PERFILES PARA DESARROLLO DE APLICACIONES TELEMÁTICAS

EMPLEADO BLUETOOTH

MARÍA VICTORIA ACOSTA ACOSTA
GUIDO ALEJANDRO GAVILANES CASTILLO

Monografía para optar al título de
Ingeniero en Electrónica y de Telecomunicaciones

Director
Javier Alexander Hurtado Guaca
Ingeniero en Electrónica y de Telecomunicaciones

FACULTAD DE INGENIERIA ELECTRONICA Y TELECOMUNICACIONES

DEPARTAMENTO DE CONMUTACION

POPAYAN

2002

CAPÍTULO 1 INTRODUCCIÓN

Las tecnologías inalámbricas y los servicios móviles se han fortalecido en los últimos años y son uno de los campos de mayor crecimiento del sector de las telecomunicaciones. Esto debido fundamentalmente a que la movilidad se ha convertido en un aspecto clave de la sociedad actual. Vivimos en una época de constante cambio donde aparecen nuevas formas de vida, las cuales a su vez generan nuevas necesidades. Las personas desean mantenerse en contacto con su familia, sus amigos y la sociedad, y tener acceso a la información en todo momento sin importar el lugar en donde se encuentren.

El desarrollo de un mundo inalámbrico cambiará fundamentalmente la forma en que los seres humanos consumen, interactúan y controlan sus vidas, abriendo paso a una serie de nuevas sensaciones, experiencias y posibilidades de desarrollo de la sociedad. Tareas tan cotidianas como el pago de una factura o la compra de un artículo en el centro comercial serán concebidas de una forma muy diferente a la que tienen hoy en día. La revolución de la información inalámbrica trae consigo un beneficio potencial para las personas y da cabida a la creación de nuevos servicios y productos que antes no hubieran podido imaginarse.

Ya que las personas se consideran como el eje central de esta nueva oleada tecnológica, es importante tener en cuenta que solo mediante la satisfacción plena de las necesidades y expectativas del usuario, será posible que los nuevos sistemas y servicios inalámbricos alcancen el éxito pronosticado. Estudios recientes pretenden descubrir qué es lo que representa, de manera definitiva, valor para los consumidores del mundo inalámbrico.

Existen cuatro aspectos relevantes que los suscriptores de los servicios inalámbricos desean obtener:

Conveniencia: Ya que el usuario podrá acceder a su información en cualquier momento y en cualquier lugar (mezclando los conceptos de movilidad e interfaz de usuario amistosa). Los dispositivos móviles actuales pueden catalogarse como aquellos que son muy convenientes y aquellos que no lo son de ninguna manera. Es verdad, que cuando una persona viaja, puede almacenar en su dispositivo móvil una gran cantidad de información

importante pero también es cierto que el usuario considerará algunos aspectos (factores de forma) del aparato como son el peso, el tamaño, la calidad de la pantalla, el teclado del dispositivo y las capacidades en cuanto a consumo de energía, cuando debe decidir si lleva consigo el dispositivo o no.

Sin embargo y dada la característica de ubicuidad de los dispositivos móviles, muchas personas los prefieren por encima de cualquier otra herramientas de cómputo de hoy en día. La información en tiempo real, la comunicación instantánea y un número creciente de oportunidades en nuevas transacciones, independientes todas estas de la situación del usuario, son factores que estimulan el incremento de dispositivos móviles en todo el mundo.

Los fabricantes de dispositivos han entendido que la conveniencia física es muy importante para el usuario, así que han dedicado gran parte de sus esfuerzos a fortalecer el nivel de amistad entre un dispositivo y su dueño, lo cual repercute directamente en el siguiente aspecto esperado por los suscriptores de servicios móviles.

Personalización: Los consumidores esperan que se les brinden servicios independientes basados en necesidades particulares del individuo así como en gustos personales y su posición geográfica. En un ambiente móvil, las limitaciones físicas del dispositivo obligan a construir y automatizar los procesos que se llevan a cabo normalmente en un ambiente fijo.

Los clientes querrán diseñar y construir sus propios procesos, en lugar de tener que aceptar que se los impongan; mas aún cuando son ellos quienes realizan sus propias transacciones desde sus dispositivos personales. Las nuevas tecnologías inalámbricas deben ser escalables e intuitivas.

Relación Costo – Beneficio: Los costos de la comunicación inalámbrica son todavía muy altos para los consumidores. Y si el lema de la tecnología inalámbrica es *“Estar siempre conectado”* debe remediarse este inconveniente cuanto antes. Una de las formas en que se ha dado solución a parte de este problema es la reducción de los costos de los dispositivos móviles en los últimos años (teléfonos fundamentalmente), pero aún queda por resolver lo relacionado con el costo de las llamadas y el valor de los servicios a los que podría tener acceso el usuario de un sistema inalámbrico.

Está claro, sin embargo, que la mayoría de las aplicaciones eficaces y por las cuales un consumidor está dispuesto a pagar son aquellas donde la información está sumamente

ligada a limitaciones de tiempo y donde el beneficio que recibe el consumidor por la obtención de información oportuna y precisa está por encima del costo que representa el tiempo al aire.

Seguridad: Existe acuerdo general en que la seguridad de la red móvil debe ser superior a la que presenta un ambiente fijo. Numerosas formas de asegurar la confiabilidad y fiabilidad de los datos cuando viajan por la red inalámbrica, así como poderosos mecanismos de encriptación y cifrado de datos, autenticación y autorización de usuarios, han surgido como respuesta a esta necesidad. Sin embargo, aun no puede asegurarse que la seguridad sea total y mientras esta no pueda certificarse a los usuarios, no podrán realizarse transacciones que involucren altas sumas de dinero o se vea comprometida la tranquilidad del cliente.

Además de los requerimientos mencionados en los párrafos inmediatamente anteriores, cobra gran importancia el fin para el cual se construyen y diseñan nuevas tecnologías: *Las aplicaciones*. En el caso de los sistemas inalámbricos, las aplicaciones son ilimitadas y muchas de ellas sorprendentes. A grandes rasgos podrían catalogarse de acuerdo a la siguiente clasificación:

Segmento de Consumo

Aplicaciones y Servicios de Entretenimiento: Destinados a la diversión y ocio de los usuarios. Entre ellos pueden citarse los juegos de vídeo, las aplicaciones multimedia, audio (MP3) y vídeo (Transmisión de imágenes, fotografía)

Aplicaciones y Servicios de Comunicación: Aquellos que están catalogados como Mensajería (SMS, e-mail), Notificación (Alertas, promociones, publicidad) y Servicios de Emergencia (Rescates y seguridad).

Aplicaciones y Servicios Transaccionales: Entre los cuales se encuentran los servicios de Pago móvil, Ventas (boletos, acciones, productos) y Finanzas (Transacciones Bancarias, Bolsa de valores).

Aplicaciones y Servicios de Información: Aquellas que proporcionan servicios de contenido. Contenido dinámico (noticias, clima) y contenido de Referencia o de consulta (catálogos, libros electrónicos, diccionarios)

Segmento de Negocios

Aplicaciones y Servicios de Comunicación: Relacionados con soporte a Fuerza de Ventas (calendario, e-mail, gestión de personas)

Aplicaciones y Servicios Transaccionales: Ventas y servicios Móviles

Aplicaciones y Servicios de Información: Por ejemplo Gestión de flotas y rastreo.

Pero, ¿cómo se soporta la *Movilidad* actualmente? Hoy en día, diferentes tipos de sistemas móviles prestan sus servicios en las redes de telecomunicaciones. Sin embargo estos sistemas no son compatibles en todo el mundo.

GSM (Sistema Global para Telecomunicaciones Móviles) es el sistema empleado en Europa Junto con TDMA y 3G (en el último año), adicionalmente se encuentran versiones de GSM 900 y GSM 1800.

Por otra parte, en América, los sistemas móviles se basan aun en sistemas analógicos, TDMA, CDMAOne y PCS.

En Japón y Asia del Pacífico, se encuentran sistemas Analógicos TMDA, cdmaOne, GSM, y 3G. El resto del mercado mundial emplea sistemas analógicos, TDMA, cdmaOne, GSM.

Los mercados regionales (America del Norte y Europa) para suscriptores, terminales y transmisores/receptores GSM están divididos además en tecnologías de 2.5G dentro de las que se incluyen GPRS, HSCSD y el estándar EDGE.

Los mercados regionales para suscriptores, terminales y transmisores/receptores 3G incluyen tecnologías como W-CDMA, cdma2000 1XRTT, cdma2000 3XRTT and TDMA-EDGE.

Todas los sistemas mencionados hacen parte de las tecnologías de radio de largo alcance, las cuales se apoyan en una amplia infraestructura de red, la cobertura está dada por estaciones base, los recursos de radio se gestionan desde una ubicación central, y los servicios están integrados en el sistema.

Sin embargo existen otros sistemas de comunicación que constituyen las llamadas tecnologías inalámbricas de corto alcance. Estas redes tienen la función principal de proporcionar conectividad y acceso a las tradicionales redes cableadas, como si se tratara de una extensión de éstas últimas, pero con la flexibilidad y movilidad que ofrecen las comunicaciones inalámbricas. Dentro de este campo pueden encontrarse las Redes de Área Local Inalámbricas(Wireless Local Área Network - WLAN) con los estándares 802.11a y 802.11b, HiperLAN, las Redes de Área Personal (Personal Área Network - PAN), las redes de comunicación infrarroja IrDA, el estándar HomeRF entre otras. La integración de estas tecnologías con las redes móviles de largo alcance permiten la consecución de entornos de comunicación totalmente ubicuos.

Las tecnologías inalámbricas e corto alcance permiten que numerosos dispositivos puedan conectarse de manera espontánea para intercambiar información y dar soporte a una nueva gama de servicios y aplicaciones. Las redes PAN son un claro ejemplo del alcance de la tecnología. Estas redes se extienden por completo al dominio del usuario. La relación del hombre con las máquinas se ha modificado sustancialmente con el paso del tiempo. Nuevos aparatos surgen y numerosas modificaciones se aplican a los ya existentes, con el fin de que puedan responder cada vez mejor a una función determinada.

No obstante, y pese a la multitud de opciones que se pueden encontrar en el mercado de la telecomunicaciones es indispensable que los proveedores de servicios, los operadores de redes, los desarrolladores de aplicaciones y todos los que hacen parte de la revolución de las comunicaciones, comprendan la necesidad de la convergencia para todos los sistemas de comunicaciones. A través del trabajo conjunto y la elaboración de normas de estandarización, todos estos actores puede asegurar que diferentes tipos de redes trabajarán conjuntamente y en armonía dentro de una misma área.

Uno de los esfuerzos recientes y que ha alcanzado gran acogida por diversos sectores de las telecomunicaciones es Bluetooth. Los líderes de las telecomunicaciones y la industria se unieron para crear Bluetooth, una tecnología de comunicación inalámbrica de voz y datos para el reemplazo de cables.

Bluetooth es una tecnología de radio de corto alcance capaz de proporcionar conexión inalámbrica entre dispositivos de una forma segura y robusta.

Como conocedor e investigador de la tecnología y teniendo en cuenta el potencial que ofrece Bluetooth para el desarrollo de novedosas aplicaciones y servicios, el proyecto de grado se realizará alrededor de esta tecnología. Es de suma importancia estar a la vanguardia en el conocimiento de nuevas tecnologías y más aún cuando representan un papel fundamental en la transformación de las telecomunicaciones, como ha podido sustentarse en los párrafos anteriores.

1.1 SINOPSIS DEL PRESENTE TRABAJO INVESTIGATIVO

Este trabajo presenta cinco aspectos fundamentales de la tecnología así:

Capítulo 2 - Generalidades Donde se hace una aproximación inicial al la tecnología Bluetooth y se definen sus aspectos básicos de operación; se posiciona la tecnología en el marco de las tecnologías inalámbricas de corto alcance. También se describen los escenarios de uso básicos en los que se aplica la tecnología.

Capítulo 3 - Descripción técnica Corresponde a un estudio profundo de los protocolos que conforman la especificación Bluetooth; se ilustran los procesos de comunicación fundamentales con el fin de aclarar el funcionamiento de tales protocolos.

Capítulo 4 - Descripción de los Perfiles de Aplicación de Bluetooth Se muestra cómo pueden implementarse los escenarios de uso de acuerdo con la especificación y de esta manera se ilustran los mecanismos de interoperabilidad de las aplicaciones que se implementen con ellos.

Capítulo 5 - Significado de la tecnología para los actores de la cadena de valor. Se exponen los aspectos clave a tener en cuenta por parte de los actores del entorno de aplicaciones telemáticas inalámbricas y así mismo las implicaciones que tiene para ellos su participación en éste entorno.

Capítulo 6 – Conclusiones y Recomendaciones del Trabajo Investigativo Se destacan los logros y los conceptos que se han adquirido y extractado del estudio de la tecnología Bluetooth, separados en varias áreas de importancia; también se dan recomendaciones para posteriores investigaciones en ésta área.

Por último, como complemento al desarrollo de la investigación se presenta la elaboración de una aplicación que se soporta en esta tecnología. Con este aporte, los investigadores dejan una base de conocimiento sólida en ésta área para soportar futuros desarrollos e investigaciones.

CAPÍTULO 2 BLUETOOTH Y LAS TECNOLOGÍAS INALÁMBRICAS DE CORTO ALCANCE

2.1 EVOLUCIÓN HISTÓRICA DE BLUETOOTH

Esta parte la descripción de la tecnología Bluetooth no se hace en términos del stack de protocolos que comprende (puesto que este es el tema principal de uno de los capítulos posteriores). La idea de este capítulo es recopilar algunos conceptos de los que hace uso y luego complementar la descripción con una comparación entre Bluetooth y otras tecnologías inalámbricas de corto alcance.

Entrando en la historia de Bluetooth como tal, se puede encontrar a la división de comunicaciones móviles de Ericsson en Suecia, que en 1994 inició estudios e investigaciones acerca de la viabilidad de una interfaz de aire de bajo costo y bajo consumo de potencia con el fin de reemplazar los cables entre sus teléfonos móviles y sus accesorios.

Con el tiempo, el concepto evolucionó gracias a que el proyecto investigativo estaba enmarcado en otro más general que pretendía desarrollar aparatos que pudieran comunicarse a través de las redes celulares. De esta manera se introduce el término *enlace Multicomunicador* (MC Link).

El objetivo de este enlace MC más tarde fue generalizado y concebido como una tecnología inalámbrica para múltiples propósitos de comunicación llamada Bluetooth (gracias al apelativo dado al rey Harald II de Dinamarca, cuyo ideal era unir a los países nórdicos bajo una sola gran nación).

Se empezó entonces a buscar a otros fabricantes que pudieran interesarse en la tecnología e implementarla en sus productos; y fue en Febrero de 1998 cuando Ericsson, Nokia, IBM, Toshiba e Intel unieron esfuerzos con el objetivo de establecer un estándar de facto para la industria. Este consorcio se llamó “Bluetooth-SIG” - G*rupo Especial de Interés en Bluetooth* (Special Interesting Group)- y en la actualidad ha producido y publicado la versión 1.1 de la

especificación. El concepto de PAN utilizado por Bluetooth también se encuentra como un estándar ante los grupos de trabajo de IEEE denominado 802.15.

Para cumplir las exigencias que debería tener inicialmente Bluetooth, se diseñó para que trabajara en la banda de 2.4GHz, que hace parte de la banda denominada ISM[1], la cual es una banda en la que se puede operar sin licencia[2]; pero cumpliendo con límites de potencia y técnicas de modulación en Espectro Ensanchado (como el caso del modo de saltos de frecuencia FHSS[3] o el de secuencia directa DSSS[4]); esta es la misma banda de trabajo que utiliza 802.11b; para Bluetooth se especificó trabajar en FHSS. Esto permite tener a disposición un canal de 720Kb/seg en un perímetro de 10 metros con su mínima potencia de radio. Para lograr alcanzar un bajo consumo de potencia y bajo costo, se han ideado soluciones en un solo chip utilizando circuitos CMOS conteniendo parte RF y procesamiento de banda base.

2.2 MODELOS DE USO

Los modelos de uso son escenarios posibles donde aplicaciones con Bluetooth puedan tener lugar. Estos escenarios se encuentran contemplados en la especificación, en el volumen dedicado a perfiles, en el cual se detallan que recursos del stack de Bluetooth deben implementarse para poder llevar a la práctica cualquiera de estos escenarios.

2.2.1 Computador sin cables

Es el escenario que mejor describe la concepción de Bluetooth como una tecnología para reemplazar cables que transportan datos. Un ejemplo de grandes cantidades de estos cables es un computador de escritorio. En este caso, los periféricos, tales como el ratón, teclado, impresoras, dispositivos de juegos, parlantes, escáners, etc. pueden tener un alto grado de libertad al no estar conectados por cables, sino por enlaces de radio al PC. Este mismo esquema puede emplearse para la compartición de recursos hardware, por ejemplo, un periférico puede que no sólo sea usado por un computador, sino también por una consola de juegos, ó una impresora puede ser de uso compartido entre 2 equipos sin necesidad de intercambiar el cable de conexión.

---

[1] Banda de uso Industrial, Científico y Médico

[2] Excepto parcialmente en países como Francia, y Japón

[3] Frequency Hopping Spread Spectrum

[4] Direct Sequence Spread Spectrum

2.2.2 Headset

Los “Manos Libres” ó Headsets son dispositivos que se han estado usando para permitir a una persona mantener una conversación telefónica sin tener que sostener el teléfono con la manos, pero usualmente estos dispositivos también están sujetos a cables restringiendo la movilidad del operario; este también es un escenario que contempla Bluetooth, dado que soporta explícitamente aplicaciones de voz. Los manos libres no solo serían aplicables en telefonía, sino también como transductores de entrada y salida para computadores y videoconferencia, ó también para equipos de sonido y de grabación.

2.2.3 Teléfono 3 en 1

Usualmente se hace uso de diversos teléfonos, tales como celulares, inalámbricos en el hogar, fijos en oficinas, etc. Un teléfono celular 3 en uno estaría equipado con Bluetooth y podría interactuar de las 3 maneras: como celular normal, como inalámbrico usando la línea telefónica a través de un punto de acceso de voz, ó como radio teléfono para comunicarse con otro igual en las proximidades. (éste último caso cobra gran importancia con los dispositivos Bluetooth de potencia clase 1 ó de 20dBm).

2.2.4 Puente a Internet

Existen dos métodos para el uso de Bluetooth como un puente inalámbrico para establecer comunicación con redes LAN ó Internet:

Marcación Telefónica: Esta forma de acceso a Internet es muy similar a la que se emplea hoy en día: un arreglo convencional implica la presencia de un computador que use una línea telefónica para conectarse a un proveedor de servicios de Internet a través de un módem. Lo que Bluetooth suprime a este escenario es la presencia de cables (entre el módem y la línea telefónica). Mediante el uso de un Computador y un teléfono (sea éste móvil ó fijo) equipados con Bluetooth que soporten un perfil para “marcar a redes”, la conexión a Internet puede realizarse de forma totalmente inalámbrica.

Acceso directo a la red: El acceso a Internet por medio de LAN se realiza a través de una Gateway sin que haya necesidad de efectuar una marcación telefónica. El acceso directo a la red por medio de Bluetooth es posible usando un tipo de dispositivo llamado punto de acceso. Un Punto de acceso permite que otros dispositivos se conecten a él con el fin de proporcionar su acceso a una LAN. Se puede decir entonces que Bluetooth proporciona un *conector* inalámbrico para la conexión a redes.

2.2.5 Sincronización Automática

La sincronización automática es un ejemplo del uso de la proximidad entre redes para hacer más fácil una tarea que ya existe. La sincronización es el proceso de reunir información de dos fuentes diferentes basándose en un conjunto de reglas por las cuales la información resultante es idéntica en las dos fuentes originales. Con Bluetooth es posible que dos dispositivos se sincronicen automáticamente siempre y cuando se encuentren en un rango de comunicación. Por ejemplo, un PDA guardado en el bolsillo de la chaqueta de una persona podría sincronizarse con el Laptop de la misma persona mientras ella camina por su oficina; esto con el fin de actualizar, por ejemplo, un conjunto de archivos en el Laptop que ella modificó en su PDA mientras se encontraba fuera de su oficina, de este modo, la información será la misma y estará actualizada en los dos dispositivos.

2.2.6 Conferencia Interactiva (Transferencia de Archivos)

La transferencia de archivos está ligada generalmente al uso de cables y unidades de almacenamiento. Un entorno inalámbrico permite establecer enlaces temporales entre dispositivos de manera que puedan intercambiarse archivos y otros arreglos de datos. Es posible establecer una conferencia interactiva en donde los participantes de la reunión pueden intercambiar tarjetas de negocios ó archivos empleando Laptops que incorporan módulos Bluetooth.

2.3 EL CONCEPTO DE PICORED

Bluetooth utiliza un concepto de red de tamaño pequeño ó personal, llamada *picored*, la cual se forma para que haya transferencia de información entre dispositivos. Bluetooth fue diseñada en forma básica para reemplazar cables de datos entre dispositivos como teléfonos móviles, Asistentes digitales personales (Personal Digital Asistant - PDA), computadores portátiles, vídeo beams y cualquier otra clase de dispositivo (hornos microondas, por ejemplo). Esta capacidad puede hacer que ellos se conecten, compartan información y recursos, lo cual en principio es el trabajo de una red. Estas redes se forman automáticamente al encontrarse sus elementos cerca unos de otros; por tal razón, las redes que se forman se denominan redes *ad – hoc.*

Se tiene entonces una red inalámbrica de poca extensión (~10m de radio) que existe siempre y cuando sus elementos permanezcan dentro del alcance de sus radios Bluetooth. El concepto de Picored no se refiere solamente al espacio de alcance físico de los dispositivos, sino más bien a redes de tipo lógico, que comparten un canal común y cuyas

áreas pueden traslaparse, inclusive por completo en un momento dado. Cuando varias picoredes tienen elementos comunes, se forman redes compuestas ó mejor llamadas "*scatternets*".

2.4 CARACTERÍSTICAS DE RADIO DE BLUETOOTH

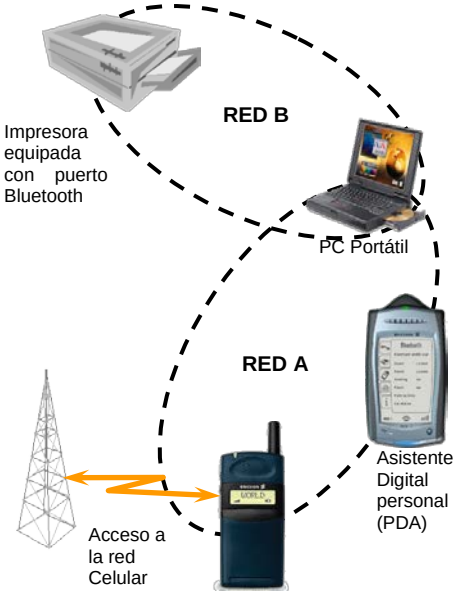
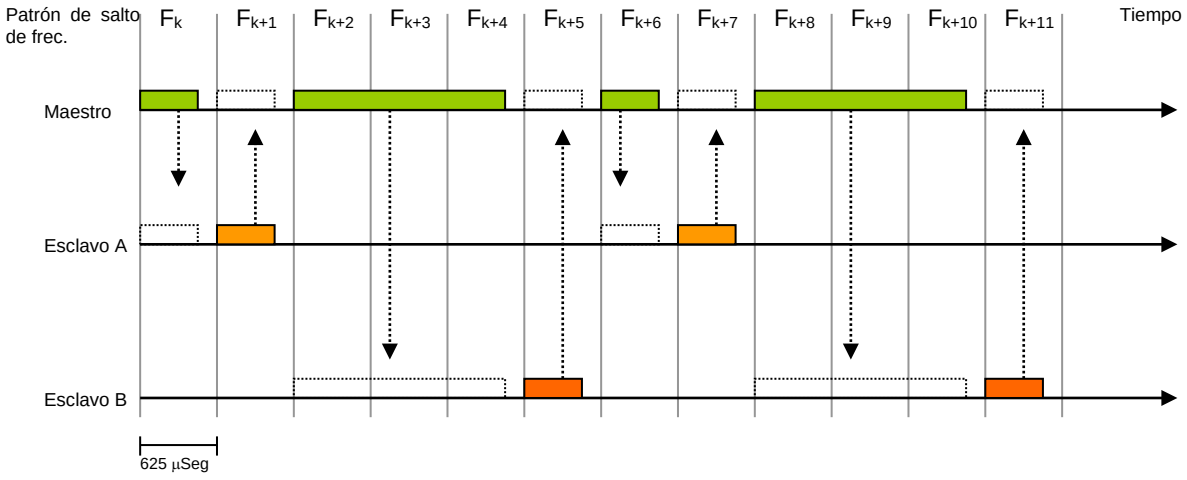


Figura 2.1 Múltiples picoredes que pueden coexistir.

Bluetooth utiliza la técnica de espectro ensanchado en saltos de frecuencia de manera conveniente a la topología que maneja. El hecho de que sea espectro ensanchado indica que la señal de radio distribuye su potencia en un rango de frecuencias ó espectro más ancho del que en realidad necesita (modulando la señal en una sola portadora) para realizar la comunicación de datos; esto significa que dicha potencia no se concentrará en una sola frecuencia, sino que será una potencia baja distribuida, lo cual es ventajoso para equipos similares que trabajen en esta misma banda de frecuencias, puesto que la señal de espectro ensanchado se percibe en los receptores de banda angosta como si fuera un ruido blanco.

La técnica de espectro ensanchado de Saltos de Frecuencia (FHSS) se explica desde el punto de vista de 802.11b en la descripción de las tecnologías inalámbricas de corto alcance (ver anexo) en esta misma monografía. Para Bluetooth este concepto no difiere, pero hay aspectos importantes que se hace necesario describir en este momento.

Se ha dividido la banda de 2.4GHz en 79 subcanales de 1MHz cada uno (en algunos países



son solo 32 canales por restricciones en la banda de aplicación industrial, científica y médica (Industrial Scientific and Medical - ISM) y que se establecen patrones de saltos definidos en el transmisor y receptor. Para Bluetooth el patrón de salto no es el que define una comunicación punto a punto, sino una Picored entera; en el caso de la figura 2.1, la Picored A y la Picored B trabajan cada una con un patrón de salto distinto; así, los elementos que pertenezcan a una Picored saben qué patrón esta siendo compartido por todos los demás elementos cercanos. Existe entonces un canal común a todos los dispositivos de la Picored, un medio físico susceptible de ser utilizado para la comunicación; es aquí donde cobra importancia la técnica de acceso a dicho medio: TDMA / TDD.

Figura 2.2 Acceso al medio en una Picored

Se puede apreciar en la figura 2.2 que se usan las dos técnicas: TDD para permitir a los esclavos transmitir hacia el maestro, y TDMA al permitir un lapso de tiempo para que cada dispositivo acceda al medio. Los esclavos transmitirán en las ranuras de tiempo impares, y el maestro transmitirá en las ranuras pares; existen varios tipos de paquetes para Bluetooth, son un total de 16 paquetes, unos tienen una duración fija de una ranura de tiempo, otros de 3 ranuras, y otros de 5. Con el esclavo B se puede observar que hay un enlace asimétrico con el maestro, hay un paquete de 3 ranuras de duración y en el otro sentido hay uno de una sola ranura. (estas definiciones pueden encontrarse mas detalladamente en la descripción de la banda base, en el capitulo 3). Esta secuencia de saltos de frecuencia se realiza con una velocidad de 1600 saltos por segundo, lo cual da una duración para cada ranura de tiempo de 625μs; esto es más rápido que la velocidad de saltos definida para la 802.11b. Durante los 625μs de duración de cada ranura, la señal de datos digital es modulada con GFSK (Gaussian Frequency Shift Keying) alrededor de la frecuencia que corresponda según la secuencia de saltos de frecuencia. Por supuesto, todos los dispositivos conocen de antemano el patrón de saltos que deben seguir.

Al existir paquetes pequeños (debido a una velocidad de saltos mayor), se disminuye la probabilidad de que el sistema se vea afectado por aparatos como hornos microondas y otros que trabajen en la banda de 2.4GHz.

Existen 3 clases de unidades Bluetooth de acuerdo a su potencia de radiación:

Clase 1: 100mW 20dBm

Clase 2: 25mW 4dBm

Clase 3: 1mW 0dBm

Actualmente son comunes los dispositivos clase 3, puesto que responden a las características de ahorro de potencia y calidad de servicio que dicta la especificación, siendo también más prácticos para ser colocados en aparatos a baterías.

La potencia de radio de salida no es siempre la que se indicó antes, ese es un tope máximo, puesto que la potencia puede cambiar de acuerdo a qué tan cerca se encuentran unos dispositivos de otros; aquí se observa la implementación de una de las características básicas de Bluetooth: la de ahorro de energía. Funciona igual como cuando se controla el volumen de voz cuando se habla con alguien que está muy lejos, y de la misma forma, un dispositivo Bluetooth puede informar a otro que la potencia de radio que está recibiendo es muy superior a la que él necesita para detectar la señal; pudiendo su interlocutor, bajar el nivel de señal de salida.

2.5 COMPORTAMIENTO DE UN DISPOSITIVO BLUETOOTH

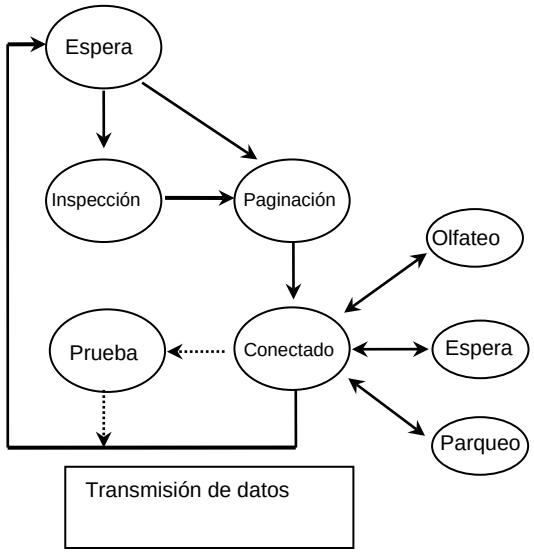


Figura 2.3. Estados del dispositivo Bluetooth

Un dispositivo Bluetooth obedece a un comportamiento descrito por estados. Cuando dos elementos ó dispositivos coinciden dentro del alcance de sus radios, cada uno empieza a inspeccionar qué otro elemento se encuentra a su alcance, qué servicios ofrece y cuál es su

dirección física. Esto ocurre dentro de un estado de inspección ó *INQUIRY*. (Referirse a la figura 2.3). Se ha establecido ya una Picored. Cada Picored deberá tener establecido un canal para comunicarse. Para regular el tráfico en dicho canal, uno de los elementos asume el papel de maestro, y los demás elementos de la red se denominarán *esclavos*. Dado que ellos comparten un mismo canal, si su número aumentara, el desempeño de la red en total disminuiría, por ello se permite que los elementos que lo necesiten puedan establecer conexiones entre sí, y con ello formar otra Picored. Varias picoredes en esta situación formarán una red compuesta ó *SCATTERNET*. En la figura 2.3 se observa que la red A puede haber estado formada desde un principio; luego pudo haberse formado otra entre la impresora y el PC portátil.

La eficiencia de la red compuesta que se ha formado es entonces mayor (puesto que hay dos canales de comunicación diferentes para ser aprovechados) y el volumen de datos (throughput) se incrementa. Una vez terminada la fase de inspección ya se tienen las direcciones de los dispositivos y estos pueden empezar a transmitir, pero no sin antes realizar una paginación ó *PAGING*. El estado de conexión es realmente el momento en el que se transfieren los datos y una vez terminada la transferencia, se puede retornar al estado de espera ó *STANDBY*, ó bien quedar en alguno de los estados especiales de bajo consumo de energía del dispositivo Bluetooth: SNIFF, HOLD y PARK. Aquí se presenta otro mecanismo para proporcionar bajo consumo de potencia, y por ello estos tres estados se diferencian en qué tan reactivo se vuelve el sistema ante señales externas y qué consumo de potencia requiere.

2.6 CARACTERÍSTICAS DE SEGURIDAD

Con el fin de dar protección y confidencialidad a la información, el sistema toma medidas de seguridad en la capa de aplicación y la capa de enlace. Estas medidas de seguridad deben ser apropiadas para un ambiente punto a punto. Esto significa que en cada unidad Bluetooth, las rutinas de autenticación y encripción se implementan de la misma forma. Cuatro tipos de identidades se usan en la capa de enlace para llevar a cabo los procedimientos de seguridad.

Dirección del dispositivo BD_ADDR: 48 bits
Clave privada de autenticación: 128 bits
Clave privada de encripción: Longitud variable 8-128 bits.
Número aleatorio RAND: 128 bits.

La naturaleza secreta de las claves impide que estas puedan conocerse por medio de los procedimientos de INQUIRY. En adición, el procedimiento de seguridad requiere un PIN que solo es conocido por el usuario (o la aplicación) para acceder un dispositivo particular.
Bluetooth define tres tipos de procedimientos que componen el proceso de seguridad:

Autenticación: Es el proceso de verificar quien está al otro lado del enlace. Esta procedimiento se realiza para verificar las direcciones de los dispositivos y se basa en el almacenamiento de una clave de enlace o por pareamiento de un PIN.

Autorización: Es el proceso por medio del cual se verifica si a un dispositivo le está permitido tener acceso a un servicio. La Autorización siempre comprende el procedimiento de Autenticación.

Encriptación: La información de usuario puede ser protegida por encripción del *payload* de los paquetes de datos; el código de acceso y encabezado del paquete nunca se encriptan.

También existe un proceso de *Gestión de Clave* mediante el cual se generan las claves necesarias para cada uno de los procedimientos anteriores. Se genera una clave de inicialización a través del PIN, su longitud, un número aleatorio y la dirección del dispositivo.

La unidad verificadora, envía un número al azar generado por un proceso especifico de autenticación. Este número se presenta cada vez que el dispositivo solicitante el cual tiene la cave de inicialización correcta y la dirección necesaria del dispositivo sean capaces de producir un número respuesta que es conocido por el verificador. Este numero se envía de regreso y es chequeado por el verificador.

Cada unidad Bluetooth tiene una clave guardada en su memoria no volátil. El dispositivo usa la clave de inicialización para encriptar esta clave única y enviarla al otro dispositivo el cual la desencripta usando la clave de inicialización intercambiada anteriormente.

El segundo dispositivo puede agregar su propia clave a la clave del primer dispositivo y generar un clave combinada si los dispositivos están en capacidad de soportarlo. De otra manera se usa la clave de uno de los dispositivos como clave de enlace. La clave de inicialización se elude después de este procedimiento.

La clave de encripción se genera a partir de la clave de enlace, un número aleatorio y un número que se obtiene de una operación de mezcla. Esta clave con algunas modificaciones se usa para encriptar la carga útil de datos.

La clave de enlace se renombra y si se establece otro enlace entre dos dispositivos se puede usar la misma clave de enlace para eliminar el envió de información sobre el canal nuevamente.

2.7 COMPARACIÓN ENTRE BLUETOOTH Y TECNOLOGÍAS INALÁMBRICAS DE CORTO ALCANCE

Tabla 2.1 Comparación entre Bluetooth y el estándar 802.11

	Bluetooth	**802.11a**	**802.11b**
Velocidad	1Mbps (712Kbps) como máximo.	Velocidades de 6, 12 y 24Mbps; opcionalmente 9, 18, 36 y 48Mbps. También 54Mbps con QAM.	Velocidades desde 2Mbps hasta 11Mbps;.
Seguridad	Parte fundamental de la especificación Adaptación automática de potencia de radio como parte del procesamiento de banda base. Encriptación Autenticación	Opcionalmente implementa WEP Encriptación y autenticación.	Opcionalmente implementa WEP Encriptación y autenticación.
Número de unidades por red	8 por piconet	Cualquier cantidad de unidades dada la técnica de acceso al medio CSMA/CA, pero con detrimento del desempeño de la red en conjunto.	Cualquier cantidad de unidades dada la técnica de acceso al medio CSMA/CA, pero con detrimento del desempeño de la red en conjunto.
Técnica de modulación	GFSK (Gaussian Frecuency Shif Keyingt)	OFDM (orthogonal Frequency Division Multiplexing)	CCK (Complementary Code Keying)
Técnica de espectro ensanchado	FFSS (1600 saltos)	--	FHSS ó DSSS
Técnica de Acceso al medio	TDMA (acceso entre dispositivos) y TDD (para recepción y transmisión)	CSMA/CA	CSMA/CA
Alcance	10 m (100m para unidades clase 1)	~50 m	50 m – 100m
Potencia	Estados de ahorro de energía (Sniff, Hold y Park) se diferencian por su nivel de actividad y receptividad. Existen Indicaciones de maestro a esclavo para salir de alguno de estos estados.	Estados de Bajo consumo de potencia con colas de almacenamiento.	Estados de Bajo consumo de potencia con colas de almacenamiento.
Topología	Piconet – Scatternet (AdHoc) Cliente/Servidor, punto a punto para distancias cortas	De infraestructura y topología AdHoc	De infraestructura y topología AdHoc
Soporte de audio y vídeo	Soporta aplicaciones de audio (voz) PCM ó CVSD; se puede lograr audio de calidad usando canales de datos. Aplicaciones que requieran sincronía y asincronía; soporta aplicaciones para compartición de recursos.	Aplicaciones de Audio y Vídeo Transferencia de archivos. Compartición de recursos.	Aplicaciones de Audio y Vídeo Transferencia de archivos. Compartición de recursos.
Manejo de interferencia	Ajuste automático de nivel de potencia de radio. Corrección hacia adelante de errores (FEC) en los encabezados de los paquetes y opcionalmente en su contenido. La técnica FHSS reduce al mínimo la interferencia entre dispositivos.	Chequeo de redundancia Cíclica (CRC) para paquetes a nivel MAC. Baja velocidad de símbolos. realiza control de secuencia de mensajes. utiliza mensajes ACK a nivel de la capa MAC.	Chequeo de redundancia Cíclica (CRC) para paquetes a nivel MAC. Baja velocidad de símbolos. realiza control de secuencia de mensajes. utiliza mensajes ACK a nivel de la capa MAC.
Aplicaciones	Sustitución de cableado de datos por Radio. Puntos de acceso a la red. Computación oculta (Hidden Computing), compartición de recursos. Instrumentación industrial inalámbrica. Sistemas empotrados y electrodomésticos.	Extensiones de LAN cableadas. Puntos de acceso a la red. Puentes entre LANs remotas. Compartición de archivos. Aplicaciones para el hogar y para PC.	Extensiones de LAN cableadas. Puntos de acceso a la red. Puentes entre LANs remotas. Compartición de archivos. Aplicaciones para el hogar y para PC.
Roaming	No hay concepto de “Celda” de cobertura, los dispositivos que se encuentren al alcance de sus radios se comunican según sea necesario.	Roaming especificado en tipos de mensajes, pero no en protocolos.	Roaming especificado en tipos de mensajes, pero no en protocolos.
Costos	Costos de fabricación bajos, (alrededor de 20 dólares) tendientes a bajar con la masificación de los chips y los productos.	Los costos de las tarjetas y dispositivos consultados están cercanos ó superiores a los 200 dólares aproximadamente.	Los costos de las tarjetas y dispositivos consultados están cercanos ó superiores a los 200 dólares aproximadamente.
Modelos de uso	Modelo Ad Hoc.	Modelos de infraestructura y modelo Ad Hoc	Modelos de infraestructura y Ad Hoc

Tabla 2.2 Comparación entre Bluetooth, HomeRF e IrDA

	Bluetooth	**HomeRF**	**IrDA**
Frecuencia de Trabajo	2.4 GHz	2.4 GHz	Infrarojo
Número de unidades por red	8 por piconet	127 por red	2 por red
Técnica de modulación	GFSK (Gaussian Frecuency Shif Keyingt)	2-4 FSK	PPM
Técnica de espectro ensanchado	FHSS (1600 saltos)	FFSS (50 saltos)	- -
Técnica de Acceso al medio	TDMA (acceso entre dispositivos) y TDD (para recepción y transmisión)	TDMA/TDD – CSMA/CA	- -
Velocidad	1Mbps (712Kbps) como máximo.	1.6Mbps - 10MHz	115.2Kbps - 16 MHz
Alcance	10 metros (100 m para unidades clase 1)	10 - 100 metros	10cm - 10 metros
Potencia	Estados de ahorro de energía (Sniff, Hold y Park) se diferencian por su nivel de actividad y receptividad. Existen Indicaciones de Maestro a esclavo para salir de alguno de estos estados.	100mW 3.3v, 250mA	2.7 - 3.6V 20nA
Topología	Piconet – Scatternet (AdHoc) Cliente/Servidor, Peer to Peer para distancias cortas	Cliente/Servidor, Peer to Peer	Sistema Punto a Punto. Ad-Hoc
Seguridad	Parte fundamental de la especificación Adaptación automática de potencia de radio como parte del procesamiento de banda base. Encriptación Autenticación	Autenticación de usuario por medio del Identificador de red NWID de 24 bits. Poderoso mecanismo de encripción de datos con una clave de 128 bits. Secuencia de saltos en frecuencia aleatoria.	Línea de vista.
Soporte de audio y vídeo	Soporta aplicaciones de audio (voz) PCM ó CVSD; se puede lograr audio de calidad usando canales de datos. Aplicaciones que requieran sincronía y asincronía; soporta aplicaciones para compartición de recursos.	HomeRF soporta aplicaciones claves de banda ancha como Internet, audio y vídeo. Hasta 6 conversaciones full duplex.	*IrMC Infrared for Mobil Communications* incluye RTCON, un componente de la especificación para la transmisión de voz full duplex sobre un enlace IrDA.
Manejo de interferencia	Ajuste automático de nivel de potencia de radio. Corrección hacia adelante de errores (FEC) en los encabezados de los paquetes y opcionalmente en su contenido. La técnica FHSS reduce al mínimo la interferencia entre dispositivos.	Mediante FHSS se logra inmunidad a interferencia para conexiones de datos asíncronas.	- -
Aplicaciones	Sustitución de cableado de datos por Radio. Puntos de acceso a la red. Computación oculta (Hidden Computing), compartición de recursos. Redes Ad Hoc. Instrumentación industrial inalámbrica. Sistemas empotrados y electrodomésticos.	Redes domesticas inalámbricas. Acceso a Internet. Activación y control de electrodomésticos a través de comandos hablados. Juguetes y consolas de juego.	Sincronización de datos entre PDA y PC. Acceso a Internet y LAN. Realización de pagos electrónicos. Terminal de datos industrial, Instrumentación y medición,
Roaming	No hay concepto de “Celda” de cobertura, los dispositivos que se encuentren al alcance de sus radios se comunican según sea necesario.	HomeRF 2.0 agrega soporte para Roaming y Hand-Off	- -
Costos	Costos de fabricación bajos, (alrededor de 20 dolares)	10 - 25 dólares por chip	1 - 2 dólares por chip

CAPITULO 3 DESCRIPCIÓN DE LA PILA DE PROTOCOLOS BLUETOOTH

La especificación de la tecnología Bluetooth se divide en dos partes, el núcleo y los perfiles. El núcleo de la especificación explica cómo trabaja la tecnología. Los perfiles son modelos de aplicaciones para caracterizar servicios de forma genérica. Su definición se hace estableciendo mensajes y procedimientos entre las capas de la pila de protocolos. A través de los perfiles, se puede describir cómo elaborar aplicaciones y construir dispositivos que utilicen el estándar y sean interoperables. Este capitulo tiene como objetivo exponer la primera parte de la especificación que corresponde el núcleo, en el cual que se describen las capas que componen la pila de protocolos Bluetooth así como las transacciones que tienen lugar para establecer la comunicación entre dos dispositivos.

3.1 PILA DE PROTOCOLOS BLUETOOTH

La pila de protocolos Bluetooth consiste en un conjunto de procedimientos software relacionados, cada uno de los cuales ejecuta tareas específicas requeridas para soportar la comunicación entre dos dispositivos Bluetooth. Los niveles de protocolos de Bluetooth trabajan juntos para asegurar que los datos se transfieran de manera confiable desde una unidad hacia otra. Así como en otros protocolos, el programa de aplicación del dispositivo se comunica en primera instancia con el nivel superior, el cual a su vez se comunica con las capas inferiores en orden descendente. El nivel más bajo se comunica con su nivel homologo en otro dispositivo, enviando paquetes de datos y control a través de enlaces Bluetooth. En el dispositivo remoto, la comunicación entre capas se ejecuta de forma inversa hasta que los datos lleguen al nivel de aplicación.

Una característica clave de la especificación, como se mencionó anteriormente es su afán por permitir que dispositivos de diferentes fabricantes trabajen unos con otros sin restricciones de ningún tipo. Por ende el protocolo Bluetooth no solo define un sistema de radio en particular, sino que además define un software inherente para que las aplicaciones elaboradas bajo esta tecnología, puedan encontrar otros dispositivos en un área, descubrir servicios y hacer uso de ellos.

La pila de protocolos Bluetooth se ilustra en la Figura 3.1.

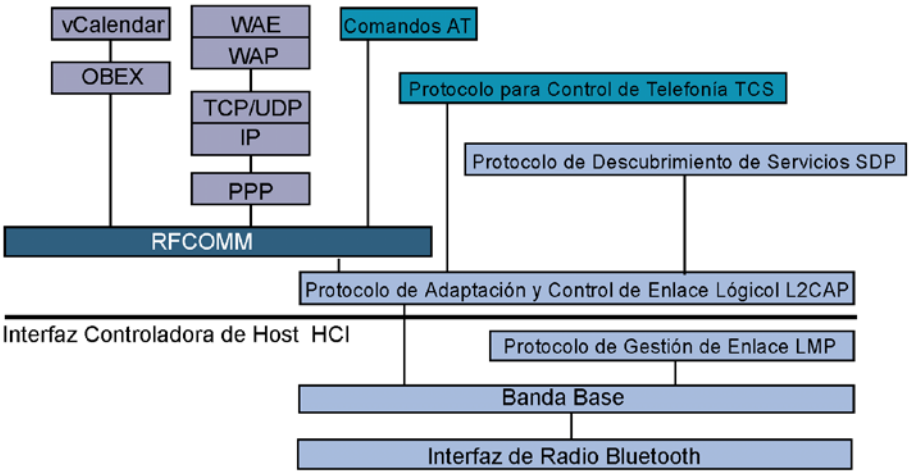


Figura 3.1. Pila de protocolos Bluetooth

Existen protocolos específicos para Bluetooth y otros que han sido adoptados con el fin de facilitar la adaptación de aplicaciones que utilizan estos protocolos ya existentes.

El SIG ha dividido la pila en cuatro capas de acuerdo con su propósito. (Ver Tabla 3.1)

Tabla 3.1. Protocolos de Bluetooth

CAPAS	PROTOCOLOS
Protocolos Base de Bluetooth	• Banda Base • Protocolo de Gestión de Enlace LMP (*Link Manager Protocol*), • Protocolo de Adaptación y Control de Enlace Lógico L2CAP (*Logical Link Control and Adaptation Protocol*) • Protocolo de Descubrimiento de Servicios SDP (*Service Discovery Protocol*)
Protocolo de sustitución de Cable	• RFCOMM (Interfaz Serial COMM RF)
Protocolos de Control De Telefonía	• Control de telefonía – Binario TCS BIN (Telephony Control – Binary), • Control de Telefonía – Comandos AT

Protocolos Adoptados	• Protocolo Punto a Punto PPP, • TCP/UDP/IP, • OBEX, • WAP

Los protocolos Base de Bluetooth, comprenden exclusivamente los protocolos desarrollados por el SIG, y en unión con la interfaz de radio Bluetooth, son requeridos por la mayoría de dispositivos.

Juntos, el protocolo para substitución de cable, el protocolo de control de telefonía y los protocolos adoptados permiten a las aplicaciones correr sobre los protocolos Base de Bluetooth.

3.1.1 Equivalencia entre el Modelo de Referencia OSI y Protocolos de Bluetooth

Aunque no existe correspondencia exacta entre los dos modelos, sus capas pueden relacionarse con el fin de establecer las responsabilidades (Funciones y Operaciones) en cada capa del protocolo Bluetooth de acuerdo con la referencia OSI.

La figura 3.2 ilustra el modelo de referencia OSI y la pila de protocolos de Bluetooth.

Figura 3.2. Equivalencia entre la pila OSI y la pila Bluetooth

El nivel físico es responsable de la interfaz eléctrica al medio de transmisión e incluye también procedimientos como la modulación y la codificación del canal. Esta parte está bajo responsabilidad de la interfaz de *Radio* y parte de la *Banda Base* en el conjunto de protocolos de Bluetooth.

La capa de enlace es responsable de la transmisión, formación de tramas y control (detección y corrección) de errores sobre un enlace particular. Esta funciones recaen sobre el segmento de control de *Banda Base* y la capa denominada *Link Manager*.

La capa de red es responsable de la trasferencia de datos a través de la red, independientemente del medio de transmisión y la topología. De estas funciones se encargan la parte superior del *Link Controler* y parte adicional del *Link Manager*.

La capa de transporte debe garantizar la confiabilidad en la trasferencia de la información y multiplexación de los datos a través de la red según el nivel determinado por la aplicación. En Bluetooth estas funciones estarán a cargo de una porción del *Link Manager* y la capa llamada *HCI* la cual posee los mecanismos para el transporte de datos.

La capa de sesión proporciona servicios de gestión y control sobre el flujo de datos lo cual se logra mediante la implementación de la capa *L2CAP* y la parte inferior de las capas *RFCOM*M y *SDP*.

La capa de presentación proporciona una representación común de los datos a la capa de aplicación agregando la estructura de servicio a las unidades de información. Esta es tarea del *RFCOMM* y *SDP*. Y finalmente la capa de aplicación es la que hace uso de la información transmitida entre dos unidades Bluetooth.

En las siguientes secciones se describen uno a uno los niveles de protocolos exclusivos de Bluetooth exceptuando la capa de Radio, que fue descrita en el capítulo anterior. Los protocolos que no son propios de la pila no están cubiertos por este documento.

3.2 NIVEL BANDA BASE

3.2.1 Introducción

La capa de Banda Base de Bluetooth se encarga de aprovechar la potencialidad y simplicidad del transceptor de radio en espectro ensanchado FHSS[1] con el fin de hacer posible la transferencia de información entre dispositivos Bluetooth. Cabe anotar que se debe enmarcar siempre a los dispositivos ó unidades Bluetooth dentro de una piconet; dicho concepto se definió en el Capítulo 2, pero su origen dentro de la tecnología nace aquí, donde los papeles de Maestro y Esclavo son de gran importancia para el proceso básico de la comunicación.

La transferencia básica de información entre dispositivos Bluetooth involucra desde el principio el establecimiento de un canal de comunicación, es decir, la formación de una Picored. Luego de esto debe haber una serie de reglas para poder utilizar dicho canal de forma eficiente; esto se refiere a las técnicas de acceso al canal, la definición de distintos formatos de tramas ó paquetes (lo cual le da versatilidad al uso del canal y permite el establecimiento de enlaces para diferentes aplicaciones).

Si se tiene en cuenta las características con las que fue diseñada la tecnología Bluetooth, se obtiene la siguiente correspondencia con las funciones que se encuentran en el nivel la banda base:

- Reemplazo de cables : Establecimiento de enlaces seguros de datos y de voz.
- Bajo consumo de energía : Definición de estados de bajo consumo de energía
- Para dispositivos móviles : Relativa simplicidad de operación (bajo consumo de recursos de procesamiento).

En éstos dos últimos casos debe tenerse en cuenta que los módulos Bluetooth incluyen la interfaz de Radio y el procesamiento de Banda base en un solo circuito integrado con el fin de cumplir con estas características.

[1] Frequency Hopping Spread Spectrum

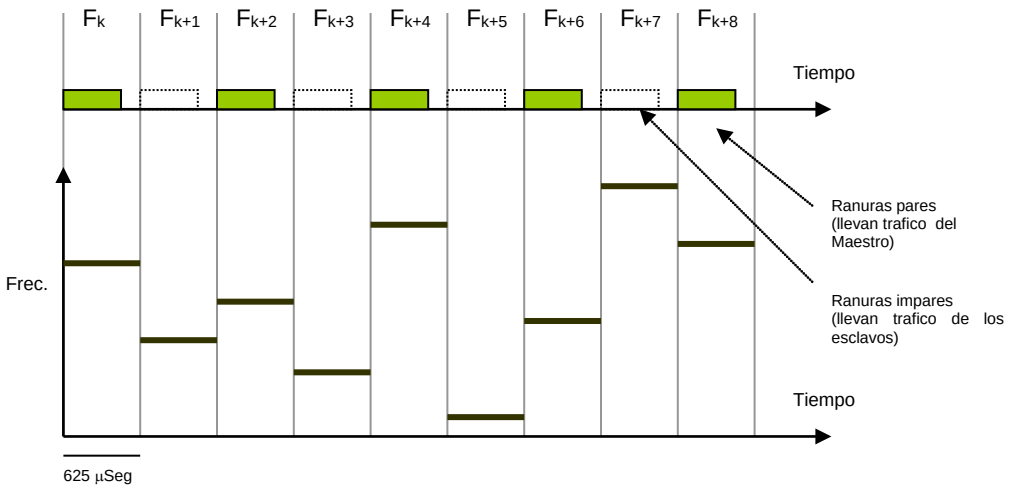


Figura 3.3. Comportamiento de una picored

3.2.1.1 Responsabilidades de la Banda Base

Las responsabilidades que puntualmente recaen en este nivel son las siguientes:

- Sincronismo y creación de la Picored
- Control de acceso al canal
- Definición de tipos de paquetes
- Detección y posible corrección de errores en la transmisión
- Garantizar seguridad de los enlaces que se establecen
- Procurar un consumo bajo de energía

Alrededor de estas responsabilidades puede enmarcarse toda la funcionalidad del procesamiento de banda base.

3.2.2 Sincronismo de la Picored y Establecimiento del Canal

La Picored y el Canal de comunicación tienen un carácter dual, puesto que al crearse la Picored se establece también un canal común de comunicación para todos los dispositivos que la conforman. Si se pudiera observar el comportamiento en el tiempo de una Picored en funcionamiento, se percibiría lo que se muestra en la Figura 3.3.

Las frecuencias se encuentran distribuidas en el tiempo, y siguen una secuencia pseudo aleatoria de saltos de frecuencia (un patrón de la técnica FHSS como se explicó en el

capítulo 2); cada ranura de tiempo ó *Slot* tiene una frecuencia determinada por dicha secuencia; las transmisiones que se realicen entre dispositivos deberán estar restringidas a estas frecuencias. Un receptor sabrá entonces que tiene que estar sintonizado a dicha frecuencia y seguir la misma secuencia ó patrón de salto para poder recibir información.

3.2.2.1 Los papeles de Maestro y Esclavo en una Picored

Una Picored tiene como actores principales a un dispositivo llamado maestro y varios dispositivos llamados esclavos; el maestro está encargado de coordinar el uso del canal de la Picored como se verá más adelante en el punto de 3.2.3 de control de acceso al canal. Cabe señalar que los dispositivos Bluetooth son construidos iguales y cualquiera de ellos puede en un caso dado ejercer el papel de maestro ó esclavo, según sea conveniente.

Por definición un maestro es aquel dispositivo que, sabiendo qué dispositivos se encuentran a su alrededor, inicia una comunicación con uno ó varios de ellos, éstos a su vez tomarán el papel de esclavos de la Picored. Las comunicaciones que existen en una Picored son en sentido maestro – esclavo ó esclavo – maestro, pero no entre esclavos (para ello dos esclavos pueden mejor formar otra Picored independiente, quedando de esta manera una configuración de Scatternet ó red compuesta).

La forma de direccionar a cada uno de los esclavos dentro de la red es por intermedio de direcciones de 3 bits, lo cual quiere decir que solo podrán haber 7 esclavos activos en la Picored. Este número puede parecer pequeño, pero cuando se trata del número máximo de dispositivos accediendo al mismo tiempo a un mismo canal, es mejor que éste número sea bajo para garantizar un buen desempeño de la red en general. Un maestro puede hacer uso de los modos de bajo consumo de energía de sus esclavos para hacer que estos permanezcan inactivos ante la Picored y de esta manera poder utilizar sus direcciones y el canal más eficientemente.

Dado que el maestro tiene la responsabilidad de manejar el acceso al canal de la Picored, tiene acceso a éste en una mayor proporción que los esclavos; esto puede visualizarse en la figura 3.3. en la que se adopta un esquema de acceso múltiple por división de tiempo (Time Division Multiple Access - TDMA) y al mismo tiempo duplexación por división de tiempo (Time Division Duplexing -TDD). El esquema TDD se utiliza para asegurar la comunicación dúplex entre esclavo y maestro, es decir, las ranuras de tiempo pares se utilizan para que el maestro transmita información; en las impares el maestro “escucha” comunicaciones de los

esclavos. El acceso de los esclavos al canal es TDMA y se soporta bajo el principio descrito anteriormente, los esclavos podrán acceder al canal a su debido tiempo en las ranuras impares (el maestro se encargará de asignar cuales de esas ranuras deberán ocupar).

3.2.2.2 Sincronismo Mediante el Reloj

Las unidades Bluetooth poseen un reloj interno que corre a 3600 ciclos por segundo, esto es el doble de la rapidez de saltos de frecuencia que utiliza el esquema de saltos de frecuencia FHSS indicado en la figura 3.3. El reloj alimenta un contador de 28 bits, lo que significa que no se reciclará sino en un lapso de tiempo de más ó menos 24 horas, esto es útil dado que este reloj es utilizado en la generación de varios parámetros de la comunicación como se verá más adelante en este capítulo. El bit 1 del reloj nativo[2] se utiliza como señal de sincronismo para efectuar los saltos de frecuencia (puesto que esta cambiando cada 625μseg). El módulo selector de frecuencia tiene como entradas:

- los 28 bits menos significativos de la dirección física del dispositivo Bluetooth (de 48 bits).
- 27 bits del contador de reloj nativo descrito anteriormente. El bit *0* solo se utiliza para generar secuencias de salto de 3600 saltos por segundo, como se verá en los procesos de indagación y de paginación.

Como resultado del procesamiento de estas dos entradas, el Modulo selector de frecuencia produce como salida la frecuencia instantánea (para cada ranura de tiempo) a la cual se debe transmitir ó recibir, la cual cumplirá con los requisitos de aleatoriedad para espectro ensanchado[3]. Una determinada entrada de dirección en un modulo selector puede generar 5 secuencias diferentes de salto:

- Una secuencia de Paginación (para el estado de paginación)
- Una secuencia de respuesta a la paginación (para el estado de respuesta a paginación)
- Una secuencia de Indagación ó Inquiry (para el estado de indagación)

---

[2] Se refiere al reloj que trae incorporado el módulo Bluetooth

[3] Según la Comisión Federal de Comunicaciones de Estados Unidos (Federal Communications Commission –FCC) el orden en que aparecen las frecuencias debe ser aleatorio ó al menos pseudoaleatorio; la secuencia generada por el módulo selector de frecuencia es aleatoria dentro de un lapso de tiempo largo sin presentar secuencias iguales ó periódicas.

- Una secuencia de respuesta a la Indagación ó Inquiry (para el estado de respuesta a indagación)
- Una secuencia de canal de comunicación (constituye el canal de radio por el que se comunicará toda la Picored si dicho dispositivo llega a ser el maestro)

El canal de comunicación entonces está formado por la secuencia de canal obtenida de la dirección física y el reloj del maestro. Para que todos los esclavos puedan estar vinculados a la Picored deben entonces conocer:

- La dirección física Bluetooth del maestro
- El valor en un instante dado del reloj del maestro para estar sintonizados a la misma frecuencia en el mismo instante de tiempo con la secuencia de saltos de la Picored. Para esto, el modulo de selección de frecuencia del esclavo se alimenta con la dirección del maestro y el reloj propio (adicionándole al reloj propio un valor de deslizamiento u *offset* con el fin de que su valor instantáneo sea igual al del maestro).

3.2.2.3 Formación de una Picored

Se debe partir del hecho de que todos los dispositivos se encuentran inactivos y no saben de la existencia de otras unidades Bluetooth cercanas, a pesar de que la distancia entre estas sea lo suficiente para entablar una comunicación.

Un dispositivo Bluetooth, dada su programación puede estar inactivo, ó por el contrario, siempre receptivo para descubrir qué unidades Bluetooth se encuentran en la cercanía, (esto se hace inicializando ciertos registros que comandan el comportamiento de banda base; estos son principalmente valores de temporizadores) este hecho lleva al primer procedimiento que tiene que llevarse a cabo para formar una Picored: el de Indagación ó *Inquiry.* Cuando ya se conoce qué dispositivos se encuentran cercanos, se les puede invitar a intercambiar información según se necesite; esto se logra por medio del procedimiento de paginación *(Paging).* Al finalizar éste procedimiento se ha establecido un canal, y por consiguiente, una Picored.

3.2.2.4 Proceso de Indagación (Inquiry)

Este proceso solamente indica al dispositivo indagador la existencia de las unidades cercanas, obtiene información acerca de las direcciones físicas de dispositivo (*Bluetooth*

Device Address[4]) y valores de reloj de cada uno de los dispositivos. Esto con el fin de que en cualquier momento al requerir comunicación con ellos se pueda formar una Picored. Dado que un dispositivo puede no siempre estar receptivo y escuchar qué dispositivos estarán cerca, se le puede programar para que cada cierto tiempo entre en estados de escaneo (receptivo y listo a responder a indagaciones desde otros dispositivos) ó realice indagaciones a otros dispositivos.

El proceso a llevarse a cabo es el siguiente. Suponiendo que existan 3 dispositivos en un mismo recinto, uno de ellos entra en el estado de indagación ó Inquiry.

1. En este estado el dispositivo selecciona un patrón de saltos de frecuencia determinado para operaciones de indagación, así que empieza a transmitir paquetes ID (ver más adelante el punto Paquetes) bajo esa secuencia repetidamente, a una velocidad de 3600 Saltos de frecuencia por segundo con el objetivo que sea mayor la probabilidad de que otra unidad lo escuche. El paquete ID es lo suficientemente pequeño (en duración) como para caber en una ranura de tiempo de la mitad del tamaño, (como en este caso) y contiene el Código de Acceso de Indagación (Inquiry Access Code - GIAC). El código de acceso GIAC es una clave genérica para ser reconocida por todos los dispositivos indagados. Existen otros valores de códigos de acceso reservados por la especificación que hacen que solo algunas clases de dispositivos respondan al Inquiry.

2. Por otro lado los otros dos dispositivos pueden tener programados sus temporizadores para revisar el canal en busca de señales de indagación desde otro dispositivo. Estos temporizadores son:
   - Tiempo entre búsquedas consecutivas de indagaciones $T_{inquiryScan}$
   - Ventana de búsqueda de indagaciones $T_{WinquiryScan}$ (se mide en cantidad de ranuras de tiempo).

   Los dos dispositivos del ejemplo en este estado recibirán el paquete ID enviado por el dispositivo indagador[5] en algún momento; la probabilidad de recibirlo esta dada por el establecimiento de los temporizadores ya mencionados.

[4] Dirección Física del dispositivo Bluetooth, dirección de 48 bits similar a la dirección MAC de las tarjetas de red. Es única para cada dispositivo.

[5] El dispositivo Indagador aun no se convierte en maestro

3. Al llegar el paquete, el correlador[6] del procesador de banda base detecta el código genérico de acceso de indagación (GIAC), (ó en su defecto alguno de los códigos reservados) y sabe que debe dar una respuesta a dicha indagación. Cabe la posibilidad que los dos dispositivos reciban el paquete y respondan al mismo tiempo, (dado que ninguno sabe de la existencia de los otros dos dispositivos) lo cual puede ocasionar una colisión de las señales de radiofrecuencia de respuesta; para solucionar esto, el procesador de banda base de las unidades espera una cantidad aleatoria de ranuras de tiempo y regresa a escuchar. Al primer paquete ID que se reconozca, la unidad responderá sin haber peligro de colisión.

4. El paquete de respuesta de cada unidad indagada es un paquete tipo FHS (ver más adelante el punto Paquetes) y contiene la dirección física propia del dispositivo Bluetooth y el valor instantáneo de su propio reloj; esta información puede ser utilizada posteriormente para entablar una comunicación, por medio del proceso de paginación.

3.2.2.5 Proceso de Paginación (Page)

Mediante este proceso, un dispositivo Bluetooth, teniendo conocimiento de las direcciones físicas y de los relojes de los otros dispositivos, puede establecer una conexión con uno de ellos. Por ejemplo, donde se tiene un computador portátil y una impresora equipada con un puerto Bluetooth. El computador portátil encontrará por medio de una indagación a la impresora y de esta manera en el momento en que él necesite realizar una impresión, establecerá una conexión con la impresora y finalmente transmitirá los datos necesarios (de lo cual se encargarán las capas superiores de la pila de protocolos). Este es un caso típico de paginación y ocurre de forma similar al Inquiry, pero con la diferencia que una paginación está dirigida hacia un dispositivo en especial. Para el ejemplo descrito anteriormente, el proceso se lleva a cabo de la siguiente manera:

1. El Computador, conoce ya la dirección física Bluetooth de la impresora, y por lo tanto la utiliza como dirección de entrada al módulo de selección de frecuencia, como también conoce el valor del reloj del otro dispositivo, puede usar estos datos para obtener la secuencia de saltos de frecuencia de paginación (ver el punto 3.2.2.2). Con la dirección Bluetooth de la impresora puede obtener su Código de Acceso de Dispositivo Bluetooth (DAC).

---

[6] El correlador es un registro de corrimiento recibe la corriente de bits provenientes de la capa de radio y lo compara a cada instante con el código de acceso que se espera recibir

2. Después de haber obtenido la secuencia de saltos a la cual se va a realizar la paginación, el computador empieza a enviar paquetes ID conteniendo el Código de Acceso al Dispositivo (DAC) ya conocido. La impresora por su parte según la programación que tenga en sus registros, revisa constantemente la secuencia de saltos de frecuencia de paginación (obtenida de la misma forma que lo hizo el computador usando la dirección y el reloj de la impresora) con el fin de recibir los paquetes que coincidan con su Código de Acceso a Dispositivo. El computador enviará paquetes ID a una velocidad de 3600 veces por segundo para aumentar la probabilidad de que la impresora lo detecte.

3. La impresora recibe paquetes sintonizándose a la secuencia de saltos de frecuencia de paginación y una vez recibido un paquete, el correlador de banda base revisa si el código de acceso contenido en él corresponde a su propio código de acceso, en cuyo caso la impresora responderá con un paquete ID utilizando la secuencia de saltos de frecuencia de respuesta a la paginación. El paquete ID (ver punto 3.2.4 Paquetes) no contiene Información útil, pero la información del instante de llegada al computador respecto a las ranuras de tiempo (625μs) se utiliza para sincronizar a la impresora con la misma fase de la secuencia de saltos que tiene el computador.

4. El computador, una vez que recibe la respuesta de la impresora envía un paquete FHS utilizando la misma secuencia de saltos de paginación. El paquete FHS contiene la información necesaria para que la impresora pueda participar en la Picored que se va a formar. Esta información es: el valor instantáneo del reloj nativo del computador, su dirección física, bits de paridad y la clase de dispositivo (ver punto 3.2.4 Paquetes).

5. Por último la impresora responde un paquete ID que contiene su DAC y utilizando la secuencia de saltos de frecuencia de respuesta a paginación; de esta manera confirma la sincronización al computador. Con los valores recibidos en el paquete FHS procedente del computador (ahora ya con el papel de maestro) la impresora (siendo automáticamente esclavo de la Picored) conoce a que secuencia de saltos de frecuencia debe sintonizarse y estará lista para recibir datos del computador según la aplicación. El esclavo avisa a su correspondiente nivel de *link manager* este estado. Queda entonces formada la Picored entre dos dispositivos, e igualmente el maestro de esta Picored puede seguir invitando más esclavos a ella, como por ejemplo, otros periféricos del computador.

3.2.2.6 Formación de redes compuestas ó Scatternets

Un dispositivo puede estar vinculado a una Picored, y formar espontáneamente otras con otros dispositivos, formándose así una scatternet. Esto se ilustra en la figura 3.4, suponiendo el caso en que el dispositivo D es el maestro de la red uno, otro de los esclavos puede formar una Picored, pero hay que tener en cuenta principalmente los siguientes aspectos:

- Un dispositivo no puede participar en dos Picoredes al mismo tiempo, puesto que cada una tiene canales de transmisión (patrones de salto de frecuencia) distintos y esto implica que al participar en una de ellas deberá adecuar su procesador de banda base para el canal y sincronización de dicha red. Por consiguiente, el dispositivo B de la figura tendrá que alternar estos dos comportamientos.

- Dado que el patrón de saltos de frecuencia de toda la Picored se determina de la dirección física del maestro, entonces un dispositivo no puede ser maestro en más de una Picored. Suponiendo el caso en que el dispositivo B participa en ambas redes como maestro, significa que las redes 1 y 2 utilizarían el mismo canal (igual patrón de saltos de frecuencia), siendo en realidad una sola Picored y no una scatternet. Para su estudio, cada red podría en principio ser tratada independientemente, pero teniendo en cuenta que los elementos compartidos deben multiplexar su comportamiento en ambas redes.

- Un dispositivo puede ser esclavo en una red y maestro en otra

- Un maestro, por las restricciones descritas anteriormente, a veces tiene que realizar un intercambio de papel con uno de sus esclavos, es decir, ceder el papel de maestro para poder vincularse a otra red ó para abandonar toda actividad. El esclavo puede ser también quien solicite el cambio. Al finalizar el intercambio, los dispositivos han intercambiado sus papeles, se ha cambiado el canal de la Picored (puesto que hay un nuevo maestro), pero hay información que no se ha transferido aún, y tiene que ver con el intercambio lógico del papel de maestro; el maestro debe comunicar la información que poseía sobre los demás esclavos, pero esto es algo de lo cual deben ocuparse las capas superiores de la pila de protocolos, como la capa de aplicación por ejemplo.

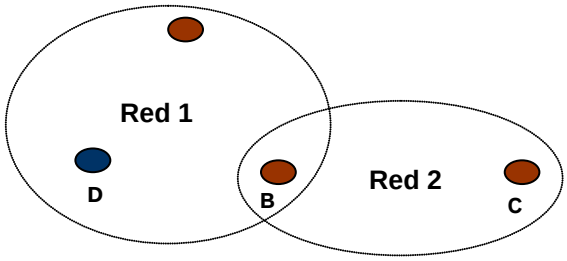


Figura 3.4. Red compuesta ó scatternet

3.2.3 Control de Acceso al Canal

En la Figura 3.3 se observó cómo se encontraba distribuido el acceso al canal, por medio de las técnicas TDMA y TDD, durante la operación normal de la Picored cada esclavo tendrá entonces ranuras pares en las cuales podrá recibir regularmente paquetes desde el maestro, y también esperará a la ranura siguiente para poder enviar paquetes al maestro.

3.2.3.1 Tipos de Enlaces

Los enlaces que se pueden formar entre maestro y esclavos son de dos tipos: Síncronos orientados a conexión (SCO[7]) y asíncronos no orientados a conexión (ACL[8]).

3.2.3.1.1 Enlaces Síncronos de Voz

Son llamados enlaces SCO. Los enlaces síncronos usados en Bluetooth, se utilizan para transportar voz con calidad telefónica (utilizada a través de enlaces de 64Kbps) ó datos en forma síncrona, es decir, requieren un flujo constante de información, a una velocidad determinada. También se asume para enlaces de voz que la información no se retransmite en caso de pérdida, puesto que debe llegar en tiempo real al destino y la pérdida de una sola muestra no sería significativa en la señal de audio reconstruida en el destino; por ello los enlaces síncronos de voz deben cumplir las siguientes características:

- Transmisión por el medio a intervalos regulares de tiempo para garantizar así una velocidad continua de información.
- No retransmisión de los paquetes que no lleguen a su destino.
- Prioridad alta para poder obedecer a una tasa de información fija.

---

[7] Synchronous Connection - Oriented

[8] Asynchronous Connectionless

3.2.3.1.1.1 Procesamiento de Audio en Banda Base

La interfaz de radio y banda base de Bluetooth ofrecen soporte para transporte de audio a 64Kbps. La información de audio que se transporte por estos enlaces deberá obedecer a dos esquemas: Lineal CVSD[9] ó logarítmico de 8 bits (aplicando la compresión μ ó A); estos parámetros deben ser negociados por las capas de *Link Manager* (ver punto 2 de este capitulo) de los dos dispositivos, a pesar de que las señales de audio si llegan directamente a la banda base. Un enlace Bluetooth puede llevar consigo 3 canales de audio dúplex (ver punto 3.2.4 Paquetes "Paquetes HV3").

En caso de que una aplicación requiera audio de mayor calidad, se deberá hacer uso de otros enlaces y transportar el audio a su destino pero siendo tratado como datos.

3.2.3.1.2 Enlaces Asíncronos

Llamados Enlaces ACL. Estos enlaces se utilizan para el transporte de datos, utilizan paquetes asíncronos y deben ser confirmados a su llegada, lo cual garantiza la integridad de la información transferida. No son orientados a conexión, pueden llegar sin necesidad de que haya una acuerdo previo de periodicidad, como ocurre en los enlaces SCO.

Los enlaces pueden ser asimétricos, alcanzando a llegar a velocidades de 723.2Kbps en una dirección y a 57.6Kbps en la otra. El carácter asíncrono de estos enlaces caracteriza sus paquetes, estos pueden transmitirse en cualquier momento (bajo las restricciones de ranuras de tiempo descritas anteriormente) y a diferencia de los enlaces síncronos pueden tener una duración variable, lo cual les permite flexibilidad en la carga útil que transportan. Para mayor referencia en este sentido, véase el punto 3.2.4 Paquetes.

3.2.4 Paquetes en Banda Base

3.2.4.1 Estructura Genérica de un Paquete

Los paquetes ó PDUs que se generan en el nivel de banda base tienen una estructura genérica que les es útil dependiendo de la aplicación y los esquemas con que trabajen. Los campos se describen a continuación.

[9] Continuous Variable Slope Delta Modulation

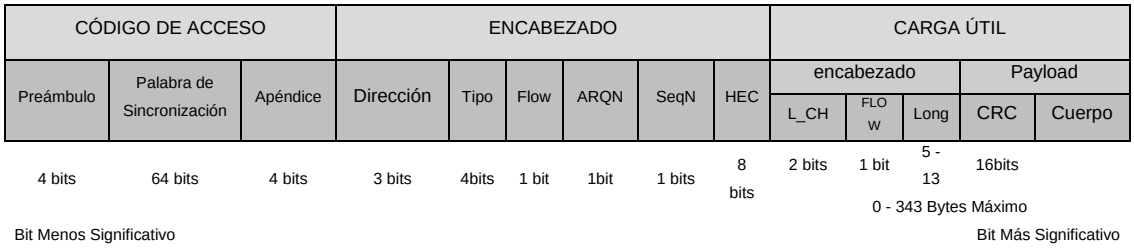


Figura 3.5. Estructura de los paquetes generados por la banda base

Los campos que comprenden los paquetes tienen diferentes funcionalidades, y puede observarse esto al tiempo con las funciones de Indagación, Paginación, formación de Picoredes y enlaces de voz y de datos.

Tabla 3.2. Descripción de los campos de un paquete en el nivel banda base Bluetooth

Campo	**Descripción**
Preámbulo	Sirve para compensar en DC el tren de bits que saldrá al aire. Puede ser 1010 ó 0101 dependiendo del primer bit de la palabra de sincronización.
Palabra de sincronización	Es la palabra que se espera como código de acceso. Puede ser de tres tipos y se generan a partir de segmentos de direcciones de 24 bits. Tiene la función de identificar los paquetes que tienen lugar en la Picored. Su detección en el receptor se hace por medio de un correlador alimentado con la dirección que se supone generó el código. La palabra puede ser de varios tipos: • Código de Acceso al Canal (CAC) – se obtiene de los 24 bits menos significativos de la dirección física Bluetooth del maestro. • Código de Acceso de Indagación - se obtiene de 24 bits predeterminados, reservados por la especificación para tal fin. • Código de Acceso al Dispositivo – se obtiene de los 24 bits menos significativos de la dirección física Bluetooth del dispositivo (esclavo).
Apéndice	Es una palabra para compensar en DC el tren de bits resultante y para agregar sincronización. Puede ser 1010 ó 0101 dependiendo del último bit de la palabra de sincronización.

Dirección	AM_ADDRESS, es la dirección de miembro activo, identifica a cada esclavo dentro de la Picored, como es de tres bits, significa que solamente se podrán direccionar hasta 7 dispositivos esclavos. La dirección 000 esta reservada para paquetes broadcast.
Tipo	Identifica el tipo de paquete, refiérase para esto a la Tabla 3.3 -Tipos de Paquetes. De acuerdo al tipo de paquete el procesador de banda base sabrá cómo deberá ser tratado el paquete.
Flow	Bandera que sirve para llevar un control de flujo de los paquetes; si el destino tiene sus buffers de recepción llenos de información sin procesar, este deberá indicar LOW = “STOP” = 0 en un paquete de respuesta, de tal manera que el transmisor baje la tasa de transferencia a un nivel más manejable por el receptor. Esto solo concierne a paquetes asíncronos. Cuando el receptor esté listo responderá con un paquete cuya bandera FLOW sea igual a “GO”, es decir, un valor de 1.
ARQN	Bandera “ACK” que sirve como confirmación al transmisor de que se recibió un paquete exitosamente en el receptor después de haber pasado la inspección del campo CRC del cuerpo de datos de usuario contenido en el paquete. Una inspección exitosa del campo CRC indicará un “ACK”, por el contrario, una inspección no exitosa del CRC devolverá un “NAK” ó ACK negativo.
SeqN	Numero de Secuencia. En un paquete que contenga datos protegidos por un CRC, esta bandera se invierte con cada nuevo paquete que se envía, permitiendo distinguir entre retransmisiones de un mismo paquete.
HEC	Header Error Check. Palabra de 8 bits generada a partir de toda la información contenida en el encabezado (10 bits), sirve para verificar la integridad de la información del encabezado, de ahí su importancia. El receptor y transmisor deben inicializar la circuitería del HEC con una dirección (la del maestro) para poder determinar si la información es correcta.
L_CH	Son dos bits que indican la existencia de información de control para la capa de link manager ó para L2CAP.
Flow	Sirve para realizar control de flujo (como se describió anteriormente en la bandera FLOW) pero a nivel de los mensajes de L2CAP.

Longitud	Longitud del payload en bytes. Es de 5 bits para cargas pequeñas (paquetes de una sola ranura de tiempo) y es de 9 bits para paquetes grandes de múltiples ranuras; en este ultimo caso, se extiende colocando 4 bits de reserva al final, quedando de 13 bits.
CRC	Chequeo de redundancia Cíclica - Cyclic Redundancy Check, es un código de 16 bits que resulta del procesamiento de toda la información del payload; Sirve para verificar la integridad de la información, la generación de esta palabra es similar al HEC; la información (payload) pasa a través de 16 registros de corrimiento realimentados; la salida de dichos registros al terminar de pasar todo el payload es la palabra CRC. En el receptor debe haber circuiteria que genere esta palabra de la misma manera y la compare con la CRC que llego, si las dos son iguales, el paquete es válido, si no son iguales, el paquete será descartado.
Cuerpo	Es la información de usuario propiamente dicha contenida en el paquete, para mayor información véase la Tabla 3.3 -Tipos de Paquetes.

3.2.4.2 Tipos de Paquetes

En los puntos anteriores se mencionaban algunos paquetes que eran útiles para procesos de formación de Picoredes y para establecer enlaces, aquí se verán cuales son los tipos de paquetes y sus posibilidades de uso.

Son 15 tipos de paquetes (4 bits), todos los paquetes están diferenciados por un código univoco que se utiliza en el campo “Tipo” descrito anteriormente. De todos los paquetes, hay 5 de control y los demás llevan información. Los que levan información pueden ser síncronos ó asíncronos, y de ello depende el tratamiento que se les dé al ser transmitidos ó recibidos.

Ya se había visto que los paquetes se transmitían en intervalos de tiempo fijos en las ranuras de tiempo ó *Slots* correspondientes (de 625μ segundos). Por simplicidad puede asumirse que un paquete se transmite en un solo slot; pero los paquetes en realidad pueden ser más grandes y ocupar más de un slot, permitiendo así que lleven más información. Para que esta ventaja no perturbe el esquema de las ranuras pares e impares de la Picored (que se describió en el punto 3.2.2), entonces los paquetes solamente podrán ser de 1, 3 ó 5 ranuras de tiempo. La Figura 3.6 puede aclarar más aún este punto. El maestro y el esclavo 1 tienen una conexión asimétrica, en la que se están utilizando paquetes asíncronos y hay uno de ellos que ocupa 3 ranuras de tiempo. Debe observarse que la sincronización de las ranuras pares e impares se conserva, puesto que después de transmitir el paquete seguirá

una ranura par que podrá ser utilizada por un esclavo para responder al maestro. También la frecuencia en la que se transmitirá el paquete será siempre la misma y es la frecuencia que le correspondería a la primera ranura de tiempo que ocupó el paquete. Al terminar la transferencia del paquete, la secuencia sigue en forma normal sin interrumpirse.

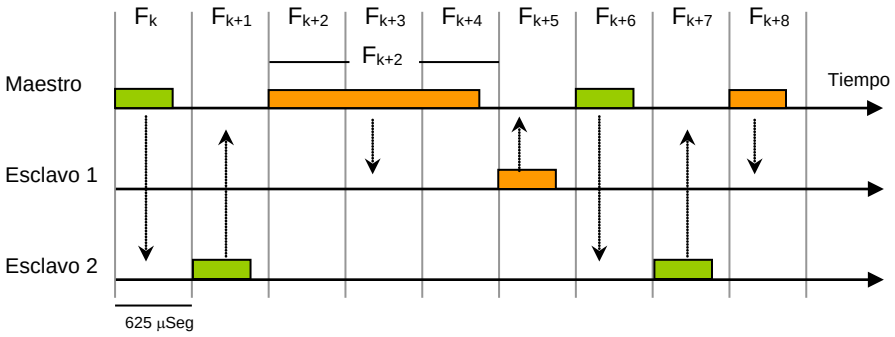


Figura 3.6. Paquetes de más de un slot

Tabla 3.3. Tipos de paquetes

Paquetes síncronos de control	1 Slot	ID	Es el paquete más sencillo de todos, pues contiene el DAC, y se utiliza en procesos de Indagación, de Paginación, y en las respuestas de los esclavos a estos procesos. Ocupa los dos primeros campos del código de acceso: preámbulo y palabra de sincronización.
		NULL	Paquete de Control que solo contiene el código de acceso al canal y el encabezado; su función es dar información de la recepción exitosa de un paquete, haciendo uso de los bits FLOW y ARQN.
		POLL	Es un paquete de la misma forma que el paquete NULL, pero este paquete requiere una confirmación de llegada por parte del receptor. Un maestro puede utilizarla para obligar a los esclavos a que respondan, independiente que tengan ó no información para enviar.
		FHS	Contiene en su payload información de sincronización, como el valor instantáneo del reloj, dirección de dispositivo activo (3 bits), la clase de dispositivo y otra información de control. No es un paquete Broadcast pero su campo de dirección destino es 000.

		DM1	Paquete de velocidad de datos media de un slot - Data Medium Rate. Aunque es un paquete de control, puede llevar información de usuario. Puede interrumpir información síncrona para transferir información de control. Está protegido por un CRC y codificado con un FEC[10] de 2/3, es decir, de 3 bits, hay 2 de información y uno para corregir errores.		
Asíncronos	Paquetes utilizados para datos, tienen diferentes capacidades y tamaños en ranuras de tiempo. Todos menos el AUX1 tienen palabra de chequeo CRC de 16 bits.				
	1 Slot		Payload	Encabezado de payload	Tipo FEC
		AUX1	29 Bytes	1 Byte	-
		DH1	27 Bytes	1 Byte	-
	3 Slots	DM3	121 Bytes	2 Bytes (ver punto 3.2.4.1)	2/3
		DH3	183 Bytes	2 Bytes	-
	5 Slots	DM5	224 Bytes	2 Bytes	2/3
		DH5	339 Bytes	2 Bytes	-
Síncronos	1 Slot	HV1	Paquete de Voz de Calidad de un slot - “High Quality Voice”. Es un paquete para aplicaciones de voz, puede llevar como carga 10 muestras de voz, lo cual equivale a llevar 1.25m s de una conversación a 64Kbps. Esto que indica que para conservar esta velocidad, el paquete debe ser transmitido cada 1.25ms, es decir, cada dos ranuras de tiempo. Los 10 bytes están protegidos por un FEC de 1/3 (es decir, por cada 3 bytes, 1 es de información y 2 son para corregir errores); no utilizan CRC.		
		HV2	Lleva 20 Bytes de información, es decir 2.5ms de conversación, lo cual significa que podrá enviarse cada 4 slots para conserva la tasa de 64Kbps. La información está protegida con un FEC de 2/3 (de tres bits de datos, dos de información y uno para corrección de errores). no utilizan CRC.		
		HV3	Lleva 30 Bytes de Información, es decir 3.75ms de conversación a 64Kbps, su intervalo de envío debe ser de 6 ranuras de tiempo. Los datos no están protegidos por FEC. no utilizan CRC.		
		DV	Paquete que combina voz y datos, existe un campo de voz de 80 Bytes (al igual que el HV1, 2.5ms), el cual no esta protegido por FEC y un campo de datos de 150 bits de longitud con encabezado y protección por CRC.		

Hay que tener en cuenta que los enlaces de voz deben ser simétricos, es decir, el mismo tipo de paquete que hay en un sentido, existe de regreso. También estos paquetes al tener

[10] Forward Error Correction – Esquema de Corrección de Errores Hacia Adelante

tiempos fijos de transferencia, dan como resultado tasas de velocidad constante, lo cual hace que las conexiones sean de 64Kbps (en el enlace de datos del paquete DV se tendrá una tasa de 57.6Kbps).

Para los paquetes asíncronos se puede obtener una velocidad de transferencia de información bastante alta, puesto que los enlaces pueden ser asimétricos y los paquetes tienen gran capacidad. Por ejemplo, asumiendo que se van a utilizar paquetes DH5 para una aplicación, la mínima velocidad simétrica (maestro y esclavo transmitiendo el mismo paquete DH5) de transferencia seria:

Vel = Cantidad de información (bits) / periodo entre transferencias contiguas
Vel = (339 bytes de payload) * 8 bits / (10 Ranuras de tiempo * 625μs)

Velocidad máxima = 433.92Kbps
Esta velocidad es en el sentido Maestro – esclavo ó viceversa.

Para calcular la tasa de bits en forma asimétrica máxima, hay que tener en cuenta algunas consideraciones. Suponiendo que se hará una conexión asimétrica con la máxima velocidad permitida en el sentido maestro – esclavo y una conexión de regreso (esclavo - maestro) lo mejor posible.

- Se transmiten paquetes DH5 constantemente desde el maestro hacia el esclavo.
- El esclavo enviará por consiguiente paquetes asíncronos de un solo slot, ó sea DH1.
- Bajo estas condiciones se obtendrá la máxima velocidad de bits asimétrica, puesto que habrá 5 ranuras para transmitir y una para recibir. Hay que considerar que el paquete DH5 se trasmitirá como máximo cada 6 ranuras de tiempo, y el DH1 de regreso se transmitirá también cada 6 ranuras.

Las velocidades quedan entonces:

$V_{Maestro\text{-}Esclavo}$ = (339 Bytes de Payload) * 8 Bits / (6 Ranuras de tiempo * 625μs)
$V_{Maestro\text{-}Esclavo}$ = 723.2 Kbps

$V_{Esclavo\text{-}Maestro}$ = (27 Bytes de Payload) * 8 Bits / (6 Ranuras de tiempo * 625μs)
$V_{Esclavo\text{-}Maestro}$ = 57.6 Kbps

La primera es la velocidad máxima asimétrica que puede lograrse, utilizando un método similar se pueden obtener las distintas velocidades de paquetes que se requieran para otras clases de enlaces.

3.2.5 Canales Lógicos

Teniendo un medio de transporte común en la Picored, es decir, un canal físico, se pueden montar canales lógicos soportados en él. Estos canales transportan información por medio de los paquetes ya conocidos, pero esta información puede ya diferenciarse entre información de control e información de usuario. Entre la información de control se encuentran los mensajes procedentes del nivel de Link Manager local de cualquiera de los dispositivos, tendientes a ser llevados al *Link Manager* del otro dispositivo con el fin de llevar a cabo alguna transacción. Los canales lógicos son los siguientes:

- Control de Enlace (*Link Control - LC)*. Se transporta en el encabezado del paquete (al cambiar la información de las banderas ARQN y FLOW), también define información en el payload.
- Manejador de enlace (*Link Manager - LM)*. Utiliza paquetes DM (Contienen FEC) y se distinguen por definir el campo L_CH del encabezado del payload en un valor determinado.
- Datos Asíncronos / Isócronos de usuario (*User Asínchronous / Isochronous Data* - UA / UI). Llevan información transparente procedente de la capa de L2CAP, la cual puede fragmentarse.
- US - *Datos Síncronos de Usuario*. Se transportan sobre paquetes SCO.

3.2.6 Características de Bajo Consumo de Energía

El nivel de banda base tiene a su cargo la ejecución de procedimientos que tengan como objetivo reducir al máximo el consumo de energía en el dispositivo. Estos procedimientos están relacionados con:

- Un dispositivo que reciba un paquete, revisa el encabezado verificando que su HEC sea válido, si esta condición no se cumple, el paquete debe quedar descartado, evitando todo procesamiento posterior del mismo; regresando de inmediato al estado de espera, en el cual consumirá menos energía.

- Sólo se envía información que sea necesaria, por ejemplo, el ACK negativo se obvia cuando ningún ACK ha llegado, lo cual hace que el receptor no esté obligado a responder un NAK a la fuente.
- Cuando un paquete es recibido por un esclavo y no está dirigido a él por la revisión del campo de dirección, entonces el esclavo puede retornar al estado de espera hasta que termine la duración del paquete.
- Un dispositivo Bluetooth está conectado a un módulo Bluetooth que agrupa los niveles de procesamiento de Radio, banda base y link manager; todo se comunica a través de una Interfaz de Equipo Controlador (Host Controller Interface – HCI). El módulo Bluetooth concentra dichos niveles y esta característica hace que en total el procesamiento Bluetooth consuma poca energía.
- Los módulos Bluetooth tienen varios modos de ahorro de energía que pueden ser programados de acuerdo a los valores de algunos registros del nivel banda base, los cuales regularmente son temporizadores.

Existen 3 modos de ahorro de energía que se diferencian entre si por la forma en la cual administran esta característica. Estos modos son Sniff, Hold y Park. Cada uno tiene un nivel distinto de consumo de energía, a pesar de ser relativamente bajo en general.

Mas Energía Consumida		Menos Energía Consumida
Sniff	Hold	Park

3.2.6.1 Estado Sniff

En este estado, el esclavo y maestro pueden ponerse de acuerdo para transmitir y recibir solo en ranuras de tiempo determinadas, lo cual resulta ventajoso para el ahorro de energía, puesto que el dispositivo solamente estará listo para recibir y procesar paquetes en dichas ranuras de tiempo, quedando inactivo el resto del tiempo. Estos intervalos de tiempo se negocian entre maestro y esclavo; la negociación la realizan las capas de *Link Manager* de cada dispositivo, cualquiera de los dos puede solicitar que se pase a este estado. Entre los parámetros que se negocian están:

- El intervalo Sniff
- El número de intentos en los cuales se dejará de escuchar el canal.
- El numero de Slots a esperar desde que ocurra una transmisión desde el maestro, después de los cuales el esclavo dejara de escuchar el canal.

Las comunicaciones entonces son periódicas, los enlaces SCO pueden seguir llegando normalmente, la carga que en realidad se disminuye es la de paquetes ACL (que eventualmente podrían llegar en cualquier momento), puesto que el Maestro solamente los enviará en los intervalos permitidos. Quiere decir que el ahorro de energía que habría en un dispositivo que ya posee enlaces SCO establecidos, se reduce a la ganancia en procesamiento de paquetes ACL. El dispositivo conserva su dirección de dispositivo activo dentro de la Picored.

3.2.6.2 Estado Hold

En el estado Hold ó de no actividad las comunicaciones no son periódicas a diferencia del estado Sniff, el dispositivo se mantiene durante un cierto tiempo en estado Hold sin importar si llega información ó no. Los enlaces SCO no se detienen, el tráfico ACL es el que se detiene por completo durante un intervalo Hold. Tanto los esclavos como los maestros pueden solicitar unos a otros entrar en este modo, el argumento que se negocia es el tiempo de Hold. Esta negociación se haría entre los niveles de Link Manager de los dos dispositivos. El dispositivo conserva su dirección de dispositivo activo dentro de la Picored.

3.2.6.3 Estado Park

El modo de Parqueo es el que más energía ahorra de los tres. En él, el dispositivo deja de participar en la Picored, queda en un estado de actividad mínima, pero no abandona por completo la receptividad a eventos que ocurran en la Picored. El podrá volver a ser miembro de la Picored sin necesidad de que se realice de nuevo una indagación ó una paginación.

1. El Maestro puede obligar al esclavo ó el esclavo también puede solicitar entrar en el estado Hold por medio de una negociación entre sus niveles *Link Manager*.
2. El esclavo en el modo de parqueo cede su dirección de dispositivo activo de 3 bits ó AM_ADDR.
3. El maestro concede al esclavo dos direcciones de 8 bits que le servirán para poder regresar a la Picored. Estas direcciones son: la dirección de dispositivo parqueado (PM_ADDR) y la dirección de solicitud de acceso (AR_ADDR).
4. En cualquier momento el Maestro puede solicitar al esclavo su reintegro a la Picored, y para ello puede hacer uso de la dirección PM_ADDR que se le entregó previamente. También podrá utilizar la dirección física del dispositivo Bluetooth para poder reintegrarlo a la Picored.
5. En cualquier momento, el esclavo podrá salir del estado de parqueo solicitándolo al maestro, y utilizando la dirección AR_ADDR que le fue dada previamente.

6. Existe un canal de “faro” ó *Beacon Channel* que sirve para que el Maestro envíe constantemente paquetes Broadcast, cuyo contenido puede ser la dirección de un dispositivo con el fin de despertarlo ó bien el contenido puede ser tal que todos los dispositivos respondan de forma ordenada de acuerdo a su dirección AR_ADDR. El Canal Faro está definido por intervalos de tiempo regulares. No es otra secuencia de saltos de frecuencia, sino que es un canal lógico, que funciona sobre el canal físico de la Picored.

3.2.7 Seguridad en el Nivel de Banda Base

Bluetooth ha definido procedimientos para garantizar cierto nivel de seguridad de las comunicaciones entre dispositivos. Estos procedimientos están relacionados con la autenticación y la encriptación, y definen 4 entidades que tienen su papel en estos procedimientos:

- Dirección Bluetooth del dispositivo (BD_ADDR) de 48 bits.
- Clave privada de autenticación de 128 bits.
- Clave privada de encriptación de longitud variable de 8 a 128 bits.
- Número Aleatorio de 128 bits, que se garantiza, será diferente cada vez que se obtenga dentro de un periodo largo de tiempo.

El procedimiento para garantizar la seguridad en la transferencia de datos entre dos dispositivos, es como sigue:

1. Con la Dirección BD_ADDR, un número PIN[11] de longitud variable y otro número aleatorio se obtiene una clave de inicialización K_{init}. El número PIN se puede obtener del usuario, por medio de la aplicación que tenga el dispositivo.
2. La clave de inicialización es utilizada para transportar la clave de unidad, que podrá ser utilizada como clave de enlace en ambos dispositivos.
3. Si las capacidades de procesamiento y almacenamiento de los dispositivos lo soportan, se puede generar una clave de combinación a partir de las claves de unidad de los dos dispositivos, lo cual incrementa la seguridad. Esta clave de combinación (si se llega a generar) también podrá ser clave de enlace.

[11] Personal Identification Number

4. Cualquiera que haya sido la clave que definida como clave de enlace, se intercambia entre los dispositivos y dicha clave será la que se utilice para la autenticación.
5. La clave de encriptación se obtiene a partir de la clave de enlace que esté activa en ese momento.

3.2.7.1 Autenticación

El esquema de autenticación puede describirse por medio de la Figura 3.7; en ella dos dispositivos antes de establecer una comunicación, se autentican y saben si son dispositivos autorizados para acceder a la información. La autenticación puede ser de acuerdo a la aplicación, en ambos sentidos ó en uno solo. Nótese que la clave de enlace debe existir previamente. El número aleatorio se combina mediante la circuitería E1 (detallada en la especificación) y este resultado se envía al otro dispositivo. El resultado de la combinación en B se retorna y al ser comparado con el obtenido localmente se determina la autenticidad del dispositivo B. Como resultado paralelo se obtiene un número Offset que se utilizará como uno de los parámetros para la consecuente encriptación de la información.

3.2.7.2 Encriptación

La encriptación se lleva a cabo como un procedimiento alterno, y que es opcional si la aplicación lo requiere; en tal caso debe activarse y generarse la clave de encriptación a partir de la clave de enlace; la encriptación se lleva a cabo en ambos sentidos de la comunicación y se aplica al payload de los paquetes. En el caso de que haya transferencia punto a multipunto de maestro a esclavos, la encriptación debe realizarse por medio de una clave maestra. Estos parámetros se incluyen en los argumentos de las solicitudes para tal fin provenientes del nivel de *Link Manager*. La Figura 3.8 muestra la forma como se realiza el proceso de encriptación; son tres pasos: la generación de la clave de payload, la generación de un flujo de bits, y por ultimo la mezcla de este flujo con los datos a encriptar.

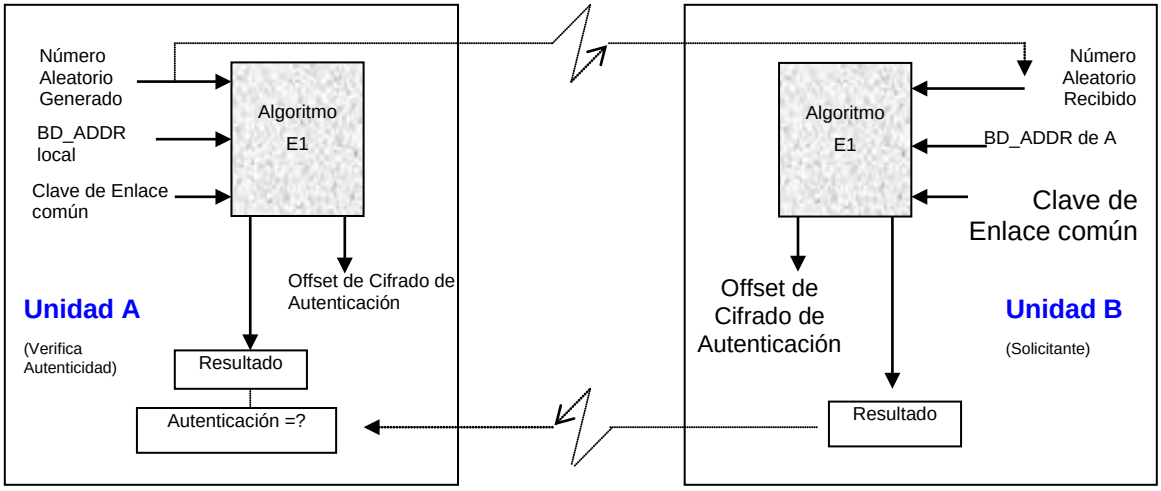


Figura 3.7. Esquema de autenticación

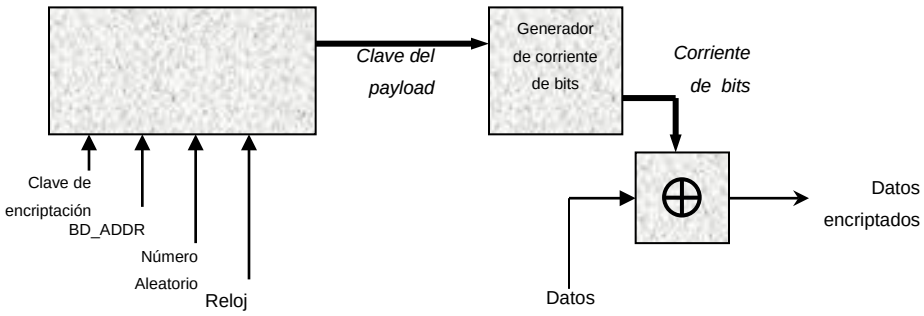


Figura 3.8. Procedimiento para la encriptación

3.3 PROTOCOLO PARA GESTIÓN DE ENLACE (LINK MANAGER PROTOCOL LMP)

3.3.1 Introducción

Las entidades del *Link Manager* o Gestionadoras del Enlace intercambian mensajes para el control del enlace entre dos dispositivos Bluetooth. Una entidad *Link Manager* (entidad LM)

puede comunicarse con la entidad LM de otro dispositivo por medio de mensajes LMP_PDU *(Link Manager Protocol_ Protocol Data Unit)* o enviando señales de control a las capas inferiores de su propio dispositivo, haciendo uso del protocolo que lleva su mismo nombre.

Los mensajes LMP_PDU tienen alta prioridad y no pueden aplazarse o atrasarse por otro tipo de tráfico; sin embargo la interacción que se presenta entre entidades LM no se realiza en tiempo real.

3.3.2 Paquetes LMP

Los paquetes utilizados por este nivel, LMP_PDU *Link Manager Protocol_ Protocol Data Unit,* se transfieren en la carga útil de un paquete ACL y se distinguen por la presencia de un valor en el campo L_CH de la cabecera de estos paquetes. Los LMP_PDU se transfieren siempre en paquetes que ocupan un solo slot como son los paquetes DM1 y en ciertas ocasiones en paquetes DV.

Figura 3.9. Formato LMP_PDU

La fuente y el destino de un paquete LM se determinan por el valor de AM_ADDR en la cabecera del paquete ACL.

Los paquetes LM pueden ser del tipo opcional u obligatorio y se emplean para realizar dos tipos de transacciones. La primera se lleva a cabo cuando una entidad LM realiza una petición a la entidad LM de otro dispositivo. El LM que recibe la petición deberá responderla diciendo si la acepta o no; en este ultimo caso expondrá también el motivo por el cual la petición se rechaza. Así mismo existe otro tipo de transacción en donde el maestro envía un comando para que sea ejecutado sin que el receptor tenga la opción de rechazarlo o modificar sus parámetros.

3.3.3 Funciones del Protocolo Link Manager

3.3.3.1 Establecimiento del Enlace

3.3.3.1.1 Conexión

Cuando un dispositivo desea comunicarse con otro y esta comunicación involucra niveles por encima del LM, se envía una petición de conexión por medio del PDU *LMP_Host_Conexión_Request*. Esta petición será recibida en el otro lado y se contestará con una respuesta de aprobación o de negación.

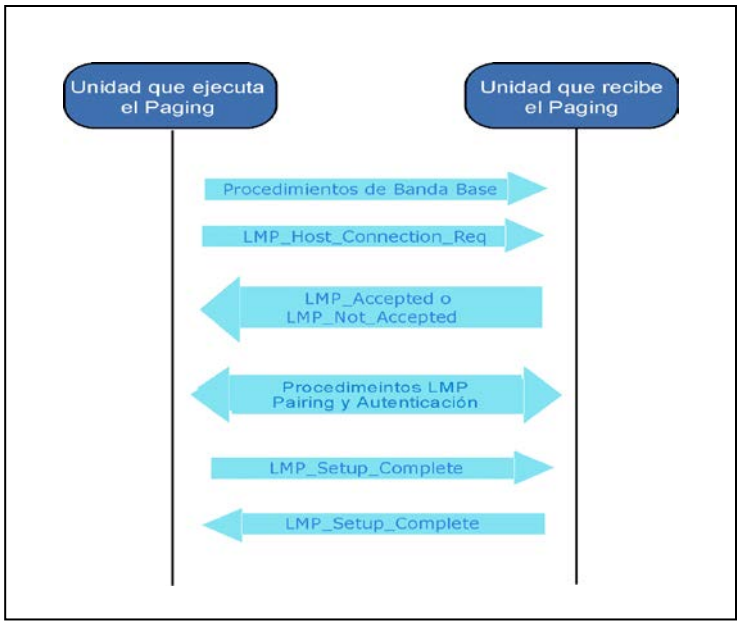


Figura 3.10. Procedimiento para establecimiento del Enlace

Si la petición se acepta se da inicio al establecimiento de otras transacciones, como son los procedimientos de seguridad o negociación de parámetros del enlace. Una vez estos procedimientos finalizan cada una de las entidades LM involucradas en la comunicación, envía un *LMP_Setup_Complete* y los PDU de otras capas pueden empezar a trasmitirse.

3.3.3.1.2 Desconexión

La conexión entre dos dispositivos puede ser finalizada en cualquier instante. Para tal efecto una de las dos partes envía a la otra un mensaje *LMP_Detach*, este mensaje puede no ser contestado, pero el enlace termina y la AM_ADDR puede ser inmediatamente reutilizada por el maestro. El mensaje de desconexión incluye la razón por la cual una de las partes desea culminar el enlace.

En caso de que el receptor nunca reciba el mensaje, se inicia un procedimiento de supervisión que concluye con la terminación del enlace. Este procedimiento se explicará más adelante.

3.3.3.1.3 Cambio en el Papel de Maestro - Esclavo

Generalmente el dispositivo que inicia el procedimiento de *PAGING*, se convierte en el maestro de una piconet. Sin embargo, en algunos casos se requiere un cambio en los papeles del maestro y el esclavo si se determina que estos no son los más apropiados para el enlace. La capa *Link Manager* proporciona mecanismos para realizar este cambio, sin afectar el funcionamiento de la piconet.

El dispositivo que desea solicitar el cambio, envía una PDU denominada *LMP_Switch_Req* que contiene el parámetro Momento de Cambio *Switch Instant*, el cual especifica el instante en el que se realiza el intercambio de slots. Este parámetro esta dado por un valor de tipo *reloj Bluetooth*, de acuerdo con el reloj del maestro. Este instante es escogido por el dispositivo que envía el mensaje y debe ser de un periodo equivalente a 2 x periodos de polling o 32 slots como mínimo. Después de enviar este mensaje, el dispositivo inicia un *Temporizador* que expira en el momento determinado por *Switch Instant*. Cuando esto ocurre se da inicio al modo de cambio de papel. Si se recibe un mensaje *LMP_Not_Accepted*, mientras el *Temporizador* aún esta corriendo, este se detiene y la operación de cambio de role no se inicia.

3.3.3.1.4 Supervisión

Cada unidad Bluetooth posee un *temporizador* que se utiliza para detectar la perdida del enlace causada porque los dispositivos, se mueven, salen del rango de cobertura o se apagan. El *Link Manager* establece el valor del tiempo de vencimiento *Timeout* para supervisión, por medio de una PDU denominada PDU_*Supervision_Time_Out*.

3.3.3.2 Gestión de Seguridad

El nivel de *Banda Base* de Bluetooth proporciona los mecanismos de encriptación y cifrado de datos característicos del esquema de seguridad de esta tecnología, sin embargo debe existir un control sobre estos procedimientos, que además este sincronizado en las unidades que establecen una conexión. El nivel *Link Manager* coordina los mecanismos para negociación de los modos y claves de encriptación, autenticación y *Pairing.*

A continuación se describen cada uno de ellos.

3.3.3.2.1 Autenticación

Como fue explicado anteriormente, el procedimiento de autenticación se basa en un esquema de petición y respuesta en donde dos partes de un enlace, *Solicitante (Claimant) y Vereficador,* (Maestro o Esclavo pueden tomar cualquiera de los papeles) intercambian información -claves en este caso-, que permiten verificar la autenticidad de un dispositivo. El *Verificador* envía al *Solicitante*, un mensaje *LMP_Au_Rand* que contiene un número aleatorio con el que el *Solicitante* calcula una respuesta *LMP_Sres* que también es función de su dirección BD_ADDR y una clave secreta que los dispositivos conocen (*LinkKey*) y la envía devuelta para que el verificador chequee si esta es correcta o no.

En caso de que la verificación resulte errónea, el verificador enviará un mensaje *LMP_Detach* con la razón “Falla en el código de Autenticación” *Code Autentication Failure* y finalizará la conexión. Si el *Solicitante,* no posee una clave de enlace, enviará al *Verificador* un mensaje *LMP_Not_Accepted*, señalándole que su clave de enlace no es aceptada.

El *Link Manager* no esta en capacidad de manejar peticiones concurrentes de autenticación, por lo tanto deberá atender cada una de estas por separado y podrá continuar con la siguiente después de recibir un *LMP_Res*.

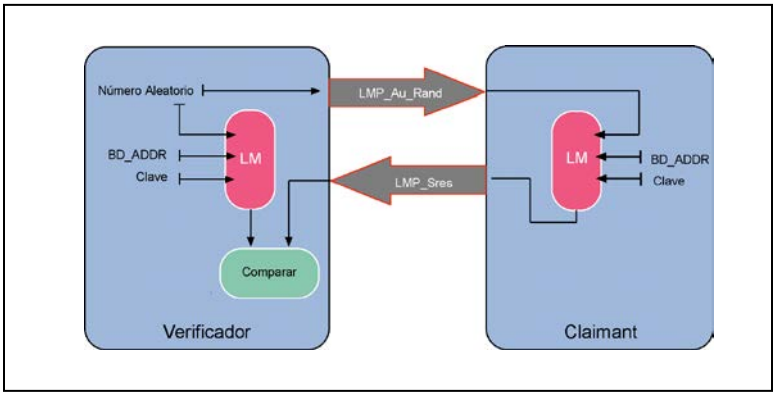


Figura 3.11. Procedimiento de autenticación entre dispositivos Bluetooth

3.3.3.2.2 PAIRING

Cuando la clave de enlace no se comparte, se genera otra clave llamada *Clave de Inicialización* con base en un PIN, la dirección BD_ADDR del dispositivo y un *número aleatorio*. La clave de Inicialización permite generar una Clave de Enlace (*LinKey*) permanente. Este procedimiento recibe el nombre de *PAIRING* y se inicia cuando uno de los dispositivos (iniciador) envía un mensaje *LMP_In_Rand* y el otro responde con un mensaje de *LMP_Accepted*. De este modo, las dos partes comprenden que deben calcular una clave de inicialización que les permita ejecutar un proceso para la creación de la Clave de Enlace (*LinKey*), que también está basada en la dirección BD_ADDR del dispositivo que hace la petición. Una vez los dispositivos comparten la clave de enlace, se puede terminar el proceso de autenticación.

Si el dispositivo que hace la petición posee un PIN fijo, generará un número aleatorio con base en este y lo devolverá al iniciador en una PDU del mismo tipo, *LMP_In_Rand*. Si el iniciador posee un PIN variable, deberá aceptar el número aleatorio que recibe y responder con un *LMP_Accepted*. En seguida el procedimiento continua con la creación de la *Clave de Inicialización*, basándose en el número aleatorio y la dirección BD_ADDR del dispositivo iniciador para que finalmente pueda crearse una Clave de Enlace (LinKey).

Si los dos dispositivos tienen un PIN fijo, se rechaza el procedimiento con un LMP_Not_Accepted y con la razón “Pairing No Allowed”.

El proceso de la creación de la clave de enlace puede tener dos resultados, una clave de enlace combinada o una de las claves de unidad de uno de los dos dispositivos (la clave de unidad se crea la primera vez que el dispositivo se enciende), este resultado depende de:

- Si una de las unidades envía su clave de unidad, se toma esta clave como clave de enlace.
- Si los dos dispositivos envían su clave de unidad, se genera una clave de enlace que es la combinación de las dos claves enviadas.

Cuando el esquema de autenticación falla, debe realizarse todo el proceso nuevamente.
Las claves de enlace pueden ser modificadas de manera permanente o temporal.

3.3.3.2.3 Encriptación

Después de realizar la *Autenticación*, los dispositivos pueden iniciar el proceso de *Encriptación*. Sin embargo, deben realizarse tres transacciones destinadas al establecimiento de los parámetros de encriptación antes de que los datos empiecen a ser transmitidos.

Ver página siguiente...

Tabla 3.4. Procedimientos para iniciar la encriptación

PROCEDIMIENTO
Negociación del Modo de Encriptación: El Link Manager emplea un *LMP_Encryption_Mode_Req* para requerir el modo de encriptación utilizado por el canal, esto es, si se trata de encriptación Punto a Punto (entre dos dispositivos) o encriptación de paquetes para Broadcast. Si el modo es aceptado se devuelve un *LMP_Accepted* y el maestro generará una clave de encriptación que se utilizará en el proceso mismo. Si el maestro recibe un *LMP_Not_Accepted* deberá realizar la petición nuevamente utilizando un modo de encriptación diferente, hasta que reciba la aprobación del esclavo.
Negociación de Tamaño de la Clave de Encriptación: Después de aceptar el modo de encriptación, los dispositivos negocian el tamaño de la clave de encriptación a ser usada. El maestro inicia el proceso enviando en una PDU la longitud máxima que soporta. Si el esclavo está en capacidad de utilizarla retornará un *LMP_Accepted*. Si no es así, el maestro deberá tratar con un tamaño menor hasta que reciba un *LMP_Accepted*.
Inicialización de la Encriptación: Si la negociación anterior tiene éxito, los dispositivos están listos para iniciar la encriptación y envían un aviso de su estado a través de un *LMP_Start_Encryption*. Una vez reciben los *LMP_Accepted*, los dispositivos pueden transmitir datos encriptados a través de un enlace ACL.

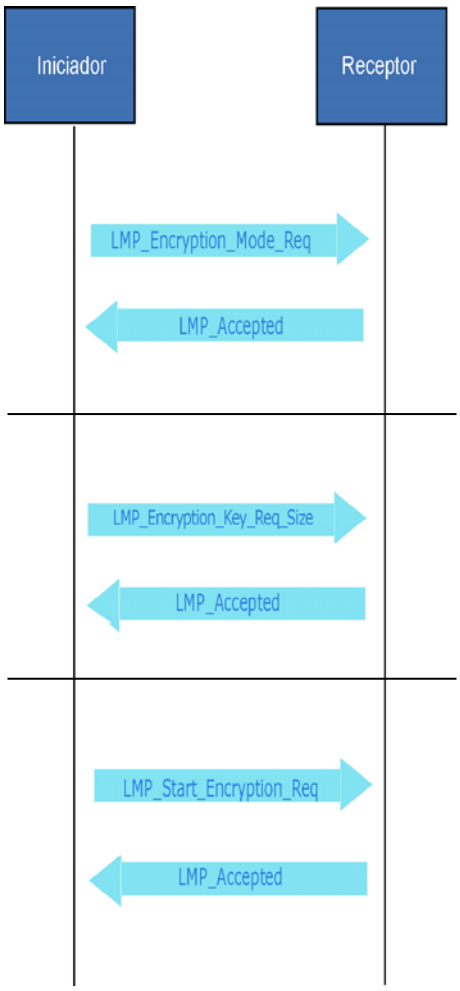


Figura 3.12. Procedimientos para iniciar la encriptación

3.3.3.3 Gestión y Control de Potencia

3.3.3.3.1 Gestión De Potencia

Debe tenerse en cuenta que el establecimiento de un enlace, puede tomar varios segundos y una ves el dispositivo alcanza el estado activo no querrá perderlo, para evitar realizar nuevamente todo el proceso de conexión. Con el fin de preservar la fuente de energía o atender otro tipo de transacción (participación en una Scatternet) los dispositivos Bluetooth pueden regular su consumo de energía modificando el tipo de asociación a una Piconet. El

Link Manager se encarga de controlar los modos de conexión con bajo consumo de energía, de la siguiente manera:

3.3.3.3.1.1 Modo Hold

Una conexión puede entrar en este modo por medio de una petición que hace el Host después de que ha recibido una solicitud previa que hace el *Link Manager* de un dispositivo remoto o porque el *Link Manager* local decide automáticamente entrar en este modo; de esta manera el *Link Manager* deberá encargarse de establecer y negociar los paramentos relacionados con este modo de consumo de energía.

Dos tipos de PDU se utilizan para establecer el modo *Hold*. *LMP_Hold* y *LMP_Hold_Req*. El primero de ellos, se utiliza en caso de que el maestro o el esclavo deseen forzar (no puede ser rechazado) al otro dispositivo a entrar en modo *Hold*. El segundo tipo se emplea para realizar una solicitud que puede ser aceptada o rechazada. Tanto el LMP_*Hold*_req como el LMP_*Hold* contiene dos tipos de parámetros:

Tabla 3.5. Formato del LMP_Hold_Req

PARÁMETROS DEL LMP_HOLD_REQ	DEFINICIÓN
Hold Time	Especifica la duración del modo Hold.
Hold Instant	Especifica el instante en que el Hold se hace efectivo, es decir, cuándo inicia.

Tanto el maestro como el esclavo pueden requerir entrar en este modo, haciendo uso del LMP_*Hold*_Req. El *Link Manager* que recibe este mensaje, tiene tres posibles opciones; aceptar la petición, a través de un *LMP_Accepted*, rechazarla, por medio de un *LMP_Not_Accepted* o enviar un *LMP_Hold_Req*, con el fin de negociar nuevamente los paramentos de la transacción. Una vez las dos capas se ponen de acuerdo en los parámetros, la negociación finaliza con un *LM_Accepted* que envían las dos unidades. En caso contrario se envía un *LMP_Not_Accepted* con su razón y la transacción finaliza.

El maestro puede forzar a la entrada de un esclavo en el modo *Hold* , una vez el esclavo haya aceptado anteriormente, una petición del maestro para entrar en este modo.
El maestro selecciona un *Hold_Instant* y envía un *LMP_Hold* con este parámetro e inicia un temporizador para esperar que el *Hold_Instant* ocurra. Cuando el temporizador expira, el

maestro ingresa al modo *Hold*. El *Link Manager* receptor compara el *Hold_Instant* con el reloj maestro e iniciará un temporizador para entrar en modo *Hold* cuando este expire.

Un Esclavo puede forzar a un maestro a entrar en este modo a través de un mensaje *LMP_Hold*, sin embargo deberá esperar que el maestro le retorne un *LMP_Hold*, después del cual, el procedimiento se desarrolla como si el maestro hubiese forzado al esclavo a entrar en este modo.

Mientras se ejecute esta transición, se suspende todo el tráfico en el dispositivo.

3.3.3.3.1.2 Modo Sniff

El protocolo *Link Manager* se utiliza para coordinar el establecimiento del modo *Sniff* en los terminales de una conexión. Para entrar en este modo el maestro o el esclavo emplean un mensaje *LMP_Sniff_Req*.

Tabla 3.6. Formato del LMP_Sniff_Req

PARÁMETROS DEL LMP_SNIFF_REQ	DEFINICIÓN
Timing Control Flags	Se emplea para calcular el primer Slot de Sniff
D_{sniff}	Especifica el instante en que el Hold se hace efectivo, es decir, cuándo inicia.
T_{sniff}	Intervalo Sniff
Sniff Attemp	Especifica la temporización para los slots de Sniff
Sniff Time Out	Duración del modo Sniff

El receptor puede iniciar la negociación de los parámetros utilizando también un *LMP_Sniff_Req* con parámetros modificados.

Si hay acuerdo en ellos, se envía un *LMP_Accepted*, de lo contrario se envía un *LMP_Not_Accepted* indicando el motivo de rechazo "*Parámetro No Soportado*".

El modo *Sniff* puede finalizar si el iniciador envía el PDU *LMP_UnSniff_Req*. El receptor deberá responder con un mensaje *LMP_Accepted*. Si es el esclavo quien hace la petición, este podrá entrar en modo activo, tan pronto reciba del maestro el *LMP_Accepted*. Si es el maestro, el esclavo entrará al modo activo después de recibir el *LMP_UnSniff_Req*.

3.3.3.3.1.3 *Modo Park*

Si un esclavo no necesita participar en el canal, pero todavía debe estar sincronizado con el maestro, puede entrar en modo *Park*. En este modo un dispositivo entrega su dirección AM_ADDR pero todavía se sincroniza con el canal, despertando en los ciertos instantes denominados instantes *faro (Beacon)*, separados un intervalo *faro*. El intervalo *Beacon*, un intervalo offset y una bandera indican como se calcula el instante *Beacon* para determinar la ocurrencia del primer *Beacon*. Después de esto, el instante *Beacon* continua periódicamente predeterminado por el intervalo *Beacon*. En instante *Beacon*, un esclavo se activa y el maestro puede cambiar los parámetros del modo *Park*, transmitir información de broadcast o permitir a los esclavos parqueados hacer peticiones de acceso al canal.

Todas los PDU que el maestro envía a los esclavos son broadcast. Estas PDU son las únicas PDU que pueden enviarse a un esclavo en modo *Park* y son las únicas PDU broadcast. Con el fin de incrementar la confiabilidad del broadcast, los paquetes serán tan pequeños como sea posible. Por lo tanto el formato de estos paquetes es un poco diferente. Los parámetros no siempre están alineados y la longitud del PDU varia.

El mensaje para controlar el modo *Park*, contiene muchos parámetros. Cuando un esclavo se pone en este modo se le asigna una única PM_ADDR, la cual es utilizada por el maestro para desparquear al esclavo. Una dirección PM_ADDR de ceros tiene un significado especial, no es una PM_ADDR válida. Si a un dispositivo se le asigna esta dirección, este debe ser identificado con su BD_ADDR, cuando sea desparqueado por el maestro.

<u>El maestro solicita a un esclavo entrar en modo Park</u>

El maestro puede solicitar el modo *Park* para un esclavo. El maestro finaliza la transmisión de los paquetes ACL, la transmisión punto a punto de paquetes ACL (si los hay) y envía un *LMP_Park_Req*. Si el esclavo acepta, entra en este modo, finaliza su transmisión L2CAP y envía un mensaje con *LMP_Accepted*. Cuando el esclavo pone en cola su *LMP_Accepted*, inicia un *temporizador* que finaliza al cabo de seis intervalos de poll (*6xTpolls slots)*. Si se recibe un ACK de Banda Base antes de que el temporizador expire, el dispositivo entra en modo *Park*, de otro modo esperará a que el temporizador expire para hacerlo. Cuando el maestro recibe un *LMP_Accepted* inicia un temporizador por *6xTpoll slots*. Cuando el temporizador expira el esclavo está en modo *Park* y el maestro puede reutilizar su

AM_ADDR. Si el maestro nunca recibe una *LMP_Accepted* inicia un procedimiento de supervisión.

Si el esclavo rechaza la solicitud para entrar en el modo *Park*, responderá con un *LMP_Not_Accepted* y el maestro reanudará su transmisión L2CAP.

El esclavo requiere entrar en el modo Park

Un esclavo puede requerir entrar en el modo *Park*. El esclavo finaliza su transmisión L2CAP y envía un *LMP_Req_Park*. Los parámetros PM_ADDR y AR_ADDR no son válidos y los otros parámetros de este paquete representan parámetros sugeridos. Si el maestro desea que el esclavo entre en este modo, finaliza sus transmisiones L2CAP y envía un *LMP_Park_Req* con parámetros que pueden ser diferentes de los sugeridos por el esclavo. Si el esclavo acepta los nuevos parámetros responderá con *LMP_Accepted*. Cuando el esclavo pone en cola el mensaje *LMP_Accepted* inicia un temporizador para *6xTpoll slots*. Si se recibe un ACK de Banda Base antes de que el temporizador expire, el dispositivo entra en modo *Park*, de otro modo esperará a que el temporizador expire para hacerlo. Cuando el maestro recibe un *LMP_Accepted* inicia un temporizador por *6xTpoll slots*. Cuando el temporizador expira el esclavo está en modo *Park* y el maestro puede reutilizar su AM_ADDR. Si el maestro no acepta que un esclavo entre en modo *Park* devolverá un *LMP_Not_Accepted*. El esclavo puede entonces reanudar su transmisión L2CAP.

Si el esclavo no acepa los parámetros del *LMP_Park_Req* enviará al esclavo un mensaje *LMP_Not_Accepted* y las dos unidades reanudarán su transmisión L2CAP.

Establecimiento del parámetro *Broadcast Scan Window*

Si se requiere más capacidad para broadcast que la dada por el tren de *Beacons*, el maestro puede indicar al esclavo que mas información broadcast seguirá al tren de *Beacons* mediante el envío de un *LMP_Set_Broadcast_Window*. Este mensaje se envía siempre en un paquete broadcast en las ranuras de tiempo (slots) *Beacon*. El escanéo de ventana inicia en el instante *Beacon* y solo es valido para el *Beacon* actual.

Modificación de los parámetros del *Instante Beacon*

Cuando los parámetros del *Beacon* cambian, el maestro lo notifica a los esclavos parqueados enviando un mensaje *LMP_Modify_Beacon*. Este mensaje se envía siempre un paquete broadcast en el slot *Beacon*.

Salir del modo Park

El maestro puede extraer o sacar del modo *Park,* a uno o varios esclavos enviando un mensaje broadcast que incluye las PM_ADDR o las BD_ADDR de los dispositivos que desea sacar de este modo de operación. El mensaje también incluye la AM_ADDR que el maestro asigna al esclavo cuando entra en este modo. Después de enviar este mensaje, el maestro debe chequear el éxito de la operación por medio de un interrogatorio a cada esclavo que sale del *Park*. El esclavo debe responder con un *LMP_Accepted* si ha cambiado de estado. Si el maestro no recibe ninguna respuesta del esclavo después de un cierto tiempo, el maestro considera que el procedimiento ha fallado y el esclavo aun sigue en el modo de parqueo. Un tipo de mensaje se emplea cuando el dispositivo parqueado es identificado con la PM_ADDR y otro tipo cuando está identificado con la BD_ADDR. Los dos mensajes tienen una longitud diferente, porque esta depende del número de esclavos que el maestro quiere retirar del modo *Park*. Para cada esclavo que el maestro extrae del Park, enviará su AM_ADDR seguida por su PM/BD_ADDR. Si los esclavos están identificados con la PM_ADDR un máximo de siete esclavos pueden ser desparqueados con el mismo mensaje. Si ellos están identificados con la BD_ADDR un máximo de dos esclavos pueden desparquearse.

Después de que el proceso de *desparqueo* sea exitoso, las dos unidades pueden reanudar su tráfico L2CAP.

3.3.3.3.2 Control de Potencia

Un dispositivo puede requerir que su nivel de potencia de transmisión sea ajustado de acuerdo a la calidad del enlace. Si una Piconet opera cerca de otra, las señales de radio tenderán a interferirse, pero si estas operan a niveles mínimos de potencia, mayor número de piconets pueden coexistir en un espacio dado.

No solamente el maestro está en capacidad de decir si la potencia de recepción es la adecuada; Bluetooth ha establecido un mecanismo en el nivel *Link Manager* que permite a cualquier receptor solicitar cambios en la potencia del transmisor. Basándose en la medida de la Intensidad de Señal Recibida RSSI (Received Signal Streight) el receptor verifica si este valor difiere mucho del valor deseado para un dispositivo Bluetooth. Si este es el caso, el esclavo puede informar al maestro que la potencia de transmisión no es la apropiada y solicitarle un aumento o decremento según sea requerido. La unidad envía un

LMP_Inc_Power_Req para incrementar la potencia y un *LMP_Dec_Power_Req* para reducirla.

No hay necesidad de enviar una respuesta de aceptación, porque si la petición no se recibe el receptor detectará que el problema persiste y enviará la petición nuevamente.

Si el receptor de la petición se encuentra transmitiendo en la máxima o mínima potencia un mensaje *LMP_Max_Power* o *LMP_Min_Power* se retorna al dispositivo que solicita el cambio. Así un dispositivo solo puede solicitar un incremento de la potencia de transmisión cuando ha solicitado previamente un decremento de ella. Del mismo modo se presenta el procedimiento en caso de que la potencia de transmisión sea mínima.

3.3.3.4 Gestión de Enlace

Los dispositivos Bluetooth tienen varias opciones para gestionar el ancho de banda que utilizan, así como también pueden cambiar dinámicamente los esquemas de codificación con el fin de obtener la mayor ventaja al enlace. El nivel *Link Manager* realiza una serie de transacciones encaminadas al logro de este propósito. Algunas de ellas son:

3.3.3.4.1 Gestión de la Calidad de Servicio QoS (Quality of Service)

Para controlar el mínimo ancho de banda, para tráfico ACL un maestro o esclavo puede solicitar un cambio en el intervalo de Polling y el parámetro NBC *Número de Repeticiones para Broadcast,* siempre que este necesite ser ajustado. Esta petición puede ser aceptada o rechazada por la otra parte.

El maestro puede forzar a un esclavo para que modifique este parámetro y en este caso la transacción se realiza sin que pueda ser rechazada por el esclavo.

Los tipos de PDU empleados en estas operaciones son respectivamente:

Tabla 3.7. Formatos del LMP_Quality_of_Service_Req y LMP_Quality_of_Service

PDU	CONTENIDO
LMP_Quality_of_Service_Req	Tpoll: Intervalo de Polling
	NBC: Número de Repeticiones para Broadcast
LMP_Quality_of_Service	Tpoll: Intervalo de Polling
	NBC: Número de Repeticiones para Broadcast

El parámetro NBC solo tiene significado cuando es enviado por el maestro, ya que está relacionado con la operación de toda la Piconet y no solo con el enlace ACL entre un maestro y un solo esclavo.

3.3.3.4.2 Enlaces SCO

Cuando se establece una conexión entre dispositivos Bluetooth, la conexión se compone de enlaces ACL. Sin embargo pueden establecerse hasta tres conexiones SCO para dar soporte a comunicaciones de voz. El LM proporciona dos clases de PDU para el establecimiento de una conexión SCO:

LMP_Sco_Link_Req

Tabla 3.8. Formato del LMP_Sco_Link_Req

CONTENIDO	DEFINICIÓN
SCO Handle	Identificador del enlace
Timing Control Flags	Se emplean para calcular el primer Slot de transmisión.
Dsco	SCO Offset
Tsco	Intervalo de separación entre slots para la la transmisión de enlaces SCO
SCO packet	Tipo de paquete SCO

LMP_Remove_Sco_Link_Req

Tabla 3.9. Formato del Remove_Sco_Link_Req

CONTENIDO	DEFINICIÓN
SCO Handle	Identificador del enlace
Razón	Motivo por el cual finaliza el enlace

3.3.3.4.3 Esquema de Paginación

Bluetooth define un esquema de paginación adicional al esquema de paginación por defecto. El nivel *Link Manager* se encarga de anunciar e incluso negociar el esquema de paginación que será usado, la próxima vez que una unidad realice un procedimiento de paginación con otra unidad utilizando las siguientes PDU:

LMP_Page_Mode_Req y LMP_Page_Scan_Mode_Req, que contienen los siguientes parámetros:

Tabla 3.10. Formatos del LMP_Page_Mode_Req y LMP_Page_Scan_Mode_Req

CONTENIDO	DEFINICIÓN	VALOR
Esquema de paginación	Define el modo de paginación	0: Esquema Obligatorio 1: Esquema I 2: Esquema II 3: Esquema III 4-255 Reservado
Parámetros de configuración del enlace	Opciones de configuración del enlace	Para el esquema obligatorio 0: RD, 1: R1, 2: R2
Razón	Motivo por el cual finaliza el enlace	Motivo por el cual finaliza el enlace

3.3.3.4.3.1 Modo de Paginación

El procedimiento es iniciado por un dispositivo A que propone el esquema de paginación a un dispositivo B, y negocia los parámetros de configuración del esquema de Paginación. El dispositivo B puede aceptar o rechazar la propuesta.

3.3.3.4.3.2 Modo de Escaneo del Esquema de Paginación

Este procedimiento se lleva a cabo de la misma forma en que se establece el modo de paginación con la diferencia de la PDU utilizada; que en este caso es el *LMP_Page_Scan_Mod_Req*.

3.3.3.4.4 Control de Paquetes Multi-Slot

El número de ranuras de tiempo (slots) empleadas por un dispositivo puede ser limitado. Un dispositivo permite a otro utilizar un número máximo de slots por medio de un mensaje llamado *LMP_Max_Slots*, el cual no puede ser rechazado. Sin embargo también existe la posibilidad de hacer la solicitud para usar el número máximo de slots disponibles, a través del envío de un mensaje *LMP_Max_Slot_Req*; esta PDU no puede ser rechazada.

3.3.3.4.5 Información de Reloj y Temporizadores

Un dispositivo puede requerir una actualización de su reloj para optimizar las operaciones del protocolo *Link Manager*. Las transacciones que se realizan para tal propósito son:

LMP_Clock_Offset_Req: PDU enviada por un maestro que tiene como respuesta la diferencia entre el reloj del esclavo y el del maestro según lo registra el esclavo.

LMP_Clock_Offset: Contiene el tiempo de compensación de una ranura de tiempo en milisegundos, entre el inicio de un slot para del maestro y el correspondiente slot de transmisión en el esclavo. Esta información se utiliza cuando se presenta un cambio de papel.

LMP_Timing_Accuracy_Req: El resultado de esta solicitud entrega el *jitter* en milisegundos del reloj del dispositivo receptor. Esta información se requiere cuando los dispositivos retornan de largos periodos de inactividad.

3.3.3.4.6 Intercambio de Información

Las entidades LM intercambian información relacionada con las características de los dispositivos y el protocolo con el fin de coordinar eficientemente su interacción. Los tipos de mensajes empleados en este tipo de transacción son:

LMP_Version_Req: Contiene la versión del protocolo LM soportada por la entidad LM que envía el mensaje. De igual forma la entidad que recibe el mensaje retorna este mismo dato en una PDU determinada *LMP_Version_Res*.

LMP_Name_Req: el nombre es una denominación amistosa que el usuario de un dispositivo le asigna a este. El nivel *Link Manager* hace una petición a otro dispositivo para conocer su nombre mediante una LMP_Name_Req, la cual es contestada por un *LM_Name_Res*.

LMP_Features_Req: Contiene las características de los niveles *Interfaz de radio*, *Banda Base* y *Link Manager* soportadas por un dispositivo Bluetooth (del que envía el mensaje en este caso). El receptor del mensaje retorna un *LMP_Features_Res* el cual contiene sus propios datos al respecto. Algunos de ellos son:

- Soporte para el control del modo de potencia
- Codificación de voz
- Esquemas de paginación diferentes entre otros.

3.3.4 Esquema General de Operación del *Link Manager*

Las operaciones básicas del *Link Manager* se ilustran en la Figura 3.13.

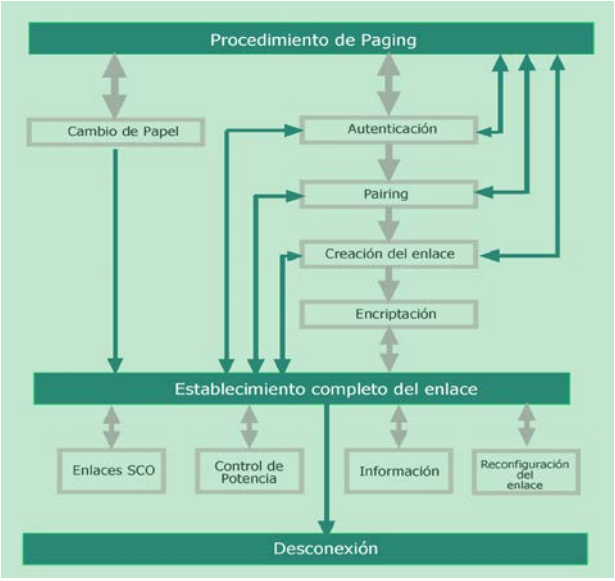


Figura 3.13. Operación del Link Manager

3.4 PROTOCOLO DE ADAPTACIÓN Y CONTROL LÓGICO DE ENLACE (LOGICAL LINK CONTROL AND ADAPTATION PROTOCOL L2CAP)

3.4.1 Introducción

El nivel L2CAP se encuentra ubicado por encima del protocolo de Gestión de Enlace (*Link Manager)* y se comunica con otros protocolos de la pila, como RFCOMM, SDP y TCS. Su labor principal es esconder las particularidades de los protocolos de transporte a las capas superiores. Por medio de esta propiedad, un gran número de aplicaciones pueden correr sobre enlaces Bluetooth sin requerir ningún tipo de modificación. La simplicidad y baja sobrecarga son algunos requerimientos esenciales para la capa L2CAP. La implementación que se haga de esta capa debe ser aplicable a dispositivos que tienen capacidades de procesamiento limitadas, debe consumir baja energía y la demanda de memoria debe ser mínima. El nivel L2CAP asume que las facilidades de transmisión en las capas inferiores proporcionan canales de comunicación *full-duplex*, los cuales transportan paquetes L2CAP_PDU en forma ordenada. La especificación solo se define para paquetes ACL, los paquetes SCO no son empleados por esta capa.

Los servicios que el protocolo L2CAP presta a las capas superiores pueden dividirse en cuatro tipos:

- Multiplexación de Protocolos:

L2CAP permite operaciones para multiplexación de protocolos ya que el nivel de Banda Base no está en capacidad de soportar campos o tipos de paquetes que pertenecen a niveles superiores, esto con el fin de lograr alta eficiencia en el manejo del ancho de banda.

- Segmentación y Re-ensamblaje:

L2CAP realiza estas operaciones para dar soporte a los protocolos que utilizan paquetes de mayor tamaño que los empleados por el nivel de Banda Base. Los paquetes que exceden el tamaño de la unidad de transmisión máxima deben ser segmentados en múltiples paquetes de tamaño igual o menor que el establecido por esta unidad. De forma similar, múltiples paquetes recibidos por el nivel de Banda Base, se re-ensamblan en paquetes L2CAP para ser transferidos a los niveles superiores.

- Calidad del Servicio QoS:

El establecimiento de una conexión L2CAP permite el intercambio de información destinada a la especificación de parámetros de los cuales depende la calidad del servicio esperada entre dos dispositivos Bluetooth. La capa L2CAP debe monitorear los recursos utilizados por el protocolo y asegurar que la QoS sea la deseada.

- Abstracción de Grupos:

Muchos protocolos incluyen el concepto de grupos de direcciones. La abstracción de grupos en L2CAP permite mapear eficientemente grupos de protocolos en las Picoredes sin exponer este concepto (piconets) a las capas superiores.

3.4.2 Canales

La comunicación entre capas L2CAP, está basada en el concepto de canales. Los canales son enlaces lógicos a través de los cuales fluye tráfico L2CAP entre 2 puntos terminales localizados en los dispositivos que comparten una conexión Bluetooth.

Cada uno de los puntos terminales en un canal L2CAP está relacionado con un *Identificador de Canal (Channel Identifier CID)* de 16 bits, por medio del cual puede diferenciarse. Los identificadores de canal son nombres locales que un dispositivo puede asignar independientemente de otro dispositivo. Los CID's, pueden ser de dos tipos:

CID Reservados empleados en funciones L2CAP, sus valores van desde 0x0001 hasta 0x003F y CID no Reservados, utilizados libremente por implementaciones especificas, con la única condición de que no sean empleados simultáneamente en el dispositivo local.

Tabla 3.11. Definición de CID empleados por el L2CAP

CID	DESCRIPCIÓN
0x000	Identificador nulo
0x0001	Canal de señalización
0x0002	Canal de recepción no orientado a conexión
0x0003 – 0x003F	Reservados
0x0040 – 0xFFFF	Empleados dinámicamente

Los puntos terminales de un canal están asociados con un Entidad Receptora de Carga Útil *(Payload Recipient Entity),* a la cual se dirige un paquete L2CAP, para procesamiento adicional. Esta entidad puede residir en la capa L2CAP misma y ser utilizada para propósitos de señalización entre capas L2CAP. También puede residir por encima de la capa L2CAP y en este caso representará una de las capas superiores cuyas PDU's son transportadas a través de L2CAP.

La especificación define tres tipos de canales en la capa L2CAP:

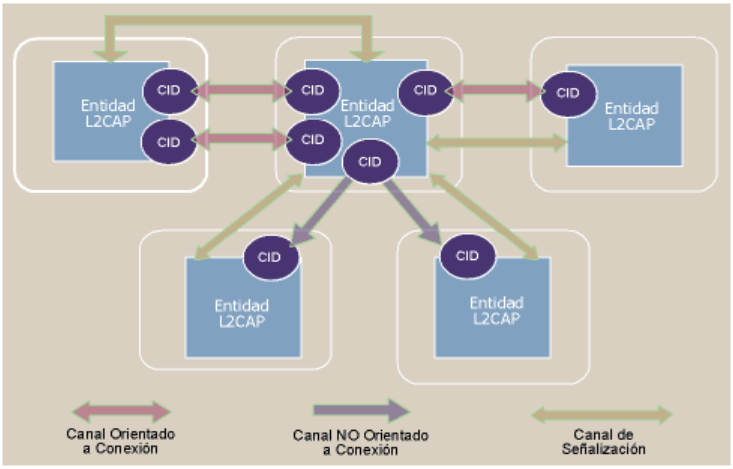


Figura 3.14. Tipos de canales L2CAP

3.4.2.1 Canales Orientados a Conexión CO:

Son canales dedicados empleados para comunicaciones bidireccionales entre dos dispositivos. La creación de uno de estos canales exige el previo establecimiento de un canal de señalización.

3.4.2.2 Canales No Orientados a Conexión CL

Son canales no dedicados unidireccionales, empleados para la transmisión de paquetes de *Broadcast* a grupos de dispositivos representados por un CID.

3.4.2.3 Canales de Señalización:

Canal dedicado y bidireccional empleado para la creación y establecimiento de canales de datos orientados a conexión, con el fin de negociar las características de configuración de los demás canales. El soporte para canales de señalización en L2CAP es obligatorio.

Los canales de señalización toman características de los canales CO y CL, pero al igual que estos últimos, no necesitan una conexión explícita previa para iniciar su comunicación.

3.4.3 Paquetes L2CAP

Existen dos clases de paquetes L2CAP, los cuales se definen de acuerdo con el tipo de canal que los emplea:

3.4.3.1 Paquetes L2CAP Orientados a Conexión CO L2CAP_PDU*:*

La Tabla 3.12 ilustra los campos que componen un CO L2CAP_PDU. Cada uno de los campos emplea un esquema de orden de bytes *little–endian*, en donde el bit menos significativo se transmite primero.

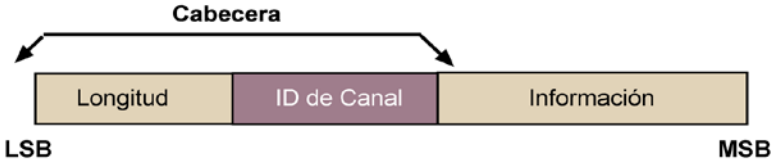


Figura 3.15. Formato de un paquete CO L2CAP

Tabla 3.12. Descripción de los campos de un paquete CO L2CAP

CAMPO		DESCRIPCIÓN
Cabecera CO L2CAP_PDU	Longitud: 2 bytes	Indica el tamaño de la carga útil en bytes, excluyendo la longitud de la cabecera. El tamaño máximo de la carga útil es 65535 bytes. Este campo sirve para chequear el paquete la integridad del paquete cuando se realizan procesos de ensamble.

	Identificador de Canal: 2 bytes	Indica el CID del punto terminal destino del paquete.
PayLoad CO L2CAP_PDU	Carga Útil	Contiene la información recibida de las capas superiores o que debe entregarse a estas. La unidad de transmisión máxima MTUco (Unidad de Transferencia Máxima para canales orientados a conexión) soportada por este tipo de paquetes es negociada durante la configuración del canal.

3.4.3.2 Paquetes L2CAP No Orientados a Conexión CL L2CAP_PDU*:*

La Figura 3.16 ilustra los campos que componen un CL L2CAP_PDU. Cada uno de los campos emplea un esquema de orden de bytes *little–endian.*

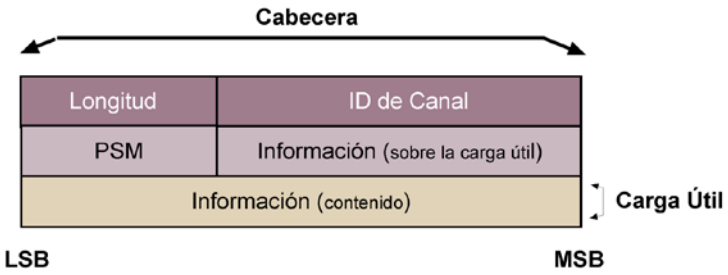


Figura 3.16. Formato de un paquete CL L2CAP

Tabla 3.13. Descripción de los campos de un paquete CL L2CAP

CAMPO		DESCRIPCIÓN
Cabecera CL L2CAP_PDU	Longitud: 2 bytes	Indica el tamaño de la carga útil en bytes mas el tamaño del campo PSM, excluyendo la longitud de la cabecera. El tamaño máximo de la carga útil es 65535 bytes. Este campo sirve para chequear el paquete la integridad del paquete cuando se realizan procesos de ensamble.
	Identificador de Canal: 2 bytes	Su valor es 0x0002, está reservado para tráfico no orientado a conexión.
	PSM Protocol Sevice Multiplexer: ≥ 2 bytes	Todo el contenido de este campo debe ser un valor impar, esto es, el bit menos significativo del octeto menos significativo debe ser 1y el bit menos significativo del octeto mas significativo debe ser 0.

PayLoad CL L2CAP_PDU	Información	Contiene la información que debe ser distribuida a todos los miembros de un grupo. Las implementaciones deben soportar un valor de unidad de transmisión máxima MTUcnl (Unidad de Transferencia Máxima para canales no orientados a conexión) mínimo de 670 bytes.

3.4.3.3 Paquetes de Señalización

Los paquetes empleados por los canales de señalización responden al concepto de Paquetes L2CAP Orientados a Conexión pero tienen un campo adicional en la cabecera.

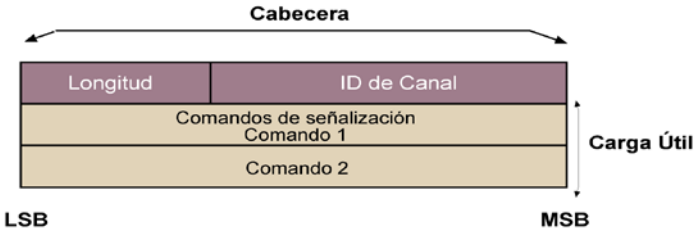


Figura 3.17. Formato de un paquete de Señalización L2CAP

La Tabla 3.14 resume las campos contenidos en este paquete.

Tabla 3.14. Descripción de los campos de un paquete de señalización L2CAP

CAMPO		DESCRIPCIÓN
Cabecera L2CAP_PDUsig	Longitud: 2 bytes	Indica el tamaño en bytes del campo de datos sin cubrir la cabecera
	Código: 1 byte	Identifica el tipo de comando de señalización
	Identificador de Canal: 1 byte	Identifica una transacción de señalización específica que ayuda a asociar una respuesta con su correspondiente petición. Los comandos de retransmisión deben utilizar el mismo tipo de identificador. Un comando que posee un identificador invalido, será descartado “silenciosamente”. Todo el contenido de este campo debe ser un valor impar, esto es, el bit menos significativo del octeto menos significativo debe ser 1y el bit menos significativo del octeto mas significativo debe ser 0.
PayLoad CO L2CAP_PDUsig	Carga Útil	Campo de datos tiene longitud variable. Contiene una colección de comandos de señalización

3.4.4 Segmentación y Ensamblaje

Las operaciones de segmentación y re-ensamblaje se utilizan para soportar una Unidad de Transmisión Máxima (MTU) más grande, que la establecida para los paquetes del nivel de Banda Base. Los paquetes de L2CAP se someten a los procedimientos de segmentación para que puedan transportarse en paquetes de Banda Base. El protocolo no realiza operaciones de segmentación y re-ensamble en si, sino que permite al formato de los paquetes L2CAP, adaptarse a tramas más pequeñas.

3.4.4.1 Procedimientos de Segmentación

La unidad de la transmisión máxima de L2CAP (MTU) se exportará usando una interfaz de servicio específica. Es la responsabilidad de las capa superiores limitar el tamaño de los paquetes al tamaño que especifica la MTU_L2CAP. Si L2CAP corre directamente sobre el nivel de Banda Base, una aplicación puede segmentar un paquete en paquetes del nivel de Banda base. Si L2CAP corre sobre el nivel *Link Manager,* una aplicación puede enviar paquetes pequeños clasificados por tamaño a la interfaz HCI, la cual realizará la conversión a paquetes del nivel de Banda Base. Todos los segmentos asociados con un paquete de L2CAP deben ser trasmitidos, antes de que otro paquete sea enviado al mismo destino.

3.4.4.2 Procedimientos de Re-ensamblaje.

La Banda Base entrega paquetes ACL en secuencia y protege la integridad de los datos utilizando un CRC de 16-bits como esquema para corrección de errores y un mecanismo ARQ para la petición de repeticiones. El nivel de Banda Base recibe paquetes ACL, e informa al L2CCAP de su arribo. También puede acumular paquetes hasta que el espacio en su buffer lo permita y posteriormente comunicarle al L2CAP acerca de su llegada.

El protocolo L2CAP debe utilizar el campo *longitud* en la cabecera de los paquetes L2CAP con el fin de verificar la consistencia de los paquetes, y si alguno de ellos presenta incoherencias, debe descartarse.

3.4.5 Gestión del Canal

Las implementaciones L2CAP siguen la arquitectura general ilustrada en la Figura 3.18. Las capas basan sus interacciones en cuatro tipos de primitivas: *Indicación, Confirmación, Requerimientos, Respuestas.*

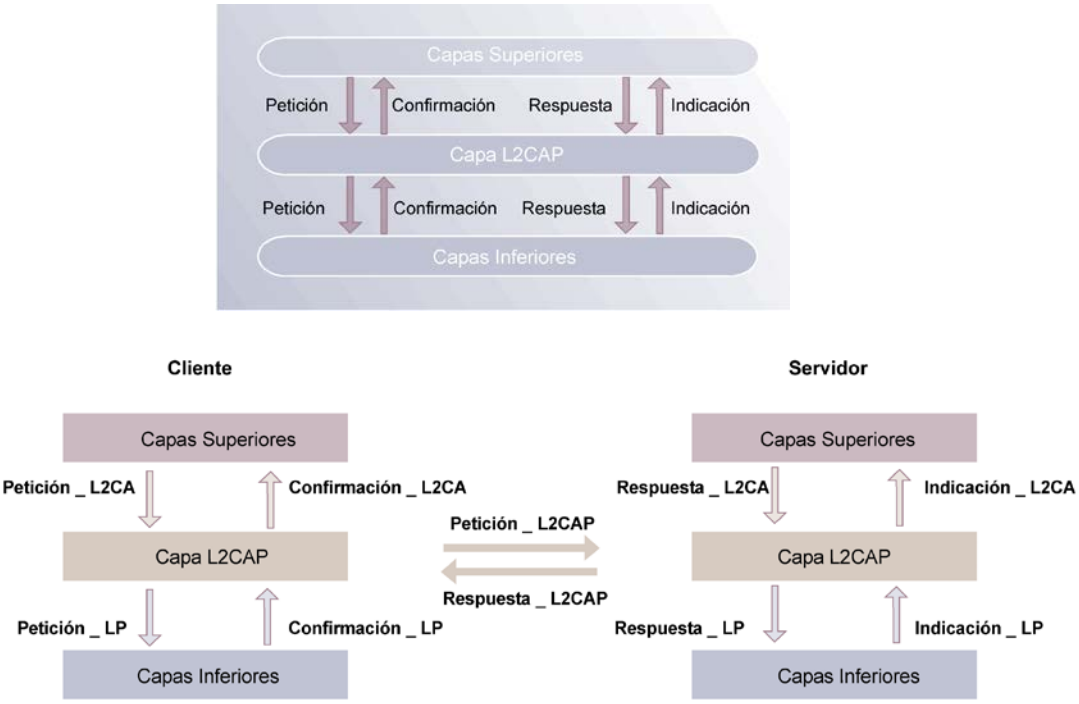


Figura 3.18. Arquitectura general de operación L2CAP

Adicionalmente las capas L2CAP deben estar preparadas para aceptar ciertos tipos de estados, eventos, que causan transición de estados y las acciones que se llevan a cabo en respuesta a los eventos.

Nota: **Observe que se ha omitido la P final en L2CA para diferenciar de los mensajes que vienen de niveles superiores o inferiores, de aquí en adelante se utilizará esta nomenclatura cuando se hable de una comunicación interna L2CAP, donde no hay intervención de niveles superiores o inferiores**

3.4.5.1 Eventos

Son mensajes entrantes al nivel L2CAP procedentes de niveles superiores o inferiores. Hay cinco categorías de eventos:

- Indicación y confirmación de los niveles inferiores.
- Petición (*Request*) y respuesta (*Responses*) de los niveles superiores.
- Datos de capas **L2CA**.
- Petición (*Request*) y Respuesta (*Responses*) de señalización.
- Eventos causados por la expiración de temporizadores.

Tabla 3.15. Eventos entre las capas inferiores y la capa L2CA

EVENTO	DESCRIPCIÓN
LP *_Connect_Cfm	Confirma el requerimiento de establecer una conexión
LP_Connect_Cfm_Neg	Negación del requerimiento
LP_Connect_Ind	Indica que el LP ha establecido exitosamente una conexión
LP_Disconnect_Ind	Indica que el LP ha sido apagado por el LMP
LP_QoS_Cfm	Confirma el requerimiento para una QoS.
PP_QoS_Cfm_Neg	Confirma una falla en el requerimiento de una QoS dada.
LP_QoS_Validation_Ind	Indica que el LP ha detectado una violación del acuerdo de QoS especificada por el LP_QoS_Req.

* LP Low Protocol (Protocolos Inferiores)

*UP Upper Protocol (Protocolos Superiores)

Tabla 3.16. Eventos entre las capas superiores y la capa L2CA

EVENTO	DESCRIPCIÓN
L2CA_Connect_Req	Requerimiento de un nivel superior para la creación de un canal en un dispositivo remoto
L2CA_Connect_Rsp	Respuesta de un nivel superior a una indicación de una petición de conexión de un dispositivo remoto
L2CA_Connect_Rsp_Neg	Respuesta negativa de una UP a la indicación de una petición de conexión de un dispositivo remoto
L2CA_Config_Req	Requerimiento de un nivel superior UP para la configuración o re configuración de un canal
L2CA_Config_Res	Respuesta de una UP a la indicación de una petición de configuración
L2CA_Config_Rsp_Neg	Respuesta negativa a la capa superior a una indicación de una petición de configuración

L2CA_Disconnect_Req	Petición que hace un nivel superior para una desconexión inmediata del canal
L2CA_Disconnect_Rsp	Respuesta que da un nivel superior a la indicación de petición de desconexión . Para este evento no hay L2CAP_Rsp_Neg.
L2CA_Data_Read	Petición de un nivel superior para realizar la transferencia de los datos recibidos de una entidad L2CAP hacia las capas superiores.
L2CA_Data_write	Petición de un nivel superior para realizar la transferencia datos desde un UP a una entidad L2CAP.

3.4.5.2 Acciones

Existen cinco categorías de acciones:

- Confirmacion e indicacion al los niveles superiores.
- Peticion (Request) y respuesta (Responses) a / de los niveles inferiores.
- Peticion (Request) y respuesta (Responses) a otras capas L2CA
- Datos entre capas l2CAP
- Establecimiento de temporizadores.

Tabla 3.17. Acciones entre las capas L2CAP y las capa inferiores

EVENTO	DESCRIPCIÓN
LP_Connect_Req	Petición de L2CAP a un LP para crear una conexión
LP_QoS_Req	Petición de L2CAP a un LP para modificar los parámetros QoS
LP_Connect_Rsp	Respuesta aceptando una indicación de petición de Conexión
LP_Connect_Rsp_Neg	Una respuesta negativa a una indicación de conexión

Tabla 3.18. Acciones entre las capas L2CAP y las capa superiores

EVENTO	DESCRIPCIÓN
L2CA_Connect_Ind	Indica una petición de conexión que ha sido recibida de un dispositivo remoto
L2CA_Connect_Cfm	Confirma que una petición de conexión ha sido aceptada
L2CA_Connect_Cfm_Neg	Confirmación negativa de una petición L2CA_Connection Request.
L2CA_Connect_Pnd	Confirma que una respuesta de conexión pendiente ha sido recibida de un dispositivo remoto
L2CA_Config_Ind	Indica que un requerimiento de configuración ha sido solicitado

L2CA_Config_Cfm	Confirma que una petición para configuración ha sido aceptada
L2CA_Config_Cfm_Neg	Confirmación negativa a una petición para configuración
L2CA_Disconnect_Ind	Indica que una petición de desconexión ha sido recibida de un dispositivo remoto o el dispositivo ha sido desconectado porque no hay peticiones de señalización.
L2CA_Disconnect_Cfm	Confirma que una petición de desconexión ha sido procesada por el dispositivo remoto. Después de recibir este evento la capa superior sabe que el canal L2CAP ha finalizado
L2CA_Time Out_Ind	Indica que un RTX o ERTX ha expirado
L2CA_QoS_Violation	Indica que un acuerdo de QoS ha sido violado

3.4.5.3 Estados

La Tabla 3.19 describe los posibles estados en transacciones L2CAP

Tabla 3.19. Eventos en transacciones L2CAP

ESTADO	DESCRIPCIÓN
CLOSED	En este estado, no hay canales asociados con un CID. Este es el único estado en el cual un enlace a nivel de banda base puede no existir.
W4_L2CAP_CONNECT _RSP	En este estado un CID representa un punto terminal local y un L2CAP_CONNECT_REQ ha sido enviado referenciando este punto terminal, que se encuentra ahora esperando un mensaje con la respuesta correspondiente a su petición.
W4_L2CA_CONNECT _RSP	En este estado existe un punto terminal remoto y la entidad L2CAP local ha recibido un L2CAL_CONNECT_REQ. También se ha enviado un L2CA_CONNECT_IND a la capa superior y se espera por la respuesta correspondiente.
CONFIG	Es este estado la conexión ya ha sido establecida, pero las dos partes negocian los parámetros del canal.
OPEN	La conexión se encuentra establecida y ya se ha configurado; el trafico de datos puede iniciarse.
W4_L2CAP_DISCONNECT _ RSP	Es este estado la conexión se finaliza. Se envía un mensaje L2CAP_DISCONNECT_REQ a la otra parte y se espera la respuesta correspondiente.
W4_L2CA_DISCONNECT_ RSP	En este estado, se cierra la conexión con una entidad L2CAP remota, y se recibe un mensaje L2CAP_Disconnect_Req. También se envía a las capas superiores un mensaje L2CAP_Disconnect _Ind para comunicar que uno de sus puntos terminales ha sido desconectado.

3.4.6 Señalización

Los procesos de señalización se emplean para establecer, configurar y finalizar los canales de comunicación L2CAP.

La señalización L2CAP se basa en peticiones y respuestas que se transportan en canales de señalización cuyos puntos terminales tienen in CID reservado cuyo valor es 0x0001.

Contrario a otro tipo de transmisión, las transferencias de paquetes de señalización son confiables en la medida que un dispositivo pueda re enviar una petición si no ha recibido una respuesta a su solicitud en un tiempo determinado; que va desde 1 segundo a 60 segundos. Sin embargo un dispositivo puede esperar hasta 300 segundos para realizar una respuesta final. Si el dispositivo remoto ha respondido indicando que ha recibido la petición pero necesita más tiempo para procesarla.

Los comandos empleados por los procedimientos de señalización son:

3.4.6.1 Establecimiento de la Conexión

3.4.6.1.1 *Petición de Conexión (C*onnection_Request)

Los mensajes *Connection_Request* se emplean para crear un canal L2CAP entre dos dispositivos. Un Canal debe ser establecido para poder enviar el procedimiento de configuración.

- Código: 0x02
- Longitud: 0x0004 o mas bytes
- PSM: 2 bytes mínimo

Tabla 3.20. Valores del campo PSM

VALOR DE PSM	PROTOCOLO CORRESPONDIENTE
0x0001	Protocolo para Descubrimiento de Servicios
0x0003	RFCOMM
0x0005	Protocolo para Control de Telefonía
< 0x1000	Reservado

- Identificador de Canal Fuente: 2 bytes. Representa al punto terminal de canal que envía la petición.

3.4.6.1.2 Respuesta de Conexión (Connection _Response)

Cuando una unidad recibe un paquete que requiere o solicita una conexión debe enviar una respuesta por medio de este paquete.

- Código 0x03
- Longitud: 0x0008 bytes
- Identificador de Canal Destino *(Destiny Channel Identifier DCID*): 2 bytes, Corresponde al CID del punto terminal que envía a respuesta.
- Identificador de Canal Fuente *(Source Cahnnel Identifier SCID)*: 2 bytes, Corresponde al CID del dispositivo que recibe la respuesta.
- Resultado:

Tabla 3.21. Significado de los valores del campo resultado a una peticion de conexión

VALOR	DESCRIPCIÓN
0x0000	Conexión Exitosa
0x0001	Conexión Pendiente
0x0002	Conexión Rechazada. No hay soporte para PSM
0x0003	Conexión Rechazada. Bloqueo de seguridad
0x0004	Conexión Rechazada. No hay recursos disponibles
Otro	Reservado

- Estado (*Status):* 2 bytes. Se define para una respuesta pendiente. Indica el estado de la conexión.

3.4.6.2 Configuración de la Conexión

3.4.6.2.1 *Petición de Configuración* (Configuration_Request)

Posterior al establecimiento del canal, este debe ser configurado. Los mensajes *L2CAP_ConfigReq* se emplean para negociar los parámetros de un canal cuando esto sea necesario.

Durante las transacciones de negociación, todo el tráfico presente en el canal debe ser suspendido. Cada uno de los parámetros concentrados en un mensaje *L2CAP_ConfigReq* están relacionados exclusivamente con el tráfico en una sola dirección, esto significa que si un dispositivo necesita establecer el valor de un parámetro determinado en dirección opuesta a la que se refiere el *L2CAP_ConfigReq* que ha recibido, deberá enviar de nuevo un *L2CAP_ConfigReq* indicando el valor deseado del parámetro en cuestión en la dirección contraria al paquete original.

- Código: 0x04.
- Longitud: 0x0004 bytes o más.
- Identificador de Canal Destino DCID: 2 bytes, CID del punto terminal que envía respuesta.
- Banderas: 2 bytes. En la versión 1.0 de la especificación, solo se emplea el bit menos significativo de este campo. Este bit denominado bit C, se utiliza como una bandera de continuación la cual indica que opciones adicionales de configuración, están por aún por recibirse. Esto ocurre cuando todas las opciones de configuración de un canal, no pueden ser enviadas en un solo paquete ya que la MTUsig (Unidad de Transmisión Máxima para señalización) del receptor, no lo permite. Por supuesto el receptor debe responder a cada una de las peticiones que recibe, aun cuando estas se refieran a segmentos de configuración.

3.4.6.2.1.1 Parámetros de Configuración

A continuación se listan los tipos de parámetros que pueden negociarse mediante procesos de configuración L2CAP. La interpretación de cada uno de ellos depende del dispositivo local y como se menciono anteriormente, el procedimiento se realiza unidireccionalmente.

- Unidad de Transmisión Máxima (MTU)

  Identifica la capacidad máxima de la carga útil en bytes que un dispositivo puede soportar sobre el canal L2CAP.
  MTU mínimo = 48bytes
  MTU máximo =672bytes

- Tiempo de Descarga (*Flush Time Out)*

  Identifica la cantidad de tiempo en múltiplos de 1.25 milisegundos durante la que el nivel *Link Manager* de un dispositivo local, continuara intentando transmitir los segmentos de un paquete L2CAP desde el nivel de Banda Base a un nivel superior, antes de que los descargue. Por defecto el time out indica el grado de confiabilidad con que las PDU's son transmitidas antes de que el enlace se pierda.

- Calidad de Servicio (QoS):

  Identifica el tipo de tráfico sobre un canal en un dispositivo local. Esta opción incluye un parámetro denominado Tipo de Servicio *(Service Type)*, el cual puede tener dos tipos de valores: *Best Effor* y *Guaranteed*. El primero de estos es un valor por defecto y que es obligatorio para cualquier implementación L2CAP. Cuando el tipo de servicio es *Guaranteed* el dispositivo local proporciona unos parámetros de QoS asociados al flujo de trafico de salida.

Estos parámetros se comunican posteriormente al *Link Manager* con el fin de que sean negociados con el *Link Manager* de otro dispositivo para determinar un intervalo de Polling que de soporte a la QoS deseada.

3.4.6.2.2 Respuesta de Configuración (Configure response)

Estos paquetes se envían como respuesta a una solicitud para configuración del canal. El valor del parámetro contenido en un paquete *Configure response* refleja un ajuste al valor de un parámetro solicitado por medio de un *L2CAP_ConfigReq*, el cual debe ser modificado solamente en la dirección de la solicitud.

- Código: 0x05.
- Longitud: 0x00086 bytes o más.
- Identificador de Canal Fuente: SCID 2 bytes. CID del dispositivo que recibe la respuesta.
- Banderas: Definidas de la misma forma en que se presentan para la Petición de Conexión.
- Opciones de Configuración: Contiene la lista de los parámetros que un dispositivo ha aceptado para que sean modificados. Si el valor de uno de los parámetros no ha sido aceptado, este parámetro será devuelto en la respuesta pero con un valor que el dispositivo sugiere y por supuesto es aceptado por este. Si se presenta una falla que responde al desconocimiento de los parámetros a negociar, la respuesta deberá contener una lista con el tipo de parámetro que no pudo entender o conocerse.

3.4.6.3 Finalización de la Conexión

3.4.6.3.1 *Requerimiento de Desconexión* (Disconnection_Request)

La finalización de un canal L2CAP requiere que un dispositivo envíe un paquete con una petición para desconexión y reciba un ACK dado por una respuesta al paquete de desconexión. El receptor debe asegurarse de que tanto el CID fuente como el CID destino inicien el procedimiento de desconexión. Una vez se envía un Disconnection_Request todo el trafico entrante y saliente al CID se descarta.

- Código: 0x06
- Longitud: 0x0004 bytes

- Identificador de Canal Destino DCID: 2 bytes, Corresponde al CID del punto terminal que envía la respuesta.
- Identificador de Canal Fuente SCID: 2 bytes, Corresponde al CID del dispositivo que recibe la respuesta

3.4.6.3.2 Respuesta de Desconexión (Disconnection_Response)

Una respuesta de desconexión debe ser enviada cada vez que un dispositivo recibe un *Disconnection_Request.*

- Código: 0x07
- Longitud: 0x0004 bytes
- Identificador de Canal Destino DCID: 2 bytes, Corresponde al CID del punto terminal que envía a respuesta.
- Identificador de Canal Fuente SCID: 2 bytes, Corresponde al CID del dispositivo que recibe la respuesta

El DCID y SCID (relativos al dispositivo que envía la petición) y el campo *Identificador* deben coincidir con los valores presentes en el comando *Request_Disconect*, de lo contrario la respuesta se descartará.

3.4.6.4 Transacciones Adicionales

3.4.6.4.1 *Rechazo de Comando* (Command Reject)

Se emplea para rechazar un paquete que contiene un comando desconocido o para notificar que la respuesta que recibe una entidad es inapropiada.

- Código: 0x01
- Longitud: 0x0002 o más bytes
- Motivo del Rechazo:

Tabla 3.22. Motivo de rechazo del comando

VALOR	DESCRIPCIÓN
0x0000	Comando desconocido
0x0001	MTU excedida
0x0002	CID Inválido
Otro	Reservado

3.4.6.4.2 *Requerimiento de Eco* (Echo_Request)

Esta petición se utiliza para probar el enlace con una entidad L2CAP remota. Este comando se asemeja al ping utilizado en redes IP.

- Código: 0x08
- Longitud: 0x0004 bytes

3.4.6.4.3 *Respuesta de Eco* (Echo Response)

Es la respuesta que se envía a un *Echo_Request*. El identificador de la respuesta debe coincidir con el identificador enviado en el *Echo_Request*.

- Código: 0x09
- Longitud: 0x0004 bytes

3.4.6.4.4 *Petición de Información* (Información Request)

Se utiliza para solicitar a una entidad L2CAP remota, información especifica sobre algún tipo de aplicación o implementación.

- Código: 0x0A
- Longitud: 0x0002 bytes
- Tipo de Información: 2 bytes

3.4.6.4.5 Respuesta de Información (Information Response)

Contiene la información solicitada a través de un *Informatión Request*.

- Código: 0x0B
- Longitud: 0x0002 bytes
- Tipo de Información: 2bytes
- Resultado: Campo que contiene valores de respuesta a solicitudes previas

Tabla 3.23. Significado de los valores del campo resultado a una petición de información

VALOR	DESCRIPCIÓN
0x0000	Transacción Exitosa
0x0001	No soportado
Otro	Reservado

- Data: su contenido depende del campo *Tipo de Información*. Si la petición se refiere al MTU, el campo data contiene el MTU de la unidad interrogada.

3.4.7 Gestión de Grupos

El nivel de Banda Base soporta el concepto de Piconet, un grupo de dispositivos que saltan sincrónicamente usando un mismo reloj.

L2CAP permite realizar una abstracción de estos grupos con el fin de que no sea necesario que las capas superiores se expongan a capas inferiores como el LM y la Banda Base para manejar grupos eficientemente. L2CAP utiliza canales no orientados a conexión para enviar paquetes de datos a todos los miembros de un grupo. Los grupos no tienen una QoS asociada a ellos y por lo tanto L2CAP no presenta un grado de confiabilidad que permita garantizar la entrega de un paquete a todos los miembros de un grupo. Sin embargo las transmisiones a un grupo no son exclusivas, ya que un dispositivo que no pertenece a un grupo determinado puede recibir este tipo de transmisiones. Si se desea que las transmisiones sean privadas puede hacerse uso de un mecanismo de encriptación de alto nivel.

La gestión de grupos se maneja a través de cinco primitivas de servicio que se describen a continuación:

3.4.7.1 Creación de Grupos L2CA_Group_Create:

Esta primitiva se emplea para solicitar la creación de un CID que represente una conexión lógica a múltiples dispositivos. El parámetro de entrada PSM sirve para nombrar el tráfico saliente y filtrar el tráfico entrante. El parámetro CID representa el punto terminal local.

- Parámetro de Entrada: PSM
- Parámetro de Salida: CID

3.4.7.2 Cierre de Grupos L2CA_Group_Close

Como su nombre lo dice esta primitiva se emplea para clausurar un grupo definitivamente.

- Parámetro de Entrada : CID, identifica el grupo que desea cerrarse.
- Parámetro de Salida: Resultado

Tabla 3.24. Significado de los valores del parámetro de salida de L2CA_Group_Close

VALOR	DESCRIPCIÓN
0x0000	Cierre exitoso del canal
0x0001	CID inválido

3.4.7.3 Adición de Miembros al Grupo L2CA_Group_Add_Member

Utilizada para adicionar un miembro al grupo.

- Parámetros de Entrada: CID, representa el grupo y BD_ ADDR, la dirección del dispositivo que desea adicionarse al grupo.
- Parámetro de Salida: Resultado. Confirma el éxito o fracaso de la petición.

Tabla 3.25. Significado de los valores del parámetro de salida de L2CA_ Group_Add_Member

VALOR	DESCRIPCIÓN
0x0000	Transacción Exitosa
0x0001	Falla al establecer la conexión al dispositivo remoto
Otro	Reservado

3.4.7.4 Remoción de Miembros del Grupo L2CA_Group_Remove_Member

Es la primitiva que realiza una operación contraria a la anterior primitiva, por medio de la cual se retira la membresía de algún dispositivo en el grupo.

- Parámetro de Entrada : CID, representa el grupo.
- BD. ADDR es la dirección del dispositivo que desea retirarse.
- Parámetro de Salida: Resultado . Confirma el éxito o fracaso de la operación .

Tabla 3.26. Significado de los valores del parámetro de salida de L2CA_Group_Remove_Member

VALOR	DESCRIPCIÓN
0x0000	Transacción Exitosa
0x0001	Falla: El dispositivo no es un miembro del grupo
Otro	Reservado

3.4.7.5 Obtención de lista de Integrantes del Grupo L2CA_Get_Group_Membership

Es una petición para obtener la lista de todos los miembros que hacen parte de un grupo especifico.

- Parámetro de Entrada : CID, representa el grupo.
- Parámetros de Salida: Los parámetros de salida son tres:
  Resultado: Indica el estado de la transacción

Tabla 3.27. Significado de los valores del parámetro de salida de L2CA_Get_Group_Membership

VALOR	DESCRIPCIÓN
0x0000	Transacción Exitosa
0x0001	Falla: El grupo no existe
Otro	Reservado

Si la solicitud encuentra respuesta, el resultado devuelve el número total de miembros y la lista de sus direcciones.

Tabla 3.28. Número total de miembros N

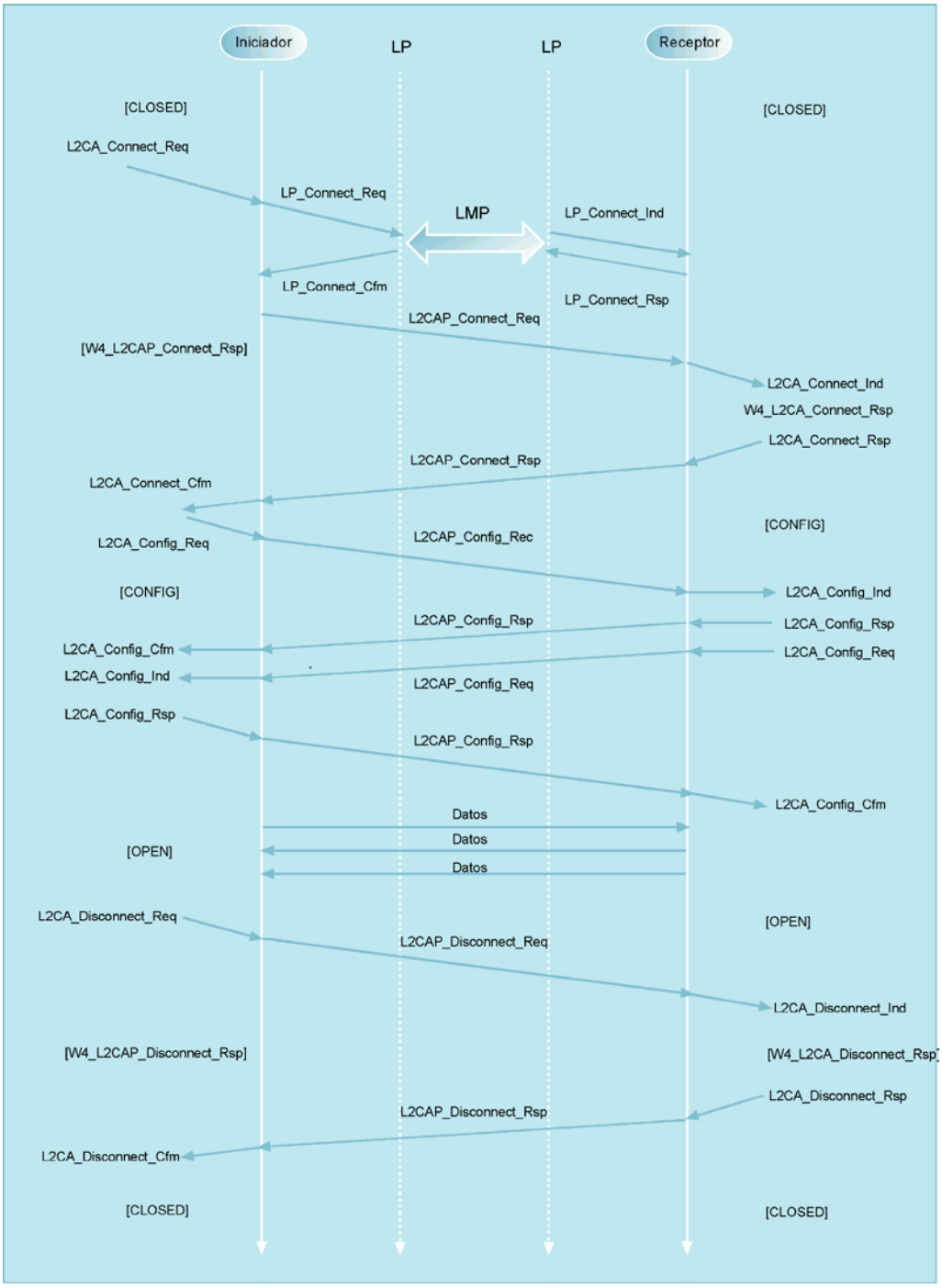
VALOR	DESCRIPCIÓN
0x0000 – 0XFFFF	Número de dispositivos en el grupo identificados por un CID.

Tabla 3.29. Lista de Direcciones

VALOR	DESCRIPCIÓN
0x0000	Transacción Exitosa
0xXXXXXXXXXXXX	Lista de la direcciones Bluetooth de los dispositivos en el grupo, identificados por un CID

3.4.8 Esquema General de Operación del L2CAP

Ver página siguiente....



Carta de Secuencia 3.1. Mensajes para las operaciónes L2CAP

3.5 INTERFAZ DEL EQUIPO CONTROLADOR

Recorriendo la pila de protocolos es posible darse cuenta que en este punto convergen la parte Hardware y Software de Bluetooth, puesto que las capas de Radio, Banda Base, y Link Manager están implementadas en firmware[12]. En la Interfaz de Equipo Controlador ó también llamada por la especificación HCI (Host Controller Interface), la lógica está implementada en parte en Firmware y otra parte en Software. También existe una interfaz hardware entre ellas, que puede variar según la necesidad de la aplicación del dispositivo Bluetooth.

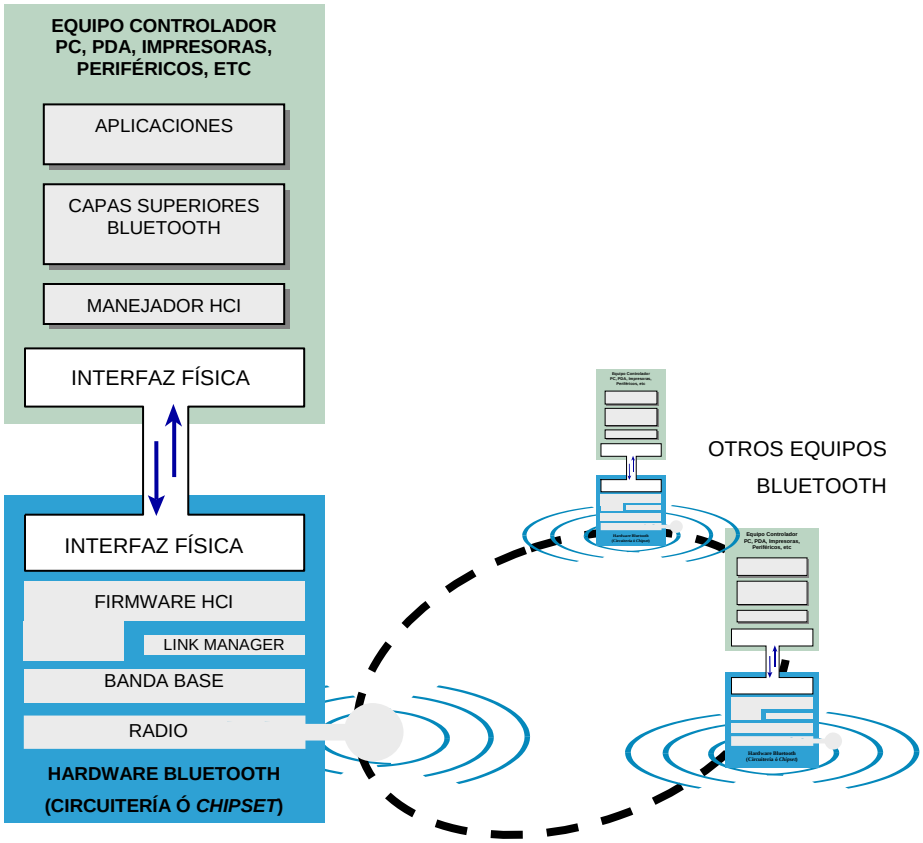


Figura 3.19. La interfaz de equipo controlador y su papel en la comunicación Bluetooth

El módulo Hardware Bluetooth según la especificación, puede integrarse al equipo ó Host que lo posea por medio de una interfaz física, que puede ser serial (USB, RS232, UART), ó tarjeta de computador (Bus ISA), aunque esta parte solamente se encuentra especificada en

[12] El Firmware en este caso es la lógica programada en la circuitería del módulo Bluetooth

un “Whitepaper” hecho por el SIG[13]. Por esta interfaz debe pasar, en absoluto, toda la información requerida por el Hardware Bluetooth, información de control, datos y comandos provenientes de niveles superiores de la pila de protocolos.

No se descartan otras posibilidades de integración del Hardware Bluetooth con el Hardware del equipo controlador, este seria el caso de dispositivos totalmente integrados; supongamos, tarjetas principales para computadores con puerto Bluetooth, ó teléfonos igualmente equipados, donde el hardware Bluetooth estaría totalmente integrado a la circuiteria del dispositivo. En éste caso la interfaz no es necesaria, pero el fabricante debe encargarse de la implementación de los comandos que deben ejecutarse en el firmware de Bluetooth. En la Figura 3.19 puede apreciarse el papel que juega esta interfaz, teniendo en cuenta que igualmente está implementado cualquier otro dispositivo que pueda entablar comunicaciones utilizando la tecnología, y que las interfaces físicas pueden variar según el tipo de dispositivo que haga las veces de equipo controlador.

3.5.1 Responsabilidades de la Interfaz HCI

Como se dijo anteriormente, por la interfaz HCI debe pasar toda la información que sea necesaria para el Hardware Bluetooth, esta incluye:

- Información de usuario proveniente de capas superiores de la pila de protocolos con el fin de ser transportada hacia otro dispositivo.
- Comandos para instruir al hardware acerca de su comportamiento.
- Información asíncrona de eventos que ocurran con el hardware (como conexiones desde otros dispositivos o cambios de estado).

La información debe serializarse para poder ser transportada, para ello se definen varios tipos de mensajes específicos para cada clase de información descrita.

También, dado que la interfaz HCI controla el hardware Bluetooth, es responsable de ejecutar comandos tendientes a cumplir tareas específicas como:

- Inicialización del dispositivo local:

[13] Grupo Especial de Interés en Bluetooth – Special Interest Group

    - Averiguar las capacidades del dispositivo local (características funcionales que soporta, información de versión de protocolos que soporta, tamaño del Buffer, y Código de Acceso a Indagación que espera recibir).
    - También realiza una configuración inicial del dispositivo, es decir, establece filtros para recibir eventos, habilitar y establecer los parámetros de escaneo de paginación ó de indagación).
- Búsqueda de otros dispositivos en la periferia:
    - Usar los comandos de indagación para descubrir a otros dispositivos.
    - Establecer parámetros para realizar un escaneo de indagaciones con el fin de ser descubierto por otros dispositivos (esta característica no es realizada por algunos dispositivos de carácter “privado”).
- Conexiones:
    - Establece los parámetros para escaneo de paginación.
    - Realiza conexiones nuevas tanto ACL como SCO.
- Intercambio de datos:
    - Transfiere información desde y hacia el equipo controlador.
    - Identifica a cada una de las conexiones que se tengan por medio del uso de Handlers[14].
- Aumentar ó disminuir el número de dispositivos de una Picored por medio de la utilización de comandos para alternar el estado de los dispositivos entre activo y de parqueo.

3.5.2 Tipos de Mensajes

Las PDU ó Unidades de Datos de Protocolo que se definen son las siguientes:

1. de Comandos
2. de Eventos
3. de Datos
    a. Asíncronos (L2CAP)
    b. Síncronos

Las estructuras que tienen estas PDU son muy similares, y se distinguen por un código específico dentro de un campo de dicha PDU. Ahora se explicarán cada una de las PDUs y su utilidad. Para aclarar la concepción que se tenga de lo que significa una PDU a este nivel

[14] Los Handlers (ó También Connection Handlers) son números identificadores de las conexiones a nivel de Link Managers

es que es una trama, lista para ser enviada por una interfaz física serial. A continuación se describe su estructura y la especialización de cada PDU.

3.5.2.1 Mensajes de Comandos

Estos comandos van dirigidos del HCI al *Host Controller*[15] ó Firmware HCI, y tienen como funciones:

- Establecen parámetros operacionales, (autenticación, claves de enlace, etc.).
- Configura el estado operacional del módulo (por ejemplo, modos de bajo consumo de energía y parámetros de funcionamiento).
- Leer y escribir entradas de registros. (valores de contadores, temporizadores, etc que sirven como parámetros al nivel de banda base).

Algunos ejemplos de estas funciones son los comandos para descubrir otros dispositivos en la vecindad (como los comandos HCI_Inquiry y HCI_Periodic_Inquiry_Mode); para crear y terminar conexiones ACL ó SCO, para aceptar o rechazar conexiones provenientes de otros dispositivos, negociación de parámetros de autenticación y encriptación, y solicitudes de información propia de dispositivo tal como su nombre (*user-friendly name*).

La estructura de los paquetes es la que se muestra a continuación.

IDENTIFICADOR DE COMANDO		PARÁMETROS DEL COMANDO	
OPCODE		LONGITUD	CARGA ÚTIL
OGF	OCF		
6 BITS	10 BITS	1 BYTE	0 - 255 BYTES
2 BYTES			

Figura 3.20. Estructura General de un Paquete HCI

Los campos que juegan papel en esta trama son:

- OpCode: es el código de operación, especifica el comando que se está transportando en esta trama. Con esto el Firmware HCI podrá reconocer qué parámetros debe buscar en el carga útil.
- Campo de Grupo (OpCode Group Field – OGF): Identificador de Grupo de operación. Los comandos HCI están divididos en 5 grupos: Comandos de Control de Enlace,

[15] El Host Controller es el firmware HCI que interpreta los comandos HCI para que sean ejecutados en el módulo Bluetooth a nivel del Link Manager y el nivel de Banda Base.

Comandos de Políticas de Enlace, Comandos de Firmware HCI y de Banda Base, Comandos para Parámetros de Información, Comandos de Estado, y Comandos para Pruebas. (ver grupos de comandos más adelante).

- Campo de Comando (OpCode Command Field – OCF): Identificador del comando específico.
- Longitud: Longitud en Bytes que ocupa la Carga útil.
- Carga útil: Es la información que lleva el comando, es decir, los *parámetros* que el comando utilizará en el firmware para ejecutarse. Gracias al código del comando se conocerán los parámetros que deberán llegar.

Los comandos se envían a través de la interfaz HCI y una vez en el hardware de Bluetooth son procesados en forma concurrente; un comando puede comenzar su ejecución cuando aun no ha terminado la ejecución de otro, lo cual hace que no se terminen de ejecutar en el mismo orden en que se recibieron.

3.5.2.1.1 Grupos de Comandos HCI

Los grupos de comandos se especializan en tareas propias de un nivel ó niveles específicos, ó también en áreas específicas según el tipo de información que involucre. Los grupos de comandos están identificados por el número correspondiente al identificador de grupo OGF descrito en la Figura 3.20. Los grupos son los siguientes:

1. *Comandos de Control de Enlace*: Permiten al Firmware HCI controlar conexiones entre dispositivos Bluetooth. Instruyen a la capa de Link Manager para crear y modificar conexiones a nivel de enlace, también realiza las respectivas indagaciones para descubrir dispositivos que estén al alcance.
2. *Comandos para Políticas de Enlace*: Provee métodos para modificar el comportamiento de las Picoredes; es decir, modifican el comportamiento del Link Manager.
3. *Comandos de Firmware HCI y Banda Base*: Da acceso a funcionalidades del Hardware (firmware HCI, banda base y Link Manager) del dispositivo local.
4. *Comandos para Parámetros de Información*: Son funciones para acceder a información de solo lectura del dispositivo, tal como su dirección física ó la versión de Bluetooth que soporta.

5. *Comandos para Información de Estado*: Permite consultar información acerca del estado del hardware; estos estados no pueden modificarse, y entre ellos están el estado de un enlace y la lectura en dB del RSSI[16].
6. *Comandos de Prueba*: Invoca funcionalidad para pruebas de los dispositivos Bluetooth, establece condiciones de prueba. Utilizados en la fabricación y calificación de los productos que saldrán al mercado. Un ejemplo de estas funcionalidades es la habilitación del modo de prueba Loopback, en el cual los comandos HCI se ejecutan emulando la presencia de otro dispositivo. (similar al comportamiento de la dirección IP 127.0.0.1 ó dirección Loopback).

3.5.2.2 Mensajes de Eventos

Ocurren de forma asíncrona desde el Firmware hacia el HCI; reportan la ocurrencia de cambios internos o de solicitudes de otros dispositivos; estos eventos pueden ser:

- La terminación exitosa ó fallida de un comando invocado previamente
- El cambio de estado de ahorro de energía de un dispositivo
- Solicitudes de conexión provenientes de otros dispositivos
- Solicitudes de autenticación

La estructura de las tramas de eventos es la siguiente:

IDENTIFICADOR DE EVENTO	PARÁMETROS DEL EVENTO	
CÓDIGO DE EVENTO	LONGITUD	CARGA ÚTIL
1 Byte	1 Byte	0 - 255 Bytes

Figura 3.21. Estructura general de las tramas de eventos

Se tiene entonces la identificación del evento. Existen definidos por la especificación 32 eventos y también están descritos por sus parámetros; el firmware HCI sabrá que parámetros debe buscar según el tipo de evento que venga en una trama; por ejemplo, cuando se ejecuta el comando HCI_Inquiry ó HCI_Periodic_Inquiry_Mode se genera, al terminar su ejecución, un evento llamado Inquiry_Complete_Event, el cual indica que ha finalizado la ejecución de la indagación ó Inquiry; este evento posee un solo parámetro llamado *Status*, que puede ser de valor 0x00 (en caso de una ejecución exitosa) ó en su

[16] Received Signal Strength Indicator, es un número indicador de la intensidad de la señal de radio recibida para poder saber si ésta es suficiente ó no (en tal caso activar mecanismos para negociar un nivel de señal mayor – esta característica es opcional).

lugar algún otro número (0x01 – 0xFF) que identificará a un error dentro de una lista específica de códigos de errores (definida en la especificación).

Los eventos entonces brindan información acerca del comportamiento del dispositivo y por ello la aplicación que se tenga haciendo uso del hardware Bluetooth debe responder a ellos de manera asíncrona si se quiere hacer énfasis en la robustez de dicha aplicación.

3.5.2.3 Mensajes de Datos

Estas tramas son más sencillas, puesto que se sabe que contienen datos, y por ello deben pasar intactas hasta el otro dispositivo hacia el cual van a ser transferidos. Por supuesto que los datos que pasen por estas tramas tendrán la misma clasificación que se les ha dado desde su definición a nivel de banda base: Asíncrona - ACL ó Síncrona – SCO. Las tramas entonces están diferenciadas en este sentido. Debe tenerse muy en cuenta que una trama de estas solamente sirve para transportar datos, y como tal, necesita que previamente haya una conexión ACL establecida con el dispositivo al cual va dirigida. Incluso la información SCO requiere que previamente haya una conexión ACL (con el fin de que se utilice como canal de control mientras ocurre el establecimiento del enlace).

Otra característica de las tramas de datos es que no se encuentran direccionadas a un dispositivo en especial, sino que a este nivel el direccionamiento de los dispositivos en la Picored está dado por un identificador de conexión, también llamado *Connection Handler*, y que da identificación a una conexión particular (independiente de a qué dispositivo esté dirigido). Al crearse una conexión el equipo controlador ó Host recibe un Handler para que la aplicación pueda distinguir entre las distintas conexiones que haya establecido. (Ver Más adelante el punto 3.5.3 - *Ejemplo de Descubrimiento de Dispositivos por Comandos HCI*). La estructura de la trama es la siguiente.

Para tramas ACL:

Identificador de Conexión	Banderas		Longitud	Datos
	PB	BC		
12 bits	4 bits		2 Bytes	0 - 65535 Bytes

Para Tramas SCO:

Identificador de Conexión	Reservado	Longitud	Datos
12 bits	4 bits	1 Byte	0 - 255 Bytes

Figura 3.22. Forma de las tramas de datos del nivel HCI

Connection Handler: Es el identificador de conexión descrito arriba.
Banderas:

- Borde de Paquete: (Packet Boundary - PB). indica si este es un fragmento de un mensaje proveniente de niveles superiores.
- Broadcast BC. hace distinción entre paquetes distribuidos a todos los esclavos de la Picored, si es punto a punto, ó si va dirigido a todos los miembros de la Picored (incluyendo los que estén en el estado de parqueo).

La longitud de los datos está restringida a 1 ó 2 bytes en cada caso.

3.5.3 Ejemplo de Descubrimiento de Dispositivos por Comandos HCI

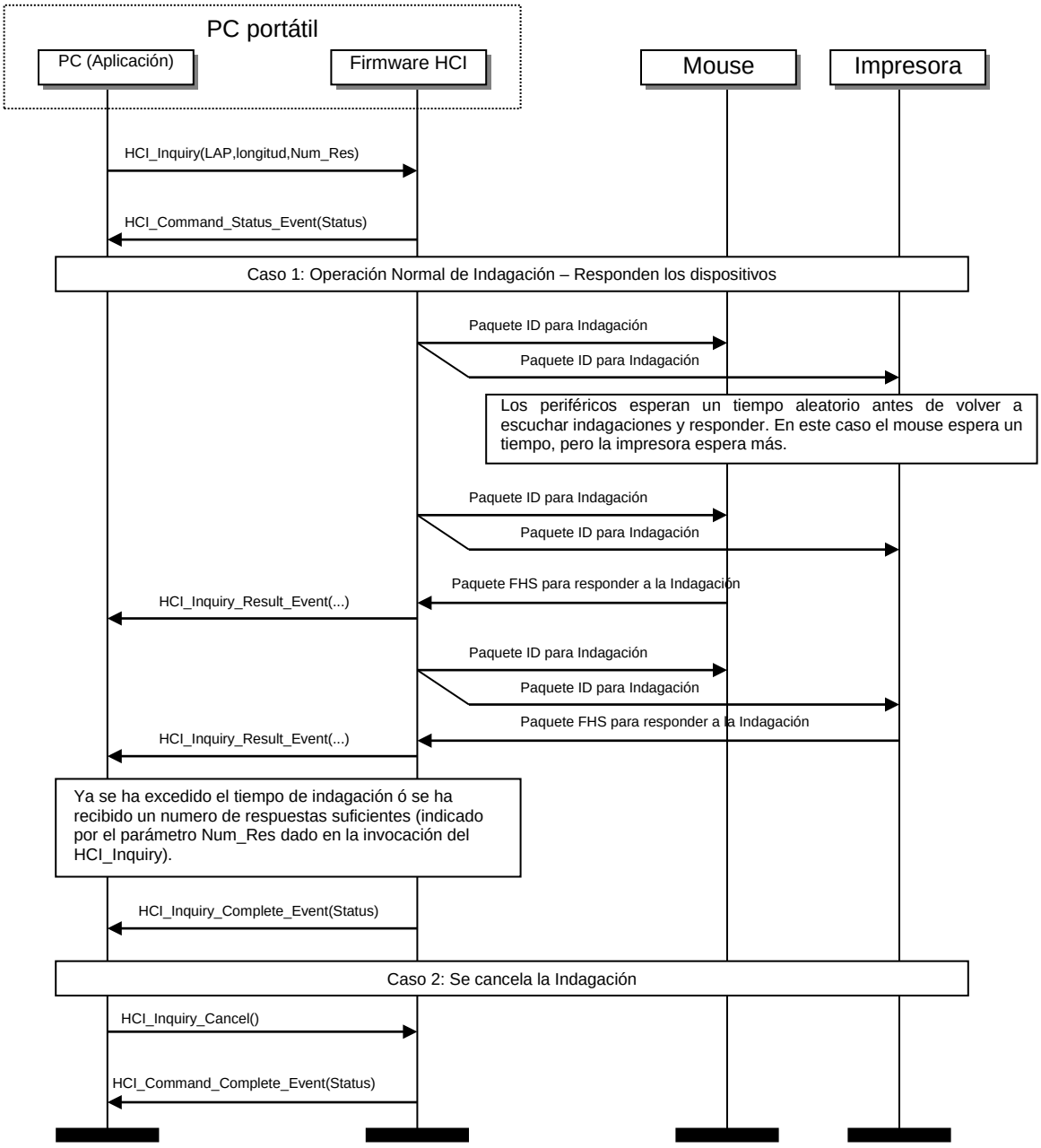
Puede también hacerse una similitud entre estas invocaciones y las llamadas a funciones que se hacen regularmente en programación, generalmente la función se invoca especificando sus parámetros, por ejemplo, para comandos HCI un ejemplo seria invocar el comando de indagación y recoger la información de otros dispositivos en la cercanía; el proceso se describe en la Carta de Secuencias 3.2 Proceso de indagación con comandos HCI

Se va a asumir que un PC hará una indagación (haciendo uso de un accesorio Bluetooth conectado a uno de sus puertos ó bien incorporado en su tarjeta principal) para saber que periféricos encuentra cercanos. El encontrará un ratón y una impresora Bluetooth. Para entender mejor el proceso, puede estudiarse la Carta de Secuencia 3.3 en conjunto con el *punto 3.2.2.4 de indagación de banda base* . Obsérvese que se han dividido las dos entidades en el portátil que intercambian tramas HCI. Por otro lado, en los periféricos que responden a la indagación solamente es necesario ver sus interfaces físicas, puesto que al responder ellos a la indagación no generan eventos que atraviesen sus interfaces HCI.

Cuando la aplicación del portátil lo indique (puede ser algún proceso constante de búsqueda de dispositivos) envía el comando HCI_Inquiry con los parámetros necesarios (LAP - 24 bits de dirección para generar el GIAC ó DIAC, Longitud - el tiempo durante el cual se hará la indagación y el número máximo de respuestas a recibir), y esto hace que el Firmware empiece a procesar dicho comando, enviando una confirmación de ello; en caso de no haberse podido ejecutar la indagación solicitada, la variable *Status* devuelta en el evento HCI_Command_Status_Event tendrá un código de error que indicará acerca de la razón de no ejecución del comando.

Si se sigue el flujo normal de funcionamiento tendremos que el Hardware Bluetooth queda programado gracias a este comando y desde ahí se encarga de enviar paquetes ID (véase la sección de Nivel de Banda Base) en la secuencia de saltos de indagación; este paquete se envía periódicamente sin importar cuál de los periféricos lo reciba. Cuando los periféricos reciben este paquete, sus procesadores de banda base sabrán que tienen que esperar un tiempo aleatorio para volver a escuchar indagaciones (con el fin de que sus respuestas no se produzcan al mismo tiempo y colisionen). Después de esperar dicho tiempo aleatorio (diferente para cada uno), los periféricos vuelven a escuchar indagaciones y responden inmediatamente; en este caso el mouse responde primero y luego la impresora.

Ver página siguiente...



Carta de Secuencia 3.2. Proceso de indagación con comandos HCI

Con cada respuesta de los periféricos el Hardware Bluetooth genera un evento HCI_Inquiry_Result que contiene: la dirección física Bluetooth del dispositivo (ó dispositivos) que respondieron, la clase de dispositivo, el Offset del reloj maestro y parámetros de paginación (que servirán si se realiza posteriormente una paginación con alguno de estos dispositivos). La indagación puede terminar de tres maneras:

- Cuando se cumple el parámetro especificado al principio de número máximo de respuestas. En este caso puede haberse establecido a solo 2. El evento que se genera una vez terminada la indagación es el de HCI_Inquiry_Complete_Event.
- Cuando el parámetro de tiempo de indagación establecido en la invocación al principio se cumple. El evento que se genera una vez terminada la indagación es el HCI_Inquiry_Complete_Event.
- Si se cancela la indagación por medio de un comando HCI_Inquiry_Cancel, lo cual hace que se retorne un evento HCI_Command_Status_Event al terminar la cancelación.

Cuando la indagación haya terminado, el PC tendrá en memoria las direcciones de los dispositivos que encontró y podrá proceder a realizar conexiones con ellos en cualquier otro momento por medio de los parámetros de paginación que le fueron enviados.

Se podría continuar enriqueciendo el ejemplo anterior con el establecimiento simple de un enlace ACL entre dos dispositivos. Supóngase el mismo computador portátil y la Impresora, que una vez conociendo sus parámetros, pueden establecer conexiones entre ellos.

Los enlaces ACL son necesarios antes de realizar otras tareas; hay comandos de control de enlace que no necesitan de una conexión previa para poderse realizar, tal es el caso de los comandos HCI_Remote_Name_Request, HCI_Inquiry y HCI_Periodic_Inquiry_Mode; pero para los demás comandos de control de enlace debe existir un enlace ACL previo con el fin de que sirva como canal de control; de ahí resulta la importancia de este ejemplo.
Primeramente, el portátil envía al hardware el comando HCI_Create_Connection, con los parámetros:

- Dirección física Bluetooth del dispositivo al cual se va a conectar (en este caso la impresora).

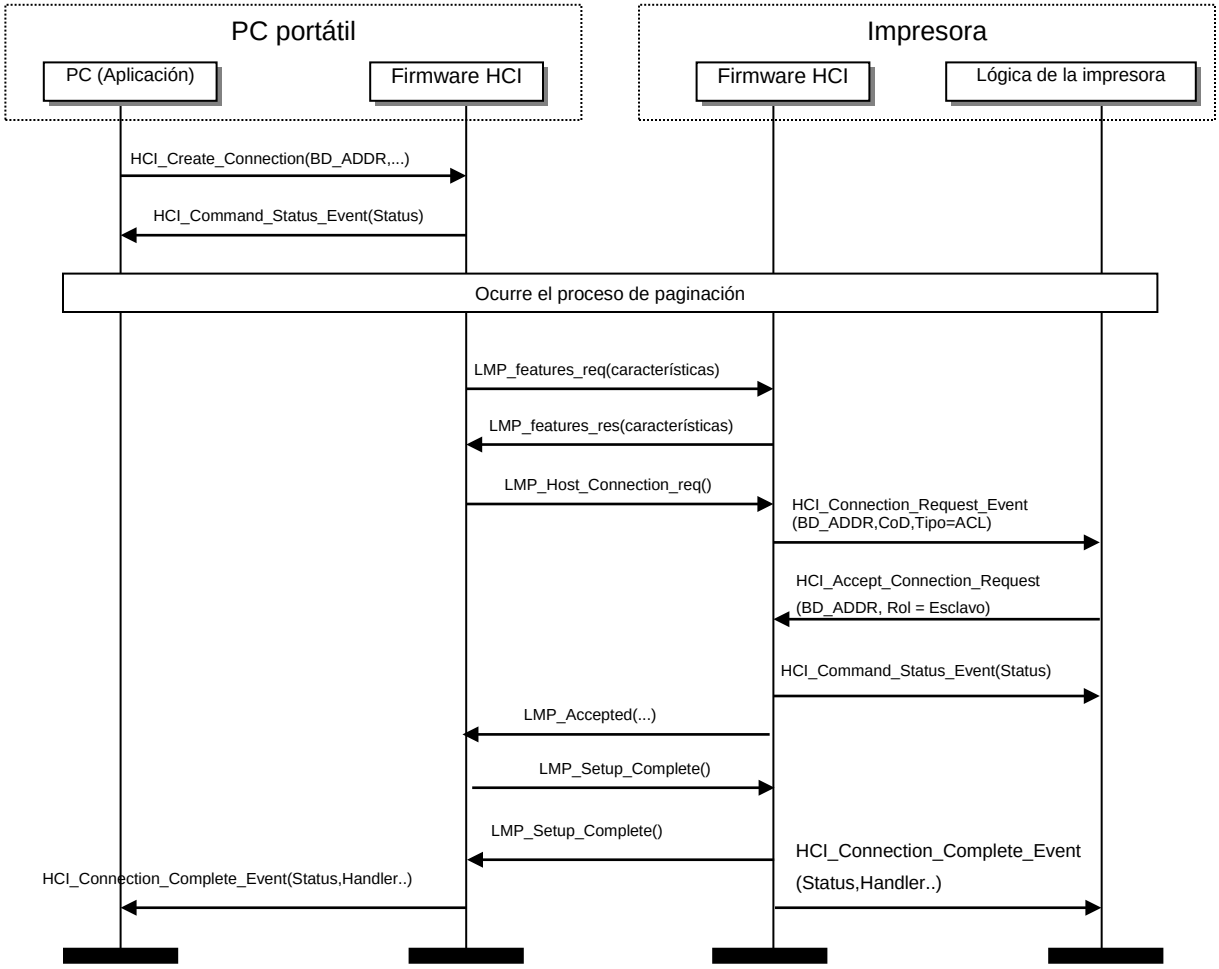
- Tipo de paquete (y por consiguiente, el tipo de conexión, en este caso el tipo de paquete sería uno de los tipos de paquetes asíncronos ACL). (ver tipos de paquetes punto 3.2.4.2 de Banda Base).
- Parámetros de escaneo de paginación.
- Si permite ó no el intercambio de papel (Maestro - Esclavo).
- El Offset ó desplazamiento de su reloj local.

De esta forma el Hardware responde con un evento HCI_Command_Status_Event, y de así se inicia el proceso de paginación, necesario para poder empezar a realizar conexiones; el proceso involucra los mensajes requeridos entre los Link Managers de ambos equipos.

Posteriormente las capas de *Link Manager* negocian parámetros ó características de la conexión que se proponen establecer, es decir, negocian los tipos de paquetes, ó si se permite intercambio de papel (maestro - esclavo), ó si se va a hacer encriptación (entre los más básicos). Dependiendo de estos parámetros la negociación se extiende, puesto que se tendrían que realizar más operaciones para configurar la conexión.

Luego de la negociación, el nivel de Link Manager del equipo portátil solicita al Link Manager de la impresora una conexión. La impresora puede aceptar ó no dicha solicitud. La impresora podría en un caso dado tener restringidos los dispositivos que puedan conectarse a ella y evitar de esta manera ser utilizada por equipos ajenos ó “no confiables”. Asumiendo que responde afirmativamente a la conexión, asume el papel de elemento esclavo y este hecho hace que se finalice la configuración de la conexión, dada de ambos lados por el mensaje LMP_Setup_Complete.

Ver página siguiente...



Carta de Secuencia 3.3. Establecimiento de un enlace ACL

Cuando el portátil y la impresora ya han confirmado el éxito de la conexión, entonces el firmware genera a cada lado el evento HCI_Connection_Complete_Event, el cual tiene como parámetros :

- La variable Status indicando que se realizo la conexión exitosamente.
- Un *Connection Handler* de 2 bytes que identifica a la conexión dentro del contexto del firmware HCI (este concepto esta definido en el punto 3.5.2.3 de Mensajes de Datos).
- La dirección física Bluetooth del dispositivo al que se está conectado
- Tipo de enlace (ACL / SCO, en este caso ACL puesto que es el primero que se realiza antes de formar un enlace SCO).

- El modo de encriptación (sin encriptación, encriptación solo para enlaces punto a punto, ó también encriptación para todos los enlaces incluyendo los de paquetes Broadcast)

Con esta información, puede ya tenerse un enlace ACL y además de servir para transferencia de datos de capas superiores, sirve para configurar y agregar conexiones SCO, por ejemplo, por medio del comando Add_SCO_Connection.

3.5.4 Conclusión del funcionamiento de la interfaz

Dada la estructura de la tramas, puede observarse que en realidad la forma de trabajo del esquema de mensajes de la interfaz HCI se basa en la invocación de funciones del Firmware, invocaciones completas identificadas por un código y cuya información la constituyen los parámetros de la invocación. Las invocaciones están debidamente serializadas para lograr abrirles paso a través de la interfaz física que las transportará hacia el Hardware HCI. Existe una contraparte firmware en el módulo Bluetooth llamada Host Controller, que se encarga de procesar los comandos y generar los eventos necesarios. Es un nodo de procesamiento que indica al link manager y a la banda base qué hacer para completar la ejecución de los comandos.

3.6 PROTOCOLO DE DESCUBRIMIENTO DE SERVICIOS (SERVICE DISCOVERY PROTOCOL SDP)

3.6.1 Introducción

A medida que las redes de computo crecen y plantean nuevos modelos de comunicación, cobra gran importancia la forma en que un servicio que está disponible en la red pueda encontrarse y comenzar a utilizarse. Adicional a estas consideraciones existen otros aspectos como la consecución de protocolos, métodos de acceso y "*drivers* "requeridos para hacer uso de un servicio así como también los procedimientos relacionados con el control de acceso al servicio, la construcción del servicio entre otros, que representan un problema a la hora de determinar la flexibilidad y versatilidad de una red. Por las razones anteriormente expuestas, muchas compañías y grupos de estandarización se dieron a la tarea de

establecer un procedimiento que permitiera resolver este inconveniente. El resultado de su trabajo dio origen al *Descubrimiento de Servicios*; este es un proceso por medio del cual los dispositivos en una red pueden localizar servicios disponibles, extraer información respecto a su operación y hacer uso de ellos. En redes tradicionales los servicios pueden ser gestionados y configurados estáticamente por un administrador de red; sin embargo en el caso de las redes móviles que se forman espontáneamente, es necesario establecer un tipo de solución dinámica y flexible, si se tiene en cuenta la forma en que operan estas redes. Por ejemplo, es fundamental que unidades en un rango de proximidad sean capaces de configurarse por si solas, es decir, que los dispositivos y servicios se descubran unos a otros y negocien las parámetros requeridos para su comunicación, sin ninguna intervención manual.

En este orden de ideas, el SIG desarrolló su propio protocolo de descubrimiento de servicios, optimizado para un entorno Bluetooth, en lugar de adoptar un SDP ya existente.
Entre los objetivos que el grupo de desarrollo pretendían alcanzar con la creación de este protocolo, se cuentan:

- Simplicidad y Versatilidad: Puesto que este protocolo se emplea en la mayoría de los casos de uso de Bluetooth, es deseable que su nivel de procesamiento sea tan bajo como sea posible. Además ya que muchos otros caso de uso están siendo desarrollados por el SIG, es necesario que el SDP sea extensible y lo suficientemente versátil para que pueda acomodarse a nuevos servicios.
- Compacto: Ya que el servicio de descubrimiento, es una operación que se realiza posterior al establecimiento del enlace, el tráfico SDP en la interfaz de aire debe ser muy reducido; esto con el fin de que el proceso de comunicación no se prolongue innecesariamente.
- Habilidad en la búsqueda de Servicios: Un usuario o un dispositivo debe estar en capacidad de localizar rápidamente un servicio, cuando conoce exactamente lo que está buscando (Conoce características particulares del servicio o instancias específicas de el). En caso contrario, es de gran ayuda conocer mediante un proceso de *Browsing*, cuáles son los servicios generales que ofrece un dispositivo.

3.6.2 Descripción del Protocolo

El *Protocolo de Descubrimiento de Servicios SDP,* proporciona un proceso específico para el descubrimiento de servicios, en un entorno de red Bluetooth. Está optimizado para satisfacer

la naturaleza dinámica de las comunicaciones Bluetooth y se enfoca en el descubrimiento de los servicios que esta disponibles en un grupo de unidades Bluetooth, más no en la forma y tampoco define los métodos que permiten acceder a ellos.
Alguna de las capacidades específicas del protocolo, se mencionan a continuación:

1. SDP proporciona la habilidad a los clientes de buscar los servicios que necesita, basándose en la clase de servicio y atributos específicos de estos servicios.
2. SDP habilita un *Browsing (*navegador) de servicios sin un conocimiento previo de las características de los servicios.
3. SDP está dotado de los medios que permiten el descubrimiento de nuevos servicios que se hacen disponibles cuando los dispositivos entran en proximidad, así como cuando un nuevo servicio se hace disponible en un dispositivo que está en la proximidad de RF. Estos medios también se aplican en el caso de que un dispositivo salga del área de cobertura o un servicio deje de estar disponible en un dispositivo.
4. SDP permite que un dispositivo descubra un servicio en otro el dispositivo sin consultar un tercer dispositivo.
5. SDP debe ser apropiado para el uso en dispositivos de capacidades limitadas.
6. SDP proporciona un mecanismo para descubrir incrementalmente, la información de los servicios que ofrece un dispositivo, si se cree que esto minimiza la cantidad de datos que deben intercambiarse para determinar que un servicio particular, no es el que un cliente requiere.
7. SDP permite el descubrimiento y uso de servicios que proporcionan el acceso a otros protocolos de descubrimiento de servicios.
8. SDP debe soportar la creación y definición de nuevos servicios sin requerir el registro de una autoridad central.

3.6.3 Esquema de Comunicación Cliente Servidor

El SDP opera bajo un esquema de comunicación Cliente-Servidor. El cliente es la entidad que busca los servicios y el servidor es el proveedor de servicios. La comunicación entre estas dos entidades se realiza por medio de peticiones y respuestas entre ellas.

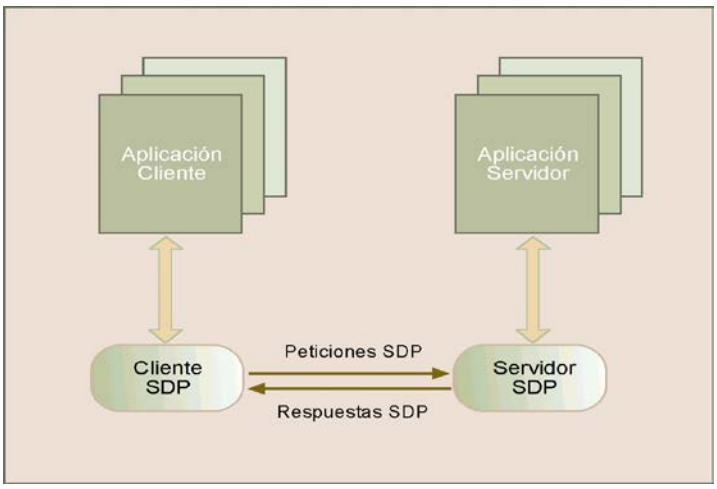


Figura 3.23. Interacción entre cliente y servidor SDP

Un dispositivo puede asumir cualquiera de los dos papeles en un tiempo dado. Sin embargo solo puede existir un servidor SDP en un dispositivo. Si este último posee múltiples aplicaciones, un solo servidor SDP puede actuar en nombre de todas ellas, para dar a conocer los servicios que prestan. El mismo procedimiento también es valido para el caso del Cliente SDP.

El número de *servidores SDP* que se encuentran disponibles para ser utilizados por un cliente SDP, puede variar de acuerdo con la cercanía de un dispositivo, ya que cada elemento de una Piconet puede modificar su grado de asociación a la red y por lo tanto, habrá casos en que se pierda la comunicación con un servidor SDP debido a la separación del dispositivo, mientras que en otras circunstancias, la presencia de servidores SDP aumentará por la inclusión en la red de una nueva unidad.

3.6.4 Servicios en SDP

La especificación define *SERVICIO* como "*Una entidad que está en capacidad de brindar información, ejecutar una acción, o controlar un recurso en nombre de otra entidad. Un servicio puede ser implementado en software, hardware o ambas formas*".

3.6.4.1 La Clase Servicio

Cada servicio es una instancia de una clase. La definición de la *Clase Servicio* proporciona las definiciones de todos los atributos que describen un servicio. Cada Clase de servicio tiene asignado un identificador único (representado por un UUID) contenido en el valor del atributo *ServiceClassIDList*. Cuando se define una clase de servicio como subclase de otra existente, la clase nueva hereda todos los atributos de la clase madre y además pueden definírsele otros atributos diferentes.

Con el fin de que un servidor SDP pueda suministrar la información de los servicios que presta una aplicación, tiene a su cargo un registro donde almacena la información que describe a cada servicio. Este registro recibe el nombre de *Registro de Servicios (Service Registry)*. La Figura 3.24 ilustra la estructura general del Registro del Servicios SDP.

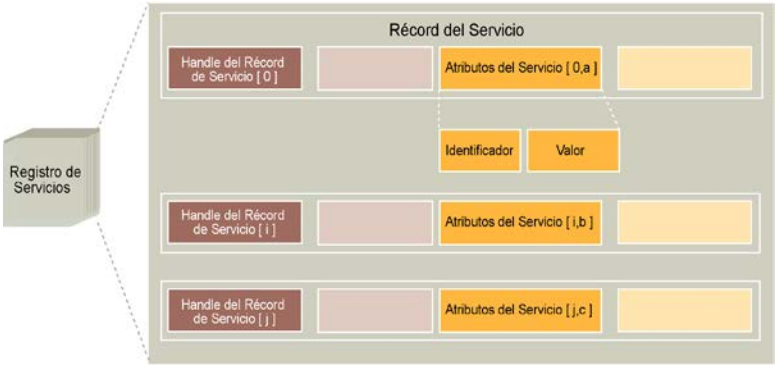


Figura 3.24. Estructura del registro de servicios SDP

Cada servicio incluido en el *Registro de servicios*, se describe por medio de una entidad denominada *Récord de Servicio.*

3.6.4.2 Récord de Servicio

Un Récord de Servicio está formado por una colección de atributos que describen una clase de servicio (como por ejemplo un servicio de FAX, audio o impresión), contienen información acerca de los protocolos que el servicio necesita para poder ejecutarse y otro tipo de

información destinada a los usuarios finales, para que estos puedan comprender en su lenguaje cuáles son las funcionalidades que incorpora un servicio.

Service Record Handle: Es un número de 32 bits que identifica un *Récord de Servicio* en el *Registro de Servicios* de un *servidor SDP*. Si dos servidores SDP pertenecientes a dispositivos diferentes, contienen *Récord De Servicio* idénticos (el mismo servicio), el *Service Record Handle* de cada uno de ellos es completamente independiente del otro y no tiene significado para el *servidor SDP* contrario. Solamente existe un *Service Record Handle,* cuyo significado es consistente para todos y cada uno de los servidores SDP en una Piconet*. Este Service Record Handle tiene como valor 0x00000000* y representa el *Récord del Servicio* del *servidor SDP* mismo. El *récord* de este servicio contiene los atributos del *servidor SDP* y la clase de protocolo que este soporta.

- Atributos de Servicio: Un Atributo de Servicio describe una característica simple de un servicio. La Figura 3.24 ilustra el formato general de un Atributo de Servicio.

- *Identificador de Atributo* (*ID Atribute):* Es un entero sin signo de 16 bits que permite distinguir un atributo de servicio dentro de un *Récord de Servicio*. La definición de una *Clase de Servicio* especifica cada uno de los *Identificadores de Atributo* y establece un significado al valor del *Atributo* asociado con un *Identificador de Atributo*.
  Los Identificadores de Atributos en SDP se tratan como elemento de datos.

- Valor del Atributo *(Value Atribute):* El valor de un atributo*,* es un campo de longitud variable cuyo significado está determinado por el *Identificador de Atributo* que se le asocia. En SDP el *Valor de un Atributo* se trata como elemento de datos.

Los Atributos de un Servicio pueden ser de dos tipos:

- Atributos de Servicio Universales: Son aquellos atributos cuyas *definiciones* son comunes para todos los *Récord de Servicios*. Sin embargo, esto no significa que cada Récord de Servicios deba asignar valores a estos atributos. Existen dos tipos de atributos que obligatoriamente deben hacer parte de un *Récord de Servicios*; estos son: El *Service Récord Handle* (el cual tiene como Identificador de Atributo, el valor 0x0000) y *Service Class ID List* (el cual tiene como Identificador de Atributo, el valor 0x0001). Este último atributo es una lista que identifica el tipo de servicio representado por un

Récord de Servicio, en otras palabras se puede decir que es una lista de clases de las que, un servicio es un instancia.

- Atributos de Servicio Específicos: Son aquellos atributos de un servicio, que son relevantes solamente para una clase o instancia específica de un servicio. Un ejemplo de esta clase de atributos puede ser el "*Control Remoto de Volumen*" para un servicio de *Headset o Manos Libres.*

3.6.5 Paquetes SDP

Cada SDP_PDU *Service Discovery Protocol_ Protocol_Data_Unit* se compone de una cabecera, seguida por parámetros específicos asociados al PDU.

Cabecera

La cabecera está constituida por tres campos que proporcionan información relacionada con la transacción, de la siguiente forma:

- Identificador *ID de PDU*: Identifica el tipo de PDU, su significado y sus parámetros específicos.
- Identificador del tipo de Transacción (*ID Transaction)*: Este parámetro solamente identifica PDU del tipo petición (*Request*) y se emplea para asignar una respuesta a su petición correspondiente. Rango 0x0000 – 0xFFFF.
- Longitud (*Length)*: Especifica la longitud en bytes de todos los parámetros contenidos en el PDU. Rango 0x0000 – 0xFFFF.

Parámetros

Conformado por una lista de parámetros que se establece según el tipo de SDP_PDU.
La Figura 3.25 ilustra el formato de un SDP_PDU:

Ver página siguiente…

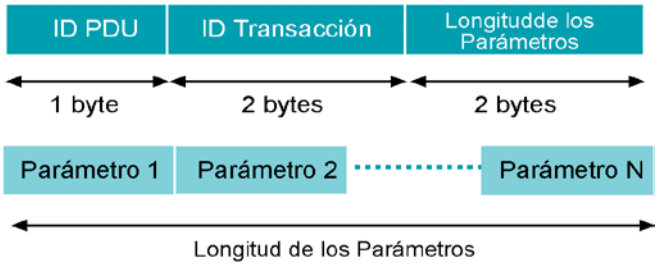


Figura 3.25. Formato de SDP_PDU

3.6.5.1 PDU_SDP Especiales

3.6.5.1.1 Respuesta Parcial y Estado de Continuación

Algunos PDU empleados para realizar peticiones pueden requerir respuestas que por su tamaño, no alcanzan en los paquetes destinados para tal fin. En estos casos el servidor SDP genera una respuesta parcial que contiene un parámetro denominado *Estado de Continuación*. Cuando una respuesta de este tipo arriba al cliente SDP, este puede devolver el parámetro *Estado de Continuación,* re-transmitiendo la petición original (con un identificador de transacción diferente) para solicitar al servidor SDP el resto de la respuesta.

3.6.5.1.2 Soporte para Errores

Cada transacción consiste de una petición y una respuesta que se relacionan entre si por un identificador de transacción. Sin embargo cuando un servidor SDP determina que ha recibido una petición inválida, no responde con la respuesta correspondiente sino que envía una PDU llamada SDP_Error_Response

3.6.6 Transacciones

El *Protocolo de Descubrimiento de Servicios* SDP, emplea un modelo de Petición y Respuesta para realizar las transacciones de búsqueda de servicios. En aquellos casos

donde el SDP utiliza L2CAP como protocolo de transporte, una sola petición o respuesta puede ser difundida en un instante dado; esto quiere decir, que un cliente SDP debe primero recibir la respuesta a cada petición que hace, antes de enviar una nueva petición.

3.6.6.1 Búsqueda de Servicios

El descubrimiento de servicios en un entorno Bluetooth se ejecuta a través de las siguientes transacciones:

1. Un cliente SDP envía un petición para buscar servicios de interés; el servidor SDP responde con los respectivos *Handles* de los servicios que coinciden con los requerimientos de la petición.
2. El cliente utiliza los Handles obtenidos en la primera transacción y envía una petición para conseguir los atributos de un servicio en especial. El servidor SDP devuelve una respuesta con los atributos del servicio solicitado.

Las SDP_PDU empleadas en las transacciones anteriores se describen a continuación.

3.6.6.1.1 Petición para Búsqueda de Servicios SDP_Service_Search_Request.

Un cliente SDP genera un *SDP_ServiceSearchRequest* para localizar *Récord de Servicios* que coincidan con un *Patrón de Búsqueda de Servicios* incluido como parámetro en este SDP_PDU.

- Identificador de PDU: 0x02.
- Parámetros:

  Patrón de Búsqueda de Servicios (Service Search Pattern):

Tabla 3.30. Descripción del parámetro patrón de búsqueda de servicios

VALOR	DESCRIPCIÓN DEL PARÁMETRO
Secuencia de Elementos de Datos	Cada elemento de la secuencia es un UUID. El número mínimo de UUID es 1 y el número máximo es 12

Contador máximo de Récord de Servicios (Maximum Service Record Count):

Tabla 3.31. Descripción del parámetro contador máximo de récord de servicios

VALOR	DESCRIPCIÓN DEL PARÁMETRO
N. Rango = 0x0001 – 0xFFF	Es un contador de 16 bits que especifica en número máximo de *Service Record Handles* que un servidor SDP debe retornar en la respuesta a la petición

Estado de Continuación (Continuation State):

Tabla 3.32. Descripción del del parámetro estado de continuación

VALOR	DESCRIPCIÓN DEL PARÁMETRO
Estado de Continuación	Es un contador de 8 bits del número de bytes de la información contenida en un mensaje Estado de Continuación, seguido por los N bytes de información que se retornaron en un respuesta previa del servidor SDP. Si no hay Estado de Continuación, este contador debe ponerse en 0

3.6.6.1.2 Respuesta a una petición de Búsqueda SDP_Service_Search_Response.

El servidor SDP genera un *SDP_Service_Search_Response* que contiene los *Service Record Handle* de su Récord de Servicios que coinciden con el patrón de búsqueda señalado en la petición. Si se genera una respuesta parcial, esta debe contener un número entero que representa el número total de *Service Record Handle* que devolverá.

Los *Service Record Handle* no pueden ser cortados para ser enviados en múltiples SDP_PDU.

- Identificador de PDU = 0x03.
- Parámetros:

Contador de Récord de Servicio Totales (Total Service Record Count):

Tabla 3.33. Descripción del parámetro contador de récord de servicio totales

VALOR	DESCRIPCIÓN DEL PARÁMETRO
N Rango 0x0000 – 0XFFFF	Entero que contiene el número total de Récord de Servicio que coinciden con el patrón de búsqueda solicitado en la petición. Cuando existen múltiples respuestas a una petición este valor debe ser el mismo para cada respuesta parcial.

Contador de Récord de Servicio Actual (Current Service Record Count):

Tabla 3.34. Descripción del parámetro contador de récord de servicio actual

VALOR	DESCRIPCIÓN DEL PARÁMETRO
N Rango = 0x0000 – 0XFFFF	Es un entero que indica el número de Service Record Handle contenidos en el próximo paquete.

Lista de Sevice Record Handle (Service Record Handle List):

Tabla 3.35. Descripción del parámetro lista de sevice record handle

VALOR	DESCRIPCIÓN DEL PARÁMETRO
Lista de Handle de 32 bits	Contiene una lista de Service Record Handles, donde cada elemento de lista coincide con un patrón de búsqueda solicitado en la petición.

Estado de Continuación (ContinuationState): Ver descripción del parámetro en SDP_Service_Search_Request.

3.6.6.1.3 Petición de los Atributos de un Servicio DP_Service_Atrtibute_Request:

Un cliente SDP genera un *SDP_Service_Atrtibute_Request* para conocer valores de los atributos especificados por un *Récord de Servicio* particular.

- Identificador de PDU : 0x04.
- Parámetros:

Handle del Récord de Servicio (Service Record Handle):

Tabla 3.36. Descripción del parámetro handle del récord de servicio

VALOR	*DESCRIPCIÓN DEL PARÁMETRO*
Handle de 32 bits	Especifica el Récord de Servicio del cual se desea conocer atributos. Este handle se obtiene por medio de una transacción *SDP_Service_Search_Request* previa.

Contador de bytes de Atributos Máximo (Maximum Attribute Byte Count):

Tabla 3.37. Descripción del parámetro contador de bytes de atributos máximo

VALOR	DESCRIPCIÓN DEL PARÁMETRO
N Rango 0x0007 – 0XFFFF	Especifica el número máximo de bytes de la información de un atributo, que debe ser devuelto en un respuesta a esta petición .

Identificador de la Lista de Atributos (AttributeIDList):

Tabla 3.38. Descripción del parámetro identificador de la lista de atributos

VALOR	DESCRIPCIÓN DEL PARÁMETRO
Secuencia de elementos de datos. Rango 0x0000 – 0XFFFF	En esta lista cada elemento representa una Identificador de Atributo o un rango de Identificadores de Atributo

Estado de Continuación (ContinuationState): Ver descripción del parámetro en SDP_Service_Search_Request.

3.6.6.1.4 Respuesta a la Petición de los Atributos de un Servicio

SDP_Service_Atrtibute_Response:

El servidor SDP generará un *SDP_Service_Atrtibute_Response* después de recibir un *SDP_ServiceAttributeRequest* válido. La respuesta contiene una lista de atributos correspondientes al *Récord de Servicio* solicitado.

- Identificador de PDU = 0x05.

- Parámetros

Contador de los bytes de la Lista de Atributos (Attribute List Byte Count):

Tabla 3.39. Descripción del parámetro contador de los bytes de la lista de atributos

VALOR	DESCRIPCIÓN DEL PARÁMETRO
N Rango 0x0002 – 0XFFFF	Contiene un contador del número de bytes que tiene el parámetro Lista de Atributos. *Ver siguiente*

Lista de Atributos (Attribute List):

Tabla 3.40. Descripción del parámetro lista de atributo

VALOR	DESCRIPCIÓN DEL PARÁMETRO
Secuencia de Elementos de datos	Contiene la lista de Atributos (Identificadores de los Atributos y sus respectivos valores) que hacen parte del Récord de Servicio indicado en la petición.

Estado de Continuación (ContinuationState): Ver descripción del parámetro en SDP_ServiceSearchRequest.

3.6.6.1.5 Petición para búsqueda de Servicios y sus Atributos

SDP_Service_Search_Atrtibute_Request:

Una transacción *SDP_ServiceSearchAtrtibuteRequest* combina las capacidades de las PDU *SDP_ServiceSearchRequest y SDP_ServiceAttributeRequest* en una única petición. Esta SDP_PDU y su correspondiente respuesta tienen un grado de complejidad mayor, sin embargo su uso reduce el número total de transacciones particularmente cuando se tratan múltiples *Récord de Servicio.*

- Identificador de PDU: 0x06.
- Parámetros:

Patrón de búsqueda del Servicio (ServiceSearchPattern): Ver descripción del parámetro en SDP_ServiceSearchRequest

Contador de bytes de Atributos Máximo (MaximumAttributeByteCount): Ver descripción del parámetro en SDP_ServiceAtrtibuteRequest:

Identificador de la Lista de Atributos (AttributeIDList): Ver descripción del parámetro en SDP_ServiceAtrtibuteRequest:

Estado de Continuación (ContinuationState): Ver descripción del parámetro en SDP_ServiceSearchRequest.

3.6.6.1.6 Respuesta a la petición para Búsqueda de Servicios y su Atributos SDP_Service_Search_Atrtibute_Response:

El servidor SDP genera un *SDP_ServiceSearchAtrtibuteResponse* después de recibir un *SDP_ServiceSearchAttributeRequest* válido. La respuesta contiene una lista de atributos de un *Récord de Servicio* que coinciden con el patrón de búsqueda requerido.

- Identificador de PDU = 0x02.
- Parámetros

Contador de los bytes de la Lista de Atributos (AttributeListByteCount): Ver descripción del parámetro en SDP_ServiceAttributeResponse.

Lista de Atributos (AttributeLists): Ver descripción del parámetro en SDP_ServiceAttributeResponse.

Estado de Continuación (ContinuationState): Ver descripción del parámetro en SDP_ServiceSearchRequest.

3.6.7 Perfil para Descubrimiento de Servicios (Service Discovery Protocol Profile SDPP)

El *Perfil para Descubrimiento de Servicios* define los protocolos y procedimientos que deben ser implementados por una aplicación, para localizar servicios en otros dispositivos utilizando el protocolo para descubrimiento de servicios (SDP) que proporciona el stack de Bluetooth. El *SDPP* describe un modelo de aplicación general y define las abstracciones de las primitivas de servicio que permiten la creación de APIS. El *SDPP* especifica como debe crearse una aplicación que hace uso del SDP y la forma en que esta debe trabajar para llevar a cabo los procedimientos asociados con la búsqueda de servicios. Además define el modo en que otros perfiles de aplicación, deben utilizar los protocolos de transporte de Bluetooth para transferir SDP_PDU cuando necesitan ejecutar transacciones SDP.

3.6.7.1 Arquitectura de protocolos

La Figura 3.26 ilustra la arquitectura de protocolos y las entidades que comprenden el perfil para descubrimiento de servicios.

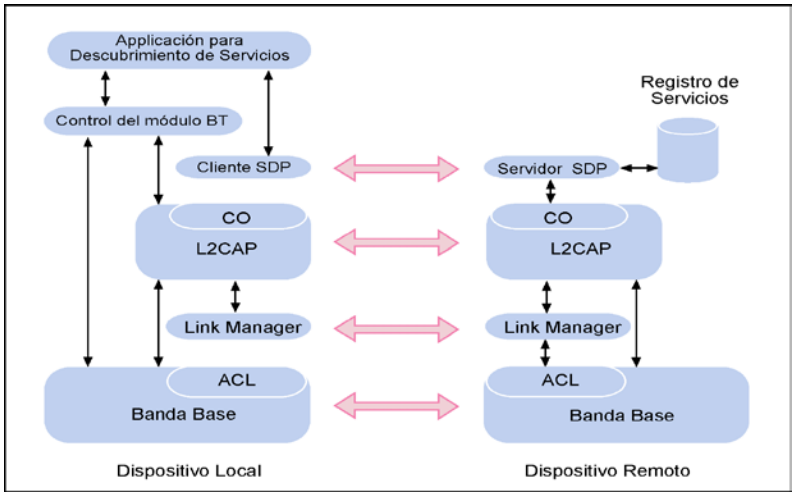


Figura 3.26. Arquitectura de protocolos y entidades que componen el SDPP

Dispositivo Local *LocDev* : Es el dispositivo donde se encuentra alojada la *aplicación para el descubrimiento de servicios* y quien inicia el procedimiento para tal fin. El dispositivo local debe incluir la porción correspondiente al *Cliente SDP* del protocolo SDP.

Dispositivo Remoto *RemDev* : Es cualquier dispositivo que participa en los procedimientos para descubrimiento de servicios, respondiendo las peticiones que hace un dispositivo local, es decir, que este es el dispositivo del cual desea conocerse qué servicios tiene disponibles. El dispositivo Remoto debe incluir la porción correspondiente al *Servidor SDP* del protocolo SDP, así como la base de datos (*Registro de Servicios*)que almacena la información correspondiente a cada servicio.

Un dispositivo Remoto puede clasificarse de acuerdo con el grado de confianza que represente para un dispositivo local, así:

Dispositivo de confianza: Son dispositivos que en el momento no se encuentran conectados al dispositivo local, pero guardan una relación de confianza con este.
Dispositivos Desconocidos: (Nuevos dispositivos) Son dispositivos que jamas han establecido conexión con el dispositivo local.
Dispositivos Conectados: Son dispositivos que se encuentran conectados a la red y con los cuales el dispositivo local ha establecido un enlace.

El papel de un dispositivo remoto o dispositivo local no es ni permanente ni exclusivo para un dispositivo, el cual podrá asumir cualquiera de los dos; e incluso al mismo tiempo, según se presenten las circunstancias de comunicación entre dos unidades Bluetooth.

Aplicación para Descubrimiento de Servicios *SrvDscApp* : La SrvDscApp que se encuentra en un dispositivo local se comunica con el cliente SDP para enviar peticiones a un servidor SDP presente en un dispositivo remoto, del cual recibirá respuestas que contienen información acerca de los servicios requeridos.

Base de Datos "Registro de Servicios": Es una entidad lógica que sirve de repositorio a información relacionada con el descubrimiento de servicios.

3.6.7.2 Escenarios de Aplicación

Los escenarios de aplicación que cubre este perfil son:

- **Búsqueda de servicios**: Por medio de esta operación, un cliente puede recuperar los *Service Record Handles* de un *Record de Servicio* particular, basado en características conocidas del servicio, como por ejemplo la clase del servicio o los atributos del mismo. Es importante que los atributos empleados para localizar un servicio, estén representados por UUID.

- **Browsing de servicios**: En ocasiones, es deseable averiguar cuales son los tipos de servicios que ofrece un dispositivo, sin tener conocimiento previo de las características que los describen. Este proceso recibe el nombre de *Browsing de Servicios* y emplea un mecanismo de búsqueda, basado en un atributo compartido por todas las clases de servicios, llamado *BrowseGroupList*. El valor de este atributo contiene un lista de UUID que representan un grupo de búsqueda denominado *Browse Group*, al cual se asocian servicios con el propósito de que puedan ser localizados por este procedimiento. Cuando un cliente desea hacer "Browsing" de un servicio, crea un patrón de búsqueda de servicios que contiene el UUID que representa al Grupo raíz de Browse *Browse Group Root*. Todos los servicios susceptibles de "Browsing" en el nivel superior, son miembros del *Browse Group Root*, y poseen un identificador tipo UUID en el atributo BrowseGroupList que los identifica como tales. Los servicios ofrecidos por un servidor SDP pueden ser organizados en una *Jerarquía de Grupos para Browsing*, definiendo Grupos de Browse adicionales por debajo del *Browse Group Root*. Cada uno de estos grupos adicionales se describe por el *Record de Servicio* de una clase llamada Descriptor del Grupo Raiz de Browse *Browse Group Root Descriptor*. En este *Record de Servicio* se encuentran los Identificadores (Atributo *Group ID*) de cada uno de los Grupos de Browse adicionales.

El primer caso (Búsqueda de Servicios), responde a la siguiente pregunta ¿Está disponible un servicio A o un servicio A con características B y C?. El segundo caso (Browse de Servicios) representa una búsqueda de servicio general que responde a la pregunta ¿Cuáles servicios se encuentran disponibles?

3.6.7.3 Capa de Aplicación

La Figura 3.27 presenta la forma en que puede implementarse la estructura de operación de una Aplicación para Descubrimiento de Servicios *SrvDscApp*.

1. La Aplicación para Descubrimiento de Servicios *SrvDscApp* activa una operación de INQUIRY, seguida de una petición del usuario para la búsqueda de servicios.
2. Para cualquier Dispositivo Remoto que se encuentre después del INQUIRY se ejecutan las operaciones para descubrimiento de servicios y una vez estas terminen, el Dispositivo Local finaliza su enlace con el dispositivo remoto y está en capacidad de conectarse con un nuevo dispositivo.
3. Si antes de realizar el Inquiry, ya se encontraban conectados otros dispositivos con el Dispositivo Local, este no los desconectará después del descubrimiento de servicios.
4. El usuario de la Aplicación para Descubrimiento de Servicios tiene la opción de conectarse con dispositivos remotos, operando en modos que tienen distintos niveles de *confianza*, así:
    - Con dispositivos remotos confiables, solamente.
    - Con Dispositivos remotos confiables, mas dispositivos remotos que solo requieren la entrada de un PIN de valor cero.
    - Con cualquiera de los dispositivos anteriores, más cualquier Dispositivo Remoto adicional que requiera la entrada de un PIN diferente de cero, por parte del usuario.

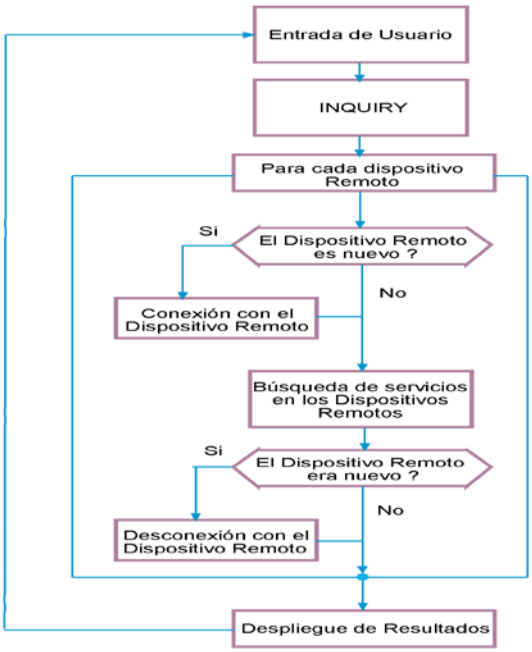


Figura 3.27. Estructura de operación de la SrvDscApp

3.6.7.4 Abstracciones de las Primitivas del Servicio.

La funcionalidad de una *SrvDscApp* se presenta en forma de abstracción de las primitivas de servicio. Esta abstracción proporciona una estructura formal para describir las expectativas que tiene el usuario de la *SrvDscApp*.

La Tabla 3.41 describe las abstracciones de las primitivas de servicio requeridas por el perfil.

Tabla 3.41. Primitivas de servicio para SDPP

PRIMITIVA	RESULTADO
Búsqueda de Servicios Service Browse (LIST(RemDev) LIST(RemDevRelation) LIST(BrowseGroup) GetRemDevName StopRule)	Se realiza un Búsqueda de servicios contenidos en un Grupo de Browse en los servicios que hacen parte de una lista de dispositivos remotos. Además la búsqueda puede calificarse con una lista de parámetros que indican la relación que un dispositivo local mantiene con un dispositivo remoto (RemDevRelation). Cuando el parámetro GetRemDevName se establece en “si”, se retorna el nombre de los dispositivos que soportan los servicios solicitados; la búsqueda continúa hasta que se hace efectiva, una regla de detenimiento de la acción.
Búsqueda de Servicios ServiceSearch (LIST(RemDev) LIST(RemDevRelation) LIST(SearchPattern, AttributeList) GetRemDevName StopRule)	Se realiza una búsqueda de servicios con el fin de saber si un dispositivo que se encuentra en la lista de dispositivos remotos, soporta los servicios coincidentes con un patrón de búsqueda y una lista de Atributos. En este caso, la búsqueda también puede calificarse con una lista de parámetros que indican la relación que un dispositivo local mantiene con un dispositivo remoto (RemDevRelation). Cuando el parámetro GetRemDevName se fija en “si”, se retorna el nombre de los dispositivos que soportan los servicios solicitados. La búsqueda continúa hasta que se hace efectiva, una regla de detenimiento de la acción.
Enumeración de Dispositivo Remotos EnumerateRemDev (LIST (ClassOfDevice) StopRule)	Ejecuta una búsqueda de dispositivos remotos en la vecindad del dispositivo local. La búsqueda puede filtrarse ocasionalmente utilizando la lista de clases de dispositivo. La búsqueda continúa hasta que se hace efectiva, una regla de detenimiento de la acción.
Primitiva de Terminación TerminatePrimitive (primitive Handle return Result)	Se presenta un término para que finalicen las acciones que se llevan a cabo.

La regla de detenimiento (*StopRule*) se emplea para garantizar la culminación de una búsqueda de servicios. La StopRule, podría representar el número total de ítems que se encontraron, así como la duración de la búsqueda.

Todas las primitivas retornan información, solamente cuando esta se encuentra. El resultado puede devolverse a través de la misma función que lo solicita, o por medio de un puntero a la estructura de datos.

La Figura 3.28, ilustra dos tipos de intercambio SDP_PDU realizar la búsqueda de un *Servicio* que tiene como *Atributo*, el soporte para un perfil determinado.

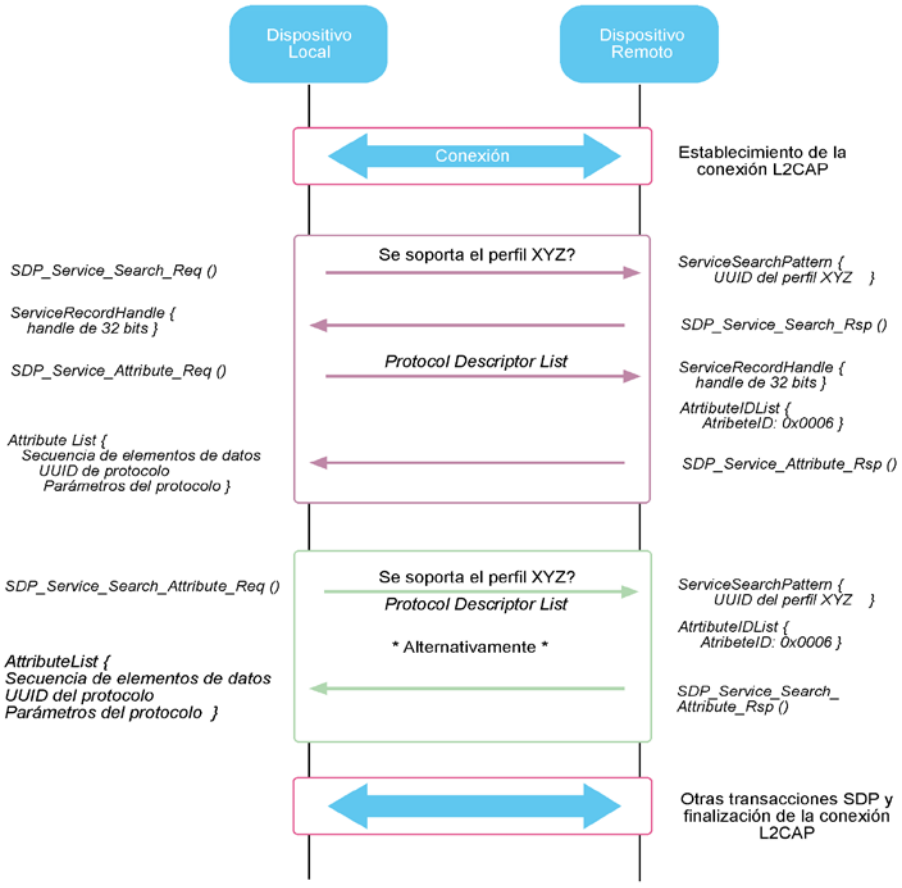


Figura 3.28. Intercambio de SDP_PDU en transacciones de búsqueda de servicios

En la primera alternativa, el dispositivo local envía un *SDP_Service_Search_Request*, el cual contiene un patrón de búsqueda compuesto por UUID asociados a un perfil deseado. Este

último, denotado como “XYZ” se identifica en la Figura 3.28, como “profile_XYZ_UUID”. En la PDU de respuesta, el servidor SDP retorna uno o mas *Service Record Handles* de 32 bits cuyos *Record de Servicio* correspondientes, contienen el *Handle* del atributo “ProtocolDescriptorList” cuyo UUID corresponde al UUID del perfil “XYZ”, en este caso.

En la segunda alternativa el dispositivo local, envía un *SDP_Service_Search_Atribute_Req,* el cual incluye el patrón de búsqueda y los atributos deseados en un servicio. En su respuesta, el servidor SDP devuelva el atributo solicitado, del *Record de Servicio* que coincida con el patrón de búsqueda transportado por la PDU que hizo la petición.

3.6.7.5 Protocolos Empleados por el Perfil Para Descubrimiento De Servicios.

Para implementar este perfil no es necesario utilizar más que los protocolos que se encuentran por debajo del SDP. A continuación se describen funcionalidades particulares de cada uno de estos protocolos, requeridas para la puesta en marcha de este perfil.

3.6.7.5.1 L2CAP

La Tabla 3.42 resume las tareas L2CAP necesarias para realizar las transacciones de descubrimiento de servicios.

Tabla 3.42. Procedimientos L2CAP empleados en el SDPP

PROCEDIMIENTO L2CAP	SOPORTE LOCDEV	SOPORTE REMDEV
Tipo de Canal		
Canales OC	Obligatorio (O)	O
Señalización		
Establecimiento de la Conexión	O	Condicional (C)
Configuración	O	O
Terminación de la conexión	O	C
Echo	O	O
Rechazo de la conexión	O	O
Opciones de Configuración		
Unidad de Transmisión Máxima	O	O
Flush Time Out	O	O
Calidad del Servicio	Opcional (OP)	OP

Aunque que las transacciones SDP comprenden una secuencia de peticiones y respuestas a través del intercambio de SDP_PDU haciendo uso de canales orientados a conexión, el SDP mismo constituye un servicio de datagramas (no orientado a conexión), en el sentido de que no se ha establecido una conexión previa para que se de el intercambio de SDP_PDU. Esta función se delega a la capa L2CAP, pero el SDP tiene la responsabilidad de comunicarle cuando debe terminar una conexión. Debe aclararse que una conexión L2CAP debe establecerse para llevar a cabo más de un solo intercambio de SDP_PDU, puesto que no se justifica crear un canal para transmitir solamente una petición y su correspondiente repuesta.

Una sesión entre un cliente y un servidor SDP, se define por lo tanto, como el intervalo de tiempo en que el cliente y el servidor mantienen presente y continuamente la misma conexión SDP. La sesión SDP termina generalmente cuando se completan las transacciones SDP, sin embargo el usuario puede intervenir para terminar el proceso antes de que se concluya adecuadamente una transacción. Normalmente el dispositivo local da la orden para dar por terminada la sesión SDP.

3.6.7.5.2 Gestionador del Enlace Link Manager.

Las característica LMP que deben implementarse para este perfil son:

Tabla 3.43. Procedimientos LMP empleados en el SDPP

PROCEDIMIENTO LMP	SOPORTE EN LMP	SOPORTE LOCDEV	SOPORTE REMDEV
Autenticación	Obligatorio (O)	Condicional (C)	C
Paring	O		
Gestión de claves	O		
Encriptación	OP	C	C
Cambio de Papel	OP		
Petición del Nombre del dispositivo	O		
Desconexión	O		
Modos de Ahorro de Energía y Control de Potencia	Opcional (OP)		
QoS	O		
Enlaces SCO	OP		
Control de paquetes MultiSlot	O		

No existen requerimientos específicos respecto al papel que debe desempeñar el dispositivo (maestro-esclavo) ni tampoco al modo de consumo de energía cuando se ejecuta de este perfil. Es decisión del *Link Manager* dar soporte al desarrollo de los procesos anteriores siempre que estos sean requeridos.

3.6.7.5.3 Link Control

3.6.7.5.3.1 Inquiry

El dispositivo Local debe advertir a la *Banda Base* de la necesidad de entrar en este modo, siempre que la *SrvDscApp* lo requiera. La entrada en este estado puede ser inmediata o no, dependiendo de los requerimientos de calidad del servicio establecidos en una conexión. El usuario de la *SrvDscApp*, debe establecer el criterio para la determinar la duración del Inquiry. El procedimiento de Inquiry debe ceñirse a lo establecido en el perfil de Acceso Genérico el cual será descrito más adelante.

3.6.7.5.3.2 Paging

Al igual que el procedimiento de Inquiry el dispositivo Local debe advertir a la *Banda Base* de la necesidad de entrar en este modo, siempre que la *SrvDscApp* lo requiera. La entrada en este estado puede ser inmediata o no, dependiendo de los requerimientos de calidad del servicio establecidos en una conexión. El tipo de *Paging* ejecutado por el dispositivo Local depende del tipo de *Paging* que soporte cada uno de los dispositivos Remotos.

3.6.7.6 Esquema General de Operación del SDPP

La siguiente Figura 3.29 ilustra de manera simplificada, las transacciones de las capas protocolares necesarias para realizar los procedimientos de Descubrimiento de Servicios.

Ver página siguiente...

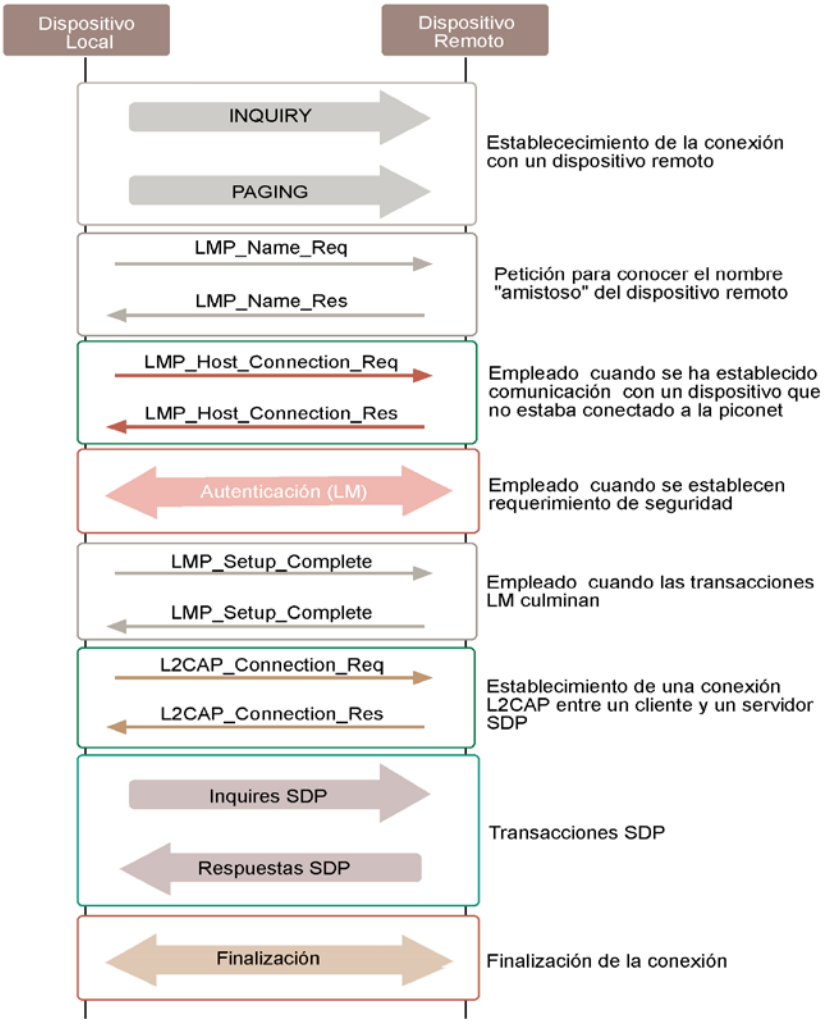


Figura 3.29. Soporte protocolar para el perfil de descubrimiento de servicios SDPP

3.7 RFCOMM

Bluetooth ha sido diseñado con el propósito de reemplazar los cables de datos que existan en variadas aplicaciones; generalmente estos cables conectan dispositivos móviles entre sí ó con otros equipos estáticos. Aplicaciones de este tipo entre las más comunes son:

- Las cámaras web y cámaras digitales, las cuales hacen uso del cable para transferir las imágenes que se han tomado y almacenado temporalmente en ellas a otros dispositivos de almacenamiento, computadores de escritorio ó a un dispositivo que pueda subirlas ó enviarlas a otro lugar a través de una red.
- Los periféricos de un computador, ya sean impresoras, ratones, escáneres, módems que utilizan puertos físicos del equipo para poder ser utilizados.
- Algunos teléfonos móviles tienen puertos de datos.
- Los PDAs[17] poseen un puerto serial que utilizan para sincronizarse (ver Modelo de uso de Sincronización en el capítulo 2) y compartir archivos con otro dispositivo.

Las interfaces que utilizan los dispositivos de estos ejemplos son puertos seriales, lo cual constituye un gran espectro de aplicaciones a cubrir con Bluetooth si este llega a reemplazar dichas interfaces físicas. La capa RFCOMM de la pila de protocolos tiene esta finalidad, hacer que las aplicaciones que actualmente hacen uso de interfaces seriales cableadas, puedan también interactuar con interfaces inalámbricas, que emulan el comportamiento de las interfaces seriales.

Esta protocolo, al igual que el nivel de Descubrimiento de Servicios fueron diseñadas específicamente para Bluetooth.

3.7.1 Requerimientos de RFCOMM

Bluetooth debe garantizar ciertos requerimientos para poder constituir un reemplazo de cables para aplicaciones telemáticas:

- Debe haber multiplexación de las comunicaciones que se lleven a cabo en forma serial, es decir, pueden haber muchas aplicaciones que utilicen un puerto serial soportadas por el mismo dispositivo Bluetooth.
- Debe garantizar que las señales de RS232 sean compatibles al ser emuladas y así dar a apariencia de ser una interfaz RS232.
- Intercambiar parámetros de conexión para poder autoconfigurarse (esto puede hacerse utilizando el SDP).

[17] Asistente Personal Digital - Personal Digital Asistant

3.7.1.1 Señales Emuladas por RFCOMM

Para poder emular un puerto serial, no sería suficiente con emular las señales de transmisión y recepción de datos, sino que también si se quiere ampliar la gama de dispositivos y de aplicaciones para soportar, se debe emular tanto las señales de datos como las de control. RFCOMM se encarga de esta tarea. Dichas señales son:

Tabla 3.44. Señales RS232 emuladas por RFCOMM

Transmit Data	TD	Señal que la aplicación usa para enviar datos
Received Data	RD	Señal por la que llega información hacia la aplicación
Request to Send	RTS	Señal de control que indica que se está listo para recibir datos
Clear to Send	CTS	Señal de control que indica que el equipo remoto (MODEM ó periférico) esta listo para recibir datos.
Data Set Ready	DSR	Indica que el periférico está listo (por ejemplo, cuando un Módem queda listo para hacer una llamada después de su inicialización)
Data Terminal Ready	DTR	Señal de control que habilita al dispositivo periférico
Data Carrier Detect	CD	Señal proveniente de un módem, indica que se esta detectando señal portadora de otro módem.
Ring Indicator	RI	Señal que le indica al PC acerca del evento de timbre de una llamada entrante.

3.7.1.2 Tipos de Dispositivos Soportados y sus Implicaciones

Existen dos clases de dispositivos que pueden conectarse a través de Bluetooth, están clasificados como dispositivos tipo 1 y 2.

3.7.1.2.1 Tipo 1

Se encuentran ubicados en los extremos de la comunicación, se denominan también DTE[18] ó Equipos Terminales de Datos, en este caso pueden colocarse la mayoría de los dispositivos computacionales, como Computadores de escritorio, PDAs, y periféricos seriales.

[18] Data Terminal Equipment

3.7.1.2.2 Tipo 2

Son equipos mediadores en la comunicación ó DCE[19], generalmente son dispositivos que brindan acceso a un terminal de datos, tal es el caso de los módems (módems externos al PC, u otros equipos transceptores para acceder a redes), que hacen uso de un puerto serial para ser utilizados por el terminal.

Si bien cuando se habla de cables seriales, esta distinción de equipos impone restricciones acerca del cable que se debe utilizar, por ejemplo, el cable con el que se puede conectar un PC con un módem externo no es el mismo que sirve para conectar dos computadores.

RFCOMM no hace distinción entre estas dos clases de dispositivos, debido a que se trata de las señales lógicas y no físicas (como en el caso de los cables) las que se van a transportar a través de RFCOMM. De esta manera la disposición de las señales es algo que no cobra importancia al hablar de la emulación.

3.7.2 RFCOMM y el Estándar ETSI TS07.10

Si se examinan los requerimientos de RFCOMM anteriormente descritos, y teniendo en cuenta que Bluetooth esta pensado en gran medida para proveer conectividad a dispositivos móviles, quiere decir que esta capa debe estar diseñada para funcionar sobre tales dispositivos, lo cual imprime en ella características de ahorro de energía y de poca exigencia de recursos de memoria y procesamiento para el hardware sobre el que se está ejecutando. Dar un vistazo a elementos software de esta índole ubican a un estándar como el más idóneo (según el SIG[20]) para servir como base para el desarrollo inicial de RFCOMM. Este estándar es el ETSI TS 07.10 (Versión 6.3, llamado también TS 101 369).

ETSI TS07.10 fue diseñado originalmente para dispositivos GSM, dadas sus características de multiplexación de puertos seriales (para el envío simultáneo de voz, SMS, Fax y otros), bajo consumo de energía y de recursos de procesamiento. Este protocolo contiene algunas características que son opcionales, pero que para RFCOMM son obligatorias, como es la consulta del estado actual del puerto.

Al estándar se le hicieron modificaciones; tales modificaciones comprenden:

---

[19] Data Communications Equipment

[20] Special Interest Group

- Adaptaciones de tramas
- Establecimiento de conexiones, ya que RFCOMM debe encargarse de abrirse paso hacia la capa equivalente en el otro dispositivo a través de L2CAP, entonces las funciones que posee TS 07.10 para tal fin dejan de tener importancia.
- Indica como se van a utilizar las funciones de TS 07.10 para multiplexación en RFCOMM.
- Se hace una modificación a los canales (ver el punto 3.7.3 sobre Canales) de multiplexación haciendo distinción entre los puertos de aplicaciones servidores y clientes.
- Consulta del estado de la conexión serial y su configuración, así como también la negociación de los parámetros de cada canal.
- Se hacen algunas modificaciones al control de flujo, con el fin de evitar que se envíe mas información por unidad de tiempo de la que la aplicación puede procesar; estos procedimientos pueden realizarse de varias formas, a nivel de L2CAP, a nivel de las tramas que se manejan a nivel de RFCOMM y TS 07.10 lo maneja a través de otras señales de control.

3.7.3 Canales

La capa de RFCOMM maneja el concepto de canales, los cuales son instancias de Conexiones de Enlaces Seriales de Datos (Data Link Connection - DLCs) y que se distinguen por un número de 5 bits, lo cual permite que hayan 30 canales en cada dispositivo (No es obligatorio que un dispositivo soporte toda esta cantidad de puertos, pueden haber por consiguiente hasta 30 aplicaciones simultáneas en un dispositivo utilizando puertos seriales). Una aplicación servidor necesitará establecer un canal de estos si quiere recibir conexiones de aplicaciones residentes en otro dispositivo. También existe un bit de dirección, cuyo valor es complementario para ambos dispositivos; este será el identificador con el cual la aplicación se registra ante la capa de *Descubrimiento de Servicios* (SDP).

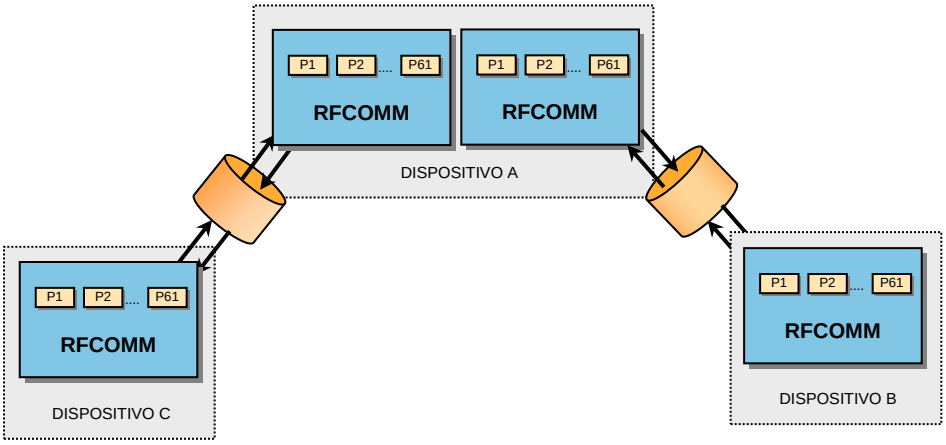


Figura 3.30. Sesiones de RFCOMM entre diferentes dispositivos

Cada dispositivo puede entablar múltiples conexiones seriales con otro una vez que se haya establecido una sesión de RFCOMM entre ellos. Entre dos dispositivos solamente puede haber una sesión establecida. El que primero inició la sesión RFCOMM se denomina *Iniciador*; el otro dispositivo se denomina *No-iniciador.* La primera aplicación que utilice el puerto serial emulado debe iniciar la sesión, y la última que termine de utilizar alguna conexión debe cerrarla. Como puede apreciarse en la Figura 3.30, el dispositivo A tiene sesiones separadas para cada dispositivo, y cada sesión puede soportar varias conexiones. Cada conexión está identificada por un Identificador de Conexión de Enlace Serial de Datos (Data Link Connection Identifier – DLCI).

3.7.3.1 DLCI

Es un identificador para cada conexión de datos que se establece en una sesión de RFCOMM. Es un identificador único en una sesión y representa un flujo de datos entre una aplicación cliente y servidor. Está relacionado con el concepto de canales descrito anteriormente de la siguiente manera: el canal definido por un identificador de 5 bits se junta con el bit de dirección y el número resultante es el DLCI de la conexión. El bit de dirección se establece en uno para el dispositivo que inició la sesión y lo contrario para el otro. Cada aplicación va a tener entonces un canal asociado y una dirección; el identificador lo utilizan los dispositivos para dirigir sus tramas de RFCOMM y dirigir los datos de acuerdo al canal al que están asociados. (para ver este punto en detalle refiérase al punto Tramas RFCOMM).

3.7.4 Modelo de comunicaciones de RFCOMM

En la Figura 3.31 puede observarse el modelo en forma gráfica, se tienen los dos tipos de dispositivos soportados, (no quiere decir que en una comunicación siempre tengan que estar presentes los dos tipos) para citar un ejemplo, entre un Computador portátil y un módem externo para acceder a la red (el PC seria el dispositivo tipo 1 y el módem el dispositivo tipo 2).

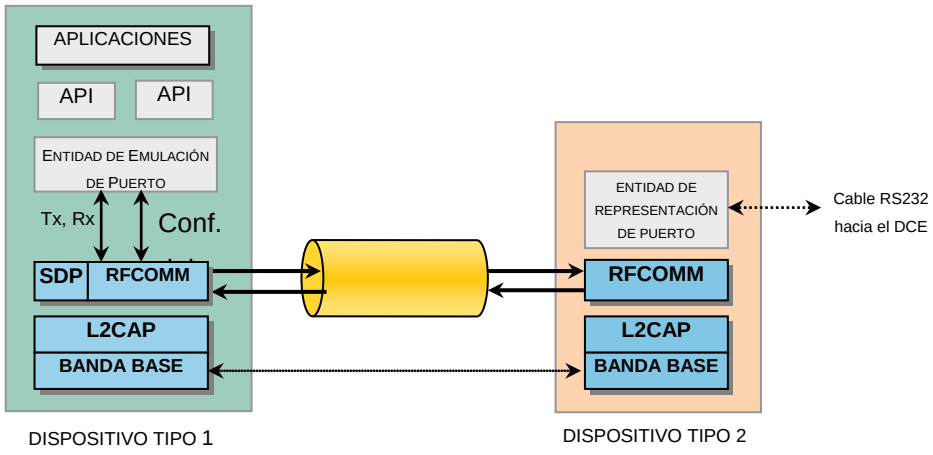


Figura 3.31. Modelo de comunicaciones de RFCOMM

Bluetooth seria el reemplazo de los cables de comunicación entre los dispositivos. Solo es necesario que haya una sesión de RFCOMM y esta puede encargarse de multiplexar los puertos que sean necesarios entre los dos dispositivos. Cabe tener en cuenta que RFCOMM además de ser la entidad de emulación del puerto serial, es quien ofrece múltiples puertos seriales para ser transportados a través de Bluetooth y que dicha interfaz de servicios de emulación del puerto serial está determinada en el estándar ETSI TS 101 369.

3.7.4.1 Entidad de Emulación de Puerto

Es el elemento que se encarga de hacer que las APIs[21] de comunicaciones del sistema puedan hacer uso de las características de RFCOMM; es decir, las funciones del sistema operativo en el que se encuentran las aplicaciones tienen una combinación equivalente en servicios de RFCOMM. Si una aplicación necesita utilizar un puerto serial, sabe que si invoca la funcionalidad que ofrece el sistema podrá utilizarlo de manera transparente

[21] Application Programming Interface

(independientemente de si la aplicación fue diseñada para funcionar sobre cable serial). De esta manera, la entidad de emulación de puerto ofrece esta misma funcionalidad del sistema (indicada en la Figura 3.32 en el manejador de puertos del sistema), pero con la diferencia de estar invocando servicios en RFCOMM.

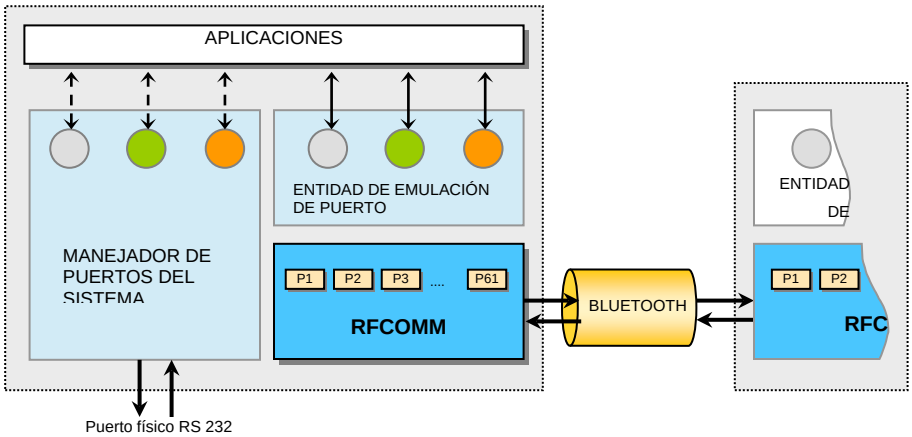


Figura 3.32. Papel de la Entidad de Emulación de Puerto en el equipo Bluetooth

La entidad de emulación de puerto, en conjunto con la capa de RFCOMM conforman un manejador de puerto serial, de la misma forma como el manejador de puertos del sistema operativo. La especificación deja a voluntad del desarrollador del manejador de puertos la responsabilidad de programar esta entidad puesto que depende del sistema operativo en el que va a funcionar.

Puede observarse también en la gráfica que RFCOMM emula varios puertos con cada dispositivo con el que está conectado; el sistema entonces tendrá estos múltiples puertos a disposición de las aplicaciones que ya se hayan implementado, incluso para dispositivos sin Bluetooth.

3.7.4.2 Entidad de Representación de Puerto

Esta entidad está presente (Ver Figura 3.31) en el modelo de comunicaciones de los dispositivos tipo 2, para los cuales hay una interfaz cableada serial; la entidad (también

llamada proxy) debe transferir la información recibida por RFCOMM y convertirla en señales de RS232 para manejar el DCE.

3.7.5 Tramas RFCOMM

Para soportar los diferentes canales que pueden existir entre dos dispositivos, RFCOMM utiliza las tramas definidas por el estándar TS07.10 y las envía a través de L2CAP (es decir, se convierten en la carga útil de los paquetes de L2CAP), estas tramas han sido adaptadas para Bluetooth. Las tramas pueden asimilarse como mensajes de bajo nivel, ellas transportan comandos y otras llevan mensajes de control.

3.7.5.1 Trama Genérica

Las tramas tienen los campos que se describen a continuación.

DIRECCIÓN				CONTROL	LONGITUD		INFORMACIÓN	FCS
EA	C/R	D	CANAL		EA	Largo		
Bit	Bit	Bit	5 Bits		Bit	7 – 15 Bits		
1 Byte				1 Byte	1 ó 2 Bytes		0 - 32767 Bytes	1 Byte

Figura 3.33. Campos de una trama genérica RFCOMM

- Dirección: Identifica a qué canal de multiplexor pertenece la trama; el Bit EA indica si hay más de un Byte para describir la dirección (es decir, se puede extender la dirección con más de un Byte, pero esto nunca se utiliza, por eso el bit tiene valor fijo). El Bit C/R indica si la trama es un comando que va en sentido de dispositivo Iniciador a No Iniciador ó si es una respuesta de No iniciador a Iniciador.

  El bit D es el sentido de la sesión RFCOMM, el dispositivo Iniciador y el No iniciador tienen este bit complementario. El indicador de canal es el que se explica en el punto 3.7.3 sobre Canales; en conjunto con el bit D forman el DLCI del canal de multiplexación. Los canales pueden ir de 1 a 30 ya que el DLCI = 0 es el canal de control del multiplexor y el 31 esta reservado por el estándar TS07.10.

- La palabra de Control indica el tipo de trama que está en curso. Las tramas que se pueden intercambiar son las siguientes:

Tabla 3.45. Tipos de tramas de RFCOMM

Nombre de la Trama	***Función***	***Descripción***
Establecer Modo Asíncrono Balanceado (Set As`ynchronous Balanced Mode - SABM)	Comando	Inicialización de un enlace serial, cuando se inicia por primera vez, inicia primero el multiplexor; posteriormente puede utilizarse para abrir nuevos DLCs.
ACK No-numerado (Unnumbered Acknowledgement - UA)	Respuesta	Respuesta cuando hay conexión
Modo Desconectado (Disconnected Mode - DM)	Respuesta	Cuando un dispositivo rechaza la solicitud SABM para iniciar una sesión de multiplexación RFCOMM.
Desconectar (Disconnect - DISC)	Comando	Desconecta el multiplexor, la implementación debe encargarse de desconectar también el enlace L2CAP que daba soporte a la sesión RFCOMM.
Información No-numerada con verificación de encabezado (Unnumbered Information with Header check - UIH)	Comando y Respuesta	Sirve para enviar información no solo de Control (Comandos) a través del canal de control (DLCI 0), sino también datos a través de los demás canales de datos.

- El *indicador de Longitud* comienza con un bit que indica cual es el tamaño en bits del campo llamado “largo” que se va a utilizar para describir el largo de los datos que le siguen en la trama; de esta manera, los dos modos que se establecen son: 7 bits (los restantes para completar un Byte completo), ó 15 bits (los restantes para completar lo 2 Bytes de longitud). Teniendo esto en mente, los tamaños permitidos para los datos transportados en la trama pueden ser de hasta 127 Bytes en el primer caso y 32767 Bytes en el segundo.

- *Los Datos* son un campo que está presente solamente en las tramas de información No-numerada (UIH). El largo de estos datos está dado por el indicador de longitud que le antecede (sea este de 7 ó de 15 bits). La longitud máxima permitida para los datos está restringida por la Unidad Máxima de Transmisión (Maximum Transmisión

Unit - MTU) de los paquetes L2CAP (ver punto 3.4.6.2 de L2CAP) que dan soporte a la sesión actual de RFCOMM.

- *La Secuencia de Chequeo de Trama (Frame Check Sequence - FCS)* es una palabra de 8 bits calculada a partir del encabezado de la trama, es decir, para los campos de Dirección, palabra de control y Longitud; (para las tramas UIH solamente se calcula para Dirección y palabra de Control); sirve para verificar que la información del encabezado ha llegado bien.

3.7.6 Comandos de Multiplexor

RFCOMM se comporta como un multiplexor de puertos seriales ó DLCs, que muestra ante la aplicación a cada uno de ellos como un puerto COM. En el punto 3.7.3.1 sobre DLCI se explica que existen identificadores para cada DLC que se establezca; existe un canal de control para el multiplexor identificado con el DLCI cero; mediante este canal, los multiplexores de ambas partes intercambian información a través de comandos. Los comandos se envían dentro de las tramas UIH a través de dicho canal y forman parte de la carga de datos de las tramas de la siguiente manera genérica:

Tipo			Longitud	Valor Parámetro 1	Valor Parámetro 2	Valor Parámetro 3	...	Valor último Parámetro
EA	C/R	Tipo						

Figura 3.34. Estructura de los comandos del multiplexor

Los bits EA y C/R tienen la misma descripción que para las tramas, dadas en el punto 3.7.5.1 - Trama Genérica (ver Más atrás). También hay un indicador de longitud para informar el numero de Bytes que ocupa el comando.

Los comandos que se intercambian pueden ser los siguientes:

Tabla 3.46. Comandos de multiplexión

Test	Verifica el enlace de comunicación, contiene un patrón de prueba que debe ser devuelto por el otro dispositivo para así demostrar que el enlace funciona correctamente.
Flow Control On	Habilitan ó deshabilitan el control de flujo de todas las conexiones seriales de la sesión. Si un terminal no puede recibir más datos, envía al otro terminal un comando Flow Control Off. Cuando ya pueda de nuevo recibir datos, enviará un comando Flow Control On, indicando que pueden reanudarse las transmisiones. El comando no tiene argumentos.
Flow Control Off	

Modem Status	Sirve para hacer control de Flujo por cada conexión; emula las señales V.24 de un módem, esto se hace por medio de bits como si se estuviera indicando el estado de cada alambre terminal de una interfaz cableada RS232. Las señales que se emulan son Data Set Ready, Data Terminal Ready, Request To Send, Clear To Send, Ring Indication, y Data Carrier Detect. (ver punto 3.7.1.1 Señales Emuladas por RFCOMM Más atrás). Este comando se utiliza antes de que se envíe cualquier dato para establecer el valor de las señales RS232 y en cualquier momento en que estas necesiten ser cambiadas.
Remote Port Negotiation	Establece parámetros de comunicación en cualquier momento en que necesiten cambiarse. Cuando se envía con una longitud de 1, contiene el DLCI de la conexión, y el comando se interpreta como una solicitud de los parámetros del enlace; el dispositivo remoto responde entonces con los parámetros. Cuando el largo es de 8, habrá 8 Bytes con valores de parámetros para establecer en la conexión. Estos parámetros a negociar son: El DLCI, la Velocidad en baudios de transferencia[22], Bits de datos, Bit de stop, paridad, tipo de paridad, y control de flujo.
Remote Line Status	Se envía este comando al otro lado de la comunicación cuando ha ocurrido un error en el lado actual. El comando contiene el DLCI y un código de error de 4 bits, el cual indica el estado de la línea, el cual puede ser cualquiera de los siguientes errores: Overrun (cuando se sobreescribe en el buffer cuando aún no se ha leído la información anterior), un error de paridad (la verificación del bit de paridad falló), ó error de entramado (un carácter se transmitió sin bit de stop) ó en su defecto, que no hay errores. En la respuesta a este comando se envía una copia de la información de estado de línea recibida.
DLC Parameter Negotiation	Realiza la negociación de parámetros de una conexión (ocurre antes del establecimiento de la conexión). Si esto no se lleva a cabo, se asumen los parámetros por defecto. Utiliza la dirección DLCI para la que se pretende negociar, el tipo de trama a utilizar en la comunicación (siempre es la trama UIH), Modo de control de Flujo (habilitar el control de flujo basado en créditos, ver punto 3.7.8 - Aspectos de Control de Flujo), Prioridad de la conexión, tiempo de espera de ACK (0 - 60 Segundos), y Tamaño máximo de la trama. Luego de sucesivas solicitudes respuestas se obtienen los parámetros de la conexión y se procede a su establecimiento.
Non-Supported	Se envía como respuesta a un comando no reconocido, (en caso de que las dos

[22] Nota: La velocidad real en Baudios no es exactamente la negociada, sino que depende de la capacidad de transferencia que pueda soportar la capa de banda base. Para efectos prácticos, en aplicaciones típicas de puertos seriales puede suponerse que la velocidad real en realidad será mayor que lo negociado. (dado que no todas las aplicaciones harán uso de los enlaces al mismo tiempo y al ser información asíncrona, se puede aprovechar al máximo el ancho de banda disponible como se puede ver en el punto de Banda Base).

Command	implementaciones de RFCOMM en los dispositivos no sean de alguna manera compatibles en sus comandos).

3.7.7 Funcionamiento de RFCOMM

3.7.7.1 Parámetros de Funcionamiento

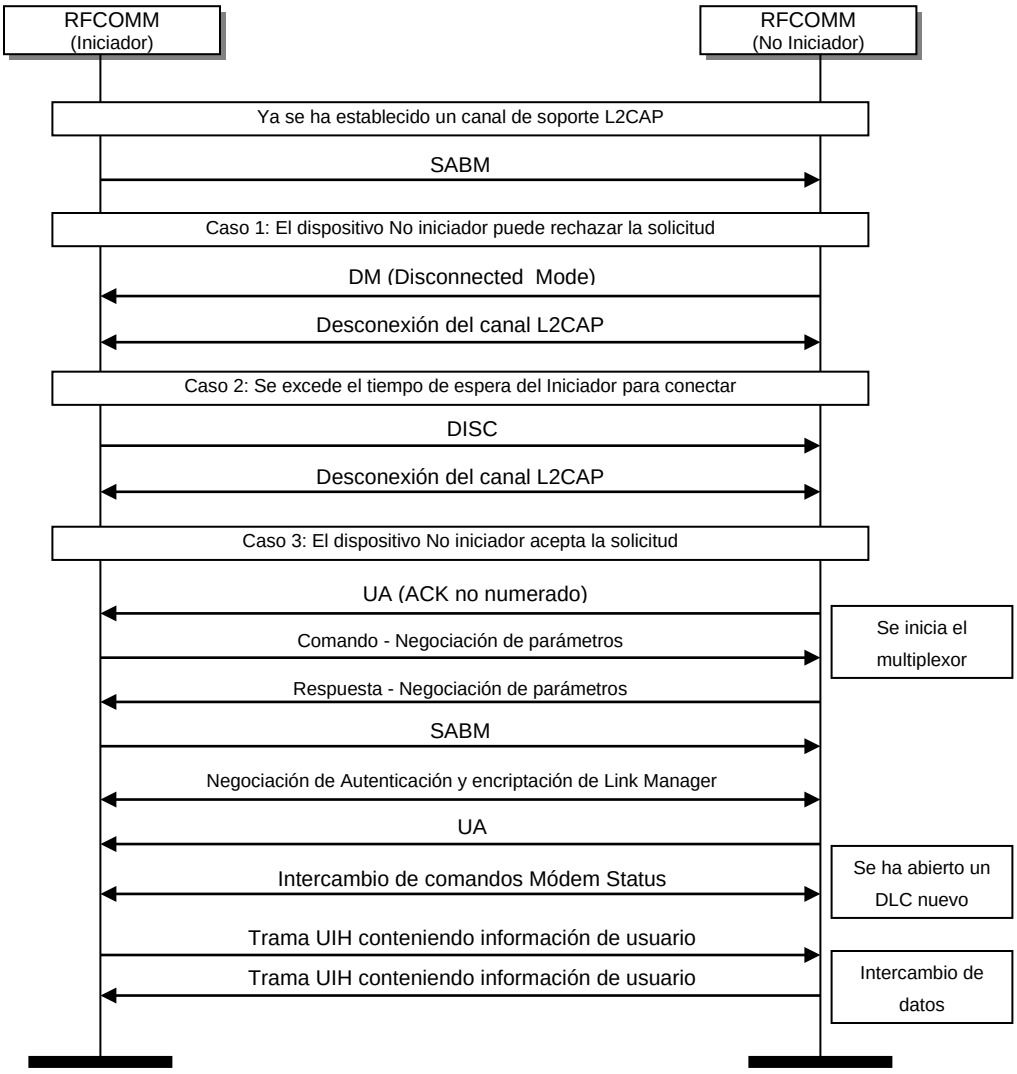
Hay varios parámetros que debe establecer RFCOMM previamente a realizar cualquier intento de conexión. Su valor depende de la implementación y de la aplicación que se tenga. Estos parámetros son:

Tabla 3.47. Parámetros de Conexión Serial a establecer previamente

Nombre	***Descripción***
Máximo Tamaño de Trama	Describe el tamaño máximo permitido para las tramas RFCOMM descritas antes. Valor por defecto: 127, puede ir de 23 a 32767 Bytes.
Temporizador de ACK	Se aplica a las respuestas a las tramas SABM y DISC, si este tiempo (en segundos) se excede, debe ocurrir una desconexión automática.
Temporizador de respuesta del canal de control del multiplexor	Temporiza las respuestas de las tramas UIH en el DLCI cero, es decir, establece un tiempo de espera para recibir respuesta a los comandos de multiplexión.

3.7.7.2 Inicialización

Para poder empezar a utilizar los puertos seriales emulados y multiplexados por RFCOMM, una aplicación debe primero revisar si ya existe una sesión RFCOMM abierta con el dispositivo que se quiere conectar, en este caso debe utilizarla, pero si esto no es así, debe dar inicio a una sesión nueva estableciendo un canal L2CAP que de soporte, e inicializando el multiplexor (enviando una trama SABM a través del DLCI cero y esperando una respuesta). Luego de esto puede empezarse a establecer DLCs para transmitir datos.



Carta de Secuencia 3.4. Conexión y transferencia de información por RFCOMM

En la Carta de Secuencia 3.4 se puede observar la interacción de las tramas y los comandos para establecer conexiones DLC.

3.7.7.3 Finalización

La última aplicación que utilice el multiplexor debe cerrar el canal correspondiente L2CAP y enviar (opcionalmente) la trama DISC, esperando la respuesta para después terminar la sesión.

Carta de Secuencia 3.5. Desconexión de la sesión

3.7.8 Aspectos de Control de Flujo

Cuando un dispositivo no puede procesar información a una velocidad mayor ó igual a la cual la está recibiendo, existe el riesgo de que se pierda información ó que el dispositivo más lento sea interrumpido tantas veces por recepción de información, que acabe por colapsar. Por esto, existen varios métodos para controlar el flujo de información que se transfiere y así evitar problemas. Estos métodos son:

- El mecanismo de L2CAP para control de flujo, el cual está soportado por el Nivel el Link Manager; dado que L2CAP soporta a RFCOMM, se aprovecha este mecanismo.
- Los puertos seriales cableados pueden realizar control de flujo haciendo uso de las señales Ready to Send, Clear to Send, Data Terminal Ready ó Data Set Ready. Como RFCOMM puede emular estas señales (mediante el comando Módem Status), entonces el control de flujo puede realizarse de esta manera; la característica especial de este método consiste en que puede aplicarse en forma unidireccional (es más probable que un periférico, que posee menos memoria y capacidad procesamiento necesite más control de flujo que un computador; en este caso, el control es de forma unidireccional).
- Los comandos de multiplexación Flow Control On y Off son una característica heredada del estándar TS07.10 y están soportados en RFCOMM.
- La Entidad de emulación de puerto puede tener que realizar control de flujo según las especificaciones del API que se está emulando, para lo cual hay un mecanismo de control de flujo llamado *Control de Flujo Basado en Créditos,* el cual se basa en el principio de que se pueden enviar tramas RFCOMM mientras el otro lado de la comunicación responda con créditos suficientes (indicados en un campo de un Byte en el área de datos); Los créditos indican la cantidad de tramas que el otro dispositivo va a

poder recibir sin llenarse. Si en alguna ocasión el indicador de créditos en la respuesta llega a ser cero, las transmisiones deben detenerse hasta recibir una respuesta diferente.

3.7.9 Descubrimiento de Servicios Soportados en RFCOMM

Cuando una aplicación ofrece un servicio, el cual está soportado en RFCOMM, debe proveer parámetros para poder realizar esta conexión. La mínima información requerida para ello es un nombre para el servicio, y un número de canal de servidor (5 bits) al cual conectarse, para mayor claridad al respecto, refiérase al perfil de puerto serial en el capítulo 4.

LISTA DE FIGURAS

Listado de tablas

cartas de secuencia

TABLA DE CONTENIDO

CAPITULO 4. PERFILES DE APLICACIÓN DE BLUETOOTH

4.1 INTRODUCCIÓN

Los modelos de uso contemplados por Bluetooth han dado origen a una serie de perfiles de aplicación, definidos como parte de la especificación Bluetooth y tienen como fin principal hacer que las aplicaciones puedan enmarcarse en tales perfiles ó modelos y ser así interoperables en caso de que las aplicaciones que se comuniquen por Bluetooth sean hechas por distintos fabricantes.

Si bien el núcleo de la especificación (estudiado en el capítulo 2) garantiza interoperabilidad entre los dispositivos Bluetooth fabricados por diferentes empresas, la especificación de los perfiles hace lo mismo pero en el campo de las aplicaciones posibles. También estos perfiles han sido diseñados de tal manera que reutilizan y heredan algunas cosas de protocolos ya existentes, como en el caso de los protocolos OBEX, las aplicaciones que utilicen puertos seriales, el perfil de descubrimiento de servicios (descrito en el capítulo 2), y los protocolos de telefonía. Los perfiles de aplicación están organizados de manera que unos dan servicios (aunque también sirven para aplicaciones) y otros sirven para aplicaciones específicas. En este sentido la organización de tales perfiles según la especificación es la siguiente:

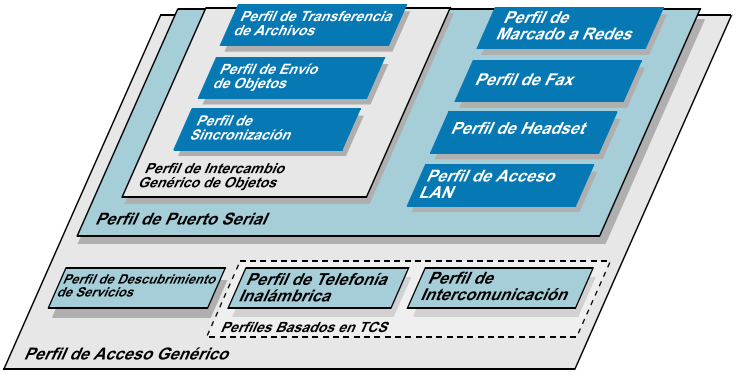


Figura 4.1 Perfiles de aplicación y sus interdependencia

Para fines del estudio que se va a realizar de los perfiles de aplicación, se hizo una agrupación de acuerdo al servicio que prestan. Por un lado se tiene los perfiles generales que por un lado sirven de apoyo para los demás perfiles dado que lo primero que debe hacer un dispositivo al comunicarse es acceder al otro dispositivo y descubrir sus servicios.

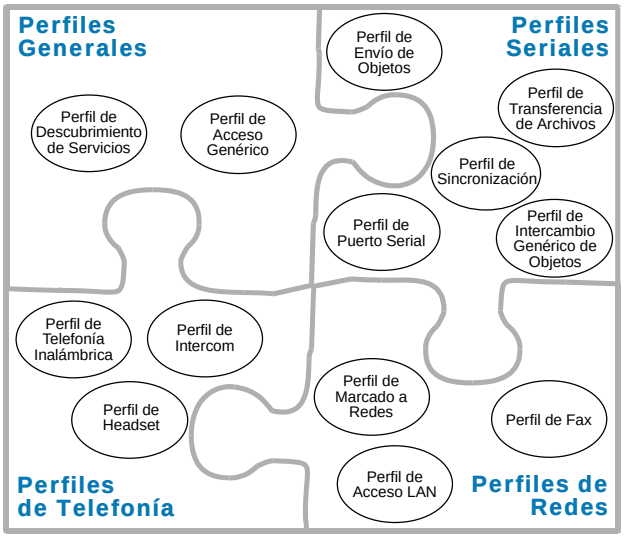


Figura 4.2 Agrupación alternativa de los perfiles de aplicación Bluetooth

Los perfiles seriales prácticamente se pueden distinguir por una característica especial: siempre hay aplicaciones cliente y servidor y una sesión de por medio.

Respecto al servicio de telefonía, se puede decir que hay perfiles que se ajustan a tal campo de aplicación.

El hecho de que haya aplicaciones que utilicen interfaces cableadas para transmitir datos ó acceder a redes hace pensar que se podría formar un grupo de esta clase de perfiles de aplicación.

4.2 Perfil de Acceso Genérico

Cuando dos dispositivos pueden conectarse a través de un cable de datos, se necesita de una u otra forma la cercanía física entre ellos. Normalmente se sabe por ejemplo, que un módem externo requiere de un cable para ser conectado a un PC y también de una configuración posterior para hacer que todo funcione; pero en el caso de conexiones inalámbricas donde no podemos ver el cable, y por supuesto, los dispositivos tampoco perciben estar "conectados" mediante este cable virtual, la situación debe solucionarse suministrando a los dispositivos cierta información básica para poder entablar una comunicación.

Figura 4.3 Problema básico de acceso entre dispositivos

Con tal información básica, un dispositivo puede estar programado para no recibir conexiones (bien sea porque se encuentre ocupado) ó también para recibirlas siempre, en este caso los dispositivos que se le conecten ó que pretendan conectársele pueden hacerlo incluso si se encuentran en recintos diferentes ó incluso bajo llave, lo cual es una posibilidad no muy agradable para dispositivos que guarden información personal ó que presten servicios, ya que cualquier otro dispositivo podría en principio atentar contra la seguridad de dicha información.

Todo lo anterior plantea tres problemas básicos:

1. Los dispositivos deben conocer con qué otros pueden conectarse.
2. Debe haber una forma de poder restringir tales conexiones.
3. Es necesario proteger la integridad de la información que se almacena e intercambia.

Frente a esto, pueden plantearse soluciones a nivel de las capas más bajas de protocolos que conforman la pila de Bluetooth, y es precisamente este hecho lo que hace que estas

capas tengan gran importancia al ser utilizadas de manera similar por todas las aplicaciones para ejecutar estas funciones básicas, y éste perfil esta orientado a resolver tales necesidades. Los tres problemas básicos se ven resueltos respectivamente en los puntos siguientes de Descubrimiento Inicial de Dispositivos, Establecimiento de Enlaces y Mecanismos de Seguridad.

4.2.1 Ejemplos de Uso del Acceso Genérico

Si se supone el caso de un punto de acceso a la red que puede estar ubicado en una oficina. Un usuario de un computador portátil desea acceder a la red, pero previamente para tal fin utilizó su celular equipado con Bluetooth; esta vez el usuario tendrá la oportunidad de elegir cuál de los dos dispositivos utilizar, si el celular (lo cual representa un cargo a pagar por utilizar la red de acceso celular) ó emplear el punto de acceso, el cual puede estar conectado a una red fija (tal vez corporativa). Dado que el punto de acceso solamente debe ser descubierto por dispositivos con capacidad de procesamiento para acceder a la red, entonces para tal aplicación especifica (que tal vez requiera de un software ejecutándose como servidor en el punto de acceso) es necesario restringir la facilidad para descubrir al punto de acceso cuando se está en el mismo entorno con otros dispositivos Bluetooth.

Esto lleva a tres estados de descubrimiento: el de No Descubrimiento, el de Descubrimiento Limitado y el de Descubrimiento General. Ver más adelante para un mejor detalle de estos estados.

Supóngase ahora el caso de una impresora ubicada en un entorno de oficina ó académico, está equipada con un puerto inalámbrico Bluetooth, lo cual permite que varios equipos puedan acceder a sus servicios. Pueden tener lugar varios escenarios de uso:

- Un computador que está fijo en la oficina accede a ella para hacer una impresión de un documento.
- Un dispositivo foráneo portátil desea hacer una impresión en el mismo aparato.
- Un equipo de una oficina contigua, gracias a que logra descubrir la impresora, desea realizar una impresión en ella.

Se pueden deducir si se tiene en cuenta que hay varios tipos de dispositivos con intención de acceder a un servicio, entonces habrá varios modos de establecer enlaces y de filtrar qué

dispositivos se quiere que tengan acceso al servicio; estos son los modos de establecimiento de enlaces y modos de seguridad.

4.2.2 Descubrimiento Inicial de Dispositivos y Servicios

Para que un dispositivo descubra la presencia de otro es necesario que el otro esté habilitado para responder a las búsquedas que efectúe el primero; esto define niveles de descubrimiento en los que puede quedar un dispositivo. Luego del descubrimiento, en el caso del punto de acceso, el usuario necesita saber a qué dispositivo recurrir para usar el servicio de acceso a la red, por ello al ser descubierto, cada dispositivo debe ofrecer una manera de saber qué servicios ofrece y cómo conectarse a ellos; este último dato se obtiene del uso del perfil de descubrimiento, pero los servicios básicos se encuentran registrados primeramente en una variable de 24 bits llamada CoD[1], la cual se intercambia en la respuesta a la indagación en el paquete FHS (Ver Capitulo 2 – Banda Base). Sus bits pueden indicar si el dispositivo es un PC, si es un dispositivo de captura (escáner, micrófono, cámara) si es de salida (impresora, parlantes), si ofrece servicios de objetos, si es un dispositivo de localización, etc. Para mayor información *ver Documento Bluetooth Assigned Numbers.*[2]

4.2.2.1 Modo de Descubrimiento General

Operando en este modo el dispositivo entra a buscar indagaciones de manera periódica utilizando como parámetro el Código de Acceso Genérico a Indagación; la utilización de este código garantiza la respuesta a todas las indagaciones que reciba su procesador de banda base. Todos los dispositivos que realicen indagaciones lo encontrarán.

4.2.2.2 Modo de Descubrimiento Limitado

El dispositivo que opere en este modo realiza indagaciones en forma periódica, pero utilizando el código de Acceso a Indagación Limitado, el cual es un código reservado para este propósito. Un dispositivo en modo de descubrimiento limitado puede ser descubierto durante menos tiempo por dispositivos que hagan indagaciones con el código genérico. Para una aplicación de servicio específica puede ser deseable que el dispositivo en el que reside entre en este modo, puesto que así solo los dispositivos que queremos que se conecten a tal servicio podrán descubrirlo gracias a la utilización de indagaciones con el código Limitado. Un ejemplo de ello es en servicios de objetos (transferencia de archivos, de sincronización,

[1] Class of Device

[2] Se encuentra en la dirección http://www.bluetooth.org/assigned-numbers/

etc.) en los cuales solo se va a emplear un cliente específico (PDA por ejemplo) para acceder a tal servidor (Computador de escritorio); si se programa el PDA y el Computador para que utilicen el código de acceso limitado, entonces otros dispositivos (probablemente no deseados) no podrán descubrir el mismo computador y por tanto, el mismo servicio.

4.2.2.3 Modo de No Descubrimiento

En este modo, un dispositivo puede no responder a indagaciones de otros dispositivos; puede utilizarse por diferentes razones, como por ejemplo, una carga de procesamiento alta en el dispositivo puede hacer que entre en este modo mientras se restablece su funcionamiento.

4.2.3 Establecimiento de Enlaces

Una vez que se ha descubierto un dispositivo, se puede establecer un enlace físico con él; la respuesta a este evento puede estar restringida también; puede tomarse de nuevo el ejemplo de la impresora y los otros equipos de oficina.

4.2.3.1 Relación de Confianza Entre Dispositivos

En el establecimiento de los enlaces interviene una condición de autenticación, dependiendo de la configuración de los elementos involucrados en la comunicación. Un elemento puede solicitar autenticación para otro que llega a conectarse a él; en tal procedimiento se intercambian números PIN y de ahí se establece una relación de confianza entre los dispositivos, puesto que ambos pueden conservar tales números y utilizarlos de nuevo en conexiones posteriores.

Figura 4.4 Lazo de confianza entre dispositivos

En el caso de la impresora existen computadores que durante su configuración ó durante la primera conexión que realizaron con la impresora fueron autenticados, de tal manera que impresiones posteriores puedan hacerse sin necesidad de solicitar al usuario un numero PIN de nuevo. (ver punto 4.2.4 - Mecanismos de Seguridad).

4.2.3.2 Modos Conectable y No Conectable

En estos modos el dispositivo puede aceptar ó no aceptar conexiones a nivel de *Link Manager*, aunque si responder a indagaciones; puede ser útil en el caso de dispositivos para recoger datos de sensores inalámbricos Bluetooth, donde los sensores son los que deben aceptar las conexiones y no el dispositivo central.

4.2.3.3 Modos Autenticable y No Autenticable

En estos modos se emplea la capacidad de autenticarse con otros dispositivos como característica principal. Cuando un dispositivo solicita conectarse a otro, el segundo puede solicitar autenticación para dar paso a la transacción; el primer dispositivo, dependiendo del modo en que se encuentre puede aceptar ó rechazar la petición de autenticación; un dispositivo puede estar programado en modo *No Autenticable* si no tiene las capacidades de despliegue ó de teclado para visualizar y entrar el número PIN.

4.2.4 Mecanismos de Seguridad

Dado el problema de proteger la integridad de la información que se almacenada como la que se transporta y también de evitar la suplantación por parte de dispositivos no deseados para acceder a servicios privados, se hace necesario establecer modos de funcionamiento en cuanto al nivel de seguridad que se ofrezca, es decir, a qué nivel se disparan los procedimientos que garanticen la seguridad.

4.2.4.1 Clasificación de los Dispositivos Según la Perspectiva de la Seguridad

- *Dispositivos Confiables:* Mantienen un vínculo de confianza fijo, ya se ha autenticado al menos una vez y también se le ha catalogado como “confiable” por parte del usuario del dispositivo al cual el dispositivo confiable quiere acceder; esto le garantiza acceso libre a todos los servicios. Por ejemplo, para la impresora, un dispositivo de confianza sería el PC que se encuentra fijo en la misma oficina, el cual no necesitará autenticarse de nuevo cada vez que necesite hacer una impresión; inicialmente el dueño debe haber configurado para que aceptara a tal equipo fijo como “confiable”.

- *Dispositivos No Confiables:* No tienen un vínculo permanente, aunque si temporal de seguridad. Se le restringe el acceso a servicios. Para la impresora del ejemplo, un dispositivo no confiable podría ser aquel equipo de una oficina adyacente ó el PC portátil que ocasionalmente entra a la oficina e imprime con permiso del propietario de la impresora.

4.2.4.2 Clasificación de los Servicios Según la Perspectiva de la Seguridad

- *Servicios Abiertos:* En esta clase de servicios, todos los dispositivos podrán tener acceso, no hay ninguna barrera de seguridad para ellos.

- *Servicios de Solo Autenticación:* En este caso los servicios se concentran en solicitar autenticación. Si pasa las pruebas de autenticación, se tendrá acceso al servicio.

- *Servicios de Autenticación y Autorización:* Requieren que haya previamente un lazo de confianza con el dispositivo solicitante del servicio. Cualquier otro dispositivo debe solicitar autorización de manera manual (con intervención del usuario). La autorización siempre implica autenticación previa.

4.2.4.3 Modo de Seguridad 1

Este nivel de seguridad no realiza ningún filtrado ni barrera de seguridad, es decir, no inicia nunca procedimientos de autenticación ó encriptación para entablar comunicaciones.

4.2.4.4 Modo de Seguridad 2

Es un modo de Seguridad más general, puesto que el primero podría considerarse como un caso específico del segundo, donde no se han establecido políticas de seguridad ni requerimientos para los servicios; estos requerimientos pueden combinar factores como Autenticación (para evitar suplantación de identidad de dispositivos), Autorización (Para establecer privilegios) y Encriptación (para proteger la privacidad de la información transportada).

Cuando un dispositivo solicita establecer un canal de comunicación a través del protocolo de control de enlace lógico (L2CAP) es cuando se inician los procedimientos de seguridad. Esto significa que la seguridad es a nivel de servicio. El nivel de permisividad a que podrá llegar

otro dispositivo es hasta establecer enlaces a nivel de *Link Manager,* pero el acceso a aplicaciones será restringido por este modo de seguridad.

4.2.4.5 Modo de Seguridad 3

En este Modo la seguridad puede iniciarse a nivel del *Link Manager.* Antes de terminar el establecimiento de un enlace a ese nivel, se debe iniciar algún procedimiento de autenticación, autorización ó encriptación según se requiera.

4.3 Perfil de Puerto Serial

Dos aplicaciones que manejan información semejante en diferentes dispositivos, normalmente necesitan comunicarse para compartir tal información. Para ello se ha habilitado a tales aplicaciones con capacidades para utilizar puertos seriales, de tal manera que las aplicaciones puedan interactuar. Usualmente estas capacidades seriales las ofrecen los manejadores de puertos seriales cableados, que proveen registros, recursos de memoria para almacenar datos de entrada y salida y funcionalidad para enviar y recibir.

Bluetooth fue creado inicialmente con el propósito de reemplazar los cables de conexión de datos existentes, y los cables seriales no podrían ser una excepción puesto que son los más utilizados. Garantizar que las aplicaciones que utilizan interfaces seriales cableadas puedan operar igualmente sobre Bluetooth es una característica casi necesaria para que esta tecnología sea una alternativa viable a la hora de elegir el tipo de interfaz de comunicaciones en el diseño de equipos y de nuevos productos y aplicaciones.

4.3.1 Carácter Serial de algunas Aplicaciones

Los ejemplos de aplicaciones seriales son muy diversos, las cámaras, los dispositivos de almacenamiento externos al computador, las transferencias de archivos y aplicaciones de sincronización de agenda y calendario son una muestra de ello. El carácter serial de las aplicaciones presente en aquellas como intercambio de objetos, envío de objetos, transferencia de archivos, sincronización y acceso a redes, es prácticamente evidente, pero lo que no es evidente es que tales aplicaciones generalmente requieren de la presencia de un cliente y servidor escuchando y transmitiendo sobre la interfaz serial (refiérase a los perfiles correspondientes a tales aplicaciones); esto conlleva a que cada una de las aplicaciones debería tener a su disposición una interfaz de este tipo; de ahí que se requiera una forma de multiplexar los múltiples puertos seriales que se necesiten, capacidad que ofrece la capa RFCOMM de la pila de protocolos.

4.3.2 Equivalente Serial Cableado e Inalámbrico

Una aplicación serial puede ser por ejemplo, la utilización de un cable serial para conectar dos computadores personales (en este caso, se utiliza la configuración de cable de cruzado ó *Null Módem* [3]), en tal caso uno de los dos equipos debe iniciar comunicaciones sabiendo de antemano que el otro está presente en el otro lado, si esto no ocurre, entonces el iniciador debe anunciar este hecho al usuario.

Aquí se han supuesto un par de cosas:

1. la primera, que el iniciador sabe de antemano que el otro equipo ya está conectado a la interfaz serial cuando recibe el comando de usuario para "*Conectar con Equipo Remoto*".
2. Que el iniciador también conoce qué número de puerto COM debe utilizar y al cual está conectado el otro equipo; esto último lo sabe seguramente gracias a una entrada del usuario.

Esta es la información que hay que suministrar a los dispositivos cuando la conexión es cableada; en el caso equivalente inalámbrico, la información es esencialmente la misma; para suministrarla y realizar la conexión, la capa de protocolo RFCOMM, el perfil de aplicación serial, en conjunto con el de *Acceso Genérico* y el *Descubrimiento de Servicios* plantean una solución que provee interoperabilidad entre las aplicaciones que cumplan con ciertos requisitos.

4.3.2.1 Implicaciones para las Aplicaciones Seriales Bluetooth

Las aplicaciones seriales inalámbricas requieren seguir varios pasos antes de poder realizar conexiones seriales con otros dispositivos:

1. El iniciador debe descubrir al otro dispositivo y averiguar si tiene capacidades para soportar servicios que tengan que ver con puertos seriales (por ejemplo, en el caso de un periférico, el parámetro *Class of Device* contiene el código correspondiente a un dispositivo periférico como joystick, teclado ó mouse[4]); para ello puede Utilizar la funcionalidad descrita en el perfil de acceso genérico para descubrir el dispositivo

[3] Configuración de cable serial que consiste en usar un cable serial para periférico, pero cruzando sus terminales de Tx y Rx, también las de RTS y CTS, y las de DTR y DSR. En esta configuración el cable se comporta como un módem visto desde ambos dispositivos, de ahí su nombre

[4] Estos números y la estructura del registro *Class of Device* se encuentran especificados en la dirección http://www.bluetooth.org/assigned-numbers/

(usando alguno de los modos de descubrimiento, general ó limitado) y luego obtener información básica acerca de si el dispositivo encontrado es en realidad al que se quiere conectar (seleccionarlo de una lista de dispositivos encontrados). Hasta aquí la funcionalidad necesaria, como se puede ver, se limita a la proporcionada por el perfil de acceso general. También habrá que tener en cuenta las características de seguridad requeridas puesto que en algún momento el otro dispositivo puede solicitar autenticación.

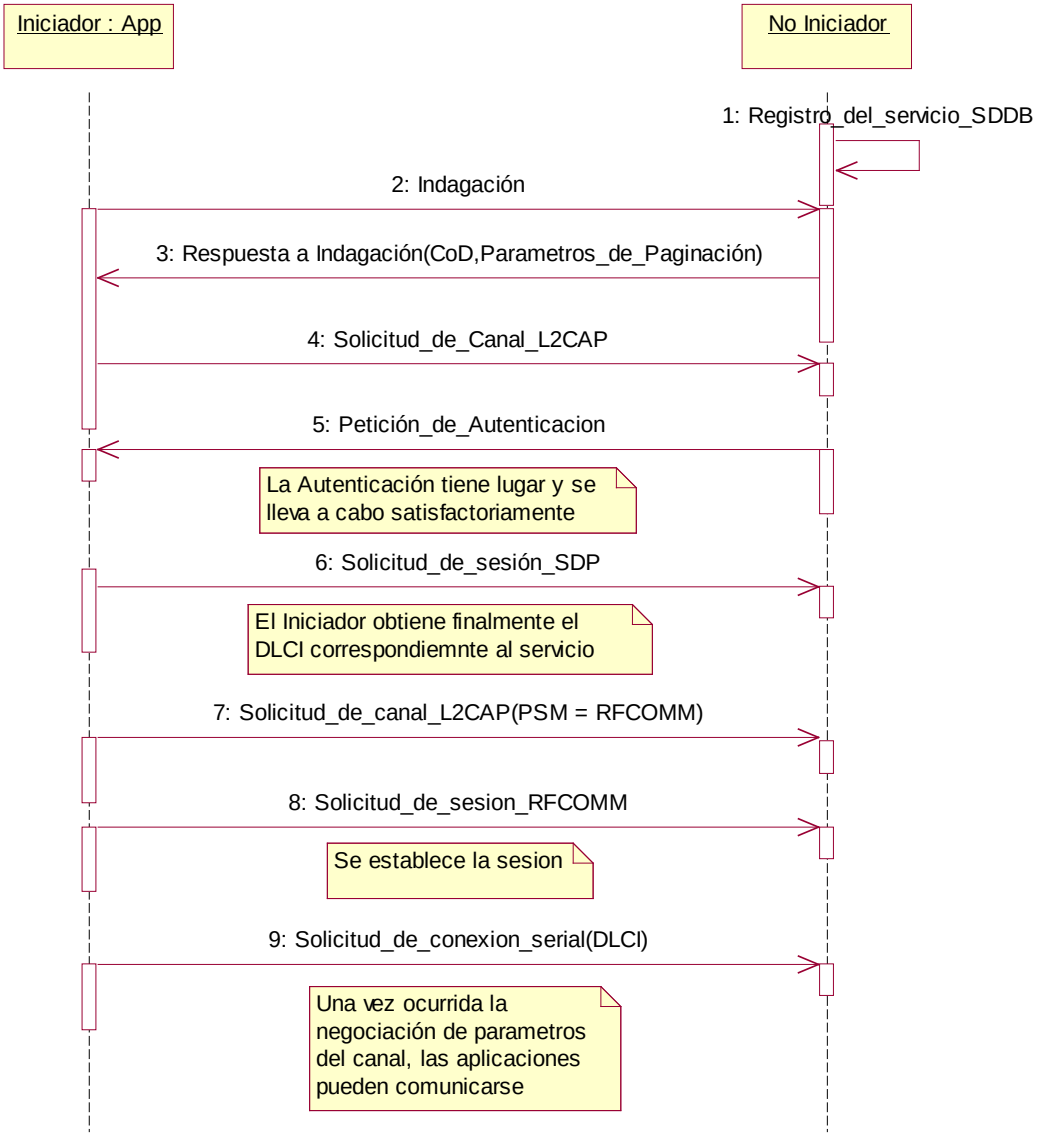
2. Luego de tener descubierto el dispositivo, el iniciador debe buscar la aplicación dentro del dispositivo remoto y para ello solicita un canal L2CAP para iniciar una sesión de descubrimiento de servicios con el dispositivo remoto; a través de sucesivas solicitudes y respuestas en esta sesión se debe obtener el identificador del canal serial (*Data link Connection Identifier ó* DLCI definido por RFCOMM ver capitulo 2 para más detalles). Cabe anotar que entonces la aplicación del dispositivo remoto debe haberse registrado en su base de datos de descubrimiento ó SDDB para lograr ser descubierta por el iniciador.
3. Una vez obtenido el DLCI, la aplicación del dispositivo iniciador sabe que en éste canal la aplicación remota esta escuchando solicitudes, un concepto muy similar al implementado por los puertos TCP donde de antemano se sabe en qué puerto encontrar una aplicación específica, es el caso de los puertos “conocidos” como por ejemplo, el 25 para encontrar un servidor de correo, ó el 80 para servidores web, 21 para ftp, etc.
4. El iniciador solicita un canal L2CAP con el parámetro PSM[5] establecido en el valor correspondiente a RFCOMM. Esto establece una sesión RFCOMM y una vez establecida, utiliza el DLCI obtenido para abrir el canal serial que comunicará a las dos aplicaciones.

Supóngase una aplicación que esta soportada por una interfaz serial como por ejemplo cualquiera de los servidores OBEX (Transferencia de archivos, sincronización y envío de objetos) explicados en los respectivos perfiles de aplicación; cada vez que un cliente se quiera conectar a estos servidores y establecer una sesión, debe ocurrir este proceso por parte del cliente, previa iniciación del servidor. A continuación puede verse una carta de secuencia de mensajes para ilustrar mejor toda la negociación.

[5] Protocol and Service Multiplexor – Multiplexor de Servicios y Protocolos; código que identifica el tipo de servicio ó protocolo que va a utilizar el canal L2CAP que se solicita.

4.3.2.2 Procedimiento Típico de Conexión

El procedimiento puede definirse en cuanto a las solicitudes en ambos sentidos que se lleven a cabo. La carta de secuencia 1 puede mostrar un caso típico sin todos los detalles.



Carta de Secuencia 4.1 Proceso de conexión con una aplicacion serial

4.3.3 El papel del Descubrimiento de Servicios

El descubrimiento de servicios es importante a este nivel en la programación de aplicaciones, ya que es la base para la utilización de otras aplicaciones remotas en otros

dispositivos; así que a cada aplicación por su parte le corresponde una tarea específica; a la aplicación servidora le corresponde registrarse en la base de datos de descubrimiento SDDB para poder ser utilizada, y por otro lado, la aplicación iniciadora (actuando como cliente) debe buscar tanto al dispositivo como a la aplicación para utilizarla.

Los servicios seriales son en cierta forma genéricos, puesto que son utilizados por otras aplicaciones ó servicios de capas superiores (como los de objetos ó acceso a redes, por ejemplo), estos servicios deben registrarse de igual manera y registrar el código DLCI ó identificador de canal servidor para acceder a ellos. Los registros en la base de datos de descubrimiento para el servicio serial deben ser como sigue:

Tabla 4. 1 Registros para los servicios seriales

Parámetros			***Valor***
Lista de Clases de Servicio	Clase de servicio0		SerialPort *
Lista de Descriptores de protocolos	Protocolo0		L2CAP
	Protocolo1		RFCOMM
		Parámetro0	<numero de canal servidor>
Nombre del servicio (opcional)			Nombre del servicio, por ejemplo, “chat”

*Ver http://www.bluetooth.org/assigned-numbers/

4.4 Perfil Genérico de Intercambio de Objetos

El perfil de Intercambio de objetos (*Object Exchange* - OBEX) es un perfil que presta servicios a otros como se verá mas adelante en los perfiles de transferencia de archivos, de Envío de Objetos y de sincronización. Lo que tienen en común estos tres perfiles es que deben transportar una entidad de datos de un dispositivo a otro. Puede entenderse por entidad de datos un arreglo, una variable de estado de alguna máquina, información de contacto de una persona, una imagen, un archivo de texto, etc.

La situación puede apreciarse en la Figura 4.5.

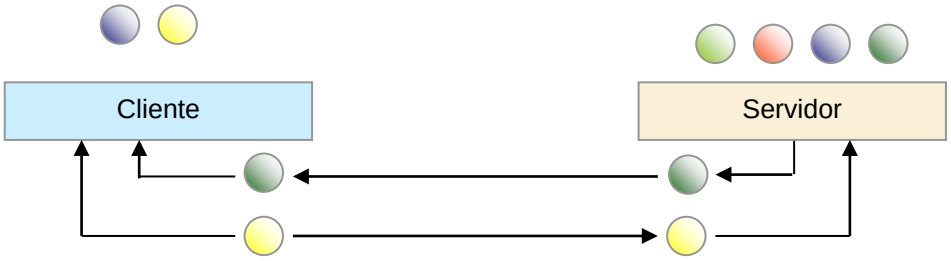


Figura 4.5 Intercambio de objetos

Los requerimientos de este perfil, están encaminados a permitir el paso de objetos de un dispositivo a otro de tal forma que para las aplicaciones esto sea transparente y más sencillo de implementar para los programadores de dichas aplicaciones.

Para resolver esta necesidad se ha adoptado el protocolo OBEX utilizado en la pila de protocolos de IrDA[6] y que constituye también una característica de interoperabilidad para que las aplicaciones que funcionan con IrDA, también puedan hacerlo con Bluetooth. El protocolo OBEX no solamente define procedimientos de intercambio de los objetos (lo cual es el protocolo mismo), sino que también define un formato de representación de objetos de datos con el fin de que estos puedan ser descritos por la aplicación y reconstruidos de nuevo al llegar a su destino. Para una descripción más a fondo de estos elementos, pueden verse los puntos siguientes.

4.4.1 Formato de Representación de Objetos

El formato de representación de objetos es la definición que se hace para poner de acuerdo a dos aplicaciones sobre como representar una entidad de datos. Primero que todo, un objeto de datos tiene varias propiedades; en el caso de un archivo, por ejemplo, tiene nombre, fecha, un contenido (quizás en texto, en html, ó binario), tal vez una descripción breve, entre otras propiedades. Un objeto puede entonces reducirse a la descripción de sus propiedades y con base en esto puede reconstruirse nuevamente.

El formato de representación se basa en encabezados preestablecidos que significan cada uno un parámetro ó propiedad diferente; se colocan de manera consecutiva a la hora de realizar la transferencia de tal forma que el objeto queda reducido a una cadena de bytes

[6] Infrared Data Asociation (Ver Capítulo2 y el Anexo sobre Tecnologías Inalámbricas de Corto Alcance para una descripción de esta tecnología).

que se puede colocar en paquetes (los definidos por el protocolo OBEX) para poder ser transportado.
Algunos de los encabezados son los siguientes:

Tabla 4. 2 Encabezados para representación de objetos

Encabezado	***Descripción***
Nombre	Nombre del Objeto. Si es un archivo, es una cadena con su nombre y extensión.
Tipo	Es el tipo de contenido (texto, html, binario, propietario de una aplicación, etc)
Longitud	Longitud del objeto en Bytes
Descripción	Cadena de caracteres que describe brevemente al objeto
Fecha	Fecha del último cambio
Cuerpo	Cuerpo del contenido del objeto (parte del total cuando se fragmenta en trozos para ser transportado)
Final de cuerpo	Parte final del Cuerpo (cuando se ha fragmentado)
Destino	Es el parámetro con el cual un dispositivo reconoce una aplicación ó servicio de intercambio de objetos específico.
Quien	Indica la aplicación ó servicio al que se acaba de conectar un dispositivo
IDConexión	Número que identifica una conexión; lo devuelve el servidor al aceptar una solicitud de conexión.
Petición de autenticación	Sirven como solicitud y respuesta respectivamente de la autenticación para acceder a un servicio en un servidor que ofrezca intercambio de objetos.
Respuesta de autenticación	
Otros	Otros encabezados que se pueden definir por parte de la aplicación (ocupando 16 espacios dentro de los códigos disponibles)

Los encabezados se colocan en los paquetes como si fueran etiquetas que indican el comienzo de la información que describen; por ejemplo, el campo “Destino” le dirá a la aplicación donde llega que de ahí en adelante dentro del paquete, se encontrará la información que describe al servicio al que se quiere conectar. (ver ejemplo de conexión en los puntos 4.4.3.1 y 4.4.3.2). Los encabezados describen que información les sigue en la trama, pero a menudo están seguidos por un numero indicador de la longitud de la información para que así se pueda diferenciar dicha información de los siguientes encabezados que haya; por ejemplo, el campo *Nombre* está seguido de un número de 2 bytes que indica la longitud de la cadena del nombre del archivo.

4.4.2 Protocolo OBEX

El protocolo tiene como bases las tramas que se pueden armar tanto para control como para transmitir objetos, existen tramas de conexión, de transferencia y mediante estas los dispositivos “conversan”.

Puede considerarse al OBEX como un protocolo de sesión parecido a HTTP pero lo suficientemente sencillo como para correr sobre dispositivos de poca capacidad de procesamiento. Trabaja bajo el esquema Cliente – Servidor y las tramas que envía se clasifican brevemente de la siguiente manera:

Tabla 4. 3. Tramas de solicitudes de OBEX

Nombre	***Descripción***
Conectar	Para negociar la conexión y sus parámetros (autenticación y tamaño de tramas)
Desconectar	Indica el fin de la sesión
Put	Envía un objeto (ó un trozo de él al no caber en una sola solicitud)
Get	Obtiene un objeto (ó algún trozo de él al no caber en una sola solicitud)
EstablecerRuta	Modifica la Ruta (de directorios, por ejemplo) en el lado que recibe esta trama.
Abortar	Cancela la operación que se esté llevando a cabo

Las tramas tienen también códigos de respuesta, los cuales pueden estar seguidos de más encabezados indicando parámetros; por ejemplo, en una solicitud *Conectar*, el otro dispositivo puede responder con una solicitud de autenticación, ofreciendo parámetros para que dicha autenticación tenga lugar.

Algunos de los códigos de respuesta son:

Tabla 4. 4 Respuestas a solicitudes OBEX

Nombre	***Descripcion***
Continúe	Para dar confirmación de que cada partición de un objeto o una transacción ha llegado y se puede continuar con la operación
Operación exitosa	Indica cuando la operación ha sido terminada (una transferencia).
NoAutorizado	Cuando una solicitud de conexión debe pasar primero por un proceso de autorización entonces se responde con este código.
NoEncontrado	Se devuelve este código como respuesta cuando el objeto solicitado no se encuentra en el servidor.
Prohibido	Cuando se esta tratando de acceder a un objeto que no esta permitido transferir (ni siquiera bajo autenticación) en un servidor. Esta respuesta depende de la implementación y de los criterios de seguridad que se

	apliquen. (para mayor claridad ver punto de Consideraciones en el perfil de envío de objetos).

Mas adelante podrá verse un ejemplo de interacción de solicitudes y respuestas entre cliente y servidor.

4.4.3 Iniciación del OBEX

Dado que el esquema de trabajo del protocolo OBEX está basado en cliente – servidor, se tiene que para poder establecer una sesión entre los dos elementos se necesita que tanto cliente como servidor estén encendidos en ambos lados de la comunicación; en el caso de que se requiera autenticación antes de establecer cualquier sesión, en ambos lados debe iniciarse la autenticación (con la correspondiente intervención del usuario), lo cual incluye la introducción de números PIN para ser utilizados por los cliente y servidor.

4.4.3.1 Establecimiento de una Sesión OBEX

Una vez que dos dispositivos han establecido una sesión con RFCOMM y conocen cada uno los servicios que se pueden encontrar en el otro (utilizando la información suministrada por el SDP), pueden intercambiar tramas de conexión con el fin de establecer una sesión de intercambio de objetos. Esta negociación se describe a continuación haciendo especial énfasis en el contenido de las tramas.

Naturaleza del mensaje	***Tipo de Trama y campos ó encabezados de contenido***	***Contenido y explicación***
Solicitud	Conectar	
	Longitud paquete	Longitud en Bytes de este paquete es un campo fijo
	Versión del OBEX	Estos son campos de parámetros fijos de esta trama, sirven para informar al otro dispositivo acerca de los parámetros de transferencia que el dispositivo soporta. Las banderas pueden tener un uso específico, como en el caso del perfil de transferencia de archivos (Ver más adelante punto 4.6 - Perfil de Transferencia de Archivos)
	Banderas	
	Longitud máxima de los paquetes	
	Destino	Es un encabezado que indica el servicio específico al cual se pretende conectar en el servidor
El dispositivo al cual va dirigida la trama solicita autorización antes de conectar		
Respuesta	No Autorizado	Código de respuesta que indica que se requiere autenticación
	Longitud del paquete	
	Versión del OBEX	El dispositivo servidor responde con sus propios parámetros de comunicación al igual que en la solicitud que le precede
	Banderas	
	Longitud máxima de los paquetes	

	Petición de autenticación	Cadena que le indica al Cliente si el acceso es completo ó de solo lectura, sugiere cual nombre de usuario y contraseña utilizar, y una cadena que es diferente cada vez que se envíe este paquete
El Cliente solicita de nuevo conectarse, ya con la autorización necesaria		
Solicitud	Conectar	
	Longitud paquete	Longitud en Bytes de este paquete es un campo fijo
	Versión del OBEX	Igual que en la primera solicitud
	Banderas	
	Longitud máxima de los paquetes	
	Destino	Igual que en la primera solicitud
	Petición de autenticación	Contiene una cadena idéntica a la que envió el servidor en su primera respuesta
	Respuesta de autenticación	Contiene el Nombre de usuario y contraseña
El servidor validó el usuario y contraseña		
Respuesta	Operación Exitosa	Código de respuesta del servidor indicando que el nombre de usuario y la contraseña son válidos
	Longitud paquete	Longitud en Bytes de este paquete es un campo fijo
	Versión del OBEX	Igual que en la primera respuesta
	Banderas	
	Longitud máxima de los paquetes	
	IDConexión	Identificador de conexión
	Quien	Debe coincidir con el campo destino del paquete
	Respuesta de autenticación	Es la misma que devolvió el cliente como respuesta de autenticación

Luego del establecimiento de la sesión, los objetos pueden transferirse mediante la utilización de tramas Put; esta transferencia ó intercambio puede tomar varias tramas, enviando un trozo del objeto en cada una.

4.4.3.2 Intercambio de Objetos

Las operaciones que se llevan a cabo para intercambiar objetos son básicamente dos: solicitar un objeto por su descripción ó por su nombre y transferir un objeto sin solicitud previa.

4.4.3.2.1 Envío de un Objeto

Mediante un ejemplo puede observarse la transferencia de un objeto sin solicitud previa mediante sucesivas tramas Put de un dispositivo a otro. Las primeras dos tramas se repiten cuantas veces sea necesario de acuerdo al número de trozos en los que se haya dividido el objeto.

Tabla 4. 5 Ejemplo del envío de un objeto

Naturaleza del mensaje	***Tipo de Trama y campos ó encabezados de contenido***	***Contenido y explicación***
Solicitud	Put	Indica la llegada de un objeto ó parte de el en esta trama
	Longitud paquete	Longitud en Bytes de este paquete. Campo fijo
	IDConexión	Identificador obtenido en el proceso de conexión a un servicio específico
	Nombre	Nombre del objeto a ser transferido
	Descripción	Descripción en texto breve del objeto (si la hay)
	(Otros parámetros del objeto)	Deben ser transferidos antes del cuerpo del objeto
	Cuerpo	Un trozo del cuerpo del objeto (primero)
Respuesta	Continúe	Se ha recibido a conformidad la trama anterior
	Longitud Paquete	Longitud en Bytes de este paquete
Se siguen transfiriendo varios trozos, confirmándose cada uno hasta que llega el último		
Solicitud	Put	Indica la llegada en esta trama del último trozo de objeto
	Longitud paquete	Longitud en Bytes de este paquete.
	IDConexión	Identificador obtenido en el proceso de conexión a un servicio específico
	Final de Cuerpo	Trozo final del cuerpo del objeto
Respuesta	Operación Exitosa	Indica que se acabo de transferir el objeto con éxito
	Longitud Paquete	Longitud en Bytes de este paquete

4.4.3.2.2 Obtención de un Objeto

Para obtener un objeto de un servidor, el procedimiento es muy parecido, pero el cliente utiliza el paquete Get para describir y solicitar el objeto en específico que necesita; el servidor encuentra el objeto y lo devuelve al cliente. El intercambio de paquetes ocurre de la siguiente manera:

Tabla 4. 6 Intercambio de paquetes para obtener un objeto del servidor

Naturaleza del mensaje	***Tipo de Trama y campos ó encabezados de contenido***	***Contenido y explicación***

Solicitud	Get	Indica la llegada de un objeto ó parte de el en esta trama
	Longitud paquete	Longitud en Bytes de este paquete. Campo fijo
	IDConexión	Identificador obtenido en el proceso de conexión a un servicio específico
	Tipo	Indica el tipo de objeto que se esta solicitando
	Nombre	Nombre del objeto solicitado
	(otros encabezados que se requieran)	Con el fin de ser más descriptiva la solicitud
Respuesta	Continúe	Indica la llegada de un objeto ó parte de el en esta trama
	Longitud paquete	Longitud en Bytes de este paquete. Campo fijo
	Nombre	Nombre del objeto a ser transferido
	Descripción	Descripción en texto breve del objeto (si la hay)
	(Otros parámetros del objeto)	Deben ser transferidos antes del cuerpo del objeto
	Cuerpo	Un trozo del cuerpo del objeto (primero)
Se siguen transfiriendo varios trozos mediante sucesivos Get - Continúe, hasta que llega el último		
Solicitud	Get	Indica la llegada en esta trama del último trozo de objeto
	Longitud paquete	Longitud en Bytes de este paquete.
Respuesta	Operación Exitosa	Indica que se acabo de transferir el objeto con éxito
	Longitud Paquete	Longitud en Bytes de este paquete
	Fin de Cuerpo	Trozo final del cuerpo objeto

Cuando un dispositivo solicita un objeto sin especificar su nombre, el protocolo devuelve el *objeto por defecto* que generalmente puede contener información sobre el servidor ó, como en el caso del perfil de envío de objetos (object push) una tarjeta de presentación virtual del usuario.

Quedan definidos así las operaciones de intercambio de objetos que necesita el perfil y que realiza el protocolo OBEX, la primera, el envío del objeto en forma espontánea (Push) y la obtención de un objeto solicitándolo al servidor (Pull).

4.4.3.2.3 Aplicaciones que Hacen Uso de Éste Perfil

Las aplicaciones que usan este perfil (como ya se ha podido ver), son aplicaciones de transferencia de archivos y de información compleja que requieren implementar las operaciones de Envío y obtención de objetos que define este perfil; este perfil es de servicio a otros tres perfiles que tienen que ver con aplicaciones más específicas: el perfil de envío de objetos, el de transferencia de archivos y el de sincronización. Pueden verse las aplicaciones en específico para tales perfiles.

4.5 Perfil de Envío de Objetos

Este perfil, llamado por la especificación “object Push” es una instanciación del perfil de Intercambio de Objetos, puesto que si bien el perfil de intercambio de objetos no es utilizado directamente por las aplicaciones, el perfil de envío de objetos sí es prácticamente un ejemplo de aplicación usando el protocolo OBEX, pero esto haciendo algunas consideraciones.

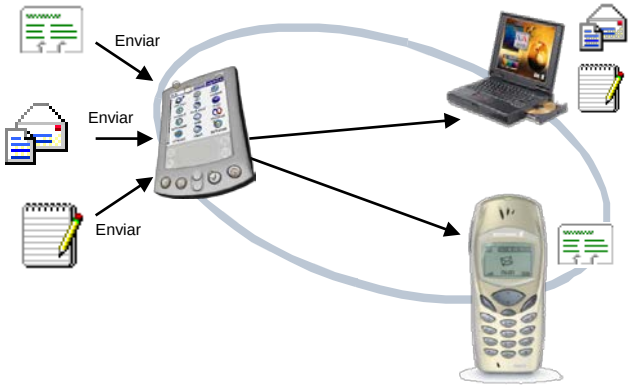


Figura 4.6 Aplicaciones típicas de envío de objetos

Una ejemplo de intercambio de información de manera típica en una reunión de negocios es el intercambio de tarjetas de presentación entre dos personas que han entablado previamente una conversación; La cercanía de las personas (y por ende, de sus PDAs) evidencia el hecho que se puede aprovechar dicha cercanía física para intercambiar esa misma información personal entre los dos PDAs (ó entre teléfonos celulares) sin que haya una tarjeta física de por medio.

Ese es el modelo de uso que inspira el desarrollo de esta y otras aplicaciones (que se describen mas adelante), el envío de información no solicitada a otro dispositivo, el cual es susceptible de aceptar ó no tal transferencia; también existe la posibilidad que la misma transferencia se dé en el sentido contrario, realizándose entonces un intercambio bidireccional.

La idea de intercambiar información de esta manera puede sonar como un hueco en la seguridad para los dispositivos, pero en realidad esto significa que la intervención del usuario debe hacer que se autorice el intercambio.

Poniendo la aplicación en términos de procedimientos, se pueden separar tres posibles casos de uso:

- Una persona quiere enviar su información a otra (el caso mas común)
- Una persona quiere obtener la información de otra persona
- Las dos personas desean intercambiar su información personal

Los dos primeros casos obedecen a operaciones básicas de intercambio de objetos; el tercero podría verse como una combinación de los otros dos, pero en realidad el caso más importante es el primero y es el de implementación obligatoria al cumplir con este perfil de especificación.

También este perfil de aplicación comparte muchas similitudes con los de transferencia de archivos ó de sincronización, puede que esto no sea tan evidente desde la perspectiva del usuario, pero si desde la del programador, ya que la funcionalidad que se implementa del protocolo OBEX bien podría reutilizarse para casi todos los casos.

Volviendo a los tres casos de uso se puede describir cada aplicación en términos de lo que ve el usuario y su correspondencia con la implementación de la funcionalidad OBEX. Cabe anotar que el dispositivo al que se envía ó del que se obtiene un objeto es llamado servidor de envío de objetos; quien los envía es el Cliente de envío de objetos. Los pasos que generalmente se siguen en cada caso empiezan con una solicitud de conexión (sin especificar el destino de tal conexión, es decir, ningún servicio en particular). Se puede complementar este proceso de conexión con autenticación, lo cual significa que en algún momento se deberá introducir un código de identificación.

4.5.1 Envío Simple del Objeto de Información

En este caso el objeto (la tarjeta de presentación como ejemplo típico) se transfiere de manera unidireccional y sin solicitud previa con los procedimientos OBEX descritos para tal fin.

Usuario	*Aplicación de envío de objetos*
El usuario del dispositivo que va a recibir (servidor) configura la recepción de objetos	El Dispositivo servidor inicia el intercambio de objetos (esto debe incluir

(por ejemplo, la tarjeta virtual de presentación).	un registro de dicho servicio ante el protocolo de descubrimiento)
El usuario indica al dispositivo cliente que quiere hacer un envío de un objeto. Selecciona de una lista el dispositivo al cual se va a enviar[7].	Envío de una trama de conexión sin utilizar el atributo *Destino.*
El usuario del dispositivo cliente ordena enviar el objeto (la tarjeta).	
El usuario del dispositivo servidor recibe una notificación de que ha llegado un objeto. El puede aceptarlo ó rechazarlo. Se supone que lo acepta.	Envío de tramas sucesivas *Put* con el contenido del objeto tal como esta descrito en el perfil de intercambio de objetos.
Al Cliente se le indica que la operación se realizó normalmente	Desconectar del servidor de intercambio OBEX.

4.5.2 Solicitud Simple de Información

La solicitud simple de información implica que se debe hacer una solicitud por parte de la aplicación lo más sencilla posible, puesto que el intercambio de tarjetas es una operación que no tiene por qué demorarse, de tal manera que el objeto que se está solicitando debe solicitarse unívocamente por medio de su tipo de contenido.

El cliente debe solicitar el objeto tarjeta virtual (cuyo formato fue definido por el Internet Mail Consortium en 1996, Ver Bibliografía de este documento).

Cabe anotar que el Protocolo OBEX define que se puede establecer un objeto por defecto para devolverlo en caso de recibir una solicitud Get sin una descripción detallada; el objeto por defecto puede establecerse como la tarjeta de presentación virtual de tal manera que sea devuelta cada vez que se reciba una solicitud Get por parte de una aplicación de envío de objetos (en el caso de intercambio de tarjetas).

[7] Previa consulta realizada mediante el perfil de Acceso Genérico sobre qué dispositivos hay y cuales de ellos soportan intercambio de objetos (mediante una consulta al servicio de descubrimiento).

Usuario	*Aplicación de envío de objetos*
El usuario del dispositivo al que se le va a solicitar el objeto (servidor) configura la solicitud de objetos (por ejemplo, la tarjeta virtual de presentación).	El Dispositivo servidor inicia el intercambio de objetos (esto debe incluir un registro de dicho servicio ante el protocolo de descubrimiento)
El usuario indica al dispositivo cliente que quiere solicitar un objeto. Selecciona de una lista el dispositivo al cual se le va a solicitar. *	• Envío de una trama de conexión sin utilizar el atributo *Destino.*
El usuario del dispositivo cliente ordena solicitar el objeto (la tarjeta del otro dispositivo).	• Trama Get desde el cliente al servidor solicitando el objeto y dando como tipo de objeto el formato “text/x-vcard”.
El usuario del dispositivo servidor recibe una indicación de que se ha solicitado una tarjeta de presentación. Se puede ó no permitir que esta transacción se complete.	• Tramas sucesivas Get del Cliente al servidor solicitando la tarjeta de presentación. • La aplicación servidora responde con tramas de continuación sucesivas que contienen el objeto. Termina con una trama “success”, tal como se describe en el protocolo de intercambio de objetos.
Al Cliente se le indica que la información ya está almacenada en el equipo.	Desconectar del servidor de intercambio OBEX.

* Previa consulta realizada mediante el perfil de Acceso Genérico sobre qué dispositivos hay y cuales de ellos soportan intercambio de objetos (mediante una consulta al servicio de descubrimiento).

4.5.3 Intercambio Bidireccional de Información

Usuario	*Aplicación de envío de objetos*
El usuario del dispositivo al que se le va a solicitar el objeto (servidor) configura el intercambio de objetos (por ejemplo, la tarjeta virtual de presentación).	El Dispositivo servidor inicia el intercambio de objetos (esto debe incluir un registro de dicho servicio ante el protocolo de descubrimiento)

El usuario indica al dispositivo cliente que quiere hacer un intercambio de un par de objetos. Selecciona de una lista el dispositivo al cual se le va a solicitar. *	Envío de una trama de conexión sin utilizar el atributo *Destino.*
El usuario del dispositivo cliente ordena realizar el intercambio de los objetos (las tarjetas ambos dispositivos).	
El usuario del dispositivo servidor recibe una notificación de que ha llegado un objeto. El puede aceptarlo ó rechazarlo. Se supone que lo acepta.	Envío de tramas sucesivas *Put* del cliente al servidor con el contenido del objeto tal como esta descrito en el perfil de intercambio de objetos.
El usuario del dispositivo servidor recibe una indicación de que se ha solicitado una tarjeta de presentación. Se puede ó no permitir que esta transacción se complete.	• Tramas sucesivas Get del Cliente al servidor solicitando la tarjeta de presentación con el campo *tipo* establecido en “text/x-vcard”. • La aplicación servidora responde con tramas de continuación sucesivas que contienen el objeto. Termina con una trama “success”, tal como se describe en el protocolo de intercambio de objetos.
Al Cliente y al servidor se les indica que la información ya está almacenada en los equipos.	Desconectar del servidor de intercambio OBEX.

* Previa consulta realizada mediante el perfil de Acceso Genérico sobre qué dispositivos hay y cuales de ellos soportan intercambio de objetos (mediante una consulta al servicio de descubrimiento).

En este caso se intercambian los objetos en ambos sentidos, lo cual es una mezcla de los dos casos anteriores.

4.5.4 Consideraciones

4.5.4.1 Consideraciones con el SDP

Para todos los casos es de esperar que el servicio esté debidamente registrado ante el protocolo de descubrimiento de servicios, puesto que en él se detallan los tipos de datos ó de objetos que va a soportar el dispositivo servidor, por ejemplo, en el caso de las tarjetas de presentación virtuales, el tipo de objeto es el *Text/x-vcard* por ejemplo, así como también pueden existir otros formatos, pero el cliente (aplicación) debe darse cuenta primero de si este tipo de contenido está soportado por el otro dispositivo antes de realizar la transferencia ó el intercambio. Para poder que los dispositivos no tengan problemas al intercambiarse objetos, es conveniente que el servidor defina los siguientes parámetros ante el protocolo de descubrimiento de servicios.

Tabla 4.7 Registros del servicio para envío de objetos

Parámetros			***Valor***
Lista de Clases de Servicio	Clase de servicio0		OBEXObjectPush
Lista de Descriptores de protocolos	Protocolo0		L2CAP
	Protocolo1		RFCOMM
		Parámetro0	<numero de canal servidor>
	Protocolo2		OBEX
Nombre del servicio (opcional)			OBEXObjectPush
Lista de Descriptores de Perfil (Opcional)	Perfil0		OBEXObjectPush
		Versión0	<Número de versión soportada>
Lista de formatos soportados			VCard2.1, vCard3.0, vCal1.0, vCal2.0, vNote, vMessage, u otros tipos de contenido

4.5.4.2 Otras Respuestas del Servidor de Envío

Otra característica que se debe incluir en la implementación de una aplicación cubierta por este perfil es la respuesta en caso de no encontrarse el objeto buscado en el servidor, el cual debe responder con el código de respuesta (ver códigos de respuesta en la descripción del perfil de intercambio de objetos) “NO ENCONTRADO”. De la misma manera el Cliente deberá implementar un filtro para reconocer este código y enviar un mensaje al usuario indicando esta situación.

También al intentar obtener un objeto por defecto, como en este caso, el cliente no puede dar sino solamente el atributo *Tipo* como argumento para que el servidor busque el objeto; si se especifica un nombre en la solicitud, esto puede significar que la aplicación no esta

tratando simplemente de “intercambiar tarjetas” (sino probablemente de buscar información sin permiso), por ello el servidor deberá responder con un código de “Prohibido”.

4.6 Perfil de Transferencia de Archivos

Los dispositivos de computación, tanto fijos como móviles donde tiene lugar algún procesamiento de información, no serían de mucha utilidad si su información no pudiera ser transportada de un equipo a otro, puesto que generalmente los documentos y archivos se crean, modifican, imprimen y consultan en algún lugar diferente; por ejemplo, si se descarga algún documento de trabajo proveniente de Internet, es probable que se almacene en algún sitio de trabajo, como una oficina, por ejemplo; la persona que quiera consultarlo puede querer llevarlo a su casa ó a otra oficina para imprimirlo, puede transportarlo ya sea a traves de una unidad de almacenamiento portátil ó descargarlo en un PDA (pudiendo consultarlo en cualquier momento).

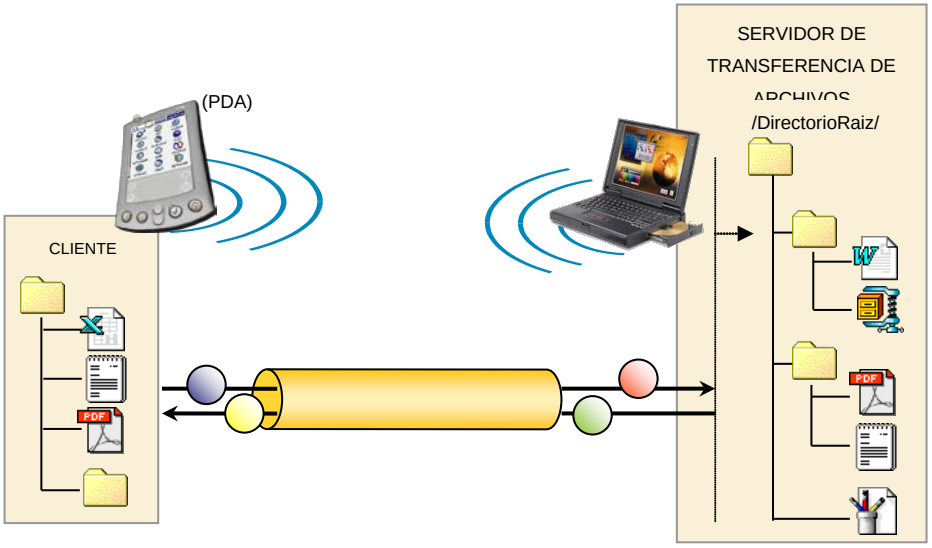


Figura 4.7 Esquema de transferencia de archivos en un caso típico

Se vé que en casos como este hay una descarga de archivos constante de un lugar a otro. Aplicaciones que, si se miran con detenimiento, existen ya en los dispositivos que manejan archivos y que pueden funcionar sobre diferentes interfaces físicas. Bluetooth tiene la facilidad de poder dar paso a estas aplicaciones gracias al perfil de transferencia de archivos.

Este perfil aprovecha las capacidades de intercambio de objetos que ya se ha venido estudiando y que son muy similares a las aplicaciones con arquitectura cliente – servidor existentes, es decir, un cliente realizando una petición y posteriormente esperando una respuesta de parte del servidor.

4.6.1 Características Básicas de una Transferencia de Archivos

Cuando se establece una sesión en un servidor FTP, generalmente surgen en la interfaz de usuario varias características necesarias y desde el punto de vista de aplicación, varios requerimientos técnicos:

Características	*Requerimientos Técnicos*
Tener una visión de cuales son los directorios que ofrece el otro dispositivo. Usualmente uno de los dispositivos (el que quedaría asumiendo un papel de servidor) ofrece un directorio de partida ó un directorio por defecto ó “compartido”. A partir de aquí se debe poder explorar la estructura de directorios consultando qué archivos y subdirectorios contiene.	Existencia de un formato para describir y enviar la información de directorios al Cliente sin tener que enviar todo su contenido. Ejecución de comandos que permitan al servidor encontrar la información requerida por el cliente
Ofrecer la posibilidad de transferir archivos hacia el servidor y desde él de acuerdo a lo que seleccione el usuario teniendo en cuenta la información obtenida en la exploración de la estructura de directorios.	Implementación de mecanismos de solicitud y envío de objetos al servidor
Poder crear directorios y poder borrar archivos y directorios; no sería lógico permitir crear archivos, puesto que si el archivo pudiera ser creado, no se podría modificar a través de la sesión de transferencia de archivos (seria lo mismo que crear el archivo en el dispositivo local, editarlo y luego transferirlo). Los directorios si se pueden crear, puesto que en ellos pueden seguirse almacenando archivos ó más directorios.	Existencia de comandos desde el cliente y respuestas de parte del servidor para administrar archivos y directorios de manera remota. Considerar permisos de escritura ó de solo lectura para el cliente ó para varios clientes

Usualmente un sistema de transferencia de archivos nos ofrece cierta funcionalidad fija para poder cumplir con las anteriores características:

1. Moverse a un directorio especificado
2. Subir un nivel en la estructura de directorios
3. Subir hasta el directorio raíz
4. Crear un directorio nuevo
5. Borrar un directorio
6. Borrar un Archivo
7. Copiar un archivo
8. Copiar un directorio con su contenido
9. Extraer el contenido de un directorio (abrirlo)
10. Extraer un archivo

La aplicación de transferencia de archivos típica debe disponer de esta funcionalidad sobre el sistema de archivos remoto (el del servidor), por ello se necesita una "plataforma" que nos brinde la manera de implementar las funciones anteriores; este soporte lo brinda el protocolo de intercambio de objetos OBEX que se explico en el punto 4.4.2. A continuación se describen las características que posee OBEX para brindar este soporte.

4.6.2 Características que brinda el Protocolo OBEX para Transferir Archivos

El protocolo OBEX y el perfil asociado de intercambio de objetos facilitan la ejecución de las funciones necesarias para el transporte de los archivos de un equipo a otro, dada su capacidad de llevar objetos completos entre clientes y servidores. Aquí hay algunas definiciones que hacen que las funciones descritas en el punto anterior puedan llevarse a cabo por medio de la utilización del protocolo OBEX.

4.6.2.1 La Ruta de Directorios del Servidor

El servidor de transferencia de archivos posee una estructura de directorios que debe poderse recorrer de manera remota desde el cliente. Así también dicha estructura debe originarse desde un directorio llamado Directorio Raíz (que es el directorio en el que por defecto quedará ubicado el servidor cuando se establezca una sesión de transferencia de archivos en él). OBEX tiene una trama para realizar este manejo, y es la trama SetPath ó EstablecerRuta (ver Perfil de intercambio de objetos OBEX), la ruta "actual" de directorios en el servidor puede cambiarse y de esta manera se puede navegar por la estructura de

directorios. La trama EstablecerRuta puede usarse de varias formas para obtener diferentes resultados dependiendo de cómo se configuren los campos de la trama. En la trama EstablecerRuta participan los campos definidos por OBEX: Código de trama, Nombre (encabezado opcional), Largo de la trama, un campo de Banderas, Constantes y encabezados opcionales; pero en realidad cobran importancia solamente el Nombre y las banderas, la trama es como sigue:

Código de Trama	Longitud del paquete	Banderas								Constantes								Otros Encabezados
EstablecerRuta	<Longitud en Bytes>	S	N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<opcionales>

En este caso las constantes son un campo no utilizado por la versión actual de OBEX (versión 1.2). De las banderas solo están definidas dos de ellas: S y N que son *SubirNivel* y *NoCrearDirectorio.* En las tramas debe incluirse un encabezado en el espacio de encabezados opcionales además del Nombre, este es el del identificador de Conexión descrito en el perfil OBEX: IDConexión; este encabezado describe la conexión que se debió establecer previamente entre los dos dispositivos que intercambian archivos.

Las combinaciones de uso de esta trama resultan en lo siguiente, (cabe anotar la correspondencia con las 4 primeras funciones del punto 4.6.1 - Características Básicas de una Transferencia de Archivos).

Tabla 4.8 Configuraciones de la trama EstablecerRuta

Código de Trama	***Cadena de Nombre***	***Banderas***		***Acción que Ejecuta en el Servidor***
		S	N	
EstablecerRuta	“Documentos”	0	1	Establece la ruta actual de directorios a la cadena: RutaActual = “RutaAnterior/Documentos/”
	<Vacío>	1	1	Sube un nivel en el árbol de directorios
	<Vacío>	0	1	Sube hasta el directorio Raíz
	“Imágenes”	0	0	Crea un directorio llamado “imágenes” dentro del directorio actual en el servidor

Los códigos de trama de las respuestas del servidor a estas solicitudes podrían ser los siguientes:

Operación Exitosa	Ocurre en caso de no haber problemas con la operación anterior.
NoEncontrado	Se devuelve en caso que el directorio solicitado no se haya encontrado ó que ya se ha llegado al directorio Raíz al intentar subir un nivel.
NoAutorizado	Ocurre si no está permitido entrar al directorio seleccionado ó si se pretende

crear un directorio nuevo dentro de un directorio al que se tiene acceso de solo lectura.

Con base en estos códigos de respuesta, la aplicación Cliente puede reaccionar para saber qué mensajes enviar al usuario del dispositivo.

4.6.2.2 Descripción del Contenido de los Directorios

Cuando un Cliente solicita a un servidor entrar a un directorio de la forma como se describió en el punto 4.6.2.1, surge para la aplicación la necesidad de saber qué archivos contiene el directorio y tal vez saber de ellos su tamaño y fechas de modificación sin tener que traerlos todos al equipo cliente; esta información queda resumida, según este concepto, a una lista de elementos (archivos y carpetas), cada uno de ellos con sus propiedades. Una forma sencilla de organizar esta lista es la que plantea OBEX de utilizar documentos basados en XML[8] para realizar la descripción; a menudo en aplicaciones de Internet se emplea XML para transportar objetos, y el listado del contenido de un directorio puede tratarse también como tal; por ello es susceptible de ser descrito mediante un documento XML. El tipo MIME de este objeto es denominado experimental, es “x-obex/folder-listing” y puede usarse como tipo de contenido en las tramas Get como se verá más adelante. La definición de este tipo de objeto se encuentra en el documento de especificación de IrOBEX relacionado en la Bibliografía.

Un ejemplo de documento que describe un objeto x-obex/folder-listing puede ser el siguiente:

```
<?xml version="1.0"?>
<!DOCTYPE folder-listing SYSTEM "obex-folder-listing.dtd">
<folder-listing version=”1.0”>
      <parent-folder />
      <folder name = “Documentos” created=”20020312T141500Z”/>
      <folder name = “Imagenes” created=”19950330T105000Z”/>
      <file name = “Notas.txt” created=”19971209T090300Z” size=“6672”
      modified=“19971222T164100Z” user-perm=”RW”/>
      <file name =”Obex.doc” created=“19970122T102300Z” size =
      “41042” type=”application/msword”
      modified=“19970122T102300Z”/>.
</folder-listing>
```

[8] Extensible Markup Language

Este documento nos dice que hay dos directorios y dos archivos en el directorio seleccionado; también indica la existencia de un directorio padre en el cual se encuentra este contenido. (“Parent Folder”).

De esta manera, el Cliente puede tener en su memoria una descripción del contenido de cualquier directorio de los que ofrece servidor, estableciendo primero la ruta actual de directorios del servidor y luego solicitando la información del directorio en el formato anteriormente descrito. Esta solicitud se hace como cualquier solicitud de un objeto, tal y como la describe el protocolo OBEX, solo que el tipo de objeto que se solicita sería un objeto tipo *x-obex/folder-listing.* A continuación en el punto 4.6.2.3.1 - Bajar un Archivo ó Directorio del Servidor, se describe como se hace esta solicitud.

También hay que anotar que la aplicación cliente debe poder procesar el documento XML y poder extraer de allí la información necesaria para poder mostrar al usuario el contenido de la carpeta. Esta característica es propia de la implementación que se haga del cliente.

4.6.2.3 Funciones de Transferencia de Archivos

De acuerdo a los procedimientos descritos por el OBEX para intercambiar objetos, tenemos el envío de tramas Get y Put para solicitar y enviar archivos hacia el servidor; también se puede solicitar un objeto *x-obex/folder-listing* para conocer el contenido de un directorio. Las tramas para el transporte de esta información se describen a continuación.

4.6.2.3.1 Bajar un Archivo ó Directorio del Servidor

Primero que todo, para poder bajar algún archivo ó directorio del servidor hay que solicitarlo con una trama Get, pero para solicitarlo es necesario tener el nombre de tal elemento y ello solo se obtiene del objeto Folder-listing que se extraiga del servidor. La trama Get puede solicitar este objeto y también archivos.

Tabla 4.9 Descripción de la trama GET

Código de Trama	***Cadena de Nombre***	***Cadena Tipo***	***Acción que Ejecuta en el Servidor***
Get	<Vacío>	“x-obex/folder-listing”	Solicita la descripción del contenido del directorio actual del servidor
	“Notas.txt”	<vacío>	Solicita el archivo Notas.txt que esta ubicado en el directorio actual del servidor

Cada una de estas tramas desencadena un proceso de transferencia de un objeto; luego de cada una de estas tramas hay una confirmación del servidor y seguidamente otra solicitud *Get*, hasta que se completa la transferencia, tal y como se describe en el perfil de intercambio de objetos OBEX en el punto 4.4.3.2.2- Obtención de un Objeto. Se puede notar la correspondencia con las funciones 9 y 10 que se mencionan en el punto 4.6.1.

Para el caso de bajar del servidor directorios completos, puede subdividirse la operación en las mismas funciones anteriores, puede utilizarse las tramas *EstablecerRuta* y *Get* (solicitando contenido de directorio) para explorar la estructura de directorios y buscar el directorio deseado; con esta información se puede reproducir la estructura de cualquier directorio en el cliente y luego realizar varios *Get* para obtener cada archivo por separado. Esta característica es parte de la implementación de la aplicación cliente.

4.6.2.3.2 Subir un Archivo ó Directorio al Servidor

Para esto el Cliente debe saber a que directorio va a subir (ó enviar en términos del OBEX) el archivo ó directorio, para ello utiliza las tramas EstablecerRuta explicadas antes y una vez hecho esto puede subir el elemento con tramas sucesivas *Put.* El proceso se lleva a cabo igual que como se transfiere un objeto en OBEX, explicado en el perfil OBEX en el punto 1.1.2.3.1 – Envío de un Objeto.

Tabla 4.10 Descripción de la trama PUT

Código de Trama	***Cadena de Nombre***	***Acción que Ejecuta en el Servidor***
Put	"IrObex.pdf"	Empieza a subir el archivo IrObex.pdf al directorio actual en el servidor (primero trozo del archivo)

Luego de esto ocurren de repetidas confirmaciones del Servidor y repetidas tramas *Put* con trozos del archivo hasta que se transfiere por completo. El servidor puede retornar en el caso que el directorio sea de solo lectura un código de trama *NoAutorizado.*

Para subir un directorio completo, es cuestión de la aplicación convertir esta operación en varias operaciones con *EstablecerRuta* para reproducir el árbol de subdirectorios en el servidor y luego enviar cada archivo con Tramas *Put* como se acaba de describir.

4.6.2.4 Funciones Adicionales de Manipulación

Las operaciones de manipulación son aquellas que permiten realizar algún cambio de los contenidos ya existentes en el servidor, tales operaciones son las de crear directorios, y

borrar Archivos y directorios del servidor; la operación de creación de directorios se explicó en el punto 4.6.2.1 - La Ruta de Directorios del Servidor.

4.6.2.4.1 Borrar un Archivo ó Directorio en el Servidor

En este caso, se utiliza la trama *Put* de una manera diferente; los encabezados que tienen importancia aquí son el Encabezado Nombre y el encabezado de Cuerpo. El encabezado Cuerpo Vacío indica que lo que se está haciendo es un borrado.

Tabla 4.11 Descripción de la trama PUT para operación de borrado.

Código de Trama	***Cadena de Nombre***	***Cuerpo***	***Acción que Ejecuta en el Servidor***
Put	"Notas.txt"	<no hay encabezado de cuerpo>	Borra el archivo Notas.txt ubicado en el directorio actual del servidor
	"Imagenes"	<no hay encabezado de cuerpo>	Borra el directorio Imágenes ubicado en el directorio actual del servidor

Los códigos de respuesta a estas solicitudes (las cuales tienen lugar una sola vez por ser una especie de "comandos" más que transferencias de archivos) por parte del servidor pueden ser los siguientes.

Tabla 4.12 Códigos de Respuesta para el caso de manipulación de archivos

Operación Exitosa	Ocurre en caso de no haber problemas con la operación anterior.
NoEncontrado	Se devuelve en caso que el directorio ó archivo solicitado para borrar no se haya encontrado en el directorio actual.
NoAutorizado	Ocurre si se pretende borrar un archivo ó directorio dentro de un directorio al que se tiene acceso de solo lectura ó si el archivo ó directorio a borrar tiene permisos de solo lectura.

Algunos servidores de transferencia de archivos no permiten borrar un directorio entero que no esté vacío; este caso es criterio de diseño y se deja en manos de la implementación específica que se requiera.

4.6.3 Consideraciones

4.6.3.1 Consideraciones para el Establecimiento de la Sesión OBEX

La inicialización del OBEX ocurre como se describió en el perfil OBEX (ver Establecimiento de una sesión OBEX), pero en la primera solicitud que hace el cliente para conectarse al servicio, el campo *Destino* debe ser diferente y establecerse en un valor fijo ó UUID igual en Hexadecimal a F9EC7BC4-953C-11D2-984E-525400DC9E09 (16 bytes). Este valor

identifica unívocamente el *Servicio de Exploración de Directorios*[9]*.* En el perfil de Envío de Objetos el establecimiento de la sesión OBEX se hacía sin especificar un Encabezado de *Destino,* puesto que lo que se quería obtener era un objeto por defecto. Ahora se quiere acceder a un servicio algo más complejo y más preciso, por ello debe usarse el UUID adecuado para exploración de directorios.

También es de importancia que todas las tramas que se intercambien deben contener el encabezado IDConexion obtenido durante el establecimiento de la sesión, esto le dice al servidor la conexión sobre la cual se está trabajando (dada la situación que pueda darse de multiplexión del servidor por medio de varias conexiones).

4.6.3.2 Modo de Descubrimiento

El dispositivo que desee quedar habilitado como servidor de transferencia de archivos, (después de quedar en este modo por intervención del usuario, lo cual se recomienda) debe quedar en el modo de descubrimiento limitado y registrar en 1 el bit de transferencia de objetos presente en la palabra CoD[10]. Con estos mecanismos solamente los dispositivos que saben que van a encontrar un servidor de transferencia a su alrededor, pueden encontrarlo y distinguir cuales ofrecen dicho servicio, puesto que saben desde el proceso de indagación (Ver Capítulo 2 – Banda base).

4.6.3.3 Consideraciones con el SDP

El registro con el SDP es de importancia para este servicio de transferencia de archivos; en el se registra que hay un servicio genérico de transferencia de archivos que pertenece al perfil de transferencia de archivos y que está soportado por el protocolo de sesión OBEX.

Tabla 4.13 Registros de servicio correspondientes

Parámetros			***Valor***
Lista de Clases de Servicio	Clase de servicio0		OBEXFileTransfer
Lista de Descriptores de protocolos	Protocolo0		L2CAP
	Protocolo1		RFCOMM
		Parámetro0	<numero de canal servidor>
	Protocolo2		OBEX
Nombre del servicio (opcional)			OBEXFileTransfer
Lista de Descriptores de Perfil (Opcional)	Perfil0		OBEXFileTransfer
		Versión0	<Número de versión soportada>

[9] Denominado por la especificación IrOBEX como Folder Browsing Service

[10] Class of Device, palabra que se intercambia durante el proceso de respuesta a la indagación.

4.7 Perfil de Sincronización

La sincronización es un procedimiento mediante el cual dos dispositivos personales (que regularmente manejan la misma clase de información y pertenecientes al mismo usuario) mantienen copias idénticas de dicha información en ambos dispositivos con el fin de que pueda ser consultada y procesada en ambos de manera independiente cuando estos no se encuentren cerca. Ejemplos de esta información son los datos de direcciones y numeros telefónicos, Las notas rápidas, mensajería y calendario; esta información puede modificarse ya sea en un PC de escritorio, un portátil, un PDA ó u teléfono celular. Una persona puede manejar esta información fácilmente en un PDA

Figura 4.8 Aplicación de sincronización entre dos dispositivos

Las aplicaciones que existen en éste campo son variadas, y asi mismo son los formatos de comunicaciones que asume cada una de ellas, lo cual hace que una interacción entre aplicaciones de diferente fabricante y sobre diferentes plataformas (PC ó PDA) sea casi imposible. Este perfil pretende adaptar para estas típicas aplicaciones de dispositivos móviles (que regularmente funcionan sobre cable) dos factores de convergencia:

- Garantizar la interoperabilidad de aplicaciones de este tipo
- Permitir a estas aplicaciones utilizar un medio de transporte diferente a los cables convencionales para ejecutar estas operaciones; es decir, utilizar los servicios de las capas inferiores de la pila de protocolos de Bluetooth.

4.7.1 Características de una Aplicación de Sincronización

Dos aplicaciones diferentes en dos dispositivos de cómputo para manejo de la información personal (notas, mensajes, contactos, calendario), regularmente poseen rutinas de sincronización, para poder comunicarse con su aplicación equivalente en el otro dispositivo y cumplir el objetivo mencionado antes (esto se llama regularmente PIM[11]). La especificación Bluetooth ofrece medios para que estas rutinas se lleven a cabo.

4.7.1.1 Sincronización de Items en dos Dispositivos de Cómputo

La sincronización de manera general se basa en mantener información idéntica en dos dispositivos de cómputo de acuerdo a los cambios que esta sufra en alguno de los dos de manera separada. Para esto las aplicaciones deben conocer qué elementos de cada dispositivo son nuevos para poder compararlos y actualizarlos en ambas partes; este procesamiento tiene lugar en uno solo de los dispositivos, el cual debe asumir la tarea de comunicarse con el otro, buscar la información nueva y copiar en ambos lados la información para que quede igual.

Desde el punto de vista de quién es el iniciador de la comunicación, tenemos que el equipo que solicita la sincronización y realiza el procesamiento mediante un *motor de sincronización* se denomina Cliente de sincronización; el equipo que recibe la solicitud es el servidor. Se dice entonces por ejemplo que un PDA (Servidor) se sincroniza con un computador personal (Cliente) y es en el cliente donde se lleva a cabo la mayor parte del procesamiento.

La sincronización, según lo anterior puede verse como un intercambio de objetos genérico con algunas consideraciones y con otras características de procedimiento según se verá a continuación.

4.7.1.2 Sincronización de Objetos

La sincronización entre dos dispositivos se puede asimilar como una comparación entre dos conjuntos de objetos en los dos dispositivos; la aplicación debe mirar las diferencias de contenido entre estos conjuntos y actualizarlos en ambos lados por medio de un intercambio de los objetos diferentes; esto no quiere decir que los objetos se almacenarán en el mismo formato en ambos lados, sino que el hecho de convertir la información en “objetos” sugiere la

existencia de un formato que se debe usar para ellos mientras ocurre el intercambio de estos objetos, de ahí en adelante las aplicaciones encargadas de administrar las información personal ya mencionada (notas, mensajes, calendario, contactos, etcétera), se encargará de hacer la conversión a su propio formato y continuar su funcionamiento después de la sincronización.

Bajo estas premisas, los papeles de Cliente y Servidor de sincronización que se definieron en el punto 4.7.1.1 quedan viéndose como Cliente y Servidor de intercambio de objetos OBEX (ver Perfil de Intercambio de objetos), lo cual hace que la sincronización pueda abrirse paso como una aplicación de tal tipo. La Figura 4.9 puede da una idea de cómo esta constituido el sistema.

Cliente y Servidor Inician una sesión, el servidor ofrece la información personal que posee en el formato de objetos definido, permite que el cliente los extraiga y posteriormente (luego de la comparación y actualización) los envíe de nuevo al servidor.

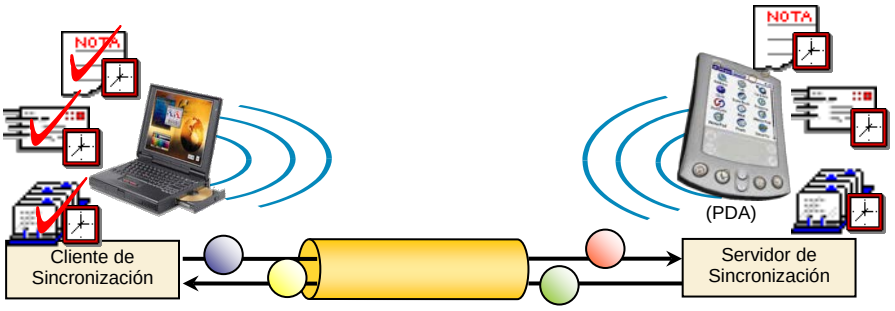


Figura 4.9 La sincronización vista como un intercambio de objetos

4.7.1.3 Los Servicios del Perfil OBEX y la Especificación IrDA

Ya que se concluyó como puede implementarse fácilmente un motor de sincronización entre los dos dispositivos mediante la reutilización de los servicios de OBEX, hay que contemplar qué opciones nos ofrece OBEX para realizar el intercambio y si hay definiciones y características nuevas que el protocolo adiciona para dar un mejor soporte a estas aplicaciones. También las aplicaciones de Sincronización, soportándose en OBEX han sido cubiertas por la especificación IrDA, la cual define procedimientos para comunicaciones

[11] Personal Information Management

móviles en el volumen de tal especificación denominado IrMC (ver referencia en la Bibliografía). Si se garantiza el cumplimiento de las definiciones adicionales puestas por tal especificación, se tendrá entonces un resultado extra: la interoperabilidad de aplicaciones que antes funcionaban con IrDA para que lo hagan a través de Bluetooth.

4.7.1.3.1 Definiciones

El protocolo OBEX que reutiliza la especificación de Bluetooth está definido en la especificación de la tecnología Infrarroja IrDA, en conjunto con IrMC para comunicaciones móviles. Esta especificación define algunos parámetros para las aplicaciones que involucren objetos, y particularmente, sincronización.

Registro de Cambios	Llamado por la especificación *Change Log*, es un objeto a manera de texto que indica los cambios que se han realizado a un determinado *Almacén de Objetos*, registra ya sea el orden en que han ocurrido los cambios, ó bien la fecha en que han ocurrido y a qué objetos de información personal. Informa del número serial del dispositivo y de si los cambios han sido por modificación, ó por borrado (borrado del sistema operativo para ahorrar espacio ó por parte de la aplicación). Este registro queda vacío cada vez que hay un traslado completo de información del almacén de objetos ó un reset del sistema.
Almacén de Objetos	Es una base de datos que contiene objetos definidos por IrMC, es decir, objetos de información personal (Notas, contactos, calendario, mensajes). Es el lugar al cual accede el cliente para extraer objetos (su concepto se asemeja al de la estructura de directorios del servidor de transferencia de archivos).
Sincronización Lenta	Cuando dos dispositivos se sincronizan por primera vez, lo más probable es que primero tengan que autenticarse y luego transferirse todo el contenido de sus Almacenes de Objetos, este procedimiento es más lento en comparación con las siguientes veces que se haga de nuevo la sincronización (Sincronización Rápida).
Sincronización Rápida	Dos dispositivos que ya han establecido un lazo de confianza (se han autenticado previamente) pueden utilizar la información que provee el registro de cambios (El cliente lo solicita como objeto al servidor) para intercambiar solamente los objetos que hayan cambiado y que sean diferentes ó nuevos en ambos almacenes de objetos. Este procedimiento hace mucho más breve la sincronización.
Registro de Información	Es un objeto que retiene información sobre un almacén de objetos específico, (por ejemplo, el almacén de objetos calendario) acerca de qué

tipo de objeto reside en dicho almacén, cuantos de ellos hay, y cuantos más podrá almacenar.

4.7.1.3.2 Funcionalidad OBEX Aplicada a la Sincronización

La funcionalidad que presta OBEX dentro del entorno de sincronización, es la de ofrecer los procedimientos de intercambio de objetos descritos en el Perfil Genérico de Intercambio de objetos OBEX, El procedimiento de Sincronización en sí lo define la especificación IrMC, pero utiliza la funcionalidad anteriormente mencionada.

4.7.2 Interoperabilidad entre Aplicaciones de Sincronización

Ya se había mencionado que para garantizar la interoperabilidad entre las aplicaciones de manejo PIM era necesario establecer formatos fijos para la información que se va a intercambiar, es decir, información personal y de control para la sincronización; a continuación están algunas definiciones de tales formatos.

4.7.2.1 Formato de la Información Intercambiada

4.7.2.1.1 Registro de Cambios

Es un objeto que contiene información de cambios a partir de cierto instante de tiempo. Detalla qué objetos han cambiado y cual ha sido este cambio. Un Registro de Cambios es típicamente asi:

`SN:1218182THD000001-2`	Numero Serial de la máquina.
`DID: 03df30423`	Identificador del almacén de objetos
`Total-Records:4`	Numero total de objetos contenidos en el almacén
`Maximum-Records:50`	Máximo número de objetos que se puede guardar
`M:4:19990104T180000Z:0A456566`	Dos Objetos modificados en las fechas dadas.
`M:5:19990114T180100Z:0FED4101`	
`H:6:19990222T000320Z:133DEFDE`	El Objeto identificado como `133DEFDE fue` borrado por el sistema (probablemente para ahorrar espacio) en la fecha dada. Contador de cambios: 6 cuando ocurrió este cambio.

Para recibir este objeto, el Cliente debió solicitarlo mediante una operación Get especificada en el perfil OBEX dando como parámetro un *Ancla de Motor de Sincronización* igual ó menor a 4 de enero de 1999 a las 6:00pm, así el Servidor al recibir esta solicitud sabe que ese fue el momento de la ultima sincronización y solo devuelve un registro de cambios con los registros necesarios para que el cliente actualice la información.

4.7.2.1.2 LUID

Es un número identificador de un Objeto que significa *Locally Unique Identifier*, sirve para localizar los objetos (contactos, Notas, etc) dentro de un servidor; estos números son únicos solamente dentro del almacén de objetos en que residen. Es el segundo número que acompaña a cada cambio en el registro de cambios. (en el ejemplo anterior los números, 4,5 y 6 son LUIDs).

4.7.2.1.3 Ancla del Motor de Sincronización

Es un valor que ha guardado el motor de sincronización en la última sincronización, bien puede ser de tipo Contador de Cambios (un identificador del último cambio realizado) ó tipo marca de Tiempo; cuando se va a realizar la siguiente sincronización, el cliente usa este valor para solicitar el *Registro de Cambios* realizados y que solo le sean retornados los registros de cambios desde la última vez que se hizo sincronización; esto con el fin de hacer más rápido el proceso.

4.7.2.1.4 Objeto Calendario

Formato vCalendar1.0, definido en la especificación IrMC, define un objeto con campos por defecto como Fecha y hora de inicio y final del evento señalado, y descripción del evento. Este es un formato definido por el *Internet Mail Consortium*[12] (IMC).

4.7.2.1.5 Objeto Mensaje

Formato vMessage definido en la especificación IrMC, define un objeto con campos por defecto como Remitente, asunto y el contenido. La versión actual definida por la especificación es la 1.1. Puede servir para intercambiar entre dispositivos móviles Mensajes de correo electrónico ó SMS.

[12] "vCalendar - The Electronic Calendaring and Scheduling Format - Version 1.0", The Internet Mail Consortium (IMC), September 18, 1996, (http://www.imc.org/pdi/vcal-10.doc)

4.7.2.1.6 Objeto Tarjeta de Datos Personales

Formato vCard2.1, definido en la especificación IrMC, define un objeto con campos por defecto como Nombre, Versión de la vCard y Teléfono asociado al nombre. Este es un formato definido por el *Internet Mail Consortium*[13] (IMC).

4.7.2.1.7 Objeto Nota

Formato vNote definido en la especificación IrMC, define un objeto con campos por defecto como la Versión, la Fecha, Resumen y Cuerpo. La versión actual definida por la especificación es la 1.1. Son prácticamente mensajes propios del usuario que posee el equipo de computo; son muy comunes en PDAs.

4.7.2.1.8 Objeto Información de Dispositivo

Mantiene información sobre el dispositivo local, información como el nombre del fabricante, el modelo, Vresióin del IrMC, Número serial del dispositivo, tipos de formatos soportados por el dispositivo (de calendario, mensajes, notas y contactos), y otros opcionales como la versión de software.

4.7.2.1.9 Objeto Reloj de Tiempo Real

Objeto que contiene la Fecha y hora de acuerdo al dispositivo donde se encuentre (puesto que puede ser ligeramente diferente). Esta definido por la especificación IrMC. El objeto puede obtenerse con operaciones Get del protocolo OBEX.

4.7.3 Tipos de Sincronización

La sincronización es el núcleo de procesamiento del motor de sincronización presente en un cliente, pero que puede desencadenarse de diferentes maneras. Hay dos maneras en las que puede iniciarse este proceso, y aquí se vuelve a poner en consideración la perspectiva del usuario: Sincronización por comando y la sincronización automática.

4.7.3.1 Cliente y Servidor de Sincronización

Usualmente estos dos papeles los asume respectivamente un PC y un PDA respectivamente; en este caso es el cliente el que debe tener mayores capacidades de procesamiento, puesto que en él ocurre la comparación de los objetos de uno y otro lado, y es quien decide cuales deben ser transferidos y cuales no, para ello usa operaciones Get y Put en el servidor.

---

[13] "vCard – The Electronic Business Card Exchange Format - Version 2.1", (http://www.imc.org/pdi/vcard-21.doc) y el "IrDA Telecom Extensions to the IMC vCard Format, Version 1.0", October 15, 1997

4.7.3.2 Proceso básico de Sincronización

Se va a asumir un caso práctico. Se tiene un PC que quiere sincronizar la libreta de direcciones de un teléfono celular con una interfaz Bluetooth con sus registros de contactos actuales. Cada uno independientemente ha agregado un número telefónico previamente a su propia lista (el PC probablemente lo obtuvo por Internet y se obtuvo otro en el celular al llegar una llamada), lo cual significa que debe haber transferencia en ambos sentidos. El proceso puede resumirse de la siguiente manera:

Primero es necesario que el usuario active las aplicaciones que hagan uso de la sincronización y seleccione el dispositivo próximo que soporte esta característica y con el cual se desea sincronizar.

En caso que sea la primera vez que los dos dispositivos se comunican, ocurre también una solicitud de los códigos PIN respectivos. Si se usa también autenticación OBEX, igualmente se llevará a cabo tal proceso.
El Teléfono (el servidor) y el PC (cliente) deben estar en modo *conectable* (ver perfil de acceso genérico), deben tener activado el bit de transferencia de objetos en la palabra CoD, y el servicio registrado en la base de datos del SDP (SDDB).

Cabe anotar la existencia de un evento que desencadene la sincronización, es decir, el cliente puede iniciarla, como alternativa básica, pero en los puntos siguientes se podrá observar que la sincronización se puede llevar a cabo automáticamente ó por medio de un comando proveniente del servidor. La conexión se hace como se describió en el Perfil de intercambio genérico de Objetos, pero el campo *Destino* de la solicitud debe establecerse a "IRMC-SYNC".

Tabla 4.14 Operaciones OBEX

Operación	***Operaciones OBEX y Respuestas del Servidor***	
	Cliente	***Servidor***
El cliente obtiene el Registro de cambios del servidor, utiliza el contador de cambios obtenido en la anterior sincronización.	GET "telecom/pb/luid/5.log" *	
El servidor devuelve el registro de cambios, el cual contiene un solo cambio posterior al 5 (que fue el solicitado) el servidor no soporta marcas de tiempo y por ello no están presentes. El LUID del objeto que ha cambiado es 998.		SN:wxy2146 DID:56789 Total-Records:4 Maximum-Records:50 M:6::998
El Cliente debe comparar el numero serial del dispositivo y el identificador del almacén de	GET "telecom/pb/luid/998.vcf" (Este es un objeto vCard, el cual es el tipo	

objetos para saber si es un dispositivo ya sincronizado antes. Si esto ocurre, entonces se solicitan los objetos que hayan cambiado, en este caso solo uno.	propio de las libretas de direcciones ver definiciones mas arriba)	
El Teléfono devuelve el contenido del vCard que se ha solicitado (998.vcf). También se devuelve el LUID del objeto. Como confirmación		begin:vcard n:Gavilanes;Guido Alejandro tel;fax:(572) 4100496 version:2.1 end:vcard X-IRMC-LUID=998
Una vez hecho este primer cambio, se establece el contador local de cambios a 6. Luego se agrega el objeto que hace falta en el teléfono. Se indica que el contenido es un objeto vCard.	PUT"telecom/pb/luid/.vcf", "begin:vcard n:Acosta;Maria Victoria tel;fax:(572) 4106743 version:2.1 end:vcard"	
El servidor adiciona el objeto recibido a su lista (en el formato propio de la aplicación de manejo de contactos) y Responde con el LUID con el cual identificará al objeto. Agrega un cambio al registro de cambios y devuelve el valor del contador:7.		Operación Exitosa, LUID=567 CONTROL-CAMBIOS=7
El PC almacena el 7 como su contador de cambios local, el cual será útil en una próxima sincronización para obtener el registro de cambios.		
Termina la sincronización		

* Este nombre de objeto indica que se esta solicitando un objeto que se localiza en el "Phone Book" (pb), y cuyo LUID es 5.

4.7.3.3 Sincronización por Comando

Hay casos en que, por ejemplo, un PDA puede contener información más actualizada de contactos que un PC en algún momento dado; entonces en tal momento es necesaria una sincronización entre los dos dispositivos para que la misma información quede en ambos. El inconveniente esta en que existe la aplicación cliente (el motor de sincronización) en el PC y la servidora (más liviana en cuanto a recursos de procesamiento) en el PDA, y quien debe iniciar en este caso la sincronización debe ser el PDA; el carácter servidor del PDA no permite iniciar la transacción; pero para solucionarlo hay una forma en la cual el Servidor puede asumir un papel temporal de Cliente y envía un *Comando de Sincronización* al PC (temporalmente como Servidor); este evento desencadena la sincronización desde el PC, pero ya asumiendo de nuevo su papel de cliente.

De esta manera en el PDA puede existir (como en general ocurre) un botón ó comando denominado "Sincronizar Ahora", e iniciar la sincronización al pulsarlo, todo esto sin tener que haber tocado nada en el PC.

La sincronización por comando ocurre de la siguiente manera:

- el PDA se conecta al PC como si fuese un Cliente utilizando como parámetro de *Destino* el valor “IRMC-SYNC” y envía el comando utilizando una trama *Put* estableciendo el nombre en “telecom/push.txt”. El objeto que se envía contiene las instrucciones de lo que el cliente debe hacer. Un ejemplo podría ser:

  SYNC: telecom/pb.vcf
  SYNC: telecom/cal.vcs

  Lo cual hace que haya una sincronización tanto de la libreta de contactos (vCard) como del calendario.

- Después que el PDA recibe la respuesta exitosa del PC, debe encargarse de desconectar la sesión OBEX y puede esperar a que le llegue la solicitud del PC para sincronización, iniciándose así el proceso básico de sincronización.

La sincronización por comando debe registrarse como servicio por parte del PC, en la SDDB, al igual que el intercambio de objetos, esto puede aclararse en el punto 4.6 - Consideraciones con el SDP.

4.7.3.4 Sincronización Automática

Esta es una segunda alternativa disparar la sincronización. En este caso, el cliente la realiza de forma convencional, pero iniciándola de manera automática cada vez que el dispositivo (PDA) se encuentre cerca. Este carácter automático quiere decir también que no hay ninguna notificación para el usuario y que lo dispositivos han establecido un lazo de confianza (previamente conocen sus códigos PIN).

4.7.4 Consideraciones

4.7.4.1 Consideraciones con el SDP

Los servicios que se han descrito en este perfil son básicamente 2: El servicio de sincronización y el de comando de sincronización, ambos servicios, si existen, deben ser registrados en la SDDB para que puedan iniciarse los procesos de sincronización. Aquí

están los parámetros que se deben registrar. El cliente que reciba el comando de sincronización es quien debe registrar dicho servicio.

Tabla 4.15 Registros de servicio correspondientes al servicio de sincronización

Parámetros			***Valor***
Lista de Clases de Servicio	Clase de servicio0		IrMCSync
Lista de Descriptores de protocolos	Protocolo0		L2CAP
	Protocolo1		RFCOMM
		Parámetro0	<numero de canal servidor>
	Protocolo2		OBEX
Nombre del servicio (opcional)			IrMCSynchronization
Lista de Descriptores de Perfil (Opcional)	Perfil0		IrMCSync
		Versión0	<Número de versión soportada>
Lista de datos soportados			Phonebook, Calendar, Notes, Messages (Referirse a Bluetooth assigned Numbers [14]).

Tabla 4.16 Registros de servicio correspondientes al servicio de sincronización por comando

Parámetros			***Valor***
Lista de Clases de Servicio	Clase de servicio0		IrMCSyncCommand
Lista de Descriptores de protocolos	Protocolo0		L2CAP
	Protocolo1		RFCOMM
		Parámetro0	<numero de canal servidor>
	Protocolo2		OBEX
Nombre del servicio (opcional)			IrMCSyncCommand
Lista de Descriptores de Perfil (Opcional)	Perfil0		IrMCSync
		Versión0	<Número de versión soportada>

4.7.4.2 Consideraciones con los Demás Perfiles de Soporte

De acuerdo con el perfil de Acceso Genérico, y con el OBEX, se debe tener en cuenta que la sincronización es un proceso que demanda de máximos niveles de seguridad, puesto que en algunos casos el usuario tiene poca ó ninguna intervención en el proceso (en la sincronización automática y en el intercambio de cada objeto durante la sincronización el usuario no interviene). Por esto, debe activarse la autenticación y la encriptación en los dispositivos y una vez que ocurra la primera vez la autenticación, se puede considerar que los dispositivos han alcanzado un nivel de confianza para posteriores sincronizaciones (puesto que ya conocen sus códigos PIN).

[14] Bluetooth Special Interest Group, Bluetooth Assigned Numbers. http://www.bluetooth.org/assigned-numbers.htm

4.8 Perfil de Telefonía inalámbrica

4.8.1 Introducción

El soporte para voz o más generalmente audio es una de las características principales de Bluetooth. Este perfil define las características y procedimientos que se requieren para cumplir con la condición de interoperabilidad, entre dos unidades Bluetooth que intervienen en el escenario de aplicación “Teléfono tres en uno”, específicamente el caso en el cual, un teléfono equipado con Bluetooth trabaja como teléfono inalámbrico. Esta funcionalidad, sin embargo, no solo se destina a los teléfonos celulares, sino también hacia *handsets* (dispositivos de mano) que hacen uso de estaciones base y hacia los parlantes y micrófonos de un computador, siempre y cuando se requiera transportar de tráfico de voz.

Este perfil basa toda su operación en la capa adoptada TCS de la pila de protocolos Bluetooth. Mientras que la especificación no defina API conformes al protocolo, TCS se considerará una interfaz funcional para originar y recibir una llamada, transferir información relacionada con ella y realizar procedimientos de gestión sobre tráfico de voz. A continuación se describe brevemente cuales son sus principales características y algunas de las funcionalidades mas relevantes con el fin de lograr una mayor comprensión de la operación del perfil.

4.8.2 TCS

La especificación para control de telefonía está basada en la recomendación Q931 de la ITU. TCS reside sobre la capa L2CAP. Las aplicaciones para telefonía pueden establecer comunicación directa con TCS y usar las funciones de control de telefonía que está capa propone.

TCS define tres áreas funcionales especificas:

- Control de Telefonía: TCS se utiliza para el control de telefonía en aspectos como el establecimiento y la terminación de llamadas.

- Conectionless: También define un método para el intercambio de información de señalización sin previo establecimiento de llamadas.
- Gestión de Grupos: Cuando un grupo de dispositivos que soportan la capa TCS, los miembros del grupo denominado WUG, pueden hacer uso de funciones especiales definidas por el protocolo TCS, entre las cuales se incluyen: Gestión de membrecía en el grupo, compartición de servicios de telefonía y métodos para el establecimiento de comunicaciones directas entre dos miembros del grupo.

4.8.2.1 Operación entre dispositivos

Cuando se habla de los aspectos de señalización, TCS utiliza señalización punto a punto y también punto a multipunto. La primera de ellas se usa cuando se conoce el punto terminal (Dispositivo Bluetooth) con el que se quiere establecer una llamada. La segunda puede ser usada en caso contrario, cuando se encuentra que mas de un punto de terminal está disponible para establecer una llamada; por ejemplo cuando una estación base necesita alertar a varios teléfonos acerca de la entrada de un llamada.

La especificación indica que múltiples instancias del TCS pueden ejecutarse a la vez con el fin de dar soporte a numerosas llamadas.

La señalización punto a punto se mapea a través de canales L2CAP CO (Orientados a conexión), mientras que la señalización punto a multipunto se realiza con canales L2CAP CL (No orientados a conexión), lo cual está de acuerdo con la funcionalidad de estos canales que se emplean para transmitir paquetes Broadcast en una Piconet.

Las Figuras (4.10 y 4.11) a continuación ilustran las operaciones de señalización

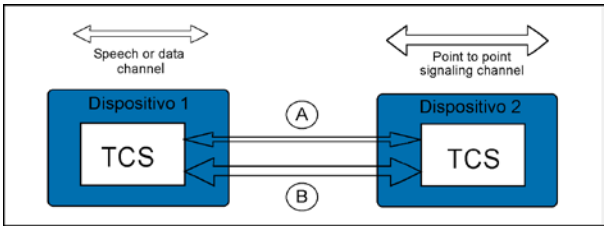


Figura 4.10 Señalización punto a punto en un configuración punto a punto

La Figura 4.10 ilustra la forma en que se realiza la señalización para establecer una llamada de voz o datos en una configuración punto a punto. En primer lugar, el dispositivo remoto es notificado acerca de una petición de llamada usando el canal A y posteriormente el canal de señalización se emplea para establecer el canal de conversación o de datos.

La Figura 4.11 por su parte ilustra el mismo proceso pero en una configuración punto a multipunto. En primer lugar, se comunica a todos los dispositivos remotos, la llegada de una petición de comunicación por medio de u canal A (señalización punto a multipunto). Después uno de los dispositivo contesta la petición por medio de un canal de señalización B (punto a punto). Y por último se emplea este canal de señalización para establecer el canal de voz a datos C.

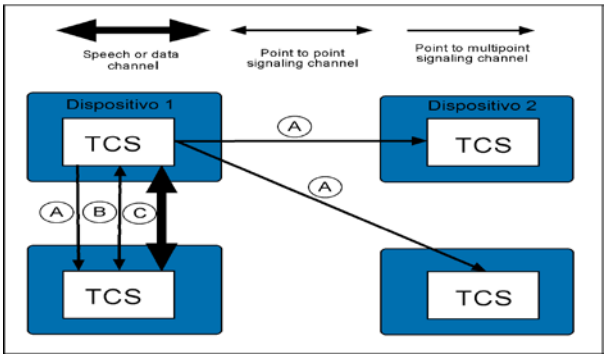


Figura 4.11 Señalización en un configuración punto a multipunto

4.8.2.2 Operación entre capas

Las implementaciones del perfil para telefonía inalámbrica que se basan TCS deben seguir la arquitectura de capas e interfaces que se describe a continuación.

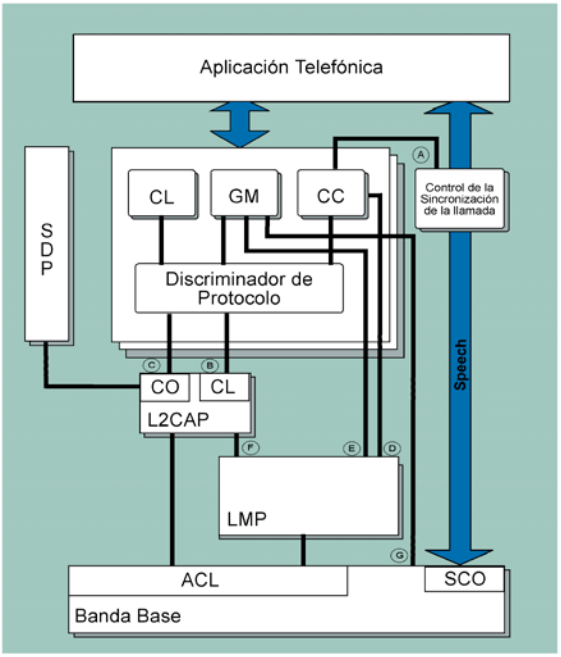


Figura 4.12 Modelo de protocolo para TCS

- La entidad de Control de Llamada (a través de la interfaz A) proporciona información de control de sincronización de la llamada cuándo se conectan (o desconectan) los caminos de conversación. Esta información se basa en el intercambio de los mensajes de control de llamada *Connect Acknowledge y Disconnect*.
- La capa L2CAP se emplea para enviar un mensaje de *Setup* usando señalización punto al multipunto, a través de un canal no orientado a conexión. L2CAP emplea la interfaz B para informar a la capa TCS de la recepción de un mensaje *Setup* recibido en un canal no orientado a conexión.
- La capa TCS emplea la interfaz C para entregar mensajes a la capa L2CAP cuando se requiere que sean enviados a través de canales orientados a conexión.
- La entidad de control de llamada emplea la interfaz D para controlar el establecimiento de canales SCO por parte del *Link Manager*.
- La entidad de Gestión de Grupos controla directamente los procedimientos de *inquiry, paging y paring* a través de las interfaces E y G.

- El discriminador de Protocolos es una entidad TCS que se encarga de enrutar el tráfico hacia la entidad funcional correspondiente.

4.8.3 Descripción del Perfil de Telefonía Inalámbrica

4.8.3.1 Configuración Y Roles

El perfil de telefonía inalámbrica describe establece dos tipos de papeles para los dispositivos que lo implementen:

Gateway: La Gateway actúa como un punto terminal de la red externa a la que un dispositivo terminal desea conectarse. La Gateway se considera entonces como el punto central con respecto al origen y recepción de llamadas; esto significa que deberá soportar todas las peticiones para el establecimiento de llamadas hacia la red externa y desde esta al dispositivo terminal. Ejemplos de una Gateway incluyen estaciones base PSTN domésticas, Gateway GSM, Gateway satelitales y Gateway H.323.

Terminal TL: Es el dispositivo que hace las veces de terminal de usuario inalámbrico que desea conectarse hacer uso de los servicios que a Gateway ofrece. Ejemplos de terminales son un teléfono inalámbrico un PC o un teléfono celular o inalámbrico que opera en modo dual.

4.8.3.2 Tipos de llamadas

- *Llamada Externa*: Es un llamada que se establece entre un terminal y otro dispositivo por medio de una red externa a la cual están conectados los dispositivos que participan el la llamada.

- *Llamada de intercomunicación*: Es una llamada que se establece entre dos terminales que pertenecen a un mismo grupo inalámbrico. Referirse al perfil de intercomunicación.

4.8.3.3 Aplicación

La Figura 4.13 a continuación ilustra una piconet para telefonía inalámbrica.

Figura 4.13 Piconet para telefonía inalámbrica

En este caso los dispositivos pueden conectarse a un punto de acceso el cual se comunica con una red de telefonía externa. El maestro es la piconet en este caso es la Gateway y es responsable del manejo de hasta siete esclavos activos y 255 en modo park. Los dispositivos a su alrededor pueden adherirse a la piconet, realizando el procedimiento de *Page* a la estación base. Una vez la estación base acepta la conexión del dispositivo remoto, este se convierte por defecto en el maestro de la piconet.

Sin embargo este modelo de roles en la red es contrario al que se requiere para la operación normal de la piconet. De esta manera se hace necesario un cambio de papeles entre los dispositivos de la piconet.

Podría contemplarse otra alternativa de operación, en donde no haya cambio de rol y en donde sea la Gateway quien realiza el procedimiento de *Page* hacia un dispositivo terminal, pero existen dos razones de peso para no hacerlo de esta forma:

1. Cuando los dispositivos terminales desean tener un acceso temporal a la piconet, es preferible que sea el cliente quien inicie la petición para coactarse a la piconet.

2. Con un modelo en que la Gateway es maestro o servidor en la piconet y los dispositivos terminales son clientes o esclavos, se está teniendo en cuenta la e deseo y necesidad que tiene el usuario de conectarse a la red, en lugar de que sea la Gateway quien esté buscando dispositivos para que se conecten a ella.

Planteados las condiciones de conexión de esta forma, el proceso de enlace a una piconet de telefonía inalámbrica, se inicia con la solicitud de un cliente (Dispositivo terminal) a la Gateway e inmediatamente esta última acepte se lleva a cabo un cambio de papeles en la piconet, de al forma que la Gateway asume el papel de maestro de la piconet.

Una vez que la piconet para telefonía inalámbrica se ha formado, se establece un WUG (*Wieless User Group*); este es el nombre que recibe un grupo de dispositivos que soportan TCS. El WUG se emplea para facilitar el uso de las funcionalidades y características de telefonía inalámbrica en forma segura.

Cuando un dispositivo terminal se conecta a la Gateway, este establece y mantiene una conexión L2CAP durante su permanencia en la piconet. El establecimiento de llamadas se realiza entonces, sobre conexiones L2CAP ya establecidas. Tanto canales L2CAP orientados a conexión como no orientados a conexión se establecen para efectos de señalización TCS durante una sesión de este tipo. La voz se transmitirá a través de canales SCO.

4.8.3.4 Escenarios de aplicación

El perfil de Telefonía cubre los siguientes escenarios de aplicación:

- Conexión a la Gateway de manera que las llamadas entrantes puedan ser enrutadas hacia los terminales y las llamadas salientes puedan originarse.
- Efectuar y recibir llamadas hacia y desde un usuario conectado a la red externa, a la cual está conectada la Gateway.
- Hacer uso de servicios suplementarios que proporcionan redes externas.

Como se mencionó anteriormente, la Gateway asumirá el papel de maestro de la piconet; como maestro también controlará los modos de operación de los esclavos y también podrá enviarles información de Broadcast cuando sea necesario. La Gateway deberá dedicar tanta

capacidad como sea posible para poder escanear por los *Pages* que periódicamente se originan desde los dispositivos terminales.

4.8.3.5 Establecimiento de las llamadas. Procedimientos TCS

4.8.3.5.1 Control de Llamada

Las funciones para control de llamada sirven para establecer llamadas que transportan tráfico de voz o datos. TCS actúa como una maquina de estado que ejecuta las operaciones necesarias para progresar de un estado al próximo hasta alcanzar un estado resultado.

Los estados de llamada utilizados por TCS, están definidos por la recomendación Q931 de la ITU, pero dadas las características de los dispositivos que hacen uso de Bluetooth en cuanto a capacidad de procesamiento memoria, solo un conjunto de estados se consideran obligatorios para las implementaciones basadas en TCS.

La tabla a continuación muestra el conjunto de estados que se requirieren en un entorno Bluetooth:

Tabla 4.17 Estados del establecimiento de llamada empleados por Bluetooth

ESTADOS GENERALES	ESTADOS PARA EL LADO SALIENTE	ESTADOS PARA EL LADO ENTRANTE
Null Active Disconnect Request Disconnect Indicatión Release Request	Call Initiated	Call Present Connect Request

Los participantes de una llamada reciben nombres de acuerdo con el papel que asuman en ella así:

En una llamada externa saliente el terminal TL es el lado saliente y la Gateway será el lado entrante. En una llamada externa entrante el terminal TL es el lado entrante y por su parte la Gateway GT será el lado saliente.

4.8.3.5.1.1 *Petición de llamada* (Call Request)

El lado saliente inicia el establecimiento de una llamada a través del envío de un mensaje *Setup*, e inicia un temporizador de 20 segundos llamado T303.
Si se trata de una configuración punto a punto el mensaje se entrega a través de un canal orientado a conexión. En caso contrario (configuración multipunto) el mensaje se entrega por medio de un canal no orientado a conexión, de este modo el mensaje se transmite como Broadcast en los instantes faro.

Si no hay respuesta por parte del lado entrante antes de que el temporizador T303 expire, el lado entrante debe:

- Si el mensaje *Setup* fue enviado en un canal CO, se retorna al estado *Nulo*.
- Si el mensaje se entrego en un canal cl, el lado saliente envía un mensaje *Release Complete* que contiene el motivo número *2 Recovery On Time Expiry*.

El mensaje *Setup* debe contener la clase de llamada (Llamada externa o entre dispositivos), también debe contener toda la información requerida por el lado entrante para que este pueda procesar la llamada. El número de dígitos en el elemento de información *Número de la Parte Llamada*, puede estar incompleto, de manera que si es necesario conocerlo, podrá emplearse el mensaje *Overlaping Sending*. Seguido a la transmisión del *Setup*, el lado saliente debe en el estado Llamada iniciada. Después de recibir el mensaje *Setup*, el lado entrante debe entrar en el estado Llamada Presente.

4.8.3.5.1.2 *Selección de Canal Portador* (Bearer Selection)

El mensaje *Setup* que se envía durante una petición de llamada, puede contener un elemento de información denominado *Capacidad de Portador* (Bearer Capability) a través de cual se indica que recursos de capas inferiores se emplearan en durante un llamada.

Un elemento de información *Bearer Capability* con el valor “Synchronous Connection – Oriented (SCO)” indica que se usará un enlace SCO. En este caso, el elemento de información contiene además el tipo de método que será empleado para codificación de audio.

Por otra parte, si un elemento de información *Bearer Capability* presenta el valor "Asynchronous Connectionless (ACL)" se entiende que la llamada empleará un enlace ACL. Antes de definir un elemento de información *Bearer Capability* con el valor ACL, debe haberse establecido un canal L2CAP con los requerimiento de calidad indicados para habilitar llamadas de datos.

4.8.3.5.1.3 Overlaping Sending

Si el mensaje *Setup* no contiene un elemento de información del lado que envía el mensaje, o el numero de la parte llamada está incompleto, el lado entrante iniciara un temporizador T302 (15 segundos), enviará un mensaje *Setup Acknowledge* al lado saliente y entrará en el estado de *Overlaping Receiving*.

Cuando se recibe este último mensaje, el estado saliente debe enviar el resto de la información del elemento de información, mediante el uso de mensajes de *Information*.

El lado saliente debe reiniciar un temporizador T304 (30 segundos), cada vez que envíe un mensaje de información. Por su parte el lado entrante debe iniciar un temporizador T302 cada vez que recibe un mensaje de información y este no contiene la información deseada.

Si el temporizador T304 expira, el lado saliente debe iniciar el procedimiento de *Call Clearing* con el motivo número 28 *Invalid Número Format*.

Si es el temporizador T302 el que expira, el lado entrante debe:

- Si se determina que la información no esta aun completa, iniciar el procedimiento *de Call Clearing* con el motivo número 28 *Invalid Número Format*.
- En otro caso, responder al lado saliente con un *Call Proceeding*, *Alerting* o con un mensaje de conexión.

4.8.3.5.1.4 Procedimiento de Llamada (Call proceeding)

- Call Proceding Enbloq Sending

Si se usa un Sending Enbloc (El lado entrante determina que ha recibido suficiente información en el *Setup*) el lado entrante debe enviar un mensaje de *Call Proceeding* al lado saliente para confirmar el mensaje de *Setup* e indicarle que la llamada está siendo procesada y posteriormente entrar en el estado *Incoming Call Proceeding*. Después de recibir un mensaje de *Call Proceeding*, el lado saliente debe entrar en el estado *Outgoing Call Proceeding*, detener el temporizador T303 (20 segundos), e iniciar el temporizador T310 (30-20 segundos),.

- Call Proceding Overlaping Sending.

Si se presenta alguna de estas condiciones:

1. El lado entrante recibe un *Sendig Complete Indication ó;*
2. El lado entrante determina que la información necesaria para establecer la llamada está completa

El lado entrante debe enviar un mensaje de *Call proceeding*, detener el temporizador T302 y si es aplicable, iniciar el temporizador T310. Después de recibir un mensaje de *Call Proceeding*, el lado saliente debe entrar en el estado *Outgoing Call Proceeding*, detener el temporizador T304 e iniciar, si es aplicable, el temporizador T310.

Si se el temporizador T310 expira:
El lado saliente debe iniciar un procedimiento *de Call Clearing*, con el motivo número 102 *Recovery On Timer Expiry*.

4.8.3.5.1.5 *Confirmación de Llamadas* (Call confirmation)

Después de recibir una indicación que alerte al usuario de que la llamada ha sido iniciada el lado entrante debe enviar un mensaje de alerta y entrar en el estado *Call Received*. Cuando el lado saliente recibe un mensaje de alerta, debe entrar en el estado *Call Delivery*, detener

el temporizador T303 o T310 e iniciar un temporizador T301(mínimo 3 minutos). Cuando el temporizador T301 expire, el lado saliente debe iniciar un procedimiento de *Call Clearing* con motivo número 102 *Recovery On Timer Expiry*.

4.8.3.5.1.6 *Conexión de Llamada* (Call Conexión)

El lado entrante indica que acepta una llamada entrante, mediante el envío de un mensaje *Connect* al lado saliente. Después de enviar este mensaje debe detener las alertas al usuario e iniciar el temporizador T313 (4 segundos).

Después de que el lado saliente recibe un mensaje *Connect*, debe:

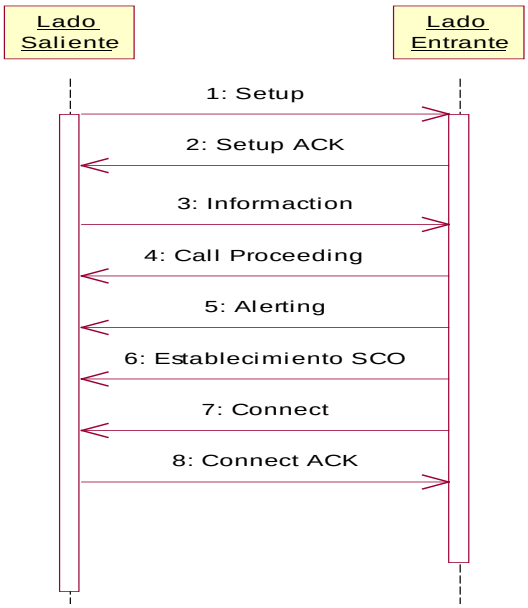
- Detener cualquier tipo de alerta interna
- Detener los temporizadores T301, T303, T304 y T310
- Completar la petición del lado saliente para soporte de canal
- Enviar un mensaje de *Connect Acknowledge* al lado entrante
- Entrar en estado *Activo*

El mensaje de *Connect Acknowledge* indica que la petición para soporte de canal ha sido completada.

Después de recibir este mensaje el lado entrante se conecta al canal de soporte, detiene el temporizador T 313 y entra en estado Activo.

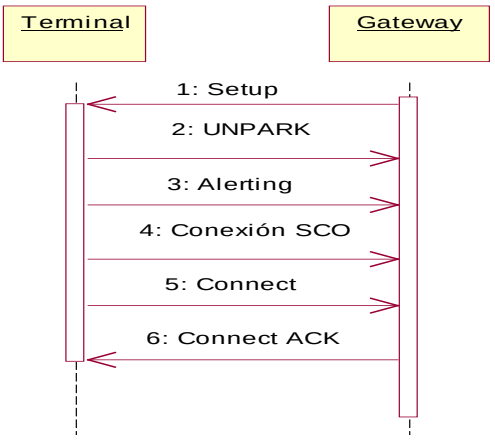
La Carta de Secuencia 4.1 ilustra el proceso para establecimiento de llamadas salientes.

Ver página siguiente...



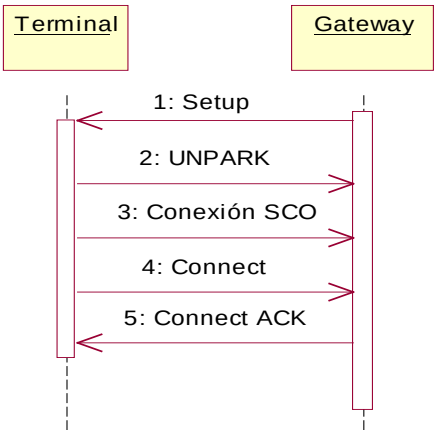
Carta de Secuencia 4.2 Establecimiento de llamada saliente

La siguiente carta de secuencia ilustra el proceso para establecimiento de llamadas, cuando se trata de una llamada entrante. Para este caso el mensaje *Setup* se entrega haciendo uso de canales orientados a conexión.



Carta de Secuencia 4.3 Establecimiento de llamada entrante 1

La Carta de Secuencia 4.4 ilustra el proceso para establecimiento de llamadas, cuando se trata de una llamada entrante. Para este caso el mensaje *Setup* se entrega haciendo uso de canales no orientados a conexión.



Carta de Secuencia 4.4 Establecimiento de llamada entrante 2

4.8.3.5.1.7 Información de Llamada (Call Informaction)

Las dos partes de una conexión pueden intercambiar información relacionada con las llamadas salientes, usando mensajes de Información.

4.8.3.5.1.8 Non Selected user clearing

Cuando una llamada ha sido entregada en un canal CL, además de enviar un mensaje *Connect Acknowledge*, el lado saliente debe enviar un mensaje *Release* a todos los demás lados entrantes que han enviado mensajes *Conncet Acknowledge, Call Proceedng, Alerting* o *Connect* en respuesta a un mensaje *Setup*.

Este mensaje de *Release*, se emplea para notificar a los lados entrantes que la llamada no se les ofrece mas.

4.8.3.5.1.9 Falla en el establecimiento de la Conexión (Failure of the call establishment)

Las causas por las cuales puede presentarse falla en el establecimiento de un llamada son:

Tabla 4.18 Posibles fallas en el establecimiento de la conexión

NÚMERO	VALOR	SIGNIFICADO
1	Unassigned (unallocated) number	Número no Signado
3	No route to destination	No hay ruta de destino
17	User busy	Usuario Ocupado
18	No user responding	El usuario no responde
19	No answer from user	No hay respuesta de Usuario
21	Call rejected by user	Llamada rechazada por el usuario
22	Number changed	El numero se ha modificado
28	Invalid number format (incomplete number)	Formato de número inválido
34	No circuit/channel available	No hay canal o circuito disposnible
44	Requested circuit/channel not available	Canal solicitado no disponible
58	Bearer capability not presently available	No hay soporte para Canal portador
65	Bearer capability not implemented	No está implementado el soporte para canal portador

4.8.3.5.2 Remoción de llamadas

*4.8.3.5.2.1 Remoción Normal de Llamada (*Normal Call Clearing*)*

Bajo condiciones normales, el procedimiento para remoción(interrupción cancelación) de un llamada se inicia cuando una de las dos partes envía un mensaje de desconexión a la otra. Los procedimientos de remoción de llamadas son simétricos y pueden ser iniciados por el lado entrante o saliente de una comunicación.

Los siguientes términos, son empleados por este perfil en la definición de los procedimientos de remoción de llamadas.

- Un canal está conectado cuando hace parte de una conexión establecida de acuerdo a la especificación.

- Un canal está desconectado cuando ya no hace parte de una conexión pero todavía no se encuentra disponible para que sea empleado en una nueva conexión.

- Un canal se ha liberado cuando ya no hace parte de una conexión y está disponible para que sea empleado en una nueva conexión.

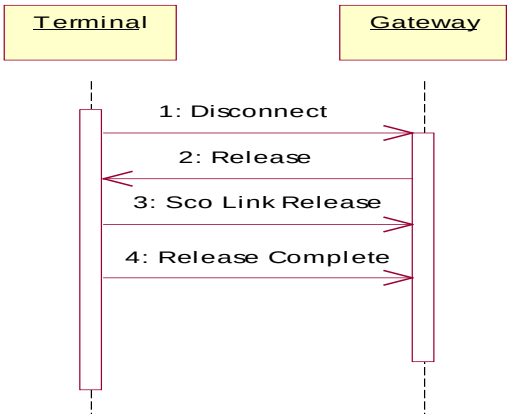
Si es el lado saliente quien desea iniciar la remoción de la llamada, debe enviar un mensaje de Desconexión *Disconnect*, iniciar un temporizador T305 (30 segundos), liberar el canal portador y entrar en el estado *Release Request*.

Después de recibir el mensaje de desconexión, el lado entrante entra al estado de Indicación de desconexión y se desconecta del canal portador. Una vez este último ha sido desconectado, el lado entrante debe enviar un mensaje *Release* al lado saliente, iniciar un temporizador T308 (4 segundos) y entrar al estado *Release Request*. Cuando el lado saliente recibe el mensaje *Release*, cancela el temporizador T305, libera el canal portador y envía un mensaje *Release Complete* al lado entrante y retorna a su estado inicial, estado *Nulo*.

Si el lado saliente no recibe el mensaje de *Release* complete antes de que expire el temporizador T305, debe eviar un mensaje *Release* al lado entrante con el número contenido originalmente en el mensaje *Disconnect*, iniciar un temporizador T308 y entrar en el estado *Release Request*.

La *Carta de Secuencia 4.5* ilustra el proceso para establecimiento de llamadas según se describe en la sección 3.3.3.3.2, cuando se trata de un llamada entrante. En esta ilustración, es el dispositivo terminal quien solicita la desconexión.

Ver página siguiente...



Carta de Secuencia 4.5 Remoción normal de llamada

*4.8.3.5.2.2 Remoción Anormal de Llamada (*Abnormal Call Clearing*)*

El procedimiento de remoción de llamadas puede ser causado de manera anormal, si se presentan los siguientes casos:

El lado Entrante rechaza el mensaje *Setup*, respondiendo con un mensaje *Release Complete* (Sin que previamente exista petición para desconexión) y entrando en el modo *Nulo*.

En el caso de una conexión punto a multipunto, el procedimiento de remoción de llamada será iniciado por el lado saliente para un usuario no seleccionado.

En el caso de una conexión punto a multipunto, cuando se ha entregado un mensaje *Setup* en un canal CL, si se recibe un indicación de desconexión por parte de un usuario durante el establecimiento de una llamada, cualquier lado entrante que haya respondido debe ser removido con un mensaje *Release*. El usuario debe realizar los procedimientos de remoción de llamada descritos anteriormente. El lado saliente debe entrar en el estado *Nulo*.

Si en el momento del establecimiento de llamada se crea un enlace SCO, la unidad podrá liberar este tipo de enlace invocando los procedimientos respectivos definidos por el *Link Manager*.

4.8.3.5.3 Gestión de grupos

La gestión de grupos se emplea en aplicaciones de telefonía , para habilitar la provisión de un conjunto de funciones que involucran no solo a uno sino muchos usuarios, por ejemplo extensiones del teléfono múltiples, reenvío de llamada, llamadas de grupos, entre otras.

Antes de dar inicio a cualquiera de los procedimientos de gestión de grupos es necesario establecer un canal L2CAP orientado a conexión.

4.8.3.5.3.1 *Grupo de usuarios inalámbricos WUG*

Las funciones para gestión de grupos emplean el concepto de WUG. Un WUG se compone de un número de unidades Bluetooth que soportan TCS. Uno de ,os dispositivos asume el papel de maestro de la WUG, normalmente este papel corresponde al maestro de la *Piconet* y en el caso de un Piconet para telefonía inalámbrica, el papel es asumido por la Gateway.

Las características principales de un WUG son:

Todas las unidades que hacen parte de un WUG, saben cual es el maestro de la WUG y conocen a todos las unidades que son miembros de la WUG.
Cuando una unidad nueva se ha unido a la WUG y ha realizado todo el procedimiento de *pairing* con la unidad maestro, puede comunicarse y realizar operaciones de autenticación y encriptación con cualquier otra unidad del grupo sin necesidad de realizar el procedimiento de *pairing* con las demás unidades.

4.8.3.5.3.2 Procedimientos para gestión de grupos

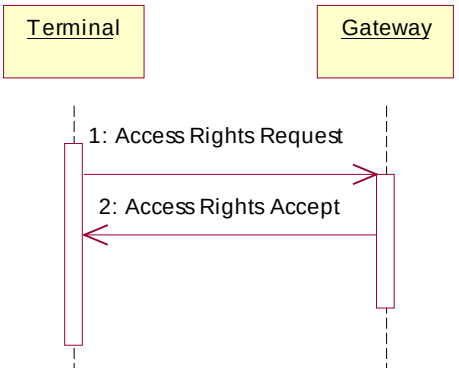
Las entidades para gestión de Grupos dan soporte a los siguientes procedimientos:

- *Obtención de Derechos de Acceso*

A través de los procedimientos para obtención de derechos de acceso, un dispositivo puede realizar peticiones para usar los servicios de telefonía que ofrece otro dispositivo miembro de la WUG. Por ejemplo un dispositivo de mano podría hacer uso de los derechos de acceso para transferir llamadas desde un dispositivo a otro.

Un dispositivo solicita derechos de acceso por medio del envío de un mensaje *Acces Rights Request* e iniciando un temporizador T401(8 segundos). El dispositivo que recibe el mensaje podrá aceptar la solicitud a través del mensaje *Acces Rights Accept*.

Si el dispositivo que envía el mensaje no recibe respuesta, debe asumir que su petición ha sido negada. La Carta de Secuencia 4.6 ilustra el proceso:



Carta de Secuencia 4.6 Procedimiento para obtención de derechos de acceso

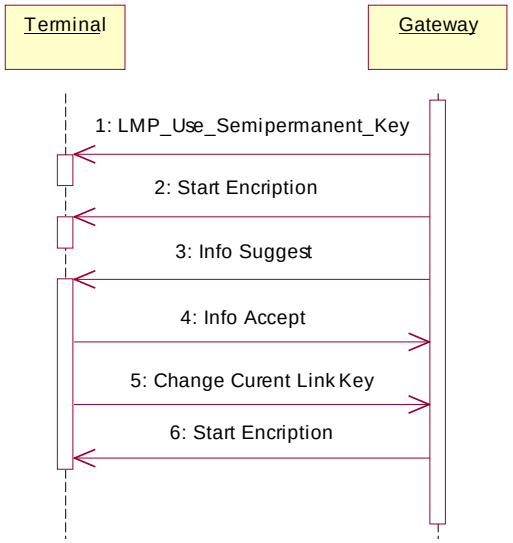
- *Distribución de la Configuración*

La distribución de la configuración es un procedimiento TSC por medio del cual se gestiona la membrecía de un dispositivo en un WUG. Todas la unidades que pertenecen a un WUG deben ser informadas acerca de los cambio que se presentan en el, por ejemplo en caso de que una unidad entre o salga del grupo.

El procedimiento para distribución de la información se inicia por el maestro del WUG a través del envío del mensaje *Info Suggest* y la iniciación del temporizador T403 (4 segundos).

Una vez recibido este mensaje, cada uno de los miembros de la red debe responderlo devolviendo al maestro un mensaje *Info Accept*.

La *Carta de Secuencia 4.7* ilustra el proceso para Distribución de la Configuración



Carta de Secuencia 4.7 Procedimiento para distribución de la configuración

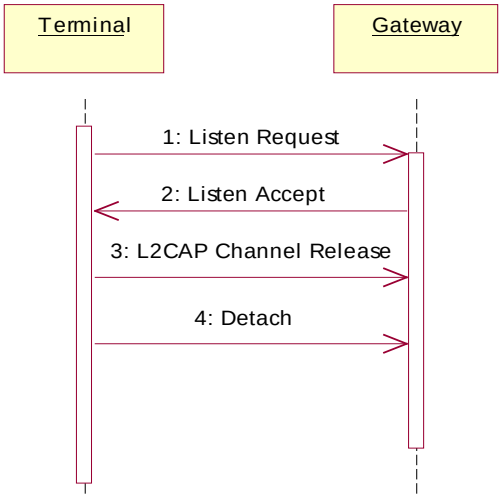
- *Acceso Inter-miembros.*

Cuando dos miembros de un WUG presentan un estado activo frente al maestro de la red, estas dos unidades pueden establecer rápidamente una conexión entre ellas.

Con la facilidad del acceso intermiembros, un miembro de la WUG puede emplear también las funciones de distribución de la información para determinar las características de un dispositivo próximo. Esta información se reenvía hacia el maestro de la WUG, quien se encarga de contactar al dispositivo con el que se desea establecer comunicación. Este dispositivo a su vez, responderá al maestro enviando su información de reloj y entrando en el modo de *PAGE SCAN*. El maestro transmitirá la información recibida, al primer dispositivo quien hace uso de ella para comunicarse directamente con e otro dispositivo.

Como resultado se obtienen una nueva Piconet dentro de la WUG, la cual consiste inicialmente de dos dispositivos.

La Carta de Secuencia 4.8 ilustra el proceso para Acceso Inter-miembros



Carta de Secuencia 4.8 Procedimiento para acceso inter-miembros

4.8.3.5.4 Servicios Suplementarios

Se denominan servicios suplementarios a aquellos servicios que hacen parte de un WUG o son proporcionados por una red externa conectada a la Gateway de la Piconet para telefonía inalámbrica.

TCS proporciona soporte para un servicio suplementario denominado *Calling Line Identity*. Si existe posibilidad de acceder a servicios suplementarios en redes externas se emplean secuencias DTMF. Con ellas puede activarse - desactivarse e interrogarse un servicio suplementario deseado.

4.8.3.5.4.1 Identificador de Llamada (Calling Line Identity)

Con el fin de informar al lado entrante acerca de la identidad de quien origina la llamada, el lado saliente incluye un elemento de información que contiene el número identificador de la parte que hace la llamada en la petición para establecimiento de llamadas, *Setup*.

Para el caso especifico de este perfil, es recomendable que la Gateway que esta conectada a una red externa que provea de este servicio, este en capacidad de entregar esta información al usuario.

4.8.3.5.4.2 Secuencias DMTF

La secuencias DTMF que se emplean a nivel de este perfil son *DMTF Star* y *DMTF Stop*, las cuales sirven para ejercer control sobre redes de tipo RTPC.

En principio los mensajes DMTF pueden ser iniciados por cualquiera de las partes de una llamada, sin embargo en la práctica, la Gateway que esté conectada a la PSTN es el receptor de los mensajes.

4.8.4 Protocolos y Perfiles de Bluetooth empleados por el Perfil de Telefonía Inalámbrica

4.8.4.1 Protocolo de descubrimiento de Servicios SDP

Como fue visto en el capitulo anterior, cada servicio que desee ofrecerse en una Piconet debe ser debidamente registrado en la base de datos del SDP, con el fin de que otros dispositivos puedan encontrarlo y hacer uso de el.

A continuación se definen las entradas hacia la base de datos (Registro del servicio) para el caso de un dispositivo que presta el servicio de Gateway dentro de una Piconet de telefonía inalámbrica:

Tabla 4.19 Registro de servicio para la gateway que se implementa en el perfil TCS

ITEM	SUB ITEM	VALOR
Lista de Identificadores de Clases de Servicio (Service Class ID List)	Clase de Servicio #0	Telefonía Inalámbrica
	C Clase de Servicio #0	Telefonía Genérica
Lista de Descriptores de Protocolo (Protocol Descriptor List)	Protocolo #0	L2CAP
	Protocolo #1	TCS
Nombre del Servicio (Service Name)		Definido por el proveedor de Servicios
Red Externa		0x01 = PSTN 0x02 = ISDM 0x03 = GSM 0x04 = CDMA

		0x05 = CELULAR ANALOGA 0x06 = CONMUTACIÓN DE PAQUETES 0x07 = OTRA
Lista de Descriptores de Perfil (Profile Descriptor List)	Perfil #0	Telefonía Inalámbrica
	Parámetros de Perfil #0	0x01000

4.8.4.2 Protocolo de Adaptación y Control de Lógico de Enlace L2CAP

En cuanto a la capa L2CAP los requerimientos exigidos por este perfil se relacionan con los tipos de canales empleados y las opciones de configuración del protocolo.

4.8.4.2.1 Tipos de canales

En este perfil se emplean canales L2CAP orientados a conexión y no orientados a conexión. Estos últimos se emplean para enviar información de Broadcast desde la Gateway a varios miembros de la piconet. Según la especificación de este perfil, solamente dispositivos terminales pueden establecer canales orientados a conexión. El valor empleado en el campo PSM de un paquete *Connection_Request* (petición de conexión) debe ser 0X0007, el cual corresponde al valor *TCS-BIN-CORDLESS*.

4.8.4.2.2 Opciones de configuración

- MTU

  El valor mínimo de MTU que una implementación L2CAP debe usar para soportar este perfil es 171 octetos. Esto significa que el número máximo de dispositivos terminales soportados por este perfil es 7.

- Flush Time Out

  El valor de tiempo de descargue tanto para la Gateway como para los dispositivos terminales debe ser el valor por defecto 0XFFFF.

- Calidad del Servicio QoS

  La negociación de la calidad del servicio es opcional.

4.8.4.3 Protocolo para Gestión de Enlace LMP

El perfil de telefonía inalámbrica tiene en cuenta las siguientes funcionalidades del protocolo *Link Manager*

- Cambio de Papel

  Al inicio del capitulo se explicaron las razones por las cuales este perfil debe soportar los procedimiento para cambio de papel. La Gateway debe realizar la petición para cambio de rol cada vez que un dispositivo terminal se conecta a ella. Si la petición es rechazada, la Gateway debe desconectarse inmediatamente del dispositivo; pues de lo contrario no puede garantizarse que la correcta prestación de los servicios que proporciona la Gateway.

- Política de Enlace

  La Gateway debe ser tan conservadora como sea posible al decidir en qué modo de consumo de energía sitúa un dispositivo terminal TL. Esto significa que cuando un TL no se encaja con el nivel de potencia deseado la Gateway podrá ponerlo en un modo de consumo de energía diferente. Los parámetros para el modo de consumo de energía bajo se escogerán de tal manera que el dispositivo terminal TL pueda regresar al estado activo en un periodo de 300 ms. Si es la Gateway quien puede ahorrar energía durante una llamada, puede hacer uso del modo *Sniff*.

- Tamaño de la clave de Encriptación:

  Con el fin de soportar en envió de paquetes Broadcast , todos los dispositivos que implementen este perfil deben estar en capacidad de soportar una clave de encriptación de cinco octetos.

4.8.4.4 Perfil de acceso genérico

Además de los procedimientos básicos respecto a descubrimiento de dispositivos y descubrimiento de servicios, el perfil de telefonía inalámbrica tiene en cuenta los aspectos de

modo de descubrimiento de los dispositivos y los modos de seguridad definidos por el perfil de Acceso Genérico.

- Modos de Descubrimiento

Tabla 4.20 Modos de descubrimiento del perfil GAP para el perfil de TCS

	PROCEDIMIENTO	SOPORTE TL	SOPORTE GW
Modos de Descubrimiento	Modo de No-Descubrimiento	No Aplica(N/A)	O
	Modo de Descubrimiento Limitado	N/A	Opcional (OP)
	Modo General de Descubrimiento	N/A	O
Modos de Conectividad	Modo de No - Conectividad	N/A	Condicional (C)
	Modo de Conectividad	N/A	O
Modo de Pairing	Modo de No-Pairing	Obligatorio (O)	O
	Modo pairing	OP	O

- Aspectos de Seguridad

Tabla 4.21 Aspectos de seguridad del perfil GAP para el perfil de TCS

PROCEDIMIENTO	SOPORTE EN EL TL	SOPORTE EN LA GW
Autenticación	O	O
Modo de Seguridad 1	Condicional (C)	N/A
Modo de Seguridad 2	C	C
Modo de Seguridad 3	C	C

4.9 Perfil de Intercomunicación Intercom

4.9.1 Introducción

Este perfil define los requerimientos necesarios para que los dispositivos Bluetooth soporten la funcionalidad de intercomunicación, la cual corresponde a la tercera forma de aplicación del escenario de uso “Teléfono tres en uno”, comúnmente llamada *Walkie Talkie.*

Podrá recordarse del capitulo anterior, que los perfiles de telefonía inalámbrica e intercomunicación soportan cada uno de ellos dos partes especificas del escenario de aplicación teléfono “Tres en uno”. Esta asignación se ha hecho basándose en los requerimientos técnicos y funcionalidades de cada una de las divisiones del escenario de aplicación.

Mientras que la parte correspondiente a telefonía inalámbrica involucra funciones avanzadas de TCS, la parte de intercomunicación o *“Intercom”* puede considerarse mas simple en comparación y por lo tanto si se han establecido requerimientos mínimos de procesamiento para los dispositivos empleados en este caso, es recomendable emplear aquellas funciones que representen la menor carga permisible.

La separación de funcionalidades también es muy útil cuando el caso de uso no se desea establecer por completo; de esta manera y según la necesidad específica se implementan solo aquellas funciones de carácter obligatorio en cada caso.

El perfil de Intercomunicación es realmente un caso de telefonía inalámbrica. Se recordará que el perfil de telefonía inalámbrica emplea el concepto de WUG con el fin de facilitar el uso de funciones especiales de telefonía inalámbrica. Una de estas funciones es a de comunicar dos dispositivos sin la necesidad de un tercero que sirva de puente entre los dos primeros. Es claro que en este caso, lo que se presenta es un llamada de intercomunicación y es así como dos dispositivos se conectan con el propósito de actuar como W*alkie Talkie*.

4.9.2 Operación entre Capas

Las implementaciones del perfil para telefonía inalámbrica que se basan TCS deben seguir la arquitectura de capas e interfaces que se describe a continuación.

Interfaz A: La entidad de Control de Llamada usa esta interfaz para control de sincronización cuando se conectan y desconectan los caminos de “conversación” al interior de TCS.

Interfaz B: Esta interfaz se utiliza para entregar mensajes TCS en la conexión orientó (el punto para apuntar) el canal de L2CAP.

Interfaz C: Son empleadas por la entidad de Control de Llamada para comunicarse directamente con el *Link Manager* con el propósito de establecer y liberar los enlaces SCO

Para efectos de inicialización (*Inquiry*, *Paging*) se requiere tener control adicional sobre la Banda Base y el Controlador de Enlace (*Link Control*).

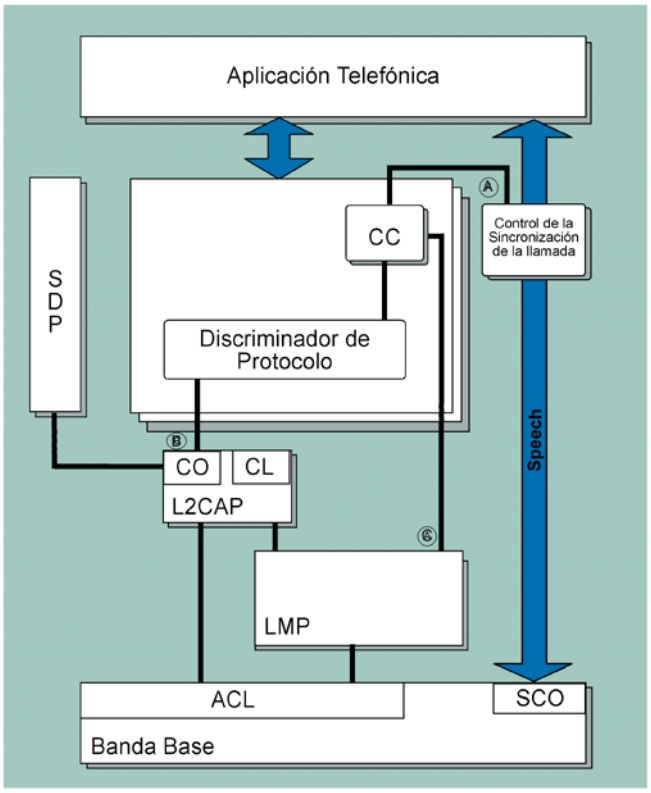


Figura 4.14 Arquitectura de capas e interfaces para el perfil de Intercomunicación

4.9.3 Escenarios de Aplicación

Los escenarios típicos cubiertos por este perfil son aquellos en donde se requiere establecer una conversación directa comunicación (teléfonos-computador-*handset*) basándose en señalización telefónica.

Por ejemplo dos personas emplea sus celulares Bluetooth para iniciar directamente una conversación teléfono a teléfono sin hacer uso de un proveedor de telefonía o de una estación base.

Si este caso se analiza sin detenimiento y se tienen en cuenta que la operación estándar entre dos dispositivos Bluetooth se limita a u rango de cobertura de diez metros a la redonda podrá concluirse que las dos personas que usan la el perfil de intercomunicación se encuentran demasiado cerca como para poder hablar directamente sin hacer uso de los "radios". Desde este enfoque el escenario para intercomunicación en el estándar de operación de Bluetooth (0 dBm a 10 metros) deja de ser interesante, aunque existen situaciones en donde la intercomunicación entre dispositivos puede ser muy útil, aun cuando la proximidad entre las unidades supera por poco los diez metros de distancia. Considérese por ejemplo el caso en que las personas se encuentran en diferentes pisos de un edificación e incluso en el mismo sitio pero desean comunicarse haciendo uso de sus unidades , lo cual podría suceder en un auditorio enorme en el momento en que los asistentes hacen preguntas al ponente acerca de su presentación.

Sin embargo, cuando situaciones similares se aplican para el escenario de 100 metros de cobertura y 4 dBm, el panoramas de aplicación cambia sustancialmente. La habilidad de establecer llamadas si hacer uso de un proveedor de servicios de telefonía , lo que incurre en gastos adicionales para el usuario, es muy atractiva. Suponga el uso de su teléfono para contactar a otra persona que se encuentra en el campus universitario mientras usted deja su vehículo en el parqueadero de la institución. El caso podría extenderse a las personas que trabajan como vigilantes o están encargadas de la seguridad de un hotel o un evento que se realizará en un área de gran extensión. En estos casos, la comunicación inalámbrica de "medio alcance" puede ser de gran efectividad, además de que ahorra la necesidad de implantar soluciones de radio diferentes, las cuales en su mayoría agregan costos y dispositivos adicionales al usuario.

4.9.4 Papel de los Dispositivos para el Perfil de Intercomunicación

El perfil de intercomunicación no establece papales predefinidos para los dispositivos que hacen uso de este. Contrario al Perfil para telefonía inalámbrica, en donde deben definirse los roles de Gateway y Terminal, el perfil de intercomunicación es completamente simétrico y todos los dispositivos que soportan el perfil se denotan como *Terminales TL.*

A continuación se plantea la descripción de las interacciones que tienen lugar, cuando un terminal desea establecer una llamada de intercomunicación con un dispositivo remoto.

4.9.5 Establecimiento de la Conexión

Si el iniciador de la llamada no conoce la dirección Bluetooth del dispositivo remoto, debe obtenerla por medio de los procedimientos de *Acceso Genérico* relacionados con el descubrimiento de servicios.

El perfil no exige el establecimiento de algún nivel de seguridad. Sin embargo si estos son requeridos, deben ser ejecutados los procedimiento de autenticación definidos por el perfil de Acceso Genérico con el propósito de establecer un conexión segura. De igual forma se tratan los aspectos relacionados con encriptación de datos.

Una se llevan a cabo los procedimientos anteriores de acuerdo con los requerimientos de usuario, se puede dar inicio al establecimiento de la llamada.

4.9.5.1 Procedimientos para control de llamada

*4.9.5.1.1 Petición de llamada (*Call Request*)*

Antes de que se establezca una llamada, es necesario establecer un canal L2CAP orientado a conexión CO, según los requisitos que se establecen en la sección 3.4.6.2.1 de capítulo 3

Los procedimientos relacionados con la solicitud del establecimiento de llamada, deben realizarse de la misma forma que se describen en el perfil de telefonía inalámbrica sección 4.8.3.5.1.1

*4.9.5.1.2 Soporte de Canal Portador (*Bearer Capability*)*

El siguiente cuadro presenta el contenido del elemento de información para una llamada de intercomunicación.

Tabla 4.22 Contenido del elemento de información bearer capability

CAMPO	VALORES
Tipo de enlace (Link Type)	SCO
Información de Usuario	CVSD

El propósito del elemento de información *Capability Bearer* es indicar si una petición o servicio del portador está disponible. Si se habla de una llamada de Intercomunicación el valor por defecto del elemento *Bearer Capability,* corresponde a un tipo de canal SCO con paquetes HV3 mientras que el método empleado para codificación de audio es Continuous Variable Slope Delta CVSD.

*4.9.5.1.3 Confirmación de Llamada (*Call Confirmation*)*

Los procedimientos relacionados con esta funcionalidad deben realizarse de la misma forma que se describen en el perfil de telefonía inalámbrica sección 4.8.3.5.1.5

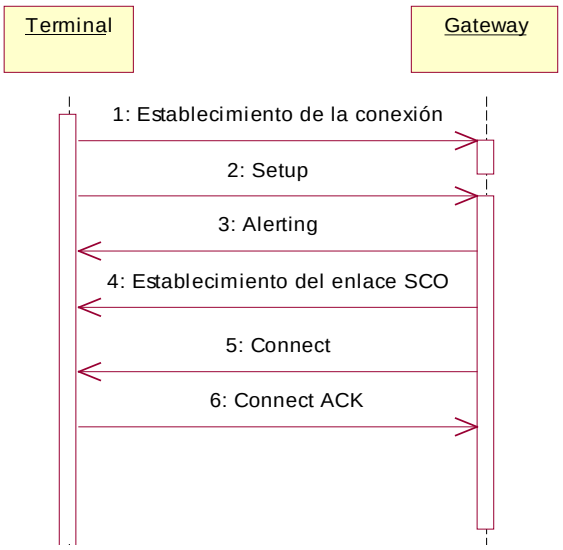
*4.9.5.1.4 Conexión de la Llamada (*Call Connetion*)*

Los procedimientos relacionados con esta funcionalidad deben realizarse de la misma forma que se describen en el perfil de telefonía inalámbrica sección 4.8.3.5.1.6

El perfil de intercomunicación emplea paquetes SCO para el transporte de información, los cuales son norma cuando se habla de tráfico de voz. El establecimiento de paquetes SCO debe realizarse antes del envío e un mensaje CONNECT según los procedimientos establecidos en la sección 3.3.3.4.2 del capítulo 3.

La Carta de Secuencia 4.9 ilustra el proceso de establecimiento de una Llamada de Intercomunicación

Ver página siguiente...



Carta de Secuencia 4.9 Establecimiento de una llamada de intercomunicación

*4.9.5.1.5 Falla en el establecimiento de la conexión (*Failure of call establishment*)*

Los procedimientos relacionados con esta funcionalidad deben realizarse de la misma forma que se describen en el perfil de telefonía inalámbrica sección 4.8.3.5.1.9

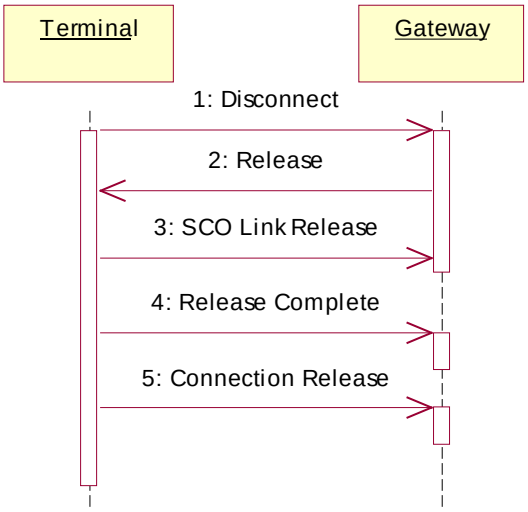
4.9.5.2 Remoción de Llamada (Call Clearing)

Los procedimientos relacionados con esta funcionalidad deben realizarse de la misma forma que se describen en el perfil de telefonía inalámbrica sección 4.8.3.5.2

Adicionalmente el perfil plantea las siguientes observaciones para el momento en que se reciba el último mensaje para remoción de llamada.

La liberación de un enlace SCO debe realizarse según los procedimientos establecidos por el *Link Manager* en la sección 3.3.3.4.2 del capítulo 3.

La Carta de Secuencia 4.10 ilustra el proceso de Remoción de Llamada de Intercomunicación



Carta de Secuencia 4.10 Remoción de llamada de intercomunicación

4.9.5.3 Perdida del enlace

Si una unidad se encuentra en el estado activo y detecta que se ha perdido el enlace con el dispositivo remoto, debe entrar al estado *Nulo*.

Los procedimientos para remoción de llamadas, no se ejecutan en este caso.

4.9.6 Protocolos y Perfiles de Bluetooth empleados por el Perfil de Intercom

4.9.6.1 Protocolo de descubrimiento de Servicios SDP

A continuación se definen las entradas hacia la base de datos (Registro del servicio) para el caso de un dispositivo que desea ofrecer el servicio de llamadas de Intercomunicación.

Tabla 4.23 Registro del servicio para el perfil de Intercomunicación

ITEM	SUBITEM	VALOR
Lista de Identificadores de Clases de Servicio (Service Class ID List)	Clase de Servicio #0	Intercomunicación
	Clase de Servicio #0	Telefonía Genérica
Lista de Descriptores de Protocolo (Protocol Descriptor List)	Protocolo #0	L2CAP
	Protocolo #1	TCS
Nombre del Servicio (Service Name)		Definido por el proveedor de Servicios. Puede emplearse "Intercomunicación"
Lista de Descriptores de Perfil (Profile Descriptor List)	Perfil #0	Intercomunicación
	Parámetros de Perfil #0	Versión del Perfil 0x01000

4.9.6.2 Protocolo de Adaptación y Control de Lógico de Enlace L2CAP

En cuanto a la capa L2CAP los requerimientos exigidos por este perfil se relacionan con los tipos de canales empleados y las opciones de configuración del protocolo.

4.9.6.2.1 Tipos de canales

En este perfil se emplean solamente canales L2CAP orientados a conexión. El valor empleado en el campo PSM de un paquete *Connection_Request* (petición de conexión) debe ser 0X0005, el cual corresponde al valor *TCS-BIN-CORDLESS*.

4.9.6.2.2 Opciones de configuración

- MTU

  El valor mínimo de MTU que una implementación L2CAP debe usar para soportar este perfil es 3 octetos.

- Flush Time Out

  El valor de tiempo de descargue tanto para la Gateway como para los dispositivos terminales debe ser el valor por defecto 0XFFFF.

- Calidad del Servicio QoS

  La negociación de la calidad del servicio es opcional.

4.9.6.3 Protocolo Para Gestión De Enlace LMP

Los procedimientos relacionados en la siguiente tabla son de obligatorio cumplimiento para el *Link Manager* si se implementa el perfil de Intercomunicación.

Tabla 4.24 Procedimientos LMP para el perfil de Intercomunicación

PROCEDIMIENTO	SOPORTE EN LMP
Autenticación	Obligatorio (O)
Establecimiento de la Conexión	O
Pairing	O
Cambio de la clave de enlace	O
Cambio de la clave de enlace actual	O
Petición por Offset del Reloj Maestro	O
Petición de Nombre	O
Detach	O
QoS	O
Supervisión del Enlace	O

4.9.6.4 Perfil de Acceso Genérico

La siguientes tablas relacionan la funcionalidad de Modos de Descubrimiento del perfil de Acceso genérico para los dispositivos que participan el Perfil de Intercomunicación

4.9.6.4.1 Modos de Descubrimiento

La *Tabla 4.25* relaciona los modos de descubrimiento asociados a este perfil.

Tabla 4.25 Modos de descubrimiento aplicables al perfil de Intercomunicación

	PROCEDIMIENTO	SOPORTE GW
Modos de Descubrimiento	Modo de No-Descubrimiento	Obligatorio (O)
	Modo de Descubrimiento Limitado	Opcional (OP)
	Modo General de Descubrimiento	O

Modos de Conectividad	Modo de No - Conectividad	No Aplica (N/A)
	Modo de Conectividad	O
Modo de Pairing	Modo de No-Pairing	O
	Modo pairing	OP

4.10 *Perfil para "Manos Libres"* Headset

4.10.1 Introducción

El perfil de *Headset* define los protocolos y procedimientos que deben ser implementados por un dispositivo que emplea un módulo denominado "Headset" o "*manos libres*". El Headset pude actuar como el mecanismo de entrada y salida de audio en un dispositivo Bluetooth.

El motivo inicial para la concepción de Bluetooth, fue la necesidad de establecer una interfaz inalámbrica para sustitución de cable entre dos dispositivos y uno de los primeros escenarios que cubre este propósito el de Headset Bluetooth.

El Headset es un dispositivo que establece comunicación inalámbrica con un teléfono sin requerir ningún tipo de conexión cableada. Un usuario que recibe un llamada, podría tener acceso a la porción de audio de la señal, a través del micrófono y el parlante de su Headset Bluetooth, sin necesidad de sacar el teléfono de su maleta de negocios y dejando sus manos libres para realizar otra tarea mientras habla.
Existen dos ventajas fundamentales que derivan del uso del Headset Bluetooth:

- Soporte para movilidad: El usuario del Headset no está ligado al dispositivo que genera el audio y por lo tanto pueda deambularen un área de gran extensión sin interrumpir su comunicación.
- Multi Uso: Ya que Bluetooth proporciona una interfaz de acceso inalámbrica estándar, el mismo Headset puede ser empleado por múltiples dispositivos. Un Headset puede utilizarse fácilmente para transportar tráfico de audio entre un computador y su operario.

Se espera que en un futuro próximo sea posible utilizar Headsets Bluetooth en estéreos, reproductores de CD y MP3, grabadoras de sonido, entre otros.

4.10.2 Soporte Protocolar para el Perfil Headset

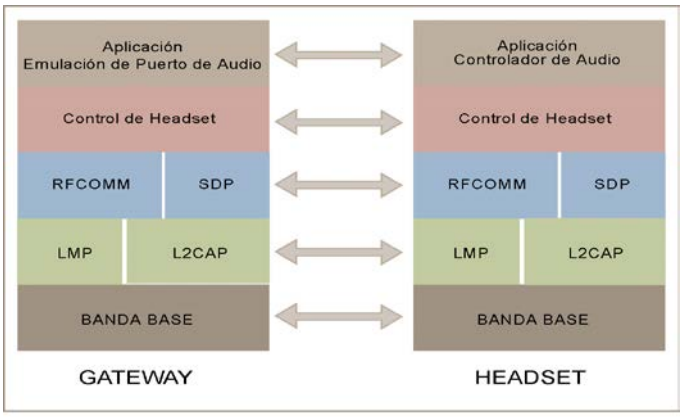


Figura 4.15 Protocolos Bluetooth y entidades de soporte para el perfil Headset

La Figura 4.15 muestra los protocolos Bluetooth y entidades que utiliza este perfil. Además de las funciones de control de telefonía que proporciona el protocolo TCS, Bluetooth introduce una segunda forma de control denominada *Control de Telefonía por comandos AT*. Aunque esta forma de control no se define como una capa adicional del stack, varios perfiles hacen uso de ella para realizar procedimientos de control, en comunicaciones que involucran tráfico de audio; este es el caso del perfil de Headset. Los comandos AT se emplean básicamente para ejercer control sobre un módem y se usan especialmente para que ciertas aplicaciones puedan comunicarse con un módem a través de un puerto serial. En Bluetooth, la implementación de comandos AT se basa en el uso del protocolo RFCOMM para emulación serial.

Entre los requerimientos que se le exigen a los dispositivos Headset, está el hecho de que puedan fabricarse a muy bajo costo, sean livianos y muy simples. Si a estos dispositivos se le agregan capacidades que requieran software sofisticado (Lo cual a su vez implica mayor capacidad de procesamiento, memoria y consumo de energía) es muy complicado cumplir con los requisitos de simplicidad mencionados; por esta razón el perfil de Headset hace uso de las funciones de control de telefonía que prestan los comandos AT y las capacidades del perfil serial, sin recurrir a la capa TCS lo que incurriría en una mayor complejidad para el dispositivo.

4.10.3 Papeles y Configuración de los Dispositivos que participan en el Perfil de Headset

El perfil no exige que dispositivo actué como maestro y el otro como esclavo; cualquiera de los dos puede asumir cualquiera de los dos papeles. Sin embargo, para efectos de definición del perfil es importante definir cuál es el dispositivo donde se genera el audio, el cual recibirá el nombre de *Gateway de Audio* (Teléfonos y computadores personales) y por otra parte cual es el dispositivo que hace las veces de Headset.

Un Headset puede acceder a los servicios ofrecidos por la Gateway sin el establecimiento de un conexión segura. Es el usuario quien debe determinar si desea tener conexiones seguras; para esto deberá hacer uso de los procedimientos de autenticación definidos en el *Perfil de Acceso Genérico*.

El establecimiento del enlace debe iniciarse cundo se recibe o se genera un llamada. Normalmente se hace necesario llevar a cabo el proceso de *Paging* entre los dos dispositivos. Para este perfil, se asume que el escenario de uso se presenta solamente entre dos dispositivos. Y entre una Gateway y un Headset, solo puede establecerse una conexión de audio al tiempo. No está incluido el soporte para múltiples llamadas. El trafico de audio se presenta en ambas direcciones, la Gateway controlará el establecimiento y desconexión de enlaces SCO y el Headset se encargará de conectar directamente enlaces SCO con el flujo de audio interno.

La Gateway y el Headset están en capacidad de emular puertos seriales a través de RFCOMM. La emulación serial se emplea, como se dijo anteriormente, para transportar comandos AT entre los dos dispositivos.

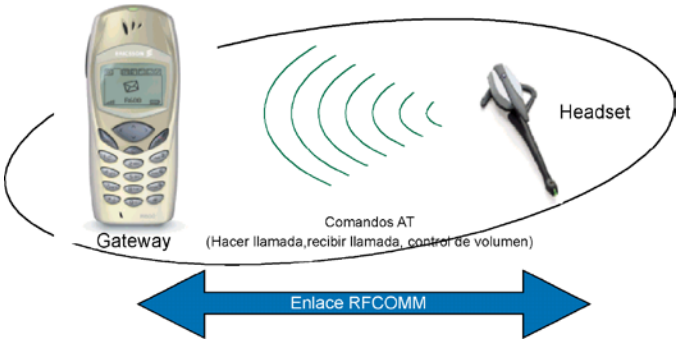


Figura 4.16 Configuración básica del perfil de Headset

4.10.4 Operaciones del Perfil de Headset

Esta sección describe los requerimientos y procedimientos relacionados con el establecimiento de llamadas y operaciones adicionales del perfil de Headset, haciendo uso de unas pocas funciones de control facilitadas por los comandos AT.

4.10.4.1 Procedimientos de Conexión

El establecimiento de una conexión puede iniciarse por cualquiera de los dos dispositivos que participan en la conexión.

4.10.4.1.1 Establecimiento de la conexión por parte de la Gateway de Audio

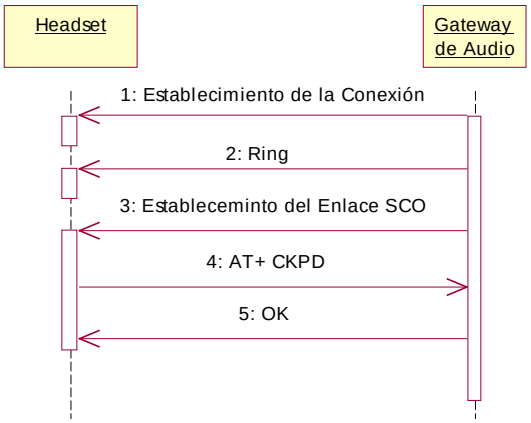
Después de que la Gateway recibe una indicación interna o un evento generado por el usuario, inicia el establecimiento de la conexión, la cual esta destinada a su vez al establecimiento de una conexión ACL.

Si no se ha establecido una sesión RFCOMM entre la Gateway y el Headset, debe llevarse a cabo esta operación por parte de l dispositivo que inicia la conexión. El establecimiento de la conexión debe realizarse según los procedimientos definidos por el *Perfil de Acceso Genérico* y el *perfil de Aplicación para Descubrimiento de Servicios*.

Una vez se ha establecido la conexión la Gateway envía el primero de los comandos AT empleados por este perfil; un RING, con el propósito de alertar al usuario de la llamada que se ha generado. El Ring puede repetirse durante todo el tiempo que tome establecer la conexión.

El establecimiento de enlaces SCO puede realizarse una vez se haya establecido la conexión ACL. En los casos donde se alerta al usuario de a presencia de una llamada, el usuario debe dar una respuesta con la que confirme que acepta la conexión de audio.

Para tal efecto el usuario generará un tipo de evento (Presionar un botón en el Headset, por ejemplo) ante el cual el Headset envía un comando AT denominado AT+CKPD (*Ver* Tabla *4.26 Lista de comandos AT que dan soporte al perfil de Headset*). La Gateway en respuesta establece la conexión SCO, si esta aun no se ha llevado a cabo.



Carta de Secuencia 4.11 Establecimiento de la conexión por parte de la gateway

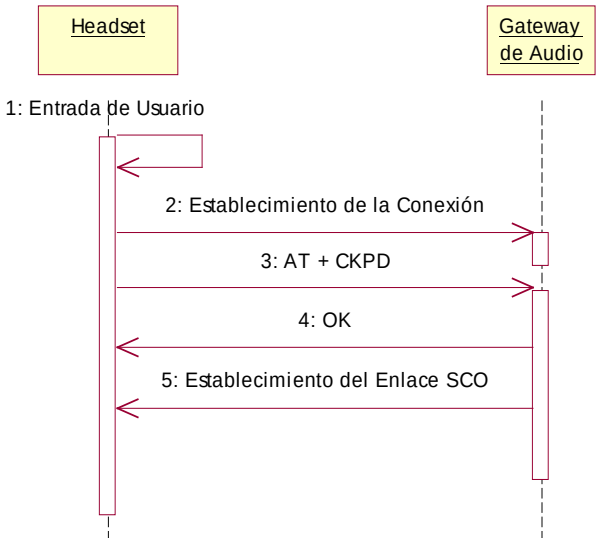
4.10.4.1.2 Establecimiento de la conexión por parte del Headset

El establecimiento de la conexión ACL por parte del Headset se presenta, cuando se recibe una indicación por parte del usuario como por ejemplo cuando este presiona un botón en el dispositivo. Después de establecer el enlace AC, el Headset debe enviar un comando AT+CKPD hacia la Gateway.

La Gateway a su vez recibe el comando AT+CKPD e inicia el establecimiento de la conexión SCO. Pueden requerirse procedimientos adicionales en la Gateway para establecer y/o enrutar trafico de audio hacia el Headset.

La ilustra el proceso para establecimiento de una conexión de audio, solicitada por el Headset.

Ver página siguiente...



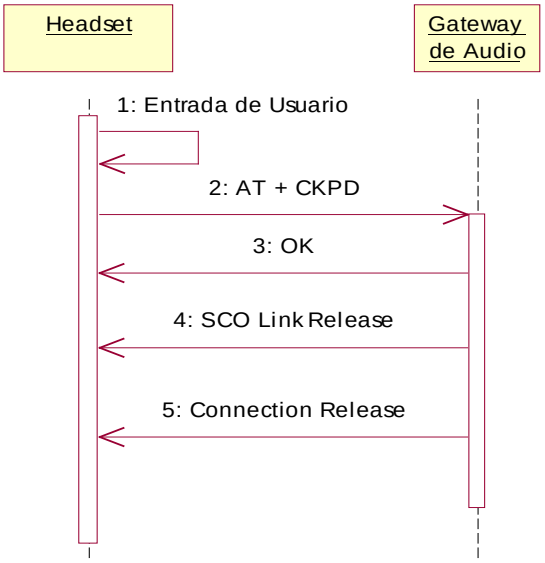
Carta de Secuencia 4.12 Establecimiento de la conexión por parte del headset

4.10.4.2 Liberación de la conexión de Audio

Una llamada puede ser terminada por la Gateway o por el Headset. Los procedimientos relacionados con esta operación, se ilustran a continuación.

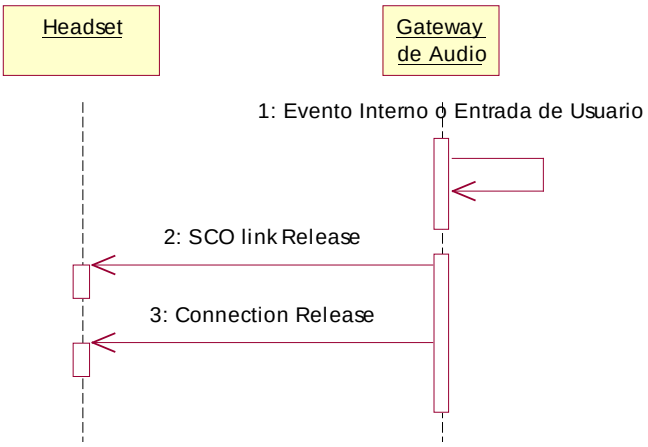
- Remoción de llamada por parte del Headset

Ver página siguiente...



Carta de Secuencia 4.13 Liberación de la conexión de audio por parte del headset

- Remoción de Llamada por parte de la Gateway.



Carta de Secuencia 4.14 Liberación de la conexión de audio por parte de la gateway

4.10.4.3 Control Remoto de Volumen

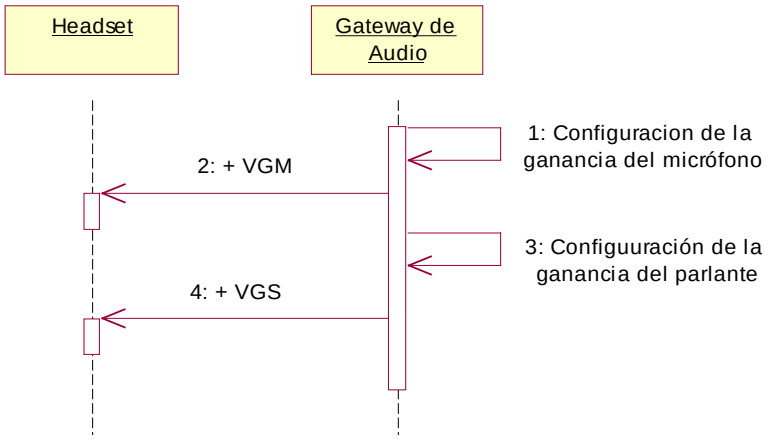
La Gateway está en capacidad de controlar la ganancia del micrófono y el parlante del Headset mediante los comandos +VGM y +VGS respectivamente.

No existe límite para la cantidad de veces que estos comando pueden transmitirse hacia el Headset, siempre y cuando la conexión este activa. A pesar de esto, no es obligatorio que una aplicación que implementa el perfil soporte la opción de control de volumen para los dos casos; micrófono y parlante.

Las ganancias tanto del micrófono y el parlante se establecen en una escala de 0 a 15. Estos valores, son valores absolutos y relacionados con un nivel de volumen que depende de la aplicación particular y que está bajo el control del Headset.

Un Headset puede almacenar la configuración de los valores +VGM y +VGS de su micrófono y parlante cuando se libera la conexión, con el fin de restablecerlos cuando se establezca nuevamente una llamada. Si estos valores se han almacenado, el Headset debe informárselo a la Gateway en el momento en que se presenta una nueva conexión.

Cuando se emplean formas mecánicas como botones *dialers* etc. para controlar el volumen en un Headset , el Headset de be usar comandos AT +VGM y +VGS, para informar a la Gateway de cualquier cambio en el nivel de volumen del dispositivo.



Carta de Secuencia 4.15 Control remoto de volumen

4.10.5 Comandos AT empleados por el Perfil Headset

Tabla 4.26 Lista de comandos AT que dan soporte al perfil de Headset

COMANDO	SINTAXIS	DESCRIPCIÓN	VALOR
Indicación de Llamada Entrante	RING	Este comando es emitido por el DCE (Gateway) para reportar al DTE (Headset) la presencia de una llamada entrante. Este comando debe repetirse cada vez que la red envié una indicación de llamada entrante.	-
Presión de botones en el Headset	+CKPD	Comando emitido por el Headset, para comunicar que se ha presionado un botón	200
Ganancia del Micrófono	+VGM=<gain>	Comando emitido por la Gateway para establecer la ganancia del micrófono del Headset. <gain> es una constante decimal relacionada con un nivel de volumen controlado por el Headset.	<gain>: 0 - 15
Ganancia del Parlante	+VGS=<gain>	Comando emitido por la Gateway para establecer la ganancia del parlante del Headset. <gain> es una constante decimal relacionada con un nivel de volumen controlado por el Headset.	<gain>: 0 – 15
Reporte del nivel de ganancia del micrófono	+VGM=<gain>	Comando emitido por el headset para reportar a la gateway la ganancia actual de su micrófono. <gain> es una constante decimal relacionada con un nivel de volumen controlado por el Headset.	<gain>: 0 – 15
Reporte del nivel de ganancia del parlante	+VGS=<gain>	Comando emitido por el headset para reportar a la gateway la ganancia actual de su parlante. <gain> es una constante decimal relacionada con un nivel de volumen controlado por el Headset.	<gain>: 0 – 15

4.10.6 Protocolos y Perfiles de Bluetooth empleados por el Perfil de Headset

4.10.6.1 Protocolo de descubrimiento de Servicios SDP

A continuación se definen las entradas hacia la base de datos (Registro del servicio) para el Perfil de Headset.

Tabla 4.27 Registro del servicio para el perfil de Headset (Dispositivo: Headset).

ITEM	SUBITEM	VALOR
Lista de Identificadores de Clases de Servicio (Service Class ID List)	Clase de Servicio #0	Headset
	Clase de Servicio #0	Audio Genérico
Lista de Descriptores de Protocolo (Protocol Descriptor List)	Protocolo #0	L2CAP
	Protocolo #1	RFCOMM
Nombre del Servicio (Service Name)		Definido por el proveedor de Servicios. Puede emplearse "Headset"
Lista de Descriptores de Perfil (Profile Descriptor List)	Perfil #0	Headset
	Parámetros de Perfil #0	Versión del Perfil 0x0100
Control remoto de Volumen		Si/No

Tabla 4.28 Registro del servicio para el perfil de Headset (Dispositivo: Gateway).

ITEM	SUBITEM	VALOR
Lista de Identificadores de Clases de Servicio (Service Class ID List)	Clase de Servicio #0	Gateway de Audio Headset
	Clase de Servicio #0	Audio Genérico
Lista de Descriptores de Protocolo (Protocol Descriptor List)	Protocolo #0	L2CAP
	Protocolo #1	RFCOMM
Nombre del Servicio (Service Name)		Definido por el proveedor de Servicios. Puede emplearse "Gateway de Voz"
Lista de Descriptores de Perfil (Profile Descriptor List)	Perfil #0	Headset
	Parámetros de Perfil #0	Versión del Perfil 0x0100

4.10.6.2 Protocolo para Gestión de Enlace LMP

En adición a los requerimientos exigidos por el perfil serial para la capa *Link Manager*, el perfil Heaset exige el soporte de enlaces SCO tanto para la Gateway como para el Headset.

Tabla 4.29 Procedimientos LMP para el perfil Headset

PROCEDIMIENTO	SOPORTE
Autenticación	Obligatorio (O)
Pairing	O
Cambio de la clave de enlace	O
Cambio de la clave de enlace actual	O
Petición por Offset del Reloj Maestro	O
Versión del LMP	O
Petición de Nombre	O
Detach	O
QoS	O
Enlaces SCO	O
Supervisión del Enlace	O
Establecimiento de la Conexión	O

4.10.6.3 Perfil de Puerto Serial

El perfil Headset emplea las funcionalidades que describe el perfil de puerto serial relacionadas con la operación de las capas RFCOMM y L2CAP.

4.10.6.3.1 RFCOMM

Estas funcionalidades son principalmente aquellas que se refieren a señales de control RS232 , indicación de estado remoto y negociación de puerto remoto.
Referirse a la sección 4.3.2.1.y a la sección 4.3.2.2.

4.10.6.3.2 L2CAP

A continuación se listan las funcionalidades de L2CAP, que emplea el perfil serial como soporte al perfil de Headset.

Tabla 4.30 Procedimientos L2CAP empleados por el perfil Headset

	PROCEDIMIENTO	SOPORTE PARA LAS DOS UNIDADES
Tipos de canales	Canales orientados a conexión	Obligatorio (O)
	Canales orientados a conexión	No Aplica (N/A)
Señalización	Establecimiento de la conexión	O
	Configuración	O
	Finalización de la conexión	O
	Eco	O
	Rechazo de comandos	O
Configuración de Parámetros	Unidad de transferencia máxima	O
	Flush Time Out	O
	QoS	Opcional (OP)

4.10.6.4 Perfil de Acceso Genérico

La siguiente tabla muestra cuales son los modos de operación que debe soportar el perfil Headset según lo estable el perfil de *Acceso Genérico.*

Tabla 4.31 Modos de operación del perfil de Acceso Genérico empleados por el Perfil Headset

ITEM	SUB ITEM	SOPORTE EN EL HS	SOPORTE EN LA GW
Modos de Descubrimiento	Modo de No-Descubrimiento	Obligatorio (O)	No Aplica (N/A)
	Modo de Descubrimiento Limitado	Opcional (OP)	N/A
	Modo General de Descubrimiento	O	N/A
Modos de Conectividad	Modo de No-Conectividad	N/A	N/A
	Modo de Conectividad	O	O
Modo de Pairing	Modo de No-Pairing	OP	OP
	Modo Pairing	OP	OP

4.11 *Perfil de Marcación Telefónica* Dial Up

4.11.1 Introducción

El perfil de Dial Up se define con el propósito de que un computador pueda iniciar o recibir una llamada de datos conectándose a una red, a través de marcación telefónica. El perfil de Dial Up es uno de los dos casos que plantea el escenario de uso de Bluetooth llamado “Puente a Internet“.

Existen dos métodos para utilizar Bluetooth como puente inalámbrico hacia Internet o otras redes externas o Intranets corporativas. El primer método es el de Intertrabajo o *Networking* por medio de acceso telefónico empleando un teléfono como módem inalámbrico.

Hoy en día, la marcación telefónica para acceso a redes es uno de las formas convencionales para entrar a Internet. Un usuario usa el módem de su computador junto con el software para marcar a redes para conectarse a un proveedor del servicios, el cual a su vez proporciona la entrada a Internet o a una red de otro tipo. Sin embargo, este método de acceso exige la presencia de cables entre el computador y el teléfono (o la línea telefónica).

Para este caso Bluetooth propone una solución inalámbrica con el fin de reemplazar cables entre el computador y el teléfono o entre un teléfono y la línea telefónica (asumiendo en este ultimo caso que se trata de un teléfono celular) y hacer que una tarea habitual se realice de modo mas simple y sea aplicable en muchos otros ambientes. En estas condiciones el teléfono se comporta como un módem para tráfico de datos.

Visto desde otro punto, el perfil de Dial Up cubre un área en donde las comunicaciones y la computación se traslapan ya que puede verse como dispositivos telefónicos acceden a redes de telefonía con el fin de que otros dispositivos, que son de computo, puedan conectarse a redes de datos.

Otro de los inconvenientes propios de la marcación a redes alambrada y que Bluetooth resuelve, es la necesidad de contar con conectores adecuados entre un computador (el módem) y el toma de la línea telefónica. Muchas personas se ven en situaciones incómodas, cuando salen de sus casas o sus sitios habituales de trabajo y no encuentran un toma para teléfono compatible con el conector del módem de su computador personal y no pueden por

lo tanto acceder a Internet u otra red de interés. Con Bluetooth no es necesario preocuparse por esta coyuntura, ya que la naturaleza inalámbrica de la comunicación evita el uso de cualquier tipo de conector adicional.

4.11.2 Configuración y Papel de los Dispositivos que intervienen en el perfil de Dial Up

El perfil de Dial Up define los protocolos y procedimientos que deben implementarse para poner en práctica el escenario de uso de Bluetooth denominado Puente a Internet. Ejemplos de las situaciones que cubre este perfil son:

- El uso de un teléfono celular o módem como módems inalámbricos por parte de un computador para conectarse a un servidor de acceso a Internet, a través de marcación telefónica.
- El uso de un teléfono celular o módem por parte de un computador para recibir llamadas de datos.

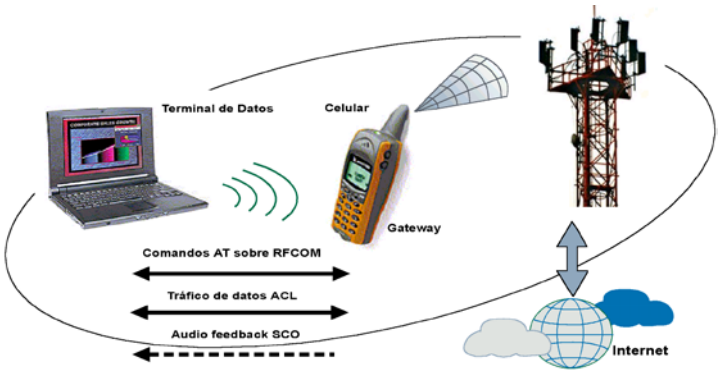


Figura 4.17 Disposición de los elementos que conforman el perfil de Dial Up

La especificación describe dos tipos de papeles para los dispositivos que participan en este perfil:

Gateway: Es el dispositivo que proporciona acceso a la red pública. Los dispositivos mas comunes que hacen el papel de Gateway son los módems y los teléfonos celulares.

Terminal de Datos: Es el servicio que utiliza el servicio de marcación telefónica ofrecido por la Gateway. Los dispositivos terminales están representados por computadores portátiles o PC.

4.11.3 Pila de Protocolos Bluetooth empleada por el Perfil de Dial UP

El perfil de Dial Up se deriva del perfil de Serial y emplea la capa RFCOMM del protocolo Bluetooth para transportar información de usuario y comandos AT entre el terminal de datos y la Gateway. La capa de emulación de módem reside en el dispositivo que hace las veces de Gateway, mientras que el software para control hace parte del terminal de datos.

Antes de que el Terminal de datos pueda acceder a los servicios que presta la Gateway, los dos dispositivos deben haber realizado los procedimientos de inicialización entre ellos. Estos procedimientos incluyen generalmente la activación manual de la aplicación y la entrada de un PIN para autenticación, según se describe en el perfil de Acceso Genérico.

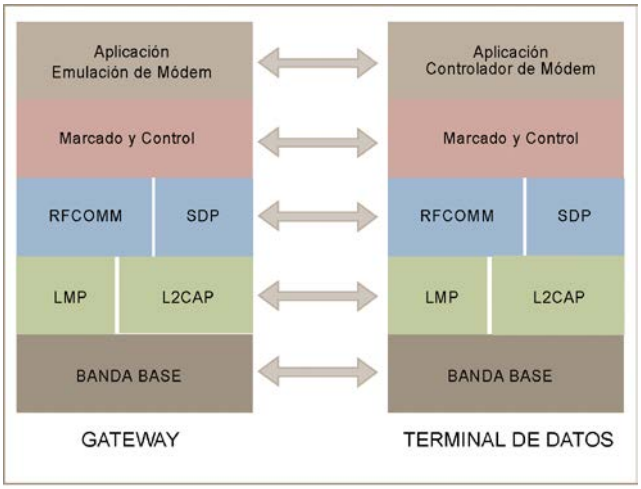


Figura 4.18 Protocolos Bluetooth y entidades de soporte para el perfil de Dial Up

El perfil solo permite configuraciones punto a punto y por lo tanto esta en capacidad de soportan una llamada a la vez. El audio se transporta a través del enlaces SCO, y se confiere seguridad al perfil, si se implementan los mecanismos de autenticación y encriptación, que proporciona la capa *Link Manager*.

4.11.4 Comandos AT Generalidades

4.11.4.1 Formato de comandos AT

Todos los comandos AT empiezan con la secuencia AT. La excepción es el comando A/ con el que se repite el último comando introducido.

Los comandos mas simples son:

- ATH: Indicación para “colgar el teléfono”.
- ATDT: Orden para marcar un número de teléfono determinado empleando la marcación por tonos.
- ATDP: Idem ATDT pero usando marcación por pulsos

Los comandos comienzan con las letras AT y siguen con las letras del alfabeto (A..Z). A medida que los modem se hicieron más complicados, surgió la necesidad de incluir mas comandos, son los comandos extendidos y tienen la forma AT&X (por ejemplo), donde el "&" marca la "X" como carácter extendido.

4.11.4.2 Control de Flujo

El control de flujo es un mecanismo por medio del cual se gestiona el intercambio de información entre la Gateway (Módem) y el Terminal. Estos mecanismos permiten detener el flujo cuando uno de los elementos no puede procesar mas información de la que le llega en determinado momento.
Existen dos tipos principales de comandos AT para procedimientos de control

- Comandos que ejecutan acciones inmediatas (ATD marcación, ATA contestación o ATH desconexión)
- Comandos que cambian algún parámetro de configuración (por ejemplo ATS7=90)

4.11.4.3 Modos de operación

Estado de comandos: En este estado, el módem responde a los comandos que envía el terminal cuando no existe conexión. En este modo es posible realizar las operaciones de marcado y conexión.

Modo en Línea: Es el estado en que se encuentran la Gateway y el Terminal, cuando se ha establecido la conexión.

4.11.4.4 Códigos de resultados

Cuando el Terminal (por ejemplo un computador personal) envía un comando al módem, la respuesta se devuelve en términos de códigos de resultado. Estos códigos de resultado pueden ser "CONNECT", "OK" o "ERROR".

1. ATV determina el tipo de código de resultado que aparecerá:
    1. ATV0 respuesta numérica
    2. ATV1 respuesta de palabras
2. ATQ1 inhibe los códigos de resultado, pone el módem en "estado silencioso"
3. ATQ0 habilita los códigos de resultado, desconecta el modo silencioso

4.11.4.5 Circuitos

Un módem se compone de tres circuitos modulares: el circuito de recepción de datos digitales, el circuito de emisión de datos analógicos y una unidad de control del módem.

El perfil de Dial Up requiere la implementación de dos circuitos que se definen en la especificación V.250 de la ITU; estos circuitos son:

Terminal de Datos Listo DTR (Data Terminal Ready) (Circuit 108) &D. Este circuito envía una señal para indicar al terminal que la Gateway está conectada y lista para iniciar la comunicación. Si la señal se pone a OFF mientras el módem esta en el Modo en Línea, el módem termina la sesión y cuelga la llamada.

Detector de Línea de Señal Recibida RTS (Received Line Signal Detector) (Circuit 109) &C. El terminal puede interceptar la señal que origina este circuito, de manera que la emisión de códigos de resultado pueda coordinarse apropiadamente con las transiciones de esta señal

4.11.4.6 Descripción de comandos

A continuación se describen los comandos AT básicos empleados por el Perfil Dial Up.

- Dial
  D[<dial_string>][;]

  Este comando proporciona las instrucciones a la Gateway para establecer una llamada. Este comando puede incluir la realización de varios pasos dependiendo del tipo de Gateway; entre ellos se tiene: conexión a la línea, espera por tono de marcado y monitoreo de línea.

- Select Tone Dialing
  T

  Indica que el tipo de Marcación se realizará por tonos.

- Select Pulse Dialing
  P
  Indica que el tipo de Marcación se realizará por pulsos.

- Hook Control
  H[<value>]
  Este comando proporciona las instrucciones a la Gateway para desconectarse de la línea y terminar cualquier proceso de ejecución de la llamada.

- Return to Line Data State
  O[<value>]
  A través de este comando, la Gateway retorna al estado "*Datos en Línea*" y transmite un código de resultado Connect o Connect <text>.

- Pause Before Blind Dialling
  SG
  Este parámetro especifica la cantidad de tiempo en segundos que debe esperar la Gateway por la información de señalización después de que se presenta la conexión con

la línea, cuando no se han implementado las funciones para detección de tono de marcado.
Valor: 2-10 segundos

- Connection Completion Timeout
  S7
  Este parámetro especifica la cantidad de tiempo en segundos que la Gateway debe esperar por una respuesta de conexión. Si la conexión no se establece durante este tiempo, la Gateway se desconecta de la línea y retorna un código resultado indicando el motivo de la desconexión.
  Valor: 1-255 segundos

- Automatic Disconnect Delay
  S10
  Este parámetro indica la cantidad de tiempo en decenas de segundos, que la Gateway debe permanecer en línea después de que ha sido notificada de la pérdida de la señal.
  Valor: 1-254

- Monitor Speaker Loudness
  L[<value>]:
  Este parámetro controla el volumen del micrófono. El nivel específico de carga depende del fabricante del dispositivo.

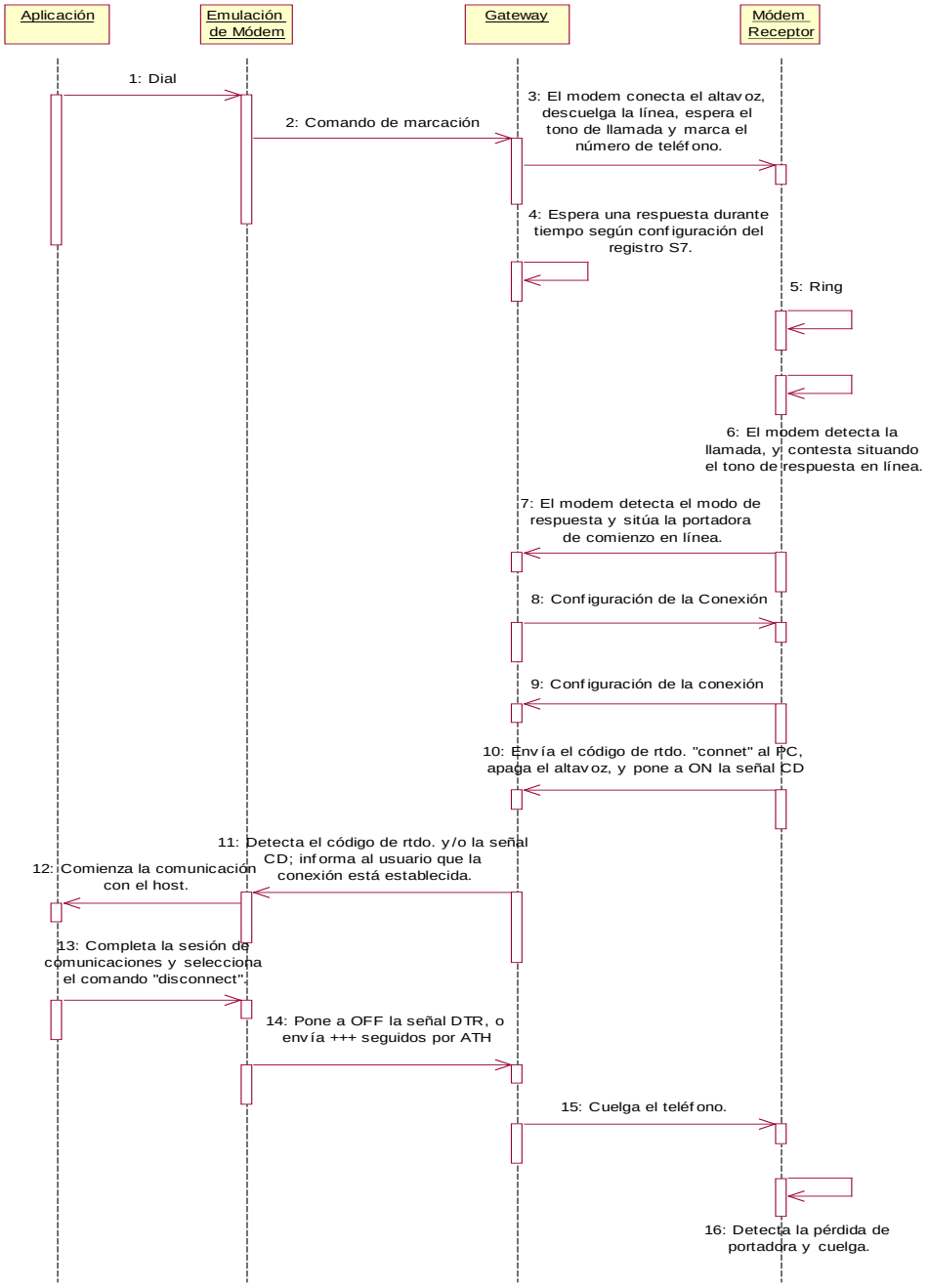
4.11.5 Audio Feedback

El perfil de Dial Up presta un soporte adicional para audio feedback. A través del Audio feedback, los tonos del módem asociados con un llamada pueden devolverse al terminal de datos para que puedan reproducirse por medio de los canales de audio SCO, y entreguen información acerca del proceso de una llamada, al usuario del servicio.

4.11.6 Cartas de Secuencia para el Establecimiento de la Conexión

A continuación se ilustra el proceso para establecimiento de una conexión entre dos dispositivos, a través de marcación telefónica.

Ver página siguiente...



Carta de Secuencia 4.16 Establecimiento de la conexión a través de marcación telefónica

4.11.7 Protocolos y Perfiles de Bluetooth empleados por el Perfil de Dial Up

4.11.7.1 Protocolo de descubrimiento de Servicios SDP

A continuación se definen las entradas hacia la base de datos (Registro del servicio) para el servicio de marcación telefónica.

Tabla 4.32 Registro del servicio para el perfil de Dial Up

ITEM	SUBITEM	VALOR
Lista de Identificadores de Clases de Servicio (Service Class ID List)	Clase de Servicio #0	Dial Up Networking
	Clase de Servicio #0	Telefonía Genérica
Lista de Descriptores de Protocolo (Protocol Descriptor List)	Protocolo #0	L2CAP
	Protocolo #1	RFCOMM
Parámetro para el Protocolo #1	Canal Servidor	1,2,3,.....30
Nombre del Servicio (Service Name)		Definido por el proveedor de Servicios. Puede usare “Dial Up Networking”
Lista de Descriptores de Perfil (Profile Descriptor List)	Perfil #0	Dial Up Networking
	Parámetros de Perfil #0	Versión del Perfil 0x0100
Soporte para Audio Feedback		Si/No

4.11.7.2 Protocolo Para Gestión De Enlace LMP

En adición al soporte de los procedimientos que establece la especificación del Link Manager, este perfil exige el soporte de las funcionalidades LMP que se establecen para el Perfil de Puerto Serial. Referirse a la sección 4.3.2.2.

Este perfil exige el soporte para el establecimiento de enlaces SCO en ambos dispositivos Gateway y Terminal de Datos.

4.11.7.3 Perfil de Puerto Serial

El perfil Dial Up emplea las funcionalidades que describe el perfil de puerto serial relacionadas con la operación de las capas RFCOMM y L2CAP.

4.11.7.3.1 RFCOMM

Estas funcionalidades son principalmente aquellas que se refieren a señales de control RS232 , indicación de estado remoto y negociación de puerto remoto. Referirse a las secciones 4.3.2.1 y 4.3.2.2.

4.11.7.3.2 L2CAP

A continuación se listan las funcionalidades de L2CAP, que emplea el perfil serial como soporte al perfil de Dial Up.

Tabla 4.33 Procedimientos L2CAP empleados por el perfil Dial Up

	PROCEDIMIENTO	SOPORTE PARA LAS DOS UNIDADES
Tipos de canales	Canales orientados a conexión	Opcional (OP)
	Canales orientados a conexión	No Aplica (N/A)
Señalización	Establecimiento de la conexión	Obligatorio (O)
	Configuración	O
	Finalización de la conexión	O
	Eco	O
	Rechazo de comandos	O
Configuración de Parámetros	Unidad de transferencia máxima	O
	Flush Time Out	O
	QoS	OP

4.11.7.4 Perfil de Acceso Genérico

La siguientes tablas relacionan la funcionalidad de Modos de Descubrimiento y aspectos de seguridad del perfil de Acceso genérico para los dispositivos que participan el Perfil de Dial Up.

- Modos de Descubrimiento

Tabla 4.34 Modos de descubrimiento aplicables al perfil de Dial Up

	PROCEDIMIENTO	SOPORTE TD	SOPORTE GW
Modos de Descubrimiento	Modo de No –Descubrimiento	No Aplica (N/A)	Opcional (OP)
	Modo de Descubrimiento Limitado	N/A	OP
	Modo General de Descubrimiento	N/A	OP
Modos de Conectividad	Modo de No – Conectividad	N/A	N/A
	Modo de Conectividad	N/A	O
Modo de Pairing	Modo de No-Pairing	Obligatorio (O)	OP
	Modo pairing	OP	O

- Aspectos de Seguridad

Tabla 4.35 Aspectos de seguridad aplicables al perfil de Dial Up

PROCEDIMIENTO	SOPORTE EN EL TL	SOPORTE EN LA GW
Autenticación	Obligatorio (O)	O
Modo de Seguridad 1	No Aplica (N/A)	N/A
Modo de Seguridad 2	Condicional (C)	C
Modo de Seguridad 3	C	C

4.12 Perfil de Acceso a LAN (LAN Access Profile LAP)

4.12.1 Introducción

El perfil de Acceso a LAN es el segundo perfil que junto con el Dial Up permite instanciar el escenario de uso “Puente a Internet”.

Así como el perfil de Dial Up, el perfil de Acceso a LAN habilita un dispositivo de computo para obtener acceso a una red de Datos, pero en lugar de emplear un teléfono o un módem utiliza un Punto de Acceso a Datos que esta conectado a la red externa.
Usar el perfil de Acceso a LAN, es similar a establecer una conexión con una red de datos mediante un cable Ethernet, pero de manera inalámbrica restringiéndose al empleo del protocolo Punto a Punto, sobre RFCOMM.

Las razones por las cuales se escogió este protocolo son:

- PPP es una de la formas mas empleadas para permitir el acceso a redes.
- Proporciona mecanismos de autenticación, encriptación, compresión de datos y soporte multi-protocolo.
- Muchos dispositivos, incluyendo los PDA’s soportan comunicación IP sobre PPP para maraca a redes IP.

4.12.2 Pila de Protocolos Bluetooth empleada por el perfil de Acceso a LAN

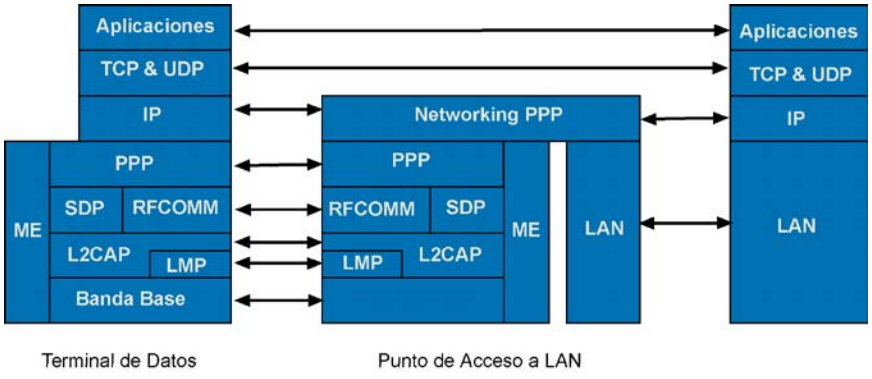


Figura 4.19 Pila de protocolos para el perfil de Acceso a LAN

La figura 4.19 ilustra la pila de protocolos para implementar el perfil de acceso a LAN. Además de las capas convencionales de Bluetooth, se incluyen las siguientes capas adicionales:

- **PPP**: Protocolo Punto a Punto.
- **Intertrabajo o *Networking PPP***: Esta capa establece la forma en que se toman paquetes IP de la capa PPP para introducirlos en la LAN.
- **Entidad de Gestión *Management Entity* ME**: Se encarga de coordinar los procedimientos de inicialización, configuración y conexión.

4.12.3 Configuración y Roles de los Dispositivos que participan en el LAP

Punto de Acceso a LAN LAP

Es el dispositivo Bluetooth que proporciona el acceso a LAN. El LAP provee de los servicios que ofrece un servidor PPP. RFCOMM se emplea para transportar paquetes PPP y controlar el flujo de los mismos.

Terminal de Datos TD

Este el es dispositivo que hace uso de los servicios ofrecidos por el LAP. Algunos ejemplos de terminales de datos son los computadores personales, *desktops* y PAD's.
El DT actúa como un cliente PPP.

4.12.4 Escenarios de Aplicación.

Los escenarios de aplicación cubiertos por este perfil son:

Un DT utiliza un LAP como una forma inalámbrica de conexión a una red de área local. Una vez conectado, el DT operará como si estuviera conectado a la LAN a través de marcación telefónica. El DT podrá acceder a todos los servicios que brinda la LAN.

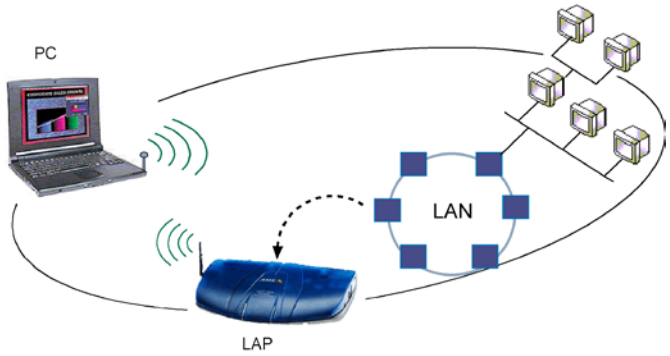


Figura 4.20 Perfil de Acceso a LAN escenario de aplicación 1

Múltiples DT usan un LAP para conectarse a una LAN. Una vez se conecten, los DT operarán como si estuvieran conectados a la LAN a través de marcación telefónica. Los DT podrán acceder a todos los servicios que brinda la LAN.

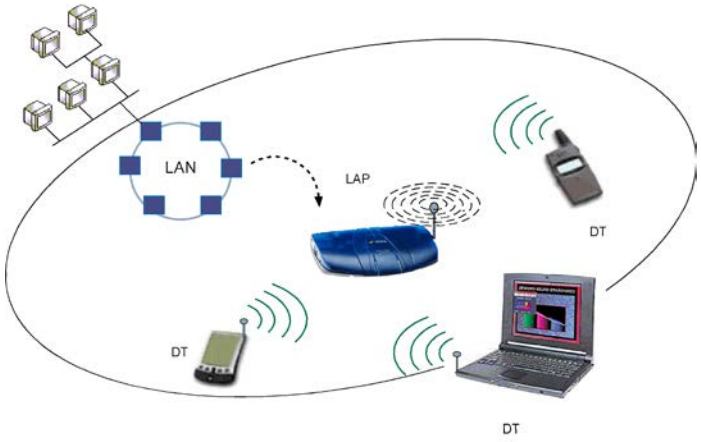


Figura 4.21 Perfil de Acceso a LAN escenario de aplicación 2

Conexión PC a PC. Este escenario ocurre cuando dos dispositivos Bluetooth (computadores personales o *desktop*) forman una conexión simple entre ellos. En este caso uno de los dispositivos toma el papel de LAP mientras que el otro asume el papel de DT.

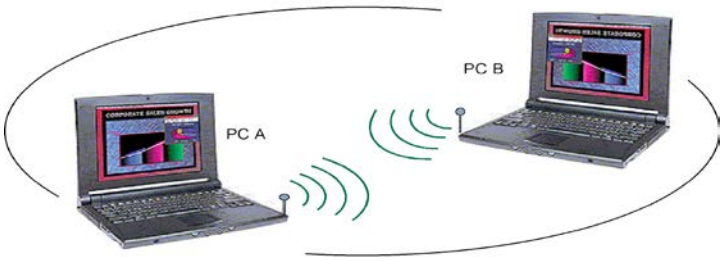


Figura 4.22 Perfil de Acceso a LAN escenario de aplicación 3

4.12.5 Fundamentos del Perfil

4.12.5.1 Establecimiento Básico de la Conexión

A continuación se describen brevemente, las interacciones que tienen lugar entre dos dispositivos que operan como LAP y DT en el perfil de Acceso a LAN.

- En primer lugar el DT emplea un tipo de aplicación para encontrar un LAP en el área de cobertura de su radio Bluetooth, con el fin de hacer uso de los servicios de los que dispone.

- Si no existe un enlace físico, a nivel de banda base, el DT debe establecer este enlace por medio de una petición de conexión al LAP.

- En seguida el DT establece la conexión PPP /RFCOMM / L2CAP.

- De forma adicional, el LAP puede hacer uso de los mecanismos de autenticación PPP, para solicitar la introducción de un PIN, un nombre de usuario y/o la clave de acceso a la red.

- Se negocia una dirección IP apropiada entre el LAP y el DT haciendo uso de procedimientos característicos de PPP.

- La conexión PPP se ha establecido; el tráfico IP puede fluir a hora a través de esta conexión.
- En cualquier instante, el DT o el LAP pueden terminar la conexión PPP.

4.12.5.2 Seguridad

En ambiente inalámbricos, la seguridad es un aspecto de gran importancia. Si se habla del perfil de Acceso a LAN, tanto el DT como el LAP deben soportar los procedimientos de encriptación, mientras que el tráfico PPP fluye entre las dos unidades.

4.12.5.3 Número de Usuarios

Según el fabricante y la aplicación, un dispositivo Bluetooth puede contar con capacidades diferentes y recursos limitados; lo cual restringe el número de usuarios que pueden hacer uso de el simultáneamente. El LAP debe proporcionar la capacidades para que pueda configurarse en uno de los siguientes modos de operación:

- Modo de Usuario Simple: En este caso el LAP puede ser accedido por un solo usuario a la vez. En este modo de operación tanto el DT como el LAP pueden asumir el papel de maestro de la Piconet.

- Modo de Usuario Múltiple: En este caso múltiples usuarios pueden acceder al as servicios del LAP de forma simultánea. En esta ocasión, es obligatorio que el LAP se establezca como maestro de la Piconet. Le corresponde a la entidad de Gestión, asegurar que los papeles se tomen de esta forma. El *Link Manager* ofrecer le soporte para Cambio de Rol, en caso de que este sea requerido.

4.12.6 Capa de Aplicación

4.12.6.1 Inicialización del Servicio a LAN

Este procedimiento permite establecer los siguientes parámetros de configuración.

- Máximo Número de Usuarios.
- Modo de descubrimiento (Acceso Genérico)
- PIN Bluetooth o claves de enlace
- Opciones de operación PPP. (Autenticación, Compresión etc.)

Cuando se completa la operación de configuración, los dispositivos están listos para aceptar conexiones PPP. Para aquellos dispositivos que están destinados únicamente a la funcionalidad de LAP, los procedimientos anteriores generalmente se llevan a cabo en momento en que se enciende el dispositivo.

4.12.6.2 Suspención del servicio de Acceso a LAN

Este procedimiento detiene la operación del LAP. En primer lugar se apaga el servidor PPP. Todas los enlaces PPP se desconectan y se remueven todas las entradas de servicio PPP de la base de datos del servidor SDP. El dispositivo puede ejecutar operaciones adicionales para eliminar la información del usuario, con el fin de evitar accesos no autorizados al servicio.

4.12.6.3 Establecimiento de la conexión

Normalmente es el DT quien inicia el establecimiento de la conexión con el LAP.

1. El primer paso es seleccionar el LAP que desea utilizarse así como el servicio PPP/RFCOMM que se encuentra disponible en el punto de Acceso a LAN.

Esta selección puede realizarse de la siguientes maneras:

- Al usuario de DT se le presenta una lista de LAP y sus servicios respectivos para que seleccione con cual de ellos desea trabajar.
- Se presenta al usuario una lista de servicios que ofrecen un grupo de LAP en el área de cobertura del terminal de datos. El usuario debe escoger el servicio ala que desea acceder y el DT se encarga de seleccionar automáticamente el LAP que presenta las mejores condiciones de prestación del servicio.
- El usuario introduce el nombre de un servicio que necesita y el DT se encarga de seleccionar el LAP que brinda acceso al servicio solicitado.
- Alguna Aplicación en el DT busca y escoge automáticamente servicios y LAP que se encuentren disponibles en momento.

En cualquiera de los casos anteriores, se emplean los mecanismos de descubrimiento de servicios del SDP para adquirir la información que desea presentarse al usuario.

2. Puede requerirse la introducción de un PIN por parte del usuario. En caso contrario se emplea un PIN de longitud cero para efectos de conexión.

3. También puede requerirse la introducción de un nombre de usuario y/o contraseña para realizar autenticación PPP.

4. Cuando el usuario (o aplicación)activa la conexión, da paso al inicio de la aplicación

4.12.6.4 Pérdida de la conexión con la LAN

Si se pierde la conexión con la LAN, el LAP debe notificar al terminal y este a su vez al usuario acerca de la falla en la conexión. Opcionalmente, la aplicación puede permitir el restablecimiento de la conexión a través de la reutilización de los parámetros de configuración y datos de la conexión que acaba de terminarse.

4.12.6.5 Desconexión del Enlace

Tanto el LAP como el DT pueden dar termino a la conexión en el momento que deseen.

4.12.7 Operaciones PPP

La operación de PPP sobre RFCOMM que se establece en este perfil es similar a la operación de PPP a través de marcación telefónica, excepto por el uso de los comandos AT que no se emplean en este perfil.

4.12.7.1 Inicialización del PPP

El LAP se encarga de registrar el Servidor PPP en la base de datos del Servidor SDP para descubrimiento de servicios. Un dispositivo en el rol de DT no está obligado a registrar PPP en su base de datos SDP.

4.12.7.2 Establecimiento de la conexión PPP

Si no existe una sesión RFCOMM entre el LAP y el DT, el dispositivo que inicia la conexión debe establecerla como primera medida.

A través de la utilización del Link Control, pueden negociarse algunos parámetros relacionados con la conexión como por ejemplo la Unidad máxima de Transferencia MTU.

4.12.7.3 Desconexión de la conexión PPP

La conexión PPP puede terminarse por las siguientes razones:

1. Intervención del Usuario
2. Falla de la conexión RFCOMM / L2CAP; por ejemplo cuando el enlace de radio falla o el dispositivo se sale del rango de cobertura de la Piconet.
3. Orden de finalización impartida por el LAP cuando este no se encuentra en capacidad de seguir prestando el servicio. Las razones por cuales puede presentarse este situación pueden ser:
    - Detección de una dirección IP duplicada
    - Perdida de la conexión con la LAN
    - Perdida de la conexión con el servidor PPP

4. Alguna política dependiente de la Aplicación puede causar también la terminación de la comunicación.

Cuando la conexión PPP se termina, deben seguirse los siguientes pasos.

- Finalizar las conexiones IPCP; esto causa que se deshabite la interfaz IP
- Finalizar las conexiones a nivel del *Link Control*
- Desconectar los enlaces RFCOMM, como se describe en el perfil de puerto serial.

Cuando la conexión se termina debido a una desconexión repentina, deben seguirse los siguientes procedimientos.

- Finalizar las conexiones IPCP; esto causa que se deshabite la interfaz IP
- Finalizar las conexiones a nivel del *Link Control*

4.12.8 Negación del Acceso por presencia de errores en la implementación del perfil

El LAP debe negar al DT el accesos a servicios PPP si se presenta una de las siguientes situaciones.

- Hay falla en el proceso de Pairing
- Ausencia de soporte para encriptación
- Ausencia de soporte de los procedimientos para cambio de rol.

El LAP debe rechazar cualquier intento por parte del DT de realizar los siguientes procedimientos.

- Realizar peticiones para que la encriptación se deshabite.
- Realizar peticiones para que el LAP que opera como maestro, ejecute un cambio de rol si está trabajando en el modo de usuario múltiple.
- Realizar peticiones para establecer una nueva conexión, cuando se han establecido el máximo número de conexiones permitidas por el valor de Número de Usuarios máximo.

4.12.9 Protocolos y Perfiles de Bluetooth empleados por el Perfil de Acceso LAN

4.12.9.1 RFCOMM

El perfil Headset emplea las funcionalidades que describe el perfil de puerto serial relacionadas con los requerimientos de interoperabilidad para RFCOMM. Sección 4.3.2.1y 4.3.2.2

En adición, debe tenerse en cuenta:

Con el fin de maximizar la throughput es recomendable que RFCOMM haga uso de paquetes a nivel de banda base que ocupan entre tres y cinco slots.

Como se define en la capa RFCOMM de la descripción de protocolos, la velocidad de las conexiones RFCOMM no puede ser configurada por el usuario. La velocidad de transferencia depende de la cantidad de trafico presente en el enlace.

4.12.9.2 Protocolo de descubrimiento de Servicios SDP

A continuación se definen las entradas hacia la base de datos (Registro del servicio) para el caso de un LAP que presta el servicio de Acceso a LAN.

Tabla 4.36 Registro de servicio para un punto de acceso a LAN

ITEM	SUB ITEM	VALOR
Lista de Identificadores de Clases de Servicio (Service Class ID List)	Clase de Servicio #0	LAN Access using PPP
Lista de Descriptores de Protocolo (Protocol Descriptor List)	Protocolo #0	L2CAP
	Protocolo #1	RFCOMM
Nombre del Servicio (Service Name)		"LAN Access using PPP"
Descripción del Servicio		Acceso a LAN
Mascara de subred		Configurable

4.12.9.3 Protocolo de Adaptación y Control de Lógico de Enlace L2CAP

El perfil Headset emplea las funcionalidades que describe el perfil de puerto serial relacionadas con los requerimientos de interoperabilidad para L2CAP Sección 4.3.2.1 y 4.3.2.2

En adición, debe tenerse en cuenta que el valor mínimo de MTU que una implementación L2CAP debe usar para soportar este perfil se determina por el parámetro *Máximun Frame Size* definido en el RFCOMM.

4.12.9.4 Protocolo Para Gestión De Enlace LMP

El perfil de Acceso a LAN debe soportar los siguientes funcionalidades del protocolo *Link Manager.*

Tabla 4.37 Procedimientos LMP empleados por el perfil de Acceso a LAN

FUNCIONALIDAD	SOPORTE EN LAP	SOPORTE EN DT
Autenticación	Obligatorio (O)	O
Pairing	O	O
Encriptación	O	O
Petición para cambio de Papeles	O	O
Procedimiento para cambio de Papeles	O	Opcional (OP)

Lista de tablas Capitulo 4

Lista de figuras

Lista de cartas de secuencia

CAPITULO 5. ENTORNO DE APLICACIÓN DE BLUETOOTH

5.1 INTRODUCCIÓN

Alrededor de una tecnología en desarrollo generalmente se ubican empresas de diversa índole, propósitos, estudios y pronósticos con el fin de hacer que tal desarrollo sea beneficioso para todos los que toman parte en él, y evaluar en el camino qué tan bien se desempeña tal tecnología, y esto no solo en el campo técnico, también juegan un papel muy importante (quizás igual de importante) la forma como se espera que la tecnología llegue a desarrollarse, la viabilidad económica, la técnica, los campos de aplicación que tenga, el impacto sobre la industria, los negocios y la vida cotidiana, los servicios que presta y las nuevas necesidades que plantea; todo esto promueve la aparición de varios actores participantes; Bluetooth y su influencia en las telecomunicaciones y en las aplicaciones telemáticas deja sentado un primer paso para el desarrollo del entorno móvil en que se desenvolverán las personas en un futuro cercano.

Este capítulo pretende mostrar las posibilidades y las pruebas que debe superar Bluetooth como tecnología en desarrollo para poder abrirse paso, tanto en el mercado, como en la industria, en el área de servicios, de suministro de equipos, en el campo de evolución de la especificación y de desarrollo de nuevas aplicaciones. Todos estos puntos de vista conforman el entorno de aplicaciones telemáticas; cada punto de vista le va a imprimir a la tecnología tendencias y posibilidades que también se mencionan y permiten realizar pronósticos en algunos de esos puntos.

5.2 Entorno de Aplicaciones Telemáticas

El entorno de las aplicaciones Telemáticas alrededor de la tecnología Bluetooth puede enmarcarse por los actores que tienen parte en su desarrollo; por ejemplo, una tecnología como Bluetooth puede hacer que haya nuevas y deslumbrantes aplicaciones muy útiles, pero hay que juzgar aspectos como

- De verdad la tecnología, ¿será accesible por todos?
- ¿Están los proveedores de equipos y software preparados para soportar los cambios que generan estas aplicaciones?
- ¿Se podrá introducir aplicaciones de cualquier tipo para promover esta tecnología?
- ¿Qué hará a estas aplicaciones interactuar si son de diferentes fabricantes?
- ¿Habrán aplicaciones ó elementos de aplicación parecidos que se puedan modificar de tal manera que no se tenga que inventar de nuevo?

Luego de realizar juicios como estos puede vislumbrarse el entorno de las aplicaciones telemáticas Bluetooth desde perspectivas muy diferentes, perspectivas de los actores de tal entorno, como los que se mencionan en los puntos siguientes.

5.2.1 Las Aplicaciones Telemáticas

Las aplicaciones que puedan desarrollarse en particular con tecnología Bluetooth se pueden enmarcar, como se vio en el capítulo 2, en modelos de uso que pretenden dar respuesta a necesidades de comunicación de dispositivos móviles como PDAs, teléfonos celulares, equipos personales y puntos de acceso; particularmente constituyen maneras de reemplazar los cables y de dar nuevas facilidades a tales dispositivos.

En general la creación de aplicaciones se ve impulsada y también frenada por diferentes factores, tales factores son por ejemplo:

Impulsores:

- La tecnología habilita a los dispositivos para que tengan a disposición comunicaciones inalámbricas y dotadas de un buen ancho de banda.

- La incertidumbre frente a los servicios de Internet inalámbrico que prestan las redes móviles.
- La eficacia de las soluciones inalámbricas de distancias cortas gracias al concepto de puntos de acceso.
- El surgimiento del comercio móvil con aplicaciones que generalmente están presentes en dispositivos personales, lo cual hace que se desarrollen transacciones a corta distancia.

Obstáculos:

- La falsa expectativa de que Bluetooth puede brindar un conjunto de servicios de Red móvil, como por ejemplo, roaming ó aspectos de infraestructura.
- La estimación que tienen los programadores de que habrá mucha penetración inicial de dispositivos Bluetooth en el mercado.
- Que los Fabricantes de dispositivos deben darse a la tarea de producir sus propios elementos de soporte Bluetooth para sistemas operativos como Windows, dado que dicha compañía decidió no brindar este soporte (hasta la fecha).

5.2.1.1 Mapa de Clasificación de las Aplicaciones

Las aplicaciones con Bluetooth pueden también clasificarse en cuanto a dos factores simultáneos: los segmentos del mercado, y el área y formas de conectividad que ofrece para los dispositivos móviles:

La primera columna ofrece tres olas de posibles aplicaciones; la primera de conectar dispositivos personales se refiere a reemplazar cables existentes en aplicaciones como manos libres, conexión de PDAs con teléfonos celulares.

La segunda se refiere a enlazar dispositivos dentro de la misma entidad, es decir, dispositivos que se encuentren en un mismo entorno, ya sea este empresarial, académico ó del hogar, de tal manera que no necesariamente será dirigido a dispositivos móviles.

La tercera señala a aquellas aplicaciones que presten servicios de acceso, usualmente entre dispositivos foráneos y de utilización esporádica; aquí se enmarcan las aplicaciones de puntos de acceso y servidores de acceso y de contenido en lugares públicos, por ejemplo.

En cuanto a los segmentos del mercado, se encuentran el de los negocios, el de los consumidores y el mercado vertical, este último se refiere a las aplicaciones dedicadas, lectores de códigos de barras, monitores de seguridad, aplicaciones de manufactura, aplicaciones en transporte y medicina, aplicaciones que serán creadas específicamente para tales fines.

Tabla 5. 1 Mapa de clasificación de las aplicaciones

	Vertical	Negocios	Consumidores
Conectar dispositivos personales	Potencial: Alto Crecimiento: lento (requiere desarrollos específicos)	Potencial: Medio Crecimiento: Medio	Potencial: Alto Crecimiento: lento (Usuarios principales: aficionados)
Conectar dispositivos en una entidad	Potencial: Alto Crecimiento: Medio (requieren de integración con equipos existentes)	Potencial: Medio Crecimiento: Medio	Potencial: Alto Crecimiento: lento (depende de la integración en una amplia gama de dispositivos)
Conectar dispositivos foráneos	Potencial: Alto Crecimiento: Muy lento	Potencial: Medio Crecimiento: lento (no se les exigirá demasiado en cuanto a su desempeño)	Potencial: Alto Crecimiento: lento (Requiere una alta penetración en el mercado de dispositivos personales)

5.2.2 Operadores de Red

Bluetooth tendrá un pequeño impacto en los ingresos de las redes móviles. A pesar de algunos desarrollos aislados, los servicios Bluetooth de acceso público todavía no tiene cabida en el mercado actual. Los modelos de negocio para tercera generación y servicios de Internet inalámbrico aún lucen inverosímiles para muchas aplicaciones.

Por el contrario, Bluetooth sacudirá el mercado de los dispositivos. Esta tecnología abre el camino para la producción de nuevos tipos de dispositivos y cambia el concepto del factor forma que se tenia hasta el momento. La *Utilidad* determinará el arribo de nuevos servicios. Bluetooth debe preocuparse por permitir que sus dispositivos puedan comunicarse y

sincronizarse mas fácil, independientemente del fabricante y de este modo evitar al usuario la presión a la que se ve sometido cuando desea comprar un dispositivo esperando que este le sirva para todo.

Es necesario recordar que Bluetooth es una de las opciones que utilizan comunicación de radio de corto alcance y seria pertinente no perder de vista tecnologías como HyperLAN2 y 802.11b, las cuales han entrado con fuerza en campos como el del comercio móvil. El costo de soportar múltiples estándares de transmisión es demasiado alto para los operadores y por esta razón es necesario tener especial cuidado a la hora de definir su posición frente a tecnologías como Bluetooth, HyperLAN2 y 802.11b cuando se piensa en ofrecer aplicaciones domésticas y de acceso publico.

5.2.3 Fabricantes de Teléfonos Móviles

5.2.3.1 Aparición de Nuevos Diseños

Bluetooth tiene la capacidad de dar posibilidades a los fabricantes para crear nuevos diseños, nuevos *Factores de forma* (es decir, configuraciones entre funcionalidad de teléfono y PDA). Permitirá nuevos tipos de dispositivos y nuevos tipos de interconectividad entre los ya existentes. También puede reducir el atractivo de los llamados “teléfonos inteligentes” ó *Smart Phones* al compararlos con la conectividad que habrá entre los teléfonos móviles corrientes y otros dispositivos como PDAs y computadores manuales, aunque la experiencia en ésta área es limitada y aun no es conveniente que los fabricantes basen sus expectativas en ello.

5.2.3.2 Oportunidades para Mejorar Productos Existentes

En principio, Bluetooth permite diferenciar entre gamas de productos, creando algunas con funcionalidad orientada al usuario final y para teléfonos del sector de los negocios. También la tecnología puede ayudar a modificar la curva de reemplazo de los teléfonos, dándoles a los consumidores las razones suficientes para cambiar sus teléfonos actuales.

Sin embargo todo depende críticamente de la disponibilidad de aplicaciones atractivas y de la interoperabilidad con otros dispositivos habilitados para Bluetooth; sin esta componente, la funcionalidad que se pueda ofrecer en los teléfonos no valdrá la pena; de aquí la dependencia de este sector hacia las aplicaciones que se desarrollen.

Posicionar a Bluetooth como un producto de lujo es algo delicado en estos momentos; pocos consumidores serán capaces de tener acceso a los dispositivos que tengan tales características. Lo conveniente sería hacer que dichas características en un producto y su costo en realidad valgan la pena; también no dejar en manos del usuario la tarea de probar si el producto final es de calidad.

5.2.3.3 Las Oportunidades para Acceso Público

Bluetooth hace que los teléfonos habilitados con tal tecnología puedan acceder a otros tipos de servicios además de aquellos que de entrada proveen las redes de teléfonos móviles; tal es el caso de aplicaciones de comercio electrónico en *billeteras electrónicas* ó en transacciones que requieran intercambio de clave de seguridad.

También debe ser posible utilizar teléfonos Bluetooth para conectarse a otros servicios de acceso públicos sin necesidad de tener que usar las redes de los operadores móviles, pero estas son aplicaciones interesantes más que modelos completos de negocio, y esto hace dudar si en realidad vale la pena reemplazar algunos servicios de operadores de red por servicios a través de Bluetooth.

5.2.4 Fabricantes de PDAs

5.2.4.1 Mejorarán los Modelos de Uso de los PDAs

Bluetooth constituye buenas noticias para los fabricantes de PDAs en el sentido que mejora la funcionalidad y uso de los productos, facilitando las comunicaciones entre PDAs (intercambio de tarjetas de presentación), con periféricos (impresoras, teclados) y en aplicaciones de sincronización.

5.2.4.2 Importa el Balance en la Clase de Procesamiento

aunque Bluetooth permite la aparición dispositivos nuevos, reduce el atractivo de dispositivos “todo en uno” comparado con aparatos especializados, es decir, la presión sobre lo fabricantes de PDAs será menor en el sentido que no tendrán que integrar características de teléfonos en sus dispositivos y mas bien buscar otras alternativas de alianzas con fabricantes de tales dispositivos.

5.2.4.3 Oportunidades de Diferenciación Fuertes

Al igual que con los fabricantes de teléfonos, hay oportunidades de diferenciación en sus productos, aunque estos estarían dirigidos hacia los consumidores con posibilidad de

adquirirlos ú orientados a personas de negocios. Puede habilitar también una gama de servicios nuevos, como por ejemplo el caso del acceso público a redes ó la facilidad de realizar transacciones de compras. Tanto para estos y otros ejemplos, el PDA puede constituir incluso el dispositivo ideal de acceso.

5.2.4.4 Hay Fuertes Competidores de Bluetooth

Bluetooth no es la única alternativa que se tiene para comunicar de forma inalámbrica los dispositivos personales. Existen otras tecnologías (estándares de la industria, mientras que Bluetooth aun esta como consorcio ó acuerdo industrial) como 802.11b que se pueden emplear ampliamente en ciertos escenarios; es necesario decidir a cual de las dos apuntar ó si mejor a ambas en paralelo.

Entre las ventajas de Bluetooth sobre otras tecnologías inalámbricas de corto alcance está su capacidad de ahorro de energía (y por consiguiente de duración de las baterías) y esto puede ser el factor determinante a la hora de escoger la alternativa tecnológica a emplear.

5.2.5 Fabricantes de Chips

Hay una gran cantidad de actores influyendo en este mercado. Los chips Bluetooth tienen que volverse más baratos, aunque este cambio en la realidad no será muy rápido. La meta de tener chips a un costo de 5 dólares no se alcanzará sino hasta el 2006.

También habrá campos para la diferenciación de productos, pero no tan abundantes como para otros actores, los fabricantes de dispositivos (los clientes de los fabricantes de Chips) buscarán componentes estándar en lugar de componentes “más que conformes” con la especificación.

Para poder tener éxito en esta capa del mercado tendrán que poder producir grandes volúmenes; usar su base de conocimiento acumulada para moverse en la cadena de valor y empezar a mirar cual sería la nueva ola de los dispositivos integrados; Por último También los fabricantes deben pensar en las posibilidades que se tengan frente a otras tecnologías de corto alcance, esto dará la oportunidad de innovar y de no concentrarse en una sola gama de productos.

5.2.6 Fabricantes de Computadores Personales

Los usuarios tanto consumidores como de negocios esperarán poder disponer de la funcionalidad Bluetooth al menos como una alternativa en productos de mediano consumo para dentro de 2 años más o menos.

Los Computadores personales habilitados con Bluetooth serán un primer eslabón entre dispositivos personales (PDAs, Teléfonos móviles ó combinaciones de ellos) y entre los dispositivos corporativos LAN. La habilidad para soportar estas características será casi obligatoria.

5.2.7 Desarrolladores de Aplicaciones con Bluetooth

Tres olas de desarrollo de aplicaciones. La primera se dedicará a enlazar dispositivos personales, tales como PDAs, PC Portátiles, Teléfonos, reproductores de MP3, manos libres, etc. La segunda tendrá que ver con aplicaciones que comuniquen dispositivos del mismo entorno (sea empresarial, académico, del hogar ó industrial) pero con énfasis en las aplicaciones empresariales. La tercera ola de desarrollo será la de comunicar a dispositivos en entornos diferentes (donde se hará necesario tener funcionalidad de gestión, por ejemplo, la gestión de los usuarios y si estos necesitan acceder a servicios, incluso si se les debe cobrar por ello). Las tres olas estarán diferenciadas por el tiempo en el cual tendrán lugar; el orden en que se mencionaron es el orden en que se espera tengan origen.

5.2.7.1 Aplicaciones Simples

Los desarrolladores de Aplicaciones con Bluetooth deben enfocarse en aplicaciones que sean simples y esenciales. La primera ola de aplicaciones (reemplazo de cables) sería un ejemplo de ello y tal vez llevará a una adopción masiva en el mercado de estas aplicaciones, lo cual es una precondición para que la tecnología evolucione junto con los productos.

También es posible desarrollar aplicaciones utilizando elementos que ya se hayan desarrollado, como por ejemplo, en las aplicaciones comerciales de ventas se puede aprovechar la infraestructura móvil de e-commerce; o por otro lado, aprovechar interfaces de programación que traigan implementadas las herramientas para interactuar con diferentes sistemas operativos.

Otra conducta de los desarrolladores es asumir que existen altos niveles de penetración de dispositivos, pero lo que si es cierto es que los fabricantes solo ofrecerán Bluetooth como

una característica de valor agregado y más aún, los usuarios pagarán por ello cuando las aplicaciones sean muy visibles. Las aplicaciones simples, que acarreen beneficios concretos al usuario final serán más viables y a la vez visibles inicialmente.

5.2.8 Impacto de Bluetooth para proveedores de servicios

Para los proveedores de servicio, el mercado inicial se encuentra en ambientes cerrados de alto valor. Los dispositivos personales no tendrán carácter universal para todos los escenarios. Por ende la prestación de servicios Bluetooth, es ideal para grupos cerrados de usuarios los cuales basan su operación en un conjunto de relaciones públicas bien definidas, como por ejemplo fabricantes y proveedores o distribuidores. En cualquier caso la descripción o promoción de un servicio o aplicación debe ir acompañada de una demostración apropiada, que ilustre al consumidor la forma en que operará la tecnología y lo que podrá conseguir si se suscribe al servicio o aplicación en cuestión. La búsqueda de buenos acuerdos comerciales puede ser un gran oportunidad para ofrecer servicios Bluetooth.

5.2.9 Impacto para los vendedores de equipos de Red

Es importante aclarar que no todos los negocios pueden hacer uso en este momento, de las tecnologías inalámbricas y por lo tanto los vendedores de equipos de red deben realizar una búsqueda extensa de los segmentos del mercado donde el nivel de movilidad y el ahorro de tiempo son dos aspectos que cobran gran importancia como fuerza de trabajo y la implementación de un solución inalámbrica ofrece buenas perspectivas.

El mercado de redes domésticas será significante para Bluetooth. El desarrollo de productos debe tener en cuenta el crecimiento actual de este mercado. Aunque algunas aplicaciones domesticas para Bluetooth son las mismas que se presentan en ambientes empresariales, hay aplicaciones únicas en el hogar que podrían beneficiarse de Bluetooth y de las que puede sacarse especial provecho si se reconocen a tiempo.

5.2.10 Mensajes para el Bluetooth SIG.

La interoperabilidad es absolutamente esencial para Bluetooth. El éxito o fracaso de la tecnología radica en la capacidad que ofrezca Bluetooth para asegurar el desarrollo de productos que comprometen aspectos de interoperabilidad. Un cuidado especial debe tenerse al estandarizar al nivel de software en la especificación ya que las aplicaciones de diferentes fabricantes pueden causar problemas del interoperabilidad adicionales.

El desarrollo de Bluetooth está atrasado y se nota la ausencia de productos en el mercado. Esto podría tener una repercusión negativa en los compradores que no consideran fuerte la entrada de la tecnología en el mercado. Sin embargo el SIG debe trabajar para asegurar que los consumidores reciban una imagen verdadera de lo que es en realidad Bluetooth y disuadir a los miembros de este grupo de poner a Bluetooth en entornos donde en realidad no tiene cabida. El grupo debe enfatizar la ventajas reales de Bluetooth como tecnología mas allá de los beneficios que ofrece la conectividad como tal. Esto con el fin de evitar una competencia falsa y desmedida con otras tecnologías de corto alcance.

El SIG debe ante todo prevenir la experiencia de WAP que pasó rápidamente de la cima a la desilusión. Debe manejar (controlar) expectativas y evitar prometer más de lo que puede entregarse; y por supuesto animar a sus miembros para que hagan lo mismo.

Otro aspecto de relevancia para el SIG es la necesidad de alentar la integración del protocolo. Ya que Microsoft ha decidido no incluir, por lo pronto, soporte para Bluetooth en sus sistemas operativos, el SIG debe exhortar a los vendedores de redes a participar en esta nueva tecnología. La capa física de Bluetooth necesita ser integrada con protocolos de alto nivel, como TCP/IP y sistemas operativos de mayor uso. Integrar Bluetooth a las variantes de Windows es una política de mercado necesaria para asegurar que Bluetooth alcance una masa crítica dentro del mercado. La integración con Linux es muy deseable dado el numero de dispositivos que usan este sistema operativo.

También debe concernir al SIG una resolución clara de los problemas de interferencia.
Variedad de criticas han sido levantadas en torno a Bluetooth y otras tecnologías de radio. Una de ellas es que estas interfieren unas con otras y también con otras aplicaciones como los hornos microondas. El SIG debe argumentar lo contrario a través evidencias independientes, con el fin de cerrar este debate lo más pronto posible.

CAPÍTULO 6 CONCLUSIONES Y RECOMENDACIONES DEL TRABAJO INVESTIGATIVO

6.1 LA TECNOLOGIA COMO ALTERNATIVA

No debe verse a Bluetooth como un competidor de las demás tecnologías inalámbricas de corto alcance (Ver Capítulo 2 y Anexo A), puesto que pertenecen a escenarios y aplicaciones diferentes.

802.11b fue pensada para ser una tecnología de LAN inalámbrica, ofrece entre otras características como soporte de Protocolo Configuración Dinámica (Dynamic Host Configuration Protocol – DHCP) y Privacidad Equivalente Cableada (Wired Equivalent Privacy - WEP), lo cual hace a esta tecnología una extensión de las LAN Cableadas.

802.11B ya ha ganado terreno con aplicaciones de acceso público, y esta será una razón para que se mantenga esta preferencia.

Bluetooth a pesar de trabajar en la misma banda del espectro radioeléctrico que 802.11b y que HomeRF, está dirigido a otras aplicaciones, tales como reemplazo de cables entre dispositivos personales y computación oculta (Hidden Computing) entre ellos.

Bluetooth ha sido diseñado para que su utilización en comunicaciones consuma pocos recursos del sistema en el que está instalado; para ello se han ideado aplicaciones simples (intercambio de voz punto a punto, aplicaciones de agenda, periféricos de computador) y en este sentido se vuelve eficiente en la escala de los dispositivos personales. Esto le reserva un campo de aplicación casi que exclusivo.

Por la misma razón anterior, los dispositivos de capacidades limitadas de procesamiento podrán beneficiarse con la tecnología, no solo dispositivos personales, sino industriales

(como sensores y transductores instrumentos de medida, etc), médicos (para monitoreo remoto, equipos de signos vitales) y en otros campos más verticales de aplicación.

También la adopción en forma parcial de protocolos y otros elementos de IrDA, GSM y aplicaciones típicas seriales hace que Bluetooth sea fácilmente implementable para aplicaciones ya existentes.

6.2 IMPLEMENTACIONES DE BLUETOOTH

6.2.1 Tecnologías Alternas de Apoyo

6.2.1.1 Hardware

Sin duda, uno de los mayores avances hardware que incorpora Bluetooth como tecnología es la parte de radio y de banda base; estas características funcionan y tienen lugar en un solo circuito integrado, junto con las ventajas de miniaturización y simplicidad que esto implica para los desarrolladores de hardware.

La posibilidad de miniaturización ofrece también una opción de utilizar antenas de diferente tipo, algunas impresas en la placa de circuito impreso en la que vienen.

Las tecnologías SMT puede hacer que la miniaturización sea mejor, y esto no solo es válido para los desarrolladores de hardware, sino también para aplicaciones específicas y colaborar con la filosofía de bajo consumo de energía.

6.2.1.2 Software

Orientación a objetos en el desarrollo de aplicaciones, con las ventajas que esto trae.

Desarrollo de manejadores ó *Drivers* para dispositivos con herramientas de desarrollo

Sistemas operativos empotrados y para dispositivos personales tipo PDA, como el caso de las Palm, Linux embebido y Epoc (de Symbian).

6.2.2 Herramientas de Desarrollo

La herramienta que debe procurarse para trabajar con Bluetooth debería cumplir con ciertos requerimientos, dada la ausencia de trabajos y desarrollos prácticos en ésta área:

En el Harware:

Traer un módulo Bluetooth dotado de la capacidad para ser manejado por una interfaz serial. Hay herramientas y kits que pueden tener tal característica disponible hacia la tarjeta de desarrollo en la que están incorporados, pero no hacia el desarrollador.

Permitir un funcionamiento independiente del software que se está empleando, si bien todos los módulos son controlados por un puerto serial, entonces cualquier implementación de la pila de protocolos (que incluya la interfaz HCI) que se tenga deberá hacer funcionar al mismo hardware de desarrollo.

En el Software:

Permitir que se tenga acceso a funcionalidades de los niveles inferiores de la pila de protocolos, es decir, evitar la verticalidad de las implementaciones que vengan con las herramientas de desarrollo, tal es el caso del kit de desarrollo de Ericsson, el cual posee un elemento Software vertical llamado SCM (Stack Control Manager), el cual no hace parte de la pila de protocolos de Bluetooth.

Preferiblemente sería hacer que tales herramientas posean una documentación en cuanto al modelado de las aplicaciones que puedan implementarse; esto hará que el desarrollador no tenga que de nuevo estudiar el API con el fin de “modelar” su funcionalidad.

Muy pocas de las herramientas software que se han encontrado (solo las de Linux) traen la implementación de la pila de protocolos en todos los niveles (incluyendo la interfaz HCI); esto puede ser necesario si se quiere tomar como base para aplicaciones de bajo nivel, por ejemplo, con microcontroladores y sistemas empotrados.

Una buena aplicación que podría servir como soporte para otros desarrollos puede ser un *Driver* ó manejador Bluetooth que sirva para múltiples aplicaciones (aunque sin pretender extender demasiado esta cobertura). El desarrollo de *Drivers* es una tarea extensiva, pero es de gran utilidad en el proceso del aprendizaje de esta tecnología, sobre todo en el estado inicial en que se encuentra Bluetooth; este trabajo ha pretendido incentivar esta clase de desarrollo.

Procurar en la escogencia del lenguaje de programación la portabilidad y la independencia del sistema operativo en el que se está trabajando, en éste sentido, pueden aprovecharse

las características de lenguajes como C ó C++, que pueden implementarse sobre microcontroladores con programas de compilación.

En lo que respecta a Java, ya existen también implementaciones de la pila de protocolos como la XJB 100 Bluetooth Host Stack de Zucotto Wireless, que también contiene un API y todo está implementado completamente en Java, aunque solo soporta los perfiles de servicio (Serial, Acceso Genérico y Descubrimiento de Servicios). El hecho que tales herramientas aparezcan pueden facilitar el trabajo para los desarrolladores, sobre todo cuando se tiene la posibilidad de que sus programas corran sobre distintas plataformas e incluso aprovechar las características de los procesadores Java para simplificar aún más la integración y rendimiento de la aplicación.

En la Investigación se encontraron las siguientes herramientas (entre otras):

Axis Communications – “AXIS OpenBT Stack”
Pila de Protocolos para Linux, implementa las capas de la pila desde HCI hasta RFCOMM y SDP, está escrita en lenguaje C y Contiene algunas aplicaciones de muestra de descubrimiento de dispositivos y de servicios.

Ericsson – “Bluetooth Application and Trainning Toolkit”
Implementación de la pila de protocolos y una API para Visual C++, utiliza un módulo Bluetooth montado en una tarjeta de evaluación que le provee la comunicación serial que necesita (por USB) y la alimentación. Trae dos aplicaciones de muestra y una aplicación llamada “comserver” que es necesario ejecutar cada vez que se utilice el API.

Bluetoolsonline – “HandyBlueStack COM object”
Implementación de la pila en forma de un objeto COM, el cual presenta 4 interfaces ó APIs a diferentes niveles, de Control, de HCI, de L2CAP y de gestión de la pila. Se pueden programar aplicaciones en tales niveles en Visual C++ y Visual Basic; también trae ejemplos de descubrimiento de dispositivos y de establecimiento de enlaces.

6.2.3 Desarrollo de Aplicaciones Telemáticas

Las aplicaciones de reemplazo de cables serán más aptas para desarrollar mientras la tecnología adquiere forma y es acogida por el mercado. Mientras esto ocurre se irá perfeccionando e incorporando en productos, lo cual permitirá a los desarrolladores pasar a crear aplicaciones más sofisticadas.

Los perfiles de aplicación son un paso adelante en la especificación de la manera como se transfiera información, ello permite que los desarrolladores no tengan que implementar por completo los protocolos, sino que indiquen qué perfiles de aplicación soporta su implementación, ahorrando tiempo en el desarrollo y garantizando la interoperabilidad entre sus diferentes aplicaciones.

El desarrollo de perfiles de aplicación es un buen ejercicio de desarrollo, ya que así pueden estudiarse alternativas de implementación de aplicaciones, por ejemplo las aplicaciones de transferencia de archivos pueden funcionar sobre un enlace serial, y en realidad puede implementarse tal aplicación y funcionar. Por otro lado, la misma aplicación puede hacerse implementando el perfil de transferencia de archivos; en ambos casos se habrá desarrollado una aplicación similar, pero en el segundo caso se ha empleado métodos de transferencia interoperables, mientras que en el primer caso, el desarrollador habrá tenido que implementar (casi sin quererlo inicialmente) su propio perfil de aplicación.

En caso de que el desarrollador no disponga de un kit de desarrollo con tarjetas de evaluación dotadas de módulos Bluetooth, existe la alternativa de utilizar algunas herramientas de emulación de la funcionalidad del módulo a través de software. De esta manera el desarrollador ejecuta un emulador simultáneamente en varios equipos conectados a una LAN; luego puede ejecutar la aplicación que desea probar en cada uno de los terminales y de esta manera los emuladores se encargan de formar las picoredes virtualmente y hacer creer a la aplicación que en realidad es un módulo Bluetooth el que están controlando. En este propósito se encuentra a IBM con el emulador “BlueHoc”.

6.3 DEBILIDADES DE LA TECNOLOGIA BLUETOOTH

Aún el costo de los dispositivos es alto mientras no se masifique su consumo y se tengan técnicas de fabricación de los chips más baratas.

También la inclusión de Bluetooth en algunos dispositivos que actualmente no lo tienen, implica requerimientos para tales dispositivos; el requerimiento principal, se refiere a las capacidades que debe tener un dispositivo para manejar datos y ofrecer un puerto serial, lo cual no necesariamente se va a encontrar en dispositivos como parlantes, micrófonos, manos libres inalámbricos existentes, televisores, etcétera. Esto implica que tales productos

tienen que variar en su configuración básica, incluyendo puertos seriales y conversores Analógico – Digitales en el caso de elementos de audio.

Si se tomara a Bluetooth como una tecnología de LAN Inalámbrica. Sería una desventaja el hecho que no soportara gestión de la movilidad, como por ejemplo, el uso de protocolos como DHCP, el cual maneja la configuración del equipo de manera dinámica a través de celdas de cobertura.

6.4 EVOLUCION DE LA TECNOLOGIA BLUETOOTH

La tecnología Bluetooth aún tiene cosas por especificar, puesto que las aplicaciones también tienden a evolucionar; por ejemplo, en un futuro (no muy cercano) habrán cosas que no serán necesarias tanto en las aplicaciones como en el hardware, la circuitería que soporte Bluetooth tal vez ya no requiera interfaces seriales y la interfaz HCI no será necesaria; esto es debido a que los criterios de simplicidad de la circuitería tenderán a que Bluetooth se integre por completo en muchos productos, eliminando la interfaz Hardware entre el módulo y el equipo.

Las aplicaciones Bluetooth también van a evolucionar, dado que es más liviano para la máquina utilizar protocolos de más bajo nivel para efectuar transferencias de información, entonces –específicamente las aplicaciones seriales- buscarán maneras más simples para realizar éstas operaciones (tal podría ser el caso de la instrumentación y sensores remotos) utilizando solamente hasta el protocolo L2CAP. Un ejemplo de estas aplicaciones sobre microcontroladores es denominado “A Minimal Bluetooth-Based Computing and Communication Platform” del *Swiss Federal Institute of Technology*.

Hay que tener en cuenta que ésta evolución de las aplicaciones en éste sentido debe ceñirse rigurosamente a la evolución paralela de los perfiles de aplicación.

6.5 SOPORTE EMPRESARIAL DE LA TECNOLOGIA BLUETOOTH

El apoyo decidido que ha recibido Bluetooth por diversos sectores de las telecomunicaciones y la industria en general, abre nuevas y más posibilidades para que la tecnología se convierta rápidamente en un estándar de aplicación mundial.

Las empresas que están desarrollando trabajos con Bluetooth son de diversa índole, desarrolladoras de software, fabricantes de dispositivos, integradoras de soluciones, fabricantes de chips, proveedores de servicios, la industria Automotriz y electrónica de consumo. Este panorama permite que la tecnología expanda sus perspectivas de cobertura, constituyendo soluciones , incluso en los entornos más inusuales.

6.6 EFECTOS EN LA SALUD HUMANA

En éste sentido se han consultado informes (que no son muy extensos en ésta área) y ellos arrojan resultados positivos respecto a la relación del uso de Bluetooth con la salud, por ejemplo, se hacen comparaciones de la potencia de radiación de un aparato Bluetooth con la de un teléfono celular, en éste caso, tendríamos una señal celular constante situada siempre en la misma frecuencia y de suficiente potencia para alcanzar la estación base; pero por otro lado la señal de Bluetooth es de muy baja potencia (0 dBm), y además esparcida en toda la banda de frecuencias del espectro ISM, lo cual hace que la exposición a ésta radiación es menos nociva en comparación con la mencionada primero.

6.7 RECOMENDACIONES PARA CONTINUAR ESTE TRABAJO INVESTIGATIVO

- Se debe continuar con la revisión e investigación de esta tecnología, ya que aparecerán nuevos perfiles de aplicación, incluso basados en los ya existentes.
- La especificación deberá seguir evolucionando, especificando ya algunas características que aparecían “reservadas” en la versión 1.1. Igualmente el grupo de trabajo en WPAN de IEEE y 802.15 van a darle a las tecnologías inalámbricas personales un lugar dentro de los estándares de la industria, pero esto aún parece demorarse; la recomendación sería para los investigadores que se enfocaran en las futuras versiones de la especificación y a permanecer atentos al estado del arte y del mercado en este campo.
- También sería un buen trabajo realizar la definición -bien sea informal ó no- de nuevos perfiles de aplicación; estos aún no han acabado de iniciar su recorrido por el mercado, y en éste sentido pueden haber buenas alternativas para “producir” perfiles en el ámbito de los microcontroladores y los sistemas empotrados, sobre todo si e habla de instrumentación, aplicaciones de reemplazo de cables y de elementos de interfaz hombre - máquina (Human Interface Device - HID).

- El trabajo investigativo puede complementarse con aplicaciones basadas en herramientas de desarrollo. Pueden iniciarse trabajos con emuladores de picoredes sobre redes LAN y luego con Kits de desarrollo que incorporen módulos Bluetooth.
- Más adelante se mencionan las herramientas que se han podido utilizar para realizar el trabajo y las que se han encontrado adicionales para qyue sean estudiadas posteriormente.
- Si se desea desarrollar aplicaciones como ejercicio de desarrollo complementarias al trabajo investigativo, se sugiere que se utilice esta monografía como una aproximación inicial a la tecnología y tomar también el modelado que se ha realizado para la implementación de la aplicación práctica adjunta a este trabajo de grado.
- Dentro del proceso de desarrollo puede surgir la necesidad de consultar valores de constantes, detalles de las negociaciones y protocolos, los cuales pueden ser consultados en la especificación 1.1 y en la bibliografía que se suministra en éste documento.