

**Contribución a la Evaluación y Diseño de
Protocolos Broadcast para Redes LAN
Ethernet y MANET**

Francesc Burrull i Mestres

Junio 2005

A l'Ann

To Ann

Agradecimientos

Este apartado, el último en ser escrito, suele ser donde abundan los errores por omisión. Uno está cansado, es ya casi el final, falta poco para leer, etc.

Dicho lo anterior, agradezco en primer lugar la paciencia que mi mujer e hijos han demostrado, porque no es fácil ver como papá vuelve del trabajo y se encierra en la habitación a trabajar más.

Como no, agradezco el buen ambiente de trabajo que se respira en el Área de Ingeniería Telemática. En esta línea agradezco a Joan y Curro, que me engañaran para venirme a Cartagena, y haber tenido así la oportunidad de participar en la creación de la Escuela de Telecomunicaciones. Agradezco los problemas técnicos de Curro por abrir caminos de Investigación, por haber generado ideas, y el haber estado codo con codo a su lado. Agradezco a Josemaria, mi director de Tesis, en primer lugar el valioso esfuerzo en el campo científico, el carácter cerebral en el análisis y discusión de ideas, y no en menor medida su amistad. De este tipo se hacen pocas.

Índice general

1. Introducción	11
1.1. Motivaciones y Objetivos	11
1.2. Distribución Capitular y Contribuciones	12
1.3. Evolución de la Tesis Doctoral	13
1.3.1. Plataforma Universal Multimedia Multiservicio	13
1.3.2. Integración en LAN: Protocolo MCDP-LAN	13
1.3.3. Movilidad en LAN: W2LAN	14
1.4. Entorno LAN y Estado del Arte	16
1.4.1. IEEE 802.3	17
1.4.2. IEEE 802.11	17
1.4.3. Redes MANET	18
1.4.4. Redes de sensores	19
1.4.5. Redes Mesh	19
1.5. Especificación formal de protocolos	20
1.5.1. Las 3 FDTs: LOTOS, Estelle y SDL	21
1.5.2. SDL	21
1.5.3. Herramientas CASE SDL	23
2. Plataforma PUMM	25
2.1. Descripción global del sistema	25
2.1.1. Distribución	27
2.1.2. Broadcast	27
2.1.3. Recepción	27
2.2. Descripción global del Set-Top-Box	27
2.3. Características del Set-Top-Box	29
2.4. Servicios del Set-Top-Box	30
3. Protocolo MCDP-LAN	33
3.1. Independencia del tráfico respecto del numero de nodos	34
3.2. Independencia del tráfico respecto de la demanda de productos	35

3.3.	Especificación del protocolo MCDP-LAN	35
3.3.1.	Servicios	35
3.3.2.	Suposiciones	36
3.3.3.	Vocabulario	36
3.3.4.	Formato	38
3.3.5.	Reglas de procedimiento	39
3.3.6.	Transmisión de contenidos multimedia	41
3.3.7.	Control de flujo y control de errores	43
4.	Protocolo W2LAN	45
4.1.	Protocolos de encaminamiento en uso	45
4.1.1.	Proactivos	46
4.1.2.	Reactivos	46
4.2.	Especificación del protocolo W2LAN	46
4.2.1.	Servicios	46
4.2.2.	Suposiciones	47
4.2.3.	Vocabulario	47
4.2.4.	Formato	48
4.2.5.	Reglas de procedimiento	50
5.	Simulación de W2LAN	65
5.1.	Análisis del Coste/Beneficio	65
5.2.	Herramienta de simulación	66
5.3.	Parámetros de simulación	70
5.3.1.	Topología en línea	70
5.3.2.	Topología en malla	71
5.3.3.	Topología aleatoria	71
5.4.	Medidas realizadas	72
5.5.	Gráficas más relevantes	73
5.5.1.	Topología en línea	73
5.5.2.	Topología en malla	75
5.5.3.	Topología aleatoria	76
6.	Aspectos teóricos de W2LAN	79
6.1.	Conceptos utilizados por W2LAN	79
6.1.1.	Modelos naturales	79
6.1.2.	Copia digital	80
6.1.3.	Anuncio, Solicitud, Datos	81
6.1.4.	Transacción W2LAN	82
6.1.5.	Conversación W2LAN	83
6.1.6.	Temporizador <i>TimerSolicitud</i>	85

<i>ÍNDICE GENERAL</i>	9
6.2. Cotas del coste por Conversación	85
6.2.1. Numero de Anuncios por Conversación	87
6.2.2. Numero de Solicitudes por Conversación	87
6.2.3. Numero de Datos por Conversación	87
7. Conclusión	89
7.1. Conclusiones Finales	89
7.2. Líneas Futuras de Investigación	89
A. Especificación SDL del protocolo W2LAN	91
B. Lista de Acrónimos	103

Capítulo 1

Introducción

1.1. Motivaciones y Objetivos

Detrás de gran parte de los avances e invenciones de la humanidad suele haber un problema. Hasta que dicho problema haya sido entendido, difícilmente podrá ser solucionado. El ámbito de la Ingeniería no es una excepción: Los problemas son atacados con el Método Científico, entendidos, y (si se puede) solucionados. El esfuerzo relativo a la comprensión de problemas es lo que se conoce por Investigación.

El caso particular de esta Tesis Doctoral no es una excepción. De manera muy resumida, la motivación ha sido la comprensión y solución a tres problemas encadenados.

La motivación inicial fue la de obtener una plataforma tipo *Set-Top-Box* de bajo coste, abierta y multiservicio para la recepción de contenidos multimedia por parte del gran público, en un intento de responder al interés de la Sociedad en disminuir la creciente brecha tecnológica entre sus diferentes sectores.

A continuación apareció una segunda motivación: distribuir dichos contenidos multimedia en una red de área local (LAN, *Local Area Network*) cumpliendo una serie de objetivos referentes a la utilización de la red. Aquí es donde aparece una contribución importante de esta Tesis Doctoral: el protocolo MCDP-LAN (*Multimedia Content Distribution Protocol over LAN*).

La tercera y última motivación fue la “migración” de la solución anterior a redes inalámbricas. En lugar de migrar el protocolo anterior se optó por una solución generalizada, que consiste en una transformación de la capa de enlace, convirtiendo desde el punto de vista de capas superiores una red Ethernet *Wireless* Ad-Hoc en una LAN convencional. El encargado de dicha transformación es el protocolo W2LAN (*Wireless to LAN*), que a mi entender es la principal contribución de esta Tesis Doctoral.

1.2. Distribución Capitular y Contribuciones

A continuación se resume el contenido de cada uno de los capítulos de esta tesis doctoral:

- **Capítulo 1:** En él se describen las motivaciones y objetivos de esta Tesis Doctoral, trazando el hilo evolutivo que interconecta los diferentes capítulos. Se hace un repaso general del estado del arte en entorno LAN, concluyendo con un razonamiento que justifica el uso de métodos formales en la especificación de protocolos.
- **Capítulo 2:** En este capítulo se introduce la Plataforma Universal Multimedia y Multiservicio (PUMM) [FMS00-1, FMS00-2]. Esta propuesta está centrada en el marco de integrar al gran público en la llamada Sociedad de la Información, a través de una plataforma de uso fácil que, mediante un aprendizaje sin esfuerzos, conduce al uso de Internet por parte del usuario final sin apenas darse cuenta.
- **Capítulo 3:** Aquí se propone un protocolo para la distribución eficiente de contenidos multimedia en redes LAN Ethernet [FBM02]. Los puntos fuertes de este protocolo son la independencia del tráfico en red respecto del número de nodos y de la demanda de contenidos por parte de los usuarios.
- **Capítulo 4:** El aumento masivo de redes 802.11 (WiFi), redes basadas en los conceptos heredados de Ethernet, conduce al florecimiento de protocolos de red para dichas redes. El protocolo W2LAN propuesto [FBM04] ataca el problema desde una perspectiva diferente: La capa W2LAN residirá entre MAC y DLC, y su uso convierte desde el punto de vista de capas superiores una red MANET 802.11 en una Ethernet LAN.
- **Capítulo 5:** El beneficio que aporta W2LAN tiene un coste. En este capítulo se analiza dicho coste, o el precio que se paga al utilizar W2LAN, mediante simulación [FBM05].
- **Capítulo 6:** Para diseñar el protocolo W2LAN se han utilizado conceptos teóricos existentes y se han tenido que pensar una serie de conceptos teóricos nuevos. Este capítulo describe cada uno de los conceptos que utiliza W2LAN, tanto los nuevos como otros ampliamente utilizados y aceptados en la Ingeniería de Protocolos.
- Finalmente, en el **capítulo 7** se enuncian las conclusiones finales y algunas de las posibles líneas de investigación futuras.

1.3. Evolución de la Tesis Doctoral

Tal como se indica en la sección 1.1, esta Tesis Doctoral consta del estudio, comprensión y solución a tres problemas encadenados: Distribución masiva de contenidos, uso eficiente en red de área local y extensión a redes MANET.

1.3.1. Plataforma Universal Multimedia Multiservicio

Un problema que aún hoy día presenta nuestra sociedad es la fractura entre la gente “tecnológicamente ilustrada” y la que no lo es. De esta manera, los avances de la Sociedad de la Información sólo son aprovechados por un pequeño sector de población, generando una fractura social creciente [JGH99].

El porqué de este problema se puede explicar de manera sencilla. En primer lugar, el usuario final suele estar interesado en los servicios (especialmente los de precio asequible o nulo), no en la tecnología que los soporta. Además, el gran mercado prefiere servicios que le sean ofrecidos en su hogar, y a poder ser sin esfuerzo.

En este marco se propone diseñar y desarrollar una Plataforma Universal Multimedia y Multiservicio (PUMM) “doméstica” [FMS00-1], consistente en un terminal de arquitectura abierta en combinación con un televisor convencional. Dicho de otro modo, un *Set-Top-Box* destinado al gran mercado de consumidores, abierto en la arquitectura hardware y en el software y servicios que soportará. El producto final tiene que ser económico y a poder ser debería usar medios de transmisión baratos o gratuitos [FMS00-2]. De este modo se propuso una configuración básica basada en una plataforma (placa base) con ranuras de expansión. En una de estas ranuras se desarrollará e insertará una placa receptora y decodificadora de datos multimedia (encapsulados en datagramas IP) que serán transportados en el sistema Datacast [DAT86] de un canal de televisión convencional. Esto proporciona al usuario un canal de bajada, ya sea aéreo o cable, con una capacidad varias veces mayor que las soluciones convencionales de acceso a Internet. Mediante el uso de encapsulamiento IP y el cifrado de datos se puede además ofrecer al usuario canales privados. Además, en prototipos con canal de retorno se puede implementar algún mecanismo de medida de QoS extremo a extremo [FCC00]. Así, la colección de servicios a ofrecer puede ser muy grande, incluyendo muchos de los que se consiguen con un ordenador personal.

1.3.2. Integración en LAN: Protocolo MCDP-LAN

El siguiente problema que aparece es la distribución de contenidos multimedia. Una vez éstos se han recibido/producido, es posible que se quieran distribuir

en un entorno SOHO (*Small Office/Home Office*), es decir, mediante red de área local. La solución clásica a este problema es el modelo cliente-servidor, pero dicho modelo no funciona si el servidor cae, e incluso disponiendo de un servidor robusto el sistema puede colapsar fácilmente cuando el número de clientes aumenta y su demandas son intensas, que es el caso del tráfico multimedia. El modelo cliente-servidor en un escenario de este estilo lleva a un servicio deficiente y un tiempo de respuesta elevado.

Así, dado un conjunto de terminales interconectados mediante un concentrador (*hub*) Ethernet, el problema es cómo distribuir contenido multimedia de manera rápida y eficiente. La solución propuesta es el protocolo MCDP-LAN (*Multimedia Content Distribution Protocol over LAN*) [FBM02].

MCDP-LAN es un protocolo broadcast [CKP01] basado en dos conceptos clave: Primero, dada la naturaleza broadcast del medio en que operan las redes LAN, las comunicaciones se implementan en modo broadcast [BAT85], utilizando directamente primitivas de nivel de enlace, con el beneficio adicional de evitar el *overhead* generado por las cabeceras de protocolos de capas superiores. El segundo concepto clave es que, debido a la similitud entre los terminales que forman la LAN, se forzará a que se comporten todos igual (*peers*, con funciones tanto de cliente como de servidor), rompiendo la asimetría clásica del modelo cliente-servidor.

La función principal del protocolo MCDP-LAN es la distribución rápida y eficiente de contenidos, que puede ser iniciada desde cualquier terminal hacia el resto de los terminales. En un momento dado, uno de los terminales de la red quizás tenga que proporcionar a su usuario un contenido generado previamente por otro terminal, y dicho contenido debe ser entregado lo antes posible. Todo esto se traduce en que todos los terminales han de disponer de una copia local del contenido generado previamente por cualquiera de los terminales, ya que a priori es desconocido cual será el terminal encargado de entregar el contenido.

1.3.3. Movilidad en LAN: W2LAN

Una solución trivial para migrar a un entorno con movilidad es substituir el *hub* Ethernet por un *hub* inalámbrico y las tarjetas Ethernet por tarjetas 802.11, pero esta solución plantea un nuevo problema: La cobertura. La solución funciona siempre que todos los terminales estén suficientemente cerca del *hub* inalámbrico.

La solución más utilizada para extender la cobertura es el encaminamiento a nivel de red utilizando o bien diversos *hubs* inalámbricos o bien dispositivos de red en modo Ad-Hoc, habiendo una gran variedad configuraciones y protocolos de encaminamiento [BMJ98] que pueden ser utilizados para este fin.

El hecho de que los terminales que usan MCDP-LAN se comporten dentro de la red como *peers* decantó la elección del modo Ad-Hoc de que disponen los dispositivos 802.11. Ahora bien, las redes MANET (*Mobile Ad-Hoc Network*) presentan el problema de la visibilidad parcial [BAK02], por lo que hay que considerar protocolos de encaminamiento de nivel de enlace (capa 2 OSI) o superior. Algunos ejemplos de protocolos de encaminamiento en uso podrían ser el protocolo AODV (*Ad Hoc On-demand Distance Vector* [CEP00]) -reactivo- y DSDV (*Destination-Sequenced Distance Vector* [CPB94]) -proactivo-. En [EMR99] se puede encontrar una comparativa extensa sobre protocolos de encaminamiento en redes MANET, observando que una característica común de estos protocolos es su complejidad.

Con todo esto en mente, y teniendo en cuenta que las redes LAN Ethernet son omnipresentes [GHE03] y sus protocolos y aplicaciones robustos y sobradamente conocidos, la idea de mantenerlos parece más que razonable. Ahora bien, hay una característica que las redes LAN tienen que las redes IEEE 802.11x MANET no tienen: la visibilidad total.

Una red inalámbrica IEEE 802.11x con un Punto de Acceso, AP (*Access Point*), no es más que un conjunto de terminales que pueden verse desde capas superiores como una red LAN Ethernet convencional, excepto que en lugar de cables hacia un *hub/switch* se utiliza un radio-enlace hacia el AP. Desde este punto de vista, un terminal fuera de cobertura del AP es equivalente a un terminal que tuviese el cable de red desconectado. En este escenario, en un momento dado un terminal o pertenece a la red o no pertenece a la red. Ahora bien, el escenario es muy diferente cuando la red inalámbrica IEEE 802.11x opera sin AP, en modo Ad-hoc. Un ejemplo sencillo sería el de la figura 1.1, donde el terminal A ve al terminal B, que a su vez ve a C, pero A no ve directamente a C.

Por construcción, el protocolo MCDP-LAN 3 explota la naturaleza broadcast de las redes Ethernet LAN, pero no puede utilizarse directamente sobre una red Ad-Hoc, porque el protocolo MCDP-LAN asume que va a operar sobre un medio broadcast con visibilidad total (como los demás protocolos Ethernet). En este punto una opción era modificar MCDP-LAN, pero una opción mucho mejor era diseñar una capa intermedia genérica que ofreciese visibilidad total entre terminales, para que MCDP-LAN (o cualquier otro protocolo Ethernet) sin modificar operase sobre “LAN” (nótese que “LAN” será en realidad una red Ad-Hoc con una capa adicional).

El protocolo W2LAN (*Wireless to LAN*) 4 es un protocolo de nivel de enlace que transforma desde el punto de vista de capas superiores una red 802.11x MANET en una red LAN Ethernet, manteniendo compatibilidad con otros protocolos ya bien establecidos y probados. Así, usando W2LAN cualquier par de

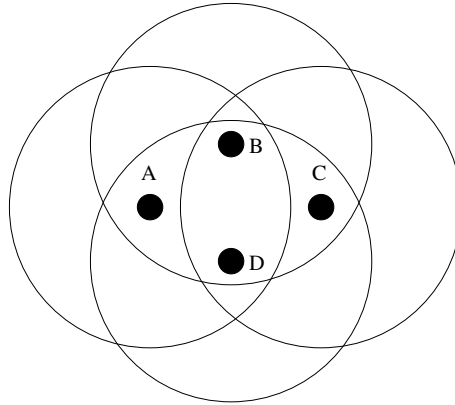


Figura 1.1: Ejemplo de una topología de red Ad-Hoc 802.11

terminales inalámbricos que pertenezcan a la misma LAN se verán entre ellos -visibilidad total-, siempre que al menos exista una ruta entre ellos (característica mínima necesaria en cualquier red).

W2LAN se basa en un modelo natural (filtrado por evolución) para diseminar información, emulando una habitación oscura con gente en su interior. Si alguien (figura 1.1, terminal A) quiere compartir una información -información 'a'-, en primer lugar se anuncia: “*¿Alguien quiere la información 'a'?*”. Los terminales que rodean la fuente (terminales B y D), si están interesados en dicha información, contestarán con la solicitud “*Sí, estoy interesado en la información 'a'*”. Entonces, si la fuente ha recibido alguna solicitud, termina la transacción entregando la información 'a'. La siguiente iteración es llevada a cabo por los alrededores de de la fuente (terminales B y D), porque a su vez se han convertido en fuentes. Obsérvese que la información 'a' se va a diseminar a todo terminal, y tarde o temprano desaparecerá por falta de interés en informaciones conocidas (nadie contestará el anuncio de C).

1.4. Entorno LAN y Estado del Arte

El entorno LAN presenta una gran variedad de soluciones tecnológicas, soluciones que suelen regirse por estándares internacionales. Por el contexto de esta Tesis Doctoral son de interés dos subconjuntos de redes de área local: las redes Ethernet, regidas por el estándar IEEE 802.3, y las redes de área local inalámbricas regidas por la norma IEEE 802.11 [IEEE802]. En función del paradigma de utilización de dichas redes se puede hablar de redes MANET, redes de sensores y redes Mesh, sobre las cuales se hará una descripción del estado del arte y esfuerzo de estandarización.

1.4.1. IEEE 802.3

IEEE 802.3 es la especificación del estándar para Ethernet, del método de comunicación física en una red de área local (*LAN, Local Area Network*), mantenida por el IEEE (*Institute of Electrical and Electronics Engineers*). En general, la norma IEEE 802.3 especifica el medio físico y las características de funcionamiento de Ethernet. El estándar Ethernet original opera a 10 Mbps y especifica los siguientes medios físicos: 10Base-2 (coaxial estrecho con longitud de segmento máxima de 185 metros), 10Base-5 (coaxial ancho con longitud de segmento máxima de 500 metros), 10Base-F (fibra óptica), 10Base-T (par trenzado telefónico), 10Base-36 (coaxial multicanal de banda ancha con longitud de segmento máxima de 3.600 metros).

La designación que IEEE hace es la siguiente: El “10”, en el campo de designación de tipo de medio, significa velocidad de transmisión de 10 Mbps. “Base” significa señalización en banda base. La “T” significa par trenzado, la “F” fibra óptica. Finalmente, el “2”, “5”, y “36” indican la longitud máxima del segmento (185 se ha redondeado a “2”, de 200).

1.4.2. IEEE 802.11

Las normas IEEE 802.11 son estándares inalámbricos *-wireless-* que especifican el interfaz aéreo entre un cliente inalámbrico y una estación base o punto de acceso, como también entre clientes inalámbricos. IEEE 802.11 especifica tanto la capa física (PHY) como la capa de acceso al medio (MAC) y están diseñadas para resolver problemas de compatibilidad entre fabricantes de equipos wireless LAN.

802.11 es una familia de especificaciones desarrolladas por un grupo de trabajo del IEEE para redes de área local inalámbricas (WLANs, *wireless LANs*). Las especificaciones de la familia son 802.11, 802.11a, 802.11b, 802.11g, 802.11e y 802.11i. Las cuatro primeras utilizan Ethernet y Acceso Múltiple con Detección de Portadora y Prevención de Colisiones (CSMA/CA, *carrier sense multiple access with collision avoidance*) para compartir el medio.

IEEE 802.11 aplica a redes inalámbricas. Éstas puedes utilizar o bien Espectro de Dispersión de Secuencia Directa (DSSS, *direct sequence spread spectrum*) o bien Espectro de Dispersión por Salto de Frecuencia (FHSS, *frequency hopping spread spectrum*). Ambos operan dentro de un intervalo de 1 a 2 Mbps.

IEEE 802.11a es una extensión de 802.11 que ofrece velocidades de hasta 54 Mbps en la banda de 5 GHz, pero lo habitual es que las comunicaciones vayan a 6, 12 o 24 Mbps. La especificación aplica a sistemas ATM inalámbricos, siendo utilizado en concentradores de acceso.

IEEE 802.11b, también conocido como Wi-Fi, se refiere a los sistemas DSSS

que operan a 1, 2; 5,5 y 11 Mbps. Todos los sistemas 802.11b cumplen con la norma de forma retrospectiva, ya que también son compatibles con 802.11 para velocidades de transmisión de datos de 1 y 2 Mbps sólo para DSSS. Los dispositivos de 802.11b logran una mayor tasa de transferencia de datos ya que utilizan una modulación diferente a la del 802.11, CCK (*complementary code keying*).

IEEE 802.11e es el primer estándar inalámbrico que cubre entorno empresarial y residencial (*home and business*). Añade características de Calidad de Servicio (QoS) y soporte multimedia a los estándares inalámbricos IEEE 802.11b y IEEE 802.11a, manteniendo compatibilidad con los mismos. El soporte de QoS y multimedia son muy importantes en redes inalámbricas en entorno residencial, por la demanda de voz, audio y vídeo existente. Los proveedores de servicios de banda ancha ven las redes domésticas con capacidades de QoS y multimedia como un ingrediente esencial para ofrecer vídeo bajo demanda, audio bajo demanda, voz sobre IP y acceso a Internet de alta velocidad.

IEEE 802.11g aplica a redes inalámbricas y provee más de 20 Mbps en la banda de 2,4 GHz. Ofrece transmisiones inalámbricas en distancias relativamente cortas a velocidades de hasta 54 Mbps. Opera en la misma banda que IEEE 802.11b, siendo de este modo compatibles.

IEEE 802.11i añade el protocolo de seguridad Estándar de Cifrado Avanzado (AES, *Advanced Encryption Standard*) al estándar 802.11. AES provee un mayor nivel de seguridad que el estándar de seguridad WI-FI Acceso Protegido.

1.4.3. Redes MANET

Las redes Ad-Hoc móviles (MANET, *Mobile Ad-Hoc Network*) son conjuntos de nodos móviles inalámbricos interconectados entre sí. Dichos nodos se pueden organizar ellos mismos, de manera automática, formando topologías arbitrarias y cambiantes. Los usuarios de las redes MANET utilizan la red como un todo, siendo transparente a ellos cualquier cambio de topología. Una característica que define a una red MANET es que no utiliza ninguna infraestructura existente, haciéndolas adecuadas para casos de desastres naturales o campo de batalla [BCG05]. Este concepto de red no es nuevo, pero la aparición de tecnologías de bajo coste, léase *Bluetooth* y IEEE 802.11, unidos al esfuerzo de estandarización del grupo de trabajo de redes MANET del IETF (*Internet Engineering Task Force, MANET Work group*) [MWG05] ha llevado a la expansión de las redes manet fuera del campo militar, pudiéndose encontrar en la literatura miles de referencias al respecto [CCL03].

1.4.4. Redes de sensores

Las redes de sensores son redes inalámbricas con una densidad alta de nodos. Dichos nodos se denominan sensores, suelen ser de tamaño reducido y bajo coste. La tarea de los sensores es recolectar información del entorno y diseminarla. Las redes de sensores permiten monitorizar y controlar entornos físicos a distancia, aplicándose en diversos campos, como monitorización del medio ambiente, aplicaciones militares y recogida de datos en zonas inhóspitas [AKY02]. Quizás sus características principales son las restricciones a las que están sujetas debido a la manera en que son instaladas y el bajo coste de los sensores: Éstos son lanzados y olvidados. Las baterías tarde o temprano se agotarán.

Los sensores, debido a su naturaleza Ad-Hoc, deben organizarse y encaminar paquetes de manera energéticamente eficiente, es decir, utilizarán algoritmos de localización y encaminamiento diseñados para minimizar el gasto energético, o la duración de la red [EGP01]. Otra característica de los sensores es que deberán ser capaces de hacer un preprocesado de los datos que recojan (suele ser una compresión de datos), de nuevo para minimizar el gasto energético, ya que el término dominante del gasto energético son las comunicaciones.

La manera en que los sensores se organizan es jerárquica, basándose en sus niveles energéticos y criterios de proximidad. Se forman *clusters*, se elige un nodo como líder del *cluster* (*cluster head*) que realizará una serie de funciones determinadas. Si falla el líder del *cluster* se reelige otro nodo.

1.4.5. Redes Mesh

La definición estricta de red Mesh es una red que utiliza uno de los dos esquemas de conexión siguientes: Topología Mesh completa o topología Mesh parcial [OWN04]. La topología Mesh completa es idéntica a la conectividad total, es decir, una red de N nodos tendrá $N(N-1)/2$ enlaces directos. En la topología Mesh parcial los nodos están conectados sólo a algunos nodos, no a todos, manteniendo siempre la conectividad completa. Esta definición no habla en ningún momento de tiempo, o dinamismo. Ahora bien, el concepto Mesh utilizado junto con redes inalámbricas suele utilizarse como sinónimo de Ad-Hoc o de red móvil. Claramente, la combinación de las características de una topología Mesh con las facilidades Ad-Hoc/MANET resulta una interesante.

Típicamente las redes Mesh consisten en una mezcla de nodos fijos y nodos móviles interconectados por enlaces inalámbricos, formando una red Ad-Hoc multisalto (*multihop*) [BCG05]. Al igual que las redes MANET, las redes Mesh heredan propiedades de las redes Ad-Hoc, por lo que quizás se puedan entender mejor describiendo algunas diferencias básicas entre redes MANET y redes Mesh. La diferencia fundamental es que las redes Mesh utilizan infraestructura

existente, que ya de por sí cubrirá una cierta zona. Esta infraestructura puede conectar la red Mesh a otras redes -típicamente Internet- por diversos puntos, y en cualquier caso ayuda al encaminamiento de paquetes dentro de la red. En este tipo de redes, los puntos de acceso intercambian información salto a salto, como si de encaminadores se tratara, y los usuarios añaden capacidad a la red. Además, los nodos móviles extenderán la cobertura de la red, haciendo también funciones de encaminamiento.

Otra diferencia importante es que las redes Mesh, a diferencia de la implementación de las redes MANET a través de simulación, desde un principio se han aplicado a casos reales. De esta manera se obtienen soluciones “suficientemente buenas” que despiertan la curiosidad del usuario y su posible adopción.

Así, el porqué de la aparición de este tipo de redes es consecuencia de cubrir unas necesidades más populares que las que cubren las redes MANET. Dichas necesidades pueden ser, por ejemplo, grupos de usuarios que quieren compartir una información, o el acceso a Internet. Así, las redes Mesh aparecen como una extensión flexible de bajo coste de redes cableadas ya existentes.

Las redes Mesh inalámbricas (MWLANs) están en proceso de estandarización por el grupo de trabajo 802.11s del IEEE 802 recientemente creado [IEE05]. Este grupo de trabajo tiene como objetivo la especificación de las capas física (PHY) y de control de acceso al medio (MAC) de las redes Mesh para mejorar la cobertura de la LAN inalámbrica (WLAN) y evitar puntos de fallo simples (*single point of failure*). También se pretende ofrecer soporte para QoS y velocidades altas (más de 100Mbps). Una extensión interesante relacionada con las redes Mesh es la llevada a cabo por el grupo de trabajo *P802.11s ESS Mesh Networking*. Se pretende extender la arquitectura y protocolos especificados por IEEE 802.11 para proporcionar un conjunto extendido de servicios (ESS, *Extended Service Set*) Mesh, como, por ejemplo, puntos de acceso capaces de establecer enlaces inalámbricos entre ellos para permitir el aprendizaje automático de topologías y configuración dinámica de rutas. La idea es extender la capa MAC de IEEE 802.11 para crear un sistema de distribución inalámbrico IEEE 802.11 que soporte tanto broadcast, multicast o unicast a nivel MAC, usando métricas sensibles al radio-enlace ejecutándose sobre topologías multisalto autoconfigurables. Este grupo de trabajo empezará a discutir propuestas para la especificación del estándar en la segunda mitad de 2005, aunque la especificación no se espera para antes de finales de 2006.

1.5. Especificación formal de protocolos

La especificación formal de protocolos es una herramienta esencial en la ingeniería de protocolos de comunicaciones. Las técnicas de descripción formal

son la base del soporte automatizado en diferentes actividades de desarrollo. Empleando las técnicas de descripción formal se pueden obtener mejoras significativas en la calidad del producto, tiempo de disponibilidad en el mercado y coste del ciclo de vida. La especificación formal permite verificar y validar protocolos de comunicaciones de manera eficiente como un paso previo al desarrollo del producto software.

Existen tres técnicas de especificación y descripción formal de protocolos aceptadas a nivel de estándar internacional, conocidas como las 3 FDTs: LOTOS (ISO 8807), Estelle (ISO 9074) y SDL (UIT Z.100). En esta Tesis Doctoral se ha utilizado SDL para la especificación de los diferentes protocolos presentados, por lo que se le hará un mayor hincapié.

1.5.1. Las 3 FDTs: LOTOS, Estelle y SDL

LOTOS (*Language of Temporal Ordering Specification* [ISO89-L]) es un lenguaje de descripción formal basado en Cálculo de Sistemas Comunicantes (CCS, *Calculus of Communicating Systems* [CCS80]), clasificable como una álgebra de procesos. La idea básica detrás de LOTOS es que los sistemas pueden ser especificados definiendo la relación temporal entre las interacciones que constituyen el comportamiento observable desde el exterior del sistema.

LOTOS es un estándar internacional de la ISO (*International Standard for Organization*, [ISO05]), el IS8807 (ISO TC97/SC21/WG21, 1989), pensado para la descripción formal de sistemas de información concurrentes y distribuidos. LOTOS posee una fuerte naturaleza algebraica, y quizás por este motivo ha quedado algo estancado en el entorno académico.

Estelle, al igual que LOTOS, es una técnica formal de descripción estandarizada por la ISO, con referencia IS9074 [ISO89-E], utilizada para describir protocolos y servicios de comunicaciones entre ordenadores. Estelle se basa en autómatas finitos extendidos.

SDL (*Specification and Description Language* [SDL05]) es un lenguaje de especificación y descripción estandarizado por la ITU (*International Telecommunications Union* [ITU05]) como Recomendación Z.100.

1.5.2. SDL

Las características principales de SDL son su versatilidad como lenguaje para usar desde la especificación de requisitos hasta la implementación, es adecuado para aplicaciones de tiempo real y sistemas estímulo-respuesta, dispone de representación gráfica, su modelo se basa en procesos comunicantes (máquinas de estado finito extendidas) y permite la descripción orientada a objetos de componentes SDL.

A pesar de que su uso en el ámbito de las telecomunicaciones esté muy extendido, también se utiliza en otras áreas tales como control de aviación y ferrocarriles, sistemas médicos, etc. De hecho, SDL es un lenguaje de propósito general para sistemas comunicantes. SDL, para describir comportamientos, se apoya en máquinas de estado finito comunicantes, representadas por procesos. Para describir comunicación se utilizan señales, que pueden ser entre procesos o comunicando el entorno del sistema con algún proceso del sistema. Algunos aspectos de la comunicación entre procesos están muy relacionados con la descripción de la arquitectura del sistema. Una máquina de estado extendida consiste en un número de estados y un número de transiciones que interconectan los estados. La máquina de estados arranca en una transición que lleva al estado inicial.

El lenguaje ha evolucionado desde la primera recomendación de 1980, con actualizaciones en 1984, 1988, 1992, 1996 y 1999. Las características orientadas a objetos se incluyeron en 1992. Finalmente se extendió a la última versión, SDL-2000, con mejor soporte a modelado orientado a objetos y generación automática de código.

El modelado orientado a objetos en SDL se potenció en la última versión para permitir su programación directa. Las características de estructura (bloques y procesos) se agruparon en agentes. También se mejoró el soporte a ASN.1 (*Abstract Syntax Notation number One* [ASN05]) para poder utilizar directamente en SDL módulos ASN.1.

Para la ingeniería de sistemas SDL suele utilizarse en combinación con otros lenguajes, como MSC (*Message Sequence Charts* [SDL05]), ASN.1 y TTCN (*Tree and Tabular Combined Notation* [TTC05]). El uso de la notación orientada a objetos de SDL-2000 en combinación de MSC, modelos de estado tradicionales SDL y ASN.1 es una combinación poderosa que cubre la mayoría de aspectos de la Ingeniería de Sistemas. Este conjunto de notaciones utiliza el mismo criterio que UML (*The Unified Modeling Language* [UML05]). Se ha avanzado también relacionando los modelos semánticos de SDL y TTCN. TTCN se usa para testear frente a estándares de validación sistemas escritos en SDL.

Todos estos lenguajes han sido estudiados por el mismo grupo de trabajo de la ITU, y consecuentemente son de interés para el *SDL Forum Society*. Los estándares Z.105(11/99) y Z.107(11/99) definen el uso de SDL con ASN.1. El estándar Z.109(11/99) define un perfil UML para SDL, y el estándar Z.120(11/99) define MSC.

Desde 1999 ha habido cambios mínimos por razones de mantenimiento de los documentos de definiciones, pero SDL-2000 apenas ha cambiado en sí. Se ha empezado una nueva revisión en 2004 tomando en cuenta la convergencia a UML2.0.

1.5.3. Herramientas CASE SDL

La industria de telecomunicaciones moderna dispone de numerosas empresas líderes en el desarrollo de software, pero debido al ritmo de aparición de nuevas tecnologías y a la constante aparición de nuevas empresas competidoras deben preocuparse de la mejora constante de sus productos y servicios [PET03]. En particular, un factor determinante del éxito de dichas empresas es el tiempo necesario en poner un producto en el mercado (*time-to-market*). Otros objetivos deseables, más tradicionales, serían la calidad del producto, la relación calidad/precio y obtener costes de desarrollo bajos [KUL03]. Bajo esta perspectiva está ampliamente aceptado que el uso de técnicas formales de especificación y descripción (FDTs, *formal description techniques*) soportadas por herramientas de ingeniería de software asistidas por ordenador (CASE, *computer-aided software engineering*) es un pre-requisito importante para alcanzar los objetivos descritos con anterioridad.

SDL, una de las FDTs de mayor éxito en los estándares de telecomunicaciones, pertenece a un conjunto de lenguajes formales estandarizados por la Unión Internacional de Telecomunicaciones (ITU, *International Telecommunications Union* [ITU05]). Este conjunto está formado por el Lenguaje de Especificación y Descripción SDL (SDL, *Specification and Description Language* [SDL05]), el lenguaje de descripción de escenarios llamado MSC (MSC, *Message Sequence Charts* [SDL05]), el lenguaje de descripción de *tests* llamado TTCN (TTCN, *Tree and Tabular Combined Notation* [TTC05]) y el lenguaje de descripción de datos ASN.1 (ASN.1, *Abstract Syntax Notation* [ASN05]).

Consecuentemente, se pueden encontrar en el mercado herramientas comerciales, utilizadas en la industria, que son capaces de analizar especificaciones SDL, hacer validaciones de especificaciones SDL basadas en algoritmos de exploración del espacio de estados, generar bancos de prueba TTCN a partir de especificaciones SDL e incluso generar implementaciones de manera automática para sistemas operativos de tiempo real [TLG05]. Se han hecho recopilaciones de estudios de caso (*case studies*) en la industria, demostrando que se obtienen costes de desarrollo inferiores, calidad mejorada y disminución de entre un 20-30 % del tiempo necesario para poner un producto al mercado, por el hecho de utilizar herramientas CASE SDL.

En SDL-Forum [SDL05] se puede encontrar una lista de fabricantes de herramientas relacionadas con SDL. En ella destaca la empresa Telelogic AB [TLG05], que dispone de abundantes relatos de éxitos (*success stories*) de uso de SDL en la industria. Dichos relatos mencionan las fases de diseño de sistema, generación automática de implementaciones y verificación formal y testeo usados por los principales líderes en la industria de las telecomunicaciones.

En el caso concreto de esta Tesis Doctoral, los diferentes protocolos presentados han sido especificados mediante la herramienta *Telelogic TAU SDL suite*, por su capacidad de validación y simulación, y quizás en un futuro permitir implementaciones de manera automática de dichos protocolos en dispositivos de tiempo real.

Capítulo 2

Plataforma PUMM

Tal como se ha introducido en 1.3.1, un problema que aún hoy día presenta nuestra sociedad es la fractura entre la gente “tecnológicamente ilustrada” y la que no lo es. En este contexto de “Sociedad de la Información” parece haber un acuerdo en el reuso de infraestructuras y redes, aceptando estándares “de facto” tales como IP, y desarrollando plataformas del tipo Set-Top-Box simples y económicas. Para conseguir de manera transparente una migración que alcance al gran público se hace necesario el esconder los aspectos tecnológicos y facilitar el uso de los dispositivos empleados.

De acuerdo con estos conceptos se ha optado por un nuevo enfoque del Set-Top-Box, un enfoque que contemple la adaptación del dispositivo al perfil del usuario, la definición del servicio proporcionado y el diseño de la plataforma hardware. Este Set-Top-Box utiliza la infraestructura existente de televisión y telefonía, dispone de una Interfaz Gráfica de Usuario (GUI, *Graphic User Interface*) adaptable, soporta Telecarga y es abierto tanto en la arquitectura como en los servicios ofrecidos. además, mediante el uso de este Set-Top-Box las diferencias entre la gente con conocimientos tecnológicos y la que no los tiene son menos traumáticas. Este hecho tiene unos beneficios no negligibles para modernizar la Sociedad.

2.1. Descripción global del sistema

El factor clave a tener en cuenta de cara al diseño del sistema es romper con los miedos y fobias que la gente “de a pié” (gente normal y corriente) puede presentar frente a las nuevas tecnologías. Para conseguir este objetivo se hace imprescindible el mirar de cerca un hogar estándar, para escoger el camino adecuado. La pregunta a hacerse es, ¿De qué sistemas de telecomunicación dispone todo el mundo en el hogar? La respuesta es televisión y teléfono. Así, el terminal

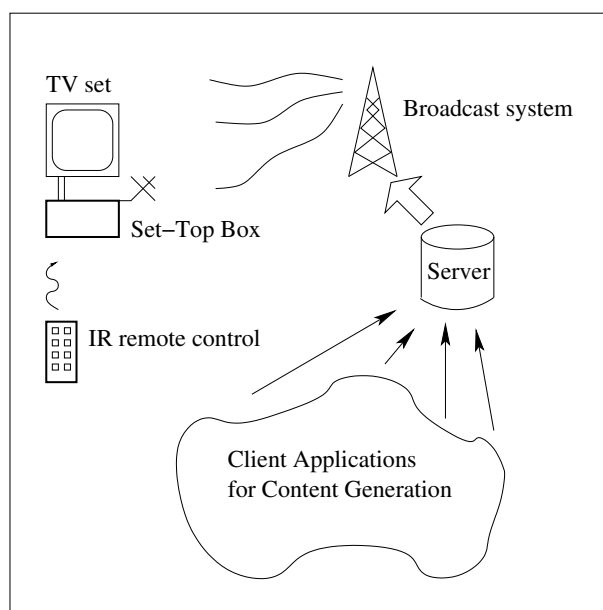


Figura 2.1: Partes de distribución, broadcast y recepción

del usuario final a desarrollar debe utilizar, como máximo, una combinación de ambos sistemas de telecomunicación actualmente desplegados de forma masiva.

Un segundo factor a tener en cuenta por parte del sistema global es la utilización de subsistemas existentes y bien conocidos. Las redes IP (un estándar “de facto”) se usarán para la distribución de contenidos hacia un servidor, responsable de la recolección y formateo de la información. Se pueden usar canales de televisión convencionales (tanto aéreo como cable) para el broadcast masivo de contenidos multimedia hacia los usuarios finales [DUT99].

Para integrar Internet con televisión se empleará encapsulado de datos IP. Así, usando datagramas IP para transportar datos multimedia sobre el sistema de Datacast, sistema ya incorporado en las señales de televisión convencionales, habilita el poder desarrollar servicios independientemente de la parte superior de la torre de protocolos que se utilice, incluso permitiendo incorporar (en prototipos con canal de retorno) mecanismos para la medida de la Calidad de Servicio (QoS) extremo a extremo.

El sistema global, por su naturaleza, se ha diseñado para ofrecer un canal de transmisión desde un centro de distribución hasta el hogar del usuario final. De este modo, el sistema se puede dividir en tres elementos funcionales diferentes: La parte de distribución, la del broadcast y la de la recepción (figura 2.1).

2.1.1. Distribución

La distribución se consigue mediante un servidor que compila información generada por aplicaciones cliente. Dichas aplicaciones cliente se pueden considerar como un sistema distribuido que genera contenidos a ser transmitidos a múltiples usuarios. El servidor incluye políticas de planificación para decidir en qué momento enviar las distintas informaciones disponibles. Esta información es formateada, comprimida y, cuando sea necesario, cifrada, para finalmente ser enviada al subsistema broadcast.

2.1.2. Broadcast

El subsistema de broadcast únicamente se encarga de modular la información y transmitirla a través del subsistema de Datacast asociado a las unidades de transmisión de televisión convencional. Usando Datacast integrado en un canal de televisión de vídeo convencional es posible transmitir hasta 204 Kbps. Esto es, 30 bytes efectivos por línea (se descartan los bytes de control y sincronismo), 34 líneas por cuadro y 25 cuadros por segundo. Utilizando un canal de televisión dedicado exclusivamente a Datacast (sin señal de vídeo) es posible transmitir hasta 3,6 Mbps (610 líneas por cuadro). Esta última cifra es prácticamente el doble de lo que se conseguiría mediante un acceso primario RDSI. Nótese que en el cálculo no se ha contemplado el *overhead* de IP y que el cálculo se ha hecho para un sistema de televisión PAL.

2.1.3. Recepción

La parte de recepción es llevada a cabo por el Set-Top-Box conectado a una antena de televisión, tal como se explica en la siguiente sección.

2.2. Descripción global del Set-Top-Box

El Set-Top-Box estará situado entre la antena de televisión y (o cable de televisión) y el televisor. Éste extrae los datos de Datacast, decodifica y descifra cuando sea necesario, almacenando los datos en memoria local. Cuando el usuario lo solicite, la información almacenada se representará de manera amigable en la pantalla del televisor.

El sistema, desde el punto de vista del usuario final, tiene que comportarse como una “caja negra” que presenta en la pantalla del televisor información potencialmente interesante. Además, el sistema debe ser muy fácil de utilizar, para así poder ser operado a través del convencional mando a distancia, dispositivo existente en cualquier hogar. El Set-Top-Box propuesto es un sistema basado en

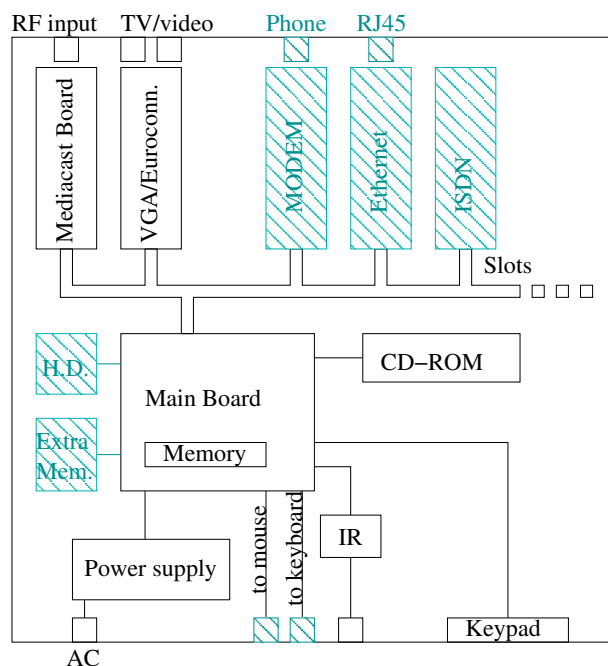


Figura 2.2: Diagrama de bloques del Set-Top-Box

una arquitectura de ordenador personal (PC) de propósito general, escondido al usuario final. Este tipo de arquitectura permite diferentes configuraciones del sistema, ajustadas a las diferentes necesidades que presenten los usuarios. Otra ventaja de esta arquitectura es el coste, en comparación con hardware específico.

La configuración básica consiste en una placa base con varias ranuras de expansión (*slots*) y una tarjeta Mediacast, que ha sido especialmente diseñada y desarrollada en el Departamento de Tecnologías de la Información y las Comunicaciones de la Universidad Politécnica de Cartagena [TIC05] para recibir y extraer datos multimedia encapsulados en datagramas IP, transportados sobre un canal Datacast de televisión convencional. Esta placa (Mediacast) es también responsable del mecanismo de telecarga. Los datos multimedia son presentados en pantalla como cualquier otro canal de televisión. Los usuarios más avanzados pueden seleccionar mediante un navegador, herramienta perfectamente conocida por el usuario de Internet, consiguiendo así algo de interactividad con sólo el canal de bajada (sin canal de retorno), debido a que la información multimedia solicitada puede estar ya almacenada en el sistema. Con esta configuración la información fluye hacia el usuario, que a su vez juega un papel pasivo en las actividades relacionadas con la conexión a red. El usuario puede visualizar la información descargada mediante el navegador, y debido a que la información es local, la velocidad de navegación es aparentemente instantánea.



Figura 2.3: Prototipo de *Set-Top-Box*

Una pequeña mejora de la configuración básica consiste en la incorporación de un acceso a Internet (modem, cable-modem, ADSL). Esta configuración permite el acceso convencional a Internet con canal de retorno, pero añadiendo el beneficio de un canal de bajada suplementario prácticamente gratuito. Debe remarcarse que esta configuración se aproxima bastante al concepto del PC convencional conectado mediante un módem a Internet, por lo que esta configuración puede ser utilizada tanto por gente ya experimentada en el uso de Internet como por gente que no lo está. En definitiva, mediante el progreso desde la configuración básica hacia la configuración con módem, la gente termina siendo usuaria de Internet sin darse cuenta, de una manera progresiva y natural.

Más allá de estas dos configuraciones se pueden diseñar fácilmente otras: La arquitectura abierta permite el añadir discos duros, lectores de DVD y CDROM, etc. El sistema operativo del Set-Top-Box se ha derivado de Linux [LIN05], que es de código abierto y las aplicaciones se desarrollan en Java, por la portabilidad e independencia de plataforma [JAV05]. De manera resumida, el sistema por completo es abierto, escalable, expandible, de bajo coste y fácilmente adaptable a futuras mejoras de hardware y nuevos servicios.

2.3. Características del Set-Top-Box

El Set-Top-Box se basa en el prototipo de la figura 2.3. A continuación se describen las características más remarcables del Set-Top-Box presentado, que son:

- **GUI adaptable al usuario:** A través de una Interfaz Gráfica de Usuario (GUI) que se inicia al nivel más principiante, los diferentes usuarios pueden adaptar el Set-Top-Box a su nivel tecnológico. El Set-Top-Box es capaz de recordar perfiles de usuario.
- **Arquitectura abierta:** Esto hace que el Set-Top-Box sea expandible en hardware, por lo que puede crecer adaptándose a nuevas necesidades, tales como televisión digital, estándares futuros, etc., solamente incorporando nuevas placas adecuadas a las nuevas funciones.
- **Multiservicio:** En contraposición con la mayoría de sistemas de tipo Set-Top-Box, dedicados o bien a televisión analógica, satélite, digital terrestre, o aplicaciones particulares (TV interactiva, acceso a Internet, telebanking, mercado de valores, etc.). En resumen, el Set-Top-Box presentado no está dedicado a ningún servicio específico.
- **Telecarga:** Ésta es una de las características más importantes. Permite la actualización de software tanto Sistema Operativo (OS) como aplicaciones. De esta manera el Set-Top-Box se adapta a los servicios emergentes. Tal y como van apareciendo nuevos servicios, de manera acorde se carga el software adecuado, sin la participación del usuario. Dicho de otra manera, los usuarios pasivos obtienen nuevos servicios sin ningún esfuerzo adicional.
- **Coste cero:** El canal de bajada es prácticamente gratuito. Utilizando la configuración con acceso a Internet, los usuarios pueden disponer de un sistema del estilo de un PC convencional con un canal de bajada adicional gratuito (la televisión convencional suele ser gratuita en la mayor parte de países) y rápido (hasta 3,6 Mbps). Cabe destacar que el canal de bajada es compartido por los diferentes usuarios.

2.4. Servicios del Set-Top-Box

Mediante la combinación de Set-Top-Box, broadcast y el uso de un canal de retorno se pueden ofrecer una gran cantidad de nuevos servicios, sólo limitados por la creatividad. Empezando por la configuración básica, se proponen los siguientes servicios:

- **Campañas Publicitarias:** Utilizando el medio broadcast, la publicidad llega a todo Set-Top-Box bajo cobertura. Esto puede ser equivalente a la recepción de panfletos en papel en el buzón de correo. Este tipo de campañas pueden substituir a las basadas en papel impreso, añadiendo funcionalidad adicional por su naturaleza multimedia, y con beneficios medioambientales.

- **Campañas Gubernamentales:** De manera parecida se puede distribuir información al contribuyente, instrucciones de voto en elecciones, información pública de interés general, etc. En general, información broadcast de las Instituciones Públicas.
- **Canales Privados:** Mediante el uso de criptografía se pueden ofrecer canales privados sobre un medio broadcast [OHT99]. Este principio permite servicios como TV-mail, que es parecido al correo convencional pero mediante soporte electrónico. Este servicio pretende ser una alternativa al correo convencional [MIV05]. El valor añadido es el ahorro de papel, corto tiempo de entrega (prácticamente instantáneo), contenidos multimedia en los documentos, amigable y fácil de utilizar. La privacidad, autenticidad e integridad se garantizan mediante el uso de criptografía. El servicio TV-mail se puede usar, por ejemplo, para enviar información bancaria a los usuarios, tal como balances de cuentas, ofertas personalizadas, etc.
- **servicios PPV:** Utilizando parcialmente técnicas criptográficas se puede proporcionar un servicio de Vídeo Club. De manera parecida se puede proporcionar música bajo demanda. Con el uso combinado de tarjeta inteligente, que puede ser leída por el Set-Top-Box, se habilitan los métodos de pago [MCC99].
- **Tele-Educación:** Otro servicio importante desde el punto de vista académico es la posibilidad de implementar servicios de tele-educación, tales como impresos de matrícula, calendarios académicos, normativas, notas, guías, ejercicios de laboratorio y notas de clase.

Capítulo 3

Protocolo MCDP-LAN

Existen varias maneras de distribuir contenidos multimedia en un entorno SOHO (*Small Office/Home Office*). La solución clásica a este problema es el modelo cliente-servidor, pero dicho modelo no funciona si el servidor cae, e incluso disponiendo de un servidor robusto el sistema puede colapsar fácilmente cuando el número de clientes aumenta y sus demandas son intensas, que es el caso del tráfico multimedia.

MCDP-LAN (ver 1.3.2) es un protocolo broadcast basado en dos conceptos clave: Primero, dada la naturaleza broadcast del medio en que operan las redes LAN, las comunicaciones se implementan en modo broadcast, utilizando directamente primitivas de nivel de enlace, con el beneficio adicional de evitar el *overhead* generado por las cabeceras de protocolos de capas superiores. El segundo concepto clave es que, debido a la similitud entre los terminales que forman la LAN, se forzará a que se comporten todos igual (*peers*, con funciones tanto de cliente como de servidor), rompiendo la asimetría clásica del modelo cliente-servidor.

La función principal del protocolo MCDP-LAN es la distribución rápida y eficiente de contenidos, que puede ser inicializada desde cualquier terminal hacia el resto de los terminales. En un momento dado, uno de los terminales de la red quizás tenga que proporcionar a su usuario un contenido generado previamente por otro terminal, y dicho contenido debe ser entregado lo antes posible. Todo esto se traduce en que todos los terminales han de disponer de una copia local del contenido generado previamente por cualquiera de los terminales, ya que a priori es desconocido cual será el terminal encargado de entregar el contenido.

Así, el proceso seguido desde que se genera un contenido hasta que un usuario lo obtiene queda separado en 3 fases: la generación propiamente dicha (la introducción en un terminal de un contenido), la distribución de dicho contenido al resto de los terminales (MCDP-LAN) y la obtención de una copia local por

parte del usuario.

3.1. Independencia del tráfico respecto del número de nodos

Una característica deseable en cualquier sistema es la escalabilidad. Siguiendo el modelo de distribución de contenidos utilizado por MCDP-LAN se puede observar que, si todo va bien -ausencia de errores-, el único momento en que se utiliza el medio compartido (LAN) es justamente después de la aparición de un nuevo contenido en algún terminal de la red. Es decir, el tráfico viene dado directa y exclusivamente por las llegadas de nuevos contenidos.

De esta manera, el hecho de tener el tráfico en la red acotado de manera proporcional al tráfico generado por las fuentes -entradas-, se puede utilizar directamente el volumen de contenidos a entregar para el dimensionado del sistema.

La fase de distribución de contenidos (MCDP-LAN) se ejecuta cada vez que un contenido nuevo aparece en el sistema, utilizando el número necesario de tramas broadcast por contenido multimedia (en función de su tamaño). Al terminar una ejecución MCDP-LAN, el resto de terminales en el sistema (sea el número que sea) disponen de una copia local del contenido multimedia.

Se ha visto el porqué de la independencia del tráfico respecto del número de nodos en ausencia de errores -hay que remarcar que éste es el caso habitual, debido a la robustez de la tecnología Ethernet sobre el que opera el sistema-. Ahora bien, a la que se producen errores de transmisión, el protocolo MCDP-LAN debe realizar alguna retransmisión de alguna trama a algún terminal. Cuando esto sucede, el resto de los terminales están forzados a esperar hasta que la trama esperada les llegue. En esta situación se pierde la independencia del tráfico respecto del número de nodos, ya que cuanto más nodos haya, más fácil es que alguno falle. Ahora bien, si un terminal falla, después de un cierto número de reintentos se le ignora -el terminal se auto-desactiva-, recuperando el resto del sistema el funcionamiento normal. Este tipo de tratamiento sería ineficaz si se utilizara en un entorno con fallos esporádicos o transitorios, pero en el caso de Ethernet se puede suponer que el enlace o funciona bien o no funciona.

3.2. Independencia del tráfico respecto de la demanda de productos

En un entorno ideal -ausencia de errores-, utilizando el modelo Cliente-Servidor aparece una relación directa entre la demanda de productos y el tráfico: Cada demanda implica una transferencia de un contenido desde el servidor hacia el cliente. Al usar MCDP-LAN los contenidos ya están distribuidos en todos los terminales, por lo que la demanda de productos no genera tráfico alguno.

El escenario de situación no ideal no es más que el de un terminal que no estaba presente en la red en el momento de la distribución de un determinado contenido que ahora le es solicitado por un usuario. Cuando esto sucede, el terminal en cuestión deberá solicitar la retransmisión del contenido a servir por parte de cualquiera de los demás terminales que lo tenga. De esta manera, el terminal que vaya a retransmitir un contenido previamente distribuido está haciendo un uso repetido de la red. Ahora bien, este tipo de situación puede etiquetarse como transitoria, ya que a un terminal en esta situación cada vez le faltan menos contenidos (a medida que sufre esta situación), por lo que se estabilizará. Otra manera de verlo es que los terminales que se han perdido algún contenido, si alguna vez alguien se lo pide, deberán pedírselo a algún otro terminal, y en el peor de los casos será el único terminal que no disponga del contenido, coincidiendo sólo en este caso con el modelo Cliente-Servidor.

3.3. Especificación del protocolo MCDP-LAN

Para la especificación de cualquier protocolo, como mínimo hay que incluir 5 elementos [HOL91]: servicios, suposiciones, vocabulario, formato y reglas de procedimiento.

3.3.1. Servicios

El principal servicio que MCDP-LAN ofrece es un acceso distribuido rápido a contenidos multimedia a través de la réplica de contenidos en todos los terminales. Es decir, cuando un usuario interactúa con un terminal, en la mayoría de los casos el terminal ya dispondrá de una copia local del contenido solicitado, que es mejor que haber de descargar bajo demanda de un servidor el producto solicitado, con el correspondiente retardo asociado. El tener una copia local del producto solicitado se consigue replicando contenidos justo en el momento de su aparición en cualquiera de los terminales hacia el resto de los mismos. MCDP-LAN detecta en cualquier terminal la aparición de nuevo contenido, asegurando su distribución hacia el resto de los terminales.

Otro servicio que MCDP-LAN ofrece es la compartición de servicios multimedia. Este servicio permite a cualquier terminal acceder a a cualquier servicio multimedia presente en la LAN, incrementando así las posibilidades de cualquier terminal de entregar contenidos. Compartir servicios significa que si algunos de los servicios están temporalmente inactivos, el resto de los mismos aún se pueden usar.

3.3.2. Suposiciones

La suposición principal es que en un entorno LAN la mayor parte del tiempo la comunicación entre terminales estará libre de errores. Esta suposición está basada en la robustez de las redes Ethernet actuales y su comprobación de CRC de 32 bits de las tarjetas Ethernet [JEG02]. El esquema de control de errores y el mecanismo de control de flujo implementado (subsección 3.3.7) se basan únicamente en esta suposición.

3.3.3. Vocabulario

El vocabulario, o conjunto de mensajes intercambiados, que utiliza MCDP-LAN se pueden clasificar en 2 tipos: mensajes Comando/Respuesta y mensajes de Datos. Es importante destacar que todos los mensajes son broadcast, para de esta forma hacer un mejor uso de la LAN, ya que en definitiva, ésta es la naturaleza del medio (al menos en su definición original).

Mensajes Comando/Respuesta

Los mensajes de Comando/Respuesta que se intercambian nacen, de nuevo (1.3.3), como consecuencia de observar un modelo natural sencillo, filtrado por la evolución. Supóngase una sala oscura con gente en ella. ¿Como sabe cada persona quién hay en la sala? ¿Cómo comparten información? ¿Qué sucede si alguien entra en la sala?. El tomar en consideración estas preguntas conduce a un conjunto de mensajes intercambiados, siempre teniendo en cuenta que se busca complejidad mínima y que MCDP-LAN se usará sobre plataforma Ethernet (disponibilidad alta y precio bajo). Así, destilando de la observación de las cuestiones anteriores, aparecen seis mensajes del tipo Comando/Respuesta. Este número reducido de mensajes encaja con los requisitos de simplicidad del modelo. Éstos son:

- *WichServicesAreThere*

Este mensaje es una pregunta a la red, al conjunto de terminales en funcionamiento. Aparece cuando un terminal tiene la necesidad de actualizar

su conocimiento de la red. Este tipo de mensaje va a generar respuestas por parte de los demás terminales distribuidos en la LAN.

- *IHaveServicesAndProducts*(*terminalID*, *services*, *products*)

Este mensaje se genera cada vez que un terminal experimenta un cambio de estado. Cuando este mensaje aparece en la LAN el resto de los terminales actualizan si les es necesario sus listas internas. Este mensaje necesita los siguientes parámetros: El número identificador del terminal que ha generado el mensaje, los servicios multimedia de que dispone dicho terminal (indica su capacidad de procesamiento) y los productos pendientes de entregar al usuario (indica la carga del terminal).

- *DeliverProduct*(*terminalID*, *product*)

Este mensaje indica al terminal que lo recibe (*terminalID*) que entregue el producto especificado al usuario.

- *WhoHasProduct*(*product*)

Este mensaje se genera cuando un terminal no dispone del producto especificado en el mensaje y lo necesita. Este tipo de mensaje va a generar respuestas por parte de los demás terminales distribuidos en la LAN.

- *IHaveProduct*(*terminalID*, *product*)

Este mensaje informa a la LAN que el terminal que lo produce (*terminalID*) dispone del producto especificado.

- *GiveMeProduct*(*terminalID*, *product*).

Este mensaje indica al terminal que lo recibe que debe empezar un broadcast MCDP-LAN del producto especificado en el mensaje.

Mensajes de Datos

Los mensajes de *Datos* se desglosan en tres tipos: *packet*, *ack* y *nack*. El hecho de que los mensajes *ack* y *nack* se utilicen en broadcast o unicast no representa ninguna diferencia en la utilización de la LAN, ya que ambos casos utilizan los mismos recursos, pero el hecho de utilizar broadcast en los mensajes *packet* significa que éstos se pueden distribuir transmitiendo una sola vez, con lo que se consigue así una utilización óptima de la red.

En este caso, los mensajes intercambiados también nacen de la observación de otro modelo natural, filtrado por la evolución, que consiste en una clase llena de estudiantes. El profesor debe ser consciente del ritmo de la clase (con el uso de vez en cuando de mensajes *ack*), a veces algún estudiante pide la repetición de algún concepto (usando un *nack*), y algunos no prestan atención, perdiendo los

contenidos. La observación de este modelo conduce a un conjunto de mensajes de *Datos* intercambiados, uno para hacer el broadcast de la información y los otros dos para el reconocimiento positivo y negativo. Así, los mensajes de *Datos* son:

- *Packet* (*terminalID*, *product*, *total number of packets*, *packet number*, *data*)

Este mensaje contiene el identificador del terminal emisor, nombre del producto, número total de paquetes en que este producto ha sido fragmentado, número de secuencia del paquete y finalmente datos multimedia (*payload*).

- *Ack* (*terminalID*, *product*, *packet number*)

Este mensaje es el reconocimiento positivo, de recepción correcta del paquete indicado del producto indicado por parte del terminal indicado.

- *Nack* (*terminalID*, *product*, *packet number*)

Este mensaje es el reconocimiento negativo, de recepción incorrecta del paquete indicado del producto indicado por parte del terminal indicado.

3.3.4. Formato

El protocolo MCDP-LAN presenta dos formatos de datos diferentes: Formato de mensajes Comando/Respuesta y formato de mensajes Datos.

Formato de mensajes Comando/Respuesta

El formato de los mensajes Comando/Respuesta se basa en *Strings*, debido a que la tarea de depuración resulta mucho más sencilla. Además, al tratarse de un protocolo específicamente diseñado para la transmisión de contenidos multimedia, el exceso de *overhead* no es un parámetro crítico. Un mensaje Comando/Respuesta siempre empieza por la palabra "COMMAND: ".

- COMMAND: <commandname>;

Los valores que puede tomar *commandname* son: *WhichServicesAreThere*, *IHaveServicesAndProducts*, *DeliverProduct*, *WhoHasProduct*, *IHaveProduct*, *GiveMeProduct*, *Ack* y *Nack*, siempre seguidos de punto y coma, ";".

Los siguientes campos mantienen una estructura similar, y son:

- TERMINALID: <terminalID>;
- NUMPACKET: <packet number>;

- NUMSERVICES: <number of services>;
- NUMPRODUCTS: <number of products>;
- PRODUCT: <product name>;

Estos campos sólo aparecen en los comandos donde son requeridos. Un ejemplo podría ser:

COMMAND: IHaveServicesAndProducts; TERMINALID: 2; NUMSERVICES: 3; NUMPRODUCTS: 7;

Este mensaje indicaría que el terminal con identificador 2 dispone de 3 servicios multimedia disponibles y 7 productos multimedia pendientes.

Formato de mensajes de Datos

El formato de mensajes de Datos también se basa en *Strings*, pero cabe remarcar que en este caso el *overhead* adicional que esto representa teniendo en cuenta que los datos que viajan llevan contenido multimedia es despreciable. Habrá dos formatos, más estrictos, diferentes: uno para *packet* y otro para *ack/nack*:

- PACKET: <terminalID>; PRODUCT: <product name>; TOTALPACKET: <numpacket>; NUMPACKET: <numpacket>; LENGHT: <datafield-length>; DATA: <data>;
- ACK/NACK: <terminalID>; PRODUCT: <product name>; NUMPACKET: <numpacket>;

Un ejemplo sería:

PACKET: 5; PRODUCT: productname; TOTALPACKET: 2048; NUMPACKET: 345; LENGHT: 1024;

Este mensaje lo manda el terminal 5, es parte del producto *productname* (el paquete 345 de 2048) y tiene un tamaño de datos de 1024 bytes.

Es importante destacar que gracias gracias al formato de datos escogido es posible la distribución de productos de manera concurrente, ya que todos los paquetes transmitidos llevan información de quien lo envía y nombre de producto, permitiendo así tanto a emisores como a receptores discernir entre diferentes transmisiones.

3.3.5. Reglas de procedimiento

El protocolo MCDP-LAN se rige por reglas de procedimiento diferentes para el régimen transitorio y para el régimen permanente.

Régimen transitorio

Cuando un terminal pasa de apagado a encendido intentará “descubrir” su entorno, que no es más que descubrir qué servicios hay disponibles por parte de los demás terminales (lista de servicios por terminal) y qué carga tienen los demás terminales (lista de carga por terminal). Una vez que el terminal ya dispone de toda esta información procederá a publicar su identidad, los servicios de que dispone y la carga local a la que está sometido. Para finalizar se alcanza el régimen permanente.

Hay que observar que un terminal que acabe de pasar de régimen transitorio a régimen permanente posiblemente no disponga de todos los contenidos multimedia de que disponen sus vecinos. Estos contenidos, en el caso de que le fuesen solicitados al terminal (no necesariamente lo serán) deberán ser solicitados a un vecino que disponga de dicho contenido para que lo retransmita mediante un broadcast MCDP-LAN, habiendo en este caso específico una ligera degradación de servicio.

Una alternativa para solucionar este problema es que el terminal que acaba de entrar en régimen permanente solicite a la LAN todos los contenidos de que dispone, y los que le falten obtenerlos de algún vecino mediante un broadcast MCDP-LAN. Desgraciadamente este planteo cargaría la LAN con contenido que ya ha sido distribuido, seguramente ya presente en la mayoría de terminales, y quizá dicho contenido nunca le sea solicitado al terminal que acaba de arrancar.

Régimen permanente

Tarde o temprano un terminal alcanzará el régimen permanente. Cuando esto ocurre habrá 4 posibles eventos a contemplar:

- Llegada de nuevo contenido a la LAN
- Llegada de nuevo contenido al terminal
- Una demanda de producto (ya sea local o desde otro terminal)
- Aparición de un nuevo terminal en la LAN

Cuando llega nuevo contenido a la LAN los terminales actualizan sus listas de contenidos distribuidos y empiezan una recepción de contenido multimedia acorde a la sección 3.3.6, recepción de contenidos.

Si llega un nuevo contenido a un terminal, éste empezará un broadcast de contenido multimedia, 3.3.6, broadcast de contenidos.

Si ocurre una demanda de un producto a un terminal, éste comprobará que el producto esté presente en local. Si no fuese así, se iniciaría una recepción de

contenidos (3.3.6, recepción de contenidos). Si no fuese posible entregar el producto de manera local, la solicitud sería reenviada a otro terminal que estuviese en condiciones de satisfacer la demanda.

Si aparece un nuevo terminal en la LAN se actualizan las listas de terminales (lista de servicios por terminal y lista de carga por terminal) de manera acorde.

3.3.6. Transmisión de contenidos multimedia

El núcleo del protocolo MCDP-LAN establece la manera en la que el contenido multimedia se va a distribuir en la LAN, sacando provecho de la naturaleza broadcast de la red sobre la que se apoya, evitando a su vez la sobrecarga que añaden los protocolos normalmente presentes entre la capa de aplicación y la capa de enlace.

El modelo en que se basa el protocolo MCDP-LAN se apoya en un entorno donde cada terminal será consciente de que cualquier transmisión en la LAN está usando un medio compartido, y consecuentemente los diferentes terminales intentarán minimizar su impacto en el recurso compartido LAN. Cuando sucede una distribución de contenido multimedia todos los terminales presentes participarán. Uno de los terminales ejecutará un broadcast de contenidos MCDP-LAN y el resto de terminales lo recibirán.

Broadcast de contenidos

El algoritmo de broadcast de MCDP-LAN empieza fragmentando el contenido a transmitir en paquetes consecutivos numerados. A pesar de que hay soporte para paquetes de longitud variable, se utilizan paquetes de longitud fija, todos iguales de tamaño excepto el último. Cuando se transmite el primer paquete, se crea una nueva lista vacía de terminales receptores. Cuando expira el *timeout* del primer paquete, los diferentes reconocimientos que se han recibido (correspondientes a los terminales receptores, incluyendo el propio) son añadidos a la lista y el segundo paquete es transmitido. Cuando todos los reconocimientos pertenecientes a la lista de receptores han sido recibidos se procede con el tercer paquete, y así sucesivamente. Hay que remarcar que las listas son utilizadas para adaptar el ritmo del proceso broadcast con el de los procesos receptores.

Un caso especial sucede cuando paquetes de reconocimiento se pierden, por ejemplo, del segundo paquete. En este caso el broadcast del tercer paquete continua, pero a continuación del *timeout*. Si expira el *timeout*, los terminales de la lista de los que no se ha recibido el correspondiente reconocimiento son eliminados de la lista.

Otro caso especial es cuando aparecen nuevos reconocimientos. En este caso los receptores recientemente aparecidos son añadidos a la lista, y sus re-

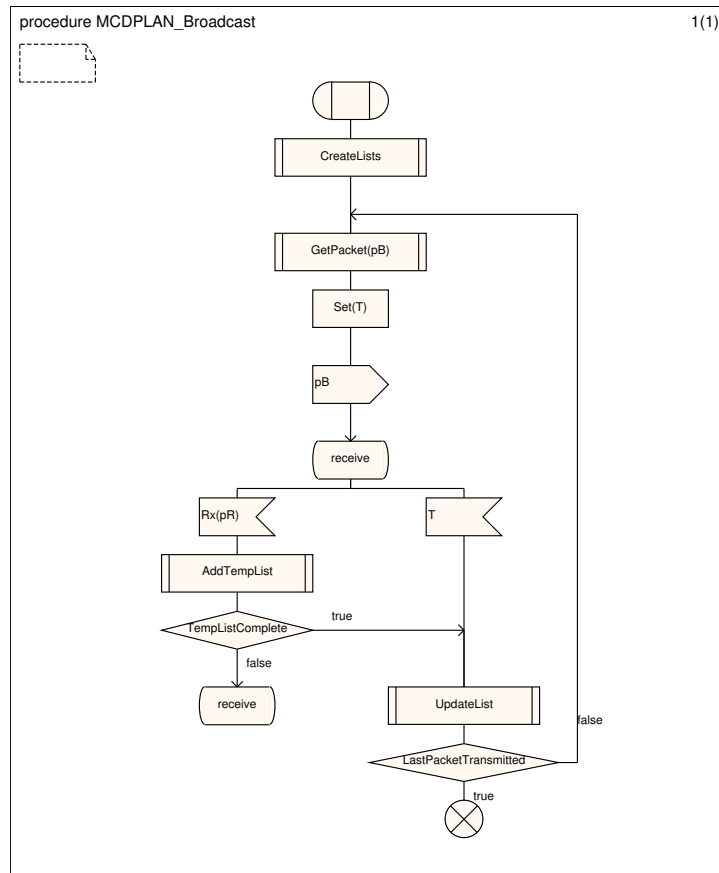


Figura 3.1: Diagrama de bloques de broadcast MCDP-LAN

conocimientos esperados para el próximo paquete transmitido, incorporándose al proceso general.

El último caso especial es la recepción de reconocimientos negativos. En este caso la lista de receptores se resetea, el terminal que ha mandado el reconocimiento negativo se añade a la lista, y el broadcast continua a partir del paquete solicitado por el reconocimiento negativo.

Resumiendo, el algoritmo broadcast del protocolo MCDP-LAN aprende que terminales están recibiendo información, actualizando de manera acorde la lista de terminales receptores.

Recepción de contenidos

La recepción de contenido multimedia simplemente consiste en enviar reconocimientos positivos cuando un paquete ha sido recibido correctamente y era esperado.

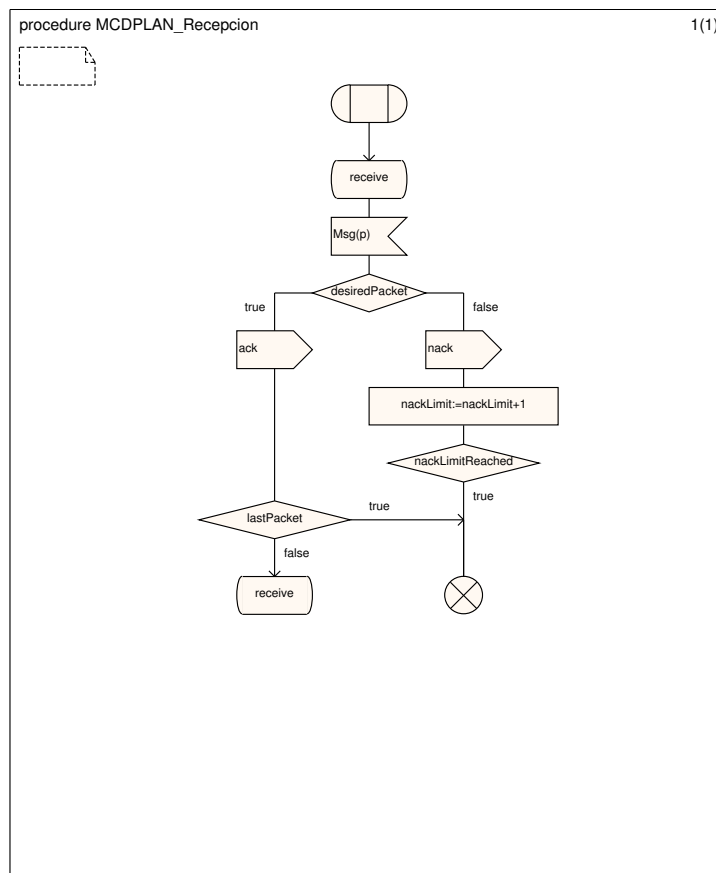


Figura 3.2: Diagrama de bloques de recepción MCDP-LAN

Un caso especial es la recepción de un paquete con un número de secuencia inesperado. En este caso se descarta el paquete y se manda un reconocimiento negativo del paquete esperado.

Si durante la transmisión de un contenido multimedia un receptor está mandando reconocimientos negativos con frecuencia, y consecuentemente, afectando negativamente a los demás terminales, automáticamente aborta la recepción del contenido multimedia, favoreciendo la recepción normal del resto de los terminales.

3.3.7. Control de flujo y control de errores

El protocolo MCDP-LAN no dispone de un mecanismo de control de errores propiamente dicho debido a las suposiciones en que se basa el protocolo, apoyándose en el mecanismo de control de flujo para la recuperación de paquetes.

El mecanismo de control de flujo de MCDP-LAN tiene ciertas similitudes al

ampliamente conocido algoritmo de parada y espera (*Stop and Wait*, [GBU96]), pero modificado [MAL02] teniendo en cuenta que las transmisiones en la LAN son broadcast. El emisor sólo manda un paquete nuevo cuando todos los receptores han reconocido el paquete anterior o expira el *timeout*. Sólo repite (en broadcast) un paquete cuando ha sido explícitamente solicitado mediante reconocimiento negativo. Es necesario destacar que el mecanismo de control de flujo del protocolo MCDP-LAN fuerza al emisor a adaptarse al ritmo global de los receptores.

Capítulo 4

Protocolo W2LAN

El protocolo W2LAN [FBM04] (*Wireless to LAN*) es un protocolo de nivel de enlace que transforma, desde el punto de vista de capas superiores, una red 802.11x MANET en una red LAN Ethernet. La elección de desarrollar W2LAN como protocolo de capa 2 en lugar de capa 3 (como ocurre en la mayoría de protocolos de encaminamiento en uso) es para dar un soporte natural a las comunicaciones broadcast/multicast. W2LAN es un protocolo reactivo, sin memoria de rutas ni posiciones.

4.1. Protocolos de encaminamiento en uso

La gran mayoría de protocolos de encaminamiento en uso [EMR99] utilizan conceptos heredados de redes fijas. No es sorprendente, debido a que inicialmente, todas las redes eran fijas. Se podría establecer un paralelismo curioso con la programación: inicialmente los programas eran secuenciales, evolucionando luego a programas concurrentes. Ahora bien, en realidad un programa secuencial no es más que un programa concurrente con un solo hilo de ejecución. Así, se podría pensar que el paradigma adecuado de diseño de una red debería suponer que la red es móvil, siendo la red fija un caso particular con velocidad de los nodos igual a cero.

Existen una gran multitud de comparativas sobre los principales protocolos de encaminamiento en uso, por lo que aquí sólo se enumeran. En [QIN04] se puede encontrar una revisión muy completa y en profundidad. Los protocolos de encaminamiento pueden clasificarse en proactivos y reactivos, en función de cómo reaccionan a los cambios del entorno.

4.1.1. Proactivos

Los protocolos proactivos son aquellos en donde los nodos buscan de manera continua información de encaminamiento, de tal manera que cuando se necesite una ruta determinada, ésta sea ya conocida. En esta categoría hay principalmente cuatro protocolos: DSDV (*Destination Sequenced Distance Vector* [CPB94]), WRP (*Wireless Routing Protocol* [MUR96]), CSGR (*Cluster Switch Gateway Routing* [CHI97]) y STAR (*Source Tree Adaptive Routing* [GAR99]).

4.1.2. Reactivos

Los protocolos reactivos, o de otra manera, los protocolos iniciados por la fuente, son ampliamente utilizados como protocolos de encaminamiento en redes Ad Hoc móviles. En este tipo de protocolos sólo se crean rutas cuando algún nodo fuente lo necesita. Cuando se necesita una ruta hacia un destino, se inicia un proceso de descubrimiento de ruta dentro de la red, que culmina cuando se ha encontrado la ruta o después de examinar todas las posibles permutaciones de rutas. Una vez descubierta y establecida, la ruta se mantiene mediante un procedimiento de mantenimiento de rutas, hasta que o bien el nodo destino se vuelve inaccesible para todo camino o la ruta deja de ser deseada. Hay bastantes protocolos que encajan en esta categoría, siendo los principales: SSR (*Signal Stability Routing* [DUB97]), DSR (*Dynamic Source Routing* [JMH03]), TORA (*Temporary Ordered Routing Algorithm* [COR95]), AODV (*Ad Hoc on Demand Distance Vector Routing* [CEP00]), RDMAR (*Relative Distance Microdiversity Routing* [AGG99]) y ZRP (*Zone Routing Protocol* [BEI02]).

El protocolo W2LAN, si hubiera de clasificarse, entraría en esta categoría, a pesar de que no utiliza información de encaminamiento.

4.2. Especificación del protocolo W2LAN

Al igual que en la sección 3.3, a continuación se especifican los 5 elementos esenciales de un protocolo: servicios, suposiciones, vocabulario, formato y reglas de procedimiento.

4.2.1. Servicios

El servicio que el protocolo W2LAN ofrece es dotar de visibilidad total a los nodos pertenecientes a una red Ad-Hoc determinada. Cuando un nodo tiene una trama Ethernet preparada para transmitir, en lugar de transmitir la trama directamente, ésta es entregada a la capa W2LAN, que de inmediato iniciará

una nueva conversación W2LAN (ver subsección 6.1.5). De manera complementaria, cuando un nodo ha recibido una conversación completa, su capa W2LAN entregará la trama Ethernet asociada a dicha conversación igual que lo haría un dispositivo Ethernet normal.

4.2.2. Suposiciones

El protocolo W2LAN se basa en 2 suposiciones:

- Opera sobre una dispositivos *wireless* Ethernet en modo Ad-Hoc

Esto significa que los nodos deben de tener al menos un dispositivo *wireless* configurado correctamente, tal como se haría en una red Ad-Hoc sin W2LAN.

- Debe existir al menos una ruta posible entre cualquier par de nodos

El protocolo sigue funcionando en redes fragmentadas, tratándolas como redes aisladas. Ahora bien, se sobreentiende que no es el caso deseado, ya que el hecho de usar W2LAN lleva implícito que se está intentando solucionar un problema de visibilidad parcial.

4.2.3. Vocabulario

El protocolo W2LAN se basa en el concepto de transacción W2LAN (subsección 6.1.4) y el de conversación W2LAN (subsección 6.1.5). Para describir el vocabulario es suficiente con desglosar las transacciones. Una transacción W2LAN consiste en una tripleta $\{announce, request, data\}$ que transportará tramas Ethernet a los distintos nodos vecinos interesados. Así, el vocabulario del protocolo W2LAN necesario para construir transacciones W2LAN será:

- *announce*

El mensaje *announce* es utilizado cuando un nodo tiene una nueva trama a transmitir, ya sea proveniente de capas superiores o proveniente de una conversación acabada de recibir que debe ser retransmitida. Este mensaje lleva asociado la dirección MAC origen del nodo, un identificador de conversación (*conversation ID*) y la cabecera de la trama Ethernet que se está distribuyendo.

- *request*

El mensaje *request* se utiliza la primera vez que un nodo recibe un *announce* de una conversación nueva. Este mensaje lleva asociado la dirección MAC destino del nodo (o de manera equivalente, la dirección MAC origen

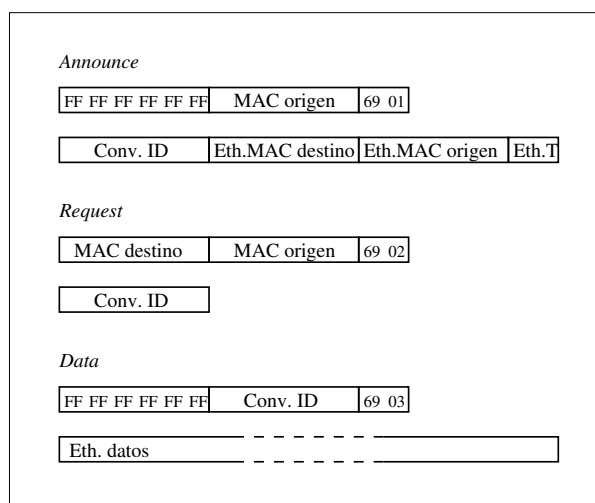


Figura 4.1: Tramas W2LAN (Cabecera/Datos)

del nodo que mandó el correspondiente *announce*) y el identificador de conversación correspondiente.

- *data*

El mensaje *data* se utiliza si un nodo que ha anunciado previamente una conversación recibe algún *request* de la conversación anunciada. Este mensaje lleva asociado el correspondiente identificador de conversación y el campo de datos de la trama Ethernet que se está distribuyendo.

4.2.4. Formato

El formato de cada mensaje (figura 4.1) se ha diseñado para que sea compatible con Ethernet, teniendo en cuenta en todo momento la simplicidad. Así, cada mensaje encaja en una trama Ethernet, evitando fragmentar los mensajes.

Los campos de cada mensaje son:

- *announce* (34 bytes)
 - Campos de la cabecera Ethernet:
 - Dirección destino (6 bytes). En las tramas *announce* siempre es la dirección de broadcast, FF FF FF FF FF FF.
 - Dirección origen (6 bytes). Corresponde con la dirección MAC del nodo que transmite.
 - Tipo de datos (2 bytes). El protocolo W2LAN utiliza para las tramas *announce* el valor 69 01.

- Campos del cuerpo Ethernet :
 - *Conversation ID* (6 bytes). Es un identificador único, que es compartido por toda trama perteneciente a una conversación determinada.
 - Dirección MAC destino de la trama Ethernet (6 bytes). Corresponde con la dirección MAC destino de la trama Ethernet que se está esparciendo.
 - Dirección MAC origen de la trama Ethernet (6 bytes). Corresponde con la dirección MAC origen de la trama Ethernet que se está esparciendo.
 - Tipo de datos de la trama Ethernet (2 bytes). Corresponde con el campo de tipo de datos de la trama Ethernet que se está esparciendo.
- *request* (20 bytes)
 - Campos de la cabecera Ethernet:
 - Dirección destino (6 bytes). En las tramas *request* corresponde con la dirección MAC del nodo que anteriormente anunció la conversación que se está llevando a cabo.
 - Dirección origen (6 bytes). Corresponde con la dirección MAC del nodo que transmite.
 - Tipo de datos (2 bytes). El protocolo W2LAN utiliza para las tramas *request* el valor 69 02.
 - Campos del cuerpo Ethernet :
 - *Conversation ID* (6 bytes). Es un identificador único, que es compartido por toda trama perteneciente a una conversación determinada.
- *data* (igual tamaño que el campo de datos Ethernet original)
 - Campos de la cabecera Ethernet:
 - Dirección destino (6 bytes). En las tramas *data* siempre es la dirección de broadcast, FF FF FF FF FF FF.
 - *Conversation ID* (6 bytes). Es un identificador único, que es compartido por toda trama perteneciente a una conversación determinada.
 - Tipo de datos (2 bytes). El protocolo W2LAN utiliza para las tramas *data* el valor 69 03.

- Campos del cuerpo Ethernet :
 - Campo de datos de la trama Ethernet (*payload*). Corresponde exactamente con el campo de datos de la trama Ethernet que se está distribuyendo.

4.2.5. Reglas de procedimiento

Para poder comprobar la corrección de las reglas de procedimiento, éstas se han expresado en el lenguaje de especificación formal SDL (*Specification and Description Language*, [SDL05]). SDL es un lenguaje de especificación formal aceptado como estándar internacional por la ITU (*International Telecommunications Union*, [ITU05]), recomendación Z101-104.

Para poder validar el protocolo de manera automática, éste se ha modelado mediante la herramienta SDL-TTCN *Suite* de Telelogic [TLG05]. Esta *suite* permite el uso combinado de MSC (*Message Sequence Charts*) con SDL directamente desde el validador. MSC es un lenguaje tanto gráfico como textual de descripción y especificación de las interacciones entre componentes de un sistema.

Previo a la descripción formal se hace una descripción informal (entiéndase como no formal), utilizando texto llano de las reglas de procedimiento, útil para un entendimiento rápido del protocolo.

Descripción informal

Desde el punto de vista de un nodo participante cualquiera, el protocolo empieza a operar cuando recibe una trama Ethernet desde capas superiores o cuando se recibe una trama W2LAN desde algún otro nodo.

En el primer caso, se construirá una trama *announce* con un nuevo *ConversationID* y se emitirá. Debido a que la cabecera Ethernet ya ha sido transmitida en la trama *announce* no hay ninguna razón para almacenarla, pero lo que sí que hay que almacenar temporalmente es el campo de datos Ethernet, debidamente identificado mediante la etiqueta *ConversationID* acabada de crear. El lugar para dicho almacenamiento es la lista de tramas pendientes, “*pending frames list*”. A continuación, el nodo debe esperar un cierto intervalo de tiempo, “*request window*”. Para permitir la concurrencia de varias conversaciones simultáneas se utilizan dos listas adicionales, la lista de temporizadores, “*timers list*”, y la lista de contadores, “*counters list*”. Un elemento que contiene el instante de caducidad del temporizador (instante actual + *request window*) se añade al final de la lista de temporizadores cada vez que se (re)activa el temporizador (que es cuando se emite *announce*). Un elemento que contiene *ConversationID* y un contador con el valor inicial cero se añade al final de la lista de contadores.

Durante el período de tiempo *request window*, si se recibe alguna trama *request* y la conversación correspondiente está en la lista de contadores, su contador se incrementará. Cuando finalice el período de tiempo *request window*, la trama pendiente se elimina de la lista de tramas pendientes y, si ha habido alguna trama *request* asociada a la trama identificada por *ConversationID*, se construirá la correspondiente trama *data* y se transmitirá.

En el caso de recibir una trama W2LAN desde otro nodo (nodo actuando como repetidor) se utilizará la lista de conversaciones, “*conversations list*”. Esta lista opera como un búfer circular sobrescribiendo los elementos más antiguos. Un elemento genérico de la lista consiste en un campo *ConversationID*, un campo Booleano que indicará si la conversación está pendiente, y una cabecera Ethernet. La trama W2LAN recibida puede ser *announce* o *data* (nótese que *request* sólo se recibe como respuesta a una transmisión *announce* previa).

Si se recibe una trama *announce* se comprueba la lista de conversaciones. Si la conversación no está en la lista se añade un nuevo elemento marcado como pendiente y se construye y emite la correspondiente trama *request*.

Si se recibe una trama *data* se comprueba la lista de conversaciones. Si la conversación está en la lista y está marcada como pendiente se construirá una trama Ethernet con la cabecera almacenada en la lista de de conversaciones y el campo de datos acabado de recibir. A continuación, sólo si la trama Ethernet reconstruida es para el nodo que la ha recibido se entregará a capas superiores, y en cualquier caso se iniciará una nueva iteración como si la trama Ethernet hubiese llegado de capas superiores.

Descripción Formal

El listado en SDL/PR (procedural) completo del sistema se incluye en el apéndice A. En este apartado se incluye un subconjunto de la descripción SDL/GR (gráfica) del proceso W2LAN que se ejecuta en un nodo. Se ha eliminado parte del código, dejando el de mayor relevancia, para obtener mayor claridad y ocupar menos espacio.

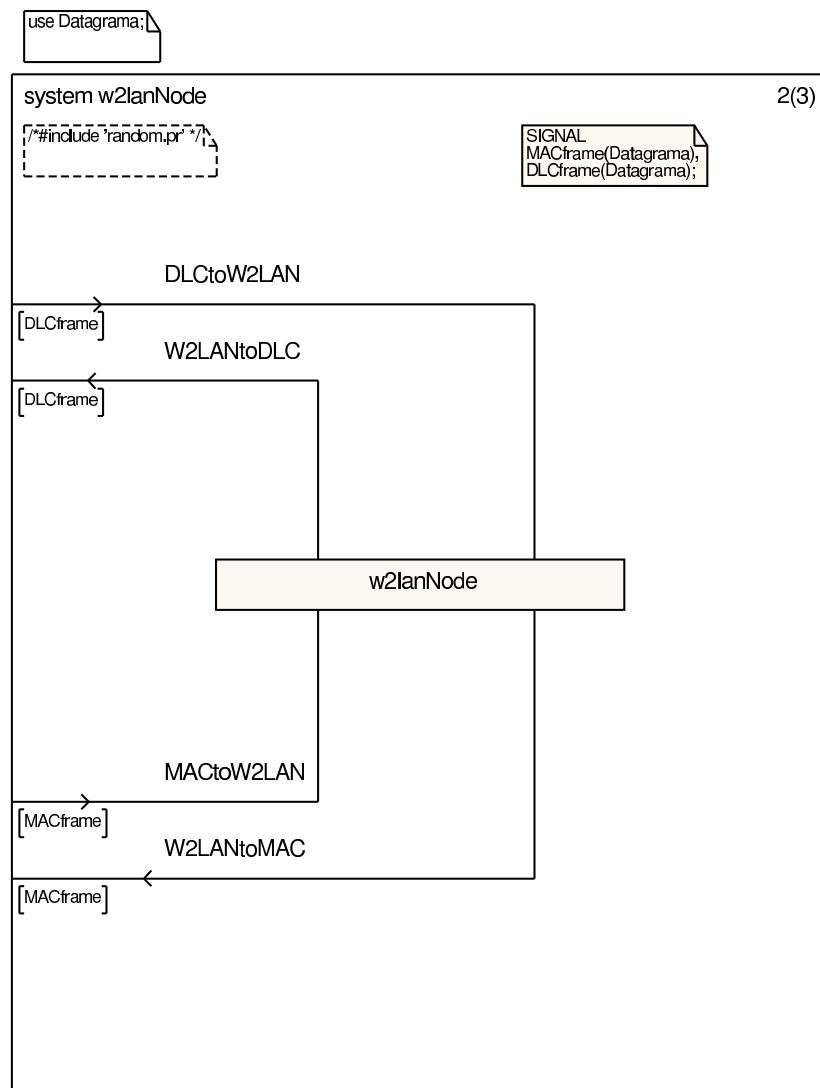


Figura 4.2: Diagrama de sistema

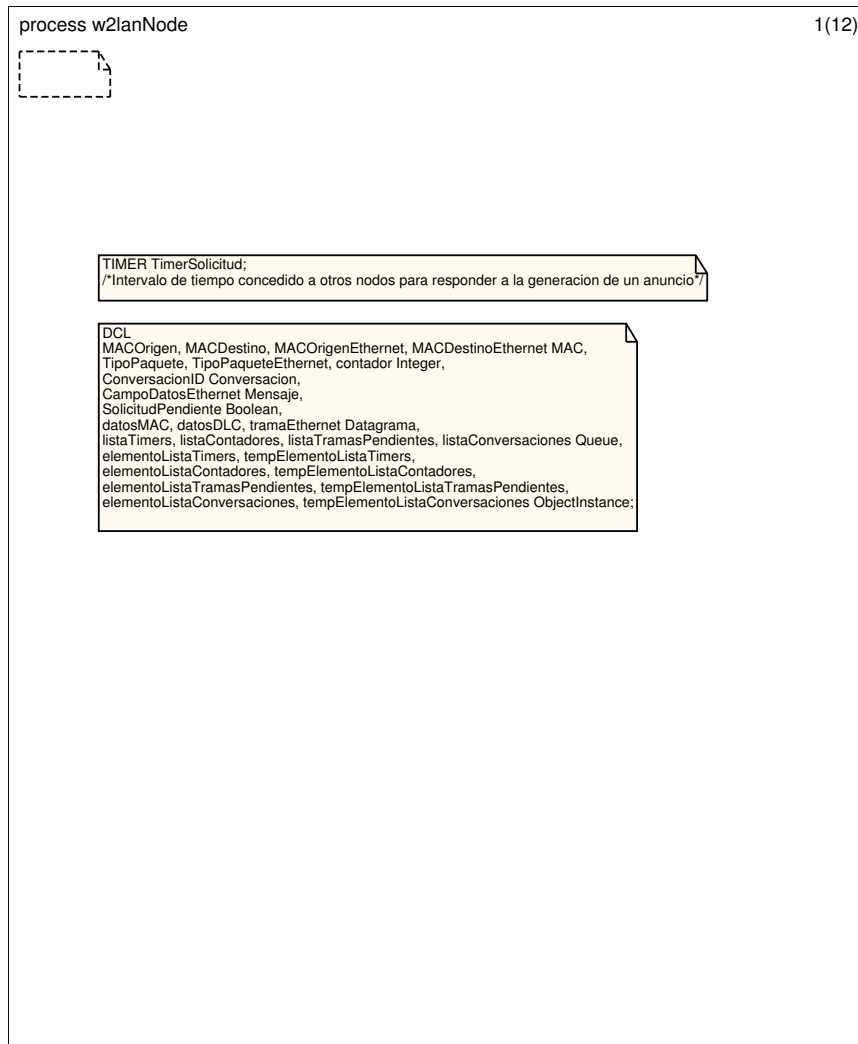


Figura 4.3: Proceso W2LAN (I)

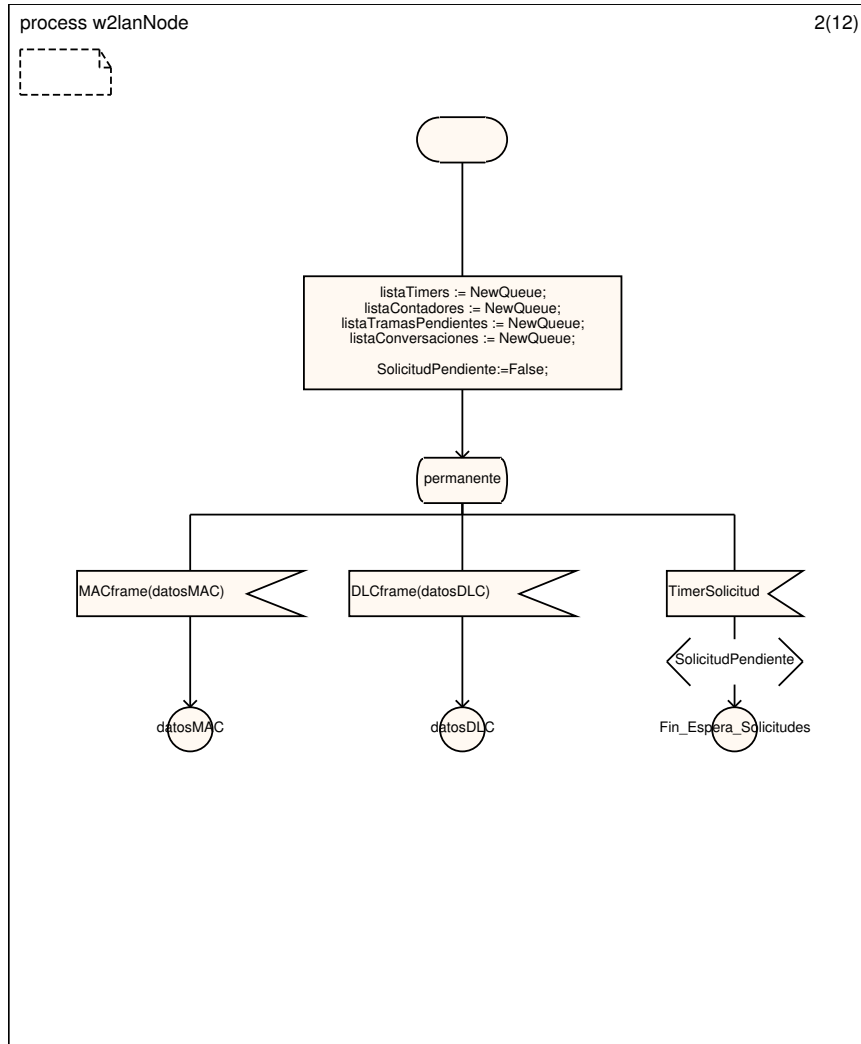


Figura 4.4: Proceso W2LAN (II)

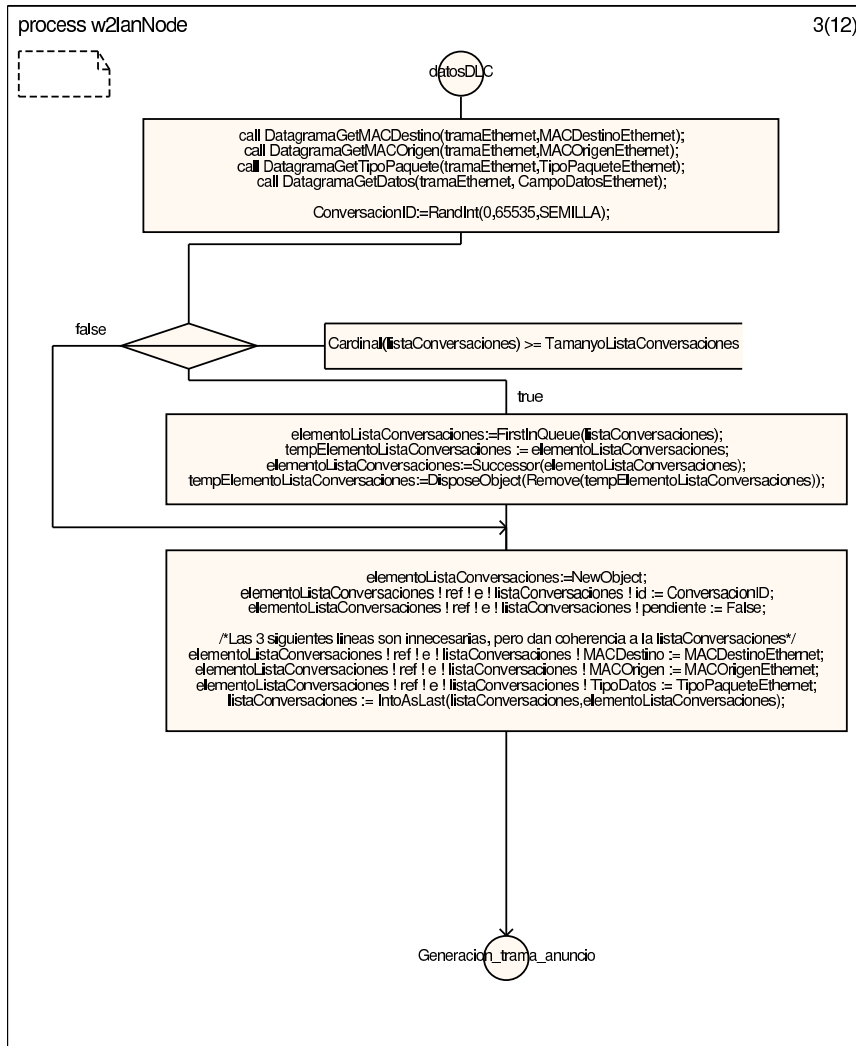


Figura 4.5: Proceso W2LAN (III)

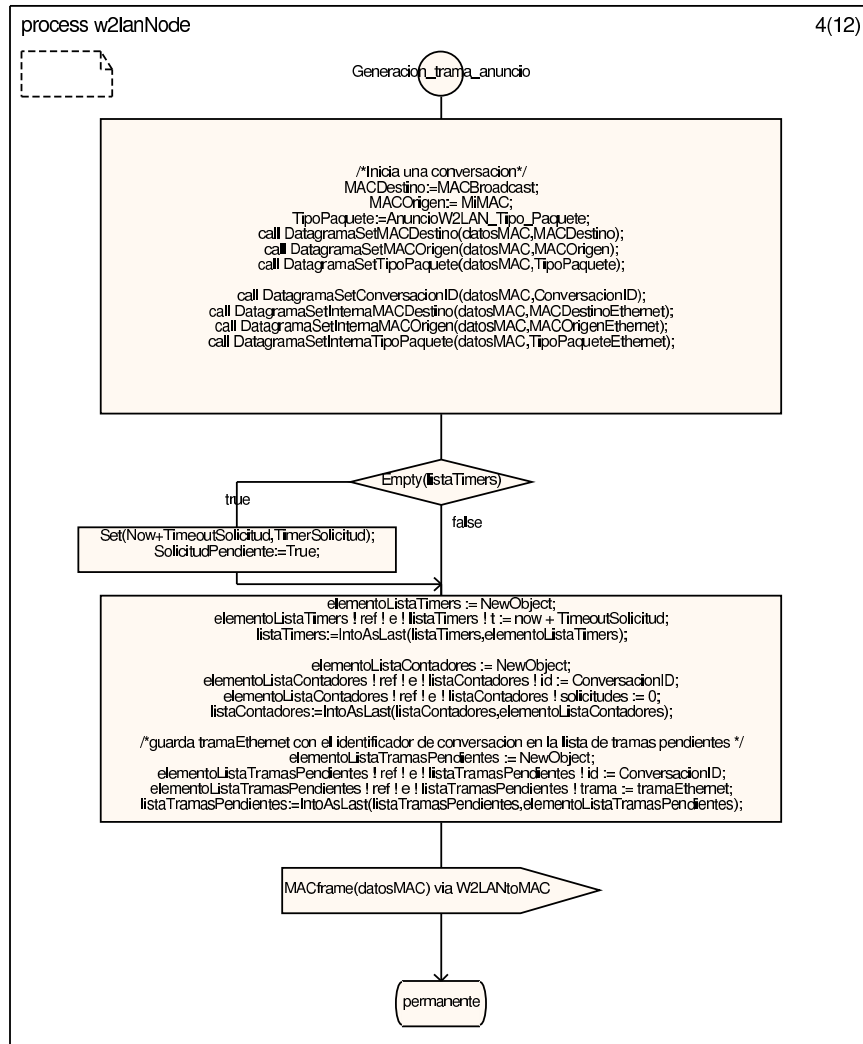


Figura 4.6: Proceso W2LAN (IV)

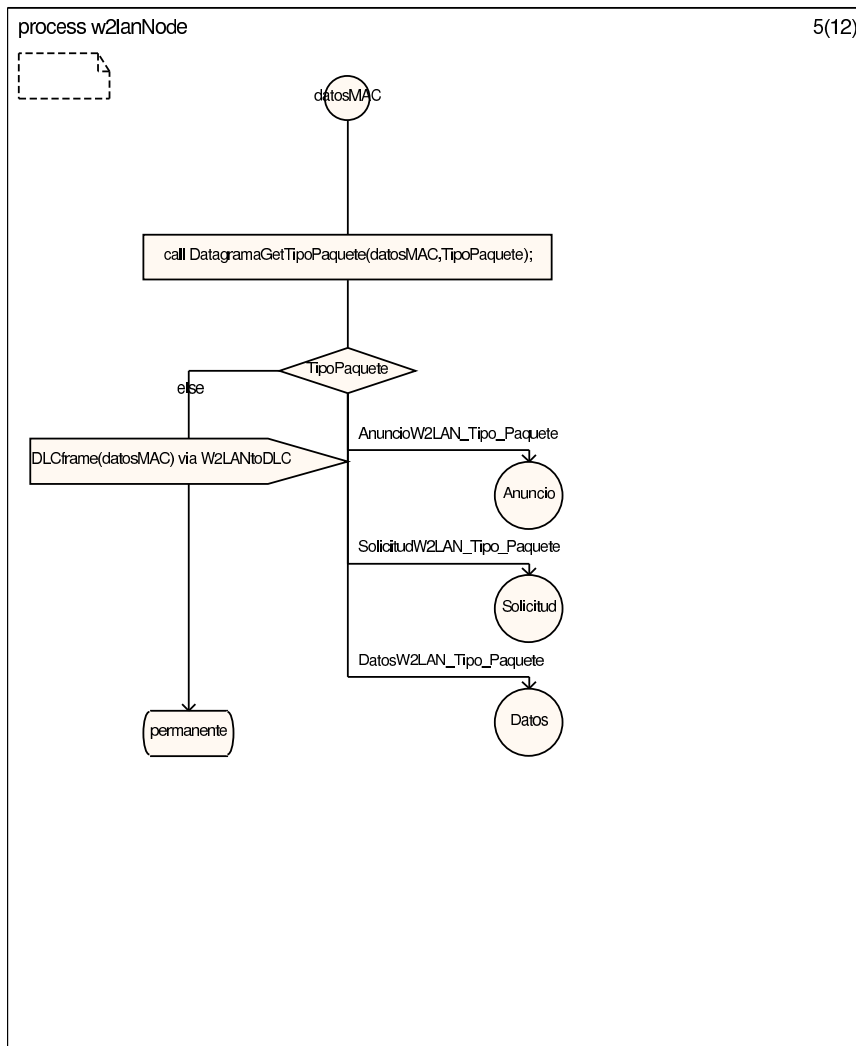


Figura 4.7: Proceso W2LAN (V)

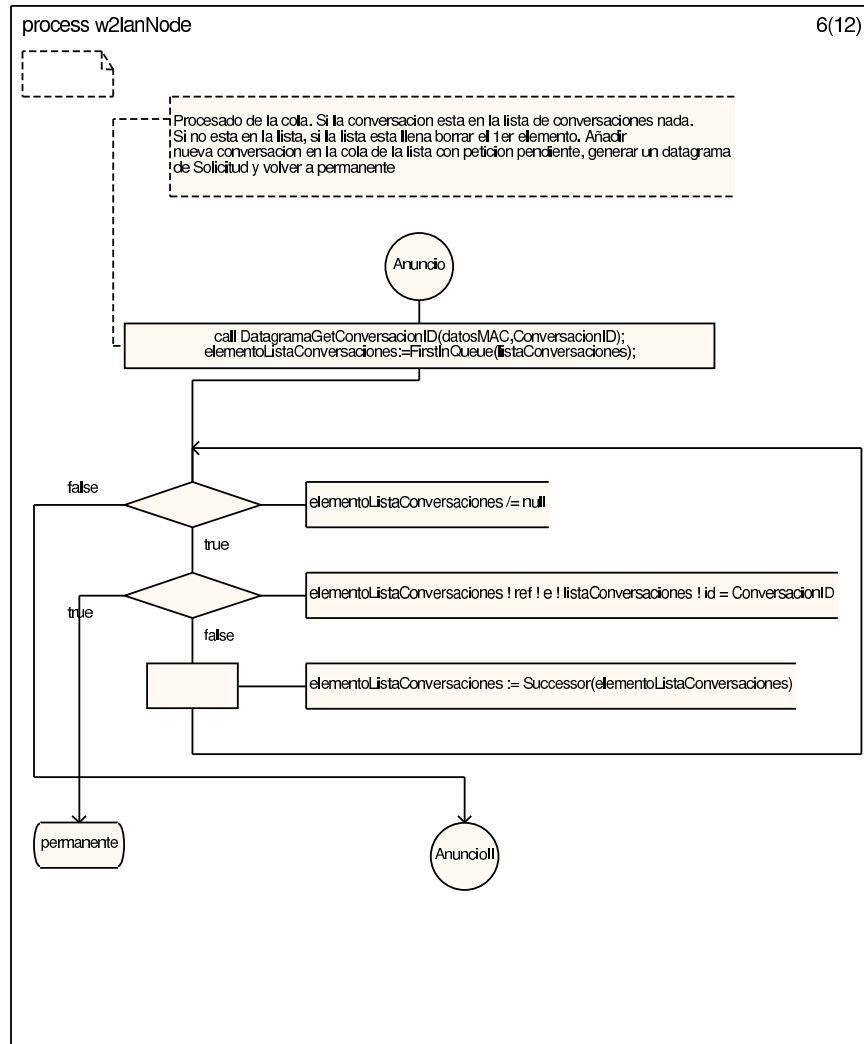


Figura 4.8: Proceso W2LAN (VI)

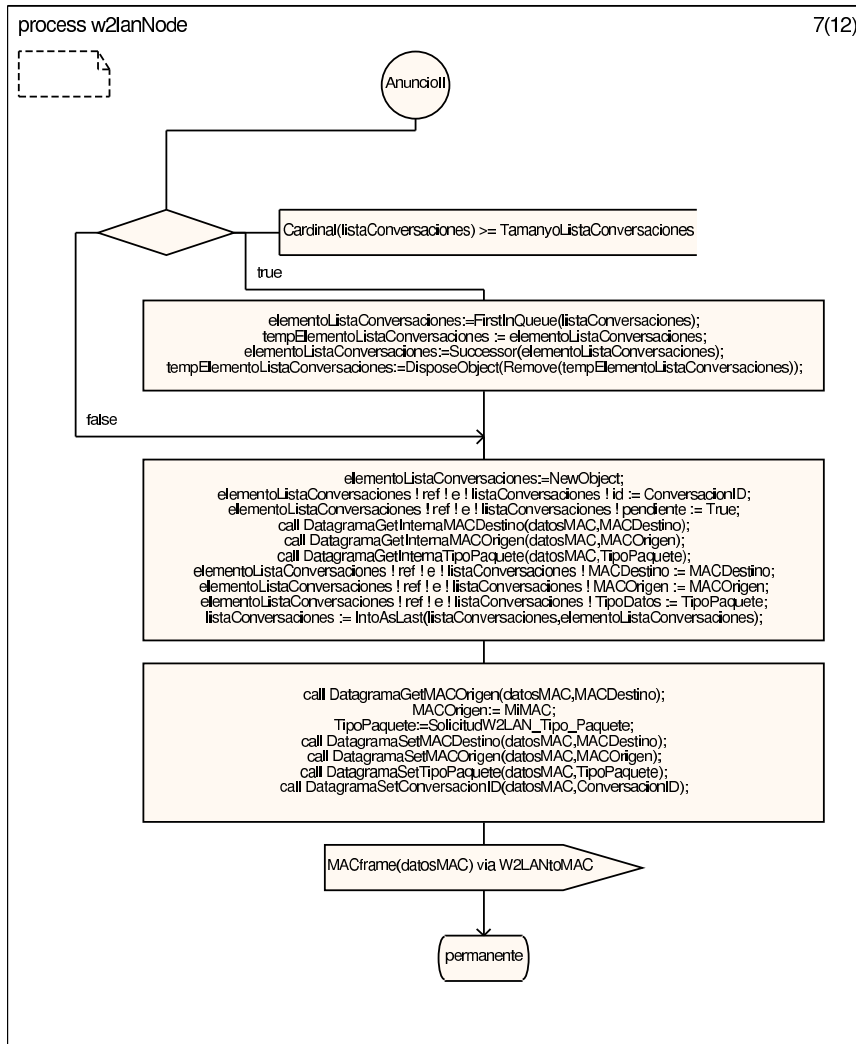


Figura 4.9: Proceso W2LAN (VII)

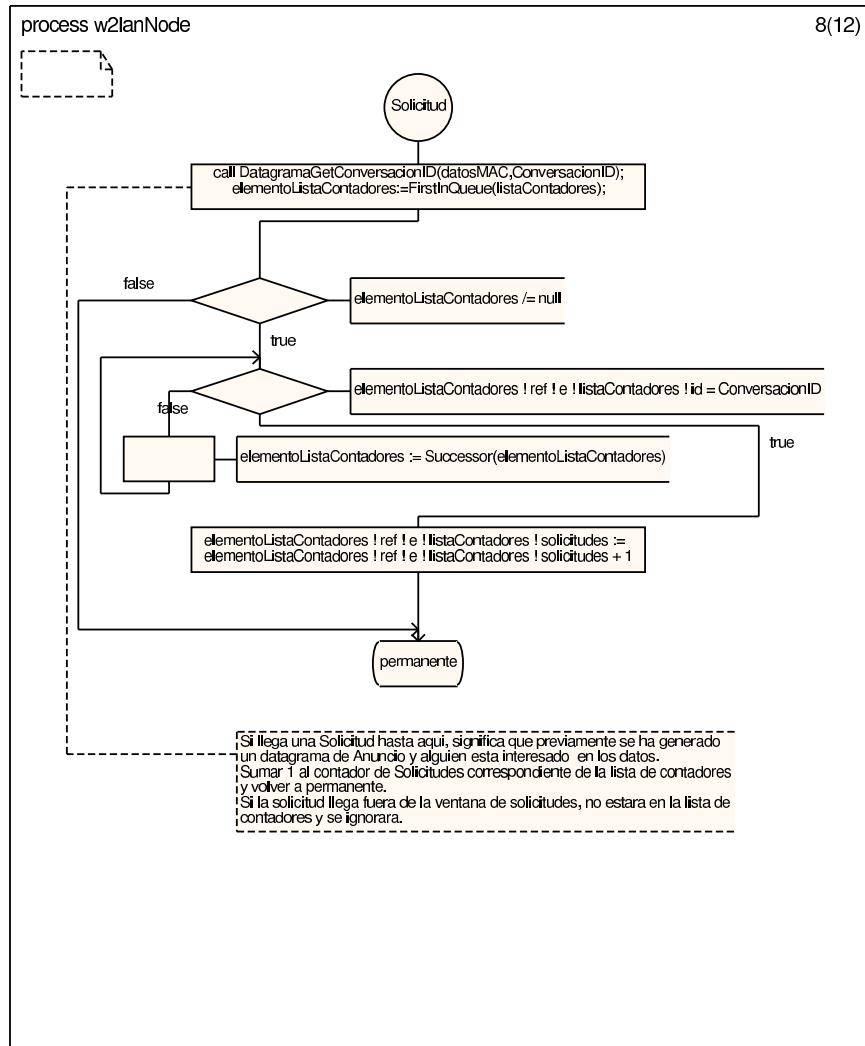


Figura 4.10: Proceso W2LAN (VIII)

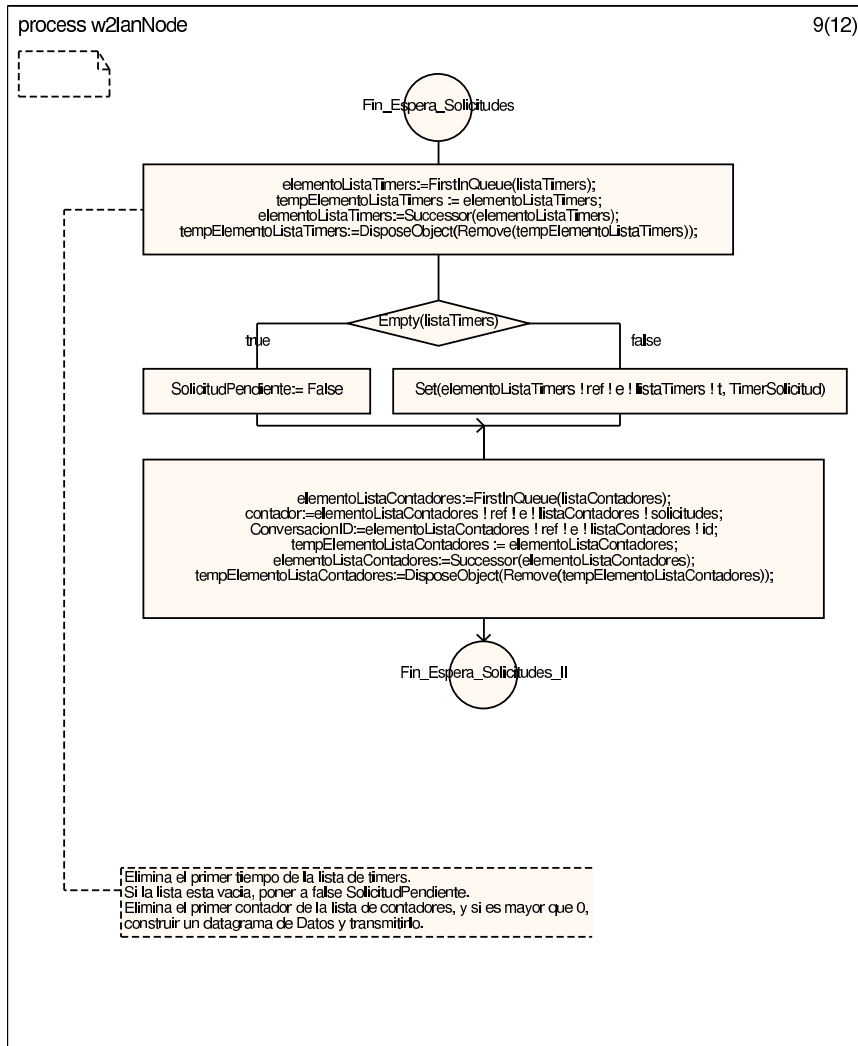


Figura 4.11: Proceso W2LAN (IX)

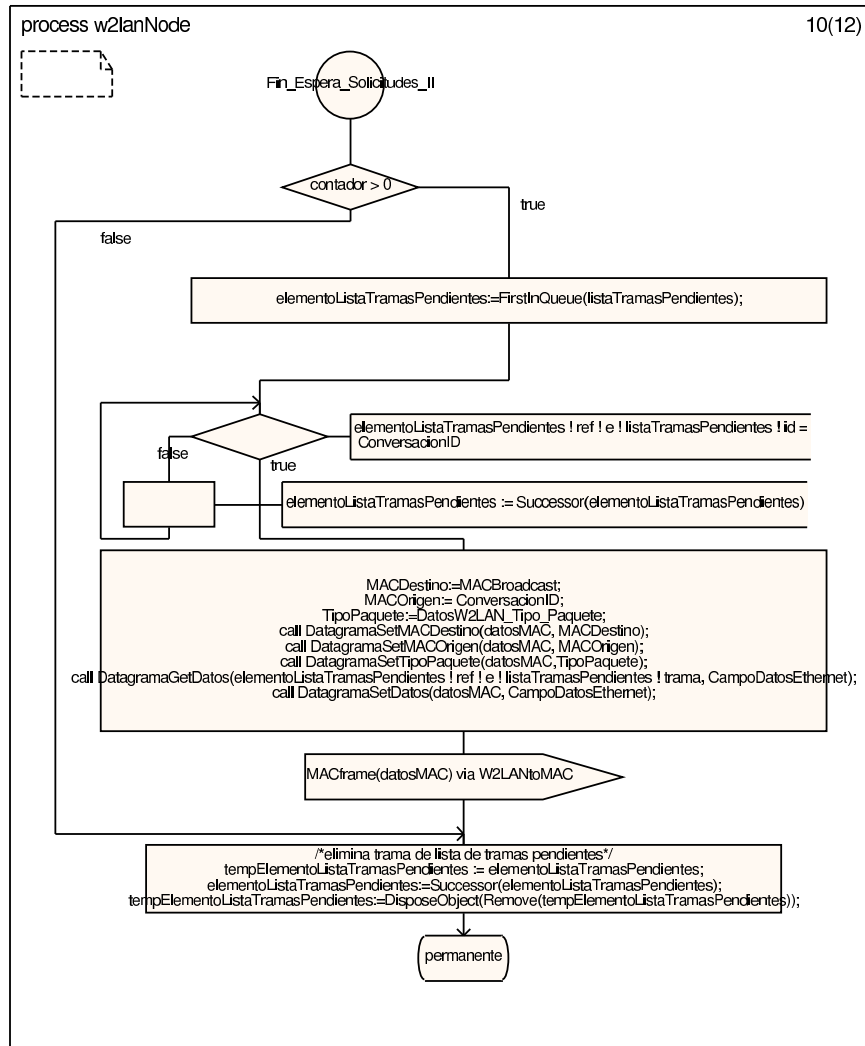


Figura 4.12: Proceso W2LAN (X)

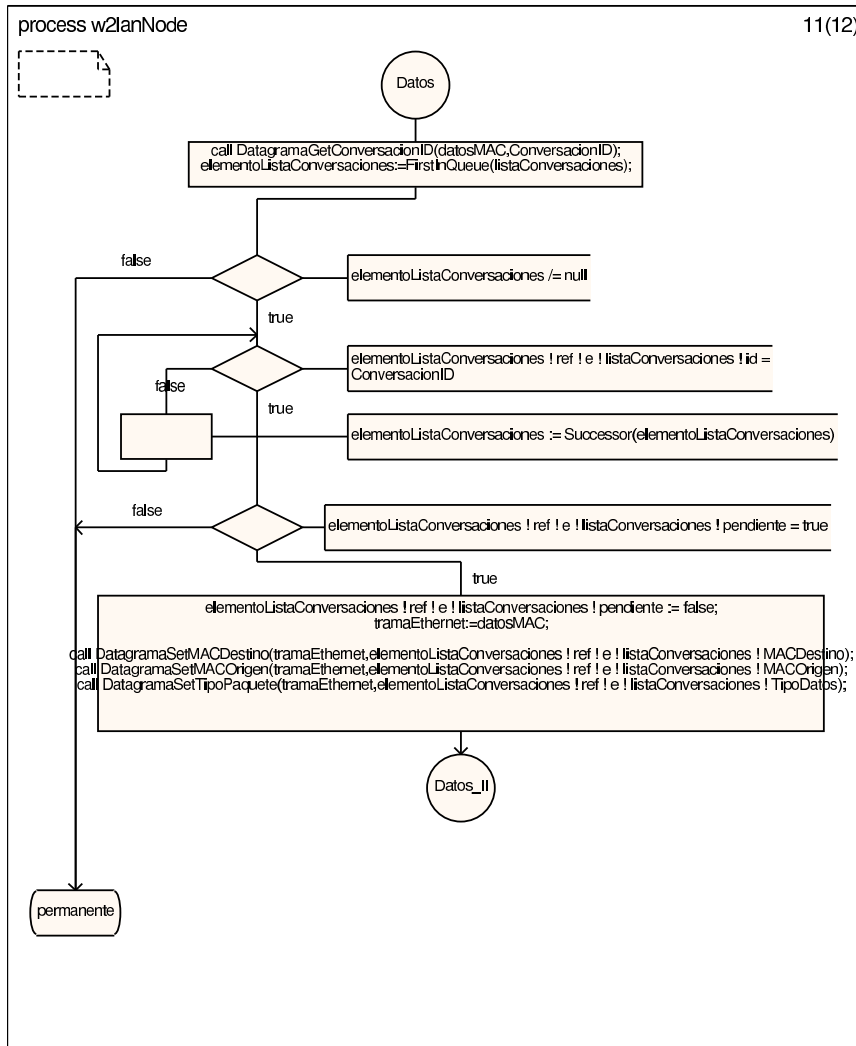


Figura 4.13: Proceso W2LAN (XI)

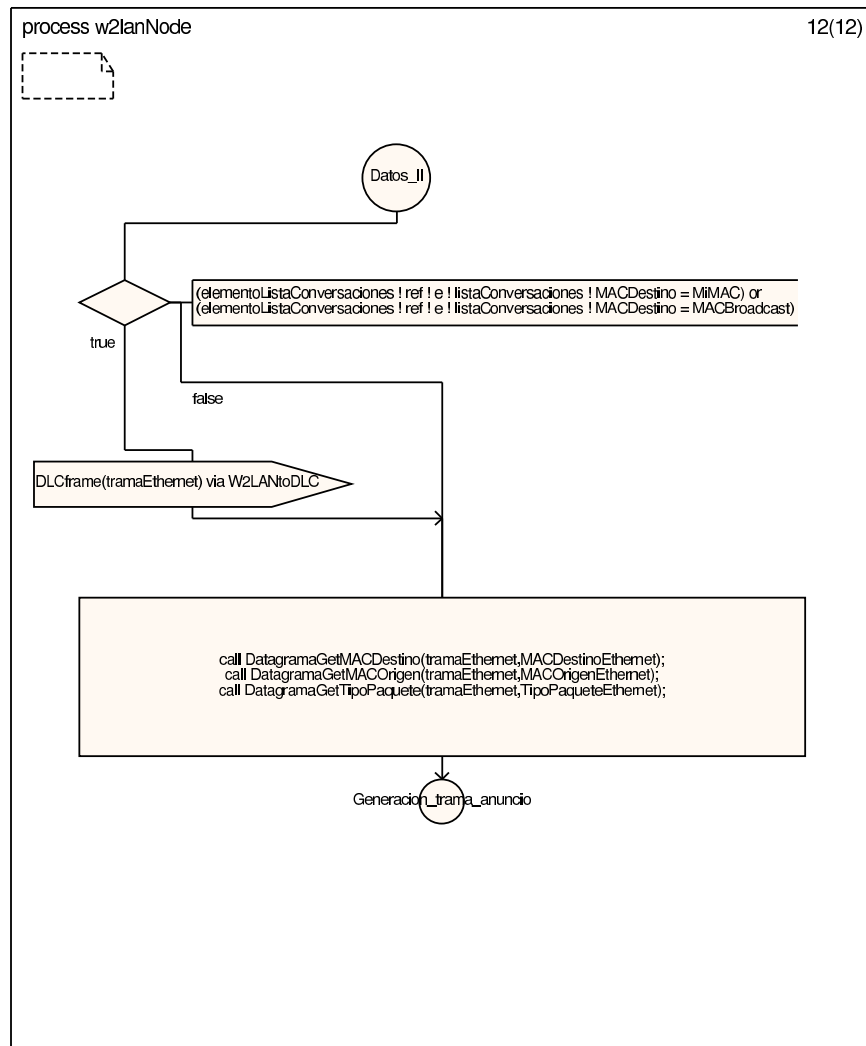


Figura 4.14: Proceso W2LAN (XII)

Capítulo 5

Simulación de W2LAN

El protocolo W2LAN (*Wireless to LAN protocol*) es un protocolo de capa 2 puro que transforma una red MANET (*Mobile Ad-Hoc network*) en una red Ethernet simple desde el punto de vista de capas superiores. El protocolo W2LAN opera como una capa situada por encima de dispositivos Ethernet, situándose entre la capa MAC (*Medium Access Control*) y la capa DLC (*Data Link Control*). El beneficio de utilizar W2LAN es que un conjunto de N nodos tendrá visibilidad total entre sí, eliminando el problema inherente de visibilidad parcial que presentan las redes MANET.

Para evaluar el protocolo W2LAN se ha construido un modelo en SDL (*Specification and Description Language*) con N nodos, y a partir del modelo SDL se ha derivado de manera automática una instancia de simulador. Finalmente se ha realizado el análisis Coste/Beneficio, que consiste en encontrar el coste en número de tramas utilizadas por el protocolo W2LAN en comparación con las utilizadas en un entorno LAN, bajo diferentes parámetros.

5.1. Análisis del Coste/Beneficio

Una manera de describir el Coste/Beneficio del protocolo W2LAN es utilizando el concepto de conversación W2LAN, es decir, el conjunto de tramas en la red que comparten el mismo identificador de conversación. En una LAN Ethernet pura, una conversación/comunicación consistiría en una trama única -desde el emisor hasta el receptor-. Ahora bien, en un entorno W2LAN una conversación consiste en un número de tramas -*Announce*, *Request* y *Data*- que eventualmente desaparecerán de la red W2LAN.

El coste de utilizar W2LAN varía dependiendo de tres parámetros: El número de nodos participantes en la red, sus radios de cobertura y la topología de los nodos.

A priori el primer parámetro, número de nodos en la red, parece que incidirá directamente en el coste: Cuanto más nodos haya, más tramas. El segundo parámetro, radio de cobertura de los nodos, se espera que tenga un impacto inversamente proporcional en el coste: A mayor cobertura, menor repetición de tramas. El tercer parámetro, topología de los nodos, añade incertidumbre, ya que existe un número ilimitado de topologías posibles.

Los principales protocolos de encaminamiento, tales como AODV (*Ad hoc on-demand distance vector* [CEP00]) y DSR (*The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks* [JMH03]), protocolos de capa 3 OSI, deben de tener mucho cuidado al considerar la movilidad de los nodos, incluso aplicando restricciones a los modelos de movilidad utilizados [CBD02], ya que la movilidad de los nodos afecta directamente el *overhead* que presentan estos protocolos. Contrariamente, las simulaciones llevadas a cabo con el protocolo W2LAN no tienen porqué tener en cuenta la movilidad de nodos. Esto es debido a que W2LAN es un protocolo de capa 2 que no trabaja ni con posiciones de nodos ni con rutas, y para crear un modelo de movilidad es necesario usar los conceptos anteriores. El modo de operación de W2LAN inunda de manera selectiva la red, de tal manera que cuando un nodo origen quiere transmitir a un nodo destino, de todas las transacciones (pertenecientes a la conversación) que ocurren en la red (algunas útiles, otras no) habrá un subconjunto de ellas que formará el camino más corto, que será el utilizado. La siguiente conversación no dispondrá de ninguna información sobre la anterior, por lo que se repetirá el proceso (sin ninguna memoria de rutas o posiciones). De este modo, los resultados de simulación son válidos y se pueden extrapolar a cualquier patrón de movilidad de nodos.

5.2. Herramienta de simulación

El modelo de N nodos ejecutando el protocolo W2LAN ha sido especificado en el lenguaje SDL (*Specification and Description Language* [SDL05]), lenguaje de descripción formal aceptado como estándar por la ITU (*International Telecommunications Union* [ITU05]), recomendación Z101-Z104. El simulador del modelo de N nodos se obtendrá de manera automática a partir de su descripción formal mediante la herramienta *suite* Telelogic TAU SDL [TLG05].

Un problema conceptual a solucionar es el de la cobertura. En un escenario W2LAN real un nodo no tiene conocimiento del entorno, por lo que cuando emita una trama, ésta será recibida por los nodos dentro de su radio de cobertura. Este escenario sigue un modelo distribuido, donde no hay un conocimiento global del sistema: El punto de vista de un nodo no tiene porque coincidir con el sistema. Para solucionar este problema se han realizado 2 modificaciones: Se ha dotado

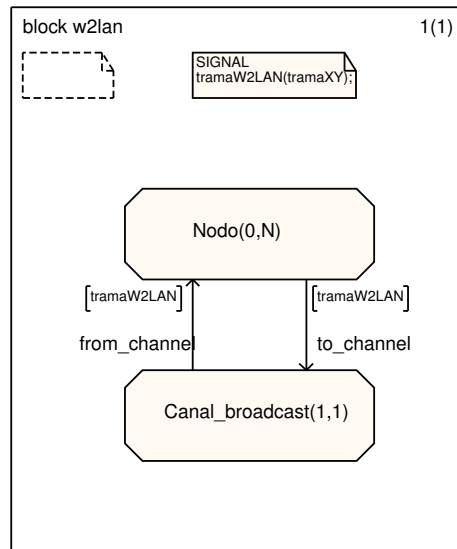


Figura 5.1: bloque W2LAN (un canal broadcast, N nodos)

a los nodos con información de su propia posición, y se ha añadido un campo en el formato de trama que incluye la posición del emisor. De esta manera, un nodo receptor sabrá la posición del emisor y podrá decidir para cada trama si está dentro de cobertura o no. Éstas 2 modificaciones afectan sólo al modelo de simulación: El protocolo W2LAN no tiene conocimiento de posiciones.

Un segundo problema (menor) es una limitación técnica, que no es más que es mucho más sencillo implementar comunicaciones punto a punto que comunicaciones multipunto. Así, si se quieren interconectar N nodos se necesitan $N \times N$ conexiones.

Para realizar simulaciones es útil disponer de toda la información del sistema de manera centralizada (visión global), porque de este modo resulta más sencillo entender el sistema en su totalidad y se tiene mayor conocimiento del mismo. Así, el modelo implementado para el simulador consiste en un canal broadcast, implementado como proceso padre, que generará N procesos hijo, los procesos nodo (ver figuras 5.1 y 5.2). Las comunicaciones serán desde un nodo al canal broadcast, y éste realizará N copias por trama, enviando una a cada nodo (ver figura 5.3). De esta manera el número de conexiones para interconectar N nodos es N . El hecho de utilizar este modelo centralizado (estrella de N puntas) hace posible tener información global del sistema desde el proceso canal broadcast.

El simulador se obtendrá de manera automática, y una vez obtenido se ejecuta reiteradamente con diferentes parámetros de entrada mediante el uso de *scripts*. Los resultados de las distintas simulaciones se van almacenando en

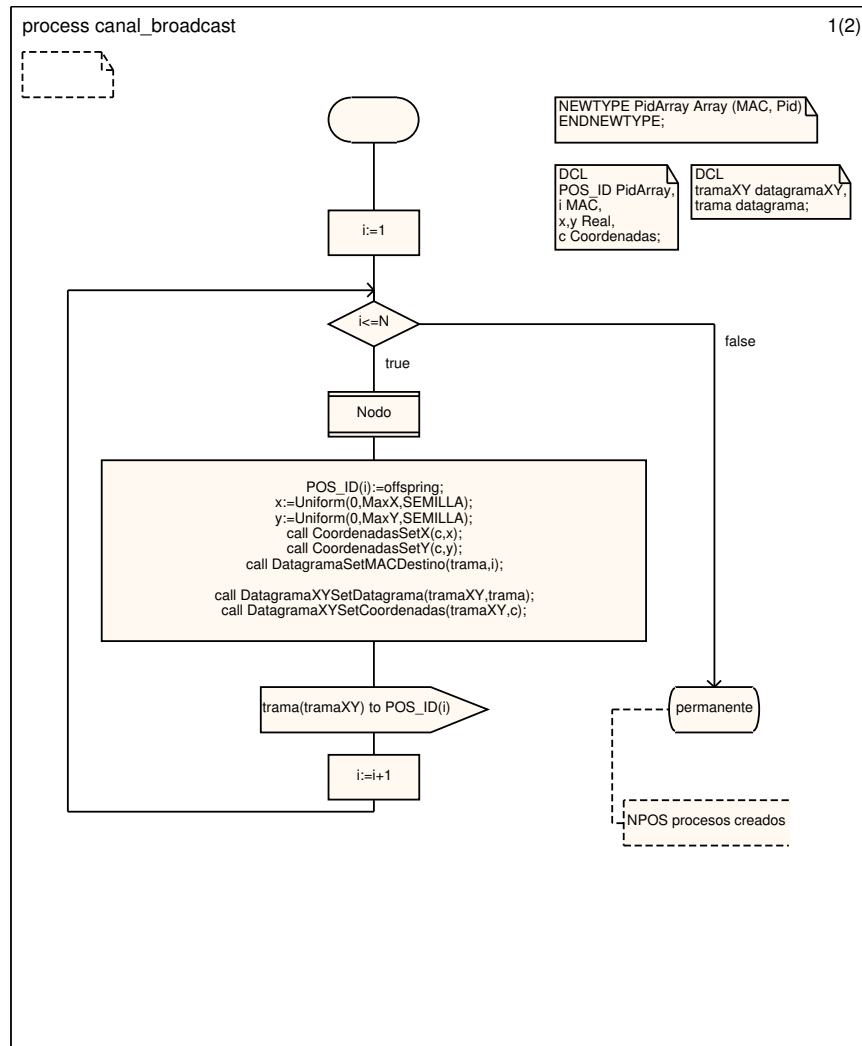


Figura 5.2: El proceso Canal_broadcast (régimen transitorio) crea N procesos Nodo y les asigna sus posiciones

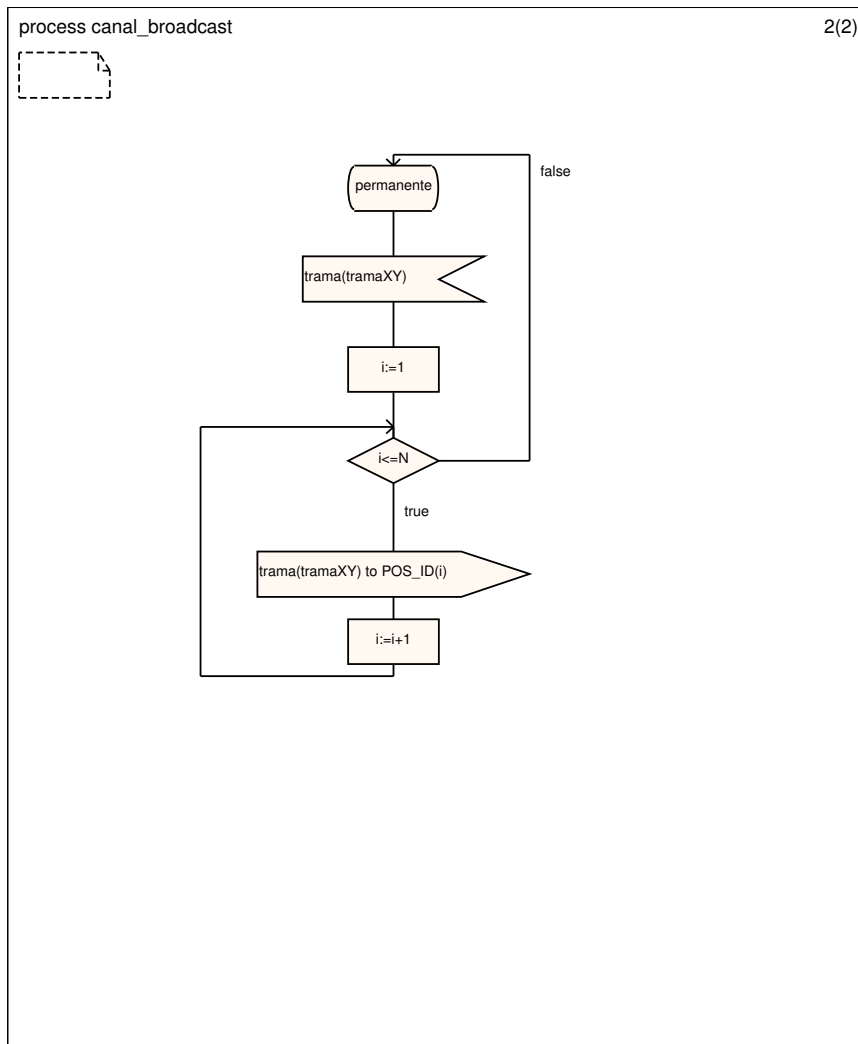


Figura 5.3: El proceso Canal_broadcast (régimen permanente) replica N veces cada trama

ficheros de texto, para ser analizados con posterioridad.

5.3. Parámetros de simulación

El simulador obtenido permite simular cualquier número de nodos, cualquier radio de cobertura y cualquier topología. Lo único que es necesario es pasarle los parámetros adecuados al simulador y ejecutarlo. Ahora bien, existen una serie de escenarios que serán de utilidad de cara a su estudio, por lo que el simulador se ha utilizado en un conjunto de escenarios bien definidos.

Un escenario consiste en un cuadrado normalizado, de lado unidad, donde se posicionarán los nodos. Los nodos tendrán un radio de cobertura que se expresa en relación con la arista del cuadrado. Por simplicidad, todos los nodos se consideran iguales. Consecuentemente, un escenario se puede describir como un conjunto de 3 parámetros:

- Un número de nodos
- El radio de cobertura de los nodos
- La topología (posiciones) de los nodos.

En este capítulo se estudian tres topologías: línea, malla y aleatoria. La elección se ha hecho por varios motivos. En primer lugar por orden de complejidad, para poder entender mejor el comportamiento del protocolo W2LAN. Otro motivo es que estas topologías reflejan casos de uso típicos: La topología en línea se puede aplicar para crear perímetros (por ejemplo autopistas, zona acordonada, etc.), la topología en malla se puede usar para dar cobertura a cierta área, y la topología aleatoria puede representar escenarios con movilidad.

5.3.1. Topología en línea

El experimento con la topología en línea es el más sencillo que se ha llevado a cabo. Consiste en una serie de nodos situados de manera equidistante en la arista inferior del cuadrado de lado unidad. Por ejemplo, el primer experimento (figura 5.4) consiste en 2 nodos separados una distancia de una unidad. El resultado trivial que se ha obtenido es, para un radio de cobertura de una unidad, es que el coste absoluto (ver 6.2) de una conversación es 4. En este caso, las conversaciones consisten siempre en una trama *Announce*, una trama *Request*, una trama *Data* y una trama *Announce*. Debe observarse que las tramas *Announce* siempre miden 34 bytes y las tramas *Request* siempre miden 20 bytes (ver 4.2.4). La trama *Data* siempre mide exactamente igual que la trama Ethernet siendo transportada por la conversación. En general, un número de N nodos, $N \geq 2$, se sitúan equidistantes en la arista inferior del cuadrado de lado unidad.

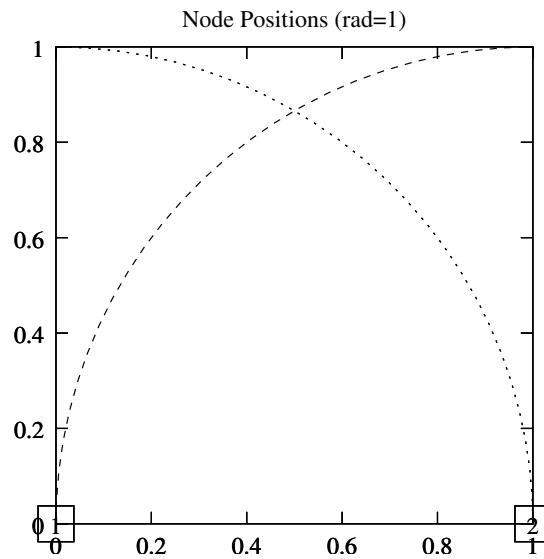


Figura 5.4: Topología lineal. Primer experimento.

5.3.2. Topología en malla

La topología en malla es la extensión natural a dos dimensiones de la topología en línea. Consiste en una serie de nodos situados desde el vértice inferior izquierdo del cuadrado de lado unidad hasta el vértice superior derecho, todos ellos equiespaciados. Por ejemplo, el primer experimento (figura 5.4) consiste en 4 nodos, uno en cada esquina. El primer resultado obtenido, para un radio de cobertura de 1, es que el coste absoluto (ver 6.2) de cualquier conversación es 9. Estas conversaciones consisten en 1 trama *Announce*, 2 tramas *Request*, 1 trama *Data*, 2 tramas *Announce*, 1 trama *Request* (hacia el primer anunciante), 1 trama *Data* (desde el primer anunciante) y 1 trama *Announce*. El segundo resultado obtenido, para un radio de cobertura de $\sqrt{2}$, es que el coste absoluto de cualquier conversación es 8. Estas conversaciones consisten en 1 trama *Announce*, 3 tramas *Request*, 1 trama *Data* y 3 tramas *Announce*. Ambos resultados son mejores que la cota superior descrita en 6.2. En general se situará un número de i^2 nodos, $i \geq 2$, en el cuadrado de lado unidad descrito anteriormente.

5.3.3. Topología aleatoria

El número de experimentos en la topología aleatoria consiste en el producto del número de nodos por el número de radios de cobertura. Por ejemplo, el primer experimento consiste en 2 nodos posicionados aleatoriamente, con la restricción de la suposición del protocolo W2LAN (ver 4.2.2) de que existe al menos una ruta entre cualquier par de nodos. Siguiendo esta directriz, el algo-

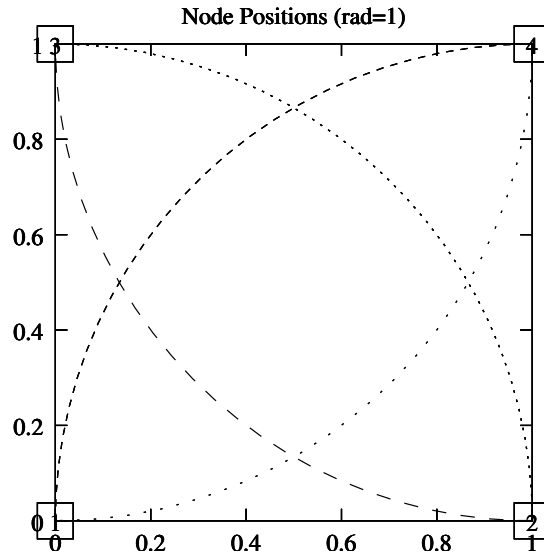


Figura 5.5: Topología en malla

ritmo de posicionamiento de nodos aleatorio selecciona un nodo (en este caso nodo 1) y sitúa el siguiente nodo (en este caso nodo 2) dentro de la cobertura del nodo previamente escogido. La figura 5.6 muestra un ejemplo arbitrario con 10 nodos y radio de cobertura de 0,5.

5.4. Medidas realizadas

La evaluación del protocolo W2LAN se ha realizado mediante simulaciones de eventos discretos, utilizando la plataforma descrita en la sección 5.2. Todas las simulaciones se han ejecutado en redes Ad-Hoc desde 2 hasta 25 nodos. Todos los nodos se han situado dentro del cuadrado de lado unidad. Todas las simulaciones se han ejecutado durante 1.000 unidades de tiempo simulado y 10.000 unidades de tiempo simulado, obteniendo variaciones en el coste de menos de 0,1 %. El tráfico ha sido generado mediante la distribución exponencial negativa (*Poisson*).

Otra consideración necesaria para entender bien las medidas son los límites teóricos explicados en la sección 6.2. Dado un conjunto de N nodos y una conversación C , el número de tramas *Announce* es siempre N , ya que habrá un nodo que inicie la conversación C y $N - 1$ que tarde o temprano tendrán la trama Ethernet asociada a la conversación C , y consecuentemente la anunciarán. De manera similar, el número de tramas *Request* es siempre $N - 1$, ya que el nodo que inicia una conversación no necesita solicitarla, y el resto de los nodos tarde

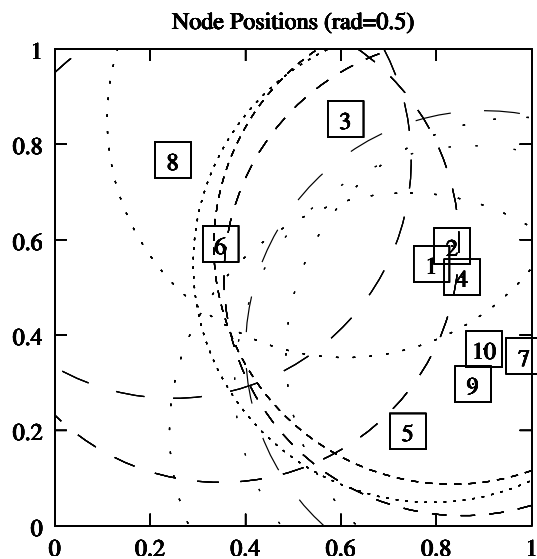


Figura 5.6: Ejemplo de topología aleatoria

o temprano la solicitarán. Finalmente, el número máximo de tramas *Data* en la peor situación posible (por ejemplo, topología lineal con radio de cobertura igual a la distancia entre nodos) es $N - 1$, que no es más que cada nodo retransmite la trama *Data*.

5.5. Gráficas más relevantes

El análisis del Coste/Beneficio del protocolo W2LAN consiste, para cada escenario bajo estudio, en hallar el número medio de tramas *Data* por conversación. Para el resto de la sección, cuando se hable de la evaluación del coste de una conversación no se estará tomando en cuenta el número constante de tramas *Announce* y *Request* (ver sección 6.2). En las siguientes subsecciones se analizarán las diferentes gráficas para diferentes numero de nodos, radio de cobertura (eje X) y número de tramas por conversación (eje Y) para las topologías lineal, malla y aleatoria. Adicionalmente y para añadir detalle al análisis, para las regiones de interés de las gráficas se añaden gráficas adicionales para un número concreto de nodos.

5.5.1. Topología en linea

Los resultados mostrados en la figura 5.7 para la topología en linea muestran que el número medio de tramas *Data* por conversación queda acotado entre 1 y 10, dentro de los límites teóricos. El primer detalle a observar es que, para

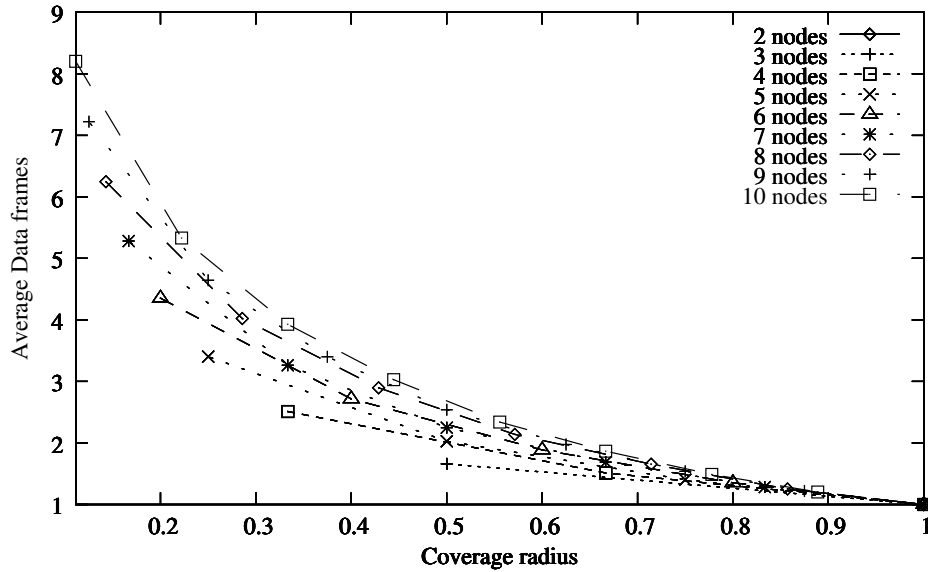


Figura 5.7: Resultados para 2..10 nodos

cualesquiera número de nodos, el experimento con radio de cobertura la unidad genera el valor unidad, que no es más que el caso donde todos los nodos tienen visibilidad directa entre ellos. Este punto, que es el que da el menor coste por conversación posible, indica que se utilizará una sola trama *Data* por conversación. Otra zona de interés es la de peor comportamiento posible, que para cualesquiera número de nodos corresponde con el radio de cobertura mínimo. Esta colección de puntos (los extremos por la izquierda de todas las curvas) tienen una particularidad, que es que el número de tramas *Data* por conversación es determinista. Sea un conjunto de N nodos posicionados según se describe en 5.3.1, el número de tramas *Data* por conversación es $N - 1$ si el nodo que originó la conversación es un extremo, y $N - 2$ si no lo es. Por lo tanto, el número de tramas *Data* por conversación será en 2 ocasiones (los extremos) $N - 1$ y en $N - 2$ ocasiones (los nodos interiores) $N - 2$. Así, el número medio de tramas *Data* se expresa mediante:

$$\frac{1}{N}(2(N - 1) + (N - 2)(N - 2))$$

Agrupando:

$$(N^2 - 2N + 2)/N$$

Así, esta fórmula genera los mismos valores que el primer punto de cada curva, y es ligeramente inferior al límite teórico superior de coste por conversación.

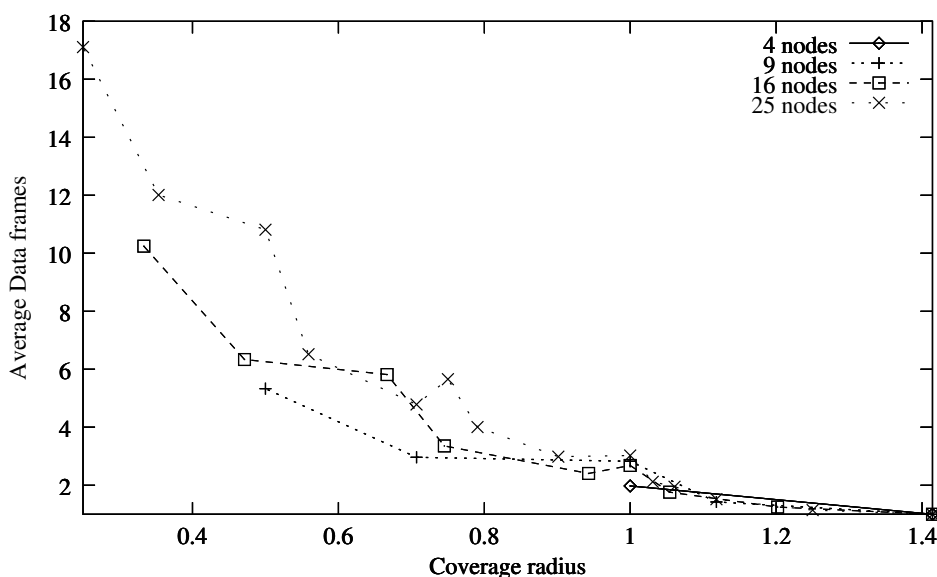


Figura 5.8: Resultados para 4, 9, 16 y 25 nodos

Los valores intermedios para cualesquiera de las curvas (ya todos no deterministas) están dentro del margen teórico de costes por conversación. El primer valor para toda curva (peor caso posible) ha sido explicado en el párrafo anterior, y los valores que le siguen decrecen de manera monótona hasta el valor de la unidad, caso de visibilidad total.

5.5.2. Topología en malla

Los resultados para la topología en malla mostrados en la figura 5.8 revelan que el número de tramas *Data* por conversación está entre 1 y 17 (curva de 25 nodos), considerablemente mejores (menor coste por conversación) que la cota máxima teórica genérica del coste por conversación (ver 6.2) de $N - 1$ (24 en el caso de 25 nodos). La explicación de este comportamiento de mejora se debe a que la topología en malla explota mejor la naturaleza broadcast del protocolo W2LAN. Al incrementar (2 dimensiones) la vecindad de los nodos aumenta el número de oyentes por transacción, pasando algunos de los nodos a no ser necesarios para propagar la conversación.

El primer detalle a observar en la topología en malla es que, para cualquier número de nodos, el experimento con radio de cobertura $\sqrt{2}$ (al trabajar en 2 dimensiones el radio de cobertura máximo pasa a ser la diagonal del cuadrado de lado unidad) genera el valor unidad, que no es más que el caso donde todos los nodos tienen visibilidad directa entre ellos. Este punto, al igual que en el

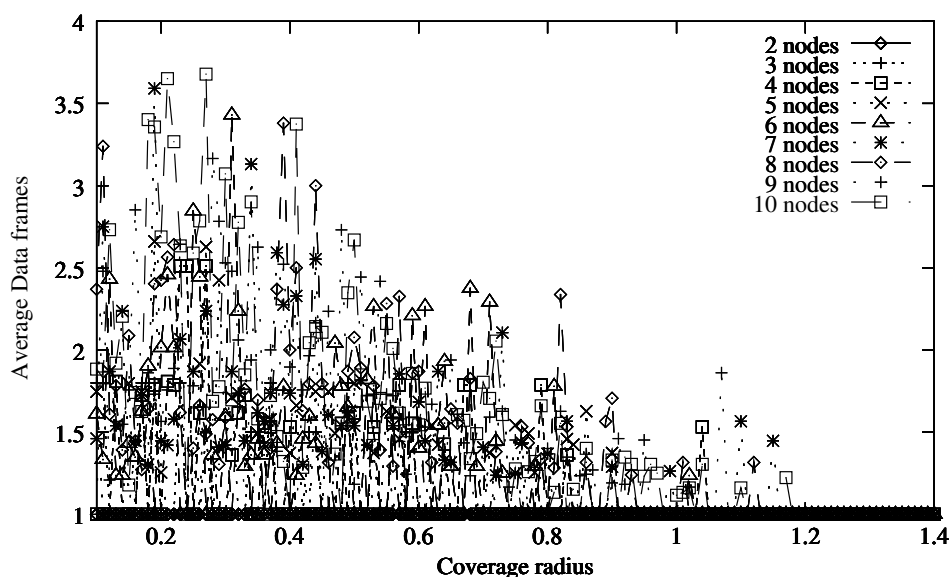


Figura 5.9: Resultados para 1 .. 10 nodos

caso de la topología lineal, es el que da el menor coste por conversación posible -la unidad-, e indica que se utilizará una sola trama *Data* por conversación.

De nuevo, el peor caso posible para todas las curvas corresponde con el radio de cobertura mínimo. Ahora bien, los valores correspondientes al peor caso posible resultan mucho mejores que en el caso de la topología lineal. Por ejemplo, la curva de 25 nodos presenta un valor de coste medio de 5,33 tramas *Data* por conversación. El valor que le correspondería con una topología lineal de 25 nodos es, mediante la fórmula obtenida en 5.3.1, de 23,08 tramas *Data* por conversación, siendo 24 tramas *Data* por conversación el límite teórico superior genérico (ver 6.2).

El resto de los valores, comprendidos entre los dos extremos de las curvas, decrecen hasta el valor de la unidad (caso de visibilidad total), siempre dentro del rango teórico de valores.

5.5.3. Topología aleatoria

Los resultados para la topología aleatoria, representados en la figura 5.9, son, en primera instancia, confusos.

La primera situación remarcable es que hay una divergencia entre puntos consecutivos de una misma curva, que hasta ahora no se daba. La explicación es simple: Para obtener los puntos de cualquiera de las curvas, se ha procedido respetando la suposición del protocolo W2LAN de que al menos existe una

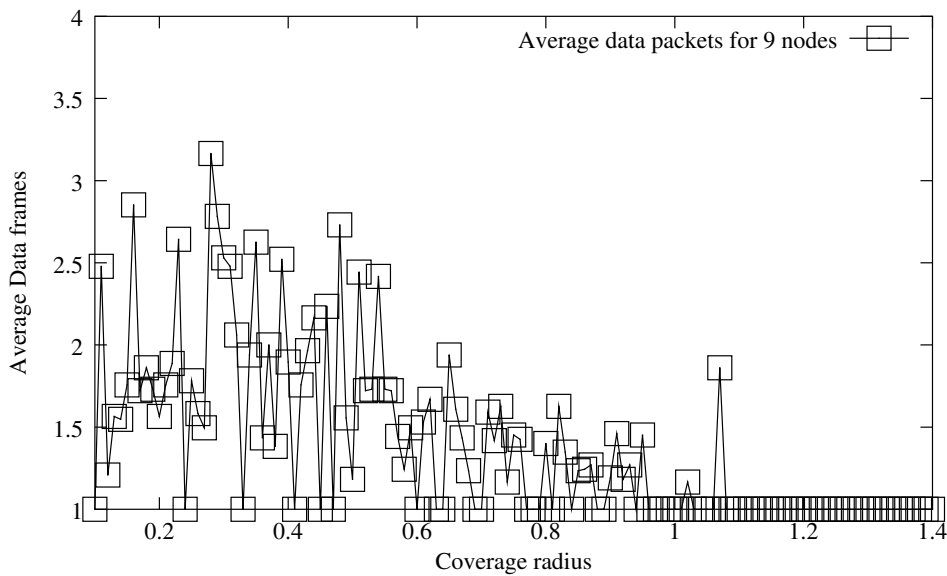


Figura 5.10: Curva de 9 nodos

ruta entre cada pareja de nodos (ver 4.2.2). Así, para cada paso incremental del radio de cobertura (se han escogido pasos de radio de cobertura de 0,01 para todas las curvas) se ha generado una topología aleatoria donde W2LAN pueda operar, es decir, una topología aleatoria de red conectada, con al menos un camino multisalto entre 2 nodos cualesquiera. Esto significa que cada punto ha sido generado con una disposición distinta de nodos. Ahora bien, a pesar de la confusión inicial se puede empezar a ver un valor medio por curva con comportamiento parecido a las topologías vistas anteriormente.

Otra observación importante es que la topología aleatoria se comporta mejor que las anteriormente vistas, pero de nuevo es un resultado influenciado por la manera de generar patrones aleatorios.

La figura 5.10 representa la curva de 9 nodos, que no es más que un caso particular de la figura 5.9, donde se puede observar mejor el comportamiento de la curva. Lo primero que se puede remarcar es que la curva se puede dividir en dos zonas.

La primera zona corresponde a radios de cobertura bajos. Debido a que en cualquier situación se fuerza a operar con una red conectada, dicha red cubrirá una parte pequeña del cuadrado de lado unidad. En esta zona, si se va aumentando el radio de cobertura, la red conectada irá cubriendo cada vez mayor proporción del cuadrado de lado unidad. Ahora bien, hay que observar que de hecho todos los experimentos en esta zona no dejan de ser el mismo experimento con distintos patrones. Así, cuando se opera con radios de cobertura que em-

piezan a permitir cubrir la superficie del cuadrado de lado unidad se estará en el límite entre zonas.

La segunda zona corresponde a radios de cobertura altos. Debido a que en esta zona el cuadrado de lado unidad está prácticamente cubierto, si se sigue aumentando el radio de cobertura de los nodos se disminuirá el coste por conversación. Esta zona corresponde con la zona útil.

Finalmente, se puede observar que los resultados son mejores con la topología aleatoria que con las demás, pero de nuevo hay la influencia del algoritmo de posicionamiento de nodos, que favorece la proximidad entre nodos, y consecuentemente en media representa menos tramas *Data* por conversación. Ahora bien, esta situación es quizás la más realista en un entorno MANET.

Capítulo 6

Aspectos teóricos de W2LAN

A la hora de diseñar el protocolo W2LAN se han tenido en cuenta ideas ya existentes y han aparecido conceptos nuevos. También ha habido la necesidad de saber qué precio se paga al utilizar W2LAN (no hay nada gratis), y poder decidir con conocimiento de causa si se usa o se usan otros protocolos.

6.1. Conceptos utilizados por W2LAN

En esta sección se describen los conceptos más relevantes que se han utilizado para el diseño del protocolo W2LAN. Como suele suceder en la mayoría de los protocolos, los diferentes conceptos utilizados son dispares, provenientes de diferentes campos del conocimiento. Dichos conceptos se utilizan a modo de bloques de construcción para obtener el modelo final.

6.1.1. Modelos naturales

El utilizar modelos naturales no es nuevo en el campo de las redes móviles [GDC04]. De hecho, la Naturaleza ha servido en muchos campos de conocimiento como inspiración, y debería de servir de inspiración con mayor frecuencia por una razón sencilla: supervivencia. Todo lo que trasciende en el tiempo, que sobrevive, es porque tiene un buen diseño. Quizás uno no entienda el porqué cierto diseño ha trascendido. Quizás uno piense que el diseño podría haber sido otro. No importa: es el que es, ha sobrevivido.

Bajo esta premisa, a la hora de diseñar un sistema artificial (entiéndase artificial en oposición a natural) la tarea del diseñador es más bien la comprensión de qué deberá hacer el sistema, seguido de una búsqueda de algún sistema natural similar. Una vez encontrado, habrá que plasmarlo en el sistema artificial a diseñar.

Para el caso concreto de W2LAN el modelo natural seguido ha sido el de una habitación, o una plaza, o un recinto, con personas. De vez en cuando, alguna de las personas puede tener algo interesante que contar. Para poner un ejemplo sencillo, supóngase un chiste. La persona a la que se le ocurre el chiste no lo contará sin más, sino que preguntará a su entorno cercano “¿saben el chiste bla bla bla? (ver 6.1.5)”. Este entorno cercano estará compuesto por un número determinado de personas, que pueden saber o no el chiste en cuestión. Si alguien no lo sabe, contestará “Cuéntamelo (el de bla bla bla).”. Si la persona dispuesta a contar el chiste ha recibido al menos una solicitud, lo contará (ver 6.1.4). A partir de este punto el entorno cercano conoce el chiste, iterando el proceso. Tarde o temprano todo el mundo sabe el chiste, y cualquiera que pregunte si alguien conoce el chiste se dará cuenta de que nadie está interesado, por lo que (¿por no hacerse pesado?) no lo contará.

6.1.2. Copia digital

Un paradigma nuevo surgido de la era digital es el de la copia de productos. Un ejemplo clarificador de dicho paradigma sería una fábrica de automóviles: Cada “copia” de automóvil tiene un “precio” asociado, medido en horas de trabajo, materiales, etc. Si en lugar de una fábrica de automóviles se estuviese hablando de una fábrica de, por ejemplo, software -cualquier producto que pueda existir en formato digital (software, música, vídeo, libros, ...)- se encontraría un escenario diferente: el coste de la copia de la parte digital del producto es prácticamente nulo.

Este hecho nuevo ha generado una multitud de consecuencias en diferentes direcciones. Una primera consecuencia ha sido el beneficio económico que empresas del sector del software, manteniendo el modelo de negocio tradicional (fábrica de automóviles), han obtenido¹. Esta consecuencia se hizo notar también en la industria audiovisual, marcado quizás por el nacimiento de Napster, la primera red Peer-to-Peer (P2P) que alcanzó grandes cotas de popularidad. De hecho, actualmente la popularidad de estas aplicaciones es tan notoria que estudios realizados en nuestra Red Regional de acceso a Internet (CTNET, Red de Ciencia y Tecnología) [GUI04] demuestran que el 80 % del tráfico que viaja por la red es de este tipo.

Una segunda consecuencia -trivial- del coste cero de la copia es la proliferación de copias. La plataforma por excelencia en que se sustentan las copias es Internet, en toda la variedad de servicios que lo permiten. Inicialmente estos servicios fueron ftp, mail y http, pero pronto aparecieron paradigmas de más alto nivel que se apoyan en los anteriores: Las redes P2P [HWP02].

¹El ejemplo clásico es Microsoft

La característica relevante de las redes P2P en relación a la copia digital es que pueden existir (suelen existir) gran multitud de copias del mismo contenido. Esta gran redundancia genera robustez, en el sentido en que si se desea descargar un contenido en concreto, por el simple hecho de estar replicado en multitud de sitios, la probabilidad de éxito es mayor.

A pesar de que la gran mayoría de redes P2P comparten contenidos (ficheros), utilizando un esquema parecido se pueden compartir otros recursos, como CPU (por ejemplo, proyecto BOINC [BOI05]) o ancho de banda. Un ejemplo de compartir ancho de banda es el de la telefonía P2P propuesta por Skype [SKY05], esquema que desde el punto de vista popular da buenos resultados². Este sistema explota el paradigma de la copia digital, mediante el uso de múltiples conexiones VoIP -copias redundantes-, utilizando en cada momento la más adecuada. De este modo Skype consigue reducir la latencia e incrementar la calidad de la llamada.

W2LAN utiliza una aproximación similar por lo que respecta al paradigma de copia digital, haciendo copias redundantes de tramas que se esparcen por la LAN. En el caso de comunicaciones unicast, de todas las tramas utilizadas en un conversación -copias-, sólo una trama será “la más adecuada”. De esta manera se puede ver a una conversación W2LAN como a una información a ser transportada, mediante el soporte de múltiples copias -algunas redundantes- de tramas. En el caso de comunicaciones multicast o broadcast se sigue exactamente el mismo principio³, con la única diferencia que ahora, en lugar de una, serán una o más de una las tramas más adecuadas. Así, utilizar W2LAN en comunicaciones multicast o broadcast es una solución de gran sencillez (e igual coste que unicast).

6.1.3. Anuncio, Solicitud, Datos

W2LAN trabaja sobre dispositivos Ethernet, dispositivos que entienden un determinado formato de trama, especificado por la norma IEEE 802.3 [802.3]. Consecuentemente, W2LAN debe respetar el formato de trama por razones de compatibilidad. Los requisitos del protocolo han introducido nueva información a ser transportada, por lo que la segmentación de la trama Ethernet original, si se quiere mantener la compatibilidad, se hace necesaria. La manera natural de segmentar la trama Ethernet es contemplando la propia trama Ethernet y el modelo que sigue W2LAN (ver 6.1.1). A nivel DLC, la trama Ethernet consta de una cabecera de 14 bytes (6 bytes dirección destino, 6 bytes dirección origen,

²PC-Actual, 8/2/2005, “Cuando se trata de hacer llamadas telefónicas gratuitas a través de Internet utilizando la tecnología de voz sobre IP, Skype es simplemente lo mejor. Con más de 50 millones de personas que han descargado el software, Skype es la aplicación más utilizada.”

³de hecho la capa W2LAN desconoce la naturaleza de las comunicaciones transportadas en capas superiores (DLC), siguiendo la filosofía OSI (*Open Systems Interconnection*).

2 bytes tipo de datos) y un cuerpo de tamaño variable, entre 64 y 1500 bytes. El modelo W2LAN consta de un anuncio, y en caso de interés una información asociada al anuncio. Teniendo en cuenta estos dos factores, la trama Ethernet se dividirá en una trama *Announce*, que incluirá como datos la cabecera Ethernet, y una trama *Data*, que incluirá como datos el campo de datos Ethernet. El último detalle a tener en cuenta es la información adicional, que no es más que un identificador único de conversación (ver 6.1.5) presente en cualquier trama W2LAN. Dicho identificador es de tamaño 6 bytes, y en las tramas *Data*, donde sólo hay espacio disponible en la cabecera, substituirá los ahora innecesarios 6 bytes de dirección origen. El formato de trama completo se encuentra en 4.2.4.

6.1.4. Transacción W2LAN

El concepto de transacción W2LAN va asociado a un nodo, al nodo que está efectuando la transacción, y no es más que el subconjunto de tramas pertenecientes a una misma conversación atendidas por el nodo en cuestión. Se pueden dar dos situaciones: ser el origen de una transacción o ser el destino de una transacción. Si un nodo es el origen de una transacción deberá emitir una trama *Announce*, esperar durante un tiempo *TimerSolicitud* la recepción de tramas *Request*, y si ha llegado alguna, concluir la transacción emitiendo una trama *Data*. Si un nodo es el destino de una transacción, es que ha recibido una trama *Announce* de una conversación desconocida, por lo que emitirá la correspondiente trama *Request*, y cuando reciba la trama *Data* solicitada el nodo pasará a ser origen de conversación, procediendo de manera acorde.

Un detalle de importancia en las transacciones es el tiempo entre la emisión de una trama *Announce* y la emisión, si corresponde, de una trama *Data*. Este tiempo viene modelado por un temporizador llamado *TimerSolicitud*. De cara a la especificación del protocolo su valor es irrelevante. En la implementación del simulador se ha utilizado un valor “lo suficientemente grande”, ya que debido a que en el simulador se trabaja con tiempo virtual, el único momento en que se consume tiempo es en los temporizadores. En un caso real se deberá tomar la precaución de escoger correctamente el valor del temporizador. Un valor demasiado pequeño no daría tiempo a finalizar las transacciones (timeout demasiado corto en comparación con el tiempo de propagación más el de transmisión), y un valor demasiado grande, a pesar de que no habría ningún problema procedural, conduciría a latencias elevadas, especialmente en topologías con múltiples saltos. Con estos condicionantes se sugiere en implementaciones reales algún mecanismo sencillo de ajuste dinámico del temporizador.

6.1.5. Conversación W2LAN

El concepto de conversación W2LAN va asociado a la red, y no es más que el conjunto de tramas en la red que comparten el mismo identificador de conversación (*ConversationID*). La operación idealizada de W2LAN será bajo dos condiciones de entorno ideales: sin repetir nunca un identificador de conversación (ver 6.1.5), o infinitos *ConversationID*, y que los nodos tienen memoria infinita de las conversaciones que han tenido (ver 6.1.5).

En un entorno W2LAN ideal una conversación se esparcirá a todos los nodos, desapareciendo de la red por falta de interés. En un entorno real se debe procurar que suceda lo mismo, por lo que habrá que tener en cuenta las limitaciones técnicas indicadas al aplicar el modelo.

Firma de la Conversación

La firma de la conversación no es más que un identificador “único” de 48 bits, que se genera de manera aleatoria en el momento de iniciar una conversación. El hecho de haber elegido el número de 48 bits es conveniente de cara al formato de trama (ver 4.2.4), pero ciertamente deja de ser único, ya que obviamente sólo habrá 2^{48} identificadores distintos.

Si se utilizase una memoria infinita en los nodos, en el caso más optimista la red dejaría de funcionar completamente después de 281.474.976.710.656 conversaciones (2^{48}), lo que no parece en principio preocupante. Ahora bien, con la hipótesis de memoria infinita los comportamientos anómalos aparecerían antes, ya que conforme se vayan utilizando identificadores, la probabilidad de generar uno repetido va aumentando (casos favorables / casos posibles), y una conversación nueva con un identificador repetido se trataría como una información ya conocida, y por tanto se ignoraría.

Memoria de la Conversación

Cada vez que un nodo recibe un anuncio de una conversación nueva, su identificador (el de la conversación) se añade a la memoria -de tipo búfer-. Con la hipótesis anterior de memoria infinita y 2^{48} conversaciones no se tendría un comportamiento deseado, debido a las repeticiones de identificadores. La solución planteada, caso realista, pasa por no usar una memoria infinita. Ahora, al tener la memoria un tamaño limitado, las conversaciones no son recordadas eternamente, por lo que se pueden reciclar identificadores ya utilizados sin generar anomalías. En el caso más desfavorable, la probabilidad de utilizar un identificador en uso aún recordado será igual al tamaño del búfer (medido en número de conversaciones que le caben) partido por 2^{48} . El caso estudiado mediante simu-

lación no dio problemas, utilizando un búfer de 2^{10} conversaciones. La probabilidad de repetición indeseada de identificador es 1 entre 274.877.906.944 (2^{38}).

Otro punto de interés es el hecho de el tiempo en memoria de un conversación sólo depende inversamente del tráfico global de la red. Si en la red hay más conversaciones, éstas estarán menos tiempo en memoria.

Equilibrio entre Firma y Memoria de Conversación

El hecho de no utilizar infinitos identificadores e infinita memoria lleva a la búsqueda de equilibrios, o puntos de trabajo. Si se usa una memoria “grande”, una conversación está “mucho tiempo” en memoria (en la lista de conversaciones), y puede aparecer una conversación nueva con un identificador de conversación que ya esté en memoria (ver puntos anteriores). Este tipo de problema se puede interpretar como trama perdida, y lo solucionarían capas superiores, por ejemplo, solicitando retransmisión. Si la memoria utilizada es “pequeña”, una conversación está “poco tiempo” en memoria, pudiéndose aceptar datos previamente aceptados, de una conversación aún no extinguida de la red. Este caso, en el mejor de los casos, se puede interpretar como trama duplicada, que también debería ser solucionado por capas superiores. En el peor de los casos, se podría dar un bucle, situación altamente indeseada. Esta situación es rebuscada, y se puede explicar mediante el ejemplo de la figura 6.1, donde los nodos están posicionados según la esfera de un reloj, con cobertura sólo de los vecinos colindantes, y memoria de 3 conversaciones. Supóngase que a cada nodo le llega alguna otra conversación por cada transacción ejecutada de la conversación bajo estudio. El caso anómalo empieza cuando el nodo 12 emite. 11 está caído. La conversación avanza por 1, 2, ..., 9, 10. Cuando llega a 10, 11 ya no está caído. cuando 11 anuncia, si 12 ya no se acuerda de la conversación, la aceptará, repitiéndose el proceso anteriormente explicado. Esta situación debe evitarse. La figura 6.2 analiza la misma conversación en un caso normal, sin anomalías. El que se dé esta situación anómala dependerá de bastantes factores (los factores que marcan la supervivencia de una conversación), la topología de red, el número de conversaciones simultáneas en la red, el funcionamiento de los nodos, etc., por lo que es conveniente curarse en salud y no quedarse corto, usando un tamaño de memoria que no sea “pequeño”. De esta manera sencilla se evita cualquier anomalía. En resumen, el tamaño de la memoria en conversaciones debe situarse dentro de un amplio rango, especialmente evitando la zona baja. Un valor típico que no da problemas de tamaño de memoria es 1000 (2^{10}) conversaciones, con cada conversación ocupando 20 bytes (cabecera Ethernet + *ConversationID*).

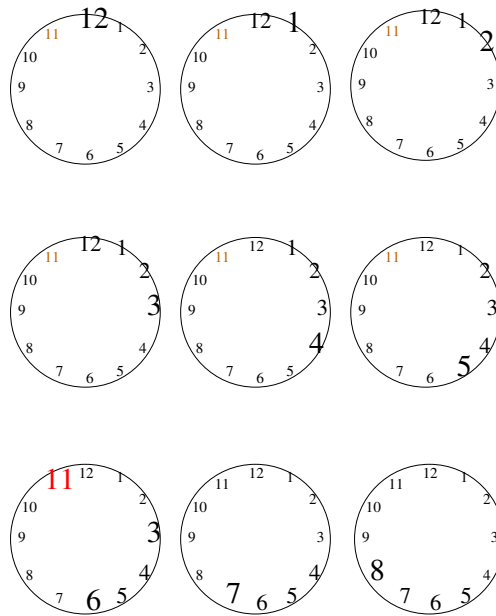


Figura 6.1: Situación de bucle anómala

6.1.6. Temporizador *TimerSolicitud*

En la especificación del protocolo W2LAN se puede observar que entre la emisión de una trama *announce* y la posible emisión de la correspondiente trama *Data* se deja transcurrir un tiempo controlado por el temporizador *TimerSolicitud*. En la validación del protocolo no se hace ninguna suposición sobre este tiempo, como es habitual en la programación concurrente. Ahora bien, este tiempo que se deja transcurrir debe servir para dar tiempo a los demás nodos a contestar anuncios, es decir, si es “demasiado” corto no servirá (los nodos vecinos solicitan tarde) y si es “demasiado” largo introducirá retardos importantes en las comunicaciones. El ajuste de este temporizador deberá hacerse de manera manual sobre una red implementada y probada, o bien se podría utilizar algún mecanismo de ajuste dinámico.

6.2. Cotas del coste por Conversación

Para establecer las cotas del coste por conversación, primero hay que definirlo sin ambigüedades. Una primera definición es la de coste absoluto, que es el número total de tramas totales, sin importar el tamaño, utilizadas en una conversación, o de manera equivalente, número de tramas W2LAN necesarias para esparcir una trama Ethernet original. Este coste absoluto constará de un número

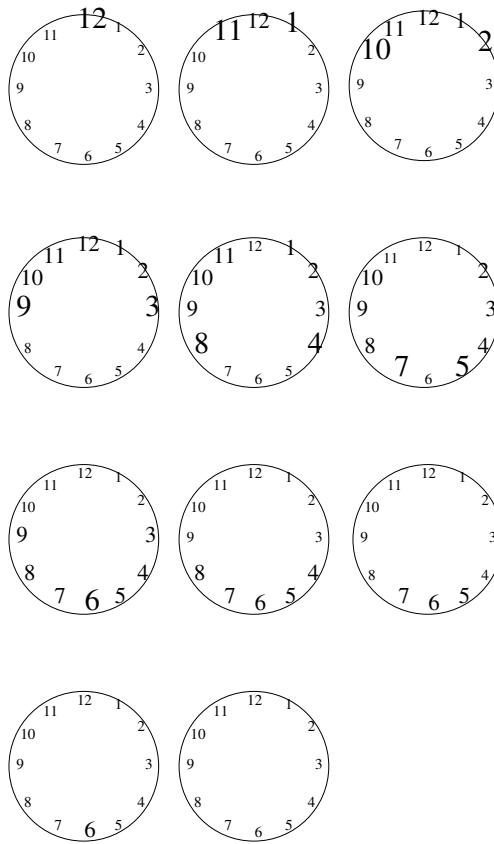


Figura 6.2: Situación correcta

de tramas *Announce*, otro de *Request* y un tercero de *Data*. Dado un escenario se puede observar que habrá un número fijo de tramas, las *Announce* y las *Request*, sumado con un número variable de tramas, las *Data*.

En el capítulo 5 se han realizado medidas de coste. El coste allí medido corresponde con la parte variable del coste absoluto, es decir, las tramas *Data*. Hay que remarcar que éstas son las tramas que realmente llenan el canal.

Otra manera de medir el coste, que no se ha utilizado, podría ser la relación entre bytes útiles (trama Ethernet de cierto tamaño medio) respecto de bytes por conversación. Este tipo de indicador daría una idea de coste (en bytes) por byte útil transportado, pero por otro lado se perdería la noción de repeticiones por trama.

6.2.1. Numero de Anuncios por Conversación

Para cualquier topología, el número de tramas *Announce* siempre coincidirá con el número de nodos en la red. Mediante la visión global de la red es difícil llegar a esta conclusión. La manera de entenderlo consiste en ver la red como un sistema distribuido, y analizarla desde el punto de vista de un nodo de la misma. Así, un nodo de la red, de cara a contabilizar tramas *Announce*, tiene dos opciones. Si ha iniciado una conversación nueva porque le han llegado datos de capas superiores significa que deberá anunciar, una vez, la nueva conversación. Si ha iniciado una nueva transacción de una conversación existente, también será hecho una sola vez.

6.2.2. Numero de Solicitudes por Conversación

De manera similar al punto anterior, el número de tramas *Request* siempre coincidirá con el número de nodos en la red menos uno. De nuevo, la manera de entenderlo es ver la red como un sistema distribuido, y analizarla desde el punto de vista de un nodo de la misma. Así, si un nodo ha iniciado una conversación nueva porque le han llegado datos de capas superiores significa que nunca solicitará la nueva conversación. Por otro lado, si un nodo está solicitando una trama *Announce* que ha recibido de la red y es nueva, sólo emitirá una trama *Request*.

6.2.3. Numero de Datos por Conversación

El número de tramas *Data* será variable en función de la topología y del nodo que inicie la conversación. Mediante la visión global de la red es difícil acotar el número de tramas *Data* por conversación. La manera de enfocararlo es, de nuevo, viendo la red como un sistema distribuido, y analizarla desde el punto de vista

de un nodo de la misma. Así, un nodo de la red, de cara a contabilizar tramas *Data*, tiene dos opciones. Previo anuncio de la conversación, el nodo recibirá un número variable de tramas *Request*. Si recibe cero tramas *Request* no transmitirá *Data*, y si recibe una o más transmitirá *Data* (una vez). Por lo tanto, el número de tramas *Data* por conversación está entre cero (caso no práctico de un nodo aislado) y el número de nodos de la red.

Capítulo 7

Conclusión

7.1. Conclusiones Finales

La motivación inicial de este trabajo fue la de obtener una plataforma tipo *Set-Top-Box* de bajo coste, abierta y multiservicio para la recepción de contenidos multimedia por parte del gran público. Se ha conseguido diseñar dicha plataforma ([FMS00-1, FMS00-2]), resolviendo así el primer objetivo.

A continuación apareció una segunda motivación: distribuir dichos contenidos multimedia en una red de área local (LAN) cumpliendo una serie de objetivos referentes a la utilización de la red. Este segundo objetivo se ha satisfecho mediante el protocolo MCDP-LAN (*Multimedia Content Distribution Protocol over LAN*, [FBM02]), protocolo que permite el escalado de la LAN por sus cualidades de independencia del tráfico respecto del número de nodos e independencia del tráfico respecto de la demanda de productos.

La tercera y última motivación fue la “migración” de la solución anterior a redes inalámbricas. En lugar de migrar el protocolo anterior se optó por una solución generalizada, que consiste en una transformación de la capa de enlace, convirtiendo desde el punto de vista de capas superiores una red Ethernet *Wireless Ad-Hoc* en una LAN convencional. Este tercer y último objetivo se ha cumplido mediante el diseño y evaluación del protocolo W2LAN (*Wireless to LAN*, [FBM04, FBM05]), que extiende de manera transparente y flexible la cobertura de una WLAN.

7.2. Líneas Futuras de Investigación

A mi entender existen bastantes líneas futuras de investigación, y constantemente aparecen nuevos problemas, que si se afrontan, generan a su vez nuevas líneas de investigación.

La principal línea futura (de hecho es presente) de investigación es la aplicación del protocolo W2LAN en topología Mesh, implementado en dispositivos reales con capacidades para la medición de distintos parámetros. El disponer de campos de prueba reales (escenario típico en redes Mesh) permitirá comprobar/modificar el funcionamiento final del protocolo W2LAN. De manera más específica, el comportamiento del temporizador *TimerSolicitud* entre tramas *announce* y *Data* (ver 6.1.6). Se está llevando a cabo una modificación del kernel de Linux (la capa DLC) para permitir ejecutar W2LAN.

Otra línea futura importante, consecuencia de la primera, es el estudio comparativo con otros protocolos a nivel de implementación. Se puede vislumbrar, empezando por lo negativo, que W2LAN saldrá perdiendo en escenarios de comunicaciones unicast, con tamaño de red grande (varios saltos entre nodos) y baja movilidad de nodos, por lo que bajo estas circunstancias se pueden plantear modificaciones del protocolo, como acotar la extensión del broadcast (por ejemplo, añadiendo un campo TTL, *Time to Live*)¹. Se puede vislumbrar también que W2LAN será una opción muy válida en escenarios de comunicaciones broadcast/multicast, con tamaño de red pequeña (pocos saltos entre nodos) y gran movilidad de nodos. Aquí es donde sufren la mayoría de protocolos, y es donde la transparencia respecto a capas superiores de W2LAN y el soporte natural e inmediato por construcción a comunicaciones broadcast/multicast y entorno móvil pueden dar sus frutos.

¹ Personalmente pienso que es mejor no utilizar W2LAN para lo que no ha sido diseñado

Apéndice A

Especificación SDL del protocolo W2LAN

En este apéndice se incluye la especificación formal del protocolo W2LAN en SDL/PR. El diseño original es en SDL/GR, pero por cuestión de espacio se ha hecho la conversión automatizada a texto. Ambas notaciones (GR y PR) son equivalentes y permiten expresar lo mismo.

```
use Datagrama;
system w2lanNode;
synonym TimeoutSolicitud Duration = 1;
/*Anunciado un datagrama, tiempo para escuchar respuestas*/
synonym SEMILLA = DefineSeed(3);
synonym MiMAC MAC = external;
synonym TamanoListaConversaciones Integer = 100;
/*Datos de arriba, DLC*/
newtype ElementoTimers
/**NAME 'ElementoTimers'*/ struct
t Time;
endnewtype;
newtype ElementoContadores
/**NAME 'ElementoContadores'*/ struct
id Conversacion;
solicitudes Integer;
endnewtype;
newtype ElementoTramasPendientes
/**NAME 'ElementoTramasPendientes'*/ struct
id Conversacion;
trama Datagrama;
endnewtype;
```

```

/* Datos de abajo, MAC*/
newtype ElementoConversaciones
/*#NAME 'ElementoConversaciones'*/ struct
id Conversacion;
pendiente Boolean;
MACDestino MAC;
MACOrigen MAC;
TipoDatos Integer;
endnewtype;
/* Elemento generico de las listas */
newtype Elemento choice
listaTimers ElementoTimers;
listaContadores ElementoContadores;
listaTramasPendientes ElementoTramasPendientes;
listaConversaciones ElementoConversaciones;
endnewtype;
/* lista */
newtype ObjectDescr
/*#NAME 'ObjectDescr'*/ struct
SysVar SysTypeObject;
e Elemento;
endnewtype;
signal
MACframe(Datagrama),
DLCframe(Datagrama);
channel DLCToW2LAN
from env to w2lanNode with DLCframe;
endchannel DLCToW2LAN;
channel W2LANtoDLC
from w2lanNode to env with DLCframe;
endchannel W2LANtoDLC;
channel W2LANtoMAC
from w2lanNode to env with MACframe;
endchannel W2LANtoMAC;
channel MACtoW2LAN
from env to w2lanNode with MACframe;
endchannel MACtoW2LAN;
block w2lanNode referenced;
endsystem w2lanNode;
block w2lanNode;
signalroute DLCToW2LAN
from env to w2lanNode with DLCframe;
signalroute W2LANtoDLC

```

```

from w2lanNode to env with DLCframe;
signalroute W2LANtoMAC
from w2lanNode to env with MACframe;
signalroute MACToW2LAN
from env to w2lanNode with MACframe;
process w2lanNode referenced;
connect DLCToW2LAN and DLCToW2LAN;
connect W2LANtoDLC and W2LANtoDLC;
connect W2LANtoMAC and W2LANtoMAC;
connect MACToW2LAN and MACToW2LAN;
endblock w2lanNode;
process w2lanNode;
timer
TimerSolicitud;
/*Intervalo de tiempo concedido a otros nodos para responder
a la generacion de un anuncio*/
dcl
MACOrigen,
MACDestino,
MACOrigenEthernet,
MACDestinoEthernet MAC,
TipoPaquete,
TipoPaqueteEthernet,
contador Integer,
ConversacionID Conversacion,
CampoDatosEthernet Mensaje,
SolicitudPendiente Boolean,
datosMAC,
datosDLC,
tramaEthernet Datagrama,
listaTimers,
listaContadores,
listaTramasPendientes,
listaConversaciones Queue,
elementoListaTimers,
tempElementoListaTimers,
elementoListaContadores,
tempElementoListaContadores,
elementoListaTramasPendientes,
tempElementoListaTramasPendientes,
elementoListaConversaciones,
tempElementoListaConversaciones ObjectInstance;
start;

```

```

task {
  listaTimers := NewQueue;
  listaContadores := NewQueue;
  listaTramasPendientes := NewQueue;
  listaConversaciones := NewQueue;
  SolicitudPendiente := False;
};
nextstate permanente;
state permanente;
input MACframe(datosMAC);
join datosMAC;
input DLCframe(datosDLC);
join datosDLC;
input TimerSolicitud;
provided SolicitudPendiente;
join Fin_Espera_Solicitudes;
endstate;
connection
datosDLC :
task {
  call DatagramaGetMACDestino(tramaEthernet, MACDestinoEthernet);
  call DatagramaGetMACOrigen(tramaEthernet, MACOrigenEthernet);
  call DatagramaGetTipoPaquete(tramaEthernet, TipoPaqueteEthernet);
  call DatagramaGetDatos(tramaEthernet, CampoDatosEthernet);
  ConversacionID := RandInt(0, 65535, SEMILLA);
};
decision Cardinal(listaConversaciones) >= TamanoListaConversaciones;
(true) :
task {
  elementoListaConversaciones := FirstInQueue(listaConversaciones);
  tempElementoListaConversaciones := elementoListaConversaciones;
  elementoListaConversaciones := Successor(elementoListaConversaciones);
  tempElementoListaConversaciones :=
  DisposeObject(Remove(tempElementoListaConversaciones));
};
(false) :
enddecision;
task {
  elementoListaConversaciones := NewObject;
  elementoListaConversaciones!ref!e!listaConversaciones!id := ConversacionID;
  elementoListaConversaciones!ref!e!listaConversaciones!pendiente := False
  /*Las 3 siguientes lineas son innecesarias, pero dan coherencia
  a la listaConversaciones*/;

```

```

elementoListaConversaciones!ref!e!listaConversaciones!MACDestino :=
MACDestinoEthernet;
elementoListaConversaciones!ref!e!listaConversaciones!MACOrigen :=
MACOrigenEthernet;
elementoListaConversaciones!ref!e!listaConversaciones!TipoDatos :=
TipoPaqueteEthernet;
listaConversaciones :=
IntoAsLast(listaConversaciones, elementoListaConversaciones);
};
join Generacion_trama_anuncio;
endconnection datosDLC;
connection
Generacion_trama_anuncio :
task {
MACDestino := MACBroadcast;
MACOrigen := MiMAC;
TipoPaquete := AnuncioW2LAN_Tipo_Paquete;
call DatagramaSetMACDestino(datosMAC, MACDestino);
call DatagramaSetMACOrigen(datosMAC, MACOrigen);
call DatagramaSetTipoPaquete(datosMAC, TipoPaquete);
call DatagramaSetConversacionID(datosMAC, ConversacionID);
call DatagramaSetInternaMACDestino(datosMAC, MACDestinoEthernet);
call DatagramaSetInternaMACOrigen(datosMAC, MACOrigenEthernet);
call DatagramaSetInternaTipoPaquete(datosMAC, TipoPaqueteEthernet);
}
/*Inicia una conversacion*/;
decision Empty(listaTimers);
(true) :
task {
set(now + TimeoutSolicitud, TimerSolicitud);
SolicitudPendiente := True;
};
(false) :
enddecision;
task {
elementoListaTimers := NewObject;
elementoListaTimers!ref!e!listaTimers!t := now + TimeoutSolicitud;
listaTimers := IntoAsLast(listaTimers, elementoListaTimers);
elementoListaContadores := NewObject;
elementoListaContadores!ref!e!listaContadores!id := ConversacionID;
elementoListaContadores!ref!e!listaContadores!solicitudes := 0;
listaContadores := IntoAsLast(listaContadores, elementoListaContadores)
/*guarda tramaEthernet con el identificador de conversacion

```

```

en la lista de tramas pendientes */)
;
elementoListaTramasPendientes := NewObject;
elementoListaTramasPendientes!ref!e!listaTramasPendientes!id :=
ConversacionID;
elementoListaTramasPendientes!ref!e!listaTramasPendientes!trama :=
tramaEthernet;
listaTramasPendientes :=
IntoAsLast(listaTramasPendientes, elementoListaTramasPendientes);
};
output MACframe(datosMAC) via W2LANtoMAC;
nextstate permanente;
endconnection Generacion_trama_anuncio;
connection
datosMAC :
task {
call DatagramaGetTipoPaquete(datosMAC, TipoPaquete);
};
decision TipoPaquete;
(AnuncioW2LAN_Tipo_Paquete) :
join Anuncio;
(SolicitudW2LAN_Tipo_Paquete) :
join Solicitud;
(DatosW2LAN_Tipo_Paquete) :
join Datos;
else :
output DLCframe(datosMAC) via W2LANtoDLC;
nextstate permanente;
enddecision;
endconnection datosMAC;
connection
Anuncio :
task {
call DatagramaGetConversacionID(datosMAC, ConversacionID);
elementoListaConversaciones := FirstInQueue(listaConversaciones
/* Procesado de la cola. Si la conversacion esta en la lista
de conversaciones nada. Si no esta en la lista, si la lista
esta llena borrar el 1er elemento. Añadir nueva conversacion
en la cola de la lista con peticion pendiente, generar un datagrama
de Solicitud y volver a permanente */);
};
grst0 :
decision elementoListaConversaciones /= null;

```

```

(true) :
decision elementoListaConversaciones!ref!e!listaConversaciones!id =
ConversacionID;
(false) :
task elementoListaConversaciones :=
Successor(elementoListaConversaciones);
(true) :
nextstate permanente;
enddecision;
(false) :
join AnuncioII;
enddecision;
join grst0;
endconnection Anuncio;
connection
AnuncioII :
decision Cardinal(listaConversaciones) >= TamanyoListaConversaciones;
(true) :
task {
elementoListaConversaciones := FirstInQueue(listaConversaciones);
tempElementoListaConversaciones := elementoListaConversaciones;
elementoListaConversaciones := Successor(elementoListaConversaciones);
tempElementoListaConversaciones :=
DisposeObject(Remove(tempElementoListaConversaciones));
};
(false) :
enddecision;
task {
elementoListaConversaciones := NewObject;
elementoListaConversaciones!ref!e!listaConversaciones!id := ConversacionID;
elementoListaConversaciones!ref!e!listaConversaciones!pendiente := True;
call DatagramaGetInternaMACDestino(datosMAC, MACDestino);
call DatagramaGetInternaMACOrigen(datosMAC, MACOrigen);
call DatagramaGetInternaTipoPaquete(datosMAC, TipoPaquete);
elementoListaConversaciones!ref!e!listaConversaciones!MACDestino :=
MACDestino;
elementoListaConversaciones!ref!e!listaConversaciones!MACOrigen :=
MACOrigen;
elementoListaConversaciones!ref!e!listaConversaciones!TipoDatos :=
TipoPaquete;
listaConversaciones :=
IntoAsLast(listaConversaciones, elementoListaConversaciones);
};

```



```

task {
call DatagramaGetMACOrigen(datosMAC, MACDestino);
MACOrigen := MiMAC;
TipoPaquete := SolicitudW2LAN_Tipo_Paquete;
call DatagramaSetMACDestino(datosMAC, MACDestino);
call DatagramaSetMACOrigen(datosMAC, MACOrigen);
call DatagramaSetTipoPaquete(datosMAC, TipoPaquete);
call DatagramaSetConversacionID(datosMAC, ConversacionID);
};
output MACframe(datosMAC) via W2LANtoMAC;
nextstate permanente;
endconnection AnuncioII;
connection
Solicitud :
task {
call DatagramaGetConversacionID(datosMAC, ConversacionID);
elementoListaContadores := FirstInQueue(listaContadores
/* Si llega una Solicitud hasta aqui, significa que previamente se ha
generado un datagrama de Anuncio y alguien esta interesado en los datos.
Sumar 1 al contador de Solicitudes correspondiente de la lista de contadores
y volver a permanente. Si la solicitud llega fuera de la ventana de
solicitudes, no estara en la lista de contadores y se ignorara. */);
};
decision elementoListaContadores /= null;
(true) :
grst1 :
decision elementoListaContadores!ref!e!listaContadores!id = ConversacionID;
(false) :
task elementoListaContadores := Successor(elementoListaContadores);
join grst1;
(true) :
task elementoListaContadores!ref!e!listaContadores!solicitudes :=
elementoListaContadores!ref!e!listaContadores!solicitudes + 1;
grst2 :
nextstate permanente;
enddecision;
(false) :
join grst2;
enddecision;
endconnection Solicitud;
connection
Fin_Espera_Solicitudes :
task {

```

```

elementoListaTimers := FirstInQueue(listaTimers);
tempElementoListaTimers := elementoListaTimers;
elementoListaTimers := Successor(elementoListaTimers);
tempElementoListaTimers := DisposeObject(Remove(tempElementoListaTimers
/* Elimina el primer tiempo de la lista de timers. Si la lista
esta vacia, poner a false SolicitudPendiente. Elimina el primer
contador de la lista de contadores, y si es mayor que 0,
construir un datagrama de Datos y transmitirlo. */));
};
decision Empty(listaTimers);
(true) :
task SolicitudPendiente := False;
(false) :
set(elementoListaTimers!ref!e!listaTimers!t, TimerSolicitud);
enddecision;
task {
elementoListaContadores := FirstInQueue(listaContadores);
contador := elementoListaContadores!ref!e!listaContadores!solicitudes;
ConversacionID := elementoListaContadores!ref!e!listaContadores!id;
tempElementoListaContadores := elementoListaContadores;
elementoListaContadores := Successor(elementoListaContadores);
tempElementoListaContadores :=
DisposeObject(Remove(tempElementoListaContadores));
};
join Fin_Espera_Solicitudes_II;
endconnection Fin_Espera_Solicitudes;
connection
Fin_Espera_Solicitudes_II :
decision contador > 0;
(true) :
task {
elementoListaTramasPendientes := FirstInQueue(listaTramasPendientes);
};
grst3 :
decision elementoListaTramasPendientes!ref!e!listaTramasPendientes!id =
ConversacionID;
(false) :
task elementoListaTramasPendientes :=
Successor(elementoListaTramasPendientes);
join grst3;
(true) :
task {
MACDestino := MACBroadcast;

```

```

MACOrigen := ConversacionID;
TipoPaquete := DatosW2LAN_Tipo_Paquete;
call DatagramaSetMACDestino(datosMAC, MACDestino);
call DatagramaSetMACOrigen(datosMAC, MACOrigen);
call DatagramaSetTipoPaquete(datosMAC, TipoPaquete);
call DatagramaGetDatos
(elementoListaTramasPendientes!ref!e!listaTramasPendientes!trama,
CampoDatosEthernet);
call DatagramaSetDatos(datosMAC, CampoDatosEthernet);
};
output MACframe(datosMAC) via W2LANtoMAC;
grst4 :
task {
tempElementoListaTramasPendientes := elementoListaTramasPendientes;
elementoListaTramasPendientes :=
Successor(elementoListaTramasPendientes);
tempElementoListaTramasPendientes :=
DisposeObject(Remove(tempElementoListaTramasPendientes));
}
/*elimina trama de lista de tramas pendientes*/;
nextstate permanente;
enddecision;
(false) :
join grst4;
enddecision;
endconnection Fin_Espera_Solicitudes_II;
connection
Datos :
task {
call DatagramaGetConversacionID(datosMAC, ConversacionID);
elementoListaConversaciones := FirstInQueue(listaConversaciones);
};
decision elementoListaConversaciones /= null;
(true) :
grst5 :
decision elementoListaConversaciones!ref!e!listaConversaciones!id =
ConversacionID;
(false) :
task elementoListaConversaciones :=
Successor(elementoListaConversaciones);
join grst5;
(true) :
decision elementoListaConversaciones!ref!e!listaConversaciones!pendiente =

```

```

true;
(true) :
task {
elementoListaConversaciones!ref!e!listaConversaciones!pendiente :=
false;
tramaEthernet := datosMAC;
call DatagramaSetMACDestino
(tramaEthernet,
elementoListaConversaciones!ref!e!listaConversaciones!MACDestino);
call DatagramaSetMACOrigen
(tramaEthernet,
elementoListaConversaciones!ref!e!listaConversaciones!MACOrigen);
call DatagramaSetTipoPaquete
(tramaEthernet,
elementoListaConversaciones!ref!e!listaConversaciones!TipoDatos);
};
join Datos_II;
(false) :
enddecision;
grst6 :
nextstate permanente;
enddecision;
(false) :
join grst6;
enddecision;
endconnection Datos;
connection
Datos_II :
decision
(elementoListaConversaciones!ref!e!listaConversaciones!MACDestino =
MiMAC) or
(elementoListaConversaciones!ref!e!listaConversaciones!MACDestino =
MACBroadcast);
(true) :
output DLCframe(tramaEthernet) via W2LANtoDLC;
(false) :
enddecision;
task {
call DatagramaGetMACDestino(tramaEthernet, MACDestinoEthernet);
call DatagramaGetMACOrigen(tramaEthernet, MACOrigenEthernet);
call DatagramaGetTipoPaquete(tramaEthernet, TipoPaqueteEthernet);
};
join Generacion_trama_anuncio;

```

```
endconnection Datos_II;  
endprocess w2lanNode;
```

Apéndice B

Lista de Acrónimos

AP	Access Point
ASN.1	Abstract Syntax Notation number One
ATM	Asynchronous Transfer Mode
Bluetooth	low cost short-range wireless specification for connecting mobile devices
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DLC	Data Link Control
FDT	Formal Description Technique
GUI	Graphic User Interface
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ISO	International Standard for Organization
ITU	International Telecommunications Union
LAN	Local Area Network
MAC	Medium Access Control
MANET	Mobile Ad-Hoc network
MCDP-LAN	Multimedia Content Distribution Protocol over LAN
MSC	Message Sequence Charts

PC	Personal Computer
PHY	Capa Física OSI
PPV	Pay Per View
PUMM	Plataforma Universal Multimedia y Multiservicio
SDL	Specification and Description Language
SOHO	Small Office/Home Office
TTCN	Tree and Tabular Combined Notation
UML	The Unified Modeling Language
W2LAN	Wireless to LAN
Wi-Fi	Wireless Fidelity
WLAN	Wireless LAN

Índice de figuras

1.1. Ejemplo de una topología de red Ad-Hoc 802.11	16
2.1. Partes de distribución, broadcast y recepción	26
2.2. Diagrama de bloques del Set-Top-Box	28
2.3. Prototipo de <i>Set-Top-Box</i>	29
3.1. Diagrama de bloques de broadcast MCDP-LAN	42
3.2. Diagrama de bloques de recepción MCDP-LAN	43
4.1. Tramas W2LAN (Cabecera/Datos)	48
4.2. Diagrama de sistema	52
4.3. Proceso W2LAN (I)	53
4.4. Proceso W2LAN (II)	54
4.5. Proceso W2LAN (III)	55
4.6. Proceso W2LAN (IV)	56
4.7. Proceso W2LAN (V)	57
4.8. Proceso W2LAN (VI)	58
4.9. Proceso W2LAN (VII)	59
4.10. Proceso W2LAN (VIII)	60
4.11. Proceso W2LAN (IX)	61
4.12. Proceso W2LAN (X)	62
4.13. Proceso W2LAN (XI)	63
4.14. Proceso W2LAN (XII)	64
5.1. bloque W2LAN (un canal broadcast, N nodos)	67
5.2. El proceso Canal_broadcast (régimen transitorio) crea N procesos Nodo y les asigna sus posiciones	68
5.3. El proceso Canal_broadcast (régimen permanente) replica N veces cada trama	69
5.4. Topología lineal. Primer experimento.	71
5.5. Topología en malla	72

5.6. Ejemplo de topología aleatoria	73
5.7. Resultados para 2..10 nodos	74
5.8. Resultados para 4, 9, 16 y 25 nodos	75
5.9. Resultados para 1 .. 10 nodos	76
5.10. Curva de 9 nodos	77
6.1. Situación de bucle anómala	85
6.2. Situación correcta	86

Bibliografía

- [AGG99] G. Aggelou, R. Tafazolli, "RDMAR: A bandwidth-efficient routing protocol for mobile ad hoc networks." Proc. of the 2nd ACM International Workshop on Wireless Mobile Multimedia (WoWMoM), Seattle, 1999.
- [AKY02] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, "A Survey on Sensor Networks", IEEE Communications Magazine, agosto 2002.
- [ASN05] The ASN.1 Consortium, <http://www.asn1.org/>
- [BAK02] F. Baker, "An outsider's view of MANET", Internet Engineering Task Force document, marzo 2002.
- [BAT85] G. Bracha and S. Toueg, "Asynchronous consensus and broadcast protocols", journal of the ACM, vol. 32, pags. 824-840, octubre 1985.
- [BCG05] Rafaele Bruno, Marco Conti, Enrico Gregori, "Mesh Networks: Commodity Multihop Ad-Hoc Networks", IEEE Communications Magazine, marzo 2005.
- [BEI02] Nicklas Beijar, "Zone Routing Protocol", <http://www.tct.hut.fi/opetus/s38030/k02/Papers/08-Nicklas.pdf>
- [BMJ98] John Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, Jorjeta Jetcheva, "A Performance Comparison of Mutli-Hop Wireless Ad-Hoc Network Routing Protocols", Proc. of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98), octubre 1998.
- [BOI05] Berkeley Open Infrastructure for Network Computing, <http://boinc.berkeley.edu>
- [CBD02] Tracy Camp, Jeff Boleng, Vanessa Davies, "A Survey of Mobility Models for Ad Hoc Networks Research", Wireless Communications

- & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, 2002.
- [CCL03] I. Chlamtac, M. Conti, J. Liu, "Mobile Ad-Hoc Networks: Imperatives and Challenges", *Ad-Hoc Networks*, vol. 1, n° 1, pags. 13-64, julio 2003.
- [CCS80] R. Milner, "A Calculus of Communicating Systems", *Lecture Notes in Computer Science*, volumen 92, 1980.
- [CEP00] C. E. Perkins, E. M. Royer and S. R. Das, "Ad hoc on-demand distance vector (AODV) routing", *draft-ietf-manet-aodv-06.txt*, julio 2000.
- [CHI97] C. Chiang, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel", *Proc. IEEE SICON'97*, pags. 197-211, abril 1997.
- [CKP01] Christian Cachin, Klaus Kursawe, Frank Petzold, Victor Shoup, "Secure and Efficient Asynchronous Broadcast Protocols", IBM Research, Zurich Research Laboratory, marzo 2001.
- [COR95] M. S. Corson and A. Ephremides, "A Distributed Routing Algorithm for Mobile Wireless Networks", *ACM Journal for Wireless Networks*, pags 61-81, febrero 1995.
- [CPB94] C. Perkins and P. Bhagvat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers", *ACM Computer Communications Review*, pags. 234-244, octubre 1994.
- [DAT86] Normative EBU SP2, "World System Teletext and Data Broadcasting System. Technical Specifications", *Data Transfer Protocol*, Chapter 23, pags. 55-58, noviembre 1986.
- [DUB97] R. Dube et al., "Signal Stability based adaptive routing for Ad Hoc Mobile networks", *IEEE Pers. Comm.*, pags 36-45, febrero 1997.
- [DUT99] A. Dutta-Roy, "Fixed wireless routes for Internet access", *IEEE Spectrum*, pags. 61-69, septiembre 1999.
- [EGP01] D. Estrin, L. Girod, G. Pottie, M. Srivastava, "Instrumenting the world with wireless sensor networks", *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 2001)*, Salt Lake City, Utah, mayo 2001.

- [EMR99] E. M. Royer and C-K. Toh, "A review of current routing protocols for ad-hoc mobile wireless networks", *IEEE Personal Communications*, 6(2):46-55, abril 1999.
- [FBM02] Francesc Burrull, Francisco Miguel Monzo-Sanchez, Joan Garcia-Haro, Josemaria Malgosa-Sanahuja, "MCDP-LAN, An efficient Multimedia Content Distribution Protocol over LAN", *Proc. of the IASTED International Conference on Communications, Internet & Information Technology (CIIT2002)*, pags. 366-370, ISBN 0-88986-327-X, noviembre 2002.
- [FBM04] Francesc Burrull, Francisco Miguel Monzo-Sanchez, Josemaria Malgosa-Sanahuja, Joan Garcia-Haro, "W2LAN: Protocol that transforms a 802.11 mobile ad-hoc network (MANET) into an Ethernet LAN", *Proc. of the IASTED International Conference on Communications, Internet & Information Technology (CIIT2004)*, pags. 317-320, ISBN 0-88986-445-4, noviembre 2004.
- [FBM05] Francesc Burrull, Francisco Miguel Monzo-Sanchez, Josemaria Malgosa-Sanahuja, Joan Garcia-Haro, "Cost/Benefit analysis of the W2LAN Protocol", pendiente de publicación en *Proc. of the IASTED International Conference on Communications Systems and Networks (CSN2005)*, septiembre 2005.
- [FCC00] Fernando Cerdan, Josemaria Malgosa-Sanahuja, Joan Garcia-Haro, Francesc Burrull, Francisco Monzo Sanchez, "Quality of Service for TCP/IP traffic: An overview", *Proc. of the Protocols for Multimedia Systems 2000 (PROMS'2000)*, pp. 91-99, ISBN: 83-88309-05-6
- [FMS00-1] Francisco Monzó Sánchez, Francesc Burrull, Fernando Cerdán, Joan García-Haro, Josemaria Malgosa-Sanahuja, "MMUP. Multimedia and Multiservice Universal Platform", *Proc. of the 10th Mediterranean Electrotechnical Conference (MEleCon'2000)*, pp. 391-394, Volume I Regional Communication and Information Technology, ISBN 0-7803-6290-X, mayo 2000.
- [FMS00-2] Francisco Monzó Sánchez, Francesc Burrull, Fernando Cerdán, Joan García-Haro, Josemaria Malgosa-Sanahuja, "Design of a cost-effective multimedia and multiservice universal platform for the Information Society mass market", *Proc. of the 4th World Multiconference on Systemics, Cybernetics and Informatics (SCT'2000)* and the 6th International Conference on Information Systems Analysis

- and Synthesis (ISAS'2000), pp. 422-425, volume IV Communication Systems and Networks, ISBN 980-07-6690-1, julio 2000.
- [GAR99] J.J. Garcia-Luna-Aceves, M. Spohn, and D. Beyer, "Source tree adaptive routing (star) protocol". Internet Draft, draft-ietf-manet-star-00.txt, 22 octubre 1999.
- [GBU96] Pawel Gburzynki, "Protocol Design for Local an Metropolitan Area Networks", Ed. John Wiley & Sons, 1996, ISBN 0-13-554270-7
- [GDC04] G. Di Caro, F. Ducatelle, and L.M. Gambardella, "AntHocNet: an ant-based hybrid routing algorithm for mobile ad hoc networks", Proc. of Parallel Problem Solving from Nature (PPSN VIII), volumen 3242 de LNCS, págs. 461-470, Springer-Verlag, 2004.
- [GHE03] Gilbert Held, "Ethernet Networks, Design, Implementation, Operation and Management", Ed. Wiley, New York, 2003, ISBN 0-470-84476-0.
- [GUI04] A. M. Guirado-Puerta, F. Burrull, M. Escudero, P. Manzanares, J.C. Sánchez, Francisco Miguel.Monzo-Sanchez, "Peer-to-Peer flow analysys in an institutional region network", Proc. of the IASTED International Conference on Communications, Internet & Information Technology (CIIT2004), pags. 333-338, ISBN 0-88986-445-4, noviembre 2004.
- [HOL91] Gerard J . Holzmann, "Design and Validation of Computer Protocols", ISBN 0-13-539925-4, Prentice Hall, 1991.
- [HWP02] Dejan S Milojicic et al., "Peer to peer computing", Hewlett-Packard company 2002.
- [IEEE802] IEEE 802 LAN/MAN Standards Committee, <http://grouper.ieee.org/groups/802/>
- [802.3] IEEE 802.3 CSMA/CD (Ethernet) Working Group, <http://www.ieee802.org/3/>
- [IEE05] Project IEEE 802.11s, IEEE P802.11 - Task Group S, http://grouper.ieee.org/groups/802/11/Reports/tgs_update.htm
- [ISO83] ISO International Standard 7185, Programming Language - Pascal, ISO/TC97/SC6/WG4, 1983.

- [ISO89-E] ISO IS9074. Information processing systems -Open systems interconnection- "Estelle, A formal description technique based on an extended state transition model", Geneva: International Organization for Standardization, 1989
- [ISO89-L] ISO IS8807. Information processing systems -Open systems interconnection- "LOTOS, an FDT based on the Temporal Ordering of Observation Behaviour", Geneva, International Organization for Standardization, 1989.
- [ISO05] International Organization for Standardization, <http://www.iso.org/iso/en/ISOOnline.frontpage>
- [ITU05] International Telecommunications Union, <http://www.itu.int>
- [JAV05] The Java Technology, <http://java.sun.com>
- [JEG02] James E. Goldman, Phillip T. Rawles, "Applied Data Communications, third edition", 2002, ISBN 0-471-37161-0.
- [JGH99] J. Garcia-Haro, "Promoting the Information Society", IEEE Communications Magazine, Global Communications Newsletter, pág 3, septiembre 1999.
- [JMH03] David B. Johnson, David A. Maltz, Yih-Chun Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)", draft-ietf-manet-dsr-09.txt, abril 2003.
- [KUL03] V. V. Kuliainin, A. K. Petrenko, A. S. Kossatchev, I. B. Burdonov, "UniTesK: Model Based Testing in Industrial Practice", Proc. of the 1st European Conference on Model-Driven Software Engineering, Nuremberg. Germany, 2003.
- [LIN05] Linux Online, <http://www.linux.org>
- [MAL02] J. Malgosa Sanauha, J. Garcia Haro, F. Cerdan, F. Burrull Mestres, "MUST, a Multicast Synchronous Transfer Application for Fast Intra-Campus Replication", Proc. of the 11th IEEE Mediterranean Electrotechnical Conference, Melecon2002, pags. 100-104, Cairo, Egipto, Mayo 2002. ISBN 0-7803-7527-0
- [MCC99] J. C. McCarthy, "The Social Impact of Electronic Commerce", IEEE Communications Magazine, pags. 53-57, Septiembre 1999.
- [MIV05] http://www.mivision.cyta.com.cy/english/periexi_tmessaging.php

- [MUR96] S. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," ACM Mobile Networks and Applications Journal, Special issue on Routing in Mobile Communication Networks, Vol. 1, No. 2, 1996.
- [MWG05] Mobile Ad-hoc Networks (manet), <http://www.ietf.org/html.charters/manet-charter.html>
- [OHT99] K. Ohta, S. Kurihara, "Encryption techniques for broadcast communication", Proc. of the IEEE Globecom'91, pp. 195-200.
- [OWN04] Tomas Krag and Sebastian Büttrich, "Wireless Mesh Networking", O'Reilly Wireless DevCenter, <http://www.oreillynet.com/wireless/>
- [PET03] O. Petrenko, V. Omelchenko, "Rapid Trainings on Specification Based Testing Tools", Proceedings of 1st South-East European Workshop on Formal Methods, Grecia, 2003.
- [QIN04] Liang Qin, Thomas Kunz, "Technical Report SCE-04-14", Carleton University, Systems and Computer Engineering, August 2004
- [SDL05] SDL Forum Society, <http://www.sdl-forum.org>
- [SKY05] Skype Technologies S.A., <http://www.skype.com>
- [TIC05] Departamento de las Tecnologías de la Información y las Comunicaciones, Universidad Politécnica de Cartagena, <http://www.upct.es>
- [TLG05] Telelogic AB, Box 4128, S-203 12 Malmö, Sweden, <http://www.telelogic.com>
- [TTC05] The European Telecommunications Standards Institute (ETSI) Protocol and Testing Competence Centre, <http://www.etsi.org/ptcc/>
- [UML05] Object Management Group, UML, <http://www.uml.org/>