

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL IPN

Departamento de Ingeniería Eléctrica

Sección de Computación



Lucrn: Un middleware para el desarrollo de aplicaciones
en redes espontáneas de dispositivos móviles Bluetooth

TESIS DE MAESTRÍA

Jorge Alfonso Briones García

Supervisada por: Dr. José Oscar Olmedo Aguirre

Mayo 2004

Contenido

Agradecimientos	VII
Resumen	IX
1. Introducción	1
1.1. Dispositivos móviles y tecnología inalámbrica	1
1.2. Planteamiento del problema	4
1.2.1. Planeación y configuración de la red	4
1.2.2. Nodos móviles	5
1.2.3. Dispositivos de recursos limitados	5
1.3. Objetivo	7
1.4. Contribuciones	9
1.5. Organización del documento	10
2. Computación móvil y Bluetooth	13
2.1. Computación móvil	13
2.2. Redes Ad-Hoc	14
2.3. Redes Espontáneas	15
2.4. Principales problemas de las redes espontáneas	16
2.5. Tecnología inalámbrica Bluetooth	18
2.5.1. La pila de protocolos	21
2.5.2. El medio de comunicación	22
2.5.3. Formación de la piconet	23
2.5.4. Mecanismos para el ahorro de energía	26
2.5.5. Descubrimiento de servicios	27
2.5.6. Perfiles	29
2.6. Conclusiones	31

3. El middleware Lucrn	33
3.1. Una vista general del middleware	33
3.2. Las capas del middleware Lucrn	35
3.2.1. Formato de los mensajes	38
3.2.2. Protocolo de anuncio de dispositivos	39
3.2.3. Protocolo de ruteo	45
3.2.4. Protocolo de movilidad	47
3.2.5. Mecanismos de colaboración	52
3.3. Resumen	59
4. Implementación del middleware	61
4.1. Selección de la plataforma	61
4.2. Ambientes de desarrollo	63
4.3. Implementación en Mac OS X	65
4.4. Diagrama de clases	69
4.5. Implementación en Palm OS	72
4.5.1. Diseño de aplicaciones para Palm	72
4.5.2. Componentes de la implementación	73
4.6. Resumen	77
5. Pruebas y Resultados	79
5.1. Escenarios que pueden ser contruidos con Lucrn.	79
5.2. Aplicaciones desarrolladas con Lucrn	82
5.2.1. Un sistema de envío de mensajes	82
5.2.2. Un pizarrón electrónico compartido	87
5.2.3. Un control remoto para la aplicación iTunes	90
5.3. Resultados	91
5.4. Resumen	92
6. Conclusiones	93
6.1. Perspectivas a futuro	95
A. La pila de protocolos en Mac OS X y Palm OS	97
A.1. La pila de protocolos en Mac OS X	97
A.1.1. IOBluetooth.framework	98
A.1.2. IOBluetoothUI.framework	99
A.2. La pila de protocolos de Bluetooth en Palm OS	100

Lista de Figuras

1.1. Capacidad máxima de memoria de una Apple iMac G4 y un asistente digital Palm m515.	2
1.2. Rango de comunicación de una red de área personal.	3
2.1. Una red ad-hoc entre 3 computadoras portátiles.	15
2.2. Comparación de diferentes estándares inalámbricos de la familia IEEE 802.	19
2.3. La relación entre las capas OSI, las capas IEEE 802 y Bluetooth.	20
2.4. Pila de protocolos de Bluetooth.	22
2.5. Estados operacionales de un dispositivo Bluetooth.	24
2.6. Mecanismo de acceso al medio (FH/TDD).	25
2.7. Un ejemplo de una scatternet.	26
2.8. Perfil de transferencia de archivos y objetos en Bluetooth.	30
3.1. El middleware en un contexto de alto nivel.	34
3.2. Los protocolos que componen al middleware Lucrn.	36
3.3. Encabezado de los mensajes.	39
3.4. Un mensaje de datos.	40
3.5. Formato del mensaje LUCRN_NEW_DEVICE.	41
3.6. Formato del mensaje LUCRN_SYNC_DEVICES.	42
3.7. Primera parte del diagrama de flujo del protocolo de anuncio de dispositivos.	43
3.8. Segunda parte del diagrama de flujo del protocolo de anuncio de dispositivos.	44
3.9. Formato del mensaje LUCRN_DATA_MSG.	47
3.10. Diagrama de flujo del protocolo de ruteo.	48
3.11. Nodos simples y complejos.	50
3.12. Formato del mensaje LUCRN_DISCONNECTING_DEVICE.	51

3.13. 1a. parte del diagrama de flujo del protocolo de movilidad. . .	53
3.14. 2a. parte del diagrama de flujo del protocolo de movilidad. . .	54
3.15. Un pizarrón electrónico compartido por varios usuarios.	55
3.16. Formato de un mensaje LUCRN_COLLECTION.	57
3.17. Diagrama de flujo del mecanismo de colaboración.	58
4.1. Componentes principales de la implementación en Mac OS X.	66
4.2. Diagrama de clases.	70
4.3. Componentes principales de la implementación en Palm OS.	74
4.4. Conexiones entre los componentes.	76
5.1. La interfaz de usuario del sistema de envío de mensajes en el sistema Mac OS X y Palm OS.	83
5.2. Buscando dispositivos que ofrezcan el servicio de envío mensajes en Palm OS.	85
5.3. Los menús Pop-Up muestran los dispositivos que componen la red.	86
5.4. Intercambiando mensajes entre los diferentes dispositivos.	86
5.5. Integración de un nuevo dispositivo.	87
5.6. La interfaz de usuario del pizarrón compartido en Mac OS X.	88
5.7. Intercambiando figuras geométricas entre dispositivos móviles.	89
5.8. La interfaz de usuario del control remoto del iTunes en Palm.	90
5.9. Buscando un dispositivo.	91
A.1. Pila de Protocolos de Bluetooth en Mac OS X.	98
A.2. Componentes de la implementación de Bluetooth en Palm.	100

Agradecimientos

Este trabajo está dedicado especialmente a mis padres y a mi hermano, de quienes he recibido y continuo recibiendo enseñanzas las cuales son invaluableles y me han servido como una luz que siempre me acompaña e ilumina mi camino.

También está dedicado a mi esposa Silvana con quien siempre estoy iniciando sueños y juntos hemos logrado convertirlos en realidad.

De manera especial agradezco al Dr. Jose Oscar Olmedo Aguirre por haber creído, apoyado y enriquecido este proyecto.

De igual forma agradezco al Dr. Sergio V. Chapa Vergara y al Dr. Adriano de Luca Pennacchia por el tiempo invertido en la revisión de este documento y por sus valiosos comentarios para mejorar el mismo.

Finalmente el agradecimiento al CINVESTAV y en especial al Departamento de Aplicación de Microcomputadoras de la Universidad Autónoma de Puebla por el apoyo brindado para la culminación de este trabajo.

Resumen

Con la reciente proliferación de dispositivos móviles –asistentes digitales, computadoras portátiles– equipados con tecnología inalámbrica Bluetooth es posible la creación de redes *ad-hoc*, es decir, redes de dispositivos inalámbricos los cuales cooperan para formar una red sin depender de una infraestructura física o de servicios. Estas redes han sido propuestas para una gran variedad de escenarios: militares, zonas de desastre, colaboración digital, etc. Sin embargo, para las aplicaciones la movilidad de los nodos provoca que se tengan que enfrentar a nuevos problemas y su ejecución en este entorno se vuelva compleja.

Considerando esta problemática, este documento explica el diseño y la implementación del middleware *Lucrn* que está compuesto por mecanismos: para anunciar la entrada y la salida de un nodo simple de la red, para el intercambio de información en una red de canales de comunicación punto a punto y para facilitar el desarrollo de aplicaciones de colaboración, los cuales, en conjunto, proporcionan a las aplicaciones de los dispositivos móviles los elementos para enfrentar una topología de red dinámica.

El middleware diseñado extiende la pila de protocolos de Bluetooth y fue implementado en los sistemas operativos *Mac OS X* y *Palm OS*, por tanto, puede ser instalado en computadoras de escritorio, portátiles y en asistentes digitales que estén equipados con tecnología Bluetooth y que utilicen los sistemas operativos antes mencionados. De igual forma, se explican las respectivas implementaciones y se muestran las aplicaciones que fueron desarrolladas entre las que destaca un pizarrón electrónico compartido que trabaja en una red espontánea de computadoras portátiles y asistentes digitales.

Abstract

With the recent proliferation of mobile devices –portable digital assistants (PDAs), computer laptops– equipped with Bluetooth wireless technology is possible to create *ad-hoc* networks, a collection of wireless nodes that configure themselves to form the network without the support of any planned infrastructure. These networks have been proposed for a variety of scenarios: military, disaster zones, entertainment, business, etc. However their construction impose several challenges for the development of applications due to the lack of mechanisms for the casual creation of the network, mobility of nodes and exchange of information.

In this thesis we present the design and implementation of Lucrn, a middleware comprising a suite of protocols that announce the creation and destruction of communication links, and enable information exchange among the nodes of a point to point network. These mechanisms as a whole provide to the applications the means to support a mobile computing environment.

Lucrn extends the Bluetooth protocol stack to facilitate the development of applications that interoperate in spontaneous networks. Currently it has been implemented in Mac OS X and Palm OS platforms and has been used to create several applications one of them is a shared whiteboard that operates between Bluetooth enabled laptops and personal digital assistants. The middleware Lucrn addresses the urgent need to provide interoperability among applications based on heterogenous mobile devices therefore is feasible the creation of new application domains.

Capítulo 1

Introducción

Los últimos avances en la tecnología de integración a gran escala, la microelectrónica y las telecomunicaciones han permitido la construcción y la proliferación de una gran variedad de dispositivos móviles equipados con tecnología inalámbrica. Esta convergencia entre dispositivos móviles y tecnología inalámbrica dio origen a nuevos escenarios de cómputo pero también introdujo nuevos problemas para las aplicaciones que se ejecutan en este entorno.

Este capítulo está organizado de la siguiente manera: en la sección que sigue se da una breve introducción a dispositivos móviles y tecnología inalámbrica, posteriormente se plantea el problema que este trabajo resuelve para después exponer los objetivos, las contribuciones y la organización global de este documento.

1.1. Dispositivos móviles y tecnología inalámbrica

Dispositivos como computadoras portátiles y asistentes digitales poco a poco han comenzado a formar parte de la vida cotidiana. A tal grado que es común identificar el uso de estos dispositivos por una gran variedad de personas y en diferentes tipos de escenarios.

El éxito de ellos radica principalmente en su capacidad para procesar y almacenar información digital. Esto conjugado con que son ligeros y pequeños da lugar a que los usuarios puedan crear, procesar y almacenar información digital en cualquier lugar y momento. En la actualidad compañías como Palm

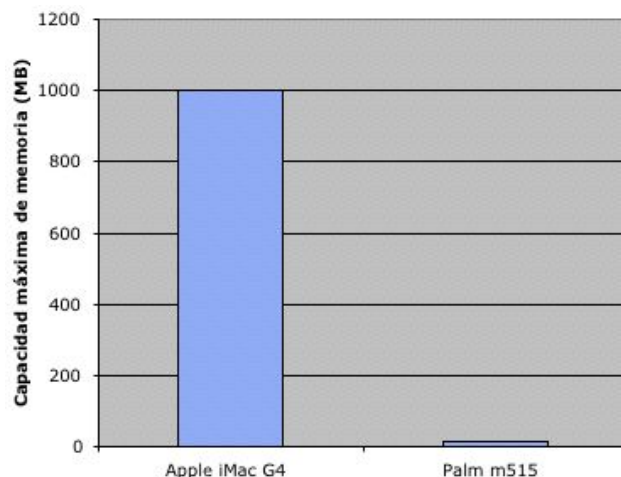


Figura 1.1: Capacidad máxima de memoria de una Apple iMac G4 y un asistente digital Palm m515.

han logrado vender más de 24,000,000 de asistentes digitales a nivel mundial [1]. Estos dispositivos tienen menos recursos con respecto a cantidad de memoria y poder de procesamiento que computadoras de escritorio o portátiles. Por ejemplo, en la figura 1.1, se presenta la capacidad de memoria máxima que puede tener una computadora *Apple iMac G4* con respecto a un asistente digital modelo *Palm m515*, se puede observar que la diferencia es notable y esto trae consigo algunas dificultades para la programación y el rendimiento de las aplicaciones.

Por otro lado, hasta hace algunos años la capacidad de estos dispositivos para transferir información hacia otros era muy limitada, la mayoría de las veces se utilizaban cables y el respectivo puerto serial de la computadora a la cual se le va a transferir la información. Esto ocasionó que se les considerara dispositivos digitales con pocas capacidades para la transferencia de información y por ello inútiles para diversas tareas. Las compañías se dieron cuenta que el futuro de estos dispositivos dependía de sus capacidades para transferir información a cualquier dispositivo digital pero sin tener que depender de una infraestructura física.

Esta necesidad de transferencia de información entre dispositivos que se encontraban a una corta distancia sin la necesidad de usar cables, sirvió para

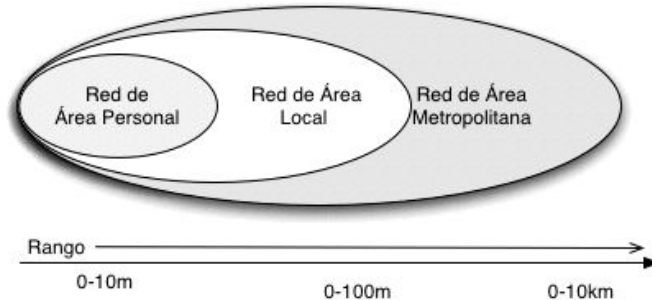


Figura 1.2: Rango de comunicación de una red de área personal.

que muchos centros de investigación y compañías se enfocaran al desarrollo de tecnologías de comunicación inalámbricas de corto alcance [2] las cuales dieron origen a las *redes de área personal*.

En la figura 1.2 se puede observar que las redes de área personal tienen un rango de comunicación de 10m, a diferencia, de las redes de área local y de área metropolitana que tienen un alcance de 100m y 10km respectivamente.

Una tecnología inalámbrica que en la actualidad sirve para la creación de redes de área personal y ha sido adoptada por una gran variedad de compañías e integrada en diversos dispositivos es *Bluetooth* [3]. Entre los dispositivos encontramos: teléfonos celulares, computadoras portátiles y asistentes digitales. Con respecto a las compañías se encuentran principalmente: Ericsson, Intel, IBM, Nokia, Toshiba, Microsoft, Motorola, 3Com, Lucent, Palm y Apple. En realidad existe el grupo de interés de Bluetooth (Bluetooth SIG) en el cual se tienen registradas más de 1200 compañías.

La incorporación de esta tecnología inalámbrica en dispositivos móviles permite que ahora sea posible la transferencia de información entre ellos sin el uso de cables. Dando lugar a que se pueda realizar el intercambio de información en cualquier lugar y momento sin tener que depender de una infraestructura física adicional. Esto permite pensar en nuevas aplicaciones las cuales no tengan que restringirse al modelo estático tradicional en el cual las computadoras permanecen inmóviles. Sin embargo, como vamos a ver en la siguiente sección, esta característica trae cambios significativos con respecto a los mecanismos que ahora las aplicaciones deben tener para tolerar una topología de red dinámica.

1.2. Planteamiento del problema

Es importante resaltar que este trabajo se fundamenta principalmente en el área de computo móvil, en la cual se ha definido el término de red *ad-hoc* para identificar a una red de dispositivos móviles que no depende de una infraestructura para su formación y, a su vez, se ha definido el concepto de *red espontánea* para indicar que los nodos móviles son asistentes digitales y computadoras portátiles y la tecnología inalámbrica es Bluetooth u 802.11b [6]. Una discusión más amplia acerca de estas redes y sus implicaciones se da en el capítulo 2.

Tomando lo anterior en cuenta y haciendo uso de la tecnología actual, la idea fundamental de este trabajo es aprovechar la proliferación de los dispositivos móviles equipados con tecnología inalámbrica Bluetooth para formar redes de dispositivos en cualquier lugar y momento. Luego, una vez que se ha formado la red, dotar a las aplicaciones de mecanismos para que puedan interoperar de una manera transparente y puedan responder adecuadamente a los eventos que suceden en la red como la entrada o la salida de algún nodo.

A partir de esta idea podemos identificar ciertos problemas los cuales no están presentes en las redes tradicionales, por ejemplo, en las redes de área local basadas en el estándar *IEEE 802.3*. Los cuales discutiremos a continuación.

1.2.1. Planeación y configuración de la red

El primero de los problemas que discutiremos trata con la planeación y configuración de la red. En una red de dispositivos móviles inalámbricos la cual se puede formar en cualquier lugar y momento no se puede determinar cuales nodos formarán parte de la red. Esto tiene consecuencias sobretodo porque antes de poder intercambiar información los nodos deben ser configurados. Otra implicación notable es que no se sabe cuales servicios son proporcionados en la red ni los dispositivos que en determinado momento los estén ofreciendo.

Para ilustrar mejor este problema pongamos el ejemplo de una red tradicional TCP/IP basada en el estándar *IEEE 802.3* (Ethernet). La puesta en marcha de esta red comprende un proceso de planeación y configuración el cual es llevado a cabo por el administrador de la red. En la planeación se hace un análisis acerca de:

- El crecimiento de la red.
- El tipo de red que se requiere, pública o privada.
- Los servicios que van a ser proporcionados en la red y cuales computadoras serán las responsables de ofrecer estos servicios.

Una vez analizados estos aspectos se procede a realizar:

- La configuración de las tarjetas de red.
- La configuración de las tablas de ruteo.
- La instalación del software necesario para que las computadoras estén preparadas para usar y ofrecer servicios.

Si se tuviera que realizar este proceso de planeación y configuración para una red que deseamos que se forme en cualquier lugar y momento, sería bastante engorroso y complejo para los usuarios no expertos en redes. Considerando esto es necesario crear mecanismos para solventar este problema. Como veremos en el capítulo 3 el middleware diseñado hace una combinación de los mecanismos de Bluetooth con los mecanismos propuestos y resuelve este problema.

1.2.2. Nodos móviles

Otro problema es el hecho de que los nodos de la red son móviles, por tanto, la topología de la red cambia frecuentemente. Este problema trae consecuencias sobretodo a las aplicaciones y al sistema ya que éste debe reconfigurar las tablas de ruteo. En el middleware se proponen tres protocolos para resolver parte de esta problemática.

1.2.3. Dispositivos de recursos limitados

Otro hecho relevante es la capacidad de los nodos en la red. Los dispositivos móviles tienen menos capacidades con respecto a memoria, poder de procesamiento y energía que aquellos que encontramos en las redes tradicionales los cuales tienen más memoria, procesadores más poderosos y están conectados siempre a una fuente de energía eléctrica.

Otro aspecto es que la programación de dispositivos móviles como asistentes digitales cambia con respecto a los dispositivos de cómputo estáticos.

Las principales consideraciones que se tienen que tomar al momento de realizar la programación en dispositivos móviles pequeños son [4]:

- El tamaño de la pantalla. (160x160 pixeles es el tamaño de la pantalla de una Palm m515).
- No se tienen periféricos adicionales como un teclado o un ratón.
- Tienen recursos limitados de memoria y procesamiento.
- Tienen una fuente de energía limitada.

Esto trae implicaciones sobre todo al momento de diseñar un middleware ya que debe ser ligero para que pueda ser instalado en alguno de estos dispositivos.

A partir de los problemas señalados podemos inferir que las redes inalámbricas de dispositivos móviles que se pueden formar en cualquier lugar y momento ocurren:

- De una manera casual.
- Entre dispositivos heterogéneos con recursos limitados.
- Entre dispositivos con distintos sistemas operativos.
- En lugares donde posiblemente no exista una infraestructura eléctrica.
- En lugares donde posiblemente no se pueda utilizar una infraestructura de comunicaciones basadas en cable.
- En lugares donde no se puede hacer uso o no existen servicios computacionales como un servidor DNS o DHCP.

Un problema adicional está relacionado con la tecnología Bluetooth ya que ésta únicamente proporciona mecanismos para generar enlaces punto a punto y punto a multipunto, esto trae como consecuencia que para enviar un mensaje entre dispositivos que no son adyacentes cada uno de los nodos necesita tener una tabla de ruteo la cual va a consultar para dirigir el mensaje a su destino. Ahora, para la generación y configuración de esta tabla es necesario realizar acciones cuando un dispositivo se integra o sale de la red.

En particular, en este trabajo se abordan los siguientes problemas:

- *La integración de un nodo a la red de dispositivos Bluetooth.*
- *La transferencia de información entre todos los dispositivos Bluetooth de la red.*
- *La salida de los nodos que tienen únicamente un enlace inalámbrico Bluetooth.*
- *La colaboración entre las aplicaciones de los dispositivos Bluetooth.*

1.3. Objetivo

Habiendo expuesto la problemática, el objetivo general del proyecto fue:

El diseño de una capa intermedia (middleware), formada por un conjunto de protocolos, que permita a las aplicaciones interoperar en una red inalámbrica de dispositivos móviles heterogéneos equipados con tecnología Bluetooth.

Para cumplir con este objetivo se diseñaron un conjunto de protocolos que extienden la pila de Bluetooth, aprovechan sus características tecnológicas, y proporcionan servicios mejorados a las aplicaciones para que puedan lidiar con los cambios frecuentes en la topología de la red.

Los protocolos diseñados fueron los siguientes:

- *Un protocolo de anuncio de dispositivos el cual sirve para que todos los dispositivos sean informados acerca de la creación de un nuevo enlace de comunicación inalámbrico.*

Este protocolo emplea los mecanismos de descubrimiento de dispositivos y de servicios proporcionados por Bluetooth. De esta manera cada vez que un nuevo dispositivo establece un canal con otro el mecanismo de anuncio de dispositivos informa de este evento a los dispositivos adyacentes y así sucesivamente.

- *Un protocolo que permite el intercambio de información entre todos los nodos que componen la red.*

Este protocolo se encuentra encima del protocolo de anuncio de dispositivos y a partir de los eventos que ocurren en el nivel inferior va actualizando las tablas de ruteo de los diferentes dispositivos.

- *Un protocolo que anuncia la salida de un nodo simple (i.e. un nodo que sólo tiene un enlace inalámbrico).*

Este protocolo tiene por objetivo avisar acerca de la salida de un nodo simple para que se actualicen las tablas de ruteo de los diferentes dispositivos.

- *Un protocolo de colaboración que permite el envío de mensajes de alto nivel entre las diferentes aplicaciones.*

Este es un protocolo de alto nivel el cual se construye encima de todos los anteriores para facilitar el desarrollo de aplicaciones colaborativas.

La metodología que se utilizó para el desarrollo de estos protocolos fue la tradicional [18] en la cual se debe especificar:

- *El servicio que ofrece el protocolo.*
- *Las suposiciones acerca del ambiente en el cual se va a ejecutar.*
- *Los mensajes válidos del protocolo.*
- *El formato de los mensajes.*
- *Las reglas que mantienen la consistencia del intercambio de mensajes que se lleva a cabo entre las diferentes entidades.*

La especificación de estos protocolos permite que éstos puedan ser implementados en cualquier sistema operativo que tenga la pila de protocolos de Bluetooth, además, se garantiza la compatibilidad entre las implementaciones.

Sin embargo para poder realizar el diseño y la implementación de estos protocolos se tuvo que cumplir lo siguiente:

- El estudio de la tecnología inalámbrica Bluetooth.
- El empleo de las interfaces de programación (API) de cada uno de los sistemas operativos Mac OS X y Palm OS.
- El estudio de los diferentes ambientes de programación para dispositivos móviles.

Una vez diseñados los protocolos se paso a su implementación en los sistemas operativos Mac OS X y Palm OS.

Finalmente, con las implementaciones se desarrollaron las siguientes aplicaciones:

- Un pizarron electrónico compartido.
- Un sistema de envío de mensajes .
- Un aplicación que permite controlar la aplicación *iTunes* desde un asistente digital.

1.4. Contribuciones

La contribución principal de este trabajo es una capa de intermedia de software que permite a las aplicaciones de los diferentes dispositivos móviles interoperar en una red inalámbrica de dispositivos Bluetooth.

Esta capa intermedia fue implementada en Mac OS X y Palm OS por tanto puede ser y es utilizada para la programación de aplicaciones las cuales funcionarán en este ambiente dinámico. Cabe resaltar que durante el transcurso de este trabajo no se tiene ninguna referencia acerca de algún middleware comercial que proporcione los mecanismos antes mencionados.

A partir de esta capa intermedia es posible la creación de nuevas aplicaciones las cuales involucren escenarios diferentes a los tradicionales en los cuales es posible formar una red de dispositivos en cualquier lugar. En particular con este trabajo se pueden mejorar los siguientes escenarios:

- *Industriales*

El sector industrial es uno de los sectores en los cuales es probable que se lleven a cabo varias reuniones a lo largo del día y varias de ellas involucren la colaboración de todos sus empleados para realizar alguna tarea. También es muy común encontrar a una gran cantidad de personas llevando consigo computadoras portátiles y asistentes digitales a esas reuniones. La mayoría de estas reuniones se llevan a cabo entre grupos de personas que están participando en algún proyecto y en las cuales todos colaboran para la definición de un proceso de negocio o la edición de un plan de trabajo que se llevará a cabo en las próximas semanas o meses. En la actualidad estos escenarios de colaboración

generalmente usan herramientas como pizarrones y hojas de papel. Al final de la sesión alguna persona es designada para transcribir lo que se realizó en la sesión de colaboración a un formato digital para su posterior envío a través de correo electrónico a todos los asistentes.

Con este middleware es posible que los participantes de la reunión utilicen los dispositivos móviles que llevan consigo, formen una red y a través de las aplicaciones colaboren en el diseño de algún proyecto y tenga la información digital inmediatamente que termina la reunión.

- *Entretenimiento*

Con este middleware es posible la creación de aplicaciones de entretenimiento las cuales permitan a los usuarios de diferentes dispositivos móviles crear una red en cualquier lugar y participar en una actividad recreativa.

- *Zonas de desastre*

Durante desastres es común que dejen de operar las plantas de energía eléctrica y la infraestructura de comunicaciones. Sin embargo, con este middleware es posible la creación de aplicaciones en las cuales los elementos de protección civil utilicen sus dispositivos móviles para formar una red y colaboren a través de las aplicaciones para trazar planes de acción.

1.5. Organización del documento

La estructura general del presente documento queda como sigue: El capítulo 2 se dedica a la comunicación inalámbrica, en particular, se describen las principales características de Bluetooth, enfatizando los mecanismos novedosos que son utilizados en este trabajo. También se hace una comparación con respecto a movilidad, poder de transmisión, consumo de energía de Bluetooth con otros estándares inalámbricos. Se presentan algunas de las limitaciones que tiene Bluetooth para el desarrollo de aplicaciones distribuidas.

El capítulo 3 corresponde a la parte medular de este trabajo. En él se especifican los protocolos de anuncio de dispositivos, ruteo, movilidad y colaboración que en conjunto forman al middleware Lucrn. Para cada uno de ellos se describe el objetivo del servicio, las suposiciones, los mensajes válidos, el formato de los mensajes, las reglas y un diagrama de flujo.

El capítulo 4 se dedica a explicar la implementación del middleware en los sistemas operativos Mac OS X y Palm OS. En él se presenta el hardware utilizado, el ambiente de desarrollo que fue empleado y los diagramas de clases en UML de las diferentes implementaciones.

El capítulo 5 presenta las aplicaciones que fueron desarrolladas con el middleware Lucrn. Resalta un pizarrón electrónico colaborativo y una aplicación que es controlada en forma remota desde un asistente digital. En este capítulo se dedica una sección, al final, para explicar cada uno de los resultados que fueron obtenidos por el diseño y la implementación de este middleware.

El capítulo 6 presenta las conclusiones de este trabajo y las perspectivas a futuro del mismo.

Adicionalmente, en el apéndice A, se explica la pila de protocolos de Bluetooth de los sistemas operativos Mac OS X y Palm OS. Además, se describe brevemente las interfaces de programación (API's) que fueron utilizadas para la implementación del middleware.

Capítulo 2

Computación móvil y Bluetooth

Este capítulo presenta el estado actual de la tecnología inalámbrica Bluetooth, además, dado que este trabajo se sitúa dentro del área de cómputo móvil, la primera parte da una introducción a ésta, presenta sus características e introduce las redes *Ad-hoc* y las redes *espotáneas*. La segunda parte presenta a Bluetooth su historia, su pila de protocolos y mecanismos novedosos e importantes que fueron utilizados.

2.1. Computación móvil

Con el advenimiento de los dispositivos móviles y la incorporación de tecnología inalámbrica a cada uno de ellos. Se creó una nueva área que se conoce como *computación móvil*. La cual trata principalmente de identificar y resolver problemas que existen cuando los dispositivos son móviles, limitados de recursos y el medio de comunicación es inalámbrico.

Las características inherentes de la computación móvil son [5]:

- *Los elementos móviles son de recursos limitados.*

Se refiere al hecho de que los dispositivos móviles tendrán recursos limitados con respecto a la velocidad del procesador, la cantidad de memoria y la capacidad del disco.

- *La movilidad envuelve ciertos riesgos.*

Se refiere al hecho de que los dispositivos móviles son más vulnerables a ser dañados, perdidos o robados.

- *El medio de comunicación es variable en términos de rendimiento y confiabilidad.*

Se refiere al hecho de que se tengan variaciones con respecto a la calidad del medio de comunicación inalámbrico. Probablemente por la interferencia que pueda existir con otros dispositivos electrónicos.

- *Los elementos móviles dependen de una fuente limitada de energía.*

Se refiere al hecho de que los dispositivos utilizan baterías como su fuente de energía.

Cabe resaltar que muchas de estas características no están presentes en los escenarios basados en redes tradicionales.

Una de las principales ventajas que ofrecen los dispositivos móviles equipados con tecnología inalámbrica es su capacidad de poder formar ligas de comunicación entre diferentes dispositivos, de una manera casual, es decir, de forma no planeada o no predeterminada.

Dentro del área de cómputo móvil encontramos un tipo especial de redes inalámbricas que se conocen como *redes ad-hoc*. Estas redes ad-hoc servirán para formar enlaces de comunicación espontáneos entre todos los dispositivos como se explica en la siguiente sección.

2.2. Redes Ad-Hoc

Una *red ad-hoc* es una colección de nodos inalámbricos los cuales cooperan para formar una red que opera sin la ayuda de infraestructura de comunicaciones física [6]. Las redes ad-hoc en la actualidad son un área de investigación muy activa dentro del cómputo móvil. En la figura 2.1 se muestra una pequeña red ad-hoc compuesta de tres nodos inalámbricos los cuales cooperan para formar la red sin el apoyo de ninguna infraestructura física como lo serían cables o en el caso de 802.11b un punto de acceso inalámbrico.

Estas redes han sido propuestas para una gran variedad de escenarios. Por ejemplo, arreglos de sensores para la recolección de información, para proporcionar una infraestructura de comunicación en áreas hostiles o de desastre, y en especial han tomado mucho auge debido a que son la infraestructura

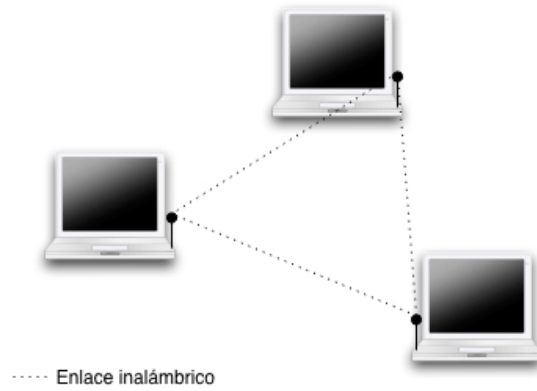


Figura 2.1: Una red ad-hoc entre 3 computadoras portátiles.

de comunicación para el área de *computación ubicua*[7] donde se trata principalmente con ambientes los cuales están saturados de dispositivos móviles inteligentes.

Las redes ad-hoc en forma general pueden tratar con miles de nodos inalámbricos. Existen una gran cantidad de trabajos al respecto, varios de ellos tratan con el problema de ruteo de información. Las redes ad-hoc en las cuales sus nodos trabajan como estaciones (hosts) y como ruteadores (routers) se conocen como redes ad-hoc de múltiples saltos (multihop ad-hoc networks).

2.3. Redes Espontáneas

Para este trabajo utilizaremos un subconjunto de las redes ad-hoc las cuales se van a generar cuando un grupo de personas se reúnen y usan sus dispositivos inalámbricos para realizar alguna actividad colaborativa digital. A este pequeño subconjunto de las redes ad-hoc se les da el nombre de *redes espontáneas*[6].

En la actualidad es común encontrar que se utiliza el término de red ad-hoc y de red espontánea como si fueran el mismo concepto. Sin embargo, son diferentes ya que las redes espontáneas son un caso especial de las redes

ad-hoc. Las cuales tienen características bien definidas [6]:

- Ocurren en espacio y tiempo bien delimitados.
- Los nodos inalámbricos que formarán la red son computadoras portátiles, asistentes digitales y teléfonos celulares.
- Los nodos inalámbricos utilizan alguna tecnología inalámbrica como 802.11b, infrarojo o Bluetooth.
- A pesar de que la naturaleza de la red espontánea es impredecible y dinámica, se espera que ocurran cambios pero que no sean tan frecuentes.

Dado que las redes espontáneas son la infraestructura de comunicación para este trabajo en la siguiente sección se presentan los principales problemas que deben ser tratados en este tipo de redes.

2.4. Principales problemas de las redes espontáneas

El problema principal de las redes espontáneas es que no dependen de la existencia de alguna infraestructura física, aún más, ni siquiera se puede suponer de la existencia de algunos servicios. A continuación presentaremos varios de los problemas que establecen la diferencia entre las redes espontáneas y las redes tradicionales IEEE 802.3 con infraestructura.

- *Los límites de la red están pobremente definidos*

Se refiere al hecho de que no se puede saber cuantos nodos formarán parte de la red. A diferencia de las redes tradicionales basadas en cable donde el número de rosetas puede delimitar el número de nodos que la red tendrá.

- *La red no está planeada*

Una red tradicional basada en infraestructura generalmente lleva un proceso de planeación donde se decide el número de nodos que se espera y además los servicios que cada una de las estaciones puede tener en esa red. Sin embargo en una red espontánea no se puede hacer alguna

2.4. PRINCIPALES PROBLEMAS DE LAS REDES ESPONTÁNEAS 17

suposición con respecto a los nodos que formarán la red y los servicios que la red pueda tener.

- *Las estaciones no están preconfiguradas*

Dado que las redes espontáneas pueden ser creadas en cualquier momento, en cualquier lugar y con cualquier número de participantes. Esto evita que se pueda realizar alguna actividad de preconfiguración en cada una de las estaciones. Como por ejemplo la asignación de direcciones a cada uno de los nodos.

- *No existen servidores centrales*

En las redes espontáneas no se puede depender de algún servidor central. Dado que el posible movimiento de los nodos obliga a que, en caso de que la red se fragmente, los nodos que estaban haciendo uso del servicio ya no tengan acceso a él. Este es uno de los problemas que ha dado pauta a que se les considere como un tipo especial de sistema distribuido.

- *Los usuarios no son expertos*

Se refiere a que los usuarios esperan que la puesta en marcha de estas redes se realice de una manera sencilla y ellos tengan que configurar lo menos posible. Esto obliga a agregar mecanismos que permitan la autoconfiguración de los nodos.

Una tecnología que es utilizada para la creación de redes espontáneas es Bluetooth, es la tecnología de comunicación que es empleada en este trabajo y de ella trata la siguiente sección. Es importante comentar que hasta el momento Bluetooth es la tecnología que más ha sido integrada en diferentes dispositivos móviles como teléfonos celulares (Ericsson T68i), asistentes digitales (Palm Tungsten, iPAQ) y computadoras portátiles (Apple iBook, Sony Vaio).

Cabe destacar que la comunicación inalámbrica no es reciente, sus orígenes datan desde 1896 cuando Marconi demostró que era posible la telegrafía inalámbrica al utilizar transmisores de alta potencia para recibir y enviar código Morse a través de una onda larga (mayor a 1Km) [8].

En la actualidad ha cobrado importancia, en principio por que los nuevos estándares de comunicación inalámbrica ya pueden transmitir datos. Además

varios de los estándares están diseñados para que puedan ser integrados físicamente en dispositivos móviles ligeros como asistentes digitales, teléfonos celulares, o computadoras portátiles. La convergencia entre estos dos avances tecnológicos –dispositivos móviles y comunicación inalámbrica– ha dado lugar al surgimiento de nuevas áreas de investigación en ciencias de la computación. Resaltan la computación móvil [5] y la computación ubicua [7]. También ha dado origen a nuevos modelos de negocio como el comercio móvil, el cual busca aprovechar la proliferación de dispositivos equipados con tecnología inalámbrica para poder realizar transacciones comerciales a través de ellos.

2.5. Tecnología inalámbrica Bluetooth

Bluetooth es una tecnología inalámbrica que surge a partir de un proyecto iniciado por Ericsson en el año de 1994 [9]. El proyecto comenzó como un estudio de factibilidad para la creación de una tecnología de bajo costo y de bajo consumo de energía que sirviera para eliminar los cables que conectaban a los teléfonos celulares con los diferentes accesorios.

Ericsson, más tarde, se dio cuenta de que la tecnología sería útil en el mercado si era apoyada por las principales compañías líderes en el desarrollo de equipo de cómputo, software y telecomunicaciones. Así se formó el Grupo de Interés de Bluetooth (SIG) que actualmente lo componen: Ericsson, Intel, IBM, Nokia, Toshiba, Microsoft, Motorola, 3Com, Lucent.

En la actualidad Bluetooth se refiere a la especificación abierta de una tecnología inalámbrica que permite la transmisión de voz y datos en distancias cortas y ha sido integrada en diferentes dispositivos móviles. Las características principales de la tecnología son:

- *Es una especificación abierta [10][11].*

El grupo de interés (SIG) ha producido una especificación abierta que está libre de regalías y está disponible públicamente.

- *De bajo consumo de energía.*

El bajo consumo de energía de Bluetooth ha permitido su integración a dispositivos móviles con recursos limitados.

- *Permite la transmisión de voz y datos de forma inalámbrica en distancias cortas (menores a 10 m).*

- *Puede ser utilizada en cualquier parte del mundo.*

Bluetooth opera sobre el espectro sin licenciar de 2.4GHz el cual está disponible mundialmente.

En la figura 2.2 podemos observar una gráfica donde se comparan diferentes estándares con respecto al consumo de energía y la velocidad de transmisión.

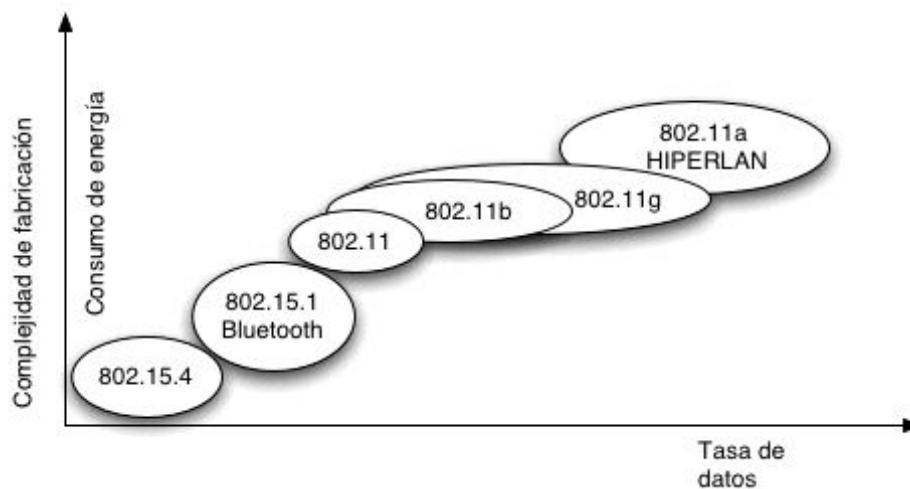


Figura 2.2: Comparación de diferentes estándares inalámbricos de la familia IEEE 802.

Se puede observar que Bluetooth tiene un bajo consumo de energía, pero también es baja su velocidad de transmisión. En la gráfica también se observa la complejidad de fabricación de los diferentes estándares y la constante que se mantiene es que un mayor consumo de energía significa que la tecnología es más difícil de fabricar –principalmente por que se necesitan procesadores de señales más poderosos– y por ende es más costosa su construcción.

Es conveniente enfatizar que cada uno de los estándares tiene características que limitan y definen el tipo de aplicaciones que pueden ser desarrolladas con ellas. En este caso, Bluetooth puede ser utilizado por aplicaciones las cuales necesitan la formación de una red entre varios dispositivos móviles sin la necesidad de depender de una infraestructura y los volúmenes de datos

a transmitir no sean muy altos. Por ejemplo, aplicaciones que hagan uso de Bluetooth para transmitir video no son adecuadas. Para aplicaciones de esa naturaleza podría emplearse el estándar 802.11b.

A partir de marzo de 1999 se creó el grupo de trabajo IEEE 802.15 el cual está enfocado al desarrollo de una familia de estándares de comunicación para redes de área personal inalámbricas (WPAN). Ese grupo escogió a Bluetooth para que formara la base del estándar 802.15.1 [12]. Por ello en la Figura 2.2 se puede observar este número encima de la leyenda Bluetooth.

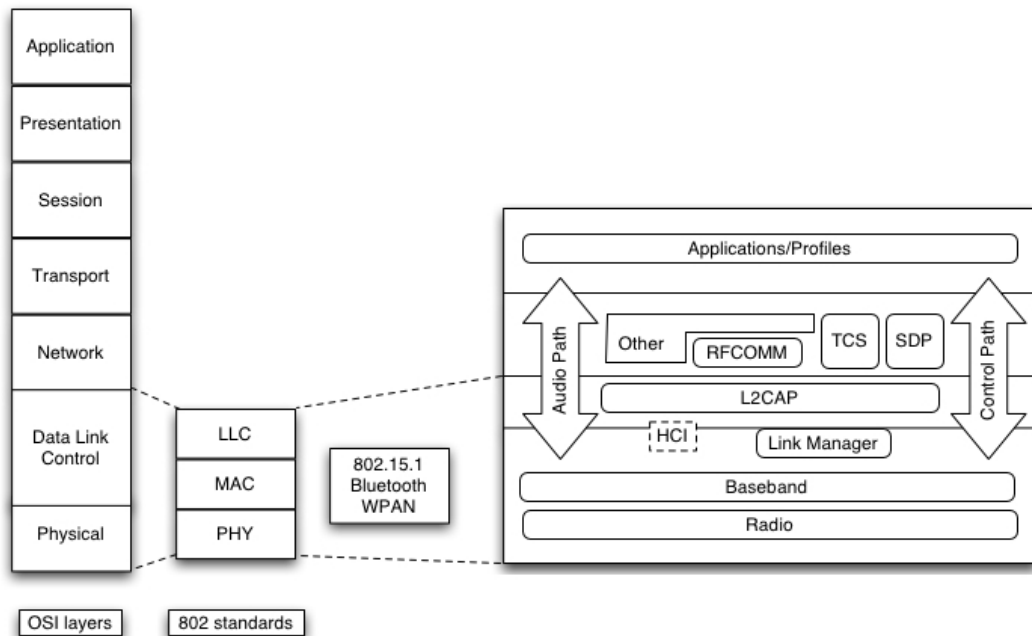


Figura 2.3: La relación entre las capas OSI, las capas IEEE 802 y Bluetooth.

El estándar IEEE 802 únicamente trata con la definición de dos de las capas del modelo OSI, la capa física (PHY) y a la capa de acceso al medio (MAC). El grupo de trabajo IEEE 802.15 decidió que los protocolos de transporte de la pila de Bluetooth serían la base del estándar IEEE 802.15.1. En la figura 2.3 se puede observar como se relacionan cada uno de los protocolos de Bluetooth con las capas del IEEE 802 y las capas OSI. Es importante mencionar que Bluetooth no implementa una capa de red y de transporte como TCP/IP.

El grupo de IEEE 802.15 sigue aún muy activo y está desarrollando el estándar 802.15.4 para crear una tecnología que consuma aún menos energía que Bluetooth y pueda ser integrada a una variedad más amplia de dispositivos.

2.5.1. La pila de protocolos

La especificación completa de Bluetooth se divide en dos volúmenes:

1. *El núcleo de la especificación [10].*

El núcleo de la especificación trata primordialmente sobre el funcionamiento de la pila de protocolos de Bluetooth y es, sin duda, el más grande de los dos volúmenes.

2. *Los perfiles [11].*

El volumen de perfiles trata sobre el uso de la pila de protocolos. Los artículos [13] [14] [15] explican a gran detalle el funcionamiento y las características de Bluetooth.

La pila de protocolos de Bluetooth la podemos dividir lógicamente en tres grupos, los cuales se muestran en la Figura 2.4:

- *Protocolos de transporte*

Estos protocolos permiten a los dispositivos localizarse unos a otros, así como, crear, manejar y configurar los enlaces físicos y lógicos. A este grupo pertenecen el radio (RF), el *baseband*, el link manager, *Logical link and adaptation protocol (L2CAP)*, *host controller interface (HCI)*. En [13] se presenta una buena explicación acerca de su funcionamiento.

- *Protocolos de Middleware*

Incluye protocolos de terceros y estándares industriales como PPP e IrDA. Además, protocolos específicos de Bluetooth como RFCOMM que es la emulación de un puerto serial, al igual que operaciones telefónicas y el protocolo de descubrimiento de servicios (SDP).

- *Protocolos de Aplicación*

Este grupo consiste de las aplicaciones que hacen uso de Bluetooth.

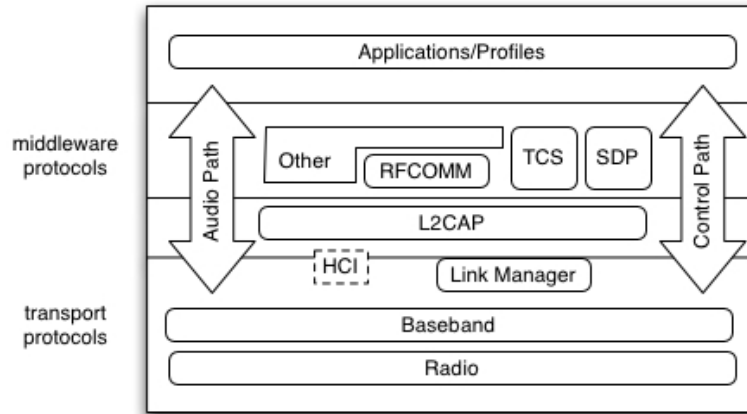


Figura 2.4: Pila de protocolos de Bluetooth.

Es importante recalcar que Bluetooth no trabaja de forma nativa con el protocolo TCP/IP, aunque, éste puede ser implementado en los *Protocolos de Middleware*.

A pesar de que en 2001 se liberó la especificación, la pila de protocolos ha sido implementada por diferentes compañías y en la actualidad existen adaptadores para computadoras de escritorio, portátiles y asistentes digitales. Además muchos de los dispositivos ya traen integrada la tecnología como: Palm Tungsten, iPAQ 3870, teléfono celular Ericsson T68i, y los modelos recientes de las computadoras fabricadas por Apple.

2.5.2. El medio de comunicación

Bluetooth emplea ondas de radio frecuencia (RF) para transmitir la información de manera inalámbrica. El uso de RF ofrece ventajas sobre otras tecnologías inalámbricas como infrarojo, ya que no necesita que el emisor y el receptor tengan una línea de visión libre de obstáculos.

El uso del espectro sin licenciar de 2.4GHz implica que Bluetooth tenga que convivir con varias tecnologías que hagan uso del mismo. Entre ellas tenemos: redes LAN inalámbricas, teléfonos inalámbricos, hornos de microondas.

Bluetooth, emplea *Frequency Hopping Spread Spectrum (FHSS)* y con esto reduce la interferencia que en determinado momento pudiera existir.

FHSS divide el espectro en canales de 1 MHz, dependiendo del país pueden llegar a ser 79 o 23 canales. Para transmitir un mensaje, éste se divide en paquetes, luego, el dispositivo emisor selecciona una frecuencia del espectro y envía un paquete, así sucesivamente va seleccionando frecuencias hasta que envía todos los paquetes. El receptor debe de estar sintonizado con el patrón de frecuencias usado por el emisor y se conoce como *patrón de salto de frecuencias* (frequency hopping pattern).

Bluetooth emplea GFSK (Gaussian Frequency Shift Keying) para realizar la modulación y un radio Bluetooth típico transmite con un poder de 1mW.

2.5.3. Formación de la piconet

Para que los dispositivos puedan comunicarse entre ellos es necesario que formen parte de una *piconet*. De una manera sencilla, una piconet comprende un canal de comunicaciones compartido a través del cual los miembros de la piconet se pueden comunicar. Trasladando esto al espacio del FHSS, el canal de comunicaciones corresponde a un patrón de saltos de frecuencias seleccionado pseudo aleatoriamente de un conjunto de frecuencias que los dispositivos comparten. Los protocolos de la *baseband*, figura 2.4, son los encargados de definir como es seleccionado ese patrón de frecuencias, como los dispositivos aprenden a seguirlo y por tanto como pueden formar parte de la piconet, además, define los mecanismos para mandar y recibir paquetes de información de una manera coordinada.

Una característica novedosa de Bluetooth es que las piconets se forman de una manera ad-hoc. Se forman por la demanda de los dispositivos que desean comunicarse.

En cada piconet puede haber solo un *maestro* y uno o más *esclavos*. Estos roles son temporales y sólo son significativos durante el tiempo de vida de la piconet. El rol del maestro es importante para las comunicaciones, porque define:

- *El patrón de frecuencias que los miembros de la piconet deben seguir.*
- *Cuándo ocurre un salto de frecuencia.*
- *Cuál es la frecuencia actual.*
- *Cuál es el esclavo que va a transmitir.*

En la figura 2.5 se muestran los estados operacionales de un dispositivo Bluetooth. En el estado *connected*, el dispositivo es miembro de una piconet. Por otro lado, cuando un dispositivo no es parte de la piconet y no realiza ninguna acción para formar o ser parte de una piconet se dice que está en el estado *standby*.

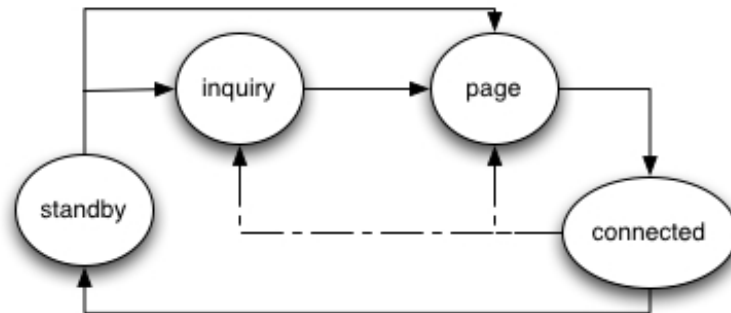


Figura 2.5: Estados operacionales de un dispositivo Bluetooth.

Para ir del estado *standby* al estado *connected* el dispositivo tiene que pasar por los estados *inquiry* y *page*. En el estado *inquiry* el dispositivo obtiene la identidad de los dispositivos que se encuentran a su alrededor, los cuales deben estar en el estado *inquiry scan*. Una vez que ha obtenido la identidad de los dispositivos pasa al estado *page* y son candidatos para ser invitados a formar parte de su piconet. Para que reciban la invitación éstos tienen que estar en el estado *page scan*. Una vez formada la piconet pasa al estado *connected*. La figura 2.5 a través de líneas punteadas indica que un dispositivo en estado *connected* puede realizar nuevas búsquedas y puede invitar a otros dispositivos.

Es importante recalcar que la especificación de Bluetooth define los intervalos de tiempo que van a emplear los dispositivos para pasar a los estados *inquiry scan* y *page scan*. Estos estados como se mencionó en el párrafo anterior son indispensables para establecer una conexión entre dos dispositivos. Por tanto aunque un dispositivo tenga el rol de maestro en una piconet éste tiene reservados ciertas ranuras de tiempo para pasar al estado *inquiry scan* y *page scan*, por consiguiente, es posible que un dispositivo sea maestro en una piconet y esclavo en otra. De igual forma sucede con un dispositivo que

tiene el rol de esclavo, éste tiene reservadas ciertas ranuras de tiempo para pasar a los estados *inquiry scan* y *page scan* de tal manera que es posible que un dispositivo tenga el rol de esclavo en dos diferentes piconets. De esto podemos inferir que estos estados corresponden a cualquier dispositivo Bluetooth sin importar el rol que en determinado momento haya adquirido.

El rol de maestro lo adquiere el dispositivo que invita a los demás dispositivos a unirse a su piconet. Pero es posible un cambio de rol entre los dispositivos. Una piconet puede tener a lo más 7 esclavos y un sólo maestro. La comunicación se da de maestro a esclavo y viceversa, pero no de esclavo a esclavo.

El protocolo de acceso al medio es *Time Division Duplex Polling Scheme* (TDD). La idea es dividir el canal en ranuras de tiempo (slots) de $625\mu\text{s}$ para que los dispositivos Bluetooth transmitan y reciban información de una manera alternada. Dado que la transmisión se lleva a cabo en ranuras de tiempo diferentes, la transmisión tiene lugar en diferentes frecuencias.

En la figura 2.6 se ilustra el mecanismo de acceso al medio. Recordemos que el canal de comunicaciones corresponde a un patrón de frecuencias de salto, cada una de estas frecuencias está representado en la figura por $f(k), f(k+1), f(k+2)$. Luego al utilizar TDD dividimos este canal en ranuras de tiempo de $625\mu\text{s}$. De tal manera que los dispositivos A y B transmiten y reciben en frecuencias y ranuras de tiempo diferentes. Es importante resaltar que el reloj que riga las comunicaciones en el canal es el del maestro. Los demás dispositivos deben estar sincronizados con éste. El maestro siempre transmite en ranuras de tiempo con número par y el esclavo en impares.

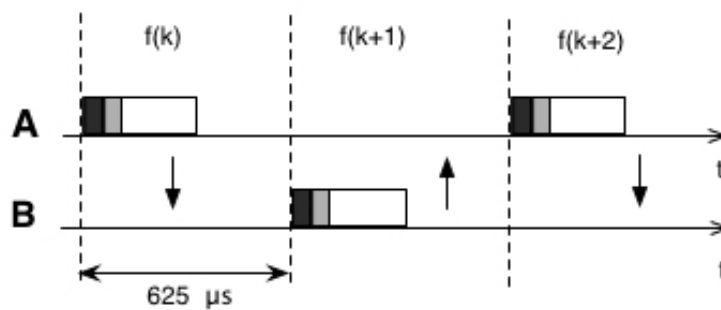


Figura 2.6: Mecanismo de acceso al medio (FH/TDD).

El hecho de que exista un maestro para coordinar las comunicaciones en la piconet, da lugar a que se tenga siempre una topología de red estrella.

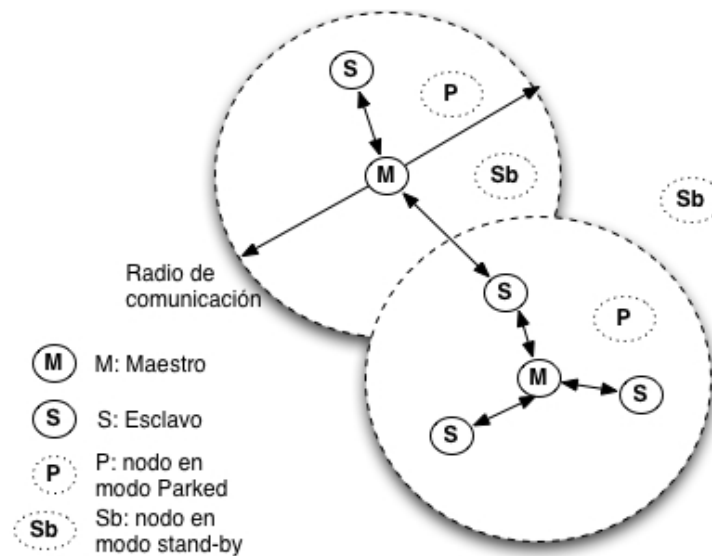


Figura 2.7: Un ejemplo de una scatternet.

Es posible que un dispositivo sea maestro en una piconet y esclavo en otra. Esto puede dar lugar a que dos o más piconets se traslapen en espacio y tiempo, a esto se le conoce como *scatternet*, vea la figura 2.7.

2.5.4. Mecanismos para el ahorro de energía

Para identificar a cada esclavo de la piconet el maestro le asigna a cada uno de ellos una dirección de 8 bits, *AM_ADDR*, que los identifica como miembros activos de la piconet. Dado que los mecanismos de ahorro de energía están ligados con el mecanismo de acceso al medio, es necesario recordar que el mecanismo de acceso al medio es controlado por el maestro. Por tanto él decide quien transmite y cuando. Además los periodos de transmisión se alternan entre el maestro y el esclavo de una manera Time Division Duplex (TDD). En particular el maestro transmite en ranuras de tiempo pares y el esclavo en ranuras de tiempo impares. Un esclavo sólo puede transmitir des-

pués de que el maestro ha transmitido al esclavo. Tomando esto en cuenta, a continuación se explican los 3 modos que Bluetooth ofrece para el ahorro de energía.

- *sniff*. En este modo el esclavo acuerda con el maestro escuchar por paquetes transmitidos por él periódicamente. Este periodo es configurado a través de transacciones que se realizan usando el Link Manager Protocol (LMP).
- *hold*. En este modo un esclavo acuerda con el maestro a no transmitir durante un período de tiempo. El dispositivo que pasa al modo hold no tiene que dejar la dirección de miembro activo, *AM_ADDR* que le asignó el maestro.
- *park*. En este modo el dispositivo deja su dirección de miembro activo, *AM_ADDR*, de la piconet y ya no interviene en alguna actividad de transferencia de información, salvo por ciertos períodos intermitentes (beacon slots) en los que el maestro utiliza para comunicarse con los dispositivos que están en modo parked y que sirven para poder incorporar nuevamente a alguno de ellos a la piconet. En caso de que un esclavo desee incorporarse nuevamente a la piconet debe transmitir un mensaje al maestro inmediatamente que ha recibido un paquete en un período intermitente.

2.5.5. Descubrimiento de servicios

El descubrimiento de servicios es un proceso mediante el cual otros dispositivos que se encuentran en una red pueden localizar los servicios disponibles sin la necesidad de depender de servidores centrales. En las redes basadas en cable el descubrimiento de servicios es de gran ayuda, sin embargo, en redes inalámbricas formadas de manera espontánea el descubrimiento de servicios es una necesidad ya que permite tratar con un ambiente dinámico en el cual los dispositivos y servicios entran y salen de la red.

El SIG de Bluetooth consideró que era de suma importancia incorporar un protocolo de descubrimiento de servicios y de esta manera surgió el SDP (Service Discovery Protocol) de Bluetooth. El cual fue liberado en julio de 1999 en la versión 1.0 de la especificación.

El protocolo de descubrimiento de servicios usa los conceptos de cliente y servidor. Cualquier dispositivo puede ser cliente o servidor en un instante dado.

El servidor necesita almacenar una lista de *entradas de servicios*, los cuales sirven para describir los servicios que el dispositivo proporciona. Esta lista es conocida como el *registro de servicios*. Una entrada del registro de servicios está formada por *atributos de servicio*, ellos contienen la información acerca de la clase de servicio, de los protocolos de la pila involucrados, de la descripción del servicio, etc.

El registro de servicios consta de:

- *Atributos universales de servicio*. Los atributos universales son aquellos que son parte de todos los tipos de servicio, tal como: la clase de servicio y los protocolos de la pila involucrados.
- *Atributos específicos de servicio*. Los atributos específicos son aquellos que sólo corresponden a un servicio en particular.

Para descubrir un servicio, el cliente especifica el servicio de interés y el servidor responde con los servicios disponibles que empaten con los que el cliente especificó. En la práctica las peticiones se realizan a través de paquetes que se conocen como SDP_PDU (Service discovery protocol protocol data units). Para garantizar la unicidad de un determinado servicio SDP hace uso de UUID (Universally unique identifiers). UUID son valores de 128 bits que pueden ser creados algorítmicamente y se puede garantizar que serán únicos a nivel mundial.

La especificación define tres operaciones para el descubrimiento de servicios:

- *Búsqueda de Servicio (Service Search)*

El cliente manda una petición para buscar por un determinado servicio.

- *Petición de un atributo del servicio (ServiceAttributeRequest)*

Utiliza la información obtenida en la transacción anterior para obtener atributos del servicio.

- *Búsqueda de servicio y petición de atributo (ServiceSearchAttributeRequest)*

Esta transacción es la unión de las dos anteriores.

2.5.6. Perfiles

En la actualidad muchos de los casos de uso de la tecnología inalámbrica, en particular Bluetooth, únicamente tratan el problema de la interacción entre dos dispositivos móviles. Dejando fuera otros posibles escenarios que involucren a más de dos dispositivos y los cuales es muy probable que también tengan éxito debido a la proliferación que existe de los dispositivos móviles.

Para ilustrar esto a continuación presento algunos de los casos de uso que el grupo de interés, por sus siglas en inglés (SIG), de Bluetooth tiene contemplados y que han llamado *perfiles*[11].

- *La computadora sin cables*

Consiste en el reemplazo de los cables que conectan a la computadora personal con los diferentes periféricos: teclado, ratón, impresora, etc.

- *Transferencia de archivos y otro tipo de objetos*

Consiste en formar una liga de comunicación inalámbrica punto a punto entre dos dispositivos con el objetivo de transferir archivos u otros objetos. Para ilustrar mejor este perfil, en la parte superior de la figura 2.8 se muestran dos computadoras portátiles las cuales transfieren un archivo entre ellas haciendo uso del dispositivo Bluetooth que tienen integrado. En la parte inferior se puede observar a una computadora que transfiere un objeto *vCard* a un teléfono celular. Un objeto *vCard* es la representación digital de una tarjeta de presentación.

Cabe destacar que este es el perfil más popular y actualmente ha comenzado a extenderse su uso.

- *Sincronización automática*

Consiste en poder fusionar conjuntos de datos de dos diferentes dispositivos a partir de un conjunto de reglas para que se tenga la misma información en ambos. Generalmente el proceso anterior se realizaba a través de cables, con este perfil se realizará de manera inalámbrica y automática. Por ejemplo, en la actualidad es común que se tenga que sincronizar la información de una computadora personal con el asistente digital. Este perfil pretende que la sincronización se haga de manera automática una vez que los dispositivos se encuentran a una determinada distancia.

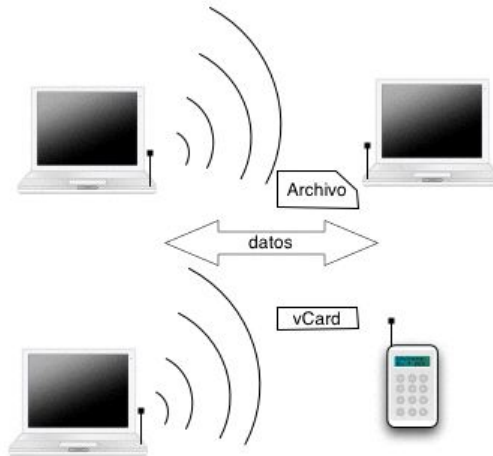


Figura 2.8: Perfil de transferencia de archivos y objetos en Bluetooth.

■ *La postal instantánea*

Debido a que Bluetooth es una tecnología que puede ser incorporada en diferentes dispositivos. Este perfil consiste de una cámara digital la cual tiene integrado Bluetooth que transmitirá una foto a un teléfono celular o una computadora portátil. El cual posteriormente enviará por correo electrónico a alguna persona.

Cada uno de los perfiles contemplan solamente dos dispositivos. Sin embargo, debido a la proliferación de los mismos es posible y necesario en algunos casos que un grupo de personas con sus dispositivos móviles puedan realizar alguna actividad en común. El objetivo no es tan solo transferir archivos o sincronizar información. Con esto me estoy refiriendo a actividades de colaboración humana las cuales sean enriquecidas con el poder de cómputo que cada uno de ellos llevan en sus bolsillos o maletas. A estos escenarios de colaboración enriquecidos con el uso de dispositivos digitales los llamaremos *escenarios de colaboración digitales*.

2.6. Conclusiones

Bluetooth en la actualidad, se ha utilizado únicamente como una tecnología para establecer un canal de comunicación entre dos dispositivos. Esto se debe principalmente a que la implementación de los protocolos de Bluetooth en los diferentes sistemas operativos no cubren una capa de red y transporte encima de los protocolos de acceso al medio como L2CAP.

En caso de que se implemente TCP/IP, se deben integrar además mecanismos como la autoconfiguración de las interfaces para que se continúen explotando características como el descubrimiento de dispositivos y servicios. Sin embargo, es necesario que exista una capa intermedia que dote a las aplicaciones de mecanismos para:

- *La formación de la red.*
- *La transferencia de información entre los diferentes nodos.*

Los mecanismos proporcionados no deben inhibir las características de la tecnología como el descubrimiento de dispositivos y servicios. El middleware que se diseñó viene a cubrir esos dos aspectos.

Otra limitación de Bluetooth es el protocolo de descubrimiento de servicios (SDP), al ser un protocolo sencillo, carece de funcionalidad especial como:

- *Anuncio del servicio.*
- *Notificación de la disponibilidad de algún servicio.*

El protocolo está orientado principalmente al descubrimiento de dispositivos y una vez que se ha localizado alguno es posible hacer una consulta. En [16] encontrará una revisión de los principales protocolos desarrollados para el descubrimiento de servicios. Para el proyecto la funcionalidad del SDP es suficiente, adicionalmente, se agregaron ciertos mecanismos para que las aplicaciones soporten la salida de algún nodo.

Es importante recalcar que las tecnologías actuales tienen serias limitaciones para la integración de dispositivos móviles. En el siguiente capítulo se explica el diseño y la implementación de un conjunto de mecanismos que complementan la tecnología inalámbrica Bluetooth y hacen posible el desarrollo de escenarios de colaboración digitales.

Capítulo 3

El middleware Lucrn

El objetivo del middleware *Lucrn* es proporcionar un conjunto de mecanismos que extiendan la pila de protocolos de Bluetooth y permitan a las aplicaciones distribuidas en dispositivos móviles heterogéneos interoperar en un ambiente en el cual la red se crea de una forma casual y, por su naturaleza, no se puede suponer la existencia de determinados servicios computacionales.

Este capítulo está dividido en dos partes, la primera parte presenta al middleware desde un punto de vista muy general. La segunda parte explica cada uno de los capas que forman parte del middleware.

3.1. Una vista general del middleware

El middleware *Lucrn* se concibe a raíz de la falta de mecanismos que permitan a las aplicaciones distribuidas en dispositivos móviles interoperar en una red que se forma de manera espontánea entre diferentes dispositivos móviles. El middleware fue diseñado para que sea implementado en una gran variedad de dispositivos móviles y sea útil para el desarrollo de aplicaciones que se ejecutan en redes espontáneas, además dota a las aplicaciones de mecanismos para:

- *La configuración de los nodos.*
- *La integración de cada uno de los nodos a la red.*
- *La salida de cada uno de los nodos de la red.*
- *La difusión de mensajes.*

- *La colaboración entre aplicaciones.*

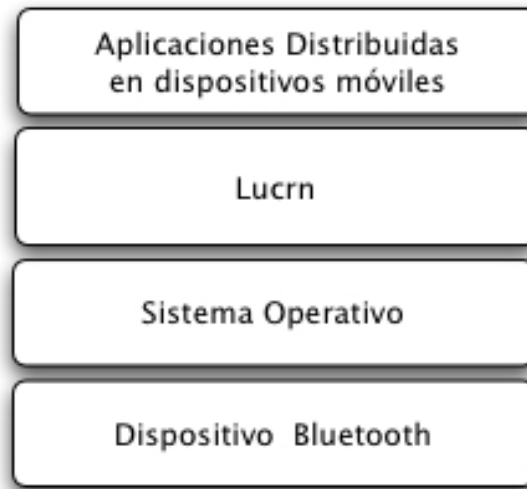


Figura 3.1: El middleware en un contexto de alto nivel.

La figura 3.1 ofrece una vista muy general de como se sitúa el middleware en los diferentes dispositivos móviles. En la parte inferior se encuentra el dispositivo Bluetooth que se encarga de transmitir la información haciendo uso de radio frecuencia. Encima de esta capa se encuentra el sistema operativo que se encarga de la administración del dispositivo. El sistema operativo, además, proporciona una interfaz a las aplicaciones que las abstrae de los detalles internos que se tienen que llevar a cabo para el uso de un determinado recurso. En particular, ciertos mecanismos están disponibles a las aplicaciones a través de interfaces de programación (API), las cuales son usadas para el manejo del dispositivo Bluetooth.

Después de la capa del sistema operativo sigue la capa que corresponde al middleware Lucrn, la cual está compuesta por un conjunto de niveles o capas los cuales dotarán a las aplicaciones de mecanismos para tratar con los problemas originados al formar parte de una red espontánea. Finalmente, las aplicaciones distribuidas se encuentran en la parte más alta de esta jerarquía y usan la interfaz de programación del middleware Lucrn para aprovechar todas las características que éste ofrece.

El diseño del middleware se realizó utilizando un enfoque orientado al uso de capas. Es decir, el middleware se dividió en capas de mecanismos que ofrecen determinados servicios. De tal manera que para las capas superiores la capa inmediata inferior representa una *caja negra* que tiene una determinada función.

Esta técnica de diseño a lo largo de los años ha mostrado tener ciertas ventajas, sobre diseños monolíticos, como [17]:

- *Diseños más sencillos.*
- *Diseños que pueden ser comprendidos más fácilmente.*
- *Diseños que sirven para la definición de estándares.*
- *Diseños en los cuales puede ser posible la sustitución o el intercambio de ciertas capas.*

Es muy importante recalcar que cada capa representa a un protocolo el cual define un conjunto de reglas que rige la interacción entre procesos que se ejecutan concurrentemente en los diferentes dispositivos móviles. En tiempo de ejecución estos procesos están distribuidos entre los diferentes dispositivos móviles, esto da lugar a que cada protocolo represente a un algoritmo distribuido.

3.2. Las capas del middleware Lucrn

Para el diseño de cada uno de las capas fue muy importante tomar en cuenta varios aspectos relacionados con los tipos de dispositivos que serán usados, las características que éstos ofrecen y la manera en que serán conectados.

Definitivamente los aspectos que fueron consideradas durante toda la etapa de diseño fueron:

1. *Los nodos que componen la red son móviles.*
2. *Los nodos que componen la red son heterogéneos.*
3. *Los nodos que componen la red son de recursos limitados.*

4. *Los nodos utilizan tecnología inalámbrica Bluetooth para transmitir la información.*
5. *La red que se forma entre los nodos es espontánea.*

De igual forma fue necesario definir criterios para el diseño de cada una de las capas:

- *Las capas deben de aprovechar los mecanismos proporcionados por la tecnología de comunicación inalámbrica para la creación de la red espontánea.*
- *El diseño de cada capa debe de tomar en cuenta que va a ser implementado en dispositivos heterogéneos con distintas capacidades y distintos sistemas operativos.*
- *El diseño de cada capa debe tomar en cuenta que los usuarios no son expertos, por tanto no debe asumir que los usuarios van a realizar algún proceso de configuración.*

Cada una de las capas que componen al middleware Lucrn las podemos observar en la figura 3.2.



Figura 3.2: Los protocolos que componen al middleware Lucrn.

En la figura 3.2 se observan 4 protocolos, cada uno de ellos define algún mecanismo que es esencial durante la ejecución de las aplicaciones distribuidas cooperativas en los diferentes dispositivos móviles.

A continuación se describe de manera muy breve la funcionalidad de cada uno de ellos:

- *Protocolo de anuncio de dispositivos.*

Este protocolo consiste de un conjunto de mecanismos y de mensajes de control que sirven para anunciar la entrada de algún dispositivo a la red. Define, además, las acciones que se deben de llevar a cabo cuando un nuevo dispositivo se integra. Por ejemplo, el agregar el canal recién creado a la lista de canales y la forma en la que se tiene que llenar la tabla de ruteo. Cabe resaltar que es la base del middleware y proporciona mecanismos para que las capas superiores puedan operar correctamente.

- *Protocolo de ruteo.*

Este protocolo consiste de un conjunto de operaciones y mensajes que sirven para que los datos puedan ser transmitidos entre los diferentes nodos de la red. Hace uso de las tablas de ruteo que fueron actualizadas durante el proceso de integración o de salida de un dispositivo a la red espontánea.

- *Protocolo de movilidad.*

Este protocolo se apoya sobre el protocolo de ruteo para poder anunciar a los demás la salida de algún nodo y define las acciones que deben realizar cada uno de ellos cuando algún evento de esta naturaleza suceda.

- *Protocolo de colaboración.*

Es el protocolo de más alto nivel, proporciona a las aplicaciones mecanismos para que estas puedan colaborar. Como ejemplo de estos mecanismos tenemos a los que replican las acciones que ocurren en cada una de las aplicaciones y que sirven para que los usuarios tengan una vista global de lo que sucede en un determinado documento.

Para explicar cada uno de los protocolos se presentarán los cinco elementos que una especificación de protocolos debe de contar para estar completa [18]:

- *El servicio que ofrece el protocolo.*

- *Las suposiciones acerca del ambiente en el cual se va a ejecutar.*
- *El tipo de mensajes válidos para el protocolo (vocabulario).*
- *El formato de cada uno de los mensajes.*
- *Las reglas que mantienen la consistencia del intercambio de mensajes que se lleva a cabo entre las diferentes entidades.*

Adicionalmente, al inicio de cada sección se explicará el problema al cual da solución cada protocolo.

3.2.1. Formato de los mensajes

Cada uno de los mensajes que se intercambien entre las diferentes entidades de los protocolos deberán seguir un determinado formato. Con esto aseguramos que los diferentes dispositivos que implementen este middleware puedan interoperar transparentemente. Además, dado que cada protocolo define mensajes de control, fue necesario definir una estructura básica que sirviera para el intercambio de información entre las entidades de los diferentes protocolos.

Un mensaje va a estar compuesto por 2 partes:

- *Encabezado*
Proporciona información acerca del tipo de mensaje
- *Cuerpo*
Lleva los datos que corresponden a cada tipo de mensaje

La función del encabezado es proporcionar información acerca del tipo de mensaje. De tal manera que cuando un mensaje llegue sólo se tengan que obtener los primeros 64 bits y a partir de la lectura y análisis de éstos se determine las acciones que se deben realizar. Por ejemplo, el reenvío a otro dispositivo o bien el paso del mensaje a una capa superior para que se realice el procesamiento de la información. En la figura 3.3 se puede observar cada uno de los campos que componen al encabezado, así como el tamaño de cada uno de ellos.

El campo de *tipo de mensaje*, figura 3.3, corresponde a los bits del 0 al 7 su función es indicar la semántica del mensaje y, además, determina el

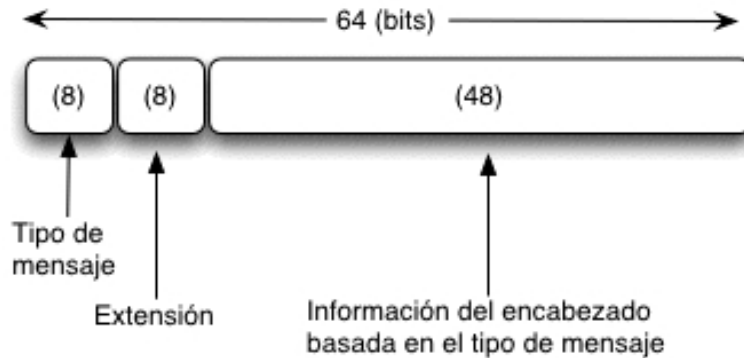


Figura 3.3: Encabezado de los mensajes.

formato del cuerpo del mensaje y de la información que se llevará de los bits 16 al 63 del encabezado.

Existen determinados mensajes en los cuales es necesario incluir mayor información en el encabezado, para ellos se ha definido el uso del campo de extensión el cual comprende los bits 8 al 15.

El campo *tipo de mensaje* define varias características del paquete o mensaje como se explicó anteriormente. En la figura 3.9 se puede observar la información que tendrá un paquete del tipo *0x00* que corresponde a un *paquete de datos*.

El *paquete de datos* contiene campos que llevan información acerca de las direcciones Bluetooth de origen, destino, el tamaño de los datos que son enviados en el paquete y los datos.

Cada protocolo define los mensajes válidos y el tipo que cada uno de ellos utilizan, en la explicación de cada uno de ellos se dedica una sección para mostrar el formato de los mensajes.

3.2.2. Protocolo de anuncio de dispositivos

La especificación de Bluetooth define los mecanismos para formar una *piconet* entre varios dispositivos. Como se vió en el capítulo anterior, la topología de la *piconet* es estrella debido a que está formada por sólo un maestro y varios esclavos. El rol de maestro es importante por que él decide el patrón

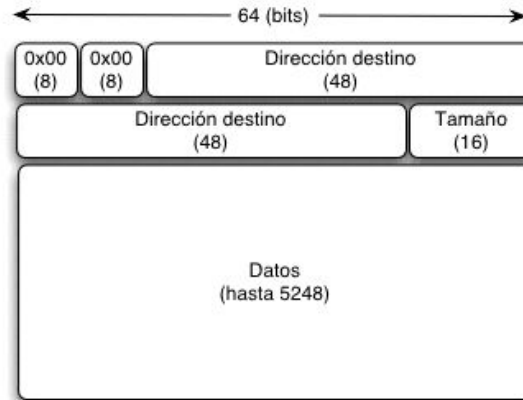


Figura 3.4: Un mensaje de datos.

de frecuencias y el acceso al medio. Otro aspecto es que la comunicación se lleva a cabo solamente entre el maestro y el esclavo. Por tanto, no es posible que la transferencia de información se pueda dar entre los esclavos. Es decir, la especificación de Bluetooth no proporciona los mecanismos para que un esclavo envíe un paquete de información a otro esclavo que se encuentra en la misma piconet. Esto limita las aplicaciones, dando lugar a que en la actualidad sólo existan aplicaciones que utilicen Bluetooth para sincronizar un conjunto de datos entre dos dispositivos y no entre varios dispositivos.

El middleware Lucrn resuelve este problema y en particular este protocolo sienta las bases para que un dispositivo envíe su identidad (dirección Bluetooth) a los demás a través de los diversos canales de comunicación que se han abierto entre los dispositivos de la red espontánea. Con esto es posible que un dispositivo sepa de la llegada de un dispositivo y además sepa que existen un canal a través del cual puede llegar a él.

Servicio

El servicio que ofrece este protocolo es anunciar a los demás dispositivos que conforman la red espontánea acerca de la creación de un nuevo enlace de comunicación L2CAP entre un dispositivo que es parte de la red y uno externo que no era parte de ella.

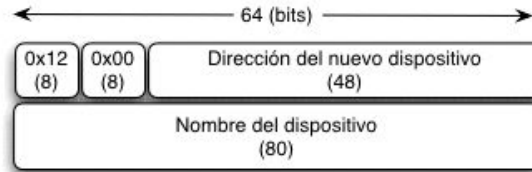


Figura 3.5: Formato del mensaje LUCRN_NEW_DEVICE.

Suposiciones

- Se ha establecido un canal L2CAP entre los dos dispositivos
- Los dispositivos tienen una lista de los canales que tienen abiertos
- Los dispositivos tienen una tabla la cual está formada por la dirección Bluetooth del dispositivo y el canal por el cual se puede tener acceso a él

Mensajes válidos

Los mensajes válidos que se intercambiarán entre las entidades del protocolo son:

- *LUCRN_NEW_DEVICE*
- *LUCRN_SYNC_DEVICES*

Formato de los mensajes

En la figura 3.5 se presenta el formato del mensaje *LUCRN_NEW_DEVICE*. Para este mensaje el campo de *tipo de mensaje* deberá llenarse con el hexadecimal *0x12*. Dado que no hace uso de ninguna extensión, el campo de *extensión* debe ser igual al hexadecimal *0x00*. Los 48 bits que siguen deben ser llenados con la dirección del dispositivo Bluetooth que acaba de formar el enlace de comunicación. Finalmente, en los últimos 80 bits debe de ir el nombre del dispositivo.

El formato del mensaje *LUCRN_SYNC_DEVICE* se puede observar en la figura 3.6 . Para este mensaje el campo de *Tipo de mensaje* deberá llenarse

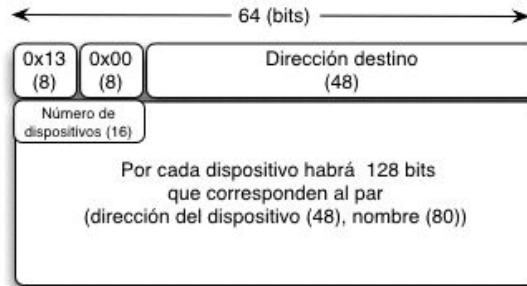


Figura 3.6: Formato del mensaje LUCRN_SYNC_DEVICES.

con el hexadecimal *0x13*. Dado que no usa extensión, el campo de *extensión* debe ser igual a *0x00*. Los últimos 48 bits del encabezado deberán llenarse con la dirección Bluetooth del dispositivo al cual se le van a sincronizar las direcciones. Con respecto al cuerpo del mensaje, los primeros 16 bits corresponderán al número de direcciones Bluetooth que van a ser enviadas. Después de esos 16 bits seguirán 128 bits por cada dirección. Los 128 corresponden a la dirección del dispositivo (48 bits) y el nombre (80 bits).

Reglas

Una vez establecido el canal entre los dos dispositivos se deben cumplir las siguientes reglas:

1. *Agregar a la lista de canales el canal recién creado.*
2. *Agregar una entrada a la tabla de ruteo con la dirección del dispositivo Bluetooth y el canal que se acaba de formar entre ellos.*
3. *Si existen canales adicionales en la lista, enviar por cada uno de ellos el mensaje LUCRN_NEW_DEVICE.*
4. *Verificar la tabla de ruteo y en caso de contar con direcciones adicionales, enviar éstas en un mensaje LUCRN_SYNC_DEVICES al dispositivo que ha creado el canal.*
5. *El dispositivo pasa al estado LUCRN_CONNECTED.*

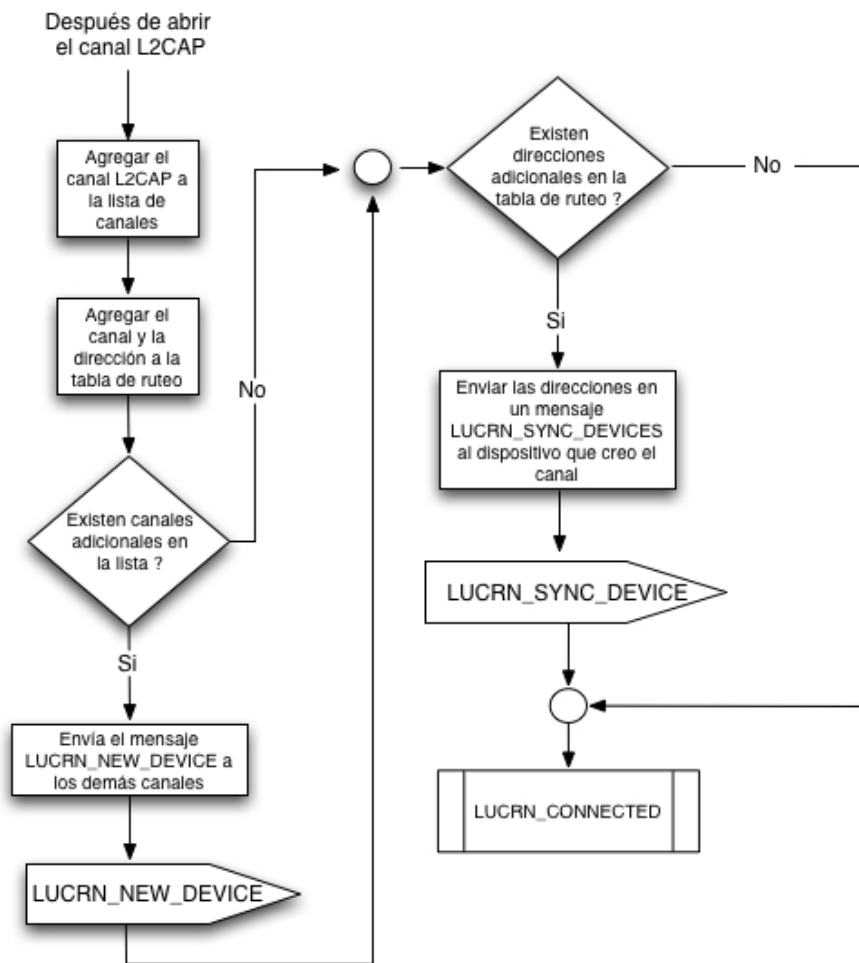


Figura 3.7: Primera parte del diagrama de flujo del protocolo de anuncio de dispositivos.

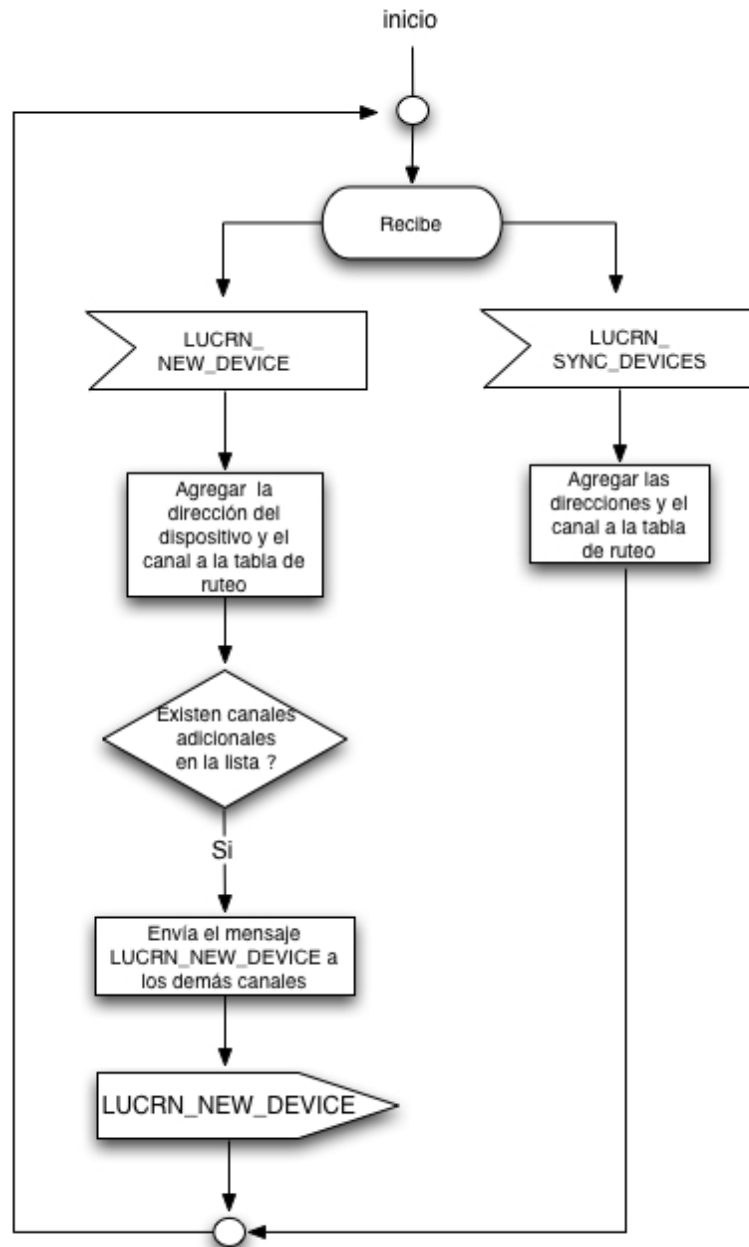


Figura 3.8: Segunda parte del diagrama de flujo del protocolo de anuncio de dispositivos.

Cuando se recibe un mensaje *LUCRN_SYNC_DEVICES*:

1. *Debe agregar cada una de las direcciones a la tabla de ruteo, junto con el canal por el cual llegaron.*

Cuando se recibe un mensaje *LUCRN_NEW_DEVICE*:

1. *Debe agregar la dirección Bluetooth del dispositivo junto con el canal por el cual llegó a su tabla de ruteo.*
2. *Debe reenviarlo a los canales que tiene en su lista y que no corresponden al canal por el cual llegó.*

Diagrama de flujo

En las figuras 3.7 y 3.8 se presentan los diagramas de flujo, en un subconjunto del lenguaje SDL [18], que corresponden a las reglas definidas para este protocolo. No está de más recordar que cada uno de estos diagramas en tiempo de ejecución corresponden a procesos concurrentes que se llevarán a cabo en cada uno de los dispositivos.

3.2.3. Protocolo de ruteo

Debido a que Bluetooth no cuenta con mecanismos nativos para la transferencia de información entre nodos esclavos de una piconet y considerando que las aplicaciones distribuidas deben de poder transferir información entre los diferentes nodos de la red, se ha diseñado este protocolo el cual hace uso de la lista de canales y de la tabla de ruteo, actualizadas en el protocolo subyacente, para lograr transferir un mensaje de un nodo a otro. Solucionando de esta manera el problema que se tiene al no poder transferir información entre los diferentes nodos.

Servicio

El servicio que ofrece el protocolo es la transferencia de paquetes de información entre los diferentes nodos que conforman la red espontánea.

Suposiciones

- *Los dispositivos se encuentran en el estado `LUCRN_CONNECTED`*
- *Los dispositivos tienen una lista de los canales locales que se encuentran abiertos*
- *Los dispositivos cuentan con una tabla de ruteo en la cual pueden encontrar los dispositivos a los cuales se desea enviar información*

Mensajes válidos

Los mensajes que son válidos para este protocolo son:

- `LUCRN_DATA_MSG`

Formato de los mensajes

El formato de un mensaje del tipo `LUCRN_DATA_MSG` se puede observar en la figura 3.9.

En la parte del encabezado se observa que el *tipo de mensaje* (bits 0-7) será igual al hexadecimal `0x00`. No tiene definida ninguna extensión, por tanto, el campo de *extensión* (bits 8-15) tendrá el valor hexadecimal `0x00`. En los últimos bits del encabezado (bits 16-63) debe llevar la información acerca de la dirección Bluetooth del dispositivo al cual va enviado el mensaje.

Con respecto al cuerpo del mensaje, los primeros 48 bits llevarán la dirección Bluetooth del dispositivo que envía el paquete de información. Los 16 bits que siguen tendrán la longitud en bytes del campo de datos. Por último, tenemos al campo de datos que llevará la información que va a ser transmitida. La longitud máxima del campo de datos es de 5248 bits o 656 bytes.

Reglas

Una vez que se recibe un mensaje del tipo `LUCRN_DATA_MSG` se deben seguir las siguientes reglas:

1. *Se debe extraer del mensaje la dirección Bluetooth del dispositivo.*
2. *Si la dirección Bluetooth es igual a la del dispositivo local enviarlo a la capa superior.*

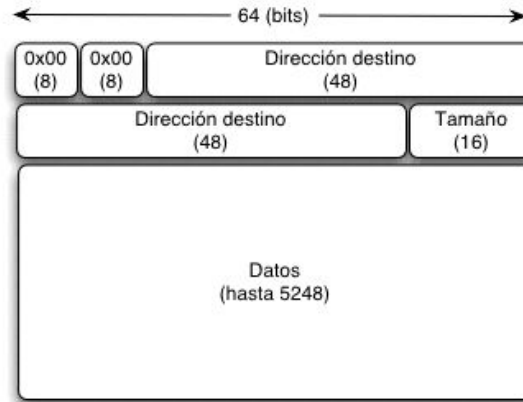


Figura 3.9: Formato del mensaje LUCRN_DATA_MSG.

3. Si la dirección Bluetooth es diferente se debe buscar ésta en la tabla de ruteo y enviarla por el canal adecuado.
4. Si la dirección Bluetooth no se encuentra en la tabla de ruteo desechar el paquete.

Diagrama de flujo

La figura 3.10 presenta el diagrama de flujo del protocolo de ruteo. Al inicio el dispositivo está esperando la llegada de algún mensaje. Luego, al recibir un mensaje del tipo *LUCRN_DATA_MSG*, extrae la dirección Bluetooth al cual va dirigido éste y la compara con su dirección Bluetooth local. Si son iguales pasa la información que contiene el mensaje a las capas superiores. En caso de que la dirección no sea igual a la suya, entonces busca en su tabla de ruteo el canal por el cual debe enviar el mensaje para que llegue al dispositivo. Si no existe ninguna entrada en la tabla que corresponda a la dirección destino entonces desecha el paquete.

3.2.4. Protocolo de movilidad

El protocolo de movilidad trata principalmente con el problema de la salida de los nodos de la red espontánea. El objetivo principal de este pro-

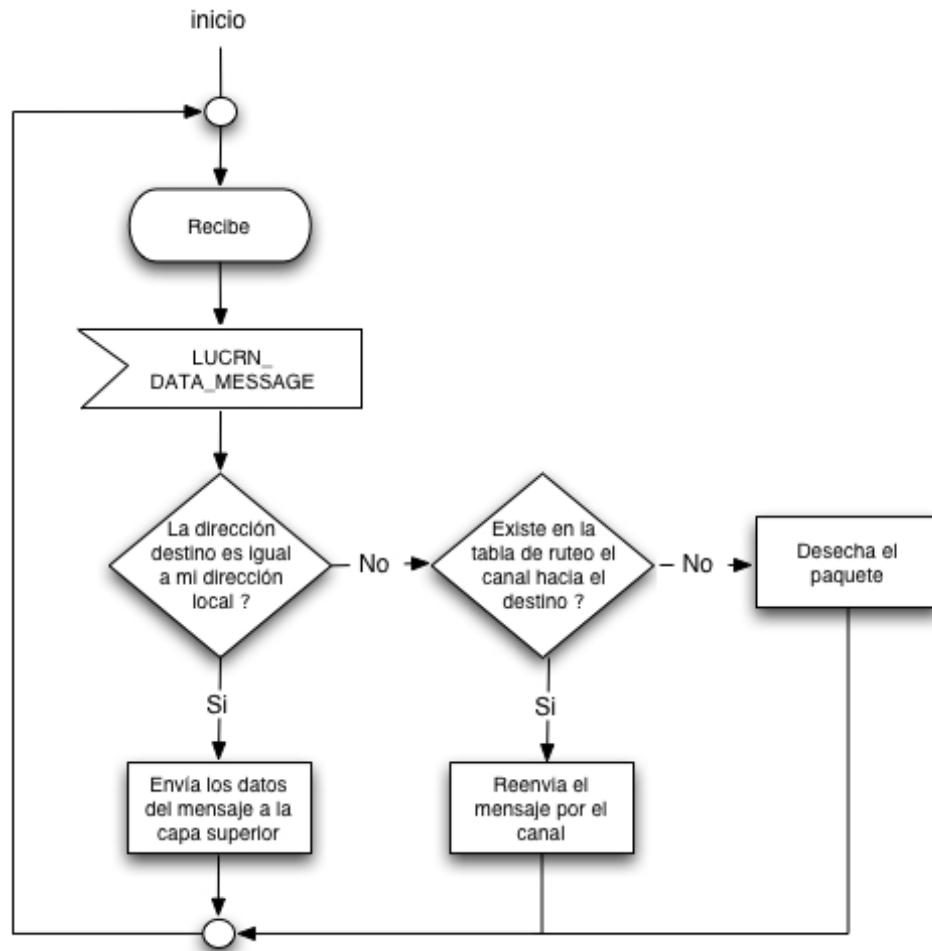


Figura 3.10: Diagrama de flujo del protocolo de ruteo.

protocolo es poder anunciar que un dispositivo va a salir o ha salido de la red espontánea para que los demás nodos actualicen sus listas de canales y tablas. Este problema es particularmente difícil sobretodo porque Bluetooth carece de un medio broadcast sobre el cual todos los nodos estén escuchando, a diferencia de Wi-Fi donde si tiene un medio broadcast.

La complejidad del problema se incrementa debido a que la topología de la red espontánea es variable . Para atacar este problema vamos a clasificar a los nodos con respecto al número de enlaces de comunicación adyacentes que ellos tienen. En base a esto tenemos:

- *Nodo simple*

Este nodo tiene solamente un enlace de comunicación. Lo que implica que sólo tiene una conexión con un dispositivo.

- *Nodo complejo*

Este nodo tiene varios enlaces de comunicación. Lo que implica que está conectado a varios dispositivos.

En la figura 3.11 se presenta una red espontánea formada por 4 dispositivos inalámbricos representados por círculos. En ella se puede distinguir que los nodos A y D son simples debido a que tienen solo un enlace de comunicación adyacente, y los nodos B y C son complejos debido a que, en este caso, ambos tienen 2 enlaces de comunicación. Los nodos simples tienen un peso menor en la red espontánea que los nodos complejos. En el sentido de que ellos van a realizar menos actividades. Por ejemplo, no van a rutear información.

El alcance de este trabajo trata solamente con el problema de la salida de un nodo simple. La salida de un nodo complejo es difícil principalmente porque el nodo complejo tiene el rol de maestro en la piconet y sería necesario la implementación de un algoritmo distribuido para la elección del nuevo maestro. Además es necesario contemplar varios factores como la distancia a la que se encuentra el dispositivo con el cual se tiene reconectar y la fusión de dos redes.

A continuación presentamos los elementos que conforman al protocolo de movilidad para un nodo simple.

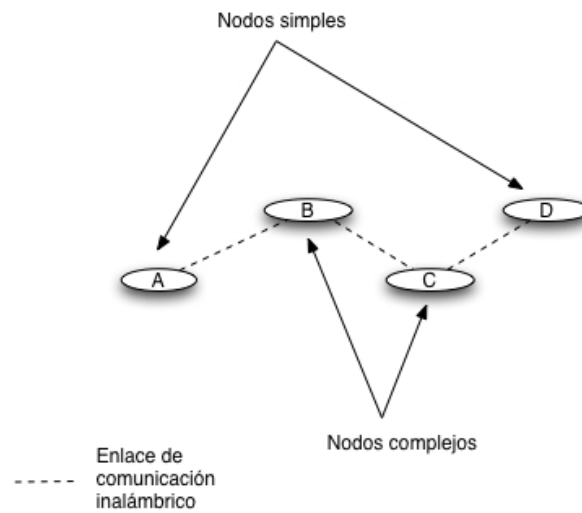


Figura 3.11: Nodos simples y complejos.

Servicio

El servicio que ofrece este protocolo es anunciar a los demás dispositivos que conforman la red espontánea acerca de la eliminación de un enlace de comunicación L2CAP de un nodo simple.

Suposiciones

- *El dispositivo recibe notificaciones de la capa baseband de Bluetooth acerca de la eliminación de un canal*
- *El enlace de comunicación eliminado corresponde al de un nodo simple*
- *Los dispositivos tienen una lista de los canales que tienen abiertos*
- *Los dispositivos tienen una tabla la cual está formada por la dirección Bluetooth del dispositivo y el canal por el cual se puede tener acceso a él*

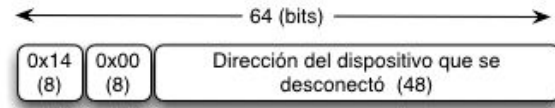


Figura 3.12: Formato del mensaje LUCRN_DISCONNECTING_DEVICE.

Mensajes válidos

Los mensajes válidos para este protocolo son:

- *LUCRN_DISCONNECTING_DEVICE*

Formato de los mensajes

El formato de un mensaje *LUCRN_DISCONNECTING_DEVICE* se puede observar en la figura 3.12. En ella se muestra que el *tipo de paquete* asociado a este mensaje es igual al hexadecimal *0x14*. No usa extensión por tanto el campo de *extensión* es igual al hexadecimal *0x00*. Los últimos 48 bits del encabezado corresponden a la dirección Bluetooth del dispositivo que se ha desconectado.

Reglas

Una vez que el dispositivo ha recibido la notificación del protocolo de la baseband de Bluetooth acerca de la eliminación de un canal. Se deben llevar a cabo las siguientes reglas:

1. *Eliminar de la lista de canales, al canal recién eliminado*
2. *Eliminar de la tabla de ruteo el registro asociado con la dirección Bluetooth del dispositivo que se ha desconectado*
3. *Crear un mensaje LUCRN_DISCONNECTING_DEVICE y enviarlo a cada uno de los canales que se encuentran en la lista de canales*

Cuando se recibe un mensaje *LUCRN_DISCONNECTING_DEVICE*:

1. *Eliminar de la tabla de ruteo el registro asociado con la dirección Bluetooth del dispositivo que se ha desconectado*
2. *Enviar el mensaje LUCRN_DISCONNECTING_DEVICE a los canales de la lista que no corresponden al canal por el cual llegó el mensaje*

Diagramas de flujo

En la figura 3.13 y 3.14 se muestra los diagramas de flujo de las reglas anteriormente expuestas.

3.2.5. Mecanismos de colaboración

Los protocolos definidos en las secciones anteriores se encargan de manejar y administrar todos los eventos que suceden en la capa de comunicaciones. Es decir, se encargan del anuncio de nuevos dispositivos a la red, de la transferencia de información entre todos los nodos y de la salida de nodos simples. El mecanismo de colaboración es un mecanismo cuyo objetivo es proporcionar mecanismos de alto nivel para que las aplicaciones puedan interoperar de una manera transparente y para que la construcción de éstas sea sencilla.

El mecanismo que se diseñó se enfoca principalmente para el desarrollo de aplicaciones colaborativas. Explota la idea de que las aplicaciones colaborativas en los diferentes dispositivos deben proporcionar la misma funcionalidad y su construcción debe ser semejante. Considerando esto, la idea fundamental es que cada aplicación tenga una interfaz similar compuesta de mecanismos o métodos públicos los cuales van a estar disponibles para las demás aplicaciones, así cualquier aplicación de la red espontánea puede hacer uso de éstos métodos para actualizar el estado de las aplicaciones remotas.

Por ejemplo, en un pizarrón electrónico compartido el objetivo principal es mostrar un área de dibujo global para todos los usuarios. De tal manera, que cuando algún usuario realice algún trazo de una figura geométrica esta acción se replique a las demás aplicaciones remotas y los usuarios tengan conocimiento de que el área de dibujo ha cambiado. Dado que las aplicaciones son semejantes en funcionalidad, entonces para replicar una acción basta con mandar está a las aplicaciones de los demás dispositivos para que ellas la reflejen en el área de dibujo.

Con el fin de ilustrar mejor el mecanismo de colaboración vamos a explicar la figura 3.15. En ella se pueden observar 3 usuarios los cuales traen consigo

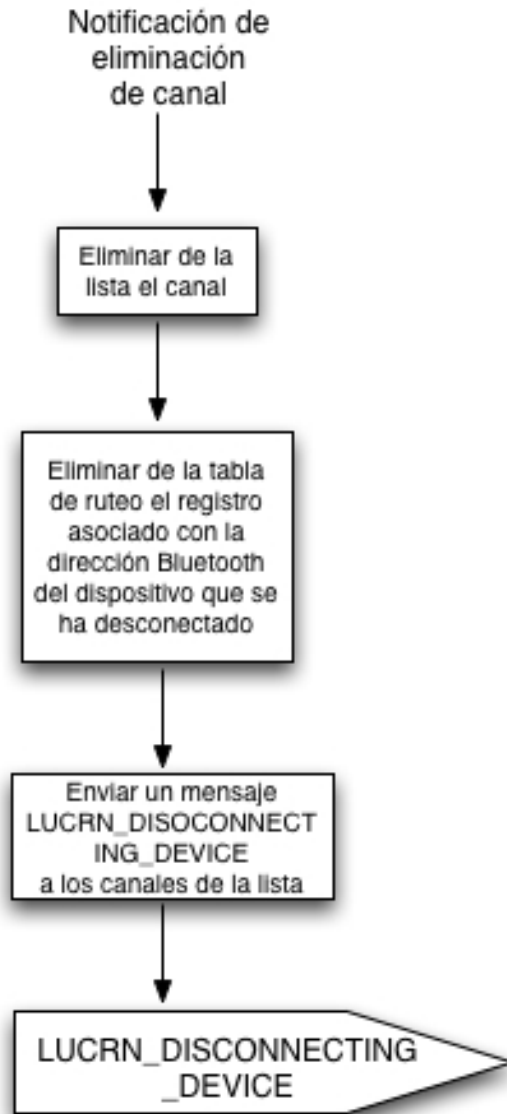


Figura 3.13: 1a. parte del diagrama de flujo del protocolo de movilidad.

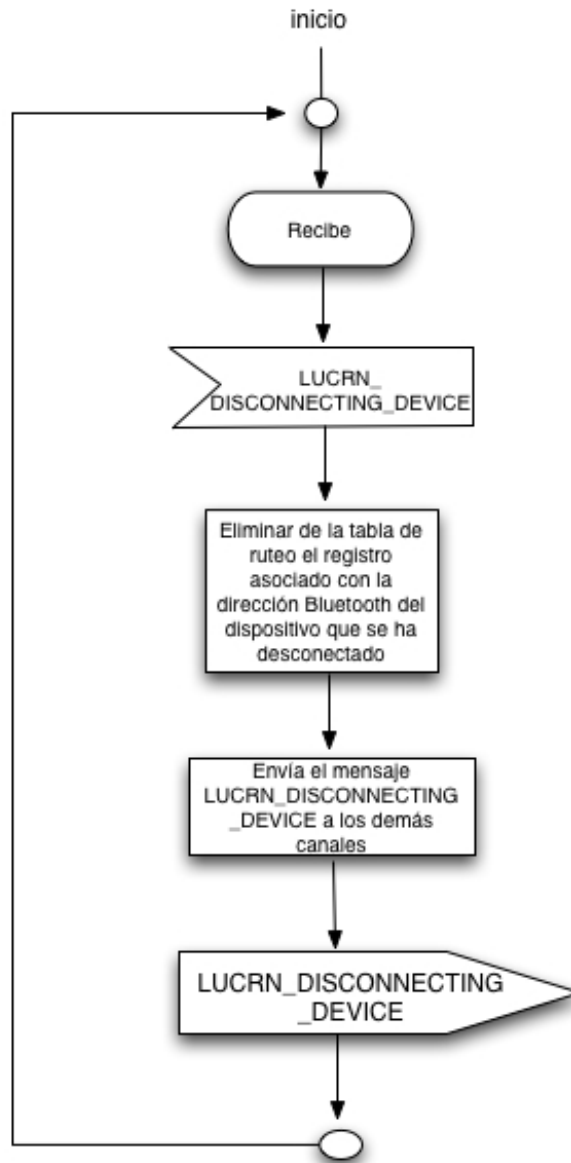


Figura 3.14: 2a. parte del diagrama de flujo del protocolo de movilidad.

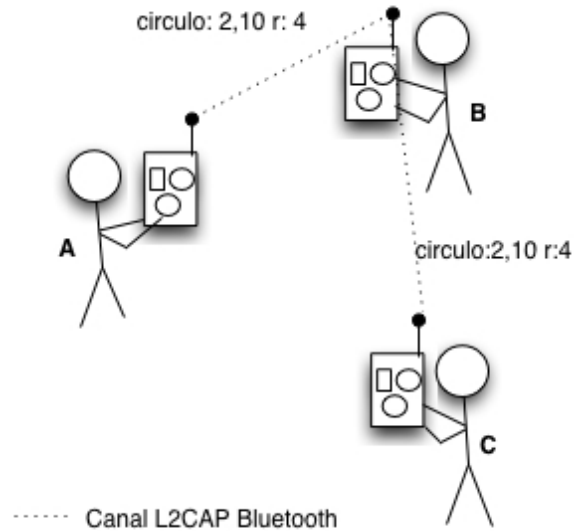


Figura 3.15: Un pizarrón electrónico compartido por varios usuarios.

un asistente digital. Se asume que estos dispositivos ya tienen implementadas las capas inferiores del middleware Lucrn. Por tanto, ya formaron una red espontánea y ya pueden transferir información entre los diferentes nodos. En la figura se muestra que ya han formado una red espontánea por las líneas punteadas que ligan a cada asistente digital. Ahora, el objetivo de las aplicaciones de cada dispositivo móvil es ofrecer un pizarrón compartido a todos los usuarios. En la figura se puede observar círculos y cuadrados en los asistentes digitales de cada usuario, con esto queremos hacer referencia al área de dibujo global. Ahora, cuando algún usuario traza una figura geométrica el mecanismo de colaboración va a actuar enviando esta acción a los demás dispositivos. Es decir, si el usuario trazó un círculo de radio 4 en la posición (2,10), esta acción se manda a las demás aplicaciones remotas, las cuales analizan el paquete, extraen la acción que se debe llevar a cabo, la ejecutan y por consiguiente el área de dibujo es actualizada. En la figura 3.15 a un lado de cada enlace de comunicación (línea punteada) se encuentra un etiqueta con el texto *circulo: 2,10 r: 4* el cual corresponde a la acción llevada a cabo por el usuario A y la cual fue enviada a los demás dispositivos.

Conceptualmente, la idea de este mecanismo de colaboración es que todas las aplicaciones de la red espontánea ofrezcan una interfaz de servicio similar, la cual pueda ser aprovechada por los demás dispositivos para la réplica de acciones. Es importante hacer notar que en las redes espontáneas las aplicaciones deben ser *peer-to-peer*. Por tanto todas son clientes y servidores a la vez. Esto se debe a que no se puede depender de un servidor central en una red espontánea principalmente por la movilidad inherente del modelo de comunicación. Esta diferencia establece una división entre los tradicionales sistemas de colaboración basados en cable y los sistemas de colaboración que proponemos en este trabajo basados en redes inalámbricas.

A continuación explicaremos cada uno de los elementos que conforman el mecanismo de colaboración.

Servicio

El servicio que ofrece este mecanismo es la cooperación entre las diferentes aplicaciones que conforman la red espontánea a través del envío de acciones.

Suposiciones

- *Las aplicaciones fueron construidas haciendo uso del middleware Lucrn.*
- *Los dispositivos han formado una red espontánea y es posible la transferencia de información entre todos los nodos a través de los mecanismos proporcionados por Lucrn.*
- *Los dispositivos cuentan con una tabla donde se publican los acciones públicas de la aplicación. Esa tabla está compuesta por campos para almacenar la acción y una referencia al objeto que va a ser invocado cuando esa acción se presente.*
- *Dado que debe ser posible implementarse en los diferentes dispositivos móviles se asume que las diferentes aplicaciones saben de antemano cuales son las acciones disponibles, ya que éstas no va a ser resueltas en tiempo de ejecución.*

Mensajes válidos

Los mensajes válidos para este mecanismo son:

- `LUCRN_COLLECTION`

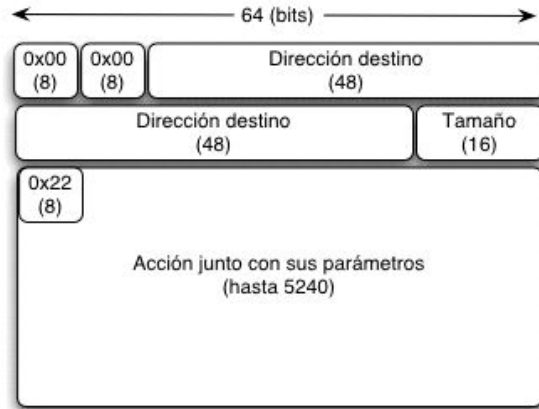


Figura 3.16: Formato de un mensaje LUCRN_COLL_ACTION.

Formato de los mensajes

El formato de los mensajes *LUCRN_COLL_ACTION* se puede apreciar en la figura 3.16. Dado que este mecanismo se encuentra encima del protocolo de ruteo es importante mencionar que este mensaje se encapsulará en un paquete del tipo *LUCRN_DATA_MSG*. Por consiguiente, en la figura se puede observar que el encabezado y los primeros 64 bits del cuerpo del mensaje corresponden a aquellos definidos para el mensaje *LUCRN_DATA_MSG*. Después de los 64 bits, los siguientes 8 servirán para identificar el tipo de mensaje *LUCRN_COLL_ACTION* el cual tiene asociado el hexadecimal 0x22. Los siguientes bits definen la acción que se debe llevar a cabo.

Reglas

Una vez que se ha recibido un mensaje del tipo *LUCRN_COLL_ACTION* se deben cumplir las siguientes reglas:

1. *Extraer la acción y los argumentos del mensaje.*
2. *Extraer de la tabla de acciones públicas el objeto que debe ser invocado para que ejecute esa acción.*
3. *Ejecutar la acción.*

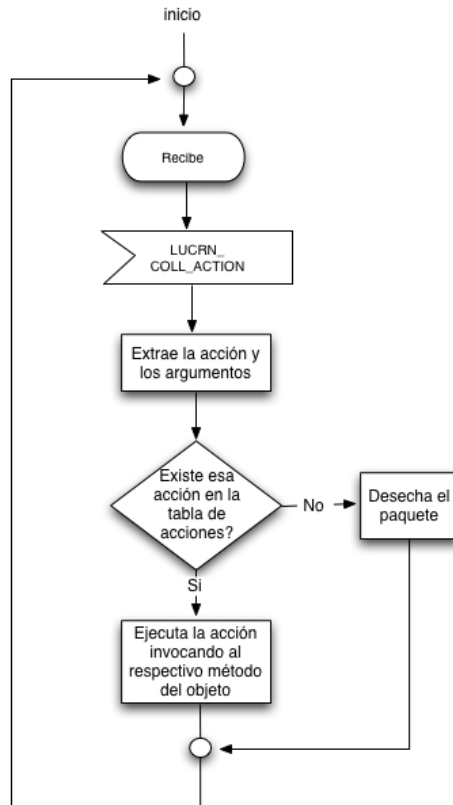


Figura 3.17: Diagrama de flujo del mecanismo de colaboración.

4. *En caso de que no exista en la tabla de acciones el objeto asociado a la acción entonces debe desecharse el paquete.*

Diagrama de flujo

En la figura 3.17 se presenta el diagrama de flujo que corresponde a las reglas anteriormente expuestas.

3.3. Resumen

En este capítulo se especificaron los protocolos de *anuncio de dispositivos*, *de ruteo*, *de movilidad* y *de colaboración* los cuales en conjunto conforman al middleware Lucrn. En cada uno de ellos se presentó el objetivo del servicio, las suposiciones, los mensajes válidos, el formato de cada mensaje, las reglas y el diagrama de flujo. Con el diseño de este middleware ahora las aplicaciones pueden contar con mecanismos para la formación de la red, el envío de información entre todos los nodos, la movilidad de los nodos simples y la colaboración entre aplicaciones.

En el siguiente capítulo se presenta la implementación de este middleware en dos sistemas operativos. La primera parte se dedica a la implementación en Mac OS X el cual administra los recursos de dispositivos móviles como computadoras portátiles. La segunda parte trata acerca de la implementación en Palm OS el cual administra los recursos de dispositivos móviles como asistentes digitales. En ambos casos se describen los diferentes componentes de cada implementación. Además, se presentan diagramas los cuales sirven para entender cada una de las clases o librerías que fueron creadas y las relaciones entre los diversos componentes.

Capítulo 4

Implementación del middleware

El middleware propuesto en el capítulo anterior fue implementado en los sistemas operativos Mac OS X y Palm OS. Con esto logramos cubrir una gran variedad de dispositivos. Desde computadoras de escritorio y portátiles con sistema operativo Mac OS X hasta asistentes digitales con sistema operativo Palm OS.

En este capítulo presentamos los lineamientos que se siguieron para la selección de cada una de las plataformas y de los dispositivos móviles. También explicamos ciertos aspectos que fueron considerados para la implementación del middleware en los diferentes sistemas operativos. Finalmente, para las dos plataformas, Mac OS X y Palm, se describen los componentes desarrollados los cuales sirven para crear un entorno en el cual diferentes aplicaciones distribuidas en dispositivos heterogéneos pueden interoperar con la mínima configuración por parte del usuario.

4.1. Selección de la plataforma

Hoy en día existe una plétora de dispositivos móviles en el mercado, entre los más populares encontramos:

- Computadoras portátiles
- Asistentes digitales
- Teléfonos celulares

El uso de los dispositivos anteriores va en aumento, sin embargo, existen diferencias entre ellos. Algunos como las computadoras portátiles tienen mayor poder de procesamiento y de almacenamiento comparados con los asistentes digitales. Esto tiene un impacto sobretodo en la programación de las aplicaciones y en la ejecución de las mismas. Sin embargo, gracias a los continuos avances tecnológicos, en un futuro va a ser posible la construcción de procesadores más poderosos con menor consumo de energía y es muy probable que se tengan dispositivos móviles muy poderosos con grandes capacidades de memoria y procesamiento. Esto viene a reforzar la importancia de este trabajo, ya que en el futuro cercano se tendrán que desarrollar aplicaciones las cuales involucren a una gran variedad de dispositivos y el factor clave será la integración transparente de todos ellos para poder desarrollar nuevos escenarios y no tan sólo para el mejoramiento de los ya existentes como se ha venido dando en los últimos años.

Para la implementación del middleware Lucrn los dispositivos móviles seleccionados fueron computadoras portátiles y asistentes digitales. Esto se debe a su proliferación en el mercado y además por que pueden equiparse con tecnología de comunicaciones inalámbrica Bluetooth.

La selección también estuvo determinada por el sistema operativo. Se buscaron sistemas operativos con las siguientes características:

- *Que implementen la pila de protocolos de Bluetooth*
- *Que proporcionen una biblioteca o un conjunto de clases para realizar la programación del dispositivo Bluetooth*
- *Que la biblioteca o el conjunto de clases proporcionado esté disponible a los programadores y sea estable*

Después de evaluar tres plataformas como: Linux, Mac OS X y Palm OS. Las plataformas seleccionadas fueron Mac OS X y Palm OS. La decisión de utilizar Mac OS X a Linux se debió a que al momento de realizar la evaluación la pila de protocolos de Bluetooth en Linux aún se encontraba en una fase preliminar y frecuentemente tenía cambios. La plataforma de Windows no se evaluó debido al costo del API de Bluetooth.

Las versiones de los sistemas operativos utilizados son las siguientes:

- *Mac OS X versión 10.2*
- *Palm OS versión 4.0*

Ambos sistemas operativos tienen implementado la pila de protocolos de Bluetooth y proporcionan APIs para el manejo y la programación de los enlaces de comunicación inalámbricos. Los APIs no tienen ningún costo y están disponibles para los programadores.

Los dispositivos que se compraron y que fueron utilizados para el proyecto tienen las siguientes características:

- *Computadora portátil Apple iBook*
 - *384 MB de memoria*
 - *Procesador Power PC G3 600MHz*
 - *20 GB disco duro*
 - *Sistema operativo Mac OS X Versión 10.2*
 - *Adaptador USB Bluetooth D-Link DBT-120*
- *Asistente digital Palm m515*
 - *16 MB de memoria*
 - *Procesador Motorola Dragonball 33MHz*
 - *Sistema Operativo Palm OS 4*
 - *Tarjeta de expansión Bluetooth SDIO para Palm*

Por razones de eficiencia y de manejo adecuado de los canales de comunicación de Bluetooth, fue importante para la implementación del middleware Lucrn que las plataformas ofrezcan mecanismos para manejar los canales L2CAP. Al menos los dos ambientes, Palm OS y Mac OS X, los ofrecen.

4.2. Ambientes de desarrollo

Para la implementación del middleware Lucrn y de las diferentes aplicaciones se utilizaron las siguientes herramientas:

- *Mac OS X*
 - *Project Builder versión 2.2*

Es una herramienta que esta diseñada para facilitar el desarrollo de aplicaciones, plug-ins y frameworks utilizando los lenguajes C, C++, Objective-C, AppleScript y Java en Mac OS X.

Proporciona mecanismos para la organización del proyecto, la edición del código, búsqueda y navegación además de un depurador.

- *Interface Builder versión 2.3*

Es una herramienta para la creación de interfaces gráficas de usuario en Mac OS X.

- *Los frameworks: Application Kit, Foundation, Bluetooth*

Son un conjunto de clases construidas encima de los APIs tradicionales de UNIX y proporcionan al desarrollador clases para el creación de objetos gráficos, para el manejo de estructuras de datos, para la configuración de los enlaces de comunicación inalámbricos. Mac OS X proporciona un gran variedad de frameworks. Pero los más usados en el proyecto fueron los mencionados anteriormente.

- *Palm OS*

- *PRC-Tools*

Es una colección de herramientas para programar aplicaciones para Palm OS utilizando los lenguajes C y C++. Estas herramientas trabajan en ambientes UNIX como Mac OS X y Linux.

- *Emulador de Palm 3.5*

El emulador de Palm OS es un software que emula el hardware de varios modelos de asistentes digitales Palm.

- *Palm OS SDK 5.0*

Es un conjunto de bibliotecas y archivos de encabezados necesarios para programar aplicaciones en Palm.

- *Palm OS Bluetooth SDK*

Es un conjunto de bibliotecas y archivos de encabezados para desarrollar aplicaciones que utilicen tecnología de comunicación inalámbrica Bluetooth.

Estas herramientas están disponibles en internet y no tienen ningún costo. Destaca el ambiente de desarrollo para Palm el cual proporciona compiladores y herramientas que pueden ser ejecutadas en Mac OS X o Linux. Así como también un emulador el cual hace que el desarrollo de aplicaciones sea más amigable y rápido. Sin embargo, debido al recién surgimiento de Bluetooth la versión actual del emulador no sirve para ejecutar aplicaciones que hagan uso

de éste, lo que implica que las aplicaciones tienen que ejecutarse directamente en el dispositivo.

4.3. Implementación en Mac OS X

La implementación en Mac OS X se realizó utilizando el lenguaje orientado a objetos *Objective-C*. Este lenguaje es una extensión del estándar ANSI C y es considerado un híbrido entre los lenguajes C y Smalltalk. El lenguaje es muy dinámico, esto se debe a que el compilador preserva gran cantidad de información de los objetos para su uso en tiempo de ejecución. Por tanto muchas decisiones que antes se hacían en tiempo de compilación ahora se realizan en tiempo de ejecución. Esto le da gran poder y flexibilidad al lenguaje. Se distingue de otros lenguajes principalmente por tener un tipo especial de datos que sirve como un apuntador a cualquier objeto [19].

Para controlar los enlaces de comunicación inalámbricos se empleó el framework de Bluetooth que proporciona el sistema y el cual consiste en un conjunto de clases que sirven para la creación y administración de los canales de comunicación. Sobre estas clases se construyeron nuevas las cuales a su vez forman un nuevo framework que corresponde al middleware Lucrn.

Como se muestra en la figura 4.1 la implementación puede verse como un conjunto de capas que proporcionan un servicio a la capa superior y abstraen de las complejidades inherentes de los protocolos que cada una implementa.

El diseño de las clases es modular, flexible y fue concebido para que evolucione conforme lo haga el middleware. Es decir, se diseñó de tal manera que sea posible la sustitución de una capa siempre y cuando las nuevas clases que la conforman adopten ciertos protocolos definidos en el lenguaje *Objective-C*. Este diseño permite experimentar con nuevos protocolos de red sin tener que reimplementar todas las capas.

Para lograr este diseño se utilizaron los protocolos que pueden ser declarados en el lenguaje *Objective-C*. Un *protocolo* es simplemente una lista de métodos los cuales no están asociados a ninguna clase y no dependen de alguna jerarquía. Para ciertas aplicaciones en determinado momento no es de interés conocer la identidad de la clase, únicamente es relevante que *adopte* el protocolo, es decir, que implemente todos y cada uno de los métodos que se declararon en éste.

Los protocolos son adecuados sobre todo cuando la clase que va a implementar los métodos se desconoce o se tiene implementar posteriormente. El

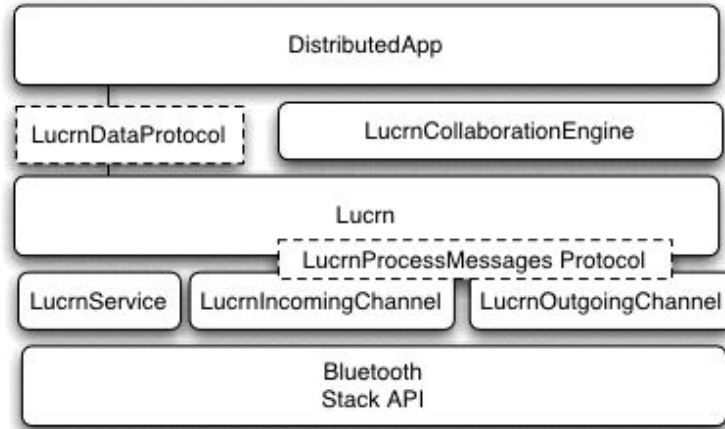


Figura 4.1: Componentes principales de la implementación en Mac OS X.

poderoso sistema de tiempo de ejecución permite obtener toda la información acerca de la clase que implementa el protocolo. En la figura 4.1 se pueden distinguir los protocolos por los rectángulos dibujados con líneas punteadas y se puede apreciar como éstos sirven para 'conectar' a los diferentes niveles.

El diseño se organizó en 5 niveles o capas los cuales se pueden ver en la figura 4.1, a continuación explicamos cada uno de ellos de una manera general, posteriormente se explican las clases de cada nivel.

- *El primer nivel*

El primer nivel corresponde al framework Bluetooth proporcionado por el sistema operativo. Este framework se divide en:

1. *IOBluetooth.framework*
2. *IOBluetoothUI.framework*

El primero proporciona un conjunto de clases y biblioteca de funciones que permiten tener acceso a los protocolos de Bluetooth implementados en el sistema operativo. El segundo corresponde a un conjunto de objetos gráficos para realizar el descubrimiento de servicios y dispositivos.

- *El segundo nivel*

El segundo nivel está formado por clases que administran en forma particular cada uno de los canales inalámbricos de entrada y salida. En este nivel se implementa el protocolo de *anuncio de dispositivos*. Su objetivo principal es que la capa superior no sepa nada acerca de los enlaces de comunicación inalámbricos con respecto a su configuración y administración. El servicio que ofrece a la capa superior es el envío y recepción de datos a través de los enlaces inalámbricos.

En este nivel se encuentran las clases:

- *LucrnService*
- *LucrnIncomingChannel*
- *LucrnOutgoingChannel*

- *El tercer nivel*

El segundo nivel está formado por la clase *Lucrn*. Ésta implementa los protocolos de ruteo y de movilidad. También implementa el protocolo *LucrnProcessMessages* y a partir de los mensajes que llegan del nivel inferior el rutea éstos o los pasa al nivel superior. Además de implementar varios de los protocolos, es responsable de la creación de los objetos que se encuentran en el nivel inferior y esto se realiza en su proceso de inicialización. La funcionalidad de esta capa permite que el nivel superior pueda enviar mensajes a todos los dispositivos que se encuentran en la red. Como si se tuvieran enlaces punto a punto entre todos los dispositivos.

- *El cuarto nivel*

El cuarto nivel está formado por la clase *LucrnCollaborationEngine* implementa el protocolo de colaboración y permite que las aplicaciones distribuidas en los diferentes dispositivos móviles puedan colaborar.

- *El quinto nivel*

El quinto nivel corresponde a la aplicación final la cual de una manera implícita hace uso de cada uno de los protocolos inferiores para el desarrollo de escenarios de colaboración sobre dispositivos móviles.

Es importante distinguir que el diseño permite que la capa de aplicación decida si desea emplear la capa de colaboración o no. Puede suceder con algunas aplicaciones, por ejemplo un control remoto de aplicaciones, en las que la funcionalidad de la clase *Lucrn* es suficiente para su desarrollo.

Cada uno de los componentes de las diferentes capas se explica con más detalle a continuación y finalmente en la sección que sigue se muestran las principales variables de instancia y métodos de cada una de ellas. También para ilustrar la relación que existe entre ellas se explica el diagrama UML que corresponde al diseño de clases en Mac OS X.

- *LucrnIncomingChannel*

La clase *LucrnIncomingChannel* trata con el manejo de los canales de entrada. Un *canal de entrada* se crea cuando algún dispositivo externo desea hacer uso del servicio que está ofreciendo el dispositivo local. Esta clase maneja canales L2CAP y los mecanismos proporcionados por framework de Bluetooth del sistema operativo le permite saber cuando un canal se ha abierto o cuando está recibiendo información por alguno de estos canales de entrada. Utiliza los mecanismos del nivel inferior para enviar información hacia el dispositivo.

- *LucrnOutgoingChannel*

La clase *LucrnOutgoingChannel* trata con el manejo de los canales de salida. Un *canal de salida* se crea cuando el dispositivo local desea conectarse con otro dispositivo para hacer uso de algún servicio que esté ofreciendo. Al igual que la clase anterior maneja canales L2CAP y los mecanismos del sistema le permiten saber cuando se ha conectado exitosamente con el dispositivo externo y cuando está recibiendo información por alguno de los canales de salida. Utiliza los mecanismos del nivel inferior para enviar información hacia el dispositivo externo.

- *LucrnService*

La clase *LucrnService* trata con la publicación de un servicio para que los dispositivos externos puedan saber de él y se conecten con él. Esta clase está relacionada con *LucrnIncomingChannel* en el sentido de que debe existir un canal de entrada para que el servicio que se ofrece por este canal pueda ser publicado.

- *Lucrn*

La clase *Lucrn* es la que tiene mayor funcionalidad de todas. Generalmente solo existirá una en las aplicaciones que utilicen el framework. Cuenta con mecanismos internos los cuales permiten realizar el ruteo de información por los canales de entrada o salida adecuados. Tiene implementado el protocolo de movilidad y presenta una interfaz para el envío de información por los canales de comunicación. También controla las clases del nivel inferior y coordina su trabajo en el sentido que puede detener la publicación de un servicio o el envío de información a un dispositivo específico o el envío de un mensaje a todos los dispositivos de la red.

Esta clase ofrece suficiente funcionalidad para que una aplicación pueda ofrecer un servicio, comunicarse con todos los dispositivos de la red, enviar información a algún dispositivo en particular o a todos los dispositivos.

- *LucrnCollaborationEngine*

La clase *LucrnCollaborationEngine* trata con la implementación del mecanismo diseñado para que las aplicaciones de los diferentes dispositivos puedan colaborar a través del envío de las acciones que ocurren en las diferentes aplicaciones. Esta clase se construyó encima de la clase *Lucrn* y proporciona una funcionalidad más especializada orientada principalmente al desarrollo de aplicaciones colaborativas.

4.4. Diagrama de clases

Una vista más detallada de cada clase que conforman al middleware *Lucrn* se muestra en el diagrama de la figura 4.2. En ella podemos observar las clases con sus variables de instancia y métodos más significativos. Además de las relaciones que existen entre cada una de ellas.

Una parte esencial del diseño son los protocolos formales definidos para que fueran adoptadas por las diferentes clases y funcionen como la conexión entre los diferentes niveles.

Dos protocolos formales fueron definidos en el diseño:

- *LucrnProcessMessage*

Este protocolo es adoptado por la clase *Lucrn*

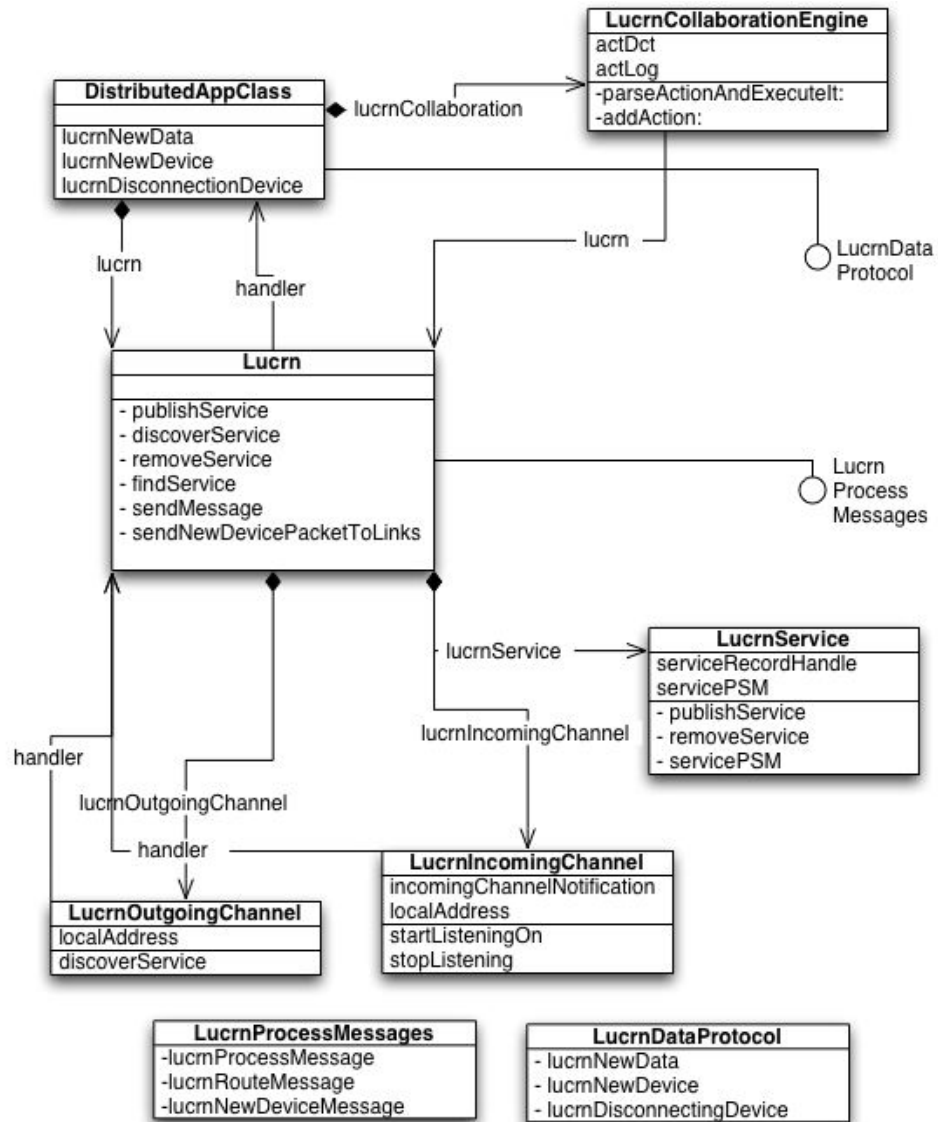


Figura 4.2: Diagrama de clases.

- *LucrnDataProtocol*

Este protocolo es adoptado por la aplicación distribuida dependiendo si ésta desea hacer uso del protocolo de colaboración o no.

El protocolo *LucrnProcessMessage*, figura 4.2 tiene declarados los siguientes métodos:

- *lucrnProcessMessage*

Se usa cuando se detecta que un mensaje debe ir a la capa de aplicación.

- *lucrnRouteMessage*

Se usa cuando se detecta que un mensaje debe ser reenviado a otro dispositivo.

- *lucrnNewDeviceMessage*

Se usa cuando se detecta que un nuevo dispositivo se ha integrado a la red.

El protocolo es adoptado por la clase *Lucrn* y funciona de la siguiente manera. Las clases *LucrnIncomingChannel* y *LucrnOutgoingChannel* tienen una variable, *handler*, que es un apuntador a un objeto genérico y en este caso particular apunta a un objeto de la clase *Lucrn*. Este objeto *handler* debe adoptar el protocolo *LucrnProcessMessage* debido a que cada vez que suceda algún evento en los canales de comunicación se invocará al respectivo método. Es decir, si algún objeto de la clase *LucrnIncomingChannel* o *LucrnOutgoingChannel* ha recibido información éste la analizará y si detecta que no es para el dispositivo local invocará el mensaje *lucrnRouteMessage* del objeto de la capa superior para que este realice las acciones necesarias. De igual manera sucede para los métodos *lucrnProcessMessage* y *lucrnNewDeviceMessage*.

El protocolo *LucrnDataProtocol* tiene declarados los siguientes métodos:

- *lucrnNewData*

Se usa cuando la aplicación ha recibido información

- *lucrnNewDevice*

Se usa cuando un nuevo dispositivo se ha integrado a la red

- *lucrnDisconnectingDevice*

Se usa cuando algún dispositivo se ha salido de la red

El protocolo *LucrnDataProtocol* es adoptado por la capa de aplicación. Este protocolo trabaja en conjunto con la clase *Lucrn* y esta clase es la responsable de invocar a cada uno de los métodos del protocolo. Es decir, si el objeto *Lucrn* recibe información que debe de ir directamente a la aplicación se la enviará a través del método *lucrnNewData*. De igual manera sucede con los demás métodos del protocolo.

En esta sección se revisaron ciertos aspectos del diseño de clases del middleware *Lucrn* para Mac OS X, se explicaron los diferentes niveles y la funcionalidad de cada una de ellos. Finalmente se describió como están relacionadas unas con otras y como los protocolos conectan a los distintos niveles. En la sección que viene explicamos la implementación del middleware en la plataforma Palm OS .

4.5. Implementación en Palm OS

Uno de los aspectos fundamentales de este trabajo es lograr que varios dispositivos móviles aprovechen el middleware *Lucrn* para lograr que aplicaciones distribuidas en los distintos dispositivos puedan cooperar y lograr la creación de nuevos escenarios. Estos escenarios deben de ser de fácil puesta en marcha y un uso sencillo. Además deben involucrar dispositivos heterogéneos los cuales en muchos casos son distintos en poder de procesamiento, consumo de energía y almacenamiento.

La programación en dispositivos móviles con recursos limitados es diferente a la programación de los dispositivos estáticos con una fuente de energía infinita y muchos recursos. En la siguiente sección explicamos algunos puntos esenciales que deben ser tomados en cuenta para la programación de aplicaciones en asistentes digitales y posteriormente explicamos el diseño y cada uno de los componentes de la implementación en Palm OS.

4.5.1. Diseño de aplicaciones para Palm

Para el diseño y la programación de aplicaciones en dispositivos Palm es necesario considerar los siguientes aspectos:

- *Una pantalla reducida*

El tamaño de la pantalla de un dispositivo Palm es de 160x160 pixeles distribuidos en un área de 6cm. Esto obliga a que las interfaces gráficas que se presentan al usuario sean rediseñadas para que sean agradables y funcionales a los usuarios de estos dispositivos.

- *Limitar el número de textos de entrada*

El dispositivo Palm carece de un teclado lo que ocasiona que la entrada de datos al dispositivo sea compleja. Por tanto es adecuado limitar el número de textos de entrada.

- *La aplicación no debe ocupar muchos recursos*

El dispositivo carece de grandes recursos en términos de velocidad de procesamiento y espacio de almacenamiento. Por tanto las aplicaciones deben desarrollarse para que ocupen los mínimos recursos necesarios.

- *La aplicación debe ser rápida*

Debido a que los usuarios están en movimiento, posiblemente algunos de ellos realizando diversas tareas, las aplicaciones deben ofrecer un mejor tiempo de respuesta.

Para realizar la implementación del middleware y de las aplicaciones estos puntos fueron tomados en cuenta. En la sección siguiente explicamos cada uno de los componentes de la implementación.

4.5.2. Componentes de la implementación

La implementación en Palm OS se realizó utilizando el lenguaje C. La razón principal para el empleo de este lenguaje es el tamaño pequeño de las librerías que genera. Como se vio en la sección anterior es un factor importante que debe ser tomado en cuenta para el desarrollo de aplicaciones en Palm OS.

En la Figura 4.3 se muestran los componentes de la implementación del middleware en Palm OS. Como se puede apreciar la implementación consta de 3 niveles encima de la pila de protocolos de Bluetooth en Palm OS. La implementación en Palm OS es mucho más compacta y no es tan sofisticada como la implementación en Mac OS X.

A continuación explicamos cada uno de los niveles:

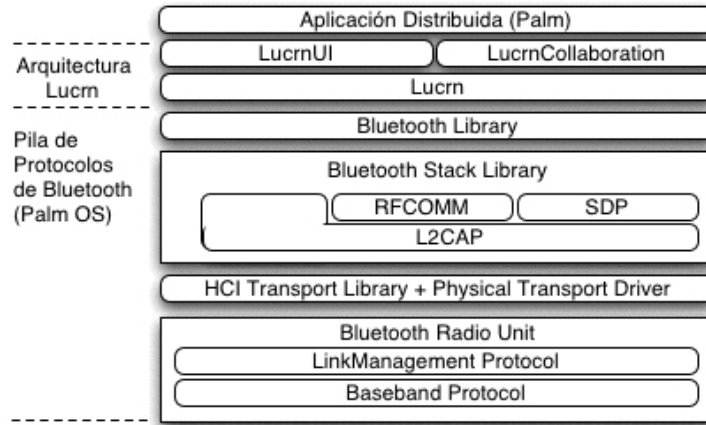


Figura 4.3: Componentes principales de la implementación en Palm OS.

- *El primer nivel*

El primer nivel corresponde a la pila de protocolos de Bluetooth implementada en el sistema operativo Palm OS. En particular, se hizo uso de la biblioteca Bluetooth la cual es un conjunto de funciones que permiten la configuración y creación de los enlaces inalámbricos. Se utilizaron principalmente las funciones para creación de canales L2CAP y las diferentes funciones para la publicación de un servicio y para el envío de información. La biblioteca trabaja principalmente a través de eventos y callbacks.

- *El segundo nivel*

El segundo nivel está formado por el módulo *Lucrn* el cual implementa cada uno de los callbacks del nivel inferior y tiene el control sobre todo lo que está sucediendo en el canal de comunicación inalámbrico. En este nivel se implementan los protocolos de anuncio de dispositivos, ruteo y movilidad. También se implementan funciones que son útiles para configurar los enlaces de comunicación, para empaquetar y desempaquetar datos. Su principal objetivo es presentar una interfaz de programación de alto nivel para que sobre ella se implemente un protocolo de colaboración más avanzado.

- *El tercer nivel*

El tercer nivel está formado por el módulo que implementa el protocolo de colaboración y un módulo que proporciona funciones especiales para manejar los objetos gráficos que se presentan en la interfaz gráfica del usuario. Este módulo fue necesario debido a que los objetos gráficos proporcionados por el SDK 5.0 carecen de cierta funcionalidad.

- *El cuarto nivel*

El cuarto nivel está formado por la aplicación final, la cual hace uso implícito de cada uno de los niveles inferiores para lograr la colaboración entre dispositivos heterogéneos.

Los componentes y una breve descripción de las actividades que realiza cada uno de ellos se hace a continuación:

- *Lucrn*

Este componente del middleware es el encargado de administrar y configurar los enlaces de comunicación Bluetooth. Para realizar la configuración y el manejo de estos enlaces implementa los callbacks del API de Bluetooth de Palm OS para tomar las acciones necesarias en base a los eventos que ocurren en el canal de comunicación inalámbrico.

Implementa los protocolos de anuncio de dispositivos, el protocolo de ruteo y el protocolo de movilidad. Para poder realizar la implementación de estos protocolos en su archivo de encabezado se definen diferentes estructuras de datos que corresponden a los mensajes que circulan por los enlaces inalámbricos. Además, proporciona funciones de alto nivel a la capa superior. En este módulo también se realiza la publicación de servicios y la administración de los canales de entrada y salida.

- *LucrnUI*

El ambiente de desarrollo utilizado carece de cierta funcionalidad para determinados objetos gráficos. Este módulo extiende la funcionalidad de algunos objetos gráficos para que la programación de aplicaciones sea más sencilla. Un ejemplo de un objeto gráfico mejorado por este módulo es un menú pop-up que se va actualizando con los dispositivos que actualmente están formando parte de la red.

- *LucrnCollaboration*

Este componente implementa el protocolo de colaboración que fue diseñado para que las aplicaciones distribuidas pudieran interoperar de una manera transparente. La comunicación con la aplicación se hace a través de punteros a función los cuales son invocados cuando se tiene información o eventos especiales como la desconexión de un dispositivo.

En el diagrama de la figura 4.4 puede observarse como se establecen las relaciones entre los distintos componentes de la implementación.

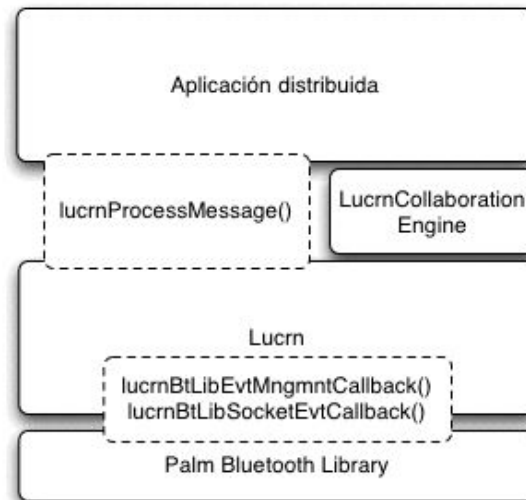


Figura 4.4: Conexiones entre los componentes.

El módulo *Lucrn* recibe información del canal a través de los callbacks que define. Un callback es un puntero a una función la cual va a ser invocada cuando suceda algún evento. En particular los callbacks que define *Lucrn* permite que este módulo sepa acerca de cuando un canal de entrada es creado e información como la negociación de la unidad máxima de transmisión o la desconexión de algún dispositivo. También a través del callback sabe cuando ha llegado información de algún canal.

El módulo *LucrnCollaborationEngine* se relaciona con *Lucrn* a través del callback `lucrnProcessMessage()`. Este callback va a ser invocado cuando el

módulo Lucrn tenga información que va a ir directo a la aplicación. Al nivel de Lucrn los mensajes tiene encabezados de control, Lucrn se encarga de distinguir cada uno de los mensajes y cuando existe alguno que tiene que ir directo al nivel superior lo va a enviar a través de este callback.

4.6. Resumen

En este capítulo se revisaron las implementaciones del middleware Lucrn en Mac OS X y Palm OS. Las dos implementaciones pueden ser vistas como capas las cuales ofrecen un servicio a la capa superior. La implementación en Mac OS X es orientada a objetos y hace uso de protocolos para conectar a los diferentes niveles. El diseño de Mac OS X ha mostrado ser flexible y permite la sustitución de clases en una capa siempre y cuando estas adopten el protocolo de la capa. La implementación en Palm OS es procedural y para establecer las conexiones entre los diferentes niveles hace uso de callbacks.

En el siguiente capítulo se utilizan cada una de estas implementaciones para desarrollar diversos tipos de aplicaciones. Se presentan los resultados que se han obtenido del diseño de este middleware y de la implementación del mismo.

Capítulo 5

Pruebas y Resultados

Con el desarrollo de este middleware ha sido posible construir diferentes escenarios los cuales ocurren de una manera casual, sin ninguna infraestructura computacional predeterminada, y entre computadoras portátiles y asistentes digitales equipados con tecnología inalámbrica Bluetooth.

Este capítulo está dividido en dos partes: La primera parte indica las características de los escenarios que pueden desarrollarse con el middleware Lucrn y muestra tres aplicaciones que funcionan en una red espontánea de computadoras portátiles y asistentes digitales. La segunda parte concluye el capítulo presentado los resultados que se han obtenido con el desarrollo de este middleware.

5.1. Escenarios que pueden ser construidos con Lucrn.

El middleware que se ha diseñado, junto con todos los mecanismos que lo conforman, permite la construcción de escenarios de cómputo móvil con las siguientes características:

- *Los escenarios ocurren de manera casual.*

Con esto enfatizamos que estos escenarios pueden ocurrir de una manera casual, entre diferentes personas y entre diferentes dispositivos. Lo que implica que pueden suceder en lugares donde no se tenga una infraestructura eléctrica o de comunicaciones predeterminada. Por tan-

to, sólo se puede depender de la batería de los dispositivos y de las capacidades inalámbricas de cada uno de ellos.

En particular, el middleware Lucrn emplea la tecnología inalámbrica Bluetooth como el medio para transferir información de manera inalámbrica entre los dispositivos y para la creación casual de los enlaces de comunicación entre ellos. Cabe destacar que ahora las aplicaciones dependen de una infraestructura de software –el middleware Lucrn– para poder interoperar de manera adecuada en este entorno.

- *El área en la que se encuentran los usuarios está bien delimitada.*

La tecnología inalámbrica sobre la cual está construido el middleware, impone ciertas características a los escenarios, en particular, Bluetooth ofrece conectividad entre los dispositivos siempre y cuando éstos no estén a una distancia mayor a diez metros. Esta restricción obliga a que los usuarios conserven una determinada distancia para que sea posible el envío de información entre los dispositivos.

- *Los usuarios utilizarán dispositivos heterogéneos con sistema operativo Mac OS X o Palm OS.*

El middleware fue diseñado para que fuera implementado en dispositivos móviles que en la actualidad han tenido una gran aceptación como: asistentes digitales y computadoras portátiles, con esto aprovechamos la proliferación que actualmente tienen éstos y proporcionamos mecanismos para que las aplicaciones de los distintos dispositivos puedan interoperar, dando lugar a nuevos dominios de aplicación. El middleware se ha implementado en los sistemas operativos Mac OS X y Palm OS, por ello los dispositivos que pueden ser empleados son computadoras portátiles, computadoras de escritorio y asistentes digitales que utilicen los sistemas operativos antes mencionados..

- *Los usuarios no son expertos*

El middleware supone que los usuarios no son expertos, por tanto, evita que ellos tengan que realizar alguna configuración para que se establezca la comunicación entre los dispositivos. De manera general, el usuario sólo tiene que presionar un botón para buscar los dispositivos que se encuentran en el rango de comunicaciones, seleccionar alguno de los que esté ofreciendo el servicio buscado y automáticamente se forma

5.1. ESCENARIOS QUE PUEDEN SER CONSTRUIDOS CON LUCRN.81

un canal de comunicación el cual es utilizado por los mecanismos del middleware para integrar al nuevo dispositivo a la red espontánea.

Una vez definidas las características de los escenarios que pueden ser creados con *Lucrn*, encontramos que el middleware es de gran utilidad para el desarrollo de:

- Escenarios de colaboración digital.

Es muy común encontrar en el sector industrial o educativo que un grupo de personas se reúnan para elaborar en conjunto un determinado documento, un esquema de procesos, definir un proyecto, etc. Estos escenarios generalmente se llevan a cabo utilizando herramientas como: pizarrón, plumones, lápiz, papel, desaprovechando los dispositivos digitales como computadoras portátiles o asistentes digitales que llevan consigo los participantes. Con el middleware es posible desarrollar aplicaciones en las cuales se aprovechen las capacidades inalámbricas de cada dispositivo y sea posible la colaboración digital.

- Escenarios de entretenimiento.

Dada la proliferación actual de los dispositivos, es posible aprovechar el middleware para la creación de aplicaciones de entretenimiento las cuales ocurran en cualquier lugar, momento y entre distintos dispositivos.

- Escenarios para el control de aplicaciones remotas.

Aprovechando que en la actualidad diferentes dispositivos de cómputo están controlando diferentes servicios informáticos o industriales, sea ha vuelto indispensable para los administradores conocer el estado actual de un determinado proceso. Con el middleware es posible el desarrollo de aplicaciones en las cuales es posible controlar de manera remota una aplicación para que en determinado momento realice una acción, la cual, por ejemplo, podría ser que regresará el estado de un determinado servidor, o en el sector industrial, que regrese el estado de un proceso de manufactura directamente a un asistente digital.

- Escenarios de colaboración para zonas de desastre

En zonas de desastre generalmente no se cuenta con una infraestructura eléctrica o computacional, por ello, es indispensable contar con algún

medio que facilite el trazo de planes de acción. El middleware puede ser usado para desarrollar aplicaciones útiles en zonas hostiles sin infraestructura y en las cuales se requiera de servicios computacionales que agilicen la planeación de tareas, la asignación de recursos humanitarios, etc.

5.2. Aplicaciones desarrolladas con Lucrn

Para ilustrar como puede ser aplicado el middleware Lucrn se han desarrollado tres escenarios los cuales funcionan entre computadoras portátiles y asistentes digitales.

5.2.1. Un sistema de envío de mensajes

El primer escenario consiste en intercambiar mensajes de texto entre distintos dispositivos como computadoras portátiles o asistentes digitales. Este escenario es útil en reuniones donde se requiere intercambiar opiniones antes de la sesión de preguntas pero sin distraer al expositor. También puede ser utilizado en zonas industriales donde el ruido es excesivo y es necesario intercambiar mensajes entre un grupo de trabajadores para que realicen alguna tarea. Estos escenarios tienen ventajas sobre los tradicionales ya que los trabajadores tienen las indicaciones inmediatamente en formato digital y pueden ser consultadas posteriormente.

Para la implementación de este escenario se desarrollaron aplicaciones en el sistema Mac OS X y Palm OS. Estas hacen uso de la implementación de Lucrn para la definición y publicación de servicios, así como para la búsqueda de dispositivos, para establecer los enlaces inalámbricos, para la difusión de los dispositivos que componen la red, y para la transferencia de información entre todos ellos.

En la figura 5.1 se pueden observar las diferentes interfaces gráficas creadas en los diferentes sistemas operativos. Del lado izquierdo se puede observar la interfaz gráfica para el sistema Mac OS X y del lado derecho se observa la interfaz gráfica para el sistema Palm.

Intencionalmente se trató de que la interfaz gráfica fuera semejante entre los dispositivos. Ambas cuentan con los siguientes objetos gráficos:

- *Un botón Find and Connect*

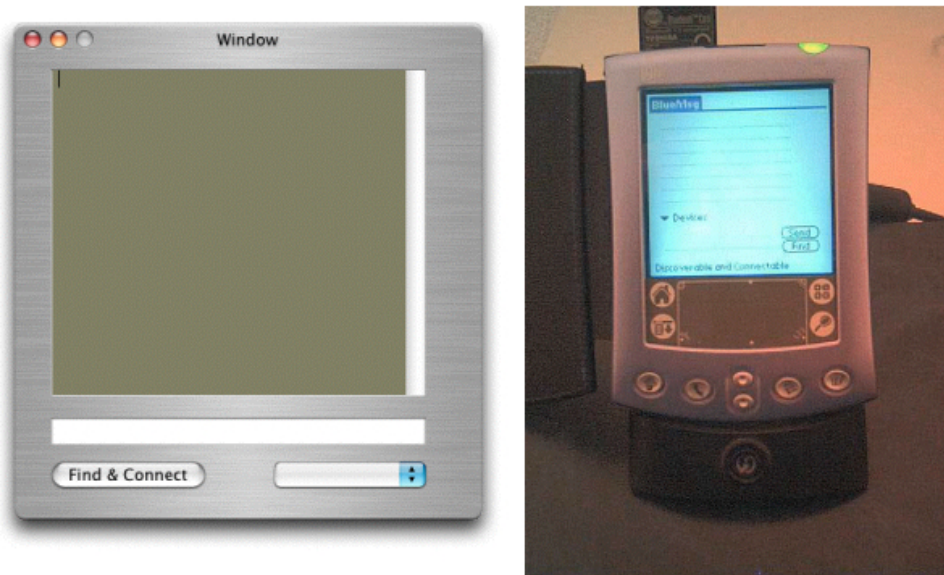


Figura 5.1: La interfaz de usuario del sistema de envío de mensajes en el sistema Mac OS X y Palm OS.

A través de este botón es posible que los usuarios puedan ver los dispositivos que tienen habilitado el servicio de envío de mensajes y conectarse a alguno de ellos. En el caso del asistente digital Palm por razones de espacio se omitió la palabra *connect*, pero la funcionalidad es la misma.

- *Un campo de texto para escribir el texto que se desea enviar*

Encima del botón *find and connect* en Mac OS X o a un lado del botón *find* en Palm OS se encuentra un campo de texto que sirve para escribir el texto que se desea enviar a los demás dispositivos.

- *Un menú pop-up el cual sirve para mostrar los dispositivos que están en la red*

Una vez que un dispositivo se ha conectado a un servicio de envío ofrecido por algún dispositivo, automáticamente se integra a la red de dispositivos y gracias al middleware Lucrn su presencia es enviada a todos los dispositivos que la conforman. En este caso se hace uso del menú pop-up para mostrar todos los dispositivos que actualmente están presentes y haciendo uso del servicio de envío de mensajes. En la figura 5.3 están señalados los menús *pop-up* en Mac OS X y Palm OS.

- *Un campo de texto en el cual se desplazan todos los mensajes que los usuarios han escrito*

El funcionamiento del sistema es como sigue: Los usuarios de las computadoras portátiles y de los asistentes digitales ejecutan la aplicación *BlueMsg*. Una vez que hacen esto aparecerán ventanas como las que se muestran en la figura 5.1. Luego, alguno de los usuarios debe presionar el botón de *find and connect*. En el caso de la Palm aparecerá una ventana como la que se muestra en la figura 5.2. En ella se puede observar que el dispositivo Palm, que en este caso corresponde a un asistente digital con el nombre de *Lucrn*, encontró a dos dispositivos: *atenea* (computadora portátil) y *apolo* (computadora iMac), ambos ofrecen el servicio de envío de mensajes. El usuario selecciona algún dispositivo y automáticamente se habilitan los campos de texto para que puedan comenzar a enviar mensajes. En la figura 5.3 se puede observar como se actualizan los menús *pop-up* para indicar los dispositivos que actualmente componen la red. Una vez establecido el enlace inalámbrico las computadoras pueden comenzar a intercambiar mensajes, como se muestra en la figura 5.4. Cabe destacar que es posible que el usuario de la portátil

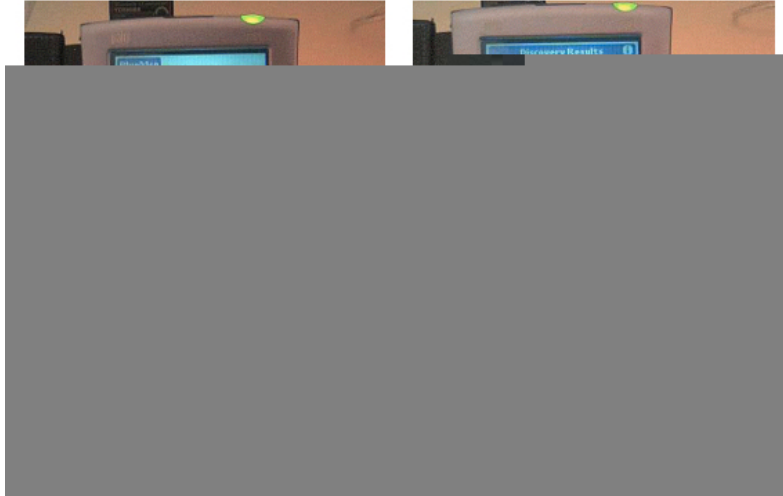


Figura 5.2: Buscando dispositivos que ofrezcan el servicio de envío mensajes en Palm OS.

se desplace dentro del recinto y ello no implica que se rompa el enlace de comunicación, a menos que se salga del rango de comunicación.

En la figura 5.5 se puede observar como los menús *pop-up* se actualizan cuando se ha integrado un nuevo dispositivo, en este caso se ha integrado la computadora *iMac* con el nombre de *apolo*.

Es importante resaltar que la primera versión de este sistema no se realizó utilizando la implementación de Lucrn. Sino que se utilizaron directamente los APIs de Bluetooth proporcionados por cada sistema operativo. El principal problema que se tuvo con esa implementación fue que sólo podía trabajar con dos dispositivos, la computadora portátil y el asistente digital. Eso se debió a la falta de una capa de red encima de la pila de protocolos de Bluetooth. Sin embargo, esta implementación fue relevante para la definición del middleware Lucrn ya que permitió conocer más a fondo los protocolos de Bluetooth y dominar los API's de los diferentes sistemas. Posteriormente se volvió a implementar esta aplicación pero ahora utilizando los API's del middleware Lucrn. Comparando ambas implementaciones la que utilizó el middleware Lucrn se implementó más rápido y puede funcionar con más de dos dispositivos.

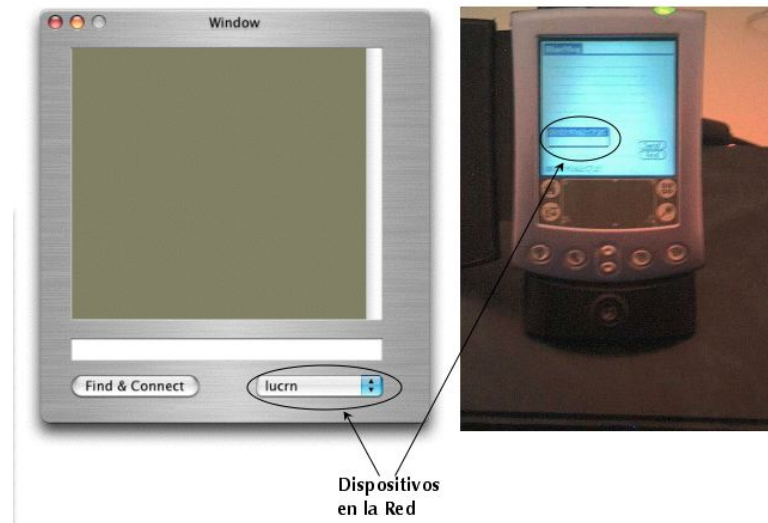


Figura 5.3: Los menús Pop-Up muestran los dispositivos que componen la red.

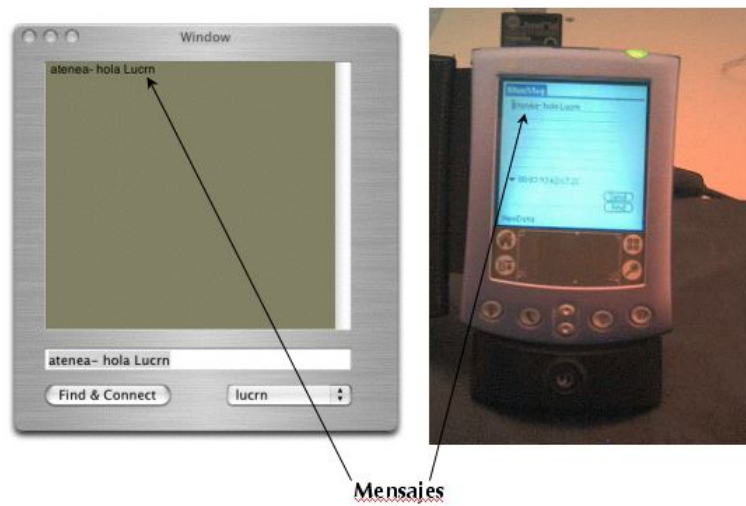


Figura 5.4: Intercambiando mensajes entre los diferentes dispositivos.

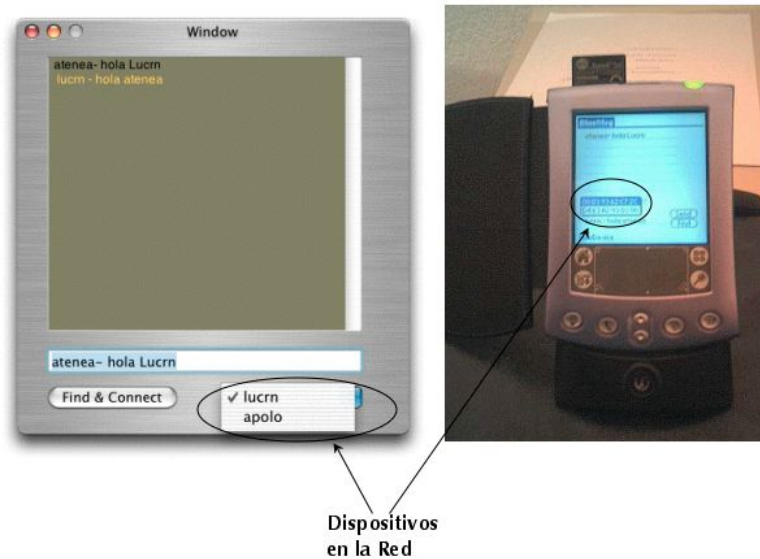


Figura 5.5: Integración de un nuevo dispositivo.

5.2.2. Un pizarrón electrónico compartido

El segundo escenario consiste en lograr que un conjunto de usuarios, haciendo uso de los dispositivos móviles que llevan consigo, colaboren a través de ellos para el desarrollo de un esquema compuesto por figuras geométricas simples como rectángulos y círculos. Explicándolo de otra manera se desea tener un pizarrón electrónico compartido que muestre a los usuarios una vista global de las acciones que cada uno de ellos realiza. Por ejemplo, una acción puede ser colocar un círculo en la posición (x,y) o borrar una determinada figura, éstas una vez realizadas van a mostrarse en las aplicaciones de los diferentes dispositivos.

Estos escenarios son de gran utilidad para un grupo de diseñadores los cuales generalmente tienen que acudir a la empresa del cliente para la recopilación de requerimientos y a partir de ellos definir un prototipo inicial. Cuando llegan a la empresa generalmente sucede que no hay la suficiente infraestructura para que conecten a sus dispositivos y a pesar de que el prototipo a diseñar sea sencillo tienen que regresar a su compañía para el desarrollo del mismo. Sin embargo, utilizando tecnología inalámbrica y desarrollando

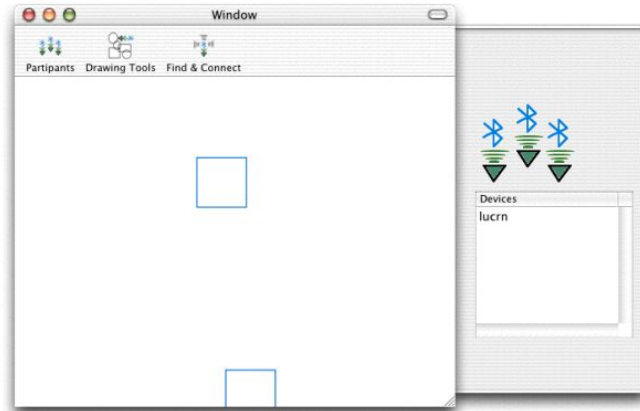


Figura 5.6: La interfaz de usuario del pizarrón compartido en Mac OS X.

aplicaciones con el middleware Lucrn es posible que ellos acudan con el cliente y a través de sus dispositivos colaboren en el diseño del prototipo y pueda ser presentado el mismo día.

Al igual que el escenario anterior, se desarrollaron aplicaciones en Mac OS X y Palm OS. En la figura 5.6 se puede observar la interfaz de usuario de las aplicaciones en Mac OS X. En este caso los dispositivos participantes son una computadora portátil con sistema operativo Mac OS X y un asistente digital con sistema operativo Palm OS.

En la interfaz de usuario podemos encontrar los siguientes objetos gráficos.

- *Un botón con la etiqueta de Find & Connect*

Al presionarse este botón el usuario puede buscar y conectarse a los dispositivos que estén ofreciendo el servicio de pizarrón compartido.

- *Un botón con la etiqueta de Participants*

Al presionarse este botón se abre un cajón del lado derecho de la ventana principal, el cual está compuesto por una tabla la cual desplegará los dispositivos que actualmente formen parte de la red.

- *Un botón con la etiqueta de Drawing Tools*

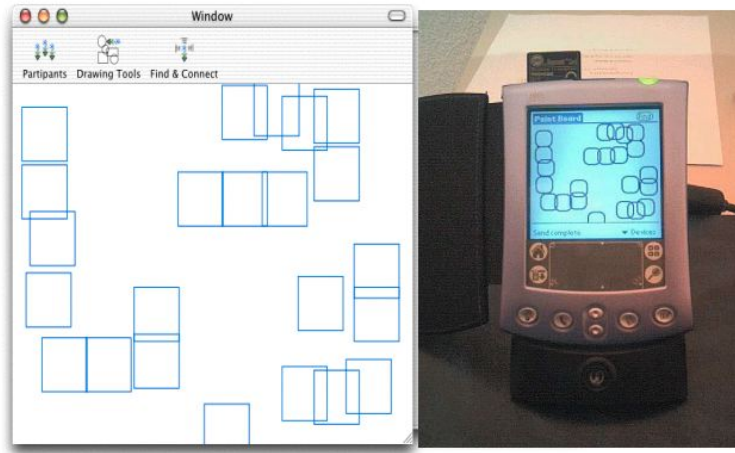


Figura 5.7: Intercambiando figuras geométricas entre dispositivos móviles.

Al presionarse este botón se abre un cajón del lado derecho de la ventana principal, este cajón tiene botones que permiten seleccionar las figuras geométricas que pueden ser dibujadas. En este caso son círculos o rectángulos. Tiene también un botón que permite borrar las figuras.

- *Un área de dibujo sobre el cual se desarrolla el esquema*

El funcionamiento de la aplicación es como sigue: los usuarios ejecutan la aplicación *whiteboard* en los diferentes dispositivos. Una vez que hacen esto aparecen las ventanas principales. Luego, los usuarios deben encontrar los dispositivos que se encuentran a su alrededor y que ofrecen el servicio de pizarrón electrónico. Para ello presionan el botón con la etiqueta de *Find & Connect* e inmediatamente aparece una ventana que muestra los dispositivos que ofrecen este servicio. Seleccionan alguno de ellos y automáticamente se habilita el área de dibujo de su aplicación y pueden comenzar a dibujar figuras geométricas. Conforme ellos van dibujando las acciones se van reflejando en las aplicaciones de los demás dispositivos como se muestra en la figura 5.7.

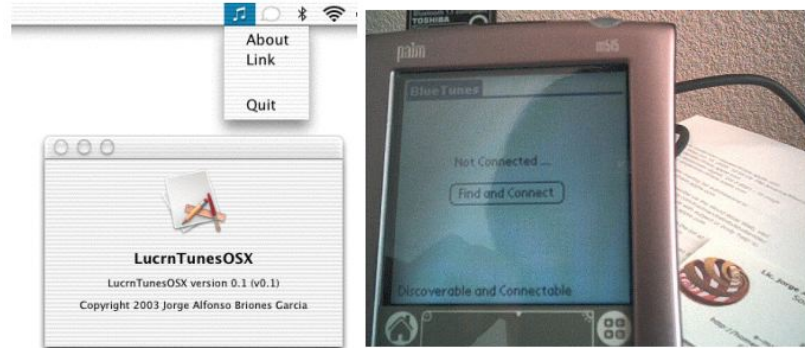


Figura 5.8: La interfaz de usuario del control remoto del iTunes en Palm.

5.2.3. Un control remoto para la aplicación iTunes

El tercer escenario consiste en que un asistente digital pueda controlar de forma remota alguna aplicación de alguna computadora portátil o computadora de escritorio. En este caso se escogió la aplicación *iTunes*, la cual reproduce música con formato MP3. Sin embargo, pudo haber sido cualquier otra como PowerPoint. Este escenario permite que el usuario esté sentado cómodamente desde su sofá y controle de manera remota el volumen de la aplicación o en forma aleatoria se reproduzca alguna canción. Una variante de este escenario sería aquel en el que se construye una aplicación en el asistente digital para obtener las últimas estadísticas de un servidor http.

El funcionamiento de la aplicación es como sigue: el usuario ejecuta la aplicación *BlueTunes* en su asistente digital (figura 5.8), luego presiona el botón *Find and connect*. Una vez hecho esto se presentará una ventana con todos los dispositivos que actualmente tienen el servicio disponible. Selecciona alguno de la lista, se establece la conexión con el dispositivo y puede comenzar a enviar algún comando que seleccionará del menú pop up que tiene en la parte superior del asistente digital (figura 5.9). Finalmente si escoge *Start* o *Random* se escogerá alguna canción de manera aleatoria de la lista actual del dispositivo y se comenzará a reproducir. En el sistema remoto (Mac OS X) se puede observar un menú que muestra de color gris el dispositivo que está conectado (figura 5.9). De igual forma se muestra la aplicación iTunes tocando la canción *My immortal*.

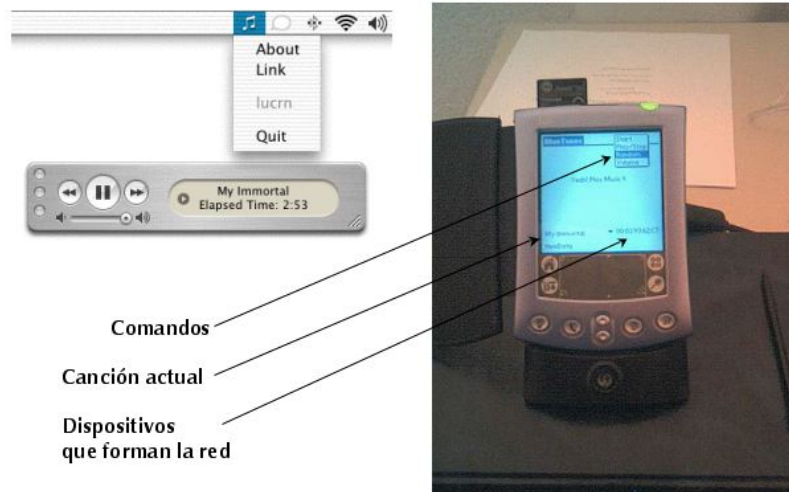


Figura 5.9: Buscando un dispositivo.

5.3. Resultados

Tomando en cuenta los escenarios construidos con el middleware Lucrn, se tienen los siguientes resultados:

- Se diseñó el middleware Lucrn que permite la interoperación transparente entre aplicaciones distribuidas en dispositivos móviles.
- Se implementó el middleware propuesto en asistentes digitales Palm y computadoras portátiles con sistema operativo Mac OS X.
- Las aplicaciones de los diferentes dispositivos móviles no requieren de una configuración especial por parte de los usuarios para poder integrarse a la red de dispositivos.
- Los escenarios desarrollados con Lucrn no dependen de la existencia de alguna infraestructura física. Pero si dependen de que cada dispositivo tenga la implementación del middleware, un adaptador Bluetooth y una batería.

- Las aplicaciones desarrolladas con Lucrn en los diferentes dispositivos móviles se comportan como servidores y clientes. Evitando que la salida de alguno de ellos implique la finalización del escenario de colaboración.
- Las aplicaciones desarrolladas con Lucrn aprovechan los mecanismos proporcionados por Bluetooth y los extienden para que se pueda trabajar con más de dos dispositivos.
- El middleware propuesto sirve para implementar una variedad de escenarios. Aún aquellos que no son escenarios de colaboración. Un ejemplo es el control remoto del iTunes.
- Las implementaciones en los diferentes sistemas operativos son flexibles, ambas proporcionan APIs que facilitan la programación de aplicaciones colaborativas.
- La implementación en Mac OS X es orientada a objetos y por capas. Las conexiones entre las diferentes capas se hacen a través de protocolos. Este diseño permite que los componentes de cada capa sean intercambiables.

5.4. Resumen

En este capítulo se presentaron las aplicaciones que se han desarrollado utilizando los APIs del middleware Lucrn en Mac OS X y Palm OS. El middleware permite el desarrollo de aplicaciones que interactúen en redes ad-hoc de dispositivos móviles equipados con tecnología inalámbrica Bluetooth. El middleware extiende la pila de protocolos de Bluetooth y agrega mecanismos para que los diferentes dispositivos que se encuentran en el rango de proximidad puedan formar una red y enviar mensajes entre todos los dispositivos que la conforman. La puesta en marcha de esta red es sencilla y no requiere que el usuario realice alguna configuración facilitando con esto el intercambio de información digital. También se presentaron los resultados que se han obtenido con este middleware y el principal es el desarrollo de aplicaciones colaborativas distribuidas en dispositivos móviles heterogéneos sobre ambientes inalámbricos.

En el siguiente capítulo se presentan las conclusiones y el trabajo a futuro que se va a realizar a partir de este proyecto.

Capítulo 6

Conclusiones

Las conclusiones que se tienen al finalizar este trabajo son las siguientes:

- *Los objetivos y metas planteadas fueron alcanzadas y se cuenta con un middleware que extiende la pila de protocolos de Bluetooth y hace posible el desarrollo de escenarios de colaboración digitales en dispositivos móviles.*
- *El middleware añade protocolos de anuncio de dispositivos, ruteo, movilidad y colaboración los cuales son ligeros y pueden ser implementados en diferentes plataformas y dispositivos móviles.*
- *El middleware utiliza los mecanismos de descubrimiento de dispositivos y servicios que ofrece la tecnología Bluetooth para lograr la formación de una red sin infraestructura entre varios dispositivos.*
- *Los protocolos de cada capa no inhiben los nuevos mecanismos incorporados por Bluetooth sino que los utilizan y los especializan.*
- *El protocolo de colaboración permite la interoperación entre aplicaciones distribuidas en dispositivos heterógeneos. Está orientado al envío de acciones y ha mostrado ser un modelo flexible y poderoso.*
- *La implementación del middleware Lucrn en los sistemas operativos Mac OS X y Palm OS, cubre un gran rango de dispositivos, desde computadoras de escritorio hasta asistentes digitales.*
- *Las APIs de las diferentes implementaciones han mostrado ser funcionales y facilitan el desarrollo de aplicaciones colaborativas.*

Cada nueva tecnología es una oportunidad para definir nuevas aplicaciones, nuevos escenarios y nuevas formas de uso de la misma. En particular las tecnologías de comunicación han influido bastante en la manera en como las personas se comunican, en como se realizan los negocios y en las formas de entretenimiento, el mayor exponente de esto es Internet. En términos de investigación, una nueva tecnología define nuevos modelos, por ejemplo ha definido modelos para el cómputo paralelo y distribuido. Con la introducción de tecnología de comunicación inalámbrica surgió la computación móvil.

En especial durante el diseño y la implementación de este middleware se pudo notar que no hay una distinción entre los diferentes nodos de la red. Es decir, los nodos ahora no se pueden distinguir en clientes y servidores, ya que todos ahora van a ser clientes y servidores en un instante dado. Esto es un cambio fundamental y va a influir significativamente en el software que se desarrolle para estos ambientes. Ahora las aplicaciones tiene que lidiar con canales de entrada y salida como se explicó en la parte de implementación del middleware. Afortunadamente middleware Lucrn comienzan a clarificar este concepto el cual posteriormente todos seguirán. Es importante hacer notar que las entidades que nos rodean son distinguibles por los servicios que proporcionan, por tanto para que un dispositivo digital sea distinguible a los demás éste debe proporcionar un servicio el cual pueda ser descubierto y pueda ser usado fácilmente. Debido a esto el middleware se apoya sobre los protocolos de descubrimiento de servicios proporcionados por Bluetooth.

Con este trabajo se distinguen características esenciales para la puesta en marcha de escenarios de colaboración digitales:

- *El anunció y descubrimiento de dispositivos y servicios.*
- *Los nodos son indistinguibles (no se clasifican como servidores y clientes).*
- *Los mecanismos de colaboración.*

El desarrollo de este proyecto fue laborioso pero muy motivante. En retrospectiva, en los primeros meses se hizo una revisión exhaustiva de la tecnología Bluetooth, se buscaron artículos en las principales revistas de investigación, se compraron libros y se entendió esta tecnología. Dado que el medio de comunicación es inalámbrico también se hizo una revisión de radio frecuencia.

Posteriormente se evaluaron las diferentes plataformas de hardware y software, de estos análisis se compró el hardware y se definieron los sistemas operativos que iban a ser utilizados. Luego se estudiaron y se desarrollaron aplicaciones utilizando los diferentes APIs hasta entenderlos y dominarlos.

Después, una vez que se entendió y se conoció como se podía extender la pila de protocolos de Bluetooth, se diseñó el middleware Lucrn. Esta etapa fue la más ardua, llevó el estudio del diseño de protocolos, la comprensión de las diferentes arquitecturas de red y a situar perfectamente a Bluetooth conociendo sus limitaciones y las capas que necesitaba.

Posteriormente se implementó este middleware en dos sistemas operativos con características muy diferentes y en dispositivos móviles con recursos de almacenamiento y de procesamiento distintos. El objetivo de la implementación siempre fue el proporcionar un API del middleware Lucrn y esto se logró en gran medida. En esta etapa también se refinaron los protocolos.

Finalmente con los APIs que surgen de las implementaciones se desarrollaron tres aplicaciones las cuales fueron presentadas en el capítulo anterior. Estas aplicaciones sirvieron para evaluar al middleware Lucrn y para notar el poder y la flexibilidad de la implementación en los diferentes sistemas operativos.

6.1. Perspectivas a futuro

Sin lugar a dudas fue un proyecto muy apasionante para el cual se han vislumbrado perspectivas a futuro como las siguientes:

- *Validación de los protocolos.*
- *Extender el protocolo de movilidad para que trate el problema de la salida de un nodo complejo.*
- *Implementación del middleware en otros dispositivos móviles como iPAQ con sistema Linux y teléfonos celulares como el t68i.*
- *Implementación de diferentes aplicaciones colaborativas, por ejemplo, aplicaciones de video y audio.*
- *Implementación del middleware en otras tecnologías inalámbricas como 802.11b.*

En el futuro se percibe claramente un incremento en el uso de dispositivos móviles ligeros, esto gracias a los avances en la tecnología de integración a gran escala y la microelectrónica. Muchos de ellos integrarán tecnología Bluetooth y será posible el intercambio de información entre varios dispositivos. Se espera que este middleware sea el inicio para definir nuevos escenarios y ya no tan solo sea la mejora de los existentes. En términos de investigación del middleware todavía tiene mucho que ofrecer y las implementaciones pueden ser extendidas y modificadas fácilmente.

En particular, este trabajo pretende ser un catalizador para que se desarrolle investigación en cómputo móvil en México ya que es un área que muy poco se ha abordado y tratado. Sin embargo, por las distintas estancias del autor en el extranjero se sabe que es un tema activo de investigación. Sobre todo por los nuevos problemas que introduce y por las ventajas que puede ofrecer.

Respecto a esto en el mes de septiembre de 2003 asistí a la *2a escuela Franco-Mexicana sobre los sistemas distribuidos cooperativos* que se celebró en Rennes Francia en las instalaciones del centro de investigación IRISA [20], la asistencia a este evento fue posible gracias a una beca otorgada por el comité organizador y al apoyo proporcionado por el Departamento de Aplicación de Microcomputadoras de la Universidad Autónoma de Puebla. En esa escuela los investigadores dieron pláticas de los proyectos que ellos están desarrollando y temas de investigación actuales, de esas pláticas destacan las de *Redes inalámbricas* y *Redes Ad-hoc*, sobretodo porque abarcan temas fundamentales de este proyecto de tesis que se inicio en septiembre de 2002.

En el mes de junio del año 2003 asistí a la conferencia mundial de desarrolladores [21] que cada año realiza la compañía *Apple Computer*, la asistencia a este evento fue posible gracias a una beca otorgada por la citada compañía, en esa conferencia se otorgó el premio de mejor producto y mejor producto inovativo del año 2003 a una aplicación que permite controlar diversas aplicaciones de Mac OS X desde un teléfono celular o un asistente digital Palm OS utilizando enlaces inalámbricos Bluetooth [22]. Como se mostró en el capítulo de resultados ese tipo de aplicaciones pueden ser desarrolladas fácilmente con el middleware que se ha diseñado. El middleware ofrece mecanismos para desarrollar aplicaciones mucho más complejas que esas, como aplicaciones colaborativas, algo que no ofrece el producto citado anteriormente.

Apéndice A

La pila de protocolos en Mac OS X y Palm OS

En este apéndice se describen brevemente las implementaciones de la pila de protocolos de Bluetooth realizadas para Mac OS X y Palm OS. Se muestran figuras que muestran los protocolos que implementan cada uno de ellos y se explican algunas de las clases y bibliotecas que proporcionan para la construcción de programas que usen a Bluetooth como medio de comunicación.

A.1. La pila de protocolos en Mac OS X

La pila de protocolos de Bluetooth fue implementada por Apple en el año de 2002 y fue liberada con la versión 10.2 del sistema Mac OS X. La implementación está basada en la especificación 1.0 liberada por el grupo de interés de Bluetooth. Ésta incluye los protocolos de la *baseband*, *L2CAP*, *RFCOMM* y *SDP* como se muestra en la figura A.1.

Para desarrollar aplicaciones con Bluetooth en Mac OS X se proporcionan dos frameworks:

- *IOBluetooth.framework*
- *IOBluetoothUI.framework*

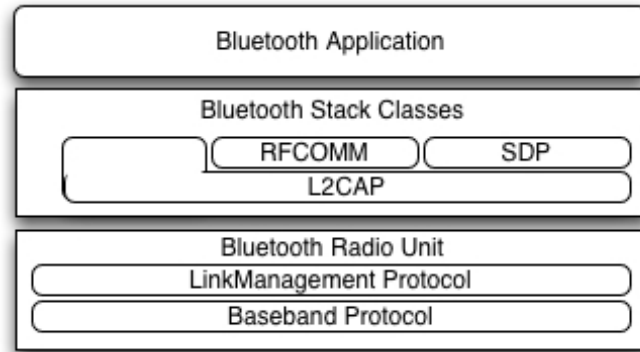


Figura A.1: Pila de Protocolos de Bluetooth en Mac OS X.

A.1.1. IOBluetooth.framework

El framework IOBluetooth contiene un conjunto de clases y bibliotecas de funciones que permiten tener acceso a los protocolos de Bluetooth implementados por Mac OS X.

A continuación se explican algunas de las clases más importantes:

- *IOBluetoothDevice*

Usando un objeto de la clase IOBluetoothDevice se puede solicitar la creación o destrucción de enlaces inalámbricos del protocolo baseband. Además se puede solicitar la creación de un canal L2CAP o RFCOMM hacia un dispositivo remoto.

- *IOBluetoothL2CAPChannel*

La clase IOBluetoothL2CAPChannel representa un canal de comunicación L2CAP. No se pueden crear objetos de esta clase directamente, se tiene que hacer indirectamente a través de la clase IOBluetoothDevice.

- *IOBluetoothRFCOMMChannel*

La clase IOBluetoothRFCOMMChannel representa a un canal RFCOMM.

- *IOBluetoothUserLib*

IOBluetoothUserLib es una biblioteca escrita en lenguaje C que contiene un conjunto de funciones para abrir canales L2CAP o RFCOMM. También proporciona funciones para la definición de un servicio Bluetooth.

A.1.2. IOBluetoothUI.framework

El framework IOBluetoothUI proporciona un conjunto de objetos gráficos que pueden ser integrados a las aplicaciones para establecer conexiones con otros dispositivos.

El framework está compuesto principalmente por las siguientes clases y bibliotecas de funciones:

- *IOBluetoothPairingController*

La clase representa a una ventana desde la cual se puede iniciar el proceso de *pairing* con otro dispositivo. El proceso de *pairing* permite a dos dispositivos definir una contraseña para cifrar el contenido de la información que se enviará entre ellos.

- *IOBluetoothServiceBrowserController*

La clase representa a una ventana desde la cual se pueden desplegar y establecer conexiones con otros dispositivos.

- *IOBluetoothUIUserLib*

Esta es una biblioteca escrita en lenguaje C que consiste de un conjunto de funciones que auxilian en la selección de algún dispositivo o servicio. Además contiene las funciones para iniciar el proceso de *pairing*

Cabe resaltar que además de los frameworks el sistema proporciona herramientas como:

- *Bluetooth Monitor*

Es una aplicación que muestra las propiedades de los enlaces inalámbricos que se tengan con otros dispositivos Bluetooth.

- *Bluetooth Packet Decoder*

Es una aplicación que muestra los paquetes que se envían o reciben a través del dispositivo Bluetooth.

A.2. La pila de protocolos de Bluetooth en Palm OS

Un sistema operativo que desde el inicio mostró gran interés en Bluetooth fue Palm OS. En la actualidad, la compañía Palm ha implementado la pila de protocolos de Bluetooth y está disponible para la mayoría de sus dispositivos móviles. Gracias a esto, ahora se pueden establecer enlaces inalámbricos Bluetooth con los diferentes asistentes digitales que ellos ofrecen.

Su implementación ha ido más allá de sólo proporcionar una librería para la creación, manejo y uso de enlaces inalámbricos Bluetooth. Ellos han integrado Bluetooth con varias de las tecnologías que son usadas para transmitir información desde el asistente digital. Logrando que ciertos códigos, que por ejemplo transmiten información a través de infrarojo, no tengan que ser modificados para que hagan uso de Bluetooth.

En la figura A.2 se puede observar como está conformado la implementación de Bluetooth en Palm OS.

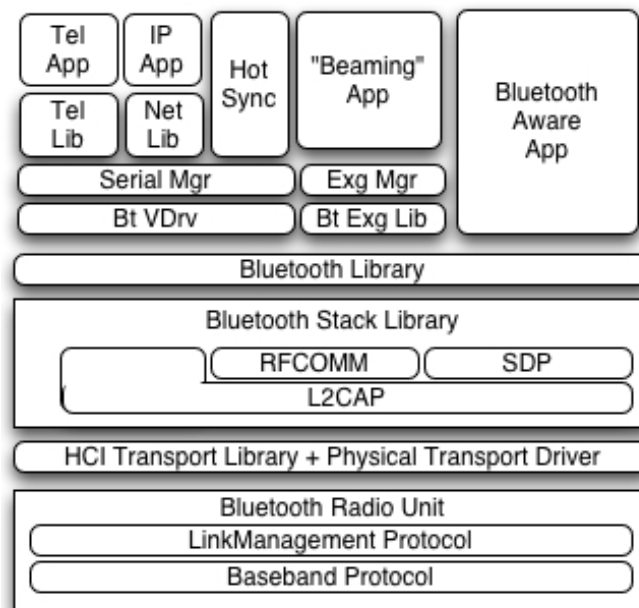


Figura A.2: Componentes de la implementación de Bluetooth en Palm.

A continuación explicamos la función de cada uno de los componentes.

■ *Bluetooth Library*

Es una biblioteca compartida (shared library) que proporciona un API para que puedan desarrollarse aplicaciones que hagan uso de Bluetooth. El API contiene funciones que permiten realizar las siguientes tareas:

- *Manejo de dispositivos, piconets y enlaces inalámbricos*
- *Comunicación con otro dispositivos a través de L2CAP y RFCOMM*
- *Consulta de servicios a través de SDP*

En la figura A.2 se puede observar que la *Bluetooth Library* es empleada por componentes y librerías que se encuentran una capa más arriba para enviar información a través de Bluetooth.

■ *Bluetooth Virtual Serial Driver*

Permite que las aplicaciones hagan uso del Palm OS Serial Manager. Utiliza el protocolo RFCOMM de Bluetooth para definir un enlace serial. En la figura A.2 el *Bluetooth Virtual Serial Driver* lo encontramos encima de la biblioteca Bluetooth y se puede identificar con la leyenda *Bt VDrv*.

■ *Bluetooth Exchange Library*

Permite que las aplicaciones hagan uso del Palm OS Exchange Manager tomando a Bluetooth como el medio de comunicación. En la figura A.2 lo podemos encontrar encima de la biblioteca de Bluetooth con la leyenda *Bt Exg Lib*.

■ *Bluetooth stack*

Es una biblioteca compartida que implementa varios de los protocolos de la especificación de Bluetooth. Los desarrolladores no tienen acceso a ella.

■ *Bluetooth Transports*

Es una biblioteca compartida que actúa como un manejador (device drivers) para varios dispositivos. Los desarrolladores no tienen acceso a ella.

- *Bluetooth Extensions*

La *Bluetooth Extensions* coordina las múltiples bibliotecas que componen la implementación de Bluetooth en Palm. Los desarrolladores no tienen acceso a ella.

La implementación del middleware Lucrn se realizó empleando las funciones de la *Bluetooth Library*. Son bastantes las funciones proporcionadas para el manejo de la piconet y el descubrimiento de servicios. Por su importancia se explican algunas de las funciones que son empleadas para enviar información a través de canales L2CAP.

- *BtLibSocketCreate*

Permite crear un socket L2CAP.

- *BtLibSocketListen*

Configura el socket para que pueda escuchar peticiones.

- *BtLibSDPServiceRecordCreate*

Inicializa un registro SDP para que represente a un socket que puede responder a peticiones.

- *BtLibSdpGetPSMByUUID*

Permite obtener el PSM (protocolo service multiplexer) de un canal L2CAP.

- *BtLibSocketConnect*

Permite crear un socket para conectarse a otro definido en un dispositivo remoto.

- *BtLibSocketSend*

Permite enviar información a través de un socket.

- *BtLibSocketClose*

Cierra un socket.

Bibliografía

- [1] Palm. Top ten reasons to choose a palm powered handheld. *Palm Source Documentation*, 2002.
- [2] Frazer Bennett and David Clarke. Piconet: Embedded mobile networking. *IEEE Personal Communications*, October 1997.
- [3] R.Bruno, M.Conti, and E. Gregori. Wlan technologies for mobile ad-hoc networks. *Proceedings of the 34th Hawaii International Conference on Systems Sciences.*, 2001.
- [4] Neil Rhodes and Julie McKeehan. *Palm OS Programming. The Developer's Guide, 2nd Edition.* O'REILLY, 2002.
- [5] M. Satyanarayanan. Fundamental challenges in mobile computing. *Proc. 15th ACM Symp. Principles of Dist. Comp.*, Philadelphia, PA, May 1996.
- [6] L.Feeney, B. Ahlgren, and A. Westerlund. Spontaneous networking: An application-oriented approach to ad hoc networking. *IEEE Comm.Magazine*, vol 39, no 6, June 2001.
- [7] M. Weiser. The computer for the 21st century. *Scientific American*, September 1991.
- [8] Ramiro Jordan and Chaouki T. Abdallah. Wireless communications and networking: An overview. *IEEE Antenna's and Propagation Magazine*, vol 44, no 1, February 2002.
- [9] Brent A. Miller and Chatschik Bisdikian. *Bluetooth Revealed. The insider's guide to an open specification for Global Wireless Communication.* Prentice Hall PTR, Upper Saddle River, NJ 07458, 2001.

- [10] Bluetooth. Specification of the bluetooth system. core. http://www.bluetooth.com/pdf/Bluetooth_11_Specifications_Book.pdf, February Version 1.1, 2001.
- [11] Bluetooth. Specification of the bluetooth system. profiles. http://www.bluetooth.com/pdf/Bluetooth_11_Profile_Book.pdf, February Version 1.1, 2001.
- [12] Chatschik Bisdikian. An overview of the bluetooth wireless technology. *IEEE Comm.Magazine*, December 2001.
- [13] Jaap C. Haartsen. The bluetooth radio system. *IEEE Personal Communications*, February 2000.
- [14] Wei-Shin Wang. Bluetooth: A new era of connectivity. *IEEE microwave magazine*, September 2002.
- [15] K.V.S.S.S Sairam, N. Gunasekaran, and S. Rama Reddy. Bluetooth in wireless communication. *IEEE communications magazine*, June 2002.
- [16] Golden G. Richard III. Service advertisement and discovery: Enabling universal device cooperation. *IEEE Internet Computing*, September-October 2000.
- [17] Dimitri Bertsekas and Robert Gallager. *Data Networks*. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1992.
- [18] Gerard J. Holzmann. *Design and Validation of computer protocols*. Prentice Hall, 1991.
- [19] Apple. The objective-c programming language. *Apple Developer Documentation*, September 2003.
- [20] IRISA. 2a escuela de sistemas distribuidos cooperativos. <http://www.irisa.fr/manifestations/2003/ESRC/>.
- [21] Apple. Worldwide developer conference. <http://developer.apple.com/wwdc/index.html/>.
- [22] Clicker. Salling cliker v.2.1. <http://homepage.mac.com/jonassalling/Shareware/Clicker/>.

Los abajo firmantes, integrantes del jurado para el examen de grado que sustenta el **Lic. Jorge Alfonso Briones García**, declaramos que hemos revisado la tesis titulada:

Lucrn: Un middleware para el desarrollo de aplicaciones en redes espontáneas de dispositivos móviles Bluetooth

Y consideramos que cumple con los requisitos para obtener el Grado de Maestría en Ciencias en la especialidad de Ingeniería Eléctrica opción Computación.

Atentamente.

Dr. José Oscar Olmedo Aguirre. _____

Dr. Sergio V. Chapa Vergara. _____

Dr. Adriano de Luca Pennacchia. _____