
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES
"ARAGÓN"

# "TRANSMISIÓN DE UNA RED INALÁMBRICA BLUETOOTH"

T E S I S
QUE PARA OBTENER EL TITULO DE:
INGENIERO MECÁNICO ELÉCTRICO
P R E S E N T A N:
AREA: ELÉCTRICA ELECTRÓNICA
MENDOZA BERNABÉ CARLOS OMAR
ARTURO MUÑOZ BAUTISTA


ASESOR ING. ADRIAN PAREDES ROMERO

MÉXICO 2006

*Agradecimientos:*

*Alguna vez mi padre me leyó estas palabras "el éxito no se logra con la suerte, es el resultado de un esfuerzo constante " y con estas palabras concluyo que mis padres y hermanos son la base importante de ese esfuerzo en mi vida, gracias por estar siempre...*

*Arturo Muñoz Bautista*

*Verae amicitiae sempiternae sunt*

*Gracias a todas las personas que han estado cerca, por hacer que vea las cosas desde un enfoque distinto a todo lo posible, por compartir sus sueños, sus consejos y sus derrotas que son siempre un pequeño obstáculo para ser aun mas grande.*

*Siempre se les agradece a los amigos en un apartado especial y solo por no olvidar a alguno empezare por los mejores que la vida me ah dado mis padres que me dan la fortaleza en sus palabras, a la Naye por ser la peque que me baja de las nubes, y a Beto que en la nobleza de su espíritu soñador lleno de sabiduría no se doblega.*

*Quién diría que la amistad es un regalo de la vida y mejor aun,*
*es hacerles creer que es un don con el que vamos caminando,*
*porque sé que a los amigos no se les escoge y por fortuna*
*siempre están ahí.*

*Carlos Omar Mendoza Bernabé*

*A nuestro asesor y amigo Ing. Adrián Paredes Romero quien ayudo a lograr este proceso facilitando que esta fase concluya con su experiencia y dedicación.*

*A nuestra alma mater la Universidad Nacional Autónoma de México que nos dio las herramientas y conocimientos para el desarrollo de nuestra formación académica, técnica y cultural.*

# Índice

## Introducción

1 Origenes ..... 1
2 Bluetooth SIG ..... 2
3 Cómo funciona ..... 2
4 La especificación Bluetooth ..... 4
5 Los perfiles de Bluetooth ..... 6
6 Bluetooth es adoptado por fin por la IEEE ..... 7

## Capitulo I
## Arquitectura

1 Descripción General ..... 8
1.1 Vista General ..... 8
2 Núcleo de la Arquitectura del Sistema ..... 10
2.1 Bloques del núcleo Arquitectónico ..... 13
2.1.1 Controlador de Canal ..... 13
2.1.2 Controlador de recursos L2CAP ..... 13
2.1.3 Controlador de Dispositivo ..... 13
2.1.4 Controlador de Enlace ..... 14
2.1.5 Controlador de recurso de Baseband ..... 14
2.1.6 Controlador de enlace ..... 15
2.1.7 RF ..... 15
3 Arquitectura de Transporte de Datos ..... 16
3.1 Portadores de tráfico del núcleo ..... 17
3.1.1 Tráfico de Datos Framed ..... 18
3.1.2 Tráfico de datos Unframed ..... 19
3.1.3 Confiabilidad de portadores de tráfico ..... 20
3.2 Entidades Arquitectura de Transporte ..... 21
3.2.1 La Estructura Genérica de Paquete Bluetooth ..... 22
3.3 Canales Físicos ..... 23
3.3.1 Canal Piconet Básico ..... 25
3.3.1.1 Visión General ..... 25
3.3.1.2 Características ..... 25
3.3.1.3 Topología ..... 26
3.3.1.4 Soporte de Capas ..... 26
3.3.2 Canal Piconet Adaptado ..... 26

3.3.2.1 Visión General ..... 26
3.3.3 Canal inquiry scan ..... 27
3.3.3.1 Visión Genera ..... 27
3.3.3.2 Características ..... 27
3.3.3.3 Topología ..... 28
3.3.3.4 Las capas de soporte ..... 28
3.3.4 Canal Page scan ..... 28
3.3.4.1 Visión General ..... 28
3.3.4.2 Características ..... 29
3.3.4.3 Topología ..... 29
3.3.4.4 Capas de soporte ..... 29
3.4 Enlaces Físicos ..... 30
3.4.1 Enlaces soportados por el canal físico piconet básico y adaptado ..... 30
3.4.1.1 En lace físico activo ..... 30
3.4.1.2 Enlace Físico Parked ..... 31
3.4.2 Enlaces soportados por los canales físicos de exploración ..... 32
3.5 Enlaces lógicos y transportes lógicos ..... 32
3.5.1 Casting ..... 34
3.5.2 Esquema Scheduling y Acknowledgement ..... 34
3.5.3 Clase de datos ..... 35
3.5.4 Conexión-orientada asíncrona (ACL) ..... 36
3.5.5 Conexión –orientada síncrona (SCO) ..... 37
3.5.6 Conexión -Orientada Sincronía extendida (eSCO) ..... 38
3.5.7 Slave activo broadcast (ASB) ..... 38
3.5.8 Slave de transmisión Parked (PSB) ..... 39
3.5.9 Enlaces lógicos ..... 41
3.5.10 ACL Enlace Lógico de Control (ACL - C) ..... 41
3.5.11 Enlace Lógico Asíncrono / Isócrono de Usuario (ACL-U) ..... 41
3.5.12 Enlaces Lógicos Síncrono y Síncrono/extendido de usuario (SCO-S / eSCO-S) ..... 42
3.6 Canales de L2CAP ..... 42
4 Topología de Comunicación ..... 44
4.1 Topología de PICONET ..... 44
4.2 Procedimientos de funcionamiento y Modos ..... 46
4.2.1 Procedimiento inquiry (descubrir) ..... 46
4.2.2 Procedimiento paging (conexión) ..... 47
4.2.3 Modo connected (conectado). ..... 47
4.2.4 Modo Hold ..... 48
4.2.5 Modo Sniff ..... 49
4.2.6 Estado Parked ..... 49
4.2.7 Procedimiento Role Switch ..... 49

# Capitulo II
# Core System Package
# [Controller volume]

## PARTE A
## RADIO ESPECIFICACION

1 Alcance ..... 50
2 Banda de frecuencia y rangos del canal ..... 51
3 Características del transmisor ..... 51
3.1 Características de Modulación ..... 53
3.2 SPURIOUS EMISSIONS (falsas) ..... 54
3.2.1 In-band spurious emissions ..... 54
3.3 Tolerancia de frecuencia de radio ..... 54
4 Características del Receptor ..... 55
4.1 Nivel de Sensibilidad Real ..... 55
4.2 Funcionamiento de la Interferencia ..... 55
4.3 Out-of-Band Blocking ..... 56
4.4 Características de Inter-modulación ..... 57
4.5 Máximo Nivel Utilizable ..... 58
4.6 Indicador de Fuerza de la Señal de Recepto ..... 58
4.7 Definición de la Señal de Referencia ..... 58

## PARTE B
## BASEBAND

1 Descripción General ..... 59
1.1 Reloj Bluetooth ..... 60
1.2 Dispositivo Bluetooth Addressing ..... 61
1.2.1 Direcciones Reservadas ..... 62
1.3 Códigos de Acceso ..... 62
2 Canales Físicos ..... 63
2.1 Definición del Canal Físico ..... 64
2.2 Canal Físico Piconet Básico ..... 64
2.2.1 Definición del Master-Slave ..... 64
2.2.2 Características Hopping ..... 64
2.2.3 Time Slots ..... 65

2.2.4 Relojes del Piconet ..... 65
2.2.5 Transmicion/recepcion timing ..... 66
2.2.5.1Timing del Canal Físico Piconet ..... 66
2.2.5.2 Re-sincronización del canal físico Piconet ..... 68
2.3 Canal Fisico Piconet Adaptado ..... 69
2.3.1 Características Hoppping ..... 69
2.4 Page Scan del Canal Físico ..... 69
2.4.1 Reloj estimado para Paging ..... 70
2.4.2 Características Hopping ..... 70
2.4.3 Procedimiento Paging Timing ..... 70
2.4.4 Page Response Timing ..... 71
2.5 Canal Físico Inquiry Scan ..... 73
2.5.1 Reloj para inquiry ..... 73
2.5.2 Características Hoppping ..... 73
2.5.3 Procedimiento Inquiry Timing ..... 74
2.5.4 Inquiry Response Timing ..... 74
2.6 Hop Selection ..... 75
2.6.1 Esquema General de Selection ..... 76
2.6.2 Selection kernel ..... 79
2.6.2.1 First addition operation ..... 79
2.6.2.2 XOR operation ..... 80
2.6.2.4 Second addition operation ..... 81
2.6.2.5 Register bank ..... 81
2.6.3 Adapted hop selection kernel ..... 82
2.6.3.1 Channel re-mapping function ..... 82
2.6.4 Control word ..... 83
2.6.4.1 Page Scan and Inquiry Scan Hopping Sequences ..... 85
2.6.4.2 Page Hopping Sequence ..... 85
2.6.4.3 Slave Page Response Hopping Sequence ..... 85
2.6.4.4 Master Page Response Hopping Sequence ..... 86
2.6.4.5 Inquiry hopping sequence ..... 86
2.6.4.6 Inquiry Response Hopping Sequence ..... 87
2.6.4.7 Basic and Adapted Channel Hopping Sequence ..... 87
3 Physical Links ..... 87
3.1 Link Supervision ..... 88
4 Logical Transports ..... 88
4.1 General ..... 88
4.2 Logical Transport Address (Lt_Addr) ..... 89
4.3 Synchronous Logical Transports ..... 89
4.4 Asynchronous Logical Transport ..... 90
4.5 Rutinas de Transmisión y Recepción ..... 90
4.5.1 Rutina TX ..... 90
4.5.1.1 El tráfico de ACL ..... 91
4.5.1.2 Tráfico SCO ..... 92
4.5.1.3 Mezcla del Tráfico de Datos/Voz ..... 93
4.5.1.4 Tráfico eSCO ..... 93

4.5.1.5 Los Tipos Predefinidos de Paquete ..... 94
4.5.2 Rutina RX ..... 94
4.5.3 Control de Flujo ..... 95
4.5.3.1 Control de Destino ..... 95
4.5.3.2 Control de la Fuente ..... 95
4.6 Slave Activo Transmisión del Transporte ..... 96
4.7 Slave PARKED Transmision del Transporte ..... 96
4.7.1 Dirección de Miembro PARKED (PM_ADDR) ..... 96
4.7.2 Dirección de Petición de Acceso (AR_ADDR) ..... 97
5 Enlaces Lógicos ..... 97
5.1 Control de Enlace Lógico (LC) ..... 97
5.2 Control de Enlace Lógico ACL (ACL-C) ..... 97
5.3 Usuario de Enlace Lógico ASÍNCRONO/ISOCRONO (ACL-U) ..... 98
5.3.1 Pausa del Enlace Lógico ACL-U ..... 98
5.4 Usuario de Datos Enlaces Síncronos Lógicos (SCO-S) ..... 98
5.5 Usuario Extendido de Datos Síncronos (eSCO-s) eSCO-s ..... 98
5.6 Prioridades del Enlace lógico ..... 98
6 Paquetes ..... 99
6.1 Formato General ..... 99
6.2 Ordenador de Bits ..... 99
6.3 Código de Acceso ..... 99
6.3.1Tipos de Código de Acceso ..... 100
6.3.2 Preámbulo ..... 101
6.3.3 Sincronización ..... 101
6.3.3.1 Definición de la Palabra Sincronización ..... 101
6.3.3.2 Generación de Secuencia Seudo-Aleatoria ..... 103
6.3.4 Trailer ..... 104
6.4 Header del Paquete ..... 105
6.4.1 LT_ADDR ..... 105
6.4.2 Tipo ..... 105
6.4.3 FLUJO ..... 106
6.4.4 ARQN ..... 106
6.4.5 SEQN ..... 106
6.4.6 HEC ..... 106
6.5 Tipos de Paquete ..... 107
6.5.1 Tipos Comunes de Paquete ..... 108
6.5.1.1 Paquete ID ..... 108
6.5.1.2 Paquete NULL ..... 108
6.5.1.3 Paquetes PULL ..... 108
6.5.1.4 Paquetes FHS ..... 108
6.5.1.5 Paquete DM1 ..... 110
6.5.2 Paquetes SCO ..... 111
6.5.2.1 Paquete HV1 ..... 111
6.5.2.2 Paquete HV2 ..... 111
6.5.2.3 Paquete HV3 ..... 111
6.5.2.4 Paquete DV ..... 111

6.5.3 Paquetes eSCO ..... 112
6.5.3.1 Paquete EV3 ..... 112
6.5.3.2 Paquete EV4 ..... 112
6.5.3.3 Paquete EV5 ..... 112
6.5.4 Paquetes de ACL ..... 113
6.5.4.1 Paquete DM1 ..... 113
6.5.4.2 Paquete DH1 ..... 113
6.5.4.3 Paquete DM3 ..... 113
6.5.4.4 Paquete DH3 ..... 113
6.5.4.5 Paquete DM5 ..... 113
6.5.4.6 Paquete DH5 ..... 114
6.5.4.7 Paquete AUX1 ..... 114
6.6 Formato PAYLOAD ..... 114
6.6.1 Campo de los Datos Síncronos ..... 114
6.6.2 Campo de los Datos Asíncronos ..... 115
6.7 Resumen de Paquete ..... 118
7 Proceso de Bitstream ..... 119
7.1 Verificacion de Error ..... 120
7.1.1 Generación de HEC ..... 120
7.1.2 Generación de CRC ..... 122
7.2 Datos Whitening ..... 123
7.3 Corrección del Error ..... 124
7.4 Código de FEC Rate 1/3 ..... 124
7.5 Código de FEC Rate 2/3 ..... 125
7.6 Esquema ARQ ..... 125
7.6.1 Unnumbered ARQ ..... 126
7.6.2 Retransmisión Filtrada ..... 129
7.6.2.1 inicialización SEQN a la Salida de una Nueva Conexión ..... 130
7.6.2.2 ACL y SCO Retransmisión filtrada ..... 130
7.6.2.3 Retransmision Filtrada en eSCO ..... 131
7.6.2.4 Retransmision Filtrada FHS ..... 131
7.6.2.5 Retransmision Filtrada en Paquetes sin el CRC ..... 131
7.6.3 Payloads Vacios Flushing ..... 132
7.6.4 Consideraciones del Multi-Slave ..... 132
7.6.5 Paquetes de la Transmisión ..... 133
8 Operación del Controlador de Enlace ..... 134
8.1 Apreciacion Global de Estados ..... 134
8.2 Estado STANDBY ..... 135
8.3 Establecimiento de Conexion de Subestados ..... 135
8.3.1 Subestado Page Scan ..... 136
8.3.2 Subestado Page ..... 137
8.3.3 Subestado Page Response ..... 140
8.3.3.1 Subestado Slave Response ..... 142
8.3.3.2 Subestado Master Response ..... 143

# PARTE C
# LINK MANAGER PROTOCOL (LMP)

1 Sincronización ..... 145
2 LMP PDUs ..... 145
2.1 Respuesta General ..... 145
2.2 Autenticación ..... 146
2.3 Pairing ..... 146
2.4 Cambio del Link Key ..... 147
2.5 Cambia la Llave Actual de la Conexión ..... 147
2.6 Encryption ..... 147
2.7 Clock Offset Request ..... 148
2.8 Slot Offset Request ..... 148
2.9 Timing Accuracy Information Request ..... 149
2.10 LMP Version ..... 149
2.11 Características soportadas ..... 149
2.12 Switch de Master-Slave Role ..... 150
2.13 Name Request (Solicitud de nombre) ..... 150
2.14 Detach (Separa) ..... 150
2.15 Hold Mode ..... 150
2.16 Sniff Mode ..... 151
2.17 Park Mode ..... 151
2.18 Power Control ..... 151
2.19 Channel Quality-Driven Change (entre DM and DH) ..... 152
2.20 Quality of Service ..... 152
2.21 SCO Links ..... 152
2.22 Control de Multi-Slot Packets ..... 153
2.23 Esquema Paging ..... 153
2.24 Link Supervision ..... 153
2.25 Connection Establishment ..... 153
2.26 Test Mode ..... 154
2.27 Error Handling ..... 155

# PARTE E
# Host Controller Interface (HCI)

1 Entidades Funcionales HCI ..... 157
1.1 HCI Firmware (ubicación Host Controller) ..... 158
1.2 HCI Driver (ubicación Host) ..... 158
2 Host Controller Transport Layer (ubicación Intermediate Layers) ..... 158

3 Comandos HCI ..... 158
3.1 HCI Events ..... 159
3.2 Autenticacion y Encriptación ..... 160
3.3 TESTING ..... 160
4 Flow Control ..... 160
4.1 Desconexion Behaviour ..... 160
4.2 Códigos de Error HCI ..... 160
5 Cambio de información HCI-Specific ..... 161
6 Comandos de Control de Enlace (Link Control Commands) ..... 161
6.1 Link Policy Commands ..... 161
6.2 Host Controller & Baseband Commands ..... 161
6.3 Parámetros de Información ..... 161
6.4 Parámetros de Posición ..... 162
6.5 Testing Commands ..... 162
7 Delimitacion - Bluetooth Host Controller Transport Layers ..... 162
7.1 UART Transport Layer ..... 162
7.2 RS232 Transport Layer ..... 162
7.3 USB Transport Layer ..... 162

# Capitulo III
# Core System Package
# [Host volume]

## PARTE A
## Logical Link Control and Adaptation Protocol L2CAP

1 Requisitos Funcionales L2CAP ..... 163
1.1 Protocol Multiplexing ..... 163
1.2 Segmentation & Reassembly ..... 164
1.3 Quality of Service ..... 164
1.4 Grupos ..... 164
2 Operación General de L2CAP ..... 166
2.1 Channel Identifiers (Identificador de Canal) ..... 166
2.2 Operacion entre Dispositivos ..... 165
2.3 Operación entre Capas ..... 165
2.4 Segmentation & Reassembly ..... 165
3 L2CAP State Machine ..... 166
4 Otras Características de L2CAP ..... 167
4.1 Formato de Paquete de Datos ..... 167
4.2 Señalización ..... 167
4.3 Configuración del Parámetro de Opcion ..... 167
4.4 Service Primitivos (Servicios primitivos) ..... 167

# PARTE B
# SERVICE DISCOVERY PROTOCOL (SDP)

1 Arreglo del Protocolo SDP ..... 168
1.1 Vista General ..... 168
1.2 Formato PDU ..... 169
1.3 Respuestas Parciales y Estado de Continuación ..... 169
1.4 Manejo de Error ..... 170
2 Servicios de SDP ..... 170
2.1 Registro de Servicio ..... 170
2.2 Atributo de Servicio ..... 170
2.3 Clase de Servicio ..... 170
3 Service Discovery ..... 171
3.1 Searching for Services ..... 171
3.2 Browsing for Services ..... 171
4 Representación de Datos ..... 172
4.1 Data Element header field ..... 172
4.2 Data Element data field ..... 172
5 Service Discovery Background info ..... 172
5.1 Service Discovery ..... 172
5.2 Bluetooth Service Discovery ..... 173
6 Perfil del Servicio de Descubrimiento (Aplicación) ..... 173
6.1 Descripción del perfil ..... 173
6.1.1 Introducción ..... 173
6.1.2 Pila del perfil ..... 174
6.1.3 Configuraciones y funciones ..... 174
6.1.4 Requisitos y escenarios de usuario. ..... 175
6.1.5 Principios del perfil ..... 175
6.1.6 Conformidad ..... 176
6.2 Aspectos de la Interfase de usuario ..... 176
6.2.1 Pairing ..... 176
6.2.2 Selección de modo ..... 176
6.3 Capa de Aplicación ..... 176
6.3.1 Aplicación Descubrimiento de servicio ..... 176
6.3.2 Service Primitivos Abstraction (Servicios de Abstracción Primitivos) ..... 178
6.3.3 Mapas de Sucesión de Mensaje ..... 179
6.4 Descubrimiento de servicio ..... 180
6.4.1 Un Ejemplo del Intercambio de SDP PDU ..... 180
6.5 L2CAP ..... 182
6.5.1 Tipos de canal ..... 182
6.5.2 Señalización ..... 182
6.5.3 Opciones de configuración. ..... 182
6.5.3.1 Unidad de Transmisión de máximo (MTU) ..... 182

6.5.3.2 Flush Time-out ..... 183
6.5.3.3 Calidad de Servicio (Quality o Service) ..... 183
6.5.4 Transacciones SDP y Vida de Conexión L2CAP ..... 183
6.6 Link Manager ..... 184
6.6.1 Vista General de Capacidad ..... 184
6.6.2 Comportamiento de error ..... 184
6.6.3 Link Policy ..... 184
6.7 Link Control ..... 185
6.7.1 Inquiry ..... 185
6.7.2 Inquiry Scan ..... 185
6.7.3 Paging ..... 185
6.7.4 Page Scan ..... 185
6.7.5 Error Behaviour ..... 186

# PARTE C
# GENERIC ACCESS PROFILE

1 Descripción del perfil ..... 187
1.1 Pila del perfil ..... 187
1.2 Configuración y función ..... 187
1.3 Requerimientos y escenarios del usuario ..... 188
1.4 Principios del perfil ..... 188
1.5 Conformidad ..... 188
2 Aspectos de Interface de usuario ..... 188
2.1 Representación de Parámetro Bluetooth ..... 188
2.2 Pairing ..... 189
3 Modos ..... 190
3.1 Modos de Detección (Discoverability Modes) ..... 190
3.2 Modos de Conectibilidad ..... 191
3.3 Modo Pairing ..... 192
4 Aspectos de Seguridad ..... 192
4.1 Autenticación ..... 193
4.2 Modos de seguridad ..... 193
5 Procedimientos del Modo Idle ..... 195
5.1 Inquiry general ..... 195
5.2 Inquiry limitado ..... 195
5.3 Descubrimiento del nombre ..... 196
5.4 Descubrimiento del dispositivo ..... 197
5.5 Bonding ..... 198
6 Procedimientos de Establecimiento. ..... 198
6.1 Establecimiento de Enlace ..... 199
6.2 Establecimiento de Canal ..... 199

6.3 Establecimiento de Conexión ..... 200
6.4 Establecimiento de Conexiones Adicionales ..... 200

# Capitulo IV
# Implementación de una Red Bluetooth

## PARTE A
## PERFIL OBJETIVO DE CAMBIO GENÉRICO

1 Vista General del Perfil ..... 201
1.1 Reglas/Configuraciones ..... 201
1.2 Escenarios de Perfil ..... 201
1.3 Fundamentos de Perfil ..... 201
2 Capa de Aplicación ..... 202
2.1 Vista General de Características ..... 202
2.2 Establece una Sesión Objetiva del Cambio ..... 202
2.3 Objeto Pushing de Datos ..... 203
2.4 Pulling Objeto de Datos ..... 203
3 Requisitos de Interoperabilidad de OBEX ..... 203
3.1 Operaciones de OBEX Utilizadas ..... 203
3.2 Inicialización OBEX ..... 203
3.3 Establecimiento de Sesión OBEX ..... 203
3.4 Pushing Servicio de Datos ..... 204
3.5 Pulling Servicio de Datos ..... 204
4 Requisitos de Interoperabilidad de Perfil de Puerto en Serie ..... 204
5 Requisitos Genéricos de Interoperabilidad de Perfil de Acceso ..... 204

## PARTE B
## PERFIL DEL OBJETO PUSH

1 Perfil General ..... 205
1.1 Reglas/Configuraciones ..... 205
1.2 Escenarios de Perfil ..... 205
1.3 Fundamentos de Perfil ..... 205
2. Interface de Usuario (Aspectos) ..... 206
2.1. Selección del Modo Servidor Push ..... 206
2.2. Selección de la Función de Clientes Push ..... 206

2.3. Acontecimientos de Uso de Aplicación..... 206
3. Capa de Aplicación ..... 207
3.1 Vista General..... 207
3.2 Característica del Objeto Push..... 207
3.3 Caracteristica de la Tarjeta Pull..... 207
3.4 Característica de Cambio de Tarjeta..... 208
4 Requisitos de Interoperabilidad de OBEX ..... 208
4.1 Las Operaciones utilizadas por OBEX ..... 208
4.2 Inicialización de OBEX ..... 208
4.3 Establecimiento de Sesión de OBEX ..... 208
4.4 Datos de Push..... 208
4.5 Datos de Pull ..... 208

# PARTE C
# PERFIL DE LA TRANSFERENCIA DE ARCHIVO

1 Perfil General ..... 209
1.1 Reglas/Configuraciones ..... 209
1.2 Escenarios de Perfil ..... 210
1.3 Fundamentos de Perfil ..... 210
2 Interface de Usuario (Aspectos)..... 210
2.1. La Selección del Modo de Transferencia de Archivo de los servidores ..... 210
2.2 Selección de la Función de Clientes ..... 211
2.3 Uso de Aplicación..... 211
3 Capa de Aplicación ..... 211
3.1 Vista General de Características..... 211
3.2 Fólder Browsing ..... 212
3.3 Transferencia de Objeto..... 212
3.4 Manipulación Objetiva ..... 212
4 Requisitos de Interoperabilidad de OBEX ..... 213
4.1 Las Operaciones utilizadas por OBEX ..... 213
4.2 Inicialización de OBEX ..... 213
4.3 Establecimiento de Sesión OBEX ..... 213
4.4 Browsing Folders..... 213
4.5 Objeto Pushing..... 213
4.6 Objeto Pull..... 214
4.7 Manipulación de Objeto..... 214
5 Descubrimiento de Servicio..... 214

# PARTE D
# PERFIL DE SINCRONIZACIÓN

1 Vista General del Perfil ..... 215
1.1 Reglas/Configuraciones ..... 215
1.2 Escenarios de Perfil ..... 216
1.3 Fundamentos de Perfil ..... 216
2 Comunicación de Usuarios Aspectos ..... 217
2.1 Selección de Modo ..... 217
2.2 Uso de Aplicación ..... 217
3 Capa de Aplicación ..... 217
3.1 Vista General de Características ..... 217
3.2 Característica de Sincronización ..... 218
3.3 Característica de Orden de Sincronización ..... 218
3.4 Característica Automática de Sincronización ..... 218
4 Requisitos de Interoperabilidad de OBEX ..... 219
4.1 Las Operaciones de OBEX ..... 219
4.2 Inicialización de OBEX ..... 219
4.3 Establecimiento de de la Sesión de OBEX ..... 219
4.4 Los Datos de Pushing de / Datos/Desconexión ..... 219
5 Descubrimiento de Servicio ..... 219

# PARTE E
# IMPLEMENTACION DE UNA RED BLUETOOTH

1 Introducción ..... 220
2 Creación de nuevos Sistemas, con lleva a nuevas necesidades. ..... 221
3. Entidades de la red ..... 222
3.1. Entidades de la red de localización ..... 222
3.1.1. Instalación ..... 224
3.1.2. Zonas de detección ..... 224
3.2. Protocolos de la Red Bluetooth ..... 227
3.2.1. Auto-configuración de la Red Bluetooth ..... 227
3.2.2. Protocolo de localización ..... 230
3.3.1. Escalabilidad ..... 232
3.3.2. Supervivencia ..... 235
3.3.3. Rendimiento del ciclo de inquiry ..... 236
3.3.4. Retardo de transmisión ..... 237
3.3.5. Estudio con Bluetooth ..... 239
4 Subestado park Bluetooth ..... 240

5 Servicios propuestos para la arquitectura m-Mall..... 241
5.1. Servicios iniciados por el sistema..... 241
5.1.1 Publicidad..... 241
5.1.2 Notificaciones..... 242
5.1.3 Servicios generales..... 242
5.2 Servicios Solicitados por el Usuario..... 242
5.2.1 Servicios de Búsqueda..... 242
5.2.2 Sistemas de Compra/Reserva..... 242
5.2.3 Servicios Guiado..... 242
5.3 Prototipo m-Mall..... 243

**CONCLUSIONES**..... 245

**NOMENCLATURA**..... 247

**BIBLIOGRAFÍA**..... 253

# Introducción

Bluetooth es una tecnología para conectividad inalámbrica de corto alcance entre dispositivos tales como PDAs (Personal Digital Assistance), teléfonos celulares, teclados, máquinas de fax, computadoras de escritorio y portátiles, módems, proyectores, impresoras, etc. El principal mercado es la transferencia de datos y voz entre dispositivos y computadoras personales. El enfoque de Bluetooth es similar a la tecnología de infrarrojo conocida como IrDA (Infrared Data Association). Sin embargo, Bluetooth, es una tecnología de radiofrecuencia (RF) que utiliza la banda de espectro disperso de 2.4 GHz.

Bluetooth intenta proveer significantes ventajas sobre otras tecnologías inalámbricas similares tales como IrDA, IEEE 802.11 y HomeRF, claros competidores en conexiones PC a periféricos. IrDA es una tecnología muy popular para conectar periféricos, pero es limitada severamente a conexiones de cortas distancias en rangos de un metro por la línea de vista requerida para la comunicación. Debido a que Bluetooth funciona con RF no está sujeto a tales limitaciones. Las distancia de conexión en Bluetooth puede ser de 10 metros o más dependiendo del incremento de la potencia del transmisor, pero los dispositivos no necesitan estar en línea de vista ya que las señales RF pueden atravesar paredes y otros objetos no metálicos sin ningún problema.

Bluetooth puede ser usado para aplicaciones en redes residenciales o en pequeñas oficinas, ambientes que son conocidos como WPANs (Wireless Personal Area Network). Una de las ventajas de las tecnologías inalámbricas es que evitan el problema de alambrar las paredes de las casas u oficinas.

## 1 Orígenes

La versión 1.0 de la especificación Bluetooth fue liberada en 1999, pero el desarrollo de esta tecnología empezó realmente 5 años atrás, en 1994, cuando la compañía Ericsson empezó a estudiar alternativas para comunicar los teléfonos celulares con otros dispositivos. El estudio demostró que el uso de enlaces de radio seria el más adecuado, ya que no es directivo y no necesita línea de vista; eran tan obvias estas ventajas con respecto a los enlaces vía infrarrojo que son utilizados para conectar dispositivos y teléfonos celulares. Existían muchos requerimientos para el estudio, los cuales incluían la manipulación tanto de voz como de datos, de tal manera se podrían conectar teléfonos a dispositivos de cómputo. Así es como nace la especificación de la tecnología inalámbrica conocida como Bluetooth. El origen del nombre de esta tecnología proviene de un Vikingo de origen Danés Harald Blatand (Bluetooth) quien en el siglo décimo unificó Dinamarca y Noruega. El nombre fue adoptado por Ericsson, quien espera que Bluetooth unifique las telecomunicaciones y la industria del cómputo.

## 2 Bluetooth SIG

El Bluetooth SIG (Special Interest Group) es un grupo de compañías trabajando en conjunto para promover y definir la especificación Bluetooth. Bluetooth SIG fue fundado en Febrero de 1998 por las siguientes compañías: Ericsson, Intel, IBM, Toshiba y Nokia. En Mayo de 1998, se anuncia públicamente el Bluetooth SIG y se invita a otras compañías para que se unan a éste. Fue en julio de 1999 cuando el SIG publica la versión 1.0 de la especificación Bluetooth. En diciembre de 1999, se unen otras compañías tales como Microsoft, Lucent, 3com y Motorola.

## 3 Cómo funciona

Bluetooth opera en la banda 2.4 GHz bajo la tecnología de radio conocida como espectro disperso. La banda de operación está dividida en canales de 1 MHz, a 1 mega símbolo por segundo puede obtenerse al ancho de banda máximo por canal. Con el esquema de modulación empleado, GFSK (Gaussian Frequency Shift Keying), esto equivale a 1 Mbps. Utilizando GFSK, un 1 binario representa una desviación positiva de la portadora nominal de la frecuencia, mientras que un 0 representa una desviación negativa. Después de cada paquete, ambos dispositivos re-sintonizan su radio transmisor a una frecuencia diferente, saltando de un canal a otro canal de radio; esta técnica se le conoce como espectro disperso con salto en frecuencia (FHSS, Frequency Hopping Spread Spectrum).
De esta manera, los dispositivos Bluetooth utilizan toda la banda de 2.4 GHz y si una transmisión se interfiere sobre un canal, una retransmisión siempre ocurrirá sobre un canal diferente con la esperanza de que este canal esté libre. Cada slot de tiempo tiene una duración de 625 microsegundos y generalmente los dispositivos saltan una vez por paquete, o sea, saltan cada slot, cada 3 slots o cada 5.

Como Bluetooth fue diseñado para aplicaciones móviles de poca potencia, la potencia del radio transmisor debe ser minimizada. Tres diferentes clases de niveles de potencias están definidas, las cuales proveen rangos de operación de aproximadamente 10, 20 y 100 metros: El nivel más bajo de potencia cubre 10 metros, el nivel más alto logra cubrir distancias de hasta 100 metros.

Aunado a las distancias cortas de conexión Bluetooth en materia de ancho de banda soporta hasta 780 Kbps, los cuales pueden ser utilizados para transferir unidireccional mente 721 Kbps y 57.6 Kbps en la dirección de retorno o hasta 432.6 Kbps de manera simétrica en ambas direcciones. Aunque estas velocidades están limitadas para cierto tipo de aplicaciones como video, aplicaciones como transferencia de archivos e impresión caen perfectas en tal ancho de banda.

Algunas de las principales características técnicas de Bluetooth son las siguientes:

- Los dispositivos en una picocelda comparten un canal de comunicación de datos común. El canal tiene una capacidad total de 1 Mbps. Los encabezados

y el control de llamada consumen cerca del 20% de esta capacidad; motivo por el cual el máximo caudal eficaz es de 780 Kbps.

- En los Estados Unidos y Europa, el intervalo de frecuencia de operación es de 2,400 a 2,483.5 MHz, con 79 canales de RF de 1 MHz. En la práctica, el intervalo de frecuencias es de 2,402 a 2,480 MHz. En México el intervalo de frecuencias va de 2,450 MHz a 2,485.5 MHz. En Japón, el intervalo de frecuencia es de 2,472 a 2,497 MHz con 23 canales de RF de 1 MHz.

- Un canal de datos salta aleatoriamente 1,600 veces por segundo los 79 (o 23) canales de RF.

- Cada canal está dividido en slots de tiempo de 625 microsegundos cada una.

- Una picocelda tiene un dispositivo maestro y hasta siete dispositivos esclavos. Un dispositivo maestro transmite en slots de tiempo pares, los esclavos en slots de tiempo impares.

- Los paquetes pueden tener una magnitud de hasta 5 slots de tiempo.

- Los datos en un paquete pueden ser de hasta 2,745 bits de longitud.

- Existen actualmente dos tipos de transferencia de datos entre dispositivos: Los orientados a conexión de tipo síncrono (SCO, Synchronous Connection Oriented) y los orientados a no-conexión de tipo asíncrono (ACL, Asynchronous Connectionless).

- En un piconet, puede hacer hasta tres enlaces SCO de 64,000 bits cada uno. Para evitar problemas de sincronización y colisión, los enlaces SCO utilizan slots de tiempo reservadas asignadas por la estación maestra.

- Un dispositivo maestro puede soportar hasta tres enlaces SCO con uno, dos o tres dispositivos esclavos

- Los slots no reservados para los enlaces SCO pueden ser usadas para enlaces ACL.

- Un maestro y un esclavo pueden compartir un enlace ACL

- Un enlace ACL puede ser punto-punto (maestro a esclavo) o multipunto (maestro a todos los esclavos).

- Un esclavo ACL puede sólo transmitir cuando se lo solicite un maestro

Bluetooth permite manipular simultáneamente transmisiones de voz y datos. Es capaz de soportar un canal de datos asíncrono y hasta tres canales de voz asíncronos o un canal que soporte ambos, voz y datos. La capacidad combinada

con los dispositivos del tipo "ad hoc" permite soluciones superiores para dispositivos móviles y aplicaciones de Internet. Esta combinación permite soluciones innovadoras como dispositivos manos libres para llamadas de voz, impresión a máquinas de fax y sincronización automática a PDAs, laptops y aplicaciones de libreta de direcciones de teléfonos celulares.

## 4 La especificación Bluetooth

La especificación de Bluetooth cubre desde el transceptor de radio hasta varias interfaces de protocolos basados tanto en hardware como en software. Algunos elementos clave y protocolos de la arquitectura de Bluetooth son descritos a continuación. Control de enlace: el hardware del control de enlace controla la transmisión y recepción de radio así como el procesamiento de la señal digital requerida para el protocolo de baseband. Sus funciones incluyen establecimientos de conexiones, soporte para enlaces asíncronos (datos) y síncronos (voz), corrección de error y autentificación. El microcódigo del Link Manager desempeña funciones a bajo nivel para el establecimiento de enlaces, autentificación y configuración de los enlaces.

Topología de la red: los dispositivos Bluetooth son generalmente organizados en grupos de 2 a 8 llamados piconet o picoredes, consistente de un dispositivo maestro y uno o más dispositivos esclavos. Un dispositivo puede pertenecer a más de una picocelda y comportarse como un esclavo en ambas o un maestro en una picocelda y como esclavo en otra.

Como Bluetooth opera en una banda de uso libre conocida como ISM (Industrial, Scientific, and Medical) donde otros dispositivos de uso común la utilizan como es el caso de puertas de cocheras, teléfonos inalámbricos, hornos de microondas, sólo por nombrar algunos. Para que los dispositivos Bluetooth puedan coexistir y operar confiablemente con los otros dispositivos, cada picocelda es sincronizada a una frecuencia específica del patrón de salto por frecuencia. Este patrón, que salta a 1,600 frecuencias diferentes por segundo, es único para una picocelda en particular. Cada "salto" de frecuencia es una slot de tiempo durante la cual los paquetes de datos son transferidos. Un paquete puede abarcar hasta 5 slots de tiempo, en la cual la frecuencia permanece constante durante la duración de esa transferencia.

Si los dispositivos van a saltar a las nuevas frecuencias después de cada paquete, ellos deben ponerse de acuerdo en la secuencia de las frecuencias que utilizarán. Como los dispositivos Bluetooth operan en 2 modos: como maestro y como esclavo. Si el maestro asigna la secuencia de salto de frecuencia. Los esclavos sincronizan al dispositivo maestro en tiempo y frecuencia seguido de la secuencia de salto del dispositivo maestro.

Enlaces baseband: La baseband de Bluetooth provee canales de transmisión para

voz y datos y es capaz de soportar un enlace asíncrono de datos y hasta tres enlaces de voz asíncronos (o un enlace soportando ambos). Los enlaces orientados a conexión síncronos (SCO) son típicamente empleados para transmisiones de voz. Esos enlaces son conexiones simétricas punto a punto que reservan slots de tiempo para garantizar la transmisión a tiempo. Al dispositivo esclavo siempre se le permitirá responder durante el slot de tiempo inmediatamente seguido de una transmisión tipo SCO del maestro. Un dispositivo maestro puede soportar hasta tres enlaces SCO a uno o varios esclavos, pero un solo esclavo puede soportar sólo enlaces SCO para diferentes dispositivos maestros. Los paquetes SCO nunca son retransmitidos.

Los enlaces orientados a no-conexión (ACL, Asynchronous Connectionless) son típicamente empleados para transmisión de datos. Las transmisiones sobre estos enlaces son establecidas en base por slot (en slots no reservadas para enlaces SCO). Los enlaces ACL soportan transferencias punto-multipunto de datos asíncronos como síncronos. Después de una transmisión ACL del maestro, sólo el dispositivo esclavo direccionado puede responder durante el siguiente slot de tiempo o si el dispositivo no está direccionado, los paquetes son considerados como mensajes difundidos (broadcast). La mayoría de los enlaces ACL incluyen retransmisión de paquetes.

Link Manager: La máquina de estado de baseband es controlada por el administrador de enlaces. Este microcódigo provee el control del enlace basado en hardware para configuración, seguridad y control de enlaces. Sus capacidades incluyen autentificación y servicios de seguridad, monitoreo de calidad de servicio y control del estado de baseband. El Link Managerse comunica con los demás utilizando el protocolo LMP (Link Management Protocol), el cual utiliza los servicios básicos baseband. Los paquetes LMP, los cuales son enviados sobre los enlaces ACL, son diferenciados de los paquetes L2CAP (Logical Link Control and Adaptation Protocol) por un bit en el encabezado del ACL. Ellos son siempre enviados como paquetes de un slot y una prioridad alta que los paquetes L2CAP. Esto ayuda a asegurar la integridad del enlace bajo una alta demanda de tráfico.

El Host controller interface, HCI (Controlador de Interface del host): Por encima del Link manager se encuentra el HCI. Este protocolo basado en hardware es usado para aislar el baseband de Bluetooth y el Link Manager de un protocolo de transporte tal como el RS-232 o USB (Universal Serial Bus). Esto permite una interface estándar para el hardware Bluetooth. Un manejador de dispositivos HCI en el host es usado para interactuar una aplicación Bluetooth con el protocolo de transporte. Actualmente existen tres mecanismos de transporte soportados: USB, RS-232 y el UART (Universal Asynchronous Receiver-Transmitter). Utilizando HCI, una aplicación Bluetooth puede acceder al hardware de Bluetooth sin el conocimiento de la capa de transporte u otros detalles de implementación del hardware.

Protocolos basados en software: el resto de los protocolos son implementados en

software. La capa más baja de L2CAP provee la interface con el Link Manager y permite la interoperatibilidad entre dispositivos Bluetooth. Provee la multicanalización de protocolos, lo cual permite el soporte de otros protocolos de más alto nivel tales como TCP/IP. El L2CAP opera sobre un enlace del tipo ACL en baseband y provee enlaces punto-multipunto para transferencias síncronas como asíncronas. L2CAP provee servicios a los protocolos de los niveles superiores al transmitir paquetes de datos sobre los canales L2CAP. Existen tres tipos de canales L2CAP: canales bidireccionales que transportan comandos; canales orientados a conexión para conexiones punto-punto y bidireccionales; y canales unidireccionales orientados a no-conexión que soporten conexiones punto-multipunto, permitiendo que una entidad local L2CAP sea conectada a un grupo de dispositivos remotos.

Varios protocolos interactúan con la capa de enlace L2CAP tales como SDP y RFCOMM. El protocolo SDP (Service Discovery Protocol) provee un medio para determinar que servicios Bluetooth están disponibles en un dispositivo particular. Un dispositivo Bluetooth puede actuar como un cliente SDP solicitando servicios o como un servidor SDP proveyendo servicios, o ambos. Un simple dispositivo Bluetooth tendrá no más de un servidor SDP, pero puede actuar como un cliente para más de un dispositivo remoto. El protocolo SDP provee acceso sólo a información acerca de servicios, la utilización de esos servicios deberá ser proveído por otro protocolo. RFCOMM es un protocolo de transporte que provee transferencia de datos serial. Una entidad de emulación de puertos es usada para mapear la comunicación de la interface de la programación de aplicaciones (API, Applications Programming Interface) a los servicios de RFCOMM, permitiendo que el software opere en un dispositivo Bluetooth. TCS (Telephony Control Protocol Specification), un protocolo para aplicaciones de telefonía es proveído para control de llamadas de voz y datos a través de señalización. La señalización tanto para punto-punto y punto-multipunto son soportados utilizando los canales L2CAP, la voz o los datos son transferidos directamente desde la baseband sobre los enlaces SCO.

Bluetooth también soporta el protocolo de sesión conocido como IrOBEX (IrDA Object Exchange Protocol), definido por IrDA. Este protocolo puede operar sobre las capas de transporte, incluyendo RFCOMM y TCP/IP. Para dispositivos Bluetooth, solo OBEX orientado a conexión es soportado. Tres perfiles de aplicación han sido desarrollados usando OBEX. Estos incluyen funcionalidades de sincronización para directorios telefónicos, calendarios, mensajes, etc.; funcionalidades de transferencia de archivos y Object Push para soporte de tarjetas de presentación.

## 5 Los perfiles de Bluetooth

Los perfiles son una parte muy importante en la tecnología Bluetooth. Los perfiles le proveen a Bluetooth una significante ventaja sobre las otras tecnologías. Los perfiles, definidos por Bluetooth SIG, tienen la intención de asegurar la

interoperatibilidad entre las aplicaciones Bluetooth y los dispositivos de diferentes fabricantes. Estos perfiles definen los roles y capacidades para aplicaciones especificas. Diferentes perfiles pueden abarcar diferentes capas y protocolos y para diferentes grados de seguridad. Además de los requerimientos de interoperatibilidad, los protocolos pueden definir servicios requeridos para otras aplicaciones o para usuarios finales.

Todos los dispositivos Bluetooth deberán soportar el perfil de acceso genérico (Generic Access Profile) como mínimo. Este perfil en particular define el descubrimiento o encuentro de dispositivos, procedimientos de conexión y procedimientos para varios niveles de seguridad. También se describen algunos requerimientos de interface al usuario. Otro perfil universal, aunque no es requerido, es el perfil de acceso a descubrimiento de servicios (Service Discovery Access Profile), el cual define los protocolos y parámetros asociados requeridos para acceder a los perfiles. Un número de perfiles han sido definidos incluyendo TCS, RFCOMM y OBEX. Algunos de estos requieren la implementación de otros, y todos ellos requieren la implementación de perfiles genéricos.

## 6 Bluetooth es adoptado por fin por la IEEE

Durante la ultima semana del mes de marzo del 2002 la IEEE aprobó finalmente el estándar IEEE 802.15.1 compatible totalmente con la tecnología Bluetooth v1.1. En este estándar se definen las especificaciones de la capa física y MAC (medium access control) para las redes WPANs. El nuevo estándar permitirá una mayor validez y soporte en el mercado de las especificaciones de Bluetooth, además es un recurso adicional para aquellos que implementen dispositivos basados en esta tecnología. Anteriormente a la estandarización, dispositivos Bluetooth no podían coexistir con los dispositivos basados en IEEE 802.11b debido a que ambos se interferían entre sí. Este esfuerzo entre Wi-Fi y Bluetooth es conocido como Blue802 y permitirá la operación simultánea de estos dos protocolos inalámbricos.

# Capitulo I
# Arquitectura

## 1 Descripción General

La tecnología Bluetooth inalámbrica es un sistema de corto alcance de comunicaciones que intenta reemplazar el cable(s) conectando dispositivos electrónicos portátiles y/o fijos. Las características de energía, son de baja potencia y bajo costo. Muchas características de la especificación centrales son opcionales, que permite la diferenciación del producto. El sistema central Bluetooth consiste en un transceptor de RF, una baseband, y en el protocol stack. El sistema ofrece los servicios que permiten la conexión de dispositivos y el cambio de una variedad de clases de datos entre estos dispositivos. Este capítulo proporciona especificaciones y una vista general de la arquitectura del sistema Bluetooth, topologías de comunicación y las características del transporte de datos. El texto en este capítulo de la especificación se debe tratar como de información y utilizado como fondo para ponerlo en contexto.

## 1.1 Vista General de Operación

El RF Bluetooth (capa física) operar en la banda libre de la banda ISM en 2,4 GHz. El sistema emplea un transceptor de salto de la frecuencia para combatir interferencia y atenuación y proporciona muchas portadoras FHSS. La operación de RF utiliza configuración, modulación binaria de FM para aminorar la complejidad de transceptor. El símbolo de la tasa es 1 Megasymbol por segundo (Ms/S) soportando una velocidad e transferencia de 1 Megabit por segundo (Mb/S).

Durante la operación típica de radio bluetooth, un canal físico es compartido por un grupo de dispositivos que son sincronizados a un reloj y frecuencia de salto comunes. Un dispositivo proporciona la referencia de sincronización y es conocido como el master (maestro). Todos los demás dispositivos son conocidos como slaves (slave). Un grupo de dispositivos sincronizados de este modo forman un piconet. Esto es la forma fundamental de comunicación en la tecnología inalámbrica Bluetooth.

Los dispositivos en un piconet utilizan una frecuencia de salto específica, el cual es determinado algorítmicamente por ciertos campos en la dirección Bluetooth y el reloj del master. El salto (hopping) básico es un orden seudo-aleatorio de las 79 frecuencias en la banda ISM. El modelo de salto se puede adaptar para excluir una parte de las frecuencias que son utilizadas por dispositivos de interferencia. La adaptive hopping technique hace la coexistencia de Bluetooth con sistemas, ISM estáticos (no-hopping) cuando éstos son co-localizados.

El canal físico es subdividido en unidades de tiempo conocidas como slots. Los datos se transmiten entre dispositivos Bluetooth en paquetes, eso se posiciona en estos slots. Cuándo las circunstancias lo permiten, varios slots consecutivos pueden ser asignados a un solo paquete. La frecuencia de salto toma lugar entre la transmisión o la recepción de paquetes. La tecnología Bluetooth provee un efecto de transmisión full dúplex a través del uso del esquema Time-División Duplex (TDD).

Encima del canal físico hay una capa de enlaces y canales y de protocolos de control asociados. La jerarquía de canales y conexiones del canal físico ascendente es el canal físico, enlace físico, transporte lógico, la conexión lógica y canal L2CAP. Estos se detallan más a lo largo de este trabajo pero son introducidos aquí para ayudar a la comprensión del resto de esta sección. Dentro de un canal físico, una conexión física se forma entre dos dispositivos cualquiera que transmiten paquetes entre ellos en cualquier dirección. En un canal físico piconet hay restricciones en cuáles los dispositivos pueden formar una conexión física. Hay una conexión física entre cada slave y el master. Las conexiones físicas no se forman directamente entre slaves en un piconet.

El enlace físico se utiliza como un transporte para uno o más enlaces lógicos que soporta unicast síncrono, tráfico asíncrono e isócrono, y transmisión de tráfico. El tráfico en enlaces lógicos se multiplexan en la conexión física ocupando slots asignados por una función programada en el director de recursos. Un protocolo de control para baseband y las capas físicas se lleva sobre los enlaces lógicos además de los datos de usuario. Esto es el Link Manager Protocol (Protocolo de Director de Conexión LMP).

Los dispositivos que son activados en un piconet tiene predefinido un transporte lógico conexión-orientada asíncrona que se utiliza para transportar la señalización del protocolo LMP. Para las razones históricas esto es conocido como el transporte lógico ACL. El predefinido transporte lógico ACL es el único que se crea siempre que un dispositivo se une a un piconet.

El transporte lógico adicional se puede crear para transportar flujos síncronos de datos cuando esto se requiere. La función del Link Manager utiliza LMP para controlar la operación de dispositivos en el piconet y provee los servicios para manejar las capas arquitectónicas más bajas (las capas de radio y baseband). El protocolo LMP es portado en el predefinido transporte lógico ACL y la transmisión del transporte lógico.

Encima de la capa baseband, la capa L2CAP provee una abstracción de canal-basado en aplicaciones y servicios. Esto cumple con la segmentación y nuevo montaje de los datos de aplicación y multiplexión y de-multiplexión de múltiples canales sobre un enlace lógico compartido. L2CAP tiene un protocolo de canal de control que es portado sobre predefinido transporte lógico ACL. Los datos de la aplicación sometidos al protocolo L2CAP se pueden ser portados en algún enlace lógico que soporte el protocolo L2CAP**.**

# 2 Núcleo de la Arquitectura del Sistema.

El núcleo del sistema Bluetooth cubre las cuatro capas más bajas y asocia los protocolos definidos por la especificación Bluetooth tanto como un servicio común, protocolo de capa, Service Protocolo Discovery (PSD) y los requisitos generales del perfil son especificados en el Perfil de Acceso Genérico (GAP). Una aplicación Bluetooth completa requiere un número de servicio adicional y una capa más alta que son definidos en la especificación Bluetooth, pero no son descritos aquí.
El núcleo de la Arquitectura del Sistema es indicada en la Figura 2.1 excepto para PSD no es mostrado con claridad.

La figura que 2.1 indica las cuatro capas más bajas, cada una se asocia con el Protocolo de comunicación. El más bajo de las tres capas es a veces agrupado a un Subsistema conocido como el controlador Bluetooth. Ésta es puesta comúnmente en práctica, involucra una interfaz de comunicación física usual entre el Bluetooth Controlador y el resto del sistema Bluetooth incluyendo L2CAP, servicio y capas más altas (conocido como el anfitrión Bluetooth.) Aunque esta interfaz es opcional la arquitectura esta diseñada para permitir su existencia y características.

La especificación Bluetooth permite la interoperabilidad entre Sistemas Bluetooth independientes definiendo los mensajes de protocolo intercambiados entre capas equivalentes, y también la interoperabilidad entre Subsistemas Bluetooth independientes definiendo una interfaz común entre Controladores Bluetooth y Bluetooth hosts.

Varios bloques funcionales son indicados y el path de servicios y datos entre éstos. Los bloques funcionales indicados en la diagrama son informativos; en general la especificación Bluetooth no define los detalles de las implementaciones excepto de donde esto es requerido para la interoperabilidad. Por lo tanto, el bloque funcional de la Figura 2.1 es demostrar para ayudar a la descripción del comportamiento del sistema. Una puesta en práctica podría ser diferente del sistema mostrado en Figura 2.1

Las interacciones usuales son definidas para toda operación entre-dispositivo, donde los dispositivos Bluetooth cambian el protocolo de señalización de acuerdo con la especificación Bluetooth. Los protocolos de sistema de núcleo de Bluetooth son el protocolo de Radio (RF), el protocolo Link Control (LC), el protocolo Link Manager (LM) y el protocolo Logical Link Control and Adaptation (L2CAP), todo esto es completamente definidos en las siguientes partes de las especificaciones Bluetooth. Además el protocolo Service Discovery Protocol (PSD) es un protocolo de capa de servicio, requerida por todas las aplicaciones Bluetooth.

El núcleo del Sistema de Bluetooth brinda servicios a través de un número de punto de acceso de servicio que son mostrados en la diagrama como elipses. Ése servicio constan de un elemento primitivo que controlan el núcleo del sistema

Bluetooth. Los servicios pueden ser divididos en tres tipos. Hay servicios de control de dispositivo que modifican el comportamiento y los modos de un dispositivo Bluetooth, el control de transporte tiende a crear, modificar y liberar a portadores de tráfico (canales y enlaces), y servicios de datos que son usados para someter datos para transmisión sobre portadores de tráfico. Esto es común para considerar que los dos primeros pertenecen al plano-C y el último corresponde al plano-U.

Una interfaz de servicio para el controlador Bluetooth del sub-sistema es definido tal que el controlador Bluetooth puede ser considerado una parte usual. En esta configuración el controlador Bluetooth opera las tres capas más bajas y la capa L2CAP es contenida con el resto de la aplicación Bluetooth en el sistema del Host. La interfaz usual es llamada Host to Controller Interface (HCI) y su punto de acceso de servicio son representados antes de las elipses sobre el borde superior del Subsistema del controlador Bluetooth en la Figura 2.1. La puesta en práctica de esta interfaz de servicio usual es opcional.

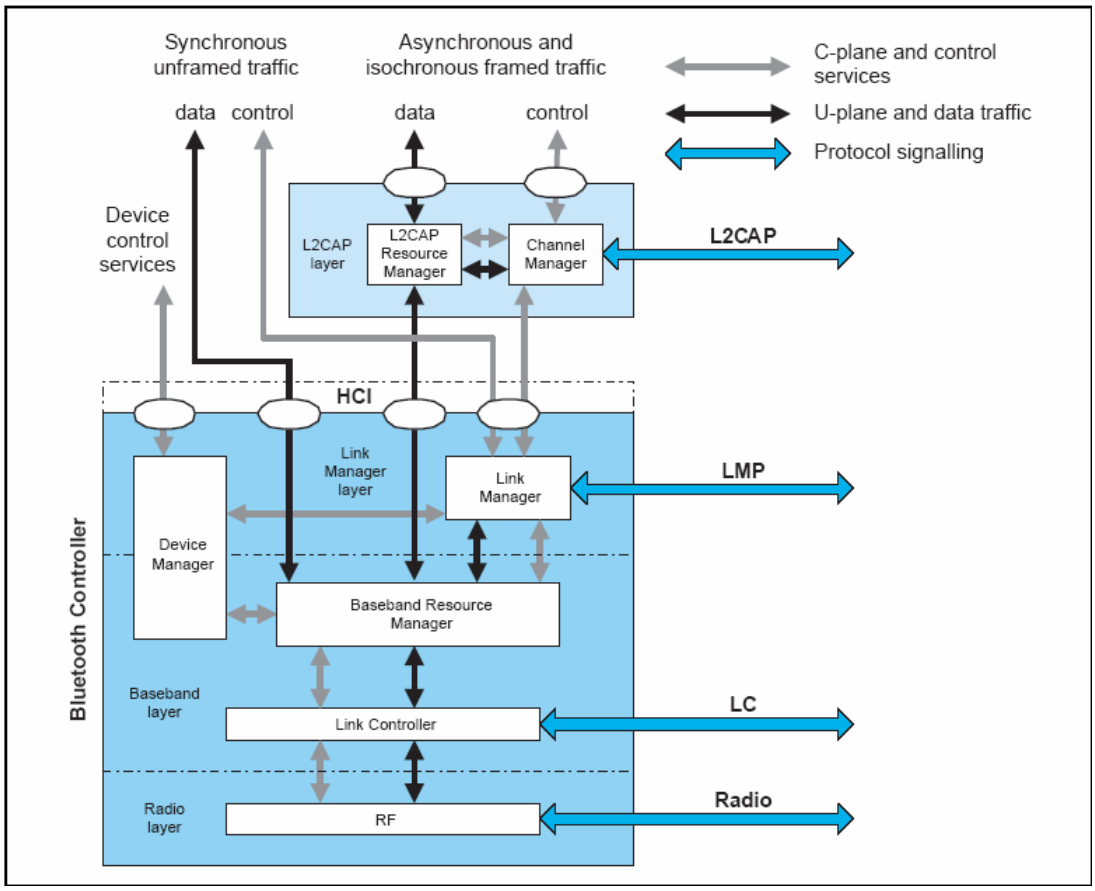



*Figura 2.1: El núcleo de la Arquitectura del Sistema de Bluetooth*

Cuando la arquitectura de Bluetooth es definida con la posibilidad de separar el Host y el controlador que se comunican a través de un HCI, varias suposiciones generales se hacen. El director de Bluetooth es asumido para limitar datos en la capacidad del buffer en comparación con el Host. Por lo tanto en la capa L2CAP se espera llevar a cabo alguna administración sencilla del recurso al someterse L2CAP PDUs al controlador para el transporte de un dispositivo semejante. Esto incluye la segmentación de L2CAP SDUs dentro de PDUs más controlables y luego la fragmentación de PDUs respecto al principio y paquetes de continuación de un tamaño apropiado para el buffer del controlador, y dirección del uso de este, para asegurar disponibilidad para canales con calidad de servicio (QoS).

La capa Baseband provee el protocolo de ARQ básico (Automatic Repeat Request) en Bluetooth. La capa L2CAP puede opcionalmente proveer una detección de error adicional y retransmisión al L2CAP PDUs. Esta característica es recomendada para aplicaciones con requerimiento de una probabilidad baja de errores inadvertidos en los usuarios de datos. Otra característica opcional de L2CAP es window-based que es un control de circulación que puede ser usado para controlar la asignación del buffer en el dispositivo de recepción. Ambas características opcionales aumentan el rendimiento QoS en ciertos escenarios.

Aunque estas suposiciones no pueden ser requeridas para Bluetooth arraigando la puesta en práctica que combinan todas las capas en un solo sistema, el uso general arquitectónico y los modelos de QoS son definidos con estas suposiciones en mente, en efecto a un mínimo común denominador.

Automáticamente la prueba de conformidad de las implementaciones del núcleo del sistema Bluetooth es requerido. Esto es conseguido permitiendo que el examinador controle la puesta en práctica a través de la interfaz de RF, que es común a todos los sistemas Bluetooth, y a través de la Interfaz de Control de Prueba (TCI), que es solamente requerido para la conformidad de la prueba.

El examinador usa intercambios con la Implemetantion Under Test (IUT) atraves de la interfaz de RF asegurando las respuestas correctas para los pedidos de los dispositivos de control remoto. El examinador controla el IUT a través del TCI para causar que el IUT cree intercambios por la interfaz de RF así éstos pueden también ser verificados de conformidad.

Los TCI usan diferentes conjuntos de comandos (la interfaz de servicio) para la prueba de cada capa arquitectónica y protocolo. Un subconjunto del conjunto de comandos HCI es usado como la interfaz del servicio de TCI para cada uno de las capas y los protocolos dentro del Subsistema del controlador de Bluetooth. Una interfaz del servicio distinta es usada para la prueba de la capa de L2CAP y el protocolo. Como una interfaz del servicio de L2CAP no es definida en la especificación del núcleo de Bluetooth, que es definido por separado en la interfaz de Control de prueba. La puesta en práctica de la interfaz de servicio L2CAP es solamente requerida para la prueba de conformidad.

## 2.1 Bloques del núcleo Arquitectónico

Esta sección describe la función y la responsabilidad de cada uno de los bloques mostrados en la figura 2.1. Una implementación no es requerida para seguir la arquitectura descrita arriba, sin embargo cada implementación se ajustará a las especificaciones de protocolo que se describen en las partes siguientes de la Especificación Bluetooth, y los aspectos conductuales de las implementaciones del sistema para dar una idea general y especificar en las partes siguientes de la Especificación Bluetooth.

### 2.1.1 Controlador de Canal

El controlador de canal es responsable de crear, dirigir y destruir canales de L2CAP para el transporte de protocolos de servicio y aplicaciones de flujo de datos. El controlador de canal usa el protocolo de L2CAP con el que interactúa con un controlador de canal (semejante) en un dispositivo remoto para crear estos canales de L2CAP y conectar sus puntos finales a las entidades apropiadas. El controlador de canal interactúa con su controlador de enlace local para crear nuevos enlaces lógicos (si es necesario) y para configurar estos enlaces para los que suministrar la calidad requerida del servicio, y para el tipo de datos que es transportado.

### 2.1.2 Controlador de recursos L2CAP

El bloque controlador de recurso de L2CAP es responsable de llevar la orden de sumisión de los fragmentos de PDU al baseband y algún programa relativo entre canales para asegurar a los canales de L2CAP con los compromisos de QoS y no negar el acceso para el canal físico debido al agotamiento de los recursos del Controlador Bluetooth. Esto es requerido porque el modelo arquitectónico no asume que el controlador de Bluetooth tiene un buffer ilimitado, o que el HCI es un conducto de Ancho de banda infinito.

El controlador de recurso L2CAP puede además controlar el tráfico conforme a las aplicaciones que son sometidas a L2CAP SDUs dentro de los límites y ajustes del escenario QoS. En general Bluetooth es un modelo de transporte que asume formalmente aplicaciones, y no se define cómo una implementación que esta esperando reparar el problema.

### 2.1.3 Controlador de Dispositivo

El controlador de dispositivo está en el bloque funcional baseband esto controla el funcionamiento general del dispositivo Bluetooth. Es responsable de toda la operación del sistema Bluetooth esto no esta directamente relacionado con

transporte de datos, tal como que inquiring para la presencia de otros dispositivos Bluetooth cercanos, conectando otros dispositivos Bluetooth, o hacer que el dispositivo local Bluetooth sea habilitado o conectado por otros dispositivos.

El controlador de dispositivo pide el acceso para el medio de transporte del controlador de recurso baseband para llevar sus funciones.

El director de dispositivo también controla el funcionamiento del dispositivo local implicado por un número de comandos de HCI, como dirigir el dispositivo local nombrado, almacenar algunas llaves de enlace y otra funcionalidad.

### 2.1.4 Controlador de Enlace

El controlador de enlace es responsable de la creación, modificación y lanzamiento de enlaces lógicos (y, si requiere, su transporte lógico asociado), también como la actualización de los parámetros relacionados con los enlaces físicos entre dispositivos. El controlador de enlace ejecuta esto, comunicándose con el controlador de enlace en el Dispositivo remoto Bluetooth que usan el protocolo de dirección de enlace (LMP.)

El protocolo (LMP) admite la creación de nuevos enlaces lógicos y transportes lógicos entre dispositivos cuando se requiere, también como el control general del enlace y transporta atributos como, él permitir la encriptación en el transporte lógico, la adaptación en el poder de transmisión del enlace físico, o los ajustes de QoS para un enlace lógico.

### 2.1.5 Controlador de recurso de Baseband

El controlador de recurso de baseband es responsable de todo el acceso al medio de radio. Tiene dos funciones principales. En su núcleo esta un programador de horarios que concede el tiempo de estar sobre los canales físicos a todas las entidades que han negociado un contrato de acceso. La otra función principal es negociar contratos de acceso con éstas entidades. Un contrato de acceso un compromiso eficaz de conseguir un determinado QoS con el que es exigido para suministrar a un usuario la aplicación con un rendimiento esperado.

El contrato de acceso y la función de planificación deben tomar en cuenta cualquier comportamiento, esto requiere el uso del radio Bluetooth. Estas inclusiones (por ejemplo) el intercambio normal de los datos entre dispositivos conectados sobre enlaces lógicos, y transportes lógicos, tanto el uso del medio de radio como llevar las inquiries, hacer conexiones, son descubiertos o conectados, para tomar las lecturas de las transportadoras inusuales durante el uso de la frecuencia adaptable del modo de salto.

En algunos casos los programadores de un enlace lógico, resulta en cambiar a un canal físico diferente de los ya usados. Esto puede ser (por ejemplo) debido a la participación en un scatternet, una función inquiry periódica, o page scan. Cuando los canales físicos no tienen el time slot alineados, entonces el controlador de recursos realinea el tiempo entre slots en los canales físico originales y el slot en el nuevo canal físico. En algunos casos los slot serán alineados naturalmente debido al mismo reloj del dispositivo, siendo usado como una referencia de ambos canales físicos.

### 2.1.6 Controlador de enlace

El controlador de enlace es responsable de la codificación y la decodificación de los paquetes payload Bluetooth de datos y parámetros relacionados con el canal físico, transporte lógico y enlace lógico.

El controlador de enlace lleva la señalización del protocolo de control de enlace (en la conjunción con la función programada del controlador de recurso), que al ser usado para comunicar el flujo de control y acknowledgemen, y la retransmisión de señales requeridas. La interpretación de estas señales es una característica del transporte lógico, se asocia con el paquete de baseband. Interpretación y control de la señalización del control de enlace es normalmente relacionada con los recursos del programa del controlador.

### 2.1.7 RF

El bloque de RF es responsable de transmitir y recibir paquetes de información sobre el canal físico. Un camino de control entre el baseband y el boque de RF admite al bloque de baseband para controlar en tiempo y frecuencia la portadora del bloque de RF. El bloque de RF transforma un flujo de datos, desde el canal físico y baseband respecto a los formatos requeridos.

# 3 Arquitectura de Transporte de Datos

El sistema de transporte de datos de Bluetooth sigue una arquitectura en capas. Esta descripción del sistema Bluetooth describe las capas de transporte del núcleo Bluetooth incluyendo los canales de L2CAP. Todos los modos Bluetooth de operaciones siguen la misma arquitectura de transporte genérica, que es indicado en la Figura 3.1

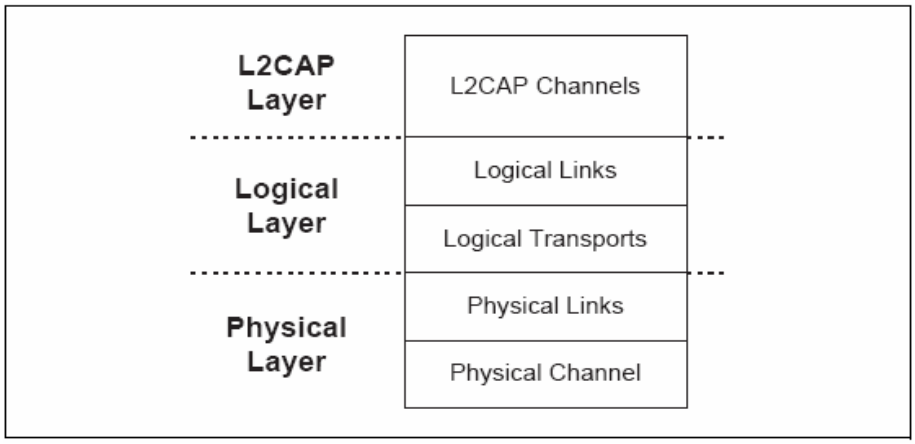



*Figura 3.1: Arquitectura genérica Bluetooth de transporte de datos*

Para las razones de eficiencia y legado, la arquitectura de transporte Bluetooth incluye una subdivisión de la capa lógica, distinguiendo entre enlaces lógicos y transportes lógicos. Esta subdivisión provee en general (y comprende comúnmente) el concepto de un enlace lógico que provee un transporte independiente entre dos o más dispositivos. La sub-capa de transporte lógico es requerida para describir la interdependencia entre algunos de los tipos de enlace lógicos (principalmente por razones del comportamiento de legado.)

La especificación Bluetooth 1.1 describió el ACL y los enlaces de SCO como enlaces físicos. Con la adición de la extensión SCO (eSCO) para una futura expansión, es bueno considerar esto como tipos de transporte lógicos, cuyo mayor propósito es concentrar una mayor exactitud. Sin embargo, no son tan independientes como se desea, atribuible a su uso compartido de recursos como el LT_ADDR y el procedimiento ARQ. Por lo tanto la arquitectura es incapaz de representar estos transportes lógicos con una sola capa de transporte.

La capa de transporte lógico adicional va de alguna manera para describir este funcionamiento.

## 3.1 Portadores de tráfico del núcleo

El núcleo Bluetooth para el cual el sistema suministra varios portadores de tráfico, para el transporte del servicio de protocolo y aplicación de datos. Son mostrados en la figura 3.2 (para la facilitar la representación, esto es mostrado, la capa más alta a la izquierda y a la derecha las capas más bajas).

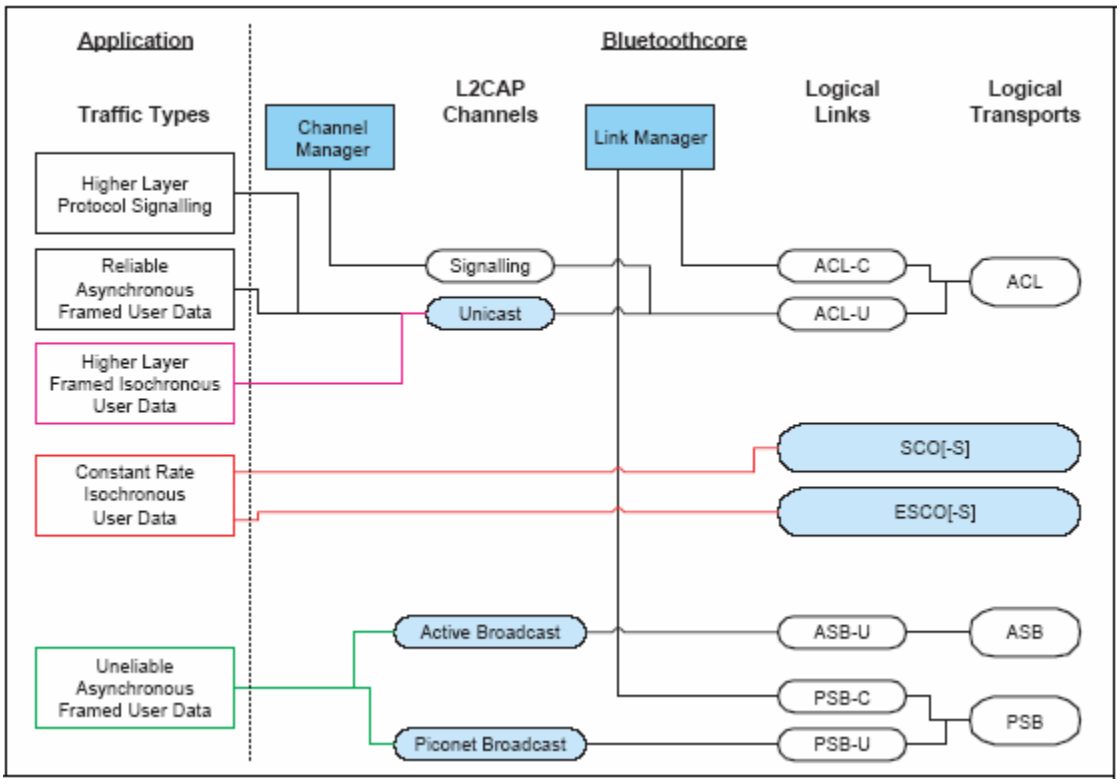



*Figura 3.2: portadores de tráfico de Bluetooth*

Los portadores del tráfico de núcleo que están disponible para las aplicaciones son mostrados en la figura 3.2 como rectángulos redondeados y sombreados. La arquitectura de las capas esta definida para suministrar estos servicios que son descritos en la parte 2. Un número de tipos de tráfico de datos son indicados sobre la izquierda del diagrama, vinculada a los portadores de tráfico que son típicamente idóneo para transportar este tipo de tráfico de datos.

Los enlaces lógicos son nombrados usando los nombres del transporte lógico asociado y un sufijo que demuestra el tipo de los datos que es transportado. (C para el control de enlaces que llevan los mensajes de LMP, U para llevar enlaces a los usuarios de datos de L2CAP (L2CAP PDUs) y S para enlaces de flujo que no llevan formato síncrono o datos isócronos.) Es común que el sufijo pueda ser retirado del enlace lógico sin presentar la ambigüedad, por lo tanto, una referencia de incumplimiento del transporte lógico de ACL puede ser resuelto para representar el enlace de lógico de ACL-C en casos donde el protocolo de LMP

esta siendo tratado, o el enlace lógico de la ACL-U, cuando la capa de L2CAP esta siendo tratada.

La correlación de los tipos de aplicación de tráfico, para portadores de tráfico de núcleo Bluetooth en la figura 3.2 se basa en combinar las características de tráfico con las características del portador. Es recomendado usar esta correlación cuando proveen el método más natural y eficiente de transportar los datos y estas características se dan.

Sin embargo, una aplicación (o una implementación del núcleo del Sistema Bluetooth) puede decidir usar un portador de tráfico diferente, o una correlación diferente parar lograr un resultado similar. Por ejemplo, en un piconet con solamente un slave, la master puede decidir transportar la transmisión L2CAP sobre el enlace lógico ACL-U un poco mejor que en el ASB-U o enlaces lógicos PSB-U. Esto será más eficiente probablemente dentro de los términos del ancho de banda (si la calidad del canal físico no es demasiado bajo.) El uso de transporte alternativo para éstos se muestra en la Fig. 3.2, es solamente aceptable si las características de los tipos de aplicación de trafico están conservadas.

La Figura 3.2 indica varios tipos de aplicación de tráfico. Éstos solían clasificar los tipos de datos que pueden ser presentados en el núcleo del sistema Bluetooth. El tipo de tráfico de datos original, no podría ser el mismo como el tipo que accedió al sistema de núcleo Bluetooth si en un proceso, interviniendo, lo modifica. Por ejemplo, los datos de video son generado en un rate constante excepto en un proceso intermedio de codificación se puede modificar esta variable, por ejemplo, para la codificación de MPEG4. Para los propósitos el núcleo del Sistema de Bluetooth, solamente la característica de presentar los datos de interés.

### 3.1.1 Tráfico de Datos Framed

La capa L2CAP suministra los servicios de un transporte frame-orientado para usuarios asíncronos e isócronos de datos. La aplicación presenta los datos a este servicio en la dimensión-variable de frame (hasta un máximo negociado para el canal) y estos frames son repartidos en la misma forma, en la aplicación correspondiente del dispositivo remoto. No hay ningún requisito para la aplicación de insertar información adicional framing en los datos, aunque lo puede hacer si esto es requerido (framing es invisible al núcleo del sistema Bluetooth.)

Los canales de L2CAP conexión -orientada pueden ser creados para el transporte unicast (Punto - a - punto) de datos entre dos dispositivos Bluetooth. Un canal L2CAP sin conexión existe para transmitir datos. En el caso de la topología piconet, el master del dispositivo es siempre el origen de los datos de transmisión y el dispositivo slave es el receptor. El tráfico sobre el canal de L2CAP de transmisión es unidireccional. Los canales L2CAP unicast podrían ser unidireccionales o bidireccionales.

Los canales de L2CAP tienen un ajuste de QoS asociado, que define las restricciones en la entrega de los frame de datos. Estos ajustes de QoS pueden ser usados para demostrar (por ejemplo) que los datos son isócronos, y por lo tanto tienen una vida limitada después de ser invalidados, o que los datos deben ser repartidos dentro de un punto de tiempo dado, o esos datos son seguros y deben ser repartidos sin error, sin embargo toma mucho tiempo.

El controlador del canal L2CAP es responsable de organizar al transporte del canal L2CAP frame data sobre un enlace de lógico baseband apropiado, es posible multiplexar este en el enlace lógico baseband con otro canal L2CAP con características similares.

### 3.1.2 Tráfico de datos Unframed

Si la aplicación no requiere la entrega de los datos en frames, posiblemente es porque incluye in-stream framing, o porque los datos son puro flujo, entonces puede evitar el uso de canales de L2CAP y hacer el uso directo de un enlace lógico baseband.

El núcleo del sistema Bluetooth soporta el transporte directo de la aplicación de datos que es isócrono y de un rate constante (bit-rate, o frame-rate para pre-framed data), usando un enlace lógico SCO-s o eSCO-s. Estos enlaces lógicos reservan el ancho de banda de canal físico y suministra el transporte de rate constante, cierra el reloj de piconet. Los datos son transportados en paquetes de tamaño fijo a intervalos fijos con ambos parámetros se negocia durante el establecimiento del canal.

Los enlaces eSCO proveen una elección más grande de bit - rates y también proveen una confiabilidad más grande, usando retransmisión limitada en caso de error. Los transportes lógicos SCO y eSCO no respaldan enlaces lógicos multiplexados o no promueven separar en capas dentro del núcleo Bluetooth. Una aplicación puede elegir la capa de un número de flujos dentro del flujo SCO/eSCO enviado, con tal de que el flujo sometido sea, o tenga la apariencia de ser un flujo de rate constante.

La aplicación elige el tipo de enlace lógico más apropiado de estos, disponible en la baseband, lo crea y configura para transportar el flujo de datos, y lo da a conocer cuando termina. (La aplicación normalmente usa también un frame canal unicast de L2CAP que transporta su información de plano-C a la aplicación semejante sobre el dispositivo remoto.)

Si la aplicación de datos es isócrona y de una variable rate, que entonces puede ser solamente llevado por el canal de unicast de L2CAP, y por lo tanto será tratado como framed de datos.

### 3.1.3 Confiabilidad de portadores de tráfico

Bluetooth es un sistema de comunicaciones inalámbrico. En ambientes de RF malos, este sistema debe ser considerado intrínsecamente poco fiable. Para contrarrestar esto el sistema provee niveles de protección en cada capa. El paquete header de baseband usa la corrección de errores (FEC.), permite codificar la corrección de errores por el receptor y un error header verificado (HEC) para detectar los errores restantes después de la corrección. Ciertos tipos de paquete Baseband incluyen FEC para los payload. Además, algunos tipos de paquete Baseband incluyen una verificación por redundancia cíclica (CRC).

En transporte lógico ACL los resultados del algoritmo de detección de error son usados para impulsar el protocolo repetición automática (ARQ). Este provee una seguridad mejorada, por la retransmisión de paquetes que no pasan en el algoritmo de verificación de errores de recepción. Es posible modificar este esquema para respaldar paquetes en estado latente-susceptibles, descartando una transmisión de paquetes sin éxito en el transmisor si la vida útil del paquete ha expirado. El enlaces eSCO usa una versión modificada de este esquema para mejorar la confiabilidad permitiendo un limitado número de retransmisiones.

El resultado de la confiabilidad ganada por el esquema ARQ es solamente tan confiable como la habilidad del HEC y CRC para codificar y detectar los errores. En muchos casos esto es suficiente, sin embargo ha sido demostrado que para tipos de paquete más largos la probabilidad de un error inadvertido es demasiado alto para soportar aplicaciones típicas, especialmente aquellos con una gran cantidad de datos que son transferidos.

La capa de L2CAP provee un nivel adicional de control de error que esta diseñado para detectar los errores inadvertidos ocasionales en la capa baseband y la retransmisión de los datos afectados. Esto provee el nivel de confiabilidad requerida por típicas aplicaciones Bluetooth.

Los enlaces broadcast no tienen ninguna ruta de realimentación, y son incapaces de usar el esquema ARQ (aunque el receptor todavía puede detectar los errores en paquetes recibidos.) En lugar de que cada paquete sea transmitido varias veces en la espera de que el receptor sea capaz de recibir al menos una de las copias con éxito. A pesar de este enfoque todavía no tiene ninguna garantía de recibir con éxito y así, estos enlaces son considerados poco fiables.

En resumen, si un enlace o canal son calificado de seguro esto significa que el receptor es capaz de detectar los errores en los paquetes recibidos y pedir retransmisión hasta que sean eliminados. Debido a que el sistema de detección de error usa algunos errores residuales (indetectados) que todavía podrían estar en los datos recibidos. Para canales de L2CAP el nivel de éstos es comparable a otros sistemas de comunicación, aunque para enlaces lógicos el nivel de error residual es algo más alto.

El transmisor puede quitar paquetes de la fila de transmisión tal que el receptor no recibe todos los paquetes de la secuencia. Si esto ocurre la detección de los paquetes faltantes es delegada a la capa de L2CAP.

Sobre un enlace poco fiable, en el que, el receptor es capaz de detectar los errores recibido en los paquetes, no puede solicitar retransmisión. Los paquetes pasados por el receptor pueden no tener errores, pero no garantiza que todos paquetes de la secuencia sean recibidos. Por lo tanto el enlace es considerado básicamente poco fiable.

Hay usos limitados para tales enlaces, y estos usos son normalmente dependientes de la repetición interrumpida de los datos de las capas más altas cuando esto es valido. Los enlaces de flujo tienen una característica de confiabilidad, en algún lugar, entre un enlace seguro y uno poco fiable, dependiendo de las condiciones operativas en curso.

## 3.2 Entidades Arquitectura de Transporte

Las entidades de arquitectura del transporte Bluetooth son mostradas en Figura 3.3 y son descritas de la capa más baja hacia arriba, en las secciones siguientes.

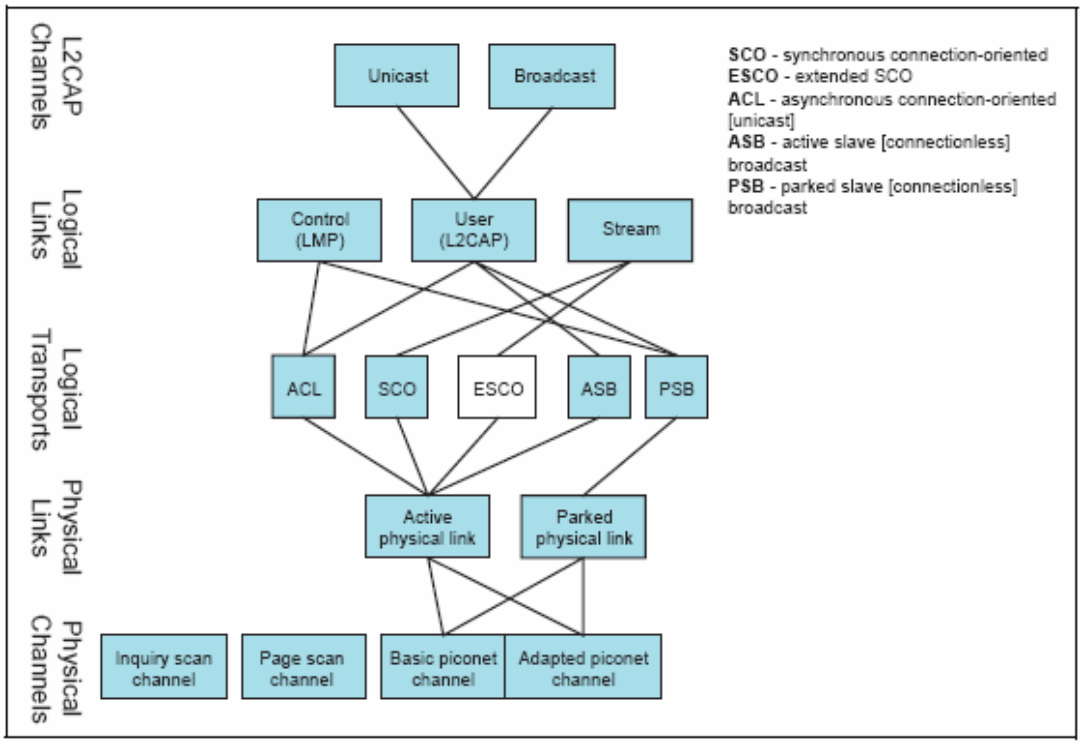



*Figura 3.3: Visión general de entidades de la arquitectura de transporte y jerarquía.*

### 3.2.1 La Estructura Genérica de Paquete Bluetooth

La estructura general del paquete en el que se refleja las capas arquitectónicas encontradas en el Sistema Bluetooth. La estructura del paquete es diseñada para el uso óptimo en operación normal. Este es mostrado en la Figura 3.4

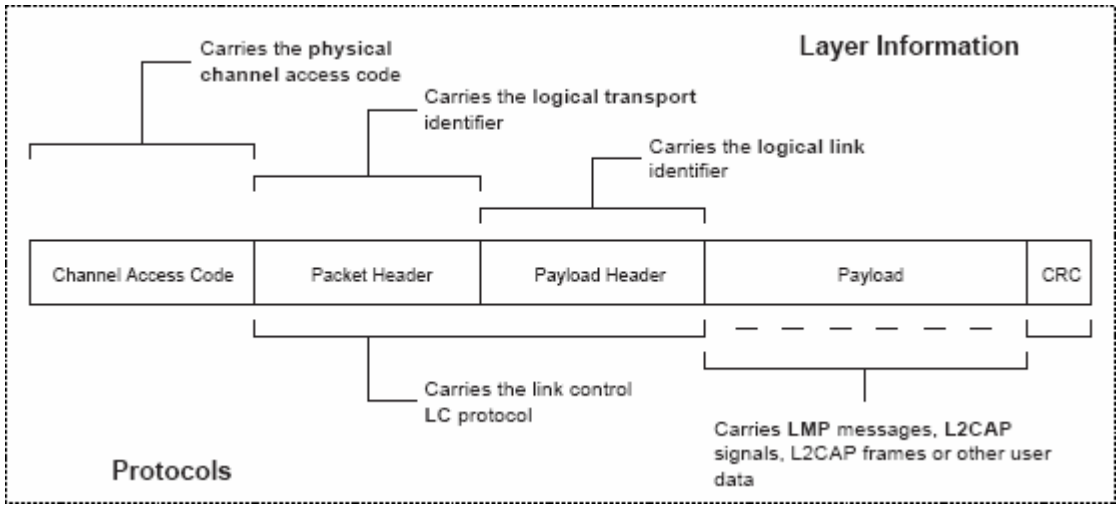



*Figura 3.4: la estructura de paquete de Bluetooth*

Normalmente los paquetes solamente incluyen los campos que son necesarios para representan las capas requeridas por la transacción. Por lo tanto, pedir una simple exploración sobre el canal físico no crea ni requiere un enlace lógico o capa alta y por lo tanto consta solamente del código de acceso del canal (se asocia con el canal físico.) La comunicación general dentro de un piconet es con paquetes, esto incluye todos de los campos, como todas las capas arquitectónicas son usadas.

Todos los paquetes incluyen el código de acceso de canal. Esto se usa para identificar las comunicaciones sobre un canal físico especial, y excluir o hacer caso omiso de paquetes con un canal físico diferente, eso pasa para usar la misma transportadora de RF en proximidades físicas.

No hay campo directo dentro de la estructura del paquete Bluetooth, esto representa o contiene la información relacionada en los enlaces físicos. Esta información esta implicada dentro de la dirección de transporte lógico (LT_ADDR) portadora del header del paquete.

La mayoría de los paquetes incluyen un paquete header. El paquete header siempre esta presente en la transmisión de paquetes en los canales físicos que soportan enlaces físicos, transporte lógico y enlaces lógicos. La portadora del paquete header LT_ADDR, que es usado por cada dispositivo receptor para

determinar si el paquete esta en el dispositivo de dirección y se usa para enrutar el paquete interiormente.

El paquete header también trae una parte del protocolo link control (LC) que se opera por el transporte lógico (excepto por ACL y transporte SCO que operan un protocolo LC compartido en cualquier transporte lógico.)

El payload header está presente en todos los paquetes del transporte lógico, este soporta múltiples enlaces lógicos. El payload header incluye un campo identificador de enlace lógico usado para enrutar el payload, y un campo que indica la longitud del payload. Algunos paquete también incluyen uno CRC después del paquete payload, este es usado para detectar mas errores en los paquetes recibidos.

El paquete payload es usado para transportar los datos de usuario. La interpretación de estos datos están en función del transporte lógico y los identificadores de enlace lógicos. Para el transporte lógico ACL, los mensajes de Link Manager Protocol (LMP) y la señalización L2CAP son transportados en el paquete payload, junto con datos generales del usuario de aplicaciones. Para SCO y transporte lógico eSCO el payload contiene los datos de usuario para el enlace lógico.

## 3.3 Canales Físicos

La capa arquitectónica más baja en el sistema Bluetooth es el canal físico. Varias clases de canales físicos son definidos. Todos los canales físicos Bluetooth se caracterizan por una frecuencia RF combinada con los parámetros temporales y restringidos por consideraciones espaciales. Para los canales físicos, piconet básico y adaptado la frecuencia de salto es utilizada para cambiar la frecuencia periódicamente para reducir los efectos de la interferencia y para razones reguladoras.

Dos dispositivos de Bluetooth usan un canal físico compartido para la comunicación. Para conseguir esto sus transceptores necesitan sintonizar la misma frecuencia RF al mismo tiempo, y necesitan estar dentro de un rango nominal de uno a otro.

Dado esto el número de portadoras de RF es limitado y muchos dispositivos Bluetooth pueden estar operando por separado dentro de la misma área espacial y temporal en que hay una enorme probabilidad de tener dos dispositivos Bluetooth independientes con sus transceptores sintonizados a la misma portadora de RF, resultando en una colisión del canal físico. Para mitigar los efectos no deseados de esta colisión cada transmisión en un canal físico empieza con un código de acceso que es usado como un código correlación por el dispositivo sintonizador del canal físico. Este código de acceso del canal es una propiedad del canal físico.

El código de acceso está siempre presente al principio de cada paquete transmitido.

Cuatro canales físicos Bluetooth son definidos. Cada uno es optimizado y usado para un propósito diferente. Dos de estos canales físicos (el canal de piconet básico y el canal de piconet adaptado) son usados para la comunicación entre dispositivos conectados y son relacionados con un piconet específico. El otro canal físico es usado para descubrir dispositivos Bluetooth (el canal inquiry scan) y para conectar dispositivos de Bluetooth (el canal de page scan.)

Un dispositivo Bluetooth puede usar uno de estos canales físicos en cualquier tiempo dado. Para soportar múltiples operaciones simultáneas, los dispositivos utilizan multiplexión por división de tiempo entre las los canales. De este modo un dispositivo Bluetooth puede aparecer para operar simultáneamente varios piconets, también como estar descubierto y conectado.

Siempre que un dispositivo Bluetooth es sincronizado en tiempo, frecuencia y código de acceso de un canal físico, se dice estar "Conectado" con este canal (Si está activamente involucrado en las comunicaciones sobre el canal.). Las especificaciones Bluetooth suponen que un dispositivo es solamente capaz de conectarse a un canal físico en cualquier momento. Los dispositivos avanzados pueden ser capaces de sintonizar simultáneamente a más de un canal físico, pero la especificación no supone que esto es posible.

## 3.3.1 Canal Piconet Básico

### 3.3.1.1 Visión General

El canal de piconet básico es usado para la comunicación entre dispositivos conectados durante la operación normal.

### 3.3.1.2 Características

El canal de piconet básico es caracterizado por una secuencia pseudo- aleatoria de saltos a través del canal de RF. La secuencia de saltos es única para el Piconet y esta determinada por el dispositivo de dirección Bluetooth del master. La fase en la secuencia de salto es determinada por el reloj Bluetooth del master. Todos los dispositivos Bluetooth que participan en el piconet son: el tiempo – el salto – y la sincronización del canal.

El canal es dividido en time slots donde cada slot corresponde a la frecuencias salto de RF. Los saltos consecutivos corresponden a diferentes frecuencias de salto de RF. Los time slots son numerados de acuerdo con el reloj Bluetooth del

piconet master. Los paquetes son transmitidos por dispositivos Bluetooth que participan en el piconet alineado, para empezar en un slot limite. Cada paquete empieza con el código de acceso del canal, que es obtenido del dispositivo de dirección Bluetooth del piconet.

En el canal piconet básico el master controla el acceso para el canal. El master empieza la transmisión solo en time slots de numero par. Los paquetes transmitidos por la master son alineados con el slot start y definen el tiempo del piconet. Los paquetes transmitidos por el master pueden ocupar hasta cinco time slots dependiendo del tipo de paquete.

Cada transmisión master, es un paquete que lleva la información sobre uno de los transportes lógicos. Los dispositivos slave podrían transmitir la respuesta sobre el canal físico. Las características de la respuesta son definidas por el transporte lógico que es una dirección.

Por ejemplo en el transporte lógico conexión -orientada asíncrona la dirección del dispositivo slave responde transmitiendo un paquete que contiene la información para el mismo transporte lógico que es alineado nominalmente con el próximo (numero impar) Slot start. Tal paquete puede habitar hasta cinco time slots, dependiendo del tipo de paquete. En un transporte lógico de transmisión ningún slave es admitido para responder.

Una característica especial del canal físico básico piconet es el uso de algunos slots reservados para transmitir un beacon train. El beacon train es solamente usado si el canal físico piconet ha colocado slaves conectados con él. En esta situación el master transmite un paquete reservado de beacon train slots (estos paquetes son usados por el slave para resincronizar al canal físico piconet.) El master puede transmitir paquetes desde cualquier transporte lógico en estos slots, siempre que allá una transmisión de arranque en cada uno de los slots. En el caso dónde hay información desde transporte lógico parked slave broadcast (PSB) que esta transmitiendo, además esto es transmitido en los beacon train slot y toma la prioridad sobre cualquier otro transporte lógico.

### 3.3.1.3 Topología

Un canal de piconet básico puede ser compartido por cualquier número de dispositivos Bluetooth, limitado solamente por los recursos disponibles sobre el dispositivo master piconet. Solamente un dispositivo es el piconet master, los demás son slaves de piconet. Toda la comunicación está entre los dispositivos master y slave. No hay comunicación directa entre el dispositivo slave sobre el canal piconet.

Sin embargo, hay una limitación sobre el número de transportes lógicos que pueden ser soportados dentro de un piconet. Esto representa que aunque no hay

límite teórico del número de dispositivos Bluetooth que comparten un canal hay un límite del número de estos dispositivos que pueden estar activamente involucrados en cambiar datos con el master.

### 3.3.1.4 Soporte de Capas

El canal básico piconet soporta varios enlaces físicos, transporte lógico, enlaces lógicos y canales L2CAP usados para propósitos generales de comunicación.

## 3.3.2 Canal Piconet Adaptado

### 3.3.2.1 Visión General

El canal de piconet adaptado es diferente del canal de piconet básico en dos formas:

Primero, las frecuencias en las cuales el slave transmite es la misma frecuencia de transmisión que el master anterior. En otras palabras la frecuencia no es recomputada entre el master y el siguiente paquete slave. La segunda, en la cuál es diferente el canal adaptado piconet al canal básico piconet, es que el tipo adaptado puede estar basado en menos de las 79 frecuencias. Un número de las frecuencias puede ser excluido del patrón de salto por ser marcado como "Sin usar". El resto de las 79 frecuencias son incluidas. Las dos secuencias son las mismas excepto cuando el fundamento de la secuencia pseudo- aleatorio de salto debería haber seleccionado una frecuencia sin usar y es reemplazada con una alternativa de elección del conjunto usado.

Porque el canal de piconet adaptado usa el mismo tiempo y código de acceso que el canal de piconet básico, los dos canales son a menudo coincidentes. Este provee un beneficio premeditado, así, permite a los slaves de cualquier canal de piconet, básico o adaptado, ajustar su sincronización al master. La topología y el soporte de capas del canal físico piconet adaptado son idénticas al canal físico piconet básico.

## 3.3.3 Canal inquiry scan

### 3.3.3.1 Visión General

Para que un dispositivo pueda ser descubierto se usa un canal de inquiry scan. Un dispositivo (descubridor) atiende las peticiones inquiry en el canal inquiry scan y envía las respuestas a estos. En el orden que un dispositivo descubre a otro,

repite (hops) a través de todas las posibles frecuencias del canal de inquiry scan. En un modo pseudo- aleatorio que, envía una petición inquiry sobre cada frecuencia y esta atento a cualquier respuesta.

### 3.3.3.2 Características

Los canales inquiry scan siguen un modelo hopping más lento y usan un código de acceso. Para distinguir ocasionalmente la misma frecuencia de radio por dos dispositivos colocados que usan canales físicos diferentes.

El código de acceso usado sobre el canal de inquiry scan es tomado de códigos de acceso de inquiry que son compartidos por los todos dispositivos Bluetooth. Un código de acceso es usado para inquiry generales, y un número de códigos de acceso adicionales es reservado para las inquiry limitadas. Cada uno de los dispositivos tiene acceso a un número diferente de Canales de inquiry scan. Como todos estos canales comparten un modelo idéntico de hopping, un dispositivo puede habitar más de una canal de inquiry scan simultáneamente, si es capaz de tener una correlación con el código de acceso.

Un dispositivo que usa un inquiry scan se queda pasivo hasta que recibe un mensaje inquiry sobre el canal de otro dispositivo Bluetooth. Este es identificado por el código de acceso apropiado de inquiry. El dispositivo inquiry scanning sigue el procedimiento de inquiry al que devolverá luego una respuesta a el dispositivo inquiring.

En el orden para que un dispositivo descubra a otros dispositivos Bluetooth utilizan el canal inquiry scan de estos dispositivos para envíar pedidos de inquiry. Cuando no tiene previos conocimientos de los dispositivos que descubre, no puede saber las características exactas del canal de inquiry scan.

El dispositivo aprovecha el hecho de que canales de inquiry scan tienen una cantidad reducida de frecuencias hop y una rate más lento de salto. El dispositivo de inquiring transmite pedidos de inquiry sobre cada una de las frecuencias hop de inquiry scan Y está listo a una respuesta de inquiry. Esto hecho es uno rate más rápido, permitiendo al inquiring el cubrir todas las frecuencias de inquiry scan en un periodo de tiempo corto razonable.

### 3.3.3.3 Topología

Los dispositivos Inquiring y Discoverable usan un intercambio simple de paquetes, el cual desempeña la función inquiring. La topología constituida durante esta transacción es una simple y transitoria conexión punto - a - punto.

### 3.3.3.4 Las capas de soporte

Durante el intercambio de paquetes entre un dispositivo Inquiring y discoverable puede ser considerado, que un enlace físico temporal exista entre éstos dispositivos. Sin embargo, el concepto es muy irrelevante cuando no tiene representación física pero es solamente implicado por la transacción breve entre los dispositivos. Mas no las capas arquitectónicas están consideradas para ser soporte.

## 3.3.4 Canal Page scan

### 3.3.4.1 Visión General

Un dispositivo Connectable (que esta preparado para aceptar conexiones), lo hace usando un canal Page scan. Un dispositivo Connectable está atento a pedidos page requests en su canal Page scan y entra en una secuencia de intercambios con este dispositivo. En el orden que un dispositivo se conecta a otro, repite (hops) a través de todas las frecuencias del canal Page scan en un modo pseudo-aleatorio, transmitiendo una página que pide la frecuencia y esta atento a cualquier respuesta sobre cada uno.

### 3.3.4.2 Características

El canal de Page scan usa un código de acceso obtenido del escanear la dirección del dispositivo Bluetooth de un dispositivo address para identificar las comunicaciones en el canal. El canal Page scan usa un hopping rate más lento que el hop rate de los canales piconet básico y adaptado. El uso del algoritmo de selección hop usa el dispositivo del Reloj Bluetooth del dispositivo scanning como una entrada.

Un dispositivo que usa su canal Page scan se queda pasivo hasta que recibe una Page Request de otro dispositivo Bluetooth. Esto es identificado por el código de acceso Page scan. Los dos dispositivos perseguirán el procedimiento page para entonces formar una conexión. Luego de una conclusión próspera del procedimiento page ambos dispositivos cambian el canal de piconet básico que es caracterizado por tener el dispositivo paging como master.

En el orden que un dispositivo se conecta a otro dispositivo Bluetooth que usa el canal Page scan del dispositivo target para envíar page requests. Si el dispositivo paging no conoce la fase del dispositivo target del canal Page scan, en consecuencia este no conoce la frecuencia hop en curso del dispositivo target. El dispositivo paging transmite page requests a cada uno de las frecuencias hop de Page scan y está atento a un page response. Esto es hecho en un hop rate más

rápido, permitiendo al dispositivo paging cubrir todas las frecuencias Page scan en un periodo de tiempo corto razonable.

El dispositivo paging podría tener un poco de conocimientos del dispositivo target del Reloj Bluetooth (indicado durante una transacción previa inquiry entre los dos dispositivos, o como consecuencia de una participación previa en un piconet con el dispositivo), en este caso puede pronosticar la fase del dispositivo target del canal Page scan. Esta información se puede usar para optimizar la sincronización de paging y el proceso Page scanning y acelerar la formación de la conexión.

### 3.3.4.3 Topología

Los dispositivos Paging y Connectable usan un cambio simple de paquetes para cumplir la función de paging. La topología constituida durante esta transacción es una simple y transitoria conexión punto - a - punto.

### 3.3.4.4 Capas de soporte

Durante el intercambio de paquetes entre dispositivos paging y connectable este puede ser considerado como un enlace físico temporal entre éstos dispositivos. Sin embargo, el concepto es muy irrelevante cuando no tiene representación física pero es solamente implicado por la transacción breve entre los dispositivos. Mas no las capas arquitectónicas están consideradas para ser soporte.

## 3.4 Enlaces Físicos

Un enlace físico representa una conexión baseband entre dispositivos Bluetooth.
Un enlace físico es relacionado exactamente con un canal físico siempre (aunque un canal físico podría soportar más que un enlace físico.)

Dentro del sistema Bluetooth un enlace físico es un concepto virtualmente que no tiene la representación dentro de la estructura de un paquete transmitido. El código de acceso del campo de paquete, junto con el reloj y la dirección del dispositivo master Bluetooth es usado para identificar un canal físico. Sin embargo no hay parte subsecuentes del paquete que identifica el enlace físico directamente. En lugar del enlace físico puede ser identificado por asociación con el transporte lógico, como cada transporte lógico es solamente recibido sobre un enlace físico.

Algunos tipos de enlace físicos tienen propiedades que pueden ser modificadas. Un ejemplo de esto, es el poder de transmisión para el enlace. Los otros tipos de enlace físicos no tienen tales propiedades. En el caso de enlaces físicos con las propiedades modificadas el protocolo de LM es usado para adaptar estas

propiedades. Como el protocolo de LM es soportado en la capa más alta (por un enlace lógico) el enlace físico apropiado es identificado por la implicación del enlace lógico que transporta la señalización de LM.

En la situación donde una transmisión se transmite sobre varias conexiones físicas diferentes, los parámetros de transmisión entonces se escogen para ser convenientes a todas las conexiones físicas.

### 3.4.1 Enlaces soportados por el canal físico piconet básico y adaptado

Los canales físicos piconet básicos y adaptado soportan un enlace físico el cuál puede ser activo o estacionario (parked). El enlace físico es un enlace de punto - a - punto entre el master y un slave. Está siempre presente cuando el slave es sincronizado dentro del Piconet.

#### 3.4.1.1 En lace físico activo

El enlace físico entre un dispositivo master y un slave está activo si existe un default ACL del transporte lógico entre los dispositivos. Los enlaces físicos activos no tienen directamente identificación de ellos mismos, pero son identificados por asociación con el default ACL del ID de transporte lógico con el que hay una correspondencia uno a uno.

Un enlace físico activo tiene la propiedad asociada del poder de radio-transmisión en cada dirección. Las transmisiones del dispositivo slave están siempre dirigidas sobre el enlace físico activo al master, y usa la potencia de transmisión que es una propiedad de este enlace en el slave en dirección al master. Las transmisiones del master pueden ser dirigidas sobre un solo enlace físico activo (a un slave específico) o sobre un número de enlaces físicos (a un grupo de slaves en el piconet.) En el caso de la transmisión punto – a – punto el master usa la potencia de transmisión apropiada para el enlace físico en cuestión. (En el caso de las transmisiones punto – a – multipunto el master usa una potencia de transmisión apropiada para la colocación del dispositivo addressed.)

Los enlaces físicos activos pueden ser puestos en hold o sniff mode. El efecto de éstos modos es para modificar los períodos cuando el enlace físico esta activo y puede portar trafico.

El transporte lógico que ha definido las características programadas no son afectadas por estos modos y continúan de acuerdo con su programación predeterminada. El default ACL del transporte lógico y otros enlaces con características programadas indefinidas están sujetos al modo del enlace físico activo.

### 3.4.1.2 Enlace Físico Parked

El enlace físico entre un master y un dispositivo slave es Parked cuándo el slave queda sincronizado en el piconet, pero no tiene default ACL del transporte lógico. Tal slave también esta puesto en parked. Un beacon train es usado para proveer la sincronización, regular a todos los slaves parked conectados con el canal físico piconet. Un slave parked broadcast (PSB) transporte lógico es usado para permitir la comunicación de un subconjunto de LMP señalización y transmisión L2CAP al slave parked. El transporte lógico PSB es estrechamente relacionado con el beacon train.

Un slave parked (su enlace activo es cambiado a un enlace parked) es usado en el procedimiento park. El master no permite colocar a un slave que tiene algún usuario creando un transporte lógico soportado por el enlace físico. Estos transportes lógicos son primero removidos, y cualquier canal de L2CAP que esta basado en estos transportes lógicos también serán removidos. La transmisión del transporte lógico y del default ACL del transporte lógico no es considerada como usuarios creados y no son explícitamente removidos. Cuándo el enlace activo reemplaza con un enlace Parked el default ACL del transporte lógico es retirado implícitamente. Los enlaces lógicos soportados y el canal L2CAP se quedan en existencia, pero están suspendidos. No es posible usar estos enlaces y los canales L2CAP para transportar señalización o datos, mientras el enlace activo está ausente.

Un slave parked puede ponerse activo usando el procedimiento unpark. Este procedimiento es requerido por el slave en una ventana de acceso e iniciado por el master. Siguiendo del procedimiento unpark el enlace físico parked es cambiado a un enlace físico activo y el default ACL del transporte lógico es recreado. Los canales de L2CAP que fueron suspendidos durante el procedimiento park más reciente son relacionados con el nuevo default ACL del transporte lógico y se convierte en activo otra vez.

Las conexiones estacionadas no sostienen el control del poder de la radio, como no hay retroalimentación del slave parked al piconet master, este puede ser utilizado para recibir la fuerza de la señal en el slave o para que el master para mida la fuerza recibida de la señal del slave. Las transmisiones se llevan a cabo a potencia nominal en el enlace parked

Los enlaces parked usan el mismo canal físico como su enlace activo asociado. Si un Master dirige un piconet que contiene a slaves parked que usan el canal físico piconet básico y además usan el canal físico piconet adaptado entonces debe crear un slave parked broadcast del transporte lógico (y transporte asociado) para cada uno de estos canales físicos. Un slave parked puede usar los períodos inactivos del slave parked broadcast del transporte lógico para salvar potencia, o este puede llevar las actividades en otro canal físico desvinculado del piconet, dentro del cual esta parked.

### 3.4.2 Enlaces soportados por los canales físicos de exploración

En el caso de inquiry scan y el canal page scan el enlace físico existe por un relativo tiempo corto y no puede ser controlado o modificado de alguna manera.

.

## 3.5 Enlaces lógicos y transportes lógicos

Una variedad de enlaces lógicos son validos para soportar diferentes aplicaciones en transporte de datos requeridos. Cada enlace lógico es relacionado con un transporte lógico el cuál tiene varias características. Estas características incluyen el control de circulación, mecanismos de acuse de recibo/repetición, la secuencia de numeración y el funcionamiento del programa. El transporte lógico puede llevar diferentes clases de enlaces lógicos (dependiendo del tipo de transporte lógico). En el caso de algunos enlaces lógicos Bluetooth 1.1 éstos son multiplexados en el mismo transporte lógico.

El transporte lógico puede ser llevado por enlaces físicos activos sobre cualquier canal físico piconet adaptado o básico.

La identificación del transporte lógico y la señalización en tiempo real (control de enlace) son portadas en el header del paquete, y para algunos enlaces lógicos la identificación es portada en el payload header. La señalización de control que no requiere respuesta de un solo slot es llevado a cabo usando el protocolo LMP.

El Cuadro 3.1 lista todos los tipos de transporte lógico, los tipos de enlaces lógicos de soporte, que tipo de enlaces físicos y canales físicos pueden soportar, y una descripción breve del propósito del transporte lógico.

Los nombres dados a los enlaces lógicos y transportes lógicos reflejan algunos de los nombres usados en Bluetooth 1.1, in orden de proveer alguno grado de familiaridad y continuación. Sin embargo estos nombres no reflejan un esquema consecuente, el cuál es bosquejado abajo.

La clasificación de cada tipo de enlace resulta de un procedimiento de selección dentro tres categorías.

| Logical transport | Links supported | Supported by | Overview |
|---|---|---|---|
| Asynchronous Connection-Oriented (ACL[1]) | Control (LMP) ACL-C<br>User (L2CAP) ACL-U | Active physical link, basic or adapted physical channel | Reliable or time-bounded, bi-directional, point-to-point. |
| Synchronous Connection-Oriented (SCO) | Stream (unframed) SCO-S | Active physical link, basic or adapted physical channel | Bi-directional, symmetric, point-to-point, AV channels. Used for 64Kb/s constant rate data. |
| Extended Synchronous Connection-Oriented (eSCO) | Stream (unframed) eSCO-S | Active physical link, basic or adapted physical channel | Bi-directional, symmetric or asymmetric, point-to-point, general regular data, limited retransmission. Used for constant rate data synchronized to the master Bluetooth clock. |
| Active slave broadcast (ASB) | User (L2CAP) ASB-U | Active physical link, basic or adapted physical channel. | Unreliable, uni-directional broadcast to any devices synchronised with the physical channel. Used for broadcast L2CAP groups. |
| Parked slave broadcast (PSB) | Control (LMP) PSB-C, User (L2CAP) PSB-U | Parked physical link, basic or adapted physical channel. | Unreliable, uni-directional broadcast to all piconet devices. Used for LMP and L2CAP traffic to parked devices, and for access requests from parked devices. |

*Tabla 3.1: Tipos de transporte lógicos.*

[1] Es claro que la abreviación más obvia para la Conexión-Orientada Asíncrona del transporte lógico es ACO. Sin embargo, esta sigla tiene un
significado de la alternativa de la especificación Bluetooth 1,1. Para evitar la confusión entre dos significados posibles para ACO la decisión se hizo para retener la abreviación de ACL para el transporte Lógico, Conexión-Orientada y Asíncrona.

### 3.5.1 Casting

La primera categoría es Casting. Esto podría ser cualquier unicast o broadcast. No hay ningún enlace multicast definido en Bluetooth 1.2

• Enlace Unicast

Los enlaces Unicast existen exactamente entre dos puntos finales. El tráfico puede envíar en cualquier dirección sobre los enlaces unicast. Todos los enlaces unicast son conexión-orientada, quiere decir que un procedimiento de conexión tiene lugar antes que el enlace pueda ser usado.

En el caso de enlace default ACL, el procedimiento de conexión es un paso implícito dentro del procedimiento general paging que usa la forma ad-hoc piconets.

•Enlaces broadcast.

Los enlaces broadcast existen entre un dispositivo fuente y el cero o más dispositivos de recepción. El tráfico es unidireccional, y de la que solamente transmite el dispositivo fuente a los dispositivos de recepción. Los enlaces broadcast son sin conexión, quiere decir que no hay ningún procedimiento para crear éste enlace, y los datos pueden enviarse sobre ellos en cualquier momento. Los enlaces broadcast son poco fiables, y hay no garantía que los datos serán recibidos.

### 3.5.2 Esquema Scheduling y Acknowledgement

La segunda categoría se relaciona con el esquema Scheduling y acknowledgement del enlace, e implica los tipos de tráfico que son respaldados por el enlace. Éstos son síncronos, isócronos o asíncronos. No hay ninguna especificación de enlace isócrona definida en Bluetooth 1.2, aunque el enlace default ACL puede ser configurado para operar en este modo.

• Enlace síncrono.

Los enlaces síncronos proveen un método de asociarse al reloj piconet de Bluetooth con los datos transportados. Esto es realizado para reservar slots regulares en el canal físico, y transmitir el tamaño fijo de paquetes en estos intervalos regulares. Tales enlaces son apropiados para un rate constante de datos isócronos.

• Enlaces asíncronos.

Los enlaces asíncronos suministran un método para transportar los datos que no tienen ninguna característica basada en tiempo. Los datos son esperados normalmente para ser retransmitidos hasta que son recibidos exitosamente, y cada entidad de datos puede procesarlos en cualquier momento después de recibirlos, sin la referencia de tiempo de recepción de algún entidad previa o sucesiva en el flujo (provee el pedido de las entidades de datos y es conservado.)

• Enlaces isócronos.

Los enlaces isócronos proveen un método para transportar los datos, tienen características basadas en tiempo. Los datos pueden ser retransmitidos hasta que son recibidos o expirados. El rate de datos sobre el enlace no necesita ser constante (esto es la principal diferencia de los enlaces síncronos.)

### 3.5.3 Clase de datos

La categoría final está relacionada con la clase de datos que son llevados por el enlace. Esto son los datos de control (LMP) o los usuarios de datos. La categoría usuarios de datos es subdividida dentro de los datos L2CAP (o framed) y el flujo de datos (o unframed).

• El enlace de control.

Los enlaces de control son solamente usados para transportar los mensajes de LMP entre dos directorios de enlaces. Estos enlaces son invisibles encima de la capa baseband, y no puede ser directamente, configurados o liberado por aplicaciones, aparte el uso del servicio de la conexión y desconexión que tiene este efecto implícitamente. El enlace de control son siempre multiplexados un enlace L2CAP equivalente en un transporte lógico ACL. Sujeto a las reglas definidas en el esquema ARQ, el enlace de control de tráfico se encarga de la prioridad del tráfico del enlace de L2CAP.

• Enlace L2CAP.

Los enlaces L2CAP son usados para transportar L2CAP PDUs, que pueden llevar el canal de señalización L2CAP (solamente en enlace lógico default ACL-U) o en usuarios de datos framed que accedieron a canales L2CAP de usuarios instantáneos en canales L2CAP, los frames baseband pueden ser más grandes que el baseband disponible para los paquetes. Un protocolo de control de enlace se incrusta dentro del campo de LLID conservando el frame -star y la semántica de frame-continuación cuando el frame transmite en varios fragmentos al receptor.

• Flujo de enlaces

Los enlaces de flujo son usados para transportar a los usuarios de datos que no tienen frame inherente, eso debe ser conservado cuando se reparten los datos. Los datos perdidos pueden ser reemplazados por el padding en la recepción.

### 3.5.4 Conexión-orientada asíncrona (ACL)

La conexión –orientada asíncrona (ACL) del transporte lógico es usada para portar señalización de control LMP y L2CAP y mejores usuarios de datos asíncronos. El transporte lógico de ACL usa un esquema de ARQN / SEQN de un 1 bit simple, provee de manera sencilla un canal confiable. Cada dispositivo slave activo dentro de un piconet tiene uno transporte lógico ACL para el piconet master, conocido como el default ACL.

El default ACL entre el master y el slave, cuando un dispositivo une a un piconet (se conectara al piconet canal físico básico). Este default ACL es asignado a una dirección de transporte lógica (LT_ADDR) por el master de piconet. Este LT_ADDR también es usado para identificar el enlace físico activo cuando se requiere (o como un miembro activo piconet identificador, eficazmente para el mismo propósito.)

El LT_ADDR para el default ACL es re-usado para la conexión–orientada asíncrona del transporte lógico entre el mismo master y el slave. (Esto es para las razones de la compatibilidad con las primeras especificaciones de Bluetooth) Por lo tanto, LT_ADDR no es autosuficiente para identificar el default ACL. Sin embargo los tipos de paquetes usados sobre ACL son diferentes de aquellos usados sobre la conexión –orientada asíncrona del transporte lógico. Por lo tanto, el transporte lógico de ACL puede ser identificado por el campo de LT_ADDR en el header de paquete en combinación con el campo de tipo de paquete.

El default ACL puede ser usado para el transporte de datos isócronos por configuración para vaciar paquetes automáticamente después de que los paquetes han expirado.

Si el default ACL es removido del enlace físico activo, después los demás transportes lógicos que existan entre el master y el slave también son retirados. En el caso inesperado de la pérdida de sincronización para el piconet canal físico, el enlace físico y todos los transportes lógicos y enlaces lógicos dejan de existir en el tiempo que la pérdida de sincronización es detectada.

Un dispositivo puede retirar este default ACL (y por la implicación de este enlace físico activo) pero queda sincronizado al piconet. Este procedimiento es conocido como parking, un dispositivo que es sincronizado al piconet, pero no tiene ningún enlace físico activo, esta en parked dentro de ese piconet.

Cuando el dispositivo pasa al estado paked el enlace lógico default ACL que es transportado sobre el default ACL del transporte lógico se queda en existencia, pero queda suspendido. Ningún dato puede ser transferido a través de un enlace lógico suspendido. Cuándo el dispositivo pasa del estado parked, regresa dentro de un estado activo, un nuevo default ACL del transporte lógico es creado (podría ser, un diferente LT_ADDR de uno ya existente) y los enlaces lógicos suspendidos son adjuntos a este default ACL, pasa a ser activo otra vez.

### 3.5.5 Conexión –orientada síncrona (SCO)

La conexión–orientada asíncrona (SCO) del transporte lógico es simétrica, el canal punto - a - punto entre el master y un slave específico. El SCO transporte lógico reserva slot en el canal físico y puede por lo tanto ser considerado como una conexión circuito-switched entre el master y el slave. El SCO transporte lógico lleva 64 kb / s de información sincronizada con el reloj piconet. Típicamente esta información es un flujo de voz codificado. Existen tres diferentes configuraciones SCO, brindando un balance entre fuerza, demora y consumo de ancho de banda.

Cada enlace lógico SCO es soportado por un simple transporte lógico SCO, el cuál es asignado al mismo LT_ADDR como default ACL del transporte lógico entre dispositivos. Por lo tanto el campo de LT_ADDR no es suficiente para identificar el destino de un paquete recibido. Porque los enlaces SCO usan slots reservados, un dispositivo usa una combinación de la LT_ADDR, el número de slots (una propiedad del canal físico) y el tipo de paquete de identificar la transmisión en el enlace SCO.

El uso repetido de LT_ADDR del default ACL para el transporte lógico SCO es debido al funcionamiento de de la especificación Bluetooth 1.1. En las primeras versiones de Bluetooth el LT_ADDR (que es conocido como el miembro de dirección activo), fue usado para identificar al miembro de piconet relacionado con cada transmisión.

Esto no fue Fácilmente de extender para permitir mas enlaces lógicos, y cierto propósito de este campo fue redefinido para las nuevas características. Algunas características de Bluetooth 1.1 lo hacen, sin embargo, no se ajusta a la arquitectura oficialmente descrita.

Aunque los slots son reservados para el SCO, está permitido usar un slot reservado para el tráfico de otro canal eso tiene una prioridad más alta. Esto puede ser requerido como consecuencia de los cometidos de QoS, a LMP de envío de señalización sobre el default ACL cuando el ancho de banda de canal físico es ocupado completamente por SCOs, como SCOs Lleva tipos de paquete diferentes a ACLs, el tipo de paquete es usado para identificar el trafico SCO (además el numero de slots y LT_ADDR.)

No hay ninguna capa arquitectónica adicional definida por la especificación del núcleo Bluetooth. Es transportado sobre un enlace de SCO. En varios formatos Standard definidos para el flujo de 64 kb /s en el que es transportado, en un flujo sin formato, es permitido dónde la aplicación es responsable de interpretar la codificación de flujo.

.

## 3.5.6 Conexión -Orientada Sincronía extendida (eSCO)

El transporte lógico (eSCO) conexión -orientada síncrona extendida es un enlace punto a - punto simétrico o asimétrico entre el master y un slave especifico, los slots reservados eSCO en el canal físico y por lo tanto pueden considerarse como una conexión circuito-switched entre el master y el slave.

Los enlaces eSCO brindan varias extensiones sobre los enlaces de SCO usuales, en esto respaldan una combinación más flexible de tipos de paquete y datos seleccionados contenidos en los paquetes y los períodos slots seleccionados, admite una extensión de bit Rates síncronos que pueden ser soportados.

Los enlaces eSCO también pueden brindar retransmisión limitada de paquetes (a diferencia de los enlaces SCO donde no hay retransmisión.) Si estas retransmisiones son requeridas ellas toman el lugar de los slots que siguen a los slots reservados, de otro modo los slots pueden ser usados para otro tráfico.

Cada enlace lógico eSCO - S es soportado por un solo transporte lógico eSCO, e identificado por un LT_ADDR que es único dentro del piconet en la duración de eSCO. Los enlaces de eSCO - S son creados usando la señalización LM siguiendo las mismas reglas para enlaces de SCO - S.

No hay ninguna capa arquitectónica adicional definida para la especificación del núcleo Bluetooth esto es transportado sobre un enlace eSCO - S. En vez de la aplicación puede usarse el flujo de datos para cualquier propósito que requieran, sujeto a las características del transporte de flujo siendo conveniente para transportar los datos.

## 3.5.7 Slave activo broadcast (ASB)

El slave activo broadcast de transporte lógico es usado para transportar al usuario de trafico L2CAP para todos los dispositivos en el piconet que esta actualmente conectado al canal físico y usado por el ASB. No hay protocolo acknowledgement y el tráfico es unidireccional del piconet master al slave. El canal ASB puede ser usado para el tráfico de grupo L2CAP (un legado de la especificación 1.1), y nunca es usado para canales de conexión -orientada L2CAP, el control de señalización L2CAP o el control de señalización LMP.

El transporte lógico ASB es intrínsecamente poco fiable debido a la falta de acknowledgement. Para mejorar la confiabilidad, cada paquete es transmitió un número de veces. Un número de secuencia idéntica es usada para ayudar a filtrar la retransmisión en el dispositivo slave.

El transporte lógico ASB es identificado por un LT_ADDR reservado. (La dirección de LT_ADDR reservado también es usada por el transporte lógico PSB.) Un slave activo recibirá tráfico en ambos transportes lógicos, y no puede distinguir fácilmente entre ellos. Como transporte lógico ASB no lleva tráfico de LMP un slave activo puede ignorar los paquetes recibidos sobre el enlace lógico LMP en el transporte lógico ASB. Sin embargo el tráfico L2CAP transmite sobre el transporte lógico PSB que es recibido por slaves activos en el transporte lógico ASB y no puede ser distinguido del trafico L2CAP enviado en el transporte de ASB.

Un ASB es creado implícitamente siempre que un piconet existe, y siempre hay un ASB asociado con cada uno de los canales físicos piconet básico y adaptado que existen dentro del piconet. Porque los canales físicos piconet básico y adaptado son principalmente coincidentes con un dispositivo slave que no puede distinguir cuál de los canales ASB esta usándose para transmitir los paquetes. Esto se añade al uso general de la falta de fiabilidad del canal de ASB. (Aunque esto, quizás, es poco fiable generalmente con paquetes perdidos.)

Un dispositivo master puede decidir usar solamente uno de sus dos ASBs posibles (cuándo tiene ambos canales físicos piconet básico y adaptado), así con suficientes retransmisiones es posible dirigir ambos grupos de slaves sobre lo mismo canal ASB.

El canal ASB nunca es usado para portar el control de señalización de LMP o L2CAP.

### 3.5.8 Slave de transmisión Parked (PSB)

El transporte lógico del slave de transmisión parked es usado para la comunicación entre el master y el slave que esta en parked (habiendo dejado su default ACL del transporte lógico.) El enlace del slave de transmisión parked es el único transporte lógico que existe entre el piconet master y slaves parked.

El transporte lógico PSB es más complicado que otros transportes lógicos, este consta de varias fases, teniendo un propósito diferente cada una. Éstas fases son la fase de control de información (solía llevar el enlace lógico LMP), la fase de información de usuario (lleva el enlace lógico L2CAP), y la fase de acceso (portando la señalización baseband). La información de control y la fase de transmisión de información son general y mutuamente exclusivas como solamente una de ellas puede ser soportar un intervalo beacon. (Incluso si no hay control o fase de información de usuario, el master es todavía requerido para transmitir un

paquete de beacon slots con el propósito de que el slave parked pueda resincronizarse). La fase de acceso está normalmente presente a menos que se cancele un mensaje de información de control.

La fase de información de control es usada por el master para envíar información al slave parked que contiene modificaciones a los atributos del transporte PSB, modificaciones para los atributos del beacon train, o una petición para un slave parked para ponerse activo en el piconet (conocido como **unparking**). Esta información de control es portada en los mensajes LMP en un enlace lógico LMP. (La fase de información de control está también presente en el caso de una fase de información de usuario donde el usuario de información requiere más de un paquete de baseband.)

Los paquetes en la fase de información de control son transmitidos en el canal físico beacon train slots, y no pueden ser transmitidos sobre cualquier otros slots. La información de control ocupa un solo paquete DM1 y es repetido en cada beacon train slots dentro de un intervalo beacon. (Si no hay información de control entonces puede haber una fase de información de usuario que usa los beacon slots. Si ninguna de las fases es usada entonces los beacon slots son usados por otro transporte lógico de tráfico o para paquetes **NULL**.)

La fase de información de usuario es usada por el master para envíar paquetes de L2CAP que esta destinado para todos los slaves piconet. La información de usuario puede ocupar uno o más paquetes de baseband. Si la información de usuario ocupa un solo paquete entonces el paquetea de información de usuario es repetido en cada uno de los canales de piconet beacon train slot.

Si la información de usuario ocupa más de un paquete baseband entonces este es transmitido en slots después del beacon train (el **broadcast scan window**) y el beacon slot es usado para transmitir un mensaje de la fase de información de control que contiene los atributos de timing de este **broadcast scan window**. Esto es requerido de manera que el slaves parked queda conectado al canal físico piconet para recibir la información de usuario.

La fase de acceso está normalmente presente a menos que se cancele temporalmente por un mensaje de control portado en la fase de transmisión de información de control. El acceso window consta de una secuencia de slots que siguen al beacon train. En el orden para que un slave parked quede activo en el piconet, esto puede envíar tal acceso al piconet master durante el acceso de window. Cada slave parked es asignada una dirección de solicitud de acceso (no necesariamente único) que es controlada cuando dura el acceso window el slave pide el acceso.

El transporte lógico PSB es identificado por el LT_ADDR reservado de 0. Esta dirección de LT_ADDR reservada también es usada por el transporte lógico ASB. Los slaves parked no son confundidos normalmente por el uso repetido de

LT_ADDR, como ellos son conectados solamente al canal físico piconet durante el tiempo que el transporte PSB está siendo usado.

### 3.5.9 Enlaces lógicos

Algunos transportes lógicos son capaces de soportar diferentes enlaces lógicos, ambos multiplexados simultáneamente, o una alternativa. Dentro del transporte lógico, el enlace lógico es identificado por el **identificador de enlace lógico (LLID),** bits en el payload header del paquete de baseband que porta datos payload. Los enlaces lógicos distinguen entre un limitado set 1 de protocolos del núcleo que pueden transmitir y recibir datos en transportes lógicos. No todos de los transportes lógicos son capaces de portar todos los enlaces lógicos (esto mostrado en Figura 3.2)

En particular los transportes lógicos SCO y eSCO son solamente capaces de portar rate streams de datos constantes, y éstos son identificados únicamente por LT_ADDR. Tales transportes lógicos solamente usan paquetes que no contienen payload header, como su longitud es conocida con anticipadamente y ningún LLID es necesario.

### 3.5.10 ACL Enlace Lógico de Control (ACL - C)

El enlace lógico de control ACL (ACL - C) es usado para portar la señalización LMP entre dispositivos piconet. El enlace de control es solamente portado en el default ACL de transporte lógico y en el transporte lógico PSB (en la fase de información de control).
El enlace ACL - C siempre da prioridad sobre el enlace ACL-U (vea abajo) cuándo porta el mismo transporte lógico.

### 3.5.11 Enlace Lógico Asíncrono / Isócrono de Usuario (ACL-U)

El enlace lógico asíncrono / isócrono de usuario (ACL-U) se usa para portar todas las tramas asíncronas e isócronas de los usuarios de datos. El enlace ACL-U es portado casi en el transporte lógico síncrono. Los paquetes en el enlace ACL-U son identificados por uno de los dos valores LLID reservados. Uno valor es usado para indicar ya sea el contenido del paquete baseband, el principio de una trama L2CAP y el otro indica una continuación de una trama previa. Esto asegura la sincronización correcta de la re-ensamblación siguiente del flujo de paquetes de L2CAP. El uso de esta técnica quita la necesidad de un L2CAP header más completo en cada paquete Baseband (el header solamente es requerido en los paquetes de inicio de L2CAP), solamente añade el requisito de una trama L2CAP completa, será transmitido antes que una nueva sea transmitida. (Una excepción

para esta regla será la habilidad para fluir una transmisión parcial de una trama L2CAP a favor de otra trama L2CAP.)

### 3.5.12 Enlaces Lógicos Síncrono y Síncrono/extendido de usuario (SCO-S / eSCO-S)

Los Enlaces Lógicos Síncrono (SCO - S) y Síncrono/extendido (eSCO - S) son usados para soportar datos isócronos repartidos en un flujo sin fraiming. Éstos enlaces están asociados con el transporte lógico, donde los datos son repartidos en unidades de tamaño constante en uno rate constante. No hay LLID dentro de los paquetes en estos transportes, cuando solamente un enlace lógico puede ser soportado, y la longitud del paquete y el período scheduling están predeterminados y se quedan fijos durante la vida del enlace.

Un rate variable de datos isócronos no puede ser portado por los enlaces lógicos SCO - S o eSCO - S. En este caso los datos deben ser portados en enlaces lógicos ACL-U, el cuál usa paquetes con payload header. Bluetooth tiene algunas limitaciones cuando soporta datos isócronos de rate variable simultáneamente con usuarios de datos confiables.

## 3.6 Canales de L2CAP

L2CAP provee un role de multiplexión que permite diferentes aplicaciones para comparte recursos de un enlace lógico ACL-U entre dos dispositivos. Las aplicaciones y el servicio del protocolo de la interfase con L2CAP usan una interfaz canal –orientada para crear conexiones a entidades equivalentes en otros dispositivos.

Los puntos finales del canal L2CAP son identificados a sus clientes por un identificador de canal (CID). Esto es asignado por L2CAP, y cada punto final del canal L2CAP en ningún dispositivo tiene un CID diferente.

Los canales L2CAP pueden ser configurados para proveer una calidad apropiada del servicio (QoS) para la aplicación. L2CAP mapea el canal sobre el enlace lógico ACL-U.

L2CAP soporta canales que son conexión -orientada y otros que son grupo-orientado. Los canales grupo -orientado pueden ser mapeados sobre el enlace lógico ASB – U, o implementado como la transmisión que repite a cada miembro dentro de un cambio en el enlace lógico ACL-U.

Aparte de la creación, la configuración y desmontaje de canales, el principal role de L2CAP es mulltiplexar las unidades de servicio de datos (SDUs) de los clientes

del canal sobre los enlaces lógico ACL-U, y realizar un simple nivel scheduling, seleccionando un SDUs de acuerdo con la respectiva prioridad.

L2CAP puede proveer para el canal de control de flujo que es semejante a la capa de L2CAP. Esto alternativa es seleccionada por la aplicación cuando el canal es establecido. L2CAP también puede proveer y aumentar la detección de errores y retransmite para reducir (a) la probabilidad de errores inadvertidos que están pasado a la aplicación y (b) recuperar la pérdida de porciones de los datos de usuario cuando la capa Baseband ejecuta un flujo en el enlace lógico ACL-U.

En el caso donde uno HCI está presente, L2CAP también es requerido al segmento L2CAP SDUs respecto a fragmentos que caben dentro de los buffers baseband, y además para operar un Token basado en el procedimiento de control de flujo sobre HCI, sometiendo fragmentos al baseband solamente cuando permite hacerlo. Esto podría afectar el algoritmo scheduling.

# 4 Topología de Comunicación

## 4.1 Topología de PICONET

En cualquier momento un enlace Bluetooth es formateado dentro del contexto de un piconet. Un piconet consta de dos o más dispositivos que habitan el mismo canal físico (cualquiera puede decir que son sincronizados a un reloj común y a una secuencia hopping.). El reloj común (piconet) es idéntico al reloj Bluetooth de uno de los dispositivos en el piconet, conocido como el master del piconet, y la secuencia hopping se deriva del reloj del master y el dispositivo de dirección Bluetooth del master. Todos los otros dispositivos sincronizados son referidos como slaves en el piconet. Los términos master y slave son solamente usados cuándo describen estos papeles en un piconet.

Dentro de una ubicación común varios piconets independientes podrían existir. Cada piconet tiene un canal físico diferente (ése es un dispositivo master diferente un reloj de piconet y un secuencia hopping independiente.)

Un dispositivo Bluetooth puede participar en dos o más piconets simultáneamente. Él hace esto, sobre una base de división de tiempo multiplexado. Un dispositivo Bluetooth nunca puede ser un master de más de un piconet. (Ya que el piconet es definido por la sincronización del reloj Bluetooth del master es imposible ser el master de dos o más piconets.) Un dispositivo Bluetooth podría ser un slave en muchos piconet independiente.

Un dispositivo Bluetooth que es un miembro de dos o más piconets esta involucrado en un scatternet. La participación en un scatternet no necesariamente implica alguna capacidad de direccionamiento de red o funcionamiento en un dispositivo Bluetooth. Los protocolos de núcleo Bluetooth no lo hacen, y no son requeridos para brindar tal funcionalidad, el cuál es responsabilidad del protocolo de más alto nivel y esta fuera del alcance de la especificación del núcleo Bluetooth.

En la Figura 4.1 se muestra un ejemplo de la topología y manifiesta un número de características arquitectónicas descritas abajo. El dispositivo A es un master en un Piconet (representado por el área sombreada, y conocido como un piconet A) y con B, C, D y E como dispositivos slaves. Otros dos piconets son mostrados:

a) Un piconet con F como dispositivo master (conocido como piconet F) y los dispositivos E, G y H como slaves.

b) Un piconet con D de dispositivo como master (conocido como piconet D) y J como slave.

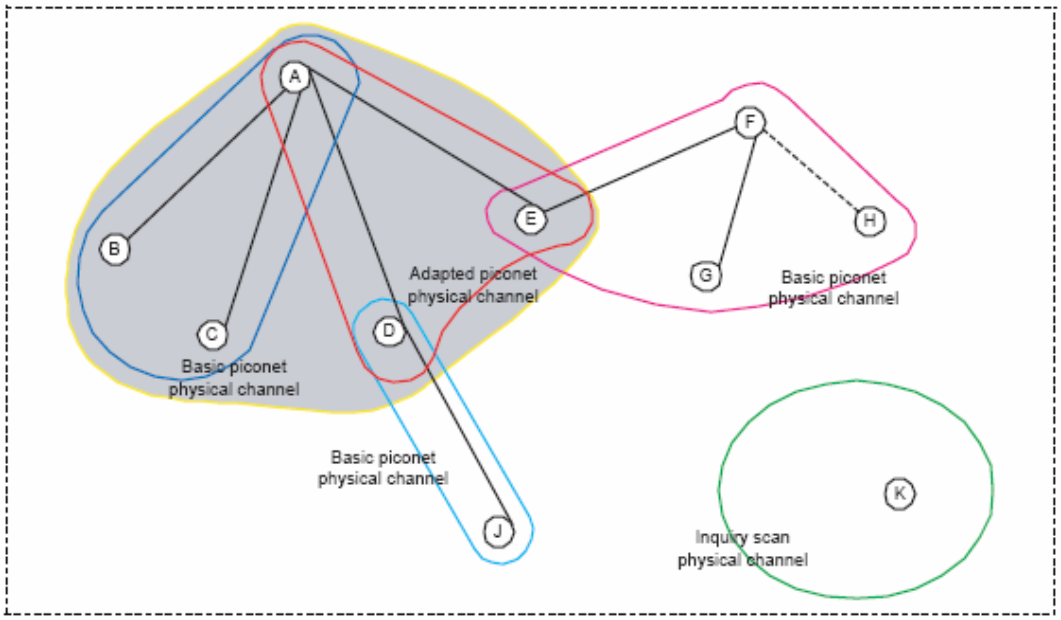



*Figura 4.1: Ejemplo de la Topología Bluetooth*

En el piconet A hay dos canales físicos. Los dispositivos B y C están usando el canal físico piconet básico (representado por un encapsulado azul) como ellos no soportan la frecuencia adaptada hopping. Los dispositivos D y E son capaces de soportar la frecuencia adaptada hopping, y estar usando el canal físico piconet adaptado (representado por un encapsulado rojo.) El dispositivo A es susceptible de la frecuencia adaptada hopping, y opera en una base de TDM sobre ambos canales físicos de acuerdo con las cuales el slave está siendo direccionado.

El piconet D y F, usan solamente un canal físico piconet básico (representado por el encapsulado de color cyan y magenta respectivamente.) En la caso del piconet D esto es porque el dispositivo J no soporta el modo adaptable hopping. Aunque el dispositivo D soporta el hopping adaptable no puede usarlo en este Piconet. En el piconet F del dispositivo F no soporta el hopping adaptable, y por lo tanto no puede ser usado en este piconet.

El dispositivo K es mostrado en la misma localidad como otro dispositivos. Este no es actualmente un miembro de piconet, pero tiene servicios que ofrece a otros dispositivos de Bluetooth. Este atiende actualmente en su canal físico inquiry scan (representado por el encapsulado verde), aguarda un pedido de inquiry request de otro dispositivo.

Los enlaces físicos (un slave por dispositivo) son representados en la diagrama por líneas conectando los dispositivos. Las líneas rectas representan un enlace físico activo, y la línea punteada representa un enlace físico parked. El dispositivo H es parked, y por lo tanto el enlace físico entre el master (F) y el slave (H) es mostrado como parked.

El transporte lógico, enlaces lógicos y el canal L2CAP son usados para suministrar las capacidades para el transporte de datos, pero no son mostrados en el diagrama.

## 4.2 Procedimientos de funcionamiento y Modos

El modo típico de operaciones de un dispositivo Bluetooth es estar relacionado con otro dispositivo Bluetooth (en un piconet) y cambiar los datos con este dispositivo Bluetooth. Cuando Bluetooth es una tecnología de comunicaciones inalámbrica ad hoc hay también varios procedimientos en funcionamiento que permiten que un piconet sea moldeado con el propósito de que las comunicaciones siguientes puedan tener lugar. Los procedimientos y los modos son aplicados en las diferentes capas de la arquitectura y por lo tanto un dispositivo puede estar empleando un número de estos procedimientos y modos simultáneamente.

### 4.2.1 Procedimiento inquiry (descubrir)

Los dispositivos de Bluetooth usan el procedimiento inquiry para descubrir dispositivos cercanos, o ser descubierto por dispositivos en su localidad.

El procedimiento inquiry es asimétrico. Un dispositivo Bluetooth que trata de descubrir otros dispositivos cercanos es conocido como un dispositivo inquiring y activa enviando inquiry request. El dispositivo Bluetooth que esta disponible para ser encontrado es conocido como dispositivos discoverable y esta atento a estos inquiry request y envía respuestas. El procedimiento inquiry usa un canal físico especial para el inquiry request e inquiry responses.

Ambos dispositivos inquiring y discoverable podrían ya estar conectados con los otros dispositivos Bluetooth en un piconet. En cualquier momento inquiring emplea u ocupa el canal físico inquiring scan que necesita ser balanceado con las demanda de QoS en los transportes lógicos existentes. El procedimiento de inquiry no utiliza ningunas de las capas arquitectónicas encima del canal físico, aunque un enlace físico transitorio puede ser considerado para estar presente durante el intercambio de la información de inquiry e inquiry scan.

### 4.2.2 Procedimiento paging (conexión)

El procedimiento para formar conexiones es asimétrico y requiere que un dispositivo Bluetooth lleve el procedimiento paging (conexión) mientras los demás dispositivos Bluetooth son conectados (paging scanning.) El procedimiento es centrado, para que el procedimiento page sea respondido solamente por un dispositivo Bluetooth especificado.

El dispositivo connectable usa un canal físico especial para atender la solicitud de paquetes en la conexión del dispositivo paging (conexión). Este canal físico tiene atributos que son específicos del dispositivo connectable, por lo tanto solamente un dispositivo paging con conocimiento del dispositivo connectable puede comunicarse sobre este canal.

Ambos dispositivos paging y connectable podrían estar ya conectados con otros dispositivos Bluetooth en un piconet. En cualquier momento paging emplea u ocupa el canal físico inquiring scan que necesita ser balanceado con las demanda de QoS en transportes lógicos existentes.

### 4.2.3 Modo connected (conectado).

Después de un procedimiento de conexión exitoso, los dispositivos físicamente son conectados dentro de un piconet. Esto quiere decir que hay un canal físico piconet al que ambos son conectados, hay un enlace físico entre los dispositivos, y hay enlaces lógico default ACL – C y ACL – U. Cuando en el modo connected es posible crear y liberar enlaces lógicos adicionales, para cambiar los modos de los enlaces físicos y lógicos mientras queda conectado al canal físico piconet. Es también posible para el dispositivo llevar a cabo los procedimientos inquiry paging y scanning o estar conectado con otros piconets sin necesidad de desconectarse del canal físico piconet original.

Los enlaces lógicos adicionales son creados usando el Link Manager que cambia el enlace en los mensajes Link Manager Protocol con el dispositivo remoto Bluetooth para negociar la creación y ajustes para estos enlaces. Los enlaces lógicos default ACL – C y ACL – U son siempre creados durante el proceso de conexión, y éstos son usados por los mensajes LMP y la respectiva señalización del canal L2CAP.

Es notado, que dos enlaces lógico default son creados cuando dos unidades son inicialmente conectadas. Uno de estos enlaces (ACL - C) transporta el protocolo de control LMP y es invisible a una capa mas alta de Link Manager. El otro enlace (ACL – U) transporta el protocolo de señalización de L2CAP y cualquier mutiplexión L2CAP. Es común hacer referencia a un transporte lógico default ACL, que puede ser resuelto por el contexto, pero típicamente refiere al enlace lógico default ACL – U. También note que estos dos enlaces lógicos comparten un transporte lógico.

Durante el tiempo que un dispositivo slave está activamente conectado con un piconet siempre hay un transporte lógico default ACL entre el dispositivo slave y el master. Hay dos métodos de eliminar el transporte lógico default ACL. El primer método es el que separa el dispositivo del canal físico piconet, cada vez que la

jerarquía total de los canales L2CAP, enlaces lógicos y transporte lógico entre los dispositivos sea eliminada.

El segundo método es poner el enlace físico al dispositivo slave en estado park cada vez qué deja el transporte lógico default ACL. Esto solamente es permitido si todos los otros transportes lógicos han sido eliminados (excepto, el transporte lógico ASB que puede ser explícitamente creado o eliminado.). No es permitido park en un dispositivo mientras tiene cualquier transporte lógico aparte de default ACL y transporte lógico ASB.

Cuando el enlace físico del dispositivo slave esta en parked, su transporte lógico default ACL es liberado y el transporte lógico ASB es reemplazado con un transporte lógico PSB. Los enlaces lógico ACL – C y ACL – U que son multiplexados sobre el transporte lógico default ACL quedan en existencia pero no puede ser usado para el transporte de datos. El Link Manager en el dispositivo master se auto restringe el uso de los mensajes LMP que pueden transportarse en los enlaces lógicos PSB - C.

El canal Manager y L2CAP Resource Manager asegura que el tráfico de datos unidireccional L2CAP no sea sometido al controlador, mientras el dispositivo esta en parked. El canal Manager puede decidir dirigir parked y unparking de los dispositivo, tan necesarios para permitir que los datos sean transportados.

### 4.2.4 Modo Hold

El modo Hold no es un modo del dispositivo general, pero es aplicable a slots sin reservar en el enlace físico. Cuando en este modo el enlace físico es solamente activo durante los slots que son reservados para la operación de los tipos de enlace síncronos SCO y eSCO. Todos los enlaces asíncronos están inactivos. Operar el modo hold una vez para cada invocación y ser retirado cuando esta completo, regresando al modo previo.

### 4.2.5 Modo Sniff

El modo Sniff no es un modo del dispositivo general, pero es aplicable al transporte lógico default ACL. Cuando en este modo la disponibilidad de estos transportes lógicos modificada definiendo un ciclo de servicio que consta de los períodos de presencia y ausencia. Los dispositivos que tienen su transporte lógico default ACL en el modo sniff pueden usar los períodos ausentes para participar en actividades en otro canal físico, o entra al modo power reduced. El modo Sniff solamente afecta al transporte lógico default ACL (por ejemplo su transporte lógico ACL compartido), y no aplica para ningún transporte lógico SCO o eSCO adicionales que pueden ser activos. Los períodos de presencia y ausencia del

enlace físico en un canal físico piconet son derivado como una unión de todos los transportes lógicos que son desarrollados en el enlace físico.

Note que los transportes lógicos broatcast no tienen ninguna expectativa definida para la presencia o ausencia. Un dispositivo master debe aspirar a programar las transmisiones para que coincidan con los períodos del enlace físico presente en el canal físico piconet, pero esto puede no ser siempre posible o práctico. La repetición de transmisiones es definida para mejorar las posibilidades de contactar a slaves múltiples sin traslapar períodos de presencia. Sin embargo, el transporte lógico broadcast no puede ser considerado seguro.

### 4.2.6 Estado Parked

Un dispositivo slave puede quedar relacionado con un piconet pero tiene su enlace físico en estado parked. En este estado el dispositivo no puede soportar ningún enlace lógicos con el master con excepción de los enlaces lógicos PSB – C y PSB – U que son usados para toda comunicación entre el piconet master y slave parked.

Cuando el enlace físico en un dispositivo slave esta en parked quiere decir que hay restricciones cuándo el master y el slave pueden comunicarse, definido por los parámetros del transporte lógico PSB. Algunas veces cuando el transporte lógico PSB esta inactivo (o ausente) entonces los dispositivos pueden participar en la actividad en otro canal físico, o entrar en el modo power reduced.

### 4.2.7 Procedimiento Role Switch

El procedimiento Role Switch es un método para intercambiar los papeles de dos dispositivos conectados en un piconet. El procedimiento involucra la variación del canal físico, canal que es definido por el dispositivo master original al canal físico que es definido por el nuevo dispositivo master. En el proceso de intercambiar un canal físico al próximo, la jerarquía de enlaces físicos y transporte lógico es retirado y reconstruido, con excepción del transporte lógico de ASB y PSB que son implicados por la topología y no están conservados. Después del Role Swich, el canal físico piconet original puede dejar de existir o puede continuar si el master original tuviera otros slaves que todavía están conectado al canal.

El procedimiento solamente copia los enlaces lógico default ACL y las capas de soporte al nuevo canal físico. Cualquier transporte lógico adicional no es copiado por este procedimiento, y si se requiere, este debe ser llevado por las capas más altas. El LT_ADDRs de cualquier transporte afectado no podría estar conservado cuando los valores pueden ya estar en uso en el nuevo canal físico.

# Capitulo II
# Core System Package
# [Controller volume]

# PARTE A
# RADIO ESPECIFICACION

## 1 Alcance

Los dispositivos Bluetooth operan en la banda 2.4 GHz sin licitación ISM (Industrial Científico y Médico). Una frecuencia de salto transceptor se aplica para combatir la interferencia y el desvanecimiento. En forma, de la modulación de FM binaria se aplica para minimizar la complejidad del transceptor. El symbol rate es de 1 Ms/s. Para la transmisión full duplex, se usa un esquema TDD. Esta especificación define los requisitos para la radio Bluetooth.

Se definen estos requisitos por dos razones:

- Provee la compatibilidad entre radios usados en el sistema.
- Define la calidad del sistema.

El radio Bluetooth debe satisfacer los requisitos bajo las condiciones de operación especificadas en el Apéndice A y Apéndice B. Los parámetros de la radio serán medidos según los métodos descritos en la prueba de Especificación de RF.
Esta especificación esta basada en las regulaciones establecidas para Europa, Japón, y América del Norte. Los documentos normales listados abajo sólo son para información, y está sujeto a cambio o revisión en cualquier momento.

Europa: Normas de la aprobación: Instituto de Normas de Telecomunicaciones europeo, ETSI, Documentos: EN 300 328, ETS 300-826, Autoridad de la aprobación: Autoridades de Aprobación de Tipo nacionales.

Japón: Normas de la aprobación: La asociación de Industrias de la Radio y Negocios, ARIB, Documentos: ARIB STD-T66 Autoridad de la aprobación: El ministerio de Poste y Telecomunicaciones, MPT.

América del Norte: Normas de la aprobación: Las Comunicaciones federales Comisionan, FCC, EE.UU., Documentos: CFR47, parta 15, Secciones 15.205, 15.209, 15.247 y 15.249, Normas de la aprobación: Industria Canadá, IC, Canadá, Documentos: GL36 Autoridad de la aprobación: FCC (EE.UU.), Industria Canadá (Canadá)

## 2 Banda de frecuencia y rangos del canal

El sistema de Bluetooth opera en la banda 2.4 GHz de ISM. Esta banda de frecuencia es de 2400 - 2483.5 MHz.

| Regulatory Range | RF Channels |
|---|---|
| 2.400-2.4835 GHz | f=2402+k MHz, k=0,…,78 |

*Tabla 2.1: Operación de las Bandas de Frecuencia*

Los canales de RF están espaciados 1 MHz y están ordenados en números de canales k como están mostrados en la Tabla 2.1. Para cumplir las regulaciones de out –of –band de cada país, una banda de guarda se usa al borde de la banda más baja y superior.

| Lower Guard Band | Upper Guard Band |
|---|---|
| 2 MHz | 3.5 MHz |

*Tabla 2.2: Bandas de Guarda*

## 3 Características del transmisor

Los requerimientos descritos en esta sección se dan como niveles de potencia al conector de la antena del dispositivo Bluetooth. Si el dispositivo no tiene un conector, se supone una antena de referencia con 0dBi de ganancia.

Debido a la dificultad de exactitud de la medida en medidas radiadas, un sistema con una antena integral pueden proveer un conector de antena temporal durante un modelo aprobado.

Si las antenas de transmisión de ganancia direccional mayor 0 dBi son usadas, los párrafos aplicables son EN 300 328, EN 301 489-17and FCC parte 15 y serán compensado por.

El dispositivo es clasificado en tres clases de potencia.

| Power Class | Maximum Output Power (Pmax) | Nominal Output Power | Minimum Output Power[1] | Power Control |
|---|---|---|---|---|
| 1 | 100 mW (20 dBm) | N/A | 1 mW (0 dBm) | Pmin<+4 dBm to Pmax<br>Optional:<br>Pmin[2] to Pmax |
| 2 | 2.5 mW (4 dBm) | 1 mW (0 dBm) | 0.25 mW (-6 dBm) | Optional:<br>Pmin[2)] to Pmax |
| 3 | 1 mW (0 dBm) | N/A | N/A | Optional:<br>Pmin[2)] to Pmax |

*Tabla 3.1: Clase de potencia*[1]

La clase de potencia del dispositivo 1 lleva a cabo un control de potencia. El control de potencia se usa para limitar la potencia de transmisión sobre los +4 dBm. La capacidad del control de potencia bajo +4 dBm es optativo y podría usarse para perfeccionar el consumo de poder y el nivel de la interferencia global. Las medidas de potencia formarán una secuencia monótona con un tamaño de medida máximo de 8 dB y un tamaño de medida mínimo de 2 dB. La clase de potencia del dispositivo 1 como máximo transmite una potencia de +20 dBm y será capaz de controlar su potencia de transmisión abajo de 4 dBm o menor.

Los dispositivos con capacidad de control de potencia perfeccionan la potencia de salida en un enlace físico con comandos LMP (vea Link Manager Protocol). Esto se hace midiendo RSSI e informando si el poder de la transmisión se aumentará o disminuye posiblemente.

En una conexión, el poder de salida no excederá la potencia de salida máximo de 2 clases de potencia para transmitir paquetes si el dispositivo receptor no soporta los mensajes necesarios para envíar mensajes de control de potencia, vea Link Manager Protocol. En este caso, el dispositivo de transmisión obedecerá las reglas de un dispositivo clase 2 o clase 3.

Si un dispositivo clase 1 está en paging o inquiry muy cerca de otro dispositivo, la entrada de potencia puede ser más grande que el requerimiento de la sección 4.5. Esto puede causar que el dispositivo receptor falle y no responda. Puede ser por consiguiente útil para page en la Clase 2 o 3 además de paging en la clase 1.

[1] 1) La potencia de salida mínima en la colocación máxima de la potencia.
2) El límite más bajo de la potencia Pmin <-30dBm se sugiere pero no es obligatorio, y puede ser escogido según las necesidades de aplicación.

## 3.1 Características de Modulación

La Modulación es GFSK (Gaussian Frequency Shift Keying) con un ancho de banda producto del periodo BT=0.5. El índice de la Modulación estará entre 0.28 y 0.35. Un 1 binario será representado por una desviación de frecuencia positiva, y un cero binario será representado por una desviación de frecuencia negativa. El símbolo de tiempo será menor a ±20 ppm.

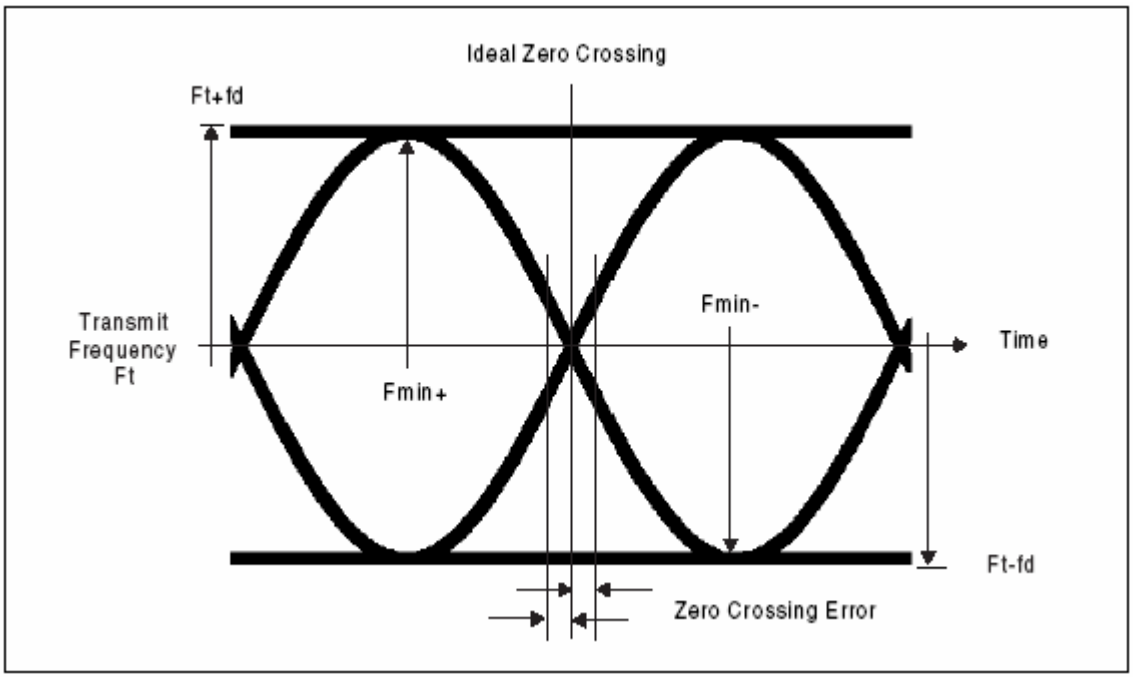



*Figura 3.1: Definición de los parámetros GFSK.*

Para cada transmisión, la desviación de frecuencia mínima, Fmin = el min. {|Fmin+ |, Fmin -} que corresponde a 1010 secuencias que no serán más pequeñas que ±80% de la desviación de frecuencia (fd) con respecto a la frecuencia transmitida Ft que corresponde a una secuencia 00001111.

Además, la desviación frecuencia mínima nunca será más pequeña que 115kHz. Los datos transmitidos tienen un symbol rate de 1 Ms/s.

El Zero Crossing Error es la diferencia de tiempo entre el periodo del symbol ideal y el tiempo de duración del crossing. Esto será menor que ± 1/8 de un periodo del symbol.

## 3.2 SPURIOUS EMISSIONS (falsas).

En la banda spurious emisiones se medirán frecuencias de radio hopping transmitiendo en un canal RF y recibiendo en un segundo canal RF; esto significa que el sintetizador puede cambiar de canal RF, entre la recepción y transmisión, pero siempre regresara al mismo canal RF de transmisión. No hay referencia en este documento a la salida de la banda ISM de las emisiones spurious; el fabricante del equipo es responsable de obedecer en determinado país su uso.

### 3.2.1 In-band spurious emissions.

Dentro de la banda ISM el transmisor pasará una máscara del espectro, dada en la tabla 3.2. El espectro cumple con los 20dB de ancho de banda definidos en FCC parte 15.247 y se medirá por consiguiente. Además de los requerimientos FCC un canal de potencia adyacente en canales adyacentes con una diferencia en el número de canales RF, dos o más son definidos. Este canal de potencia adyacente se define como la suma de las potencias en 1 MHz del canal RF. La potencia de transmisión se medirá en un ancho de banda de 100kHz usando el hold máximo. El dispositivo transmitirá en un canal RF - M y el canal de potencia adyacente se medirá en un numero de canales RF- N.
El transmisor transmitirá pseudo-aleatoriamente datos en el payload a lo largo de la prueba.

Las excepciones se permiten bajo las tres bandas de 1MHz centradas en una frecuencia qué es un entero múltiplo de 1 MHz. Ellos cumplirán con un valor absoluto de -20 dBm.

| Frequency offset | Transmit Power |
|---|---|
| ± 500 kHz | -20 dBc |
| 2MHz (\|M-N\| = 2) | -20 dBm |
| 3MHz or greater (\|M-N\| ≥ 3) | -40 dBm |

*Tabla 3.2: Transmita máscara del Espectro.*

Nota: Si la potencia de salida es menor a 0dBm, entonces dondequiera que se destine, los 20dB de FCC son un requisito relativo sobre las reglas del canal de potencia adyacente absoluto, declarado en la tabla anterior.

## 3.3 Tolerancia de frecuencia de radio

La frecuencia central inicial transmitida estará dentro de ±75 Khz. de $F_c$. La frecuencia inicial exacta es definida para ser la frecuencia exacta antes de que

cualquier paquete de información sea transmitida. Nota: la frecuencia drift requerida no es incluida en los ±75 Khz.

Los límites en la frecuencia drift de la transmisión central dentro de un paquete se especifica en la Tabla 3.3. Los diferentes paquetes son definidos en la Especificación Baseband.

| Duration of Packet | Frequency Drift |
|---|---|
| Max length one slot packet | ±25 kHz |
| Max length three slot packet | ±40 kHz |
| Max length five slot packet | ±40 kHz |
| Maximum drift rate[1] | 400 Hz/µs |

*Tabla 3.3: La frecuencia admisible máxima drifts en un paquete*[2].

## 4 Características del Receptor

Las características del receptor serán medidas usando loopback.
El nivel de sensibilidad de referencia en este capítulo es -70 dBm.

### 4.1 Nivel de Sensibilidad Real

El nivel de sensibilidad real se define como el nivel a la entrada para que un bit error rate (BER) de 0.1% sea recibido. La sensibilidad del receptor será debajo o igual a - 70 dBm con cualquier transmisor Bluetooth cumpliendo la especificación del transmisor especificado en la Sección 3.

### 4.2 Funcionamiento de la Interferencia

El funcionamiento de la interferencia en el Co-canal y adyacente de 1 MHz y 2 MHz se medirá con el señal requerida de 10 dB encima del nivel de sensibilidad de referencia. Para el funcionamiento de la interferencia en todos los otros canales RF la señal requerida será de 3 dB encima del nivel de sensibilidad de referencia.

Si la frecuencia de una señal de interfencia estuviera fuera de la banda de 2400-2483,5 MHz, la especificación out-of-band blocking (vea Sección 4.3) será

[2] 1) El máximo drift rate se permite en cualquier parte en un paquete.

aplicada. La señal de interferencia Bluetooth-modulada será (vea sección 4.7). El BER será de 0.1% para todas las señales de interferencia listadas en la Tabla 4.1

Estas especificaciones sólo serán probadas a las condiciones de temperatura nominales con un dispositivo que recibe en un canal RF y transmitiendo en un segundo canal de RF; esto significa que el sintetizador puede cambiar de canal RF entre la recepción y la transmisión, pero siempre regresa al mismo canal RF de recepción.

| Frequency of Interference | Ratio |
|---|---|
| Co-Channel interference, $C/I_{co\text{-}channel}$ | 11 dB |
| Adjacent (1 MHz) interference, $C/I_{1MHz}$ | 0 dB |
| Adjacent (2 MHz) interference, $C/I_{2MHz}$ | -30 dB |
| Adjacent (≥3 MHz) interference, $C/I_{\geq 3MHz}$ | -40 dB |
| Image frequency Interference[1 2], $C/I_{Image}$ | -9 dB |
| Adjacent (1 MHz) interference to in-band image frequency, $C/I_{Image\pm 1MHz}$ | -20 dB |

*Tabla 4.1: Función de interferencia*[3]

Los canales RF donde los requisitos no se encuentran son llamados canales RF de repuesta spurious. Cinco respuestas spurious en los canales RF se permiten en los canales de RF con una distancia = 2 MHz de la señal requerida. En estos canales RF de respuesta spurious un requisito de interferencia baja será C/I = -17 dB.

## 4.3 Out-of-Band Blocking

La out - of - band supresión (o rechazo) se medirá con una señal requerida de 3 dB encima del nivel de sensibilidad de referencia. La señal de interferencia será

[3] 1. In-band image frequency
2. Si la frecuencia de la imagen es diferente n*1 MHz, entonces la frecuencia de referencia de imagen se define como la más cercana n*1 MHz de frecuencia. Si dos canales adyacentes especificados en la Tabla 4,1 son aplicables al mismo canal, la especificación más discreta aplica.

una señal de onda continúa. El BER será de 0.1%. El out-of-band-blocking deberá cumplir con los requisitos siguientes:

| Interfering Signal Frequency | Interfering Signal Power Level |
|---|---|
| 30 MHz - 2000 MHz | -10 dBm |
| 2000 - 2399 MHz | -27 dBm |
| 2484 – 3000 MHz | -27 dBm |
| 3000 MHz – 12.75 GHz | -10 dBm |

*Tabla 4.2:Requerimientos Out-of-band suppression (or rejection).*

se permiten 24 excepciones qué son dependientes en el canal de RF dado y centrado a una frecuencia que es un entero múltiplo de 1 MHz. Para menor a 19 de estas frecuencias de respuestas spurious, un nivel de la interferencia reducido, de por lo menos -50dBm, se permite lograr los requerimientos de funcionamiento BER=0.1%, considerando que para un máximo de 5 de las frecuencias spurious el nivel de interferencia puede asumirse arbitrariamente bajo.

## 4.4 Características de Inter-modulación

El funcionamiento de sensibilidad de referencia, BER = 0.1%, se conoce bajo las siguientes condiciones:

La señal requerida será a una frecuencia $f_0$ con un nivel de potencia de 6 dB debajo del nivel de sensibilidad de referencia.

Una señal de onda senoidal estática estará en una frecuencia $f_1$ con un nivel de potencia de -39 dBm.

Una señal modulada Bluetooth (vea Sección 4.7), estará $f_2$ con un nivel de potencia de -39 dBm.

Las Frecuencias $f_0$, $f_1$ y $f_2$, se escogerán tal que $f_0=2\ f_1 - f_2$ y $|f_2 - f_1| = n*1MHz$, donde n puede ser 3, 4, o 5. El sistema cumplirá uno de las tres alternativas por lo menos (n=3,4, o 5).

## 4.5 Máximo Nivel Utilizable

El máximo nivel de entrada utilizable al que el receptor opera será mayor de -20 dBm. El BER será menor o igual a 0.1% a -20 dBm de potencia de entrada.

## 4.6 Indicador de Fuerza de la Señal de Receptor

Si un dispositivo soporta un Indicador de la fuerza de la señal de receptor (Receiver Signal Strength Indicador RSSI por sus siglas en ingles) hace posible que un transceptor que desea tomar parte en la conexión power-controlled debe ser capaz de medir su fuerza de la señal del receptor y determinar si el transmisor en el otro lado de la conexión debe aumentar o debe disminuir su nivel de salida de la potencia. La manera de controlar la potencia especifica deberá tener un **golden receive power range** Este golden receive power se define como un rango con un bajo y alto nivel umbral y más alto del límite . El nivel más bajo del umbral corresponde a un poder recibido entre -56 dBm y 6 dB encima de la verdadera sensibilidad del receptor. El nivel superior del umbral es 20 dB encima del nivel más bajo de umbral con una precisión de + /- 6 dB. Las instrucciones para alterar el poder de TX se lleva en la conexión de LMP.

## 4.7 Definición de la Señal de Referencia

Una señal de interferencia modulada Bluetooth se definirá como:
Modulación = GFSK
Índice de la modulación = 0.32±1%
BT = 0.5±1%
Bit Rate = 1 Mbps ±1 ppm
Datos modulados para el señal requerida = PRBS9
Datos modulando por la señal de interferencia = PRBS 15
Exactitud de frecuencia mejor que ±1 ppm.

# CAPITULO II

# PARTE B
# BASEBAND

## 1 Descripción General

Esta parte especifica el funcionamiento normal de una baseband Bluetooth.

El sistema Bluetooth proporciona una conexión punto-a-punto o una conexión punto-a-multipunto, vea (a) y (b) en Figura 1.1. En una conexión punto-a-punto el canal físico es compartido entre dos dispositivos Bluetooth. En una conexión punto-a-multipunto, el canal físico es compartido entre varios dispositivos de Bluetooth. Dos o más dispositivos que comparten el mismo canal físico forman un piconet. Un dispositivo Bluetooth actúa como master del piconet, considerando que el otro dispositivo actúa como slave(s). Hasta siete slaves pueden estar activos en el piconet. Adicionalmente, muchos más slaves pueden permanecer conectados en un estado parked. Estos slaves parked no están activos en el canal, pero permanecen sincronizados al master y puede ponerse activos sin usar el procedimiento del establecimiento de la conexión. El acceso al canal es controlado por el master.

Piconets que tienen dispositivos comunes son llamados scatternet, vea (c) en Figura 1.1. Cada piconet sólo tiene un solo master, sin embargo, los slaves pueden participar en piconets diferentes en una multiplexión por división de tiempo base. Además, un master en un piconet puede ser un slave en otro piconets. Los Piconets no estarán en frecuencia sincronizada y cada piconet tiene su propia secuencia hopping.

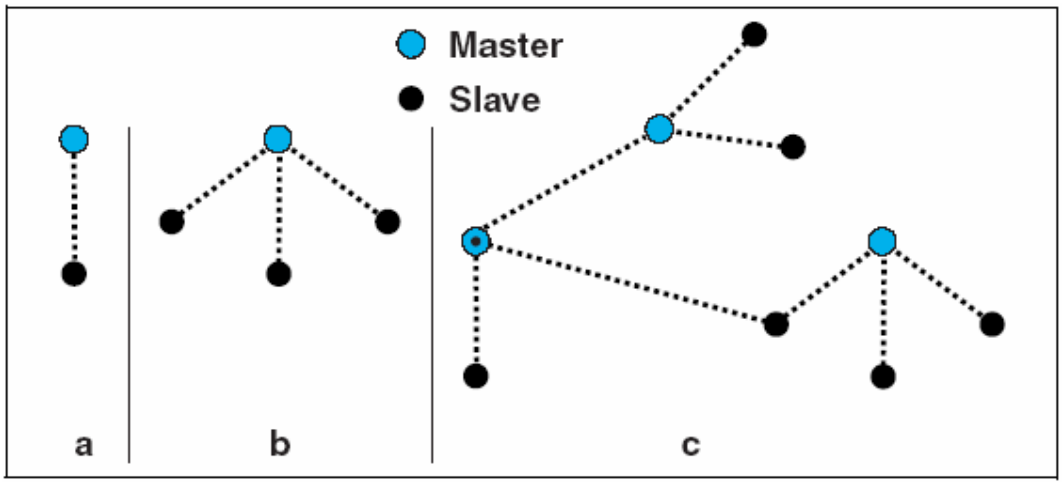


Figura 1.1: Piconets con un solo funcionamiento del slave (a), un funcionamiento del multi-slave (b) y un funcionamiento scatternet (c).

Los datos se transmiten sobre el aire en paquetes. El formato del paquete general se muestra en la Figura 1.2. Cada paquete consiste en 3 entidades: el código de acceso, el header, y el payload.

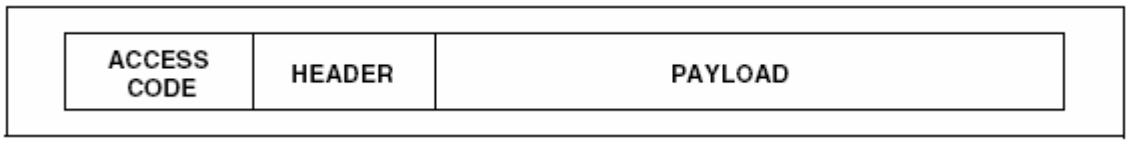



Figura 1.2: El formato del paquete Normal.

## 1.1 Reloj Bluetooth

Cada dispositivo Bluetooth tendrá un reloj nativo del que se derivará un reloj del sistema corriente. Para la sincronización con otros dispositivos, compensando usualmente, cuando se agregan al reloj nativo, proporciona relojes Bluetooth temporales que son mutuamente sincronizados. Debe notarse que el reloj de bluetooth no tiene ninguna relación al tiempo de día; puede inicializarse por consiguiente en cualquier valor. El reloj tiene un ciclo de aproximadamente de un día. Si el reloj se implementa con un contador, un contador de 28-bit se requiere alrededor de $2^{28}$-1. El bit menos significativo (LSB) marcara una señal en unidades de 312.5 µs (por ejemplo en un periodo de time slot), dando un rate al reloj de 3.2 Khz.

El reloj determina los periodos críticos y trigger a los eventos en el dispositivo. Cuatro periodo son importantes en el sistema Bluetooth: 312.5 µs, 625 µs, 1.25 ms, y 1.28 s; estos periodos corresponden al reloj de bits CLK0, CLK1, CLK2, y CLK12, respectivamente, vea Figura 1.3.

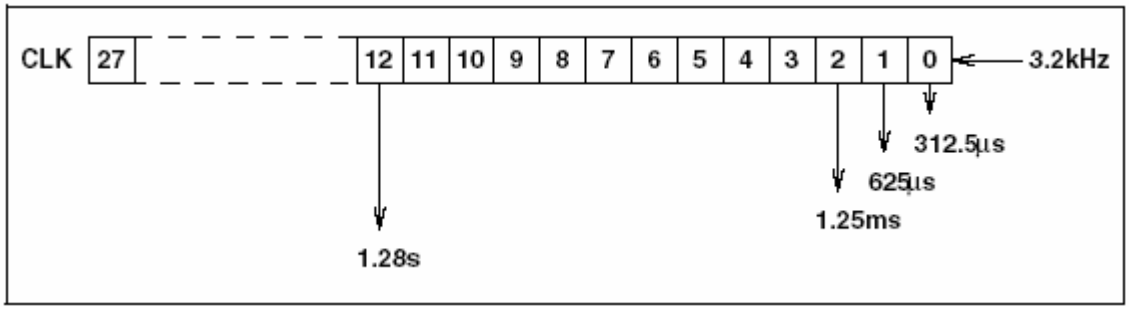



Figura 1.3. El reloj Bluetooth.

En los diferentes modos y estados un dispositivo puede residir en el reloj y tiene diferente apariencias:

- CLKN reloj nativo
- CLKE reloj estimado
- CLK reloj master

CLKN es el reloj nativo y será la referencia a todas las otras apariencias del reloj. En standby y parked, modos hold y sniff el reloj nativo puede estar manejando un oscilador de baja potencia (LPO) con peor exactitud del caso de (+/-250ppm). Por otra parte, el reloj nativo será manejado por la referencia en el oscilador de cristal con peor exactitud del caso de +/-20ppm. Vea Sección 2.2.4 para la definición de CLK y Sección 2.4.1 y la definición de CLKE. El master nunca ajustará su reloj nativo durante la existencia en el piconet.

## 1.2 Dispositivo Bluetooth Addressing

Cada dispositivo Bluetooth se asignará una dirección única de 48-bits (BD_ADDR). Esta dirección se obtendrá del Registro autorizado de IEEE. La dirección es dividida en lo siguiente tres campos:

- LAP campo: parte de dirección más baja que consiste en 24 bits
- UAP campo: parte de dirección superior que consiste en 8 bits
- NAP campo: parte de dirección no-significante que consiste en 16 bits

El LAP y UAP forman la parte significante del BD_ADDR. El modelo de bit en la Figura 1.4 es un ejemplo BD_ADDR.

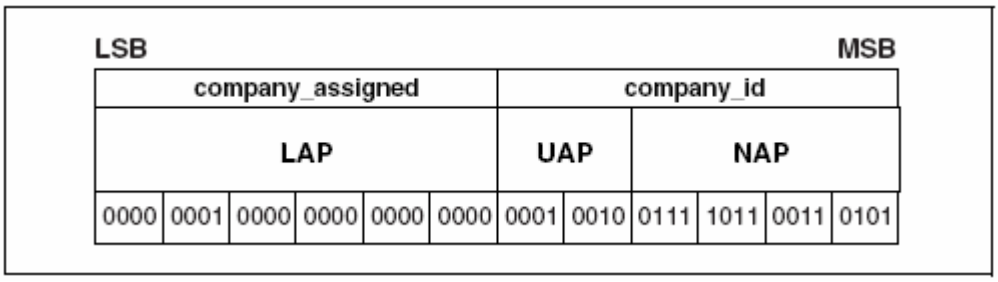



Figura 1.4 Formato de BD_ADDR

El BD_ADDR puede tomar cualquier valor excepto los 64 valores reservados para LAP y dedicadas en inquiries generalmente (vea Sección 1.2.)

### 1.2.1 Direcciones Reservadas

Un bloque de 64 bits contiguos LAPs son reservados para la operación de inquiry; un LAP comúnmente es reservado a todos los dispositivos inquiry, el resto de los 63 LAPs, son reservados para dedicar a inquiry, clases específicas de dispositivos (vea Asignación de Números en los website[4]). Los mismos valores LAP se usan sin tener en cuenta los volúmenes de UAP y NAP. Por consiguiente, ninguno de estos LAPs puede ser parte de un usuario BD_ADDR.

Las direcciones LAPs reservadas son 0x9E8B00-0x9E8B3F. El inquiry general LAP es 0x9E8B33. Todas las direcciones tienen el LSB al rightmost posición, notación hexadecimal. El default check intialization (DCI) se usa como el UAP siempre que se use una de las direcciones reservadas LAP. El DCI se define por 0x00 (hexadecimal).

## 1.3 Códigos de Acceso

En el sistema Bluetooth todas las transmisiones sobre el canal físico comienzan con un código de acceso. Se definen tres códigos de acceso diferentes, vea Sección 6.3.1

- Dispositivo de código de acceso (DAC)
- Código de acceso de canal (CAC)
- Código acceso Inquiry (IAC)

Todos los códigos de acceso se derivan del LAP de un dispositivo de dirección y dirección inquiry. El código de acceso de dispositivo es usado durante page, page scan y page response substates y se derivará del BD_ADDR del dispositivo Paged.
El código de acceso de canal es usado en el estado de CONEXION y forma el comienzo en todos los paquetes intercambiados en el canal físico piconet. El código de acceso de canal se derivará del LAP del BD_ADDR del master. Finalmente, el código de acceso inquiry se usará en el substate inquiry. Hay generalmente un IAC (GIAC) para la operación general de inquiry y hay 63 IACs (DIACs) dedicados para la operación de inquiry.

El código de acceso también indica al receptor la llegada de un paquete. Se usa para cronometrar sincronizar y compensar. El receptor correlaciona la palabra de sincronización contra el código de acceso, proporcionando una señalización robusta.

[4] https://www.bluetooth.org/foundry/assignnumb/document/assigned_numbers

## 2 Canales Físicos

La capa arquitectónica más baja en el sistema Bluetooth es el canal físico. Se definen varios tipos de canal físico. Todos los canales físicos Bluetooth son caracterizados por la combinación de una a secuencia pseudo-aleatoria de salto de frecuencia (hopping), el slot timing específico de las transmisiones, el código de acceso, y paquete header codificado. Estos aspectos, junto con el rango de transmisión, define la sintonía del canal físico. Para el piconet básico y adaptado, los canales frecuencia hopping se usan para cambiar la frecuencia periódicamente, reducir los efectos de interferencia y satisfacer los requisitos de regulación local.

Dos dispositivos que desean comunicarse usan un canal físico compartido para esta comunicación. Llevar a cabo esto, sus transceivers deben ponerse a un punto en Frecuencia RF al mismo tiempo, y deben estar dentro de un rango nominal cada uno.

Dado que el número de portadoras de RF está limitado y muchos dispositivos Bluetooth pueden estar operando independientemente dentro del mismo espacio y área temporal, hay una probabilidad fuerte que dos dispositivos Bluetooth independientes tengan su transceivers sintonizados a la misma portadora de RF, produciendo una colisión en el canal físico. Para mitigar los efectos no deseados de esta colisión cada transmisión comenzara con un código de acceso que se usara como una correlación codificada por los dispositivos sintonizados en el canal físico. Este código de acceso del canal es una propiedad del canal físico. El código de acceso siempre está presente al comienzo de cada paquete transmitido.

Cuatro canales físicos Bluetooth se definen. Cada uno se perfecciona y se usa para un propósito diferente. Dos de estos canales físicos (el canal piconet básico y adaptado) se usa para comunicación entre dispositivos conectados y es asociado con un piconet específico. El resto del canal físico se usa para descubrir (el canal inquiry scan) y conectar (el canal page scan) los dispositivos Bluetooth.

Un dispositivo Bluetooth puede usar sólo uno de estos canales físicos en cualquiera momento. Para soportar múltiples operaciones concurrentes en el dispositivo, usa multiplexión por división - tiempo entre los canales. De esta manera un dispositivo Bluetooth puede operar simultáneamente en varios piconets, así como estar descubriendo y conectando.

Siempre que un dispositivo Bluetooth se sincronice en tiempo, frecuencia y código de acceso de un canal físico se dice que esta conectado a este canal (si esta o no activamente involucrado en comunicaciones sobre el canal.) Por lo menos, un dispositivo necesita ser capaz de conexión a un canal físico en un momento, sin embargo, los dispositivos avanzados pueden ser capaces de conectarse simultáneamente a más de un canal físico, pero la especificación no asume que esto sea posible.

## 2.1 Definición del Canal Físico

Los canales físicos son definidos por un canal RF, secuencia pseudo-aleatoria hopping, el paquete (slot) tiempo y un código de acceso. La secuencia hopping es determinada por el UAP y LAP de un dispositivo Bluetooth de dirección y de secuencia hopping seleccionada. La fase en la secuencia hopping es determinada por el reloj Bluetooth. Todos los canales físicos se subdividen en los time slot donde la longitud es diferente dependiendo del canal físico. Dentro del canal físico, cada recepción o evento de transmisión es asociado con un time slot o time slots. Para cada recepción o transmisión un canal RF es seleccionado por el hop selección kernel (vea Sección 2.6), la máxima hop rate es 1600 hops/s en un estado de CONEXION y el máximo en inquiry y page substates es 3200 hops/s.
Se definen los siguientes canales físicos:

- Canal físico piconet básico
- Canal físico piconet adaptado
- Canal físico page scan
- Canal físico inquiry scan

## 2.2 Canal Físico Piconet Básico

Durante el estado de CONEXION el canal físico piconet básico es usado por default. El canal físico piconet adaptado también puede usarse. El canal físico piconet adaptado es idéntico al canal físico piconet básico salvo las diferencias listadas en Sección 2.3.

### 2.2.1 Definición del Master-Slave

El canal físico piconet básico es definido por el master del piconet. El master controla el tráfico en el canal físico piconet básico por un esquema polling. (Vea Sección 8.5) Por definición, el dispositivo que comienza una conexión por paging es el master. Una vez que un piconet se ha establecido, pueden intercambiar papeles, master-slave. Esto se describe en la Sección 8.6.5

### 2.2.2 Características Hopping

En el canal físico piconet básico se caracteriza un hopping pseudo-aleatorio a través de los 79 canales RF. La frecuencia que brinca en el canal físico piconet básico es determinada por el reloj de Bluetooth y BD_ADDR del master. Cuando el piconet se establece, el reloj del master se comunica a los slaves. Cada slave agregará una compensación a su reloj nativo sincronizado con el reloj master. Puesto que los relojes son independientes, las compensaciones deben

actualizarse regularmente. Todos los dispositivos que participan en el piconet están en un tiempo-sincronizado y un hopping-sincronizado en el canal.

El canal físico piconet básico usa el canal básico de secuencia hopping que se describe en la Sección 2.6.

### 2.2.3 Time Slots

El canal físico piconet básico es dividido en time slots, cada uno es de 625 µs de longitud. Los time slots se enumeran según los 27 bits mas significativos del reloj Bluetooth CLK28-1 del piconet master. El time slot enumera rangos de 0 a $2^{27}$-1 y es cíclico, con un ciclo de longitud de $2^{27}$. El número de time slot es denotado como k.

Un esquema de TDD es usado cuando un master y un slave transmiten alternativamente, vea la Figura 2.1. El comienzo del paquete se alineará con el slot de comienzo. Los paquetes pueden extenderse hasta cinco time slot.

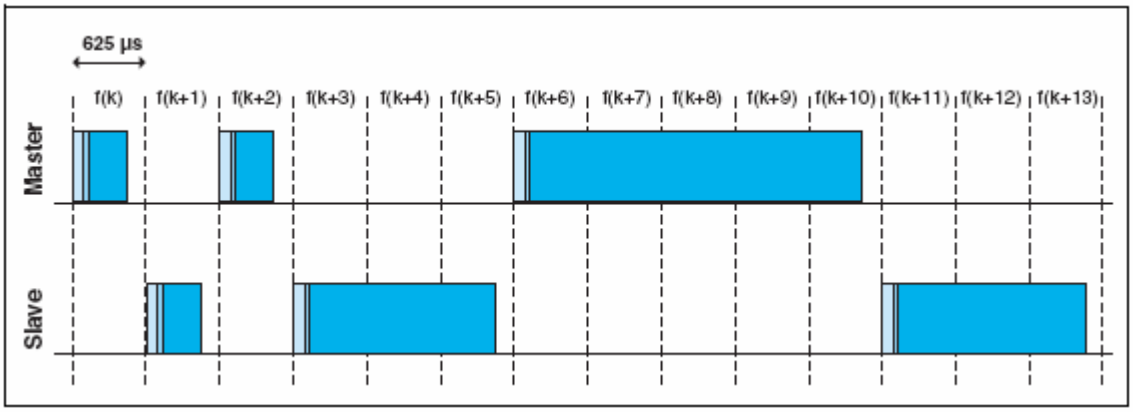


Figura 2.1. Paquetes Multi-slot

El término slot pairs es usado para indicar dos time slot starting adyacentes con un slot de transmisión de master-a-slave.

### 2.2.4 Relojes del Piconet

CLK es el reloj del piconet master. Este es usado por todas las actividades timing y scheduling en el piconet. Todos los dispositivos usan el CLK para fijar su transmisión y recepción. El CLK se derivará del reloj nativo CLKN (vea Sección 1.1) agregando una compensación, vea Figura 2.2. La compensación será cero para el master desde que CLK es idéntico a su propio reloj nativo CLKN. Cada slave agregará una compensación apropiada a su CLKN tal que el CLK corresponda al CLKN del master. Aunque todos los CLKN en unos dispositivos corran a la misma proporción nominal, la tendencia mutua causa inexactitudes en el CLK. Por consiguiente, las compensaciones en los slaves deben ponerse al día regularmente, tal que, CLK es aproximadamente el CLKN del master.

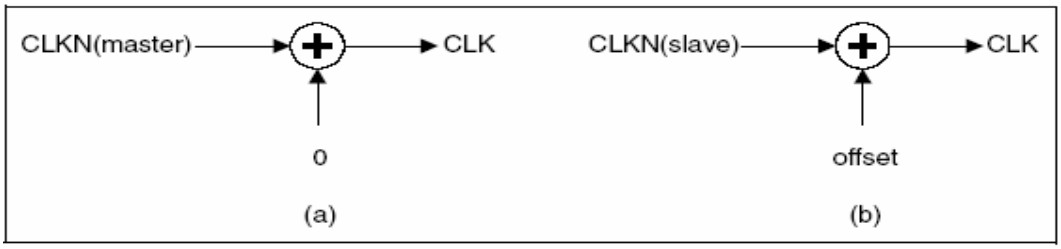



Figura 2.2: La Derivación de CLK en master (a) y en slave (b).

## 2.2.5 Transmisión/recepción timing

La transmisión master siempre empezará en time slots pares ($CLK_1$=0) y la transmisión del slave siempre empezará en time slots nones ($CLK_1$=1). Debido a los tipos de paquetes que cubren más que un simple slot, la transmisión master puede continuar en slots impares y la transmisión del slave puede continuar incluso en slots pares, vea Figura 2.1.

Todos los diagramas timing mostrados en este capítulo son basado en las señales como se presenta en una antena. El término "exact" se usa cuando se describe timing y se refiere a una transmisión o recepción ideal y fuera del timing jitter e imperfecciones en el reloj de frecuencia. El promedio timing de la transmisión de paquetes no fluirá (drift) más rápido que 20 ppm relativos al slot timing ideal de 625 µs. El timing instantáneo no se desvía más de 1 µs del timing promedio. Así, el timing de la transmisión de paquetes absoluta $t_k$ del slot limite *k* cumplirá la siguiente ecuación:

$$t_k = \left( \sum_{i=1}^{k} (1 + d_i) T_N \right) + j_k + \text{offset},$$

(EQ 1)

Donde $T_N$ es la longitud del slot nominal (625 µs), jitter denota ($| j_k | \leq 1$ µs) al comenzar el slot, *k* y $d_k$ denota el flujo ($| d_k | \leq 20$ ppm) dentro del slot *k*. El jitter y el drift pueden variar arbitrariamente dentro de los límites dados para cada slot, mientras offset es arbitrario pero una constante fija. Para el soporte, de park y sniff los parámetros drift y jitter son especificados en el Protocolo Link Manager

### 2.2.5.1 Timing del Canal Físico Piconet

En las figuras, se muestran sólo paquetes de slot simples como un ejemplo.
El TX/RX timing master se muestra en Figura 2.3. En la Figura 2.3 y la Figura 2.4 el canal de la frecuencias hopping es indicado por f (k) donde k es el número del time slot. Después de la transmisión, se espera un paquete de retorno N × 625 µs, después de comenzar el paquete TX donde N es un impar entero más grande que 0. N depende del tipo de los paquetes transmitidos.

Permitir un tiempo de espera en una ventana de incertidumbre, esta definido alrededor de “exact receive timing”. Durante el funcionamiento normal, la longitud de la ventana será 20µs, que permiten al paquete RX llegar anticipado 10 µs o 10 µs demasiado tarde. Se recomienda que la implementación del slave varié el tamaño de la ventana o el tiempo de rastreo para acomodar la ausencia del master a más de 250ms.

Durante el comienzo del ciclo RX, los correlatores de acceso buscarán el código de acceso del canal correcto, sobre la ventana de incertidumbre. Si un evento no ocurre provoca que el receptor este dormido hasta el próximo evento de RX. Si en el curso de la búsqueda, se pone claro que el rendimiento de la correlación nunca excede el umbral final, el receptor puede ir a dormirse antes. Si el evento ocurre, el receptor permanecerá abierto para recibir el resto del paquete a menos que el paquete sea para otro dispositivo, sea descubierto un error header no-recuperable, o un error del payload no-recuperable.

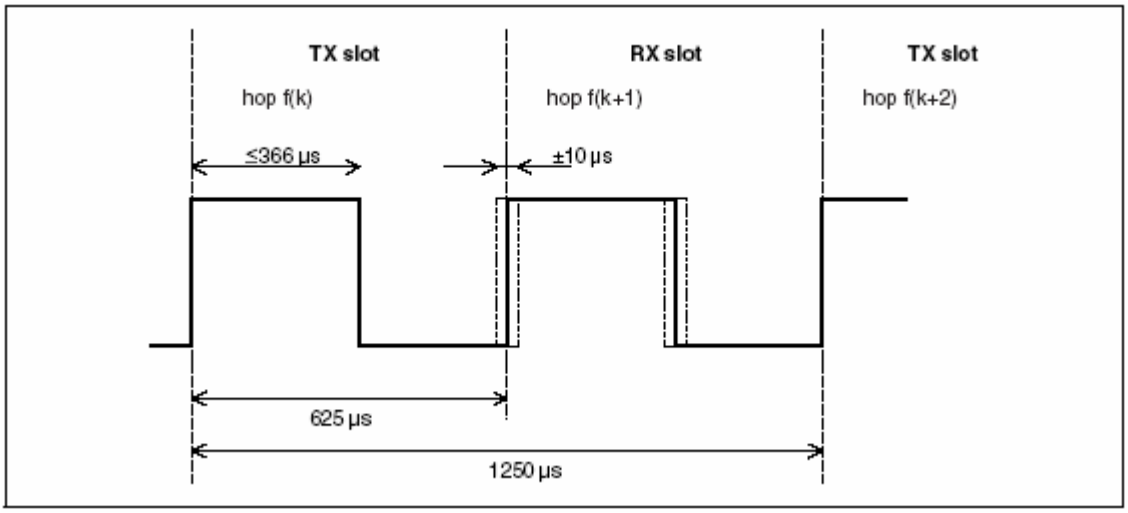



Figura 2.3: Ciclo de RX/TX del transerver master en modo normal para los paquetes single-slot.

Cada transmisión master se derivará del bit 2 del reloj nativo Bluetooth del master, así la transmisión actual se fijará Mx1250µs después de comenzar una ráfaga TX master anterior donde M depende en la transmisión y recepción del tipo de paquetes, y es un igual, al entero más grande que 0. El timing TX master se derivará del reloj Bluetooth nativo del master, y así no será afectado por el timing drifts en el slave(s).

Los slaves mantienen una estimación del reloj nativo del master agregando un timing offset al reloj nativo del slave (vea Sección 2.2.4). Este offset debe ser puesto al día cada vez que un paquete sea recibido del master. Comparando el exact RX timing del paquete recibido con el RX timing estimado, los slaves deben corregir el offset para cualquier timing misalignments (desalineación). Subsecuentemente sólo el código de acceso al canal se requiere para sincronizar

al slave, el timing RX slave pueden corregirse enviando cualquier paquete de transmisión del slot master-al-slave.

El TX/RX timing del slave se muestra en Figura 2.4 La transmisión del slave se fijará N × 625µs después de comenzar con el paquete RX del slave donde N es un entero impar, positivo más grande que 0; así será timing TX el timing drift RX del slave. Durante los periodos cuando un slave está en el modo activo (vea Sección 8.6) y no puede recibir algún código de acceso válido del master, el slave puede incrementar la recepción de la ventana de incertidumbre y/o el uso pronosticado del timing drift para aumentar la probabilidad de recepción en la ráfaga del master cuando se reanuda la recepción.

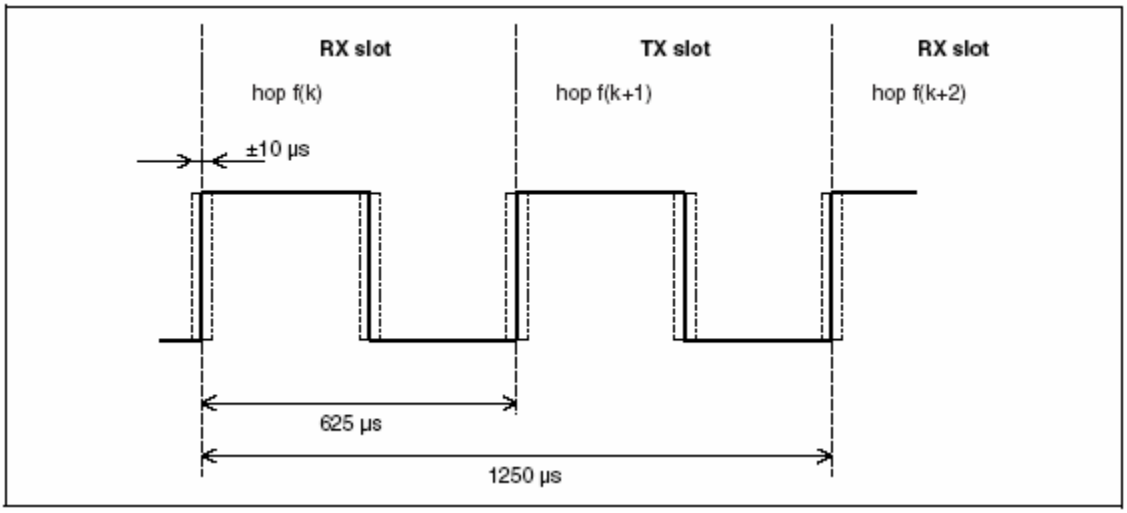



Figura 2.4: Ciclo de RX/TX de trans-recibidor del slave en modo normal para los paquetes single-slot.

## 2.2.5.2 Re-sincronización del canal físico Piconet

En el canal físico Piconet, un slave puede perder sincronización si no recibe un paquete del master por lo menos cada 250ms (o menos si se usa el bajo poder del reloj). Esto puede ocurrir en sniff, hold, park, en un scatternet o debido a la interferencia. Al re-sincronizar al canal físico piconet, un dispositivo slave escuchara al master antes de que pueda envíar información. En este caso, puede aumentarse la longitud de la ventana de la búsqueda en el dispositivo slave de 20µs a un valor más grande Xµs como se ilustró en la Figura 2.5. Note que solo la frecuencia hop RX es usada. La frecuencia hop usada del master -al-slave (RX) slot también se usará en la ventana de incertidumbre, incluso cuando se extiende precediendo en el intervalo de tiempo normalmente usado para el slot slave-a-master (TX).

Si la longitud de la ventana de búsqueda, X, excede 1250 µs, las ventanas consecutivas deben evitar solapar ventanas de búsqueda. Las ventanas consecutivas deben ser en cambio centradas al ***f (k), f (k+4),... f (k+4i)*** (donde "i"

es un entero) que da un valor máximo de X=2500µs, o incluso f (k), f (k+6),...f (k+6i) qué da un valor máximo de X=3750 µs. Las frecuencias hop RX usadas corresponden a la transmisión slot master -a-slave. Se recomienda que los paquetes single-slot sean transmitidos por el master durante la re-sincronización del slave.

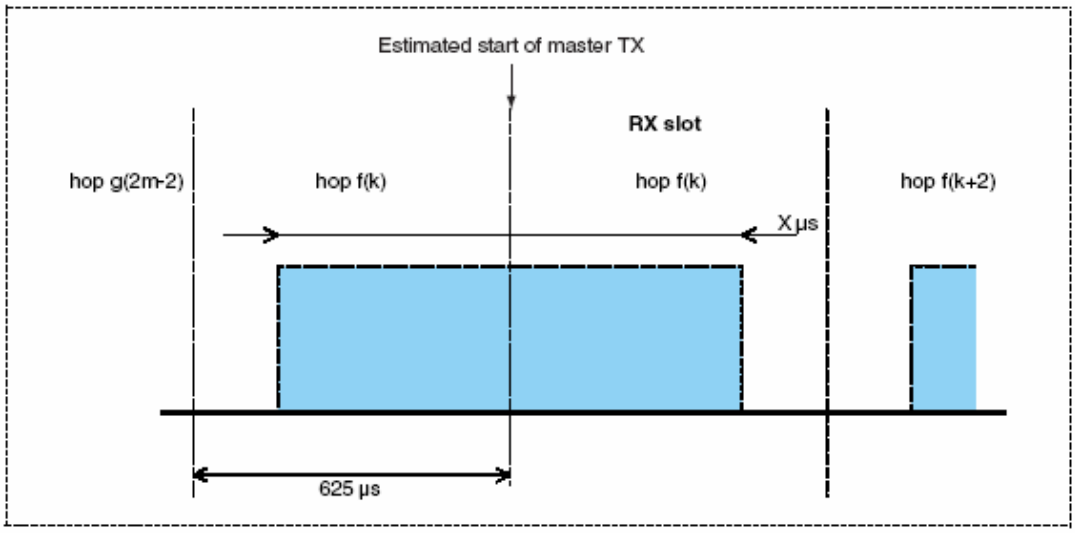



Figura *2.5: RX timing del slave regresando del modo hold.*

## 2.3 Canal Físico Piconet Adaptados

### 2.3.1 Características Hopping

El canal físico piconet adaptado usará por lo menos los canales RF $N_{min}$ (donde $N_{min}$ es 20). El canal físico piconet adaptado usa la secuencia hopping del canal adaptado descrito en la Sección 2.6

El canal físico piconet adaptado pueden usarse para los dispositivos conectados que tengan frecuencia hopping adaptable (AFH) habilitada. Hay dos distinciones entre canal físico piconet básico y adaptado. La primera distinción es que el mismo mecanismo del canal que hace la frecuencia del slave, es igual a la precedida en la transmisión del master. El segundo aspecto es que el canal físico piconet adaptado puede estar basado en menos de 79 frecuencias del canal físico piconet básico.

## 2.4 Page Scan del Canal Físico.

Aunque no se definen los papeles del master y slave previo a una conexión, el término master se usa para el dispositivo paging (éste se hace un master en el estado de CONEXIÓN) y el slave se usa para el dispositivo page scanning (este se convierte en un slave en el estado de CONEXION).

## 2.4.1 Reloj estimado para Paging

Un dispositivo paging usa una estimación del reloj nativo del dispositivo page scanning, CLKE; es decir un desplazamiento se agregará al CLKN del page para aproximar el CLKN del destinatario, vea Figura 2.6. CLKE será derivado de la referencia CLKN agregando un offset. Usando el CLKN del destinatario, el page podría acelerar el establecimiento de la conexión.

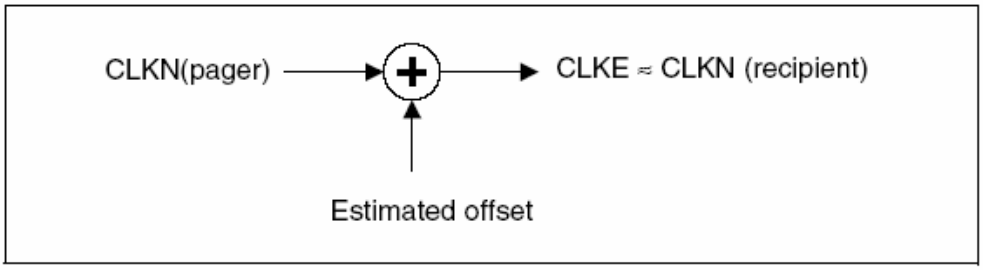



**Figura 2.6: Derivación de CLKE.**

## 2.4.2 Características Hopping

El Page scan del canal físico sigue un modelo hopping más lento que en el canal físico piconet adaptado y es una secuencia hopping pseudo-aleatoria corta a través de los canales RF. El timing del canal page scan puede ser determinado por el reloj Bluetooth nativo del dispositivo scanning. La secuencia de la frecuencia hopping es determinada por la dirección Bluetooth del dispositivo scanning.
El canal físico page scan usa el page, la respuesta master page, la respuesta slave page, y la secuencia hopping page scan especificadas en la Sección 2.6

## 2.4.3 Procedimiento Paging Timing.

Durante el procedimiento paging, el master transmitirá mensajes paging (vea la tabla 8.3) correspondiendo al slave que esta conectado. Desde el mensaje paging es un paquete muy corto, el hop rate es 3200 hops/s. En un intervalo single slot TX, el dispositivo paging transmitirá en dos frecuencias hop diferentes. En la Figura 2.7 a través de la Figura 2.11, f (k) se usa para las frecuencias de la secuencia page hopping y f '(k) denota las frecuencias de la secuencia de la respuesta page correspondiente. La primera transmisión empieza donde $CLK_0 = 0$ y la segunda transmisión empieza donde $CLK_0 = 1$.
En un intervalo single slot RX, el dispositivo paging atenderá el mensaje page response del slave en dos frecuencias hop diferentes. Similar a la transmisión, la recepción nominal empieza donde $CLK_0 = 0$ y la segunda recepción nominal comienza donde $CLK_0 = 1$; vea Figura 2.7. Durante el slot TX, el dispositivo paging enviará un mensaje paging a las frecuencias hop TX *f (k) y f (k+1)*. En el slot RX,

atenderá una respuesta o unas frecuencias hop RX correspondientes *f'(k) y f'(k+1).* Los periodos atendidos se cronometrarán exactamente a 625 µs después de los paquetes paging correspondientes, e incluirá una ventana de incertidumbre de ±10 µs.

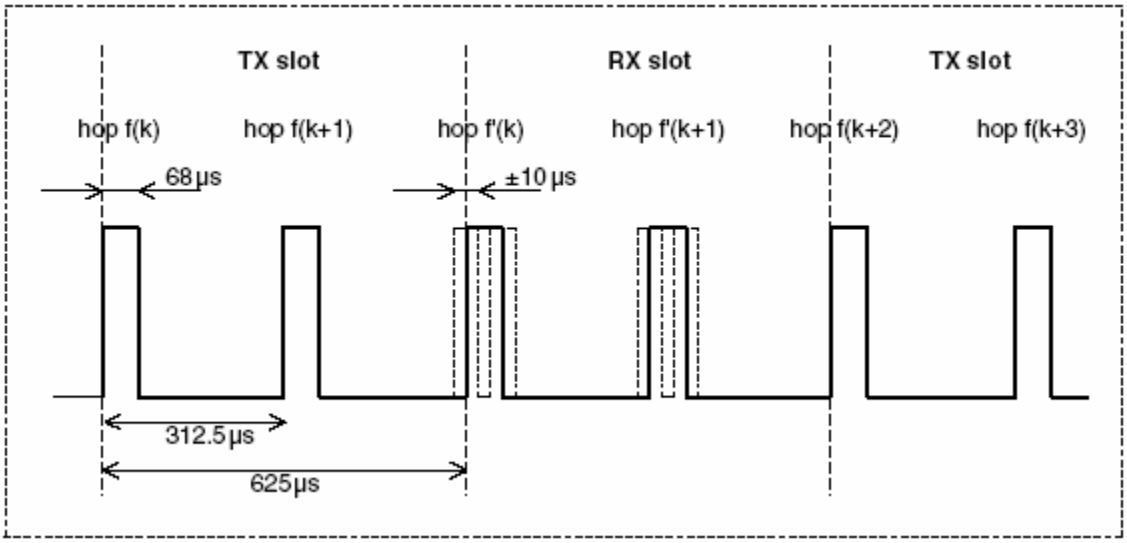



Figura 2.7: Ciclo RX/TX del transceiver en modo PAGE.

## 2.4.4 Page Response Timing

Al arreglo de la conexión, un paquete page response del master se transmite del master al slave (vea Tabla 8.3). Este paquete establece el timing y sincroniza la frecuencia. Después de que el dispositivo slave ha recibido el mensaje page, devolverá un mensaje de respuesta que consiste en el paquete page response del slave y sigue 625 µs después de recibir el mensaje page. El master enviará el paquete page response del master en el slot TX que sigue al slot RX en el que recibió la respuesta del slave, según el RX/TX timing del master. La diferencia de tiempo entre el paquete page response del slave y el paquete page response del master dependerán del timing del mensaje page que el slave recibió. En la Figura 2.8, el slave recibe el mensaje paging, primer envío en el slot del master-a-slave. Entonces responde con un primer paquete page response del slave en la primer mitad del slot del slave-a-master. El timing del paquete page response del master esta basado en el timing del primer envío del mensaje page en el slot precedido del master-a-slave: hay un retardo exacto de 1250 µs entre el primer mensaje page y el paquete page response del master. El paquete se envía a la frecuencia hop f (k+1) qué es la frecuencia hop siguiente a la frecuencia hop f (k) del mensaje page que se recibió.

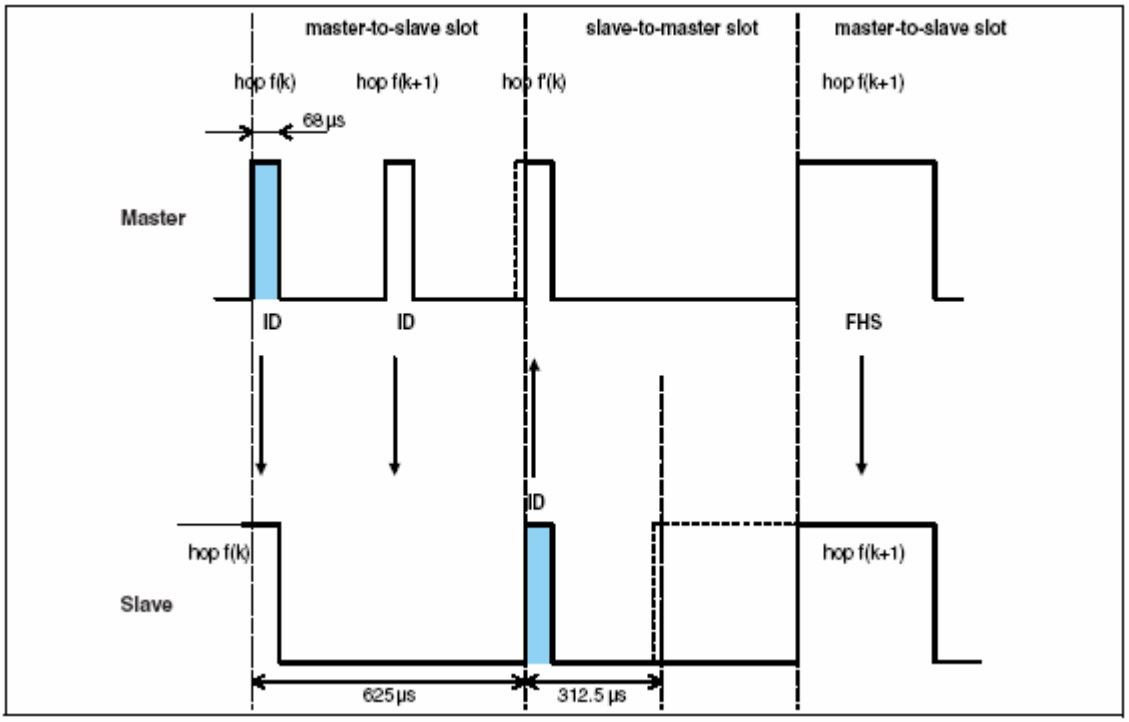



Figura 2.8: Timing del paquete page response en un page exitoso en el primer medio slot

En la Figura 2.9, el slave recibe el mensaje page enviado, el segundo envío en el slot del master-a-slave. Entonces responde con un paquete page response del slave en la segunda mitad del slot del slave-a-master exactamente 625 µs después de recibir el mensaje page. El timing del paquete page response del master está inmóvil basado en el timing del mensaje page del primer envío del mensaje page en el slot precedido del master-a-slave: hay un retardo exacto de 1250 µs entre el primer mensaje page y el paquete page response del master. El paquete se envía a la frecuencia hop f(k+2) qué es la frecuencia hop siguiente a la frecuencia hop f(k+1) en el mensaje page que se recibió.

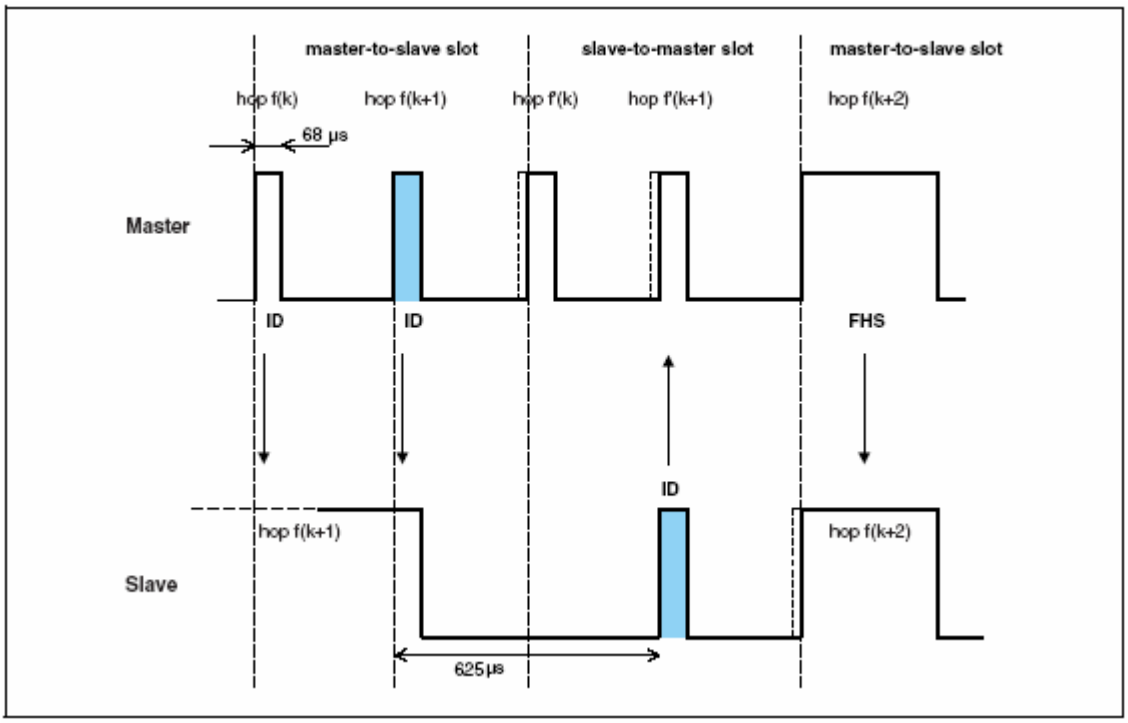



Figura 2.9: Timing del paquete page response en un page exitoso en el segundo medio slot

El slave ajustará su RX/TX timing según la recepción del paquete page response del master (y no según la recepción del mensaje page). Es decir, el segundo paquete page response del slave que reconoce la recepción del paquete page response del master se transmitirá 625 µs después de la salida del paquete page response del master.

## 2.5 Canal Físico Inquiry Scan

Aunque los papeles master y slave no se definen previo a una conexión, el término, master se usa para el dispositivo inquiring y el slave se usa para el dispositivo inquiry scan.

### 2.5.1 Reloj para inquiry

El reloj usado para la inquiry y inquiry scan será el reloj nativo del dispositivo.

### 2.5.2 Características Hopping

El canal inquiry scan sigue un modelo hopping más lento que el canal físico piconet y es una secuencia corta pseudo-aleatoria hopping a través del canal RF. El timing del canal inquiry scan es determinado por el reloj Bluetooth nativo del dispositivo scanning mientras la frecuencia de secuencia hopping es determinada por el código de acceso general de inquiry.

El canal físico inquiry scan usa inquiry, inquiry response, y secuencia hop inquiry scan descritas en la Sección 2.6.

### 2.5.3 Procedimiento Inquiry Timing

Durante el procedimiento inquiry, el master transmitirá mensajes inquiry con el código de acceso inquiry general o especializado. El timing para inquiry es el mismo que paging (vea Sección 2.4.3).

### 2.5.4 Inquiry Response Timing

Un paquete inquiry response se transmite del slave al master después de que el slave ha recibido un mensaje inquiry (vea Tabla 8.5). Este paquete contiene información necesario para el inquiring master al page del slave (vea definición del paquete FHS Sección 6.5.1.4) y sigue 625 µs después de recibir el mensaje inquiry. En la Figura 2.10 y Figura 2.11, f(k) se usa para las frecuencias de la secuencia hopping inquiry y f'(k) denota la frecuencia de la secuencia inquiry response correspondiente. El paquete es recibido por el master en la frecuencia hop f'(k) cuando el mensaje inquiry es recibido por el slave es el primer slot master-a-slave.

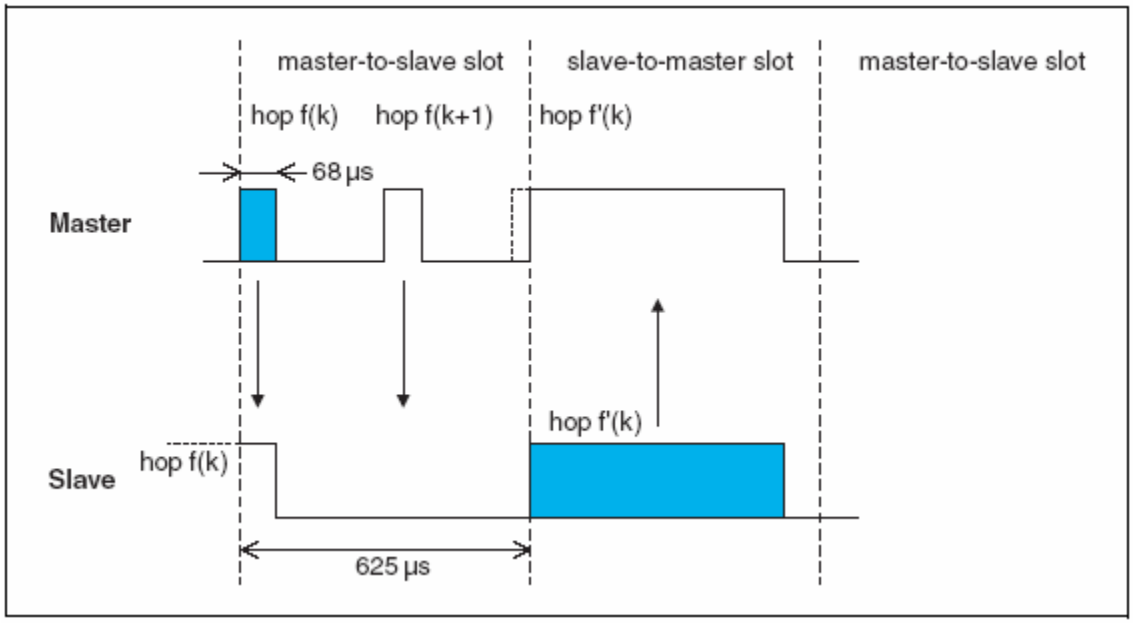



Figura 2.10: Timing en el paquete inquiry response en un inquiry exitoso en el primer medio slot

Cuando el mensaje inquiry es recibido por el slave en el segundo slot del master-a-slave el paquete es recibido por el master en la frecuencia hop f'(k+1).

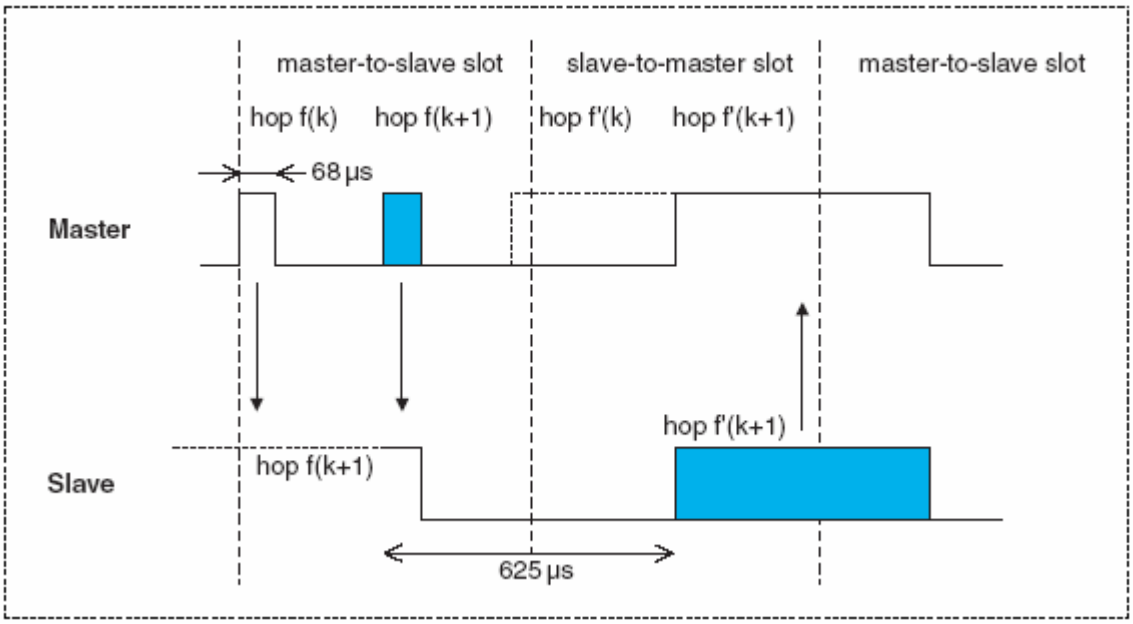



Figura 2.11: Timing en el paquete inquiry response en un inquiry en el segundo medio slot

## 2.6 Hop Selection

Los dispositivos Bluetooth utilizarán el hopping kernel como se define en las secciones siguientes. En suma, seis tipos de hopping kernel se definen; cinco para el sistema básico de salto y uno para un conjunto adaptado de hop locations utilizadas por adaptive frequency hopping (AFH).

Estas secuencias son:

• A **page hopping sequence con** 32 frecuencias distribuidas igualmente sobre los 79 MHz, con una longitud de período de 32.
• A **page response hopping sequence** cubriendo 32 frecuencias de respuesta que están una-a-una en la corriente page hopping sequence. El master y el slave utilizan diversas reglas para obtener la misma secuencia;
• An **inquiry hopping sequence** con 32 frecuencias distribuidas igualmente sobre los 79 MHz, con una longitud de período de 32
• An **inquiry response hopping sequence** cubriendo 32 frecuencias de respuesta que están una-a-una en la corriente inquiry hopping sequence.
• A **basic channel hopping sequence** la cual tiene una longitud de periodo larga, que no muestra las pautas repetitivas sobre un intervalo corto de tiempo, y que distribuye las frecuencias hop igualmente sobre los 79 MHz durante un intervalo corto de tiempo.
• An **adapted channel hopping sequence derivado del** basic channel hopping sequence que utiliza el mismo mecanismo del canal y puede utilizar menos de las 79 frecuencias. El adapted channel hopping sequence se utiliza solo en lugar del canal básico. Todas las otras hopping sequences no son afectadas por la hop sequence adaptation.

## 2.6.1 Esquema General de Selection

El esquema de selección consiste en dos partes:
• Selección de secuencia.
• Trazado de secuencia en las frecuencias hop.

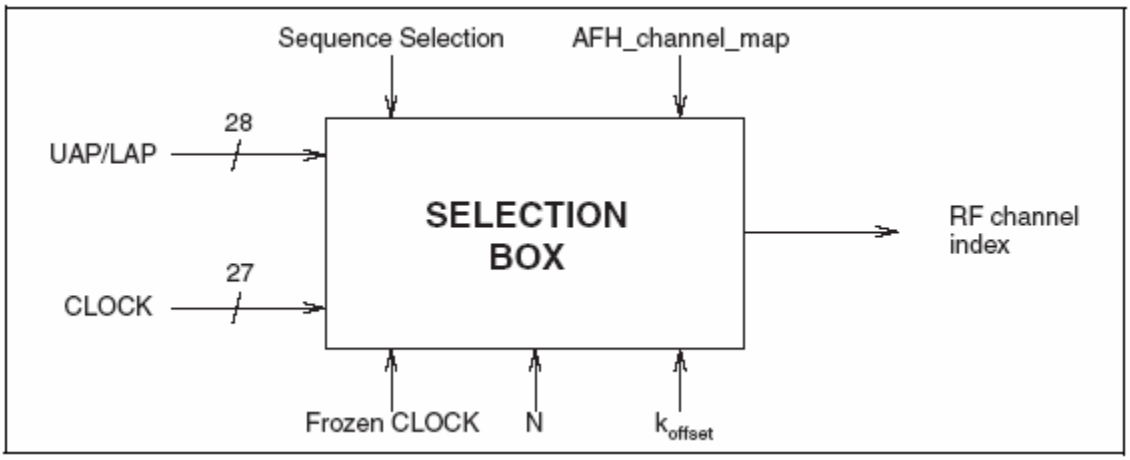
El diagrama de bloques general del hop selection scheme se muestra en la Figura 2.12. Se traza la entrada a un canal RF particular de un índice en la caja de selección. Las entradas en la caja de selección son el reloj seleccionado, frozen clock, N, k offset, address, sequence selection y AFH_el channel_map. La fuente de la entrada del reloj depende de hopping sequence selected. Adicionalmente, cada secuencia hopping utiliza diferentes bits de reloj.

La *sequence selection* se puede colocar con los valores siguientes:
• Page scan
• Inquiry scan
• Page
• Inquiry
• Master page response
• Slave page response
• Inquiry response
• Basic channel
• Adapted channel

El address imputa consiste en 28 bits incluyendo el LAP entero y los 4 LSBs del UAP. Esto se designa como el UAP/LAP. Cuándo el canal básico o adaptado secuencia hopping es seleccionado, la dirección del dispositivo Bluetooth del master (BD_ADDR) será utilizado. Cuándo page, master page response, slave page response, or page scan hopping sequences son seleccionadas el BD_ADDR dado por el Host del dispositivo paged se utilizará. Cuándo inquiry, inquiry response, o inquiry scan hopping sequences son seleccionados, el UAP/LAP correspondiente al GIAC se utilizará incluso si concierne un DIAC.

Siempre que uno de los reservado BD_ADDRs es utilizado para generar un frequency hop sequence, el UAP será reemplazado por el default check initialization (DCI). El hopping sequence es seleccionado por la sequence selection input de la caja de selección.

Cuándo el adapted channel hopping sequence se selecciona, el A*FH_channel_map* es una entrada adicional a la caja de selección. El *AFH_channel_map* indica que canales se utilizarán y cuales no serán usados.




La salida, *RF channel index*, constituye una secuencia seudo-aleatoria. El índice del canal RF es trazado en frecuencias del canal RF utilizando la ecuación de la Tabla 2.1de Radio especificación.

El esquema de selección elige un segmento de las 32 frecuencias hop que atraviesan los 64 MHz y visitan estos hops en un orden seudo-aleatorio. Próximo, a un segmento de 32 hop diferentes se elige, etc. En page, master page response, slave page response, page scan, inquiry, inquiry response and inquiry scan hopping sequences, el mismo segmento de 32-hop se utiliza todo el tiempo (el segmento es seleccionado por la dirección; diferentes dispositivos tendrán diferentes segmentos paging)

Cuándo el basic channel hopping sequence se selecciona la salida constituye una sucesión seudo-aleatorio que se desliza por los 79 hops. El principio se representa en la Figura 2.13.

.

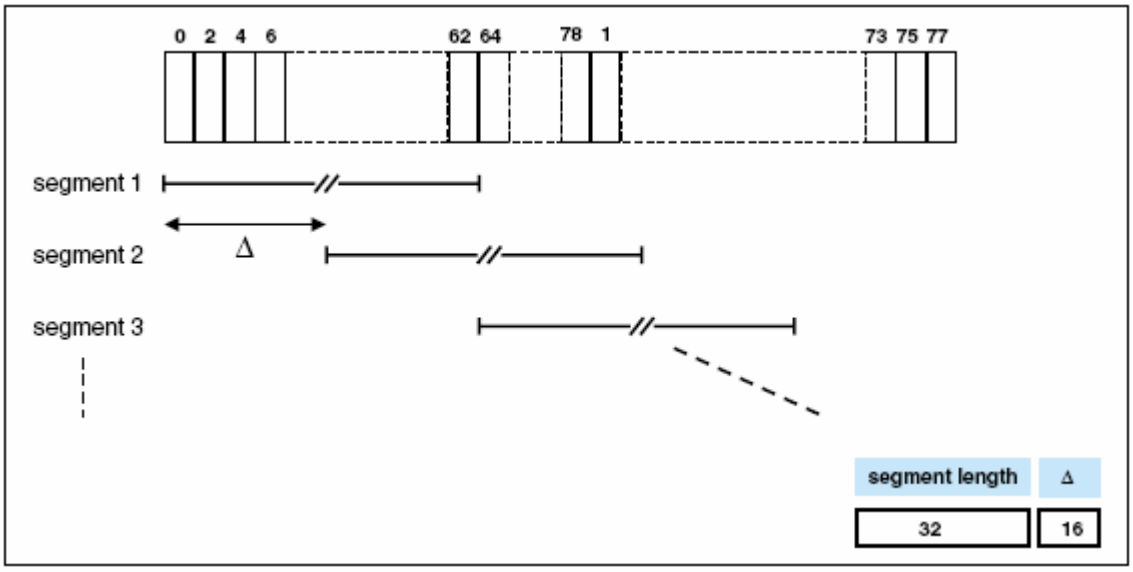



Figura 2.13: Hop selection scheme in CONNECTION state.

La frecuencia RF se quedará fija durante el paquete. La frecuencia RF para el paquete se derivará del valor de reloj Bluetooth en el primer slot del paquete. La frecuencia RF en el primer slot después de un paquete multi-slot utilizará la frecuencia como lo determina el valor del reloj Bluetooth para ese slot. La Figura 2.14 ilustra la definición del hop simple y un paquete multi-slot.

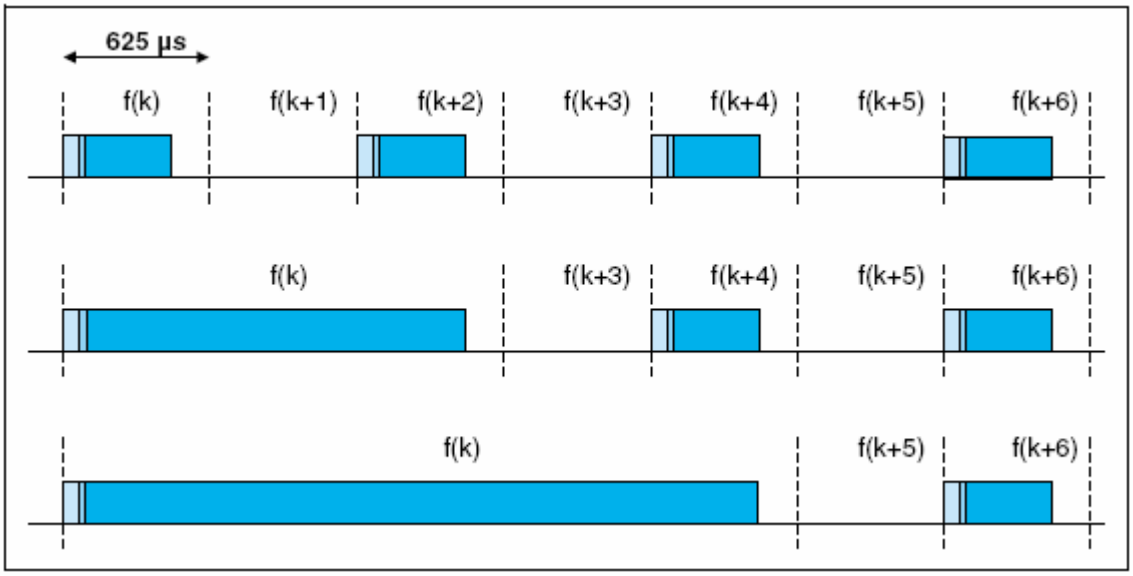



*Figura 2.14: Single- and multi-slot packets.*

Cuándo el adapted channel hopping sequence se utiliza, la sucesión seudo-aleatorio contiene sólo frecuencias que están en el conjunto del canal RF definido por la entrada *AFH_channel_map*. La adapted sequence tiene propiedades estadísticas semejantes al non-adapted hop sequence. Además, el slave responde con un paquete en el mismo canal de RF que fue utilizado por el master para direccionar ese slave. Esto se llama *same channel mechanism* de AFH. Así, el canal RF utilizado por el master al paquete del slave se utiliza también para el slave siguiente inmediato al paquete del master. Un ejemplo del *same channel mechanism* se ilustra en la Figura 2,15. El mismo mecanismo de canal se utilizará siempre que el adapted channel hopping sequence se selecciona.

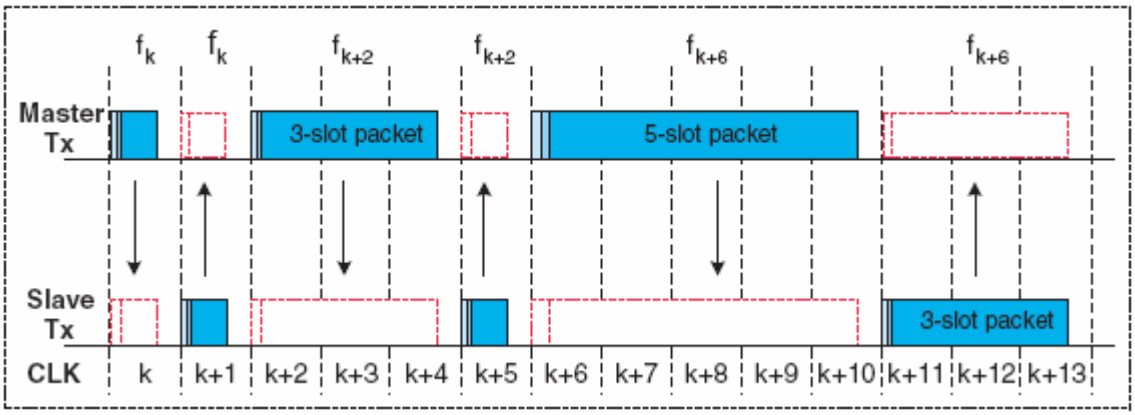



*Figura 2.15: Example of the same channel mechanism.*

## 2.6.2 Selection kernel

El basic hop selection kernel es como se muestra en la Figura 2.16 y es usado por page, page response, inquiry, inquiry response y basic channel hopping selection kernels. En estos subestados el AFH_channel_map la entrada no es usada. La entrada X determina la fase en el segmento de 32 hop, mientras que Y1 y Y2 escogen entre master a slave y slave a master. Las entradas A a D determina el ordenar dentro del segmento, las entradas E y F determinan el mapeo de las frecuencias hop. El kernel addresses contiene un registro de los índices del canal RF. Se pide esta lista para primero enumerar todos los índices pares del canal RF y después todas las frecuencias impares hop. De esta manera, un segmento de 32 hop atraviesa los 64 MHz.

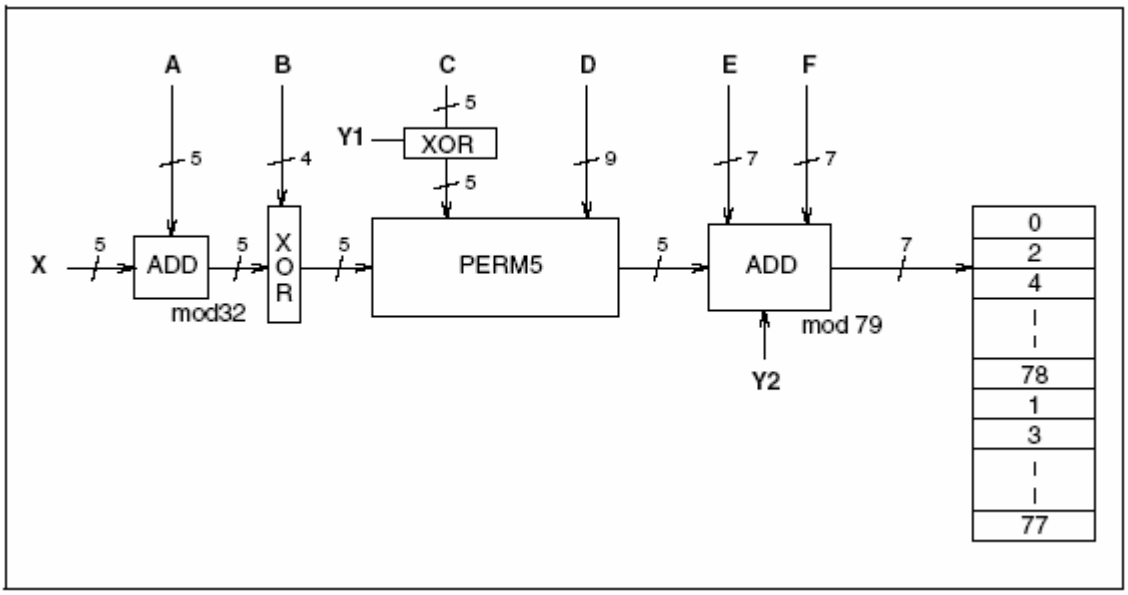



*Figura 2.16: Diagrama bloques del basic hop selection kernel para el hop system.*

El procedimiento de la selección consiste en una adición, una operación XOR, una operación de permutación, y finalmente una selección de registro. En el resto de este capítulo, la anotación Ai se utiliza para el bit i del BD_ADDR.

### 2.6.2.1 First addition operation

The first addition operation sólo agrega una constante a la fase y aplica un modulo de 32 operaciones. Para page hopping sequence, la first addition es superflua desde que sólo cambia la fase dentro del segmento. Sin embargo, cuándo diferentes segmentos se enlazan (como en el basic channel hopping sequence), la first addition operation tendrá un impacto en la sucesión resultante.

## 2.6.2.2 XOR operation

Z' permite denotar la entrada de la primera adición. En la operación XOR, los cuatro LSBs de Z' es el modulo-2, añadidos a los bits A22-19 de la dirección. La operación se ilustra en la Figura 2.17**.**

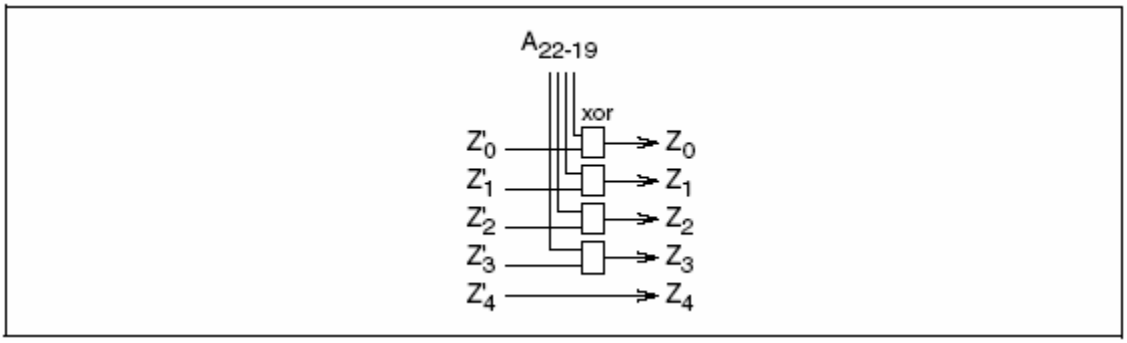



Figura 2.17: XOR operation para el hop system

La operación de la permutación implica la conmutación de 5 entradas a 5 salidas para el sistema hop, controlado por la palabra de control. La caja de la permutación o la conmutación es como se muestra en la Figura 2.18. Consiste en 7 etapas de operaciones butterfly. El control de las butterflys por las señales de control P se muestra en la Tabla 2.1. P0-8 corresponde a D0-8, y, corresponde a la Figura 2.16. La entrada Z es la salida de la operación XOR como se describió en la sección previa. La operación butterfly se puede aplicar con multiplexores como se representó en la Figura 2,19.

| Control signal | Butterfly | | Control signal | Butterfly |
|---|---|---|---|---|
| $P_0$ | $\{Z_0,Z_1\}$ | | $P_8$ | {Z1,Z4} |
| $P_1$ | $\{Z_2,Z_3\}$ | | $P_9$ | $\{Z_0,Z_3\}$ |
| $P_2$ | $\{Z_1,Z_2\}$ | | $P_{10}$ | $\{Z_2,Z_4\}$ |
| $P_3$ | $\{Z_3,Z_4\}$ | | $P_{11}$ | $\{Z_1,Z_3\}$ |
| $P_4$ | $\{Z_0,Z_4\}$ | | $P_{12}$ | $\{Z_0,Z_3\}$ |
| $P_5$ | $\{Z_1,Z_3\}$ | | $P_{13}$ | $\{Z_1,Z_2\}$ |
| $P_6$ | $\{Z_0,Z_2\}$ | | | |
| $P_7$ | $\{Z_3,Z_4\}$ | | | |

Table 2.1: Control de butterflies por el hop system

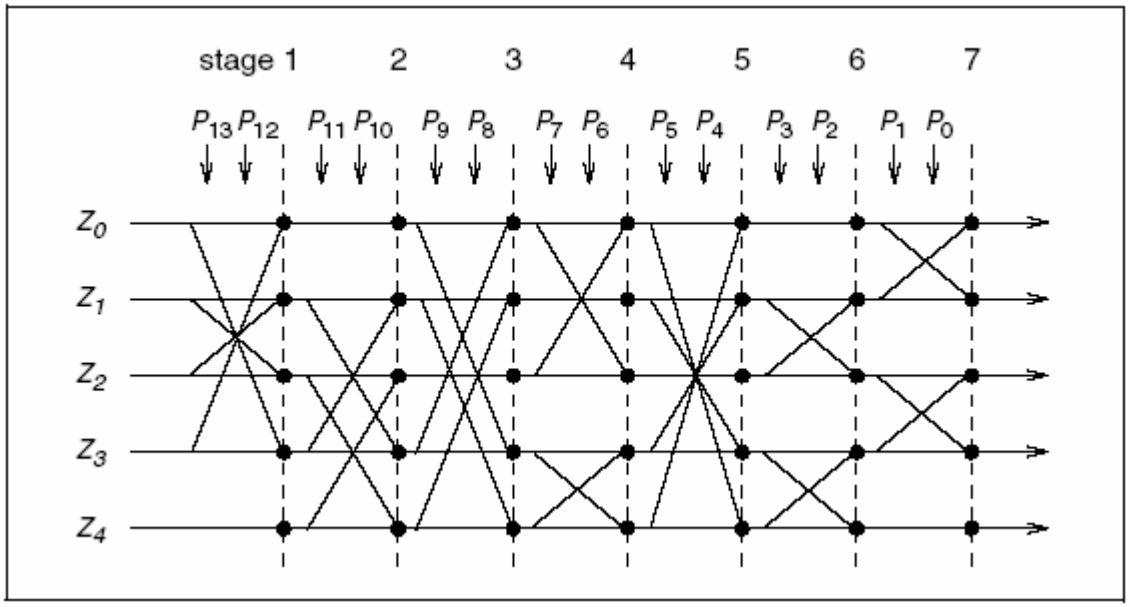



Figura 2.18: Permutation operation for the hop system.

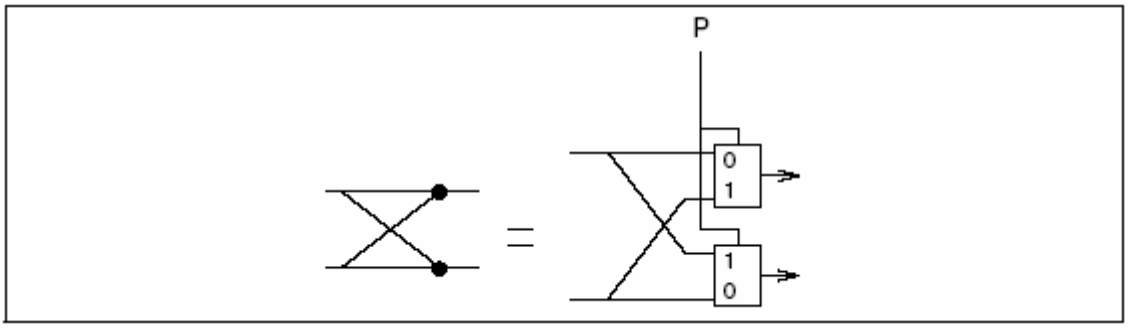



Figura 2.19: Butterfly implementation

### 2.6.2.4 Second addition operation

La operación de la adición sólo agrega una constante a la salida de la operación de permutación. La adición se aplica en el modulo 79.

### 2.6.2.5 Register bank

La salida de la adición dirige un banco de 79 registros. Los registros se cargan con las palabras en clave del sintetizador correspondiendo a las frecuencias hop de 0 a 78. Note que la mitad superior del banco contiene las frecuencias pares hop, mientras que la mitad más baja del banco contiene las frecuencias impares hop.

## 2.6.3 Adapted hop selection kernel

El adapted hop selection kernel se basa en el basic hop selection kernel definido en las secciones anteriores. Las entradas al adapted hop selection kernel son igual que para el basic hop selection kernel excepto que el AFH_channel_map es utilizado. El *AFH_channel_map* indica que canales RF se utilizarán y cuales no. Cuándo hop sequence adaptation es habilitado, el número de canales utilizados de RF se puede reducir de 79 a algún valor más pequeño N. Todos los dispositivos son capaces de operar en una adapted hop sequence (AHS) con $Nmin \leq N \leq 79$, con cualquier combinación de canales RF utilizados dentro del *AFH_channel_map* que encuentra esta limitación

La adaptación de la hopping sequence se logra por dos adiciones al basic channel hopping sequence según la Figura 2.16

• *Unused* RF channels es re-trazado uniformemente en los canales utilizados de RF. Eso es, si el hop selection kernel del sistema básico genera un canal no usado de RF, un canal alternativo de RF fuera del conjunto de canales utilizados RF, son seleccionados seudo-aleatoriamente.

• El *used* RF channel generado para el paquete master-al-slave es utilizado también inmediatamente siguiendo al paquete slave-al-master

### 2.6.3.1 Channel re-mapping function

Cuando el adapted hop selection kernel es seleccionado, el adapted hop selection kernel acorde a la Figura 2.16 son inicialmente utilizados para determinar un canal RF. Si este canal RF es *unused* según el *AFH_channel_map*, el *unused* RF channel es re-trazado por la función de re-mapeado a uno de los canales utilizados de RF. Si el canal RF determinado por el basic hop selection kernel esta listo en el conjunto de canales utilizados de RF, no se realiza ningún ajuste. La secuencia hop (non-adapted) del basic hop iguala la secuencia del adapted selection kernel en todas las ubicaciones donde *used* RF channels es generado por el basic hop. Esta propiedad facilita a non-AFH slaves quedar sincronizado mientras otros slaves en el piconet utilizan el adapted hopping sequence.

Un diagrama de bloques del mecanismo de re-mapeo se muestra en la Figura 2.20. La función de re-mapeo es un paso de post-procesamiento del selection kernel, denotado como ‘Hop selection of the basic hop’. La salida *f(k)* del basic hop selection kernel es un número del canal RF entre el rango de 0 y 78. Este canal RF hace o está en el conjunto de canales utilizados RF o en el conjunto de canales no usados RF.

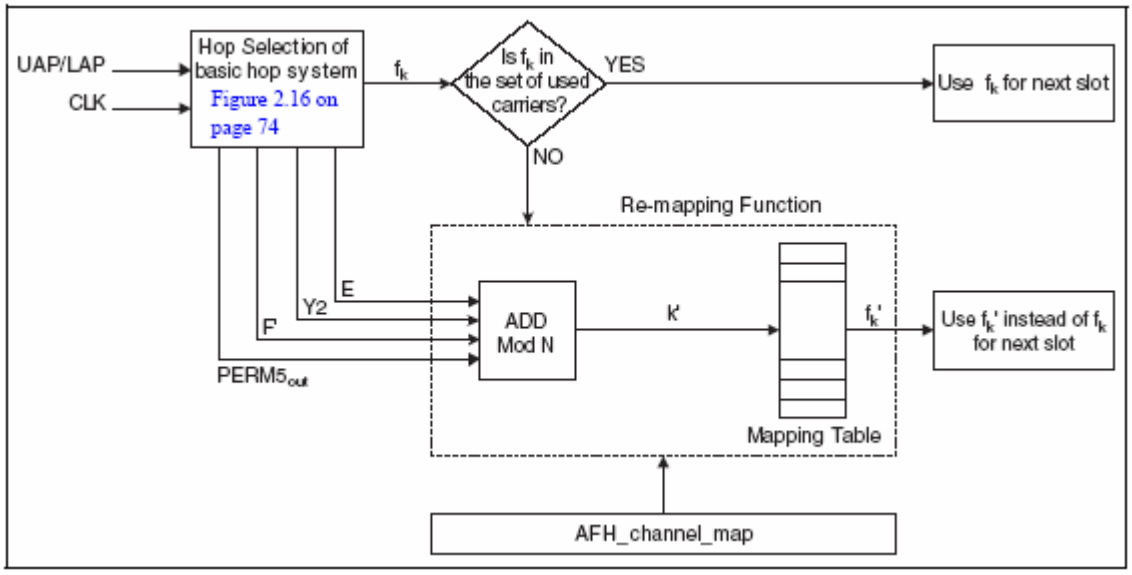



*Figura 2.20: Block diagram of adaptive hop selection mechanism*

Cuándo un unused RF channel es generado por el basic hop selection mechanism, este es re-trazado al conjunto de canales utilizados de RF como sigue. Un índice nuevo *k' ..* {0, 1,..., *N*-1} es calculado utilizando algunos de los parámetros del basic hop selection kernel:

*k'* = (*PERM5out* + *E* + *F'* + *Y2*) mod *N*

Donde *F'* es definido en la Tabla 2.2 El índice *k*' entonces es utilizado para seleccionar el canal re-trazado de una tabla mapping que contiene todo los *used* RF channels pares en orden ascendente seguido por los *used* RF channels impares de RF.

## 2.6.4 Control Word

En la siguiente sección, *Xj-i*, *i<j*, denotará los bits *i*, *i*+1,...,*j* del bit del vector X. Por la convención, X0 es el bit menos significativo del vector X.

La palabra del control del kernel es controlada por el conjunto de señales de control X, Y1, Y2, A a F, y F' como se ilustra en la Figura 2.16 y Figura 2.20. Durante paging and inquiry, las entradas A a E utiliza los valores de la dirección dados en las columnas correspondientes de la Tabla 2.2. Además, las entradas X, Y1 y Y2 se utilizan. Las entradas F y F' no son usadas. Los bits de reloj CLK6-2 (es decir, entrada X) especifica la fase dentro de la longitud de las 32 secuencias. CLK1 (es decir, las entradas Y1 y Y2) son utilizadas para seleccionar entre TX y RX. Las entradas de la dirección determinan el orden de la secuencia dentro de los segmentos. El mapeo final en las frecuencias hop es determinado por el contenido del registro.

Durante el estado de CONEXIÓN, las entradas A, C y D se derivarán de los bits de la dirección siendo bit-wise XORed con los bits de reloj como se muestran en el "el estado de la Conexión" la columna de la Tabla 2.2 (los dos bits mas significativos, MSBs, es XORed juntos, los dos segundos MSBs es XORed juntos, etc.)

| | Page scan / Interlaced Page Scan / Inquiry scan / Interlaced Inquiry Scan | Page/Inquiry | Master/Slave page response and Inquiry response | Connection state |
|---|---|---|---|---|
| X | $CLKN_{16-12}$ / $(CLKN_{16-12}+16) mod 32$ / $Xir_{4-0}$ / $Xir_{4-0}+16) mod 32$ | $Xp_{4-0}/Xi_{4-0}$ | $Xprm_{4-0}$/ $Xprs_{4-0}$/ $Xir_{4-0}$ | $CLK_{6-2}$ |
| Y1 | 0 | $CLKE_1/CLKN_1$ | $CLKE_1/CLKN_1/1$ | $CLK_1$ |
| Y2 | 0 | $32\times CLKE_1/$ $32\times CLKN_1$ | $32\times CLKE_1$ / $32\times CLKN_1$ / $32\times 1$ | $32\times CLK_1$ |
| A | $A_{27-23}$ | $A_{27-23}$ | $A_{27-23}$ | $A_{27-23}\oplus CLK_{25-21}$ |
| B | $A_{22-19}$ | $A_{22-19}$ | $A_{22-19}$ | $A_{22-19}$ |
| C | $A_{8,6,4,2,0}$ | $A_{8,6,4,2,0}$ | $A_{8,6,4,2,0}$ | $A_{8,6,4,2,0}\oplus CLK_{20-16}$ |
| D | $A_{18-10}$ | $A_{18-10}$ | $A_{18-10}$ | $A_{18-10}\oplus CLK_{15-7}$ |
| E | $A_{13,11,9,7,5,3,1}$ | $A_{13,11,9,7,5,3,1}$ | $A_{13,11,9,7,5,3,1}$ | $A_{13,11,9,7,5,3,1}$ |
| F | 0 | 0 | 0 | $16\times CLK_{27-7} \bmod 79$ |
| F' | n/a | n/a | n/a | $16\times CLK_{27-7} \bmod N$ |

Tabla 2.2

Los cinco bits de entrada X varían dependiendo del estado actual del dispositivo. En page scan and inquiry scan substates, el reloj nativo (CLKN) será utilizado. En el estado de CONEXION el reloj master (CLK) será utilizado como de entrada. La situación se complica mas para los otros estados.

### 2.6.4.1 Page Scan and Inquiry Scan Hopping Sequences

Cuando la entrada de la selección de secuencia se pone en page scan, el dispositivo de dirección Bluetooth del dispositivo scanning será utilizado como entrada de dirección. Cuándo la entrada de selección de secuencia es puesta en inquiry scan, el GIAC LAP y los 4 LSBs del DCI (s), serán utilizados como entrada de dirección para el hopping sequence. Para el código de acceso transmitido y en el correlator de recepción, el GIAC o DIAC apropiado se utilizarán. La aplicación decide cuál código de acceso inquiry utilizara dependiendo del propósito de inquiry.

### 2.6.4.2 Page Hopping Sequence

Cuando la entrada de la selección de secuencia es puesta en page, el dispositivo paging empezará a utilizar el A-Train es decir, dónde está la estimación de la fuente de frecuencia actual del receptor en el dispositivo paging. El índice K es una función de todas las entradas en la Figura 2.16. Hay 32 frecuencias paging posibles dentro de cada intervalo de 1,28 segundo. La mitad de estas frecuencias pertenece al A-Train, los demás pertenece al B-Train. Para lograr las 8 desviaciones del A-Train, una constante de 24 se añadirá a los bits de reloj (que equivale a -8 debido al modulo de 32 operaciones). El B-Train se obtiene poniendo la desviación a 8. Un cambio cíclico del orden dentro de los trains es también necesario para evitar una incompatibilidad repetitiva posible entre paging y scanning devices.

Así,

$$Xp = [\mathrm{CLKE}_{16-12} + k_{offset} + (\mathrm{CLKE}_{4-2,0} - \mathrm{CLKE}_{16-12}) \mod 16] \mod 32, \quad \text{(EQ 2)}$$

Donde

$$k_{offset} = \begin{cases} 24 & \text{A-train,} \\ 8 & \text{B-train.} \end{cases} \quad \text{(EQ 3)}$$

Alternativamente, cada interruptor entre el A- y los B- Trains pueden ser alcanzados agregando 16 al valor actual (inicializa originalmente con 24).

### 2.6.4.3 Slave Page Response Hopping Sequence

Cuando la entrada de la selección de secuencia es puesta en *slave page response*, para eliminar la posibilidad de perder la conexión debido a discrepancias del reloj nativo CLKN y la estimación de reloj master CLKE, los cuatro bits serán congelados en su valor actual. El valor se congelará en el contenido que tiene el slot donde el código de acceso del recipiente es detectado. El reloj nativo no se parará; es meramente el valor de los bits utilizados para crear el X input que se mantiene fijo por un tiempo. Un valor congelado es denotado por un asterisco (*) mas abajo.

Para cada response slot el dispositivo paged utilizará una X-Input valor más grande (modulo 32) que en el response slot anterior. Sin embargo, la primera respuesta se hará con la X-Input mantenido en el mismo valor como cuando el código de acceso fue reconocido. Permite ser un contador que comienza en cero. Entonces, el X-Input en el –slot de respuesta th (el primer response slot es el siguiente inmediato page slot ahora respondiendo a) del slave response substate es:

$$Xprs = [\mathrm{CLKN}^*_{16-12} + N] \mod 32,$$

(EQ 4)

El contador se pondrá a cero en el slot donde el slave acknowledges the page. Entonces, el valor será aumentado por uno cada vez que es puesto a cero, el cual corresponde al comienzo de un slot master TX. La X-Input se construirá de esta manera hasta que el primer paquete FHS se reciba y el paquete inmediato siguiente de la respuesta se ha transmitido. Después este slave entrará en el estado de CONEXION utilizado los parámetros recibidos en el paquete FHS.

### 2.6.4.4 Master Page Response Hopping Sequence

Cuando la entrada de la selección de secuencia es puesta en *master page response*, el master congelará su reloj estimado del slave al valor que provocó una respuesta del dispositivo paged. Equivale a utilizar los valores de reloj estimados al recibir la respuesta del slave (desde que sólo diferirá de la transmisión page correspondiente). Así, los valores se congelan cuando el paquete ID del slave se recibe. Además de los bits de reloj utilizan, los valores actuales que son congelados también. El master ajustará su X-Input de la misma manera que el dispositivo paged lo hace, es decir, incrementando este valor uno cada vez que es puesto a cero. El primer incremento se hará antes de mandar el paquete FHS al dispositivo paged. Permite al contador empezar en uno. La regla para la formar X-Input es:

$$Xprm = [\mathrm{CLKE}^*_{16-12} + k_{offset}{}^* + (\mathrm{CLKE}^*_{4-2,0} - \mathrm{CLKE}^*_{16-12}) \bmod 16 + N] \mod 32, \quad \text{(EQ 5)}$$

El valor será incrementado cada vez que es puesto a cero, que corresponde al comienzo de un slot master TX

### 2.6.4.5 Inquiry hopping sequence

Cuando la entrada de la selección de secuencia es puesta en *inquiry,* la X-Input es semejante a la utilizada *page hopping sequence*. Desde que ningún dispositivo particular se dirige, el reloj nativo CLKN del inquiry se utilizará.

Consecuentemente,

$$Xi = [\mathrm{CLKN}_{16-12} + k_{offset} + (\mathrm{CLKN}_{4-2,0} - \mathrm{CLKN}_{16-12}) \mod 16] \mod 32, \quad \text{(EQ 6)}$$

Dónde es definido por (EQ 3). La elección inicial del offset es arbitraria.
El GIAC LAP y los cuatro LSBs del DCI (as) serán utilizados como entrada de dirección para el hopping sequence generator.

### 2.6.4.6 Inquiry Response Hopping Sequence

El *inquiry response* hopping sequence es similar al *slave page response* hopping sequence con respecto a la X-Input. La entrada del reloj no se congelará, así la ecuación siguiente aplica:

$$Xir = [\mathrm{CLKN}_{16-12} + N] \mod 32,$$

(EQ 7)

Además, el contador se incrementa no en base, sino después que cada paquete FHS ha sido transmitido en respuesta a inquiry. No hay restricción en el valor inicial pues es independiente del valor correspondiente en la unidad inquiring. El GIAC LAP y los cuatro LSBs del DCI (as) serán utilizados como entrada de dirección para el hopping sequence generator. Los otros bits de entrada al generador serán igual para page response

### 2.6.4.7 Basic and Adapted Channel Hopping Sequence

En el *basic* and *adapted channel hopping sequences*, los bits de reloj utilizados en el *basic* and *adapted channel hopping sequences* siempre se derivaran del reloj master, CLK. Los bits de la dirección se derivarán de la dirección del dispositivo master Bluetooth.

## 3 Physical Links

Un physical link representa una conexión baseband entre dispositivos. Un physical link siempre se asocia exactamente un canal físico. El physical link tienen las propiedades comunes que aplican a todos los transportes lógicos en la conexión física. Las propiedades comunes del physical link son:

• Power control
•Link supervision
• Encryption
• Channel quality-driven data rate change
• Multi-slot packet control

## 3.1 Link Supervision

Una conexión puede colapsarse debido a varias razones tales como, que un dispositivo se mueve fuera de rango, encontrando interferencia severa o una condición de falla del suministro eléctrico. Desde que existe esta posibilidad y puede suceder sin ninguna advertencia previa, es importante controlar la conexión en el lado del master y en el lado del slave para evitar las posibles colisiones cuando el logical transport address (Sección 4.2) o parked member address (Sección 4.7.1) son reasignados a otro slave.

Para ser capaz de detectar la pérdida de conexión, el master y el slave utilizara un tiempo de supervisión de enlace, la supervisión T. Sobre la recepción de un paquete header válido de con uno de las direcciones del slave (vea la Sección4.2) en la physical link, el tiempo estará en reset. Si en un tiempo en el estado de CONEXION, el tiempo alcanza el *supervisionTO* value, la conexión se considerará desconectada. El mismo tiempo de supervisión del enlace se utiliza para los transportes lógicos SCO, eSCO, y para ACL. El período de tiempo muerto, supervisión TO, es negociado por el Link Manager. Su valor se elegirá para que el tiempo muerto de la supervisión sea más largo que los períodos hold and sniff. La supervisión de enlace de un parked slave es realizado por unparking y el re-parking del slave.

# 4 Logical Transports

## 4.1 General

Entre el master y los slave(s), se pueden establecer diferentes tipos de transportes lógicos. Se han definido cinco transportes lógicos:

• Synchronous Connection-Oriented (SCO) logical transport
• Extended Synchronous Connection-Oriented (eSCO) logical transport
• Asynchronous Connection-Oriented (ACL) logical transport
• Active Slave Broadcast (ASB) logical transport
• Parked Slave Broadcast (PSB) logical transport

Los transportes lógicos síncronos son transportes lógicos punto a punto entre un master y un solo slave en el piconet. Los transportes lógicos síncronos típicamente soportan información time-bounded como la voz o datos síncronos generales. El master mantiene los transportes lógicos síncronos utilizando slots reservados en intervalos regulares. Además de los slots reservados el transporte lógico eSCO puede tener una ventana de retransmisión después de los slots reservados.

El transporte lógico ACL es también un transporte lógico punto a punto entre un master y un slave. En los slots no reservados para el transporte lógico síncrono(s),

el master puede establecer un transporte lógico ACL en una base por-slot a cualquier slave, inclusive el slave(s) enganchado ya en un transporte lógico síncrono. El transporte lógico ASB es utilizado por un master para comunicase con los slaves activos. El transporte lógico PSB es utilizado por un master para comunicarse con parked slaves**.**

## 4.2 Logical Transport Address (Lt_Addr)

Cada slave activo en un piconet le es asignado una dirección lógica de transporte (LT_ADDR) primario de 3-bit. El all-zero LT_ADDR se reserva para transmisiones de mensajes. El master no tiene un LT_ADDR. Un LT_ADDR secundario es asignado al slave para cada transporte lógico eSCO en el uso del piconet. Sólo tráfico eSCO (por ejemplo NULL, POLL y uno de los tipos de paquete EV según lo negociado en la disposición lógica del transporte eSCO) puede ser mandado en estos LT_ADDRs. El tráfico ACL (inclusive LMP) siempre será enviado en el LT_ADDR primario. El LT_ADDR se lleva en el header del paquete. El LT_ADDR sólo será válido mientras un slave este en modo activo. Tan pronto como se desconecta o esta en parked, el slave perderá todos sus LT_ADDRs.

El LT_ADDR primario será asignado por el master al slave cuando este es activado. Esto es el establecimiento de cualquier conexión, en el papel de interruptor, o cuando el slave esta en unparked. En el establecimiento de conexión y en el papel de interruptor, el LT_ADDR primario se lleva dentro del FHS payload. Al unparking, el LT_ADDR primario se llevan en el mensaje unpark.

## 4.3 Synchronous Logical Transports

El primer tipo de transporte lógico síncrono, el transporte lógico SCO es simétrico, la conexión es punto a punto entre el master y un slave específico. El transporte lógico SCO reserva slots y puede por lo tanto ser considerado como una conexión circuito-switched entre el master y el slave. El master puede soportar hasta tres conexiones SCO con el mismo slave o a diferentes slave. Un slave puede soportar hasta tres conexiones de SCO con el mismo master, o dos conexiones SCO si las conexiones originales son de diferentes master. Los paquetes SCO nunca se retransmiten.

El segundo tipo de transporte lógico síncrono, el transporte lógico eSCO, es un transporte lógico punto a punto entre el master y un slave específico. Los transportes lógicos eSCO pueden ser simétricos o asimétricos. Semejante a SCO, eSCO reserva slots y puede por lo tanto ser considerado una conexión circuito-switched entre el master y el slave. Además de los slots reservados, eSCO soporta una ventana de retransmisión inmediatamente después de los slots reservados. Junto, los slots reservadas y la ventana de retransmisión forman la ventana completa eSCO.

## 4.4 Asynchronous Logical Transport

En los slots no reservados para transportes lógicos síncronos, el master puede cambiar paquetes con cualquier slave en una base por-slot. El transporte lógico ACL proporciona una conexión paquete- switched entre el master y todos los slaves activos que toman parte en el piconet. Ambos servicios asíncronos e isócronos se soportan. Entre un master y un slave sólo un transporte lógico ACL existirá. Para la mayoría de los paquetes ACL, la retransmisión de paquete se aplica para asegurar la integridad de datos. Los paquetes ACL no diseccionados a un slave específico son considerados como paquetes de transmisión y deben ser leídos por cada slave. Si no hay datos que se enviarán en el transporte lógico ACL y no se requiere ningún polling, ninguna transmisión se requiere.

## 4.5 Rutinas de Transmisión y Recepción

Esta sección describe la manera de utilizar los paquetes son definidos en la Sección 6 el soportar el tráfico en el ACL, SCO y las enlaces de eSCO. Ambas configuraciones del solo-slave y el multi-slave se consideran. Además, el uso de búferes para TX y rutinas de RX se describen. El TX y las rutinas de RX se describieron en 4.5.1, y secciones 4.5.2 son informativos solamente.

### 4.5.1 Rutina TX

La rutina de TX se lleva a cabo separadamente para cada enlace asíncrono y síncrona. La figura 4.1se muestran los búferes asíncronos y síncronos y como utilizan como la rutina TX. En esta figura, sólo un solo TX el búfer asíncrono y un solo TX el búfer síncrono se muestran. En el master, hay un TX separado del búfer asíncrono para cada slave. Es posible que haya además uno o más TX los búferes síncronos para cada slave síncrono (los transportes lógicos SCO o eSCO pueden o vuelven a emplear el mismo TX del búfer síncrono, o cada uno tiene su propio TX búfer síncrono). Cada búfer de TX consiste en dos registros de FIFO: un registro actual que puede ser conseguir acceso a y ser leído por el Director de enlace para componer los paquetes, y uno registra después eso y pueda conseguir un acceso por el Director Recurso de Baseband para cargar
Información nueva. Las posiciones de los interruptores S1 y S2 determinan cuál registro es actual y que registro es el próximo; los interruptores son controlados por el Director de enlace. Los interruptores en la entrada y la producción de los registros de FIFO nunca pueden ser conectados al mismo registro simultáneamente. La figura 4.1: esquema Funcional de buffering de TX.

Los paquetes son comunes en el transporte lógico ACL y SCO (NULL, POLL y DM1) sólo el paquete DM1 lleva una carga útil que se cambia entre el Director de enlace y Director de la Conexión; este paquete común utiliza el búfer asíncrono.

Todos los paquetes de ACL utilizan el búfer asíncrono. Todo SCO y los paquetes de eSCO utilizan el búfer síncrono menos el paquete de DV donde la parte síncrona de datos es manejada por el búfer síncrono y la parte de datos es manejada por el búfer asíncrono. La operación para el tráfico de ACL, para el tráfico de SCO, para el tráfico de eSCO, y para el tráfico combinado de datos-voz se describe en el transporte lógico SCO.

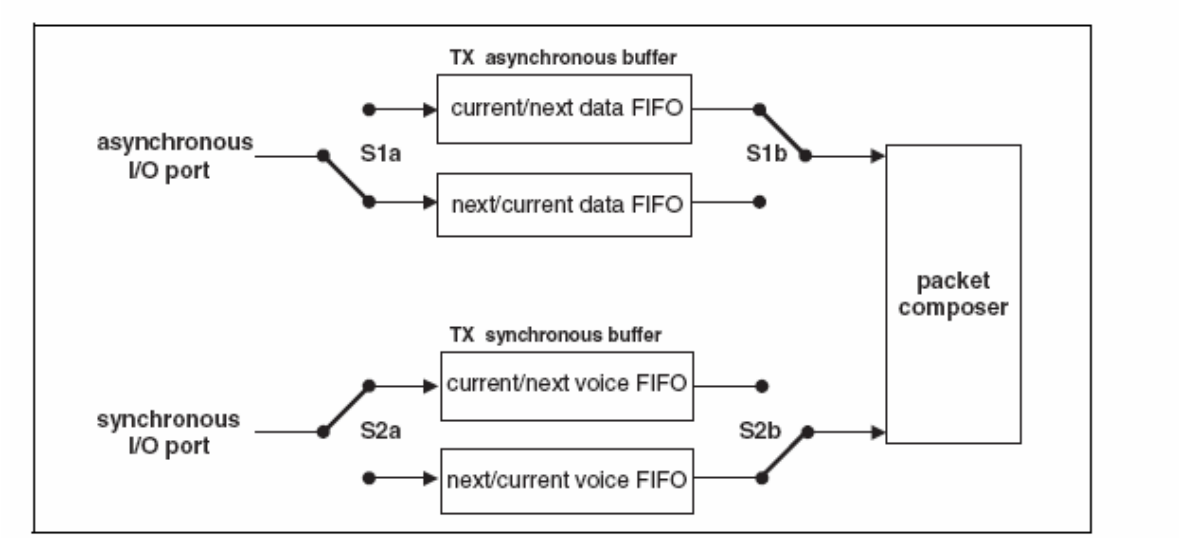



*Figure 4.1: Functional diagram of TX buffering.*

### 4.5.1.1 El tráfico de ACL

En el caso de datos asíncronos sólo el búfer de TX ACL en la Figura 4.1 en la tienen que ser considerados. En este caso, sólo en este tipo de paquete DM o DH se utilizan, y éstos pueden tener las longitudes diferentes. La longitud se indica en el header de la carga útil. La selección de DM o paquetes de DH deben depender de la calidad de la enlace. a Parte C] 4.1.7.

El tipo de paquete predefinido en puros datos negocia NULL (ver la Sección 6.5.1.2) Esto significa que, si no hay los datos de ser mandados (el tráfico de datos es asíncrono, y por lo tanto las pausas ocurren en datos que no están disponibles) ni ningunos slaves necesitan ser polled, paquetes NULL se mandan en lugar de mandar la enlace de información de control al otro dispositivo (por ejemplo. información de ACK/STOP para datos recibidos). Cuándo la información de control de enlace está disponible (cualquier necesidad acknowledgeand o que no necesita parar el flujo de RX) paquete no se manda en todo.

Los trabajo de la rutina de TX siguen como. El Director del Recurso de Baseband carga información nueva de datos en el registro a que el interruptor S1 señala. Próximo, da una orden al Director de enlace, el interruptor S1cambia (tanto S1a como S1b cambian síncronamente). Cuándo las necesidades de carga útil se

mandan el paquete lee el registro actual y, dependiendo del tipo de paquete, construye una carga útil que es añadida al código de acceso del canal y el header se transmite subsecuentemente. En el paquete response (que llega en el slot siguiente de RX si concreto una transmisión perfecta, o se puede aplazar hasta que algunos slots posteriormente de RX si concernió una transmisión de slave), el resultado de la transmisión se informa detrás. En caso de un ACK, el interruptor S1 cambia la posición; si un NAK (explícito o implícito) es recibido en vez de eso, el interruptor S1 no cambiará la posición. En ese caso, la misma carga útil se retransmite en la próxima ocasión de TX.

Tan largo como el Director del Recurso de Baseband mantiene cargando los registros con información nueva, el Director de enlace transmitirá automáticamente la carga útil; además, son realizados retransmisiones automáticamente en caso de errores. El Director de enlace mandará NULL o nada cuando ninguno de los datos nuevos se carga. Si ningunos de los datos nuevos se han cargado en el próximo registro, durante la última transmisión, el paquete estará señalando a un registro vacío después en la última transmisión se ha reconocido y el próximo registro llega a ser el registro actual. Si los datos nuevos se cargan en el próximo registro, una orden paring se requiere a cambiar el interruptor S1 al registro apropiado. Tan largo como el Director del Recurso de Baseband mantiene cargando los datos y escribe registros antes de cada slot de TX, los datos son procesados automáticamente por el Director de enlace desde el interruptor S1 es controlado por la información de ACK recibida en la respuesta. Sin embargo, si el tráfico del Director del Recurso de Baseband se interrumpe una vez y un paquete predefinido se manda en vez de eso, una orden paring es necesaria para continuar el flujo en el Director de enlace.

La orden paring se puede utilizar también en caso de time-slot (isócrono) los datos. En caso de un enlace malo, muchos son retransmisiones necesarias. En ciertas aplicaciones, los datos son time-slots: si una carga útil se retransmite todo el tiempo a causa de errores de enlace, puede llegar a ser caduco, y el sistema quizás decida continuar con datos más recientes en lugar y se salta la carga útil que no se envía. Esto es alcanzado por la orden paring también. por, el interruptor S1 se fuerza a cambiar y el Director de enlace es forzado a considerar la próxima carga útil de datos y predomina el control de ACK. De tipo ACL de paquete puede ser utilizado para mandar los datos o información de control de enlace a cualquier otro slave de ACL.

### 4.5.1.2 Tráfico SCO

En el transporte lógico SCO sólo HV y tipos de paquete de DV se utilizan, Ver la Sección 6.5.2. El puerto síncrono puede cargar continuamente el próximo registro en el búfer síncrono. Los interruptores S2 se cambian según el intervalo de Tsco. Este intervalo de Tsco se negocia entre el master y el slave en el tiempo del transporte lógico SCO se establece. Para cada slot nuevo de SCO, el paquete lee

el registro actual después que el interruptor S2 se cambia. Si el slot de SCO se tiene que utilizar para mandar el control información con la prioridad alta con respecto a un paquete del control entre el master y el slave de SCO, o un paquete del control entre el master y cualquier otro slave, el paquete desechará la información de SCO y utilizará la información del control. Esta información del control se mandará en un paquete DM1. Los datos o la información del control de la enlace se pueden cambiar también entre el master y el slave de SCO utilizando el DV o los paquetes DM1.

### 4.5.1.3 Mezcla del Tráfico de Datos/Voz

En la sección 6.5.2 de la Sección en la página 109, un paquete de DV se ha definido eso puede sostener ambos datos y expresar simultáneamente en un solo SCO el transporte lógico. Cuándo el TIPO es DV, el Director de enlace lee los datos registran para llenar el campo de datos y el registro de voz para llenar el campo de la voz. Después, el interruptor S2 se cambia. Sin embargo, la posición de S1 depende del resultado de la transmisión como en el ACL transporte lógico: sólo si un ACK se ha recibido hace el cambio de interruptor S1 su posición. En cada paquete de DV, la información de la voz es nueva, pero la información de datos quizás se retransmita si la transmisión previa falló. Si no hay los datos de ser mandados, el SCO transporte lógico cambiará automáticamente de DV de tipo paquete al HV actual de tipo paquete utilizado antes la transmisión mezclada de datos/expresa.

Note que una orden paring es necesaria cuando la corriente de datos se ha interrumpido y los datos nuevos han llegado. La transmisión combinada de datos-expresa puede ser alcanzada también utilizando por separado el transporte lógico ACL además del transporte lógico SCO(los transportes) si la capacidad del canal permite esto.

### 4.5.1.4 Tráfico eSCO

En el transporte lógico eSCO sólo EV, el PULL y paquete NULL se describen, ver la Sección 6.5.3. El puerto síncrono puede cargar continuamente el próximo registro en el búfer síncrono. Los interruptores S2 se cambian según el intervalo de TeSCO. Este intervalo de TeSCO se negocia entre el master y el slave en el tiempo del transporte lógico eSCO se establece. Para cada slot nuevo de eSCO, el paquete lee el registro actual después que el interruptor S2 se cambia. Si el slot eSCO se tiene que utilizar para mandar el control información con la prioridad alta con respecto a un paquete del control entre el master y el slave de eSCO, o un paquete de ACL entre el master y cualquier otro slave, el paquete desechará la información de eSCO y utilizará la información del control. Control de información al slave del eSCO es mandada en un paquete DM1 en el primario LT_ADDR.

### 4.5.1.5 Los tipos Predefinidos de Paquete

En las enlaces de ACL, el tipo predefinido es siempre NULL para el master y el slave. Esto significa que si ninguna necesidad de información de usuario es mandada, ni un paquete NULOL se manda si hay ACK o información de STOP, o ningún paquete se mandan en todo. El paquete NULL puede ser utilizado por el master para asignar al próximo slave en un slot a un cierto slave (a saber donde es dirigido). Sin embargo, el slave no es forzado a responder al paquete NULL del master. Si el master requiere una respuesta, manda un paquete PULL.

El SCO y los tipos del paquete de eSCO se negocian en el nivel de LM cuando el SCO o eSCO el transporte lógico se establece. El tipo de paquete es también el predefinido para el SCO reservado o slots de eSCO.

## 4.5.2 Rutina RX

La rutina de RX se lleva a cabo separadamente para el transporte lógico ACL y los transportes lógicos síncronos. Sin embargo, por contraste al TX master el búfer asíncrono, un solo búfer de RX se comparte entre todos los esclavos. Para el búfer síncrono, cómo los transportes lógicos, síncronos y diferentes se distinguen depende de si los búferes síncronos extra se requieren o no. La figura 4,2 en la página 91 exposiciones los búferes asíncronos y síncronos utilizaron como en la rutina de RX. El RX el búfer asíncrono consiste en dos registros de FIFO: un registro que puede ser conseguir acceso a y puede ser cargado por el Director de la Conexión con la carga útil del último paquete de RX, y de un registro que puede ser conseguir acceso a por el Director del Recurso de Baseband para leer la carga útil previa. El RX el búfer síncrono consiste también en dos registros de FIFO: un registro que es llenado de información nuevamente llegada de voz, y un registro que puede ser leído por la unidad de procesamiento de voz.

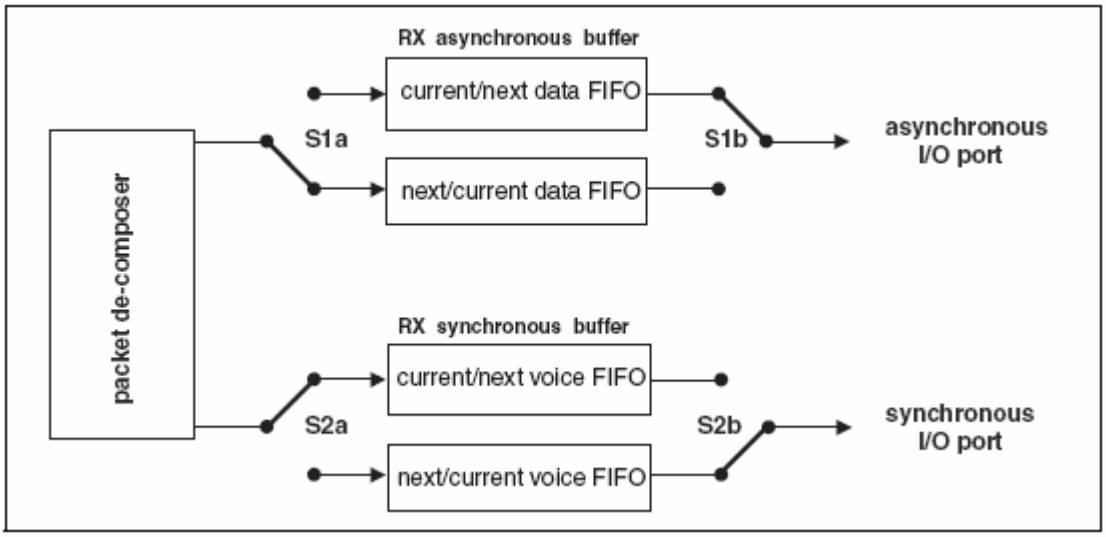



Figura 4.2. Esquema Funcional de buffering de RX

Desde que la indicación del TYPE en el header (ver la Sección 6.4.2) del paquete recibido indica si la carga útil contiene los datos y/o la voz, el de-composer de paquete puede dirigir automáticamente el tráfico a los búferes apropiados. El interruptor S1 cambia cada vez el Director del Recurso de Baseband y lee el registro viejo. Si la próxima carga útil llega antes el registro de RX se vacía, una indicación de STOP y se incluye en el header de paquete del próximo paquete de TX que se vuelve. La indicación de STOP se quita otra vez tan pronto como el registro de RX se vacía. El campo de SEQN se verifica antes una carga útil nueva de ACL se almacena en el registro asíncrono (la indicación paring) LLID y transmite los mensajes e influyen la interpretación del campo de SEQN ve la Sección 7,6). El interruptor S2 se cambia cada TSCO o TeSCO para SCO y eSCO respectivamente. Si, debido a los errores en el header, ninguna carga útil síncrona nueva llega, el interruptor checa los cambios. Los datos síncronos que procesan la unidad entonces procesan los datos síncronos para justificar las partes perdidas.

### 4.5.3 Control de Flujo

Desde que el búfer de RX ACL puede estar repleto mientras una carga útil nueva llega, el control del flujo se requiere. El flujo del campo de header en el paquete del regreso TX puede utilizar STOP o GO a controlar la transmisión de datos nuevos.

#### 4.5.3.1 Control de Destino

Tan largo como datos no se puede recibir, una indicación de STOP se transmitirá automáticamente por el Director de enlace en el header del paquete del regreso. STOP se volverá tan largo como el búfer de RX ACL no es vaciado por el Director del Recurso de Baseband. Cuándo datos nuevos se pueden aceptar otra vez, el la indicación GO se volverá. GO será el valor predefinido. Todos tipos de paquete e inclusive de datos se podrán recibir. Comunicación de voz por ejemplo no es afectada por el control del flujo. Aunque un dispositivo no pueda recibir información nueva, puede continuar todavía transmitir información: el control de flujo será separado para cada dirección.

#### 4.5.3.2 Control de la Fuente

En la recepción de una señal de STOP, el Director de enlace cambiará automáticamente al predefinido de tipo paquete. El paquete de ACL transmitirá apenas antes la recepción de la indicación STOP se mantendrá hasta que una señal GO se reciba. Se puede retransmitir tan pronto como una indicación GO se reciba. Los paquetes sólo predefinidos se mandarán tan largos como la indicación de la STOP es recibida. Cuándo un paquete no se recibe, GO será asumido implícitamente. Note que los paquetes predefinidos contienen información de control de enlace (en el header) para recibir la dirección (que puede estar todavía abiertos) y puede contener los datos síncronos (HV o paquetes EV). Cuándo una

indicación GO se recibe, el Director de enlace puede reasumir transmitiendo los datos que están presentes en los búferes de TX ACL. En una configuración de multi-slave, sólo la transmisión al slave, que publicó la señal STOP será atascada. Esto significa que el master irá sólo a la transmisión del búfer STOP de TX ACL correspondiendo al slave que momentáneamente no puede aceptar los datos.

## 4.6 Slave Activo Transmisión del Transporte

El slave activo transmisión del transporte lógico se utiliza para transportar el tráfico de usuario L2CAP a todos dispositivos en el piconet que son conectados actualmente al piconet del canal físico que es utilizado por el ASB. No hay protocolo de reconocimiento y el tráfico es unidireccional del master de piconet a los slaves. El ASB transporte lógico puede sólo ser utilizado para el tráfico del grupo L2CAP y nunca será utilizado para L2CAP los canales enlace-orientados, del control L2CAP que señala o el control de LMP. El ASB transporte lógico es informal. Para mejorar la calidad de cada paquete se transmite varios tiempos. Un número idéntico de la sucesión se utiliza para ayudar para filtrar retransmisiones en el dispositivo slave. El ASB transporte lógico es identificado por la dirección reservada, all-cero, LT_ADDR. Los paquetes en ASB transporte lógico pueden ser mandados por el master en tiempo.

## 4.7 Slave PARKED Transmision del Transporte

El slave PARKED transmite el transporte lógico y se utiliza para la comunicación del master a los slaves que son PARKED. El transporte lógico PSB es más complejo que los otros transportes lógicos como consisten en varias fases, tiene un propósito diferente. Estas fases son la fase de información de control (utilizada para llevar el LMP al enlace lógico), la fase de información de usuario (utilizada para llevar el L2CAP al enlace lógico), y la fase del acceso (llevando la señal baseband). El PSB del transporte lógico es identificado por la dirección reservada, all-cero, LT_ADDR.

### 4.7.1 Dirección de Miembro PARKED (PM_ADDR)

UN slave en el estado PARKED puede ser identificado por su BD_ADDR o por una dirección PARKED dedicada de miembro (PM_ADDR). Esta última dirección es una dirección de 8 bits de miembro que separa a los slaves estacionados. El PM_ADDR será válido tan largo como el slave PARKED. Cuándo el slave es activado se asignará un LT_ADDR pero perderá el PM_ADDR. El PM_ADDR es asignado al slave por el master durante el procedimiento de PARKING (ver [la Parte C] 4.5.2 de Sección). El All-cero PM_ADDR se reservará para slaves estacionados que sólo utilizan BD_ADDR para ser unparked.

### 4.7.2 Dirección de Petición de Acceso (AR_ADDR)

La dirección de petición de acceso (AR_ADDR) es utilizado por el slave PARKED para determinar al slave a medio slot master en la ventana de acceso donde se permite mandar el acceso los mensajes de petición, El AR_ADDR será asignado al slave cuando entra el estado parked y será válido tan largo como el slave parked. El AR_ADDR no es necesariamente extraordinario; es decir los slaves parked, diferentes pueden tener el mismo AR_ADDR.

## 5 Enlaces Lógicos

Cinco enlaces lógicos se definen:

• El Control de la Conexión (LC)
• El Control de ACL (ACL-C)
• Usuario Asíncrono/Isócrono (ACL-U) • Usuario Síncrono (SCO-S)
• Usuario Extendió Síncrono (eSCO-s)

El control lógico LC y ACL-C se utiliza en el nivel del control del enlace y el nivel de director de enlace, respectivamente. La ACL-U y el enlace lógico se utilizan para llevar información asíncrona o isócrona de usuario. El SCO-S, y eSCO-s enlace lógicos se utilizan para llevar información síncrona de usuario. El LC enlace lógico se lleva en el header de paquete, todos los otros enlaces lógicos se llevan en la carga útil de paquete. La los enlaces lógicos ACL-C y ACL-U se indican en la enlace lógico identificación, LLID, el campo en el header de la carga útil. El los enlaces lógicos SCO-S y eSCO-s son llevados por los transportes lógicos síncronos sólo; la enlace ACL-U es llevada normalmente por el transporte lógico ACL; sin embargo, ellos pueden ser llevados también por los datos en el paquete DV en el transporte lógico eSCO. El enlace ACL-C se puede llevar por el SCO o el transporte lógico ACL.

### 5.1 Control de Enlace Lógico (LC)

El control de enlace lógico LC se trazará en el header de paquete. Este enlace lógico lleva información baja de control de enlace de nivel como ARQ, como el control del flujo, y como caracterización de carga útil. El enlace lógico LC se lleva en cada paquete menos en el paquete de identificación que no tiene header de paquete.

### 5.2 Control de Enlace Lógico ACL (ACL-C)

El enlace lógico ACL-C llevará el control información cambió entre los directores del enlace del master y el slave (slaves). El enlace lógico ACL-C utilizará paquetes

DM1.El enlace lógico es indicado por el código de LLID 11 en el header de la carga útil.

## 5.3 Usuario de Enlace Lógico ASÍNCRONO/ISOCRONO (ACL-U)

Enlace lógico ACL-U llevará a L2CAP los datos asíncronos e isócronos de usuario. Estos mensajes se pueden transmitir en uno o más paquetes de baseband. Para mensajes fragmentados, el paquete del comienzo utilizará un código de LLID de 10 en el header de la carga útil. Los paquetes restantes de la continuación utilizarán código de LLID 01. Si no hay fragmentación, todos los paquetes utilizarán el código del comienzo de LLID 10.

### 5.3.1 Pausa del Enlace Lógico ACL-U

El Director de enlace transmite el paquete actual con información de ACLU, hasta que un ACK recibe, opcionalmente, hasta que un NACK explícito se recibe. Mientras el enlace lógico ACL-Use detiene, el Director de enlace no transmitirá ningún paquete con información ACL-U si la ACL-U se detuvo después de un ACK, el próximo número de la sucesión se utilizará en el próximo paquete. Si la ACL-U se detuvo después de un NAK, el mismo número de la sucesión se utilizará en el próximo paquete y el paquete de un-acknowledged será transmitido una vez el enlace lógico ACL-U es un-paused. Cuándo el enlace lógico ACL-U es un-paused por LM, el Director de enlace puede reasumir paquetes que transmiten con información ACL-U.

## 5.4 Usuario de Datos Enlaces Síncronos Lógicos (SCO-S)

El enlace lógico SCO-S lleva los datos síncronos transparentes de usuario. Este enlace lógico se conserva en el transporte lógico síncrono SCO.

## 5.5 Usuario Extendido de Datos Síncronos (eSCO-s) eSCO-s

El enlace lógico lleva también los datos síncronos transparentes de usuario. Este enlace lógico se conserva en el eSCO lógico, síncrono y prolongado del transporte.

## 5.6 Prioridades del Enlace lógico

El enlace lógico ACL-C tendrá una prioridad más alta que el enlace lógico ACL-U al planificar el tráfico en el ACL compartido el transporte lógico, menos en el caso de retransmisiones de paquetes no reconocidos de ACL será dado la prioridad sobre el tráfico en el enlace lógico ACL-C. El enlace lógico ACL-C debe tener también la prioridad sobre el tráfico en los enlaces SCO-S y eSCO-s pero en las oportunidades lógicas para interpolar los enlaces lógicos se deberán tomar.

# 6 Paquetes

Dispositivos Bluetooth utilizarán los paquetes como esta definido en las secciones siguientes.

## 6.1 Formato General

El formato general de paquete se muestra en la Figura 6,1 Cada paquete consiste en 3 entidades: el código de acceso, el header, y la carga útil. En la figura, el número de bits por la entidad se indica.

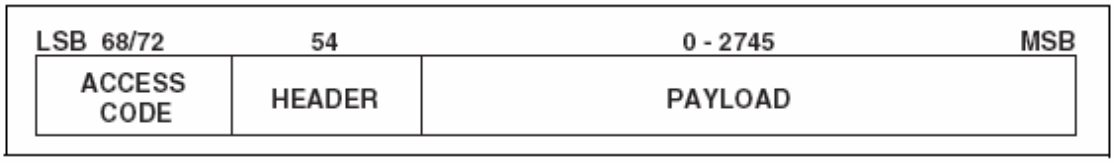



Figura 6.1. Formato General de paquete.

El código de acceso es 72 o de 68 bits y el header es de 54 bits. La carga útil recorre del cero a un máximo de de 2745 bits. Los tipos diferentes del paquete se han definido. El paquete puede consistir en:
• El código de acceso acortado sólo (ver identificación de paquete)
• El código de acceso y el header de paquete
• El código de acceso, el header de paquete y la carga útil.

## 6.2 Ordenador de Bits

El Bit que ordena al definir paquetes y mensajes en la Especificación de Baseband, sigue el formato pequeño de envío. Las reglas siguientes aplican:
• El bit menos significativo (LSB) corresponder a $b_0$;
• El LSB es el primer bit mandado sobre el aire.
• En ilustraciones, el LSB se muestra en el lado izquierdo;

Además, campos de datos generados internamente en nivel de baseband, tal como los campos de header de paquete y longitud de header de carga útil, se transmitirán con el LSB primero. Por ejemplo un parámetro X de 3 bits = 3 son mandados como:

$b_0 b_1 b_2 = 110$

Sobre el aire donde 1 se manda primero y 0 es mandado como último.

## 6.3 Código de Acceso

Cada comienzo de paquete tiene un código de acceso. Si un header de paquete sigue, el código de acceso tiene 72 bits de longitud, de otro modo el código de acceso tiene 68 bits de longitud y es conocido como un código de acceso

acortado. El código de acceso acortado no contiene un trailer. Este código de acceso se utiliza para la sincronización, la compensación de la desviación de DC e identificación. El código de acceso identifica todos paquetes cambiados en un canal físico: todos los paquetes mandados en el mismo canal físico son precedidos por el mismo código de acceso. En el receptor del dispositivo, un correlator que desliza poner en correlación contra el código de acceso y trigger cuando un umbral se excede. Esta señal de trigger se utiliza para determinar el tiempo de recepción.

El código de acceso acortado es utilizado a paging, inquiry, y parked. En este caso, el código de acceso que él mismo se utiliza como un mensaje que señala y ni un header ni una carga útil son presentes. El código de acceso consiste en un preámbulo, una palabra de sincronización, y posiblemente una trama, ver la Figura 6.2. Para detalles ve la Sección 6.3.1.

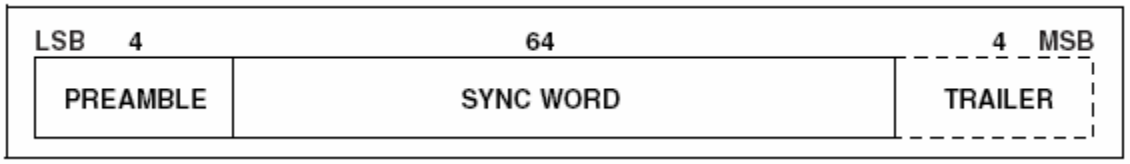



Figura 6.2. Formato de código de acceso.

### 6.3.1 Tipos de Código de Acceso

Los diferentes tipos de código de acceso utilizan las Partes más Bajas y diferentes de la Dirección (los LAPs) de sincronización. El campo LAP del BD_ADDR se explica en la Sección 1,2. Un resumen de los tipos diferentes de código de acceso está en la Tabla 6.1.

| Code type | LAP | Code length | Comments |
|---|---|---|---|
| CAC | Master | 72 | See also Section 1.3 on page 56 |
| DAC | Paged device | 68/72[1] | |
| GIAC | Reserved | 68/72* | |
| DIAC | Dedicated | 68/72* | |

*Table 6.1: Summary of access code types.*

El CAC consiste en un preámbulo, palabra de sincronización, y la trama y su longitud total son de 72 bits. Los mensajes cuando se usa como independientes sin un header, el DAC y IAC no incluyen los bits de trama y son de la longitud de 68 bits.

### 6.3.2 Preámbulo

El preámbulo es una pauta fija de un-cero de 4 símbolos utilizados para facilitar la compensación de DC. La sucesión es o 1010 o 0101, dependiendo de si el LSB de la palabra siguiente de sincronización es 1 o 0, respectivamente. El preámbulo se muestra en la Figura 6.3.

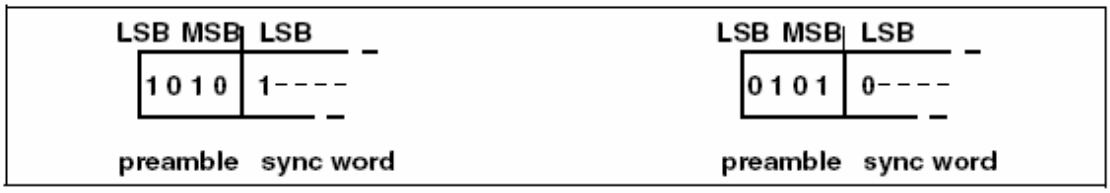



Figura 6.3. Preámbulo

### 6.3.3 Sincronización

Sincronización es una palabra en clave de 64 bits derivada de una dirección de 24 bits (el LAP); para el CAC el LAP del master se utiliza; para el GIAC y el DIAC, los LAP de reserva y se utilizan; para el DAC, el LAP del slave se utiliza. La sincronización garantiza la distancia del Hamming sincronización se basó en LAPs diferentes. Además, las propiedades buenas de la correlación del auto de la palabra de sincronización mejoran la adquisición de tiempo.

#### 6.3.3.1 Definición de la Palabra Sincronización

La palabra sincronización se basa en un código (64.30) de bloque con una cubierta (XOR) de una longitud de 64 bits el ruido seudo-aleatoria (PN) la sucesión. El código expurgado garantiza la distancia grande de Hamming () entre palabras de sincronización se basó en direcciones diferentes. La sucesión de PN mejora las propiedades de la correlación del código de acceso. Los pasos siguientes describen cómo la palabra de sincronización se genera.

1. Genera la secuencia de información
2. XOR con el “información cubierta la parte de la secuencia de cubierta de PN
3. Genera la palabra en clave
4. XOR la palabra en clave con 64 bits de la secuencia de cubierta de PN

La sucesión de la información es generada añadiendo de 6 bits al LAP de 24 bits (da un paso 1). Los bits añadidos son si el MSB del LAP iguala 0. Si el MSB del LAP es 1 los bits añadidos son. El LAP MSB juntos con los bits añadidos constituye una sucesión de XORing de longitud-siete. El propósito es inclusive una sucesión de XORed está a mejora aún más las propiedades de la correlación del auto. En da un paso 2 la información es pre-trepado por XORing con los bits de la secuencia $p_{34} \cdots p_{63}$ de PN (definido en la sección 6.3.3.2). Después que engendrar la palabra en clave (da un paso 3), la sucesión completa de PN es

XORed a la palabra en clave (da un paso 4). Este paso descodifica la parte de información de la palabra en clave. Al mismo tiempo los bits de igualdad de la palabra en clave son trepados. Consecuentemente, la sucesión original del LAP y el Voceador se asegura un papel como una parte de la palabra de sincronización de código de acceso, y de las propiedades cíclicas del código fundamental se quitan. El principio se representa en la Figura 6.4.

Las secuénciales binarias serán denotadas por su D-TRANSFORMADA correspondiente (en que se representa una demora de las unidades de tiempo). Permita ser la sucesión de 63 bits $p'(D) = p'_0 + p'_1 D + \ldots + p'_{62} D^{62}$ PN, dónde $p'_0$ está el primer bit (LSB) saliendo el PRNG (ver la Figura 6,5 es el último bit (MSB). $p'_{62}$ para obtener 64 bits, un cero extra se añade a fines de esta sucesión (así, es igual). $p'(D)$ . Para la conveniencia de marcaje, la cantidad recíproca de este prolongado polinomio $p(D) = D^{63} p'(1/D)$ será utilizada en la sucesión siguiente.

$a(D) = a_0 + a_1 D + \ldots + a_{23} D^{23}$ Esta sucesión en la orden inversa. Denotamos los 24 bits de la dirección en la parte (del LAP) de la dirección de dispositivo Bluetooth por (LSB de la dirección LSB del dispositivo Bluetooth).

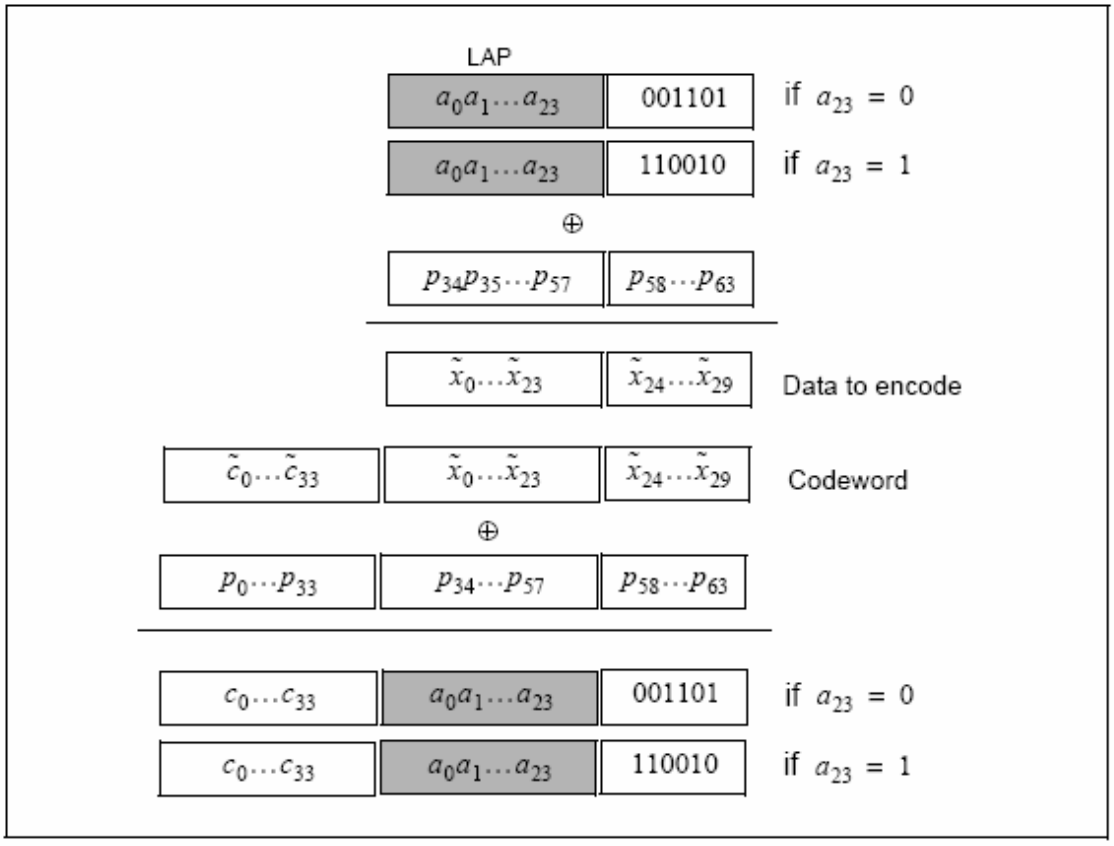



Figura 6.4. Creación de la palabra de sincronización.

El generador de código (64.30) de bloque polinomio se denota $g(D) = (1+D)g'(D)$, dónde $g'(D)$ es el generador del polinomio 157464165547 (la anotación octal) de un código binario primitivo (63.30) código de BCH. Así, en la anotación octal ( $g(D)$ ) es

$$g(D) = 260534236651$$

(EQ 8)
El bit de extremo izquierdo corresponde a la alto-orden () el bit de DC-freefour de coeficiente. La secuencia 0101 y 1010 pueden ser descrita.

$$\begin{cases} F_0(D) = D + D^3, \\ F_1(D) = 1 + D^2, \end{cases}$$

(EQ 9)
Respectivamente.

$$\begin{cases} B_0(D) = D^2 + D^3 + D^5, \\ B_1(D) = 1 + D + D^4, \end{cases}$$

(EQ 10)

Que se utiliza para crear la longitud siete secuenciales de XOR Entonces, el código de acceso será generado por el procedimiento siguiente:

1. Formatear los 30 bits de información para codificar
$x(D) = a(D) + D^{24}B_{a_{23}}(D).$
2. Agregue la información que cubre la parte de la sucesión de cubierta de PN
$\tilde{x}(D) = x(D) + p_{34} + p_{35}D + \ldots + p_{63}D^{29}.$
3. Genere los bits de la igualdad del código (64.30) de bloque
$\tilde{c}(D) = D^{34}\tilde{x}(D) \mod g(D).$
4. Cree la palabra en clave:
$\tilde{s}(D) = D^{34}\tilde{x}(D) + \tilde{c}(D).$
5. Agregue la sucesión de PN: 6. Añada el (DC-LIBERTA) preámbulo y trailer

$$y(D) = F_{c_0}(D) + D^4 s(D) + D^{68}F_{a_{23}}(D).$$

## 6.3.3.2 Generación de Secuencia Seudo-Aleatoria

Para generar el PN en la secuencia el polinomio primitivo $h(D) = 1 + D + D^3 + D^4 + D^6$ será utilizado. El LFSR y su estado se muestran en la Figura 6.5. La secuencia de

PN generada (inclusive el exceso que termina en cero) llega a ser (la anotación hexadecimal) 83848D96BBCC54FC. La producción de LFSR empieza con el bit de extremo izquierdo de esta secuencia de PN. Esto corresponde a $p'(D)$ de la sección previa. Así, utilizando la cantidad $p(D)$ recíproca como cubierta da la sucesión de 64 bits:

$$p = 3F2A33DD69B121C1,$$

(EQ 11)

Donde el bit de extremo izquierdo $p_0 = 0$ es (hay dos ceros iniciales en la representación binaria del dígito hexadecimal 3), y $p_{63} = 1$ es el bit más de la derecha. La figura 6,5: LFSR y el estado que empieza a generar.

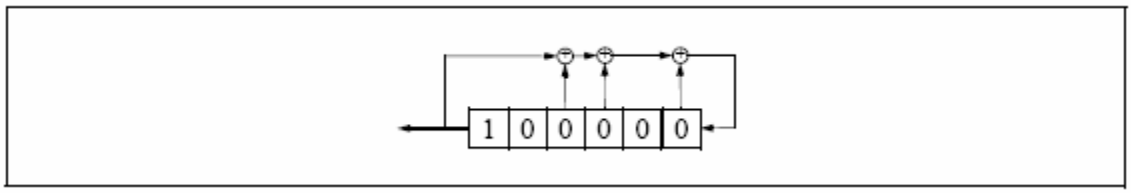



Figure 6.5: LFSR and the starting state to generate p' (D )

.

## 6.3.4 Trailer

Trailer es añadido a la palabra de la sincronización tan pronto como el header de paquete sigue el código de acceso. Esto es típicamente el caso con el CAC, pero como el trailer utiliza también el DAC y IAC, cuando estos códigos se utilizan en paquetes de FHS cambiados durante la respuesta de página y respuesta de inquiry. El trailer es una pauta fija de un-cero de cuatro símbolos. El trailer junto con el tres MSBs del syncword forma una pauta de 7 bits de alternar unos y ceros que se pueden utilizar para la compensación prolongada de DC. La sucesión del trailer es o 1010 o 0101 dependiendo de si el MSB de la palabra de sincronización es 0 o 1, respectivamente. La elección de trailer se ilustra en la Figura 6,6 trailer CAC cuando MSB de la palabra de sincronización es 0 (a), y cuándo MSB de palabra de sincronización es 1 (b).

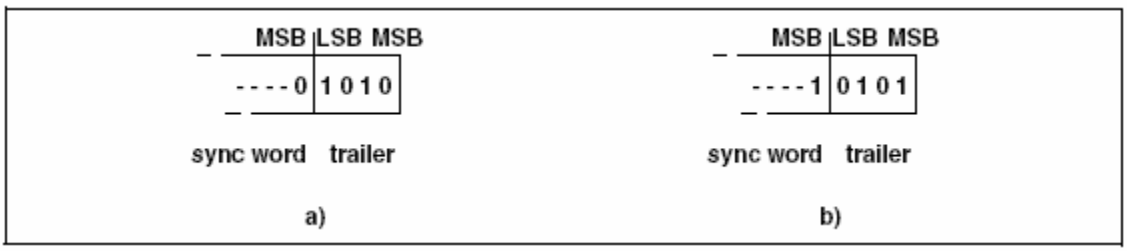



Figure 6.6: Trailer in CAC when MSB of sync word is 0 (a), and when MSB of sync word is 1 (b).

## 6.4 Header del Paquete

El header contiene el control de enlace (LC) de información y consiste en 6 campos:

• LT_ADDR: 3- mordió la dirección lógico del transporte
• TIPO: código de 4 bits de tipo
• El FLUJO: el control de 1 bit del flujo
• ARQN: de 1 bit reconoce la indicación
• SEQN: el número de 1 bit de sucesión
• HEC: error de 8 bits del header

El header total, inclusive el HEC, consiste en de 18 bits, ver Figura 6.7 en página 103, y es codificado con una tasa 1/3 FEC (no mostrado pero descrito en la Sección 7.4) teniendo como resultado un header de 54 bits. El LT_ADDR y los campos del TIPO se mandarán LSB primero.

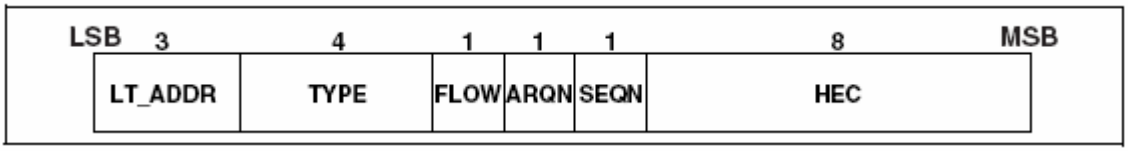



Figura 6.7. Formato del Header.

### 6.4.1 LT_ADDR

El campo LT_ADDR de 3 bits contiene la dirección del transporte lógico para el paquete (ver la Sección 4.2 Este campo indica al slave el destino para un paquete en una slot de la transmisión de master a slave e indica al slave de la fuente para un slave al slot de la transmisión.

### 6.4.2 Tipo

Dieciséis tipos diferentes de paquetes se pueden distinguir. El TIPO de código de 4 bits, especifica cuál de tipo paquete es utilizado. La interpretación del TIPO de código, depende de la dirección del transporte lógico en el paquete. Primero, se determinará si el paquete se manda en un transporte lógico SCO, o en un transporte lógico eSCO, o en un transporte lógico ACL. Entonces se puede determinar cuál tipo de paquete de SCO, paquete de eSCO, o paquete de ACL se han recibido. El TIPO del código determina cuántos slots del paquete actual ocupará (ver la columna de la ocupación de slot (en la Tabla 6.2) Esto permite los receptores no-dirigidos para abstenerse del escuchar el canal durante los slots restantes. En la Sección 6.5. Cada de tipo de paquete es descrito con más detalle.

### 6.4.3 FLUJO

El bit de FLUJO es usado para el control de flujo de paquetes sobre el transporte lógico ACL. Cuando el buffer de RX para el transporte lógico ACL en el destinatario está lleno, hace una Indicación STOP (FLOW=0) retomara STOP de otro dispositivo para transmitir datos temporalmente. La señal STOP sólo afecta paquetes de ACL. Incluyendo sólo paquetes de información de control de enlace (ID, POLL, y paquetes NULL), paquetes SCO y eSCO puedan recibirse. Cuando el buffer de RX puede aceptar datos, una indicación GO (FLOW=1) se retomara. Cuando ningún paquete es recibido, o el HEADER recibido está en error, un GO se asumirá implícitamente. En este caso, el slave puede recibir un nuevo paquete con CRC aunque su buffer de RX todavía no se vacié. El slave devolverá un NAK entonces en respuesta a este paquete aun cuando el paquete pasó el check de CRC.

El bit de FLUJO no se usa en el transporte lógico eSCO o en el enlace lógico ACL-C. Se pondrá a uno en la transmisión y se ignorará en la recepción.

### 6.4.4 ARQN

La indicación 1-bit acknowledgment ARQN es usada para informar la fuente de un traslado exitoso de datos payload con CRC, y puede ser positivo acknowledge (ACK) o negativo acknowledge (NAK) Vea Sección 7.6 para la inicialización y el uso de este bit.

### 6.4.5 SEQN

El bit de SEQN proporciona un esquema de numeración secuencial para ordenar los datos de flujo de paquete. Vea sección 7.6.2 para la inicialización y uso del bit SEQN. Para la transmisión de paquetes, y el uso de un método del sequencing modificado, vea la Sección 7.6.5.

### 6.4.6 HEC

Cada header tiene un header-error-check para checar la integridad del header. El HEC es una palabra de 8-bit (la generación HEC es especificada en la Sección 7.1.1). Antes de generar el HEC, el generador HEC se inicia con un valor de 8-bit para paquetes de FHS enviados en el substate master response, el slave superior usa parte de la dirección (UAP). Para paquetes FHS enviados en inquiry response, el default check intialization (DCI, vea Sección 1.2.1) será usado. En todos los otros casos, el UAP del dispositivo del master será usado.

Después de la inicialización, un HEC será calculado por el header de 10 bits. Antes de checar el HEC, el receptor inicializará los HEC check circuity con el UAPde 8bit apropiados (o DCI). Si el HEC no verifica, el paquete entero será descartado. Más información puede encontrarse en Sección 7.1

## 6.5 Tipos de Paquete

Los paquetes usados en el piconet están relacionados a los transportes lógicos que ellos usan. Se definen tres transportes lógicos con distintos tipos de paquete (vea Sección 4) el transporte lógico SCO, el transporte lógico eSCO o/y transporte lógico ACL. Para cada uno de estos transportes lógicos, pueden definirse 15 diferentes tipos de paquete.

Para indicar los diferentes paquetes en un transporte lógico, es usado el código TYPE 4-bit. Los tipos de paquete son divididos en cuatro segmentos. El primer segmento es reservado para el control de paquetes. Todos los paquetes de control ocupan un time slot. El segundo segmento es reservado para paquetes que ocupan un solo time slot. El tercer segmento es reservado para paquetes que ocupan tres time slot. El cuarto segmento es reservado para paquetes que ocupan cinco time slot. La ocupación de un time slot se refleja en la segmentación y puede derivarse directamente del tipo de código. Tabla 6.2 se resume los paquetes definidos para SCO, eSCO, y transporte lógico ACL tipos de transporte.

| Segment | TYPE code $b_3b_2b_1b_0$ | Slot occupancy | SCO logical transport | eSCO logical transport | ACL logical transport |
|---|---|---|---|---|---|
| 1 | 0000 | 1 | NULL | NULL | NULL |
| | 0001 | 1 | POLL | POLL | POLL |
| | 0010 | 1 | FHS | reserved | FHS |
| | 0011 | 1 | DM1 | reserved | DM1 |
| 2 | 0100 | 1 | undefined | undefined | DH1 |
| | 0101 | 1 | HV1 | undefined | undefined |
| | 0110 | 1 | HV2 | undefined | undefined |
| | 0111 | 1 | HV3 | EV3 | undefined |
| | 1000 | 1 | DV | undefined | undefined |
| | 1001 | 1 | undefined | undefined | AUX1 |
| 3 | 1010 | 3 | undefined | undefined | DM3 |
| | 1011 | 3 | undefined | undefined | DH3 |
| | 1100 | 3 | undefined | EV4 | undefined |
| | 1101 | 3 | undefined | EV5 | undefined |
| 4 | 1110 | 5 | undefined | undefined | DM5 |
| | 1111 | 5 | undefined | undefined | DH5 |

*Tabla 6.2: Paquetes definidos por Synchronous no se por que ustedes tan solo dan de alta and Asynchronous logical transport types.*

### 6.5.1 Tipos Comunes de Paquete

Hay cinco tipos comunes de paquetes. Además de los tipos listados en el segmento 1 de la tabla anterior, el paquete ID también es un tipo común de paquete común pero no se listó en el segmento 1 porque no tiene un paquete header.

#### 6.5.1.1 Paquete ID

La identidad o el paquete ID consiste en el código de acceso de dispositivo (DAC) o código de acceso inquiry (IAC). Tiene una longitud fija de 68 bits. Es un paquete muy robusto desde que usa el receptor de un del bit de correlación para igualar el paquete recibido a la secuencia del bit conocida del paquete ID.

#### 6.5.1.2 Paquete NULL

El paquete NULL no tiene ningún payload y consiste en el código de acceso de canal y sólo del paquete header. Su total longitud fija es de126 bits. El paquete NULL puede devolver el enlace de información de la fuente con respecto al éxito de la anterior transmisión (ARQN), o el estado del buffer RX (FLUJO). El paquete NULL posiblemente no tiene que ser **acknowledged**.

#### 6.5.1.3 Paquetes PULL

El paquete PULL es muy similar al paquete NULL. No tiene un payload, en contraste con el paquete NULL, requiere una confirmación del receptor no es una parte del esquema de ARQ. El paquete PULL no afecta el ARQN y campos de SEQN. En la recepción de un paquete PULL el slave debe responder con un paquete incluso cuando el slave no tenga información para enviar, a menos que el slave tenga compromisos del scatternet en ese time slot. Este retorno de paquete es implícito **acknowledged** del paquete PULL. Este paquete puede ser usado por el master del piconet para los slaves de PULL. Los slaves no transmitirán el paquete PULL.

#### 6.5.1.4 Paquetes FHS

El paquete FHS es un control de paquete con contenido especial, entre otras cosas, el dispositivo de dirección Bluetooth y el reloj de envió. El payload contiene 144 bits de información más un código 16-bit CRC. El payload es codificado con un rate de 2/3 FEC con un grosor de longitud de payload de 240 bits. En la Figura 6.8 se ilustra el formato y contenido del payload de FHS. El payload consiste en once campos. El paquete FHS es usado en page master response, inquiry response y en un role Switch. El paquete FHS contiene información del reloj en tiempo-real. Esta información del reloj se pondrá al día antes de cada retransmisión. La retransmisión del FHS payload es diferente que las

retransmisiones de payloads de datos ordinarios donde el mismo payload se usa para cada retransmisión. El paquete FHS es usado para frecuencias hop, sincronizadas antes del establecimiento del canal, o cuando un piconet existente cambia a un nuevo piconet.

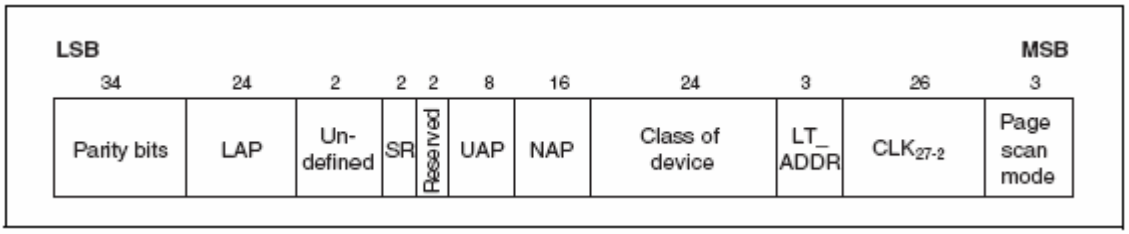



Figura 6.8. El Formato del payload de FHS.

| | |
|---|---|
| **Parity bits** | This 34-bit field contains the parity bits that form the first part of the sync word of the access code of the device that sends the FHS packet. These bits are derived from the LAP as described in Section 1.2 on page 55. |
| **LAP** | This 24-bit field shall contain the lower address part of the device that sends the FHS packet. |
| **Undefined** | This 2-bit field is reserved for future use and shall be set to zero. |
| **SR** | This 2-bit field is the scan repetition field and indicates the interval between two consecutive page scan windows, see also Table 6.4 and Table 8.1 on page 135 |
| **Reserved** | This 2-bit field shall be set to 10. |
| **UAP** | This 8-bit field shall contain the upper address part of the device that sends the FHS packet. |
| **NAP** | This 16-bit field shall contain the non-significant address part of the device that sends the FHS packet (see also Section 1.2 on page 55 for LAP, UAP, and NAP). |
| **Class of device** | This 24-bit field shall contain the class of device of the device that sends the FHS packet. The field is defined in Bluetooth Assigned Numbers (https://www.bluetooth.org/foundry/assignnumb/document/assigned_numbers). |
| **LT_ADDR** | This 3-bit field shall contain the logical transport address the recipient shall use if the FHS packet is used at connection setup or role switch. A slave responding to a master or a device responding to an inquiry request message shall include an all-zero LT_ADDR field if it sends the FHS packet. |
| **$CLK_{27-2}$** | This 26-bit field shall contain the value of the native clock of the device that sends the FHS packet, sampled at the beginning of the transmission of the access code of this FHS packet. This clock value has a resolution of 1.25ms (two-slot interval). For each new transmission, this field is updated so that it accurately reflects the real-time clock value. |
| **Page scan mode** | This 3-bit field shall indicate which scan mode is used by default by the sender of the FHS packet. The interpretation of the page scan mode is illustrated in Table 6.5. |

Tabla 6.3.Descripción del payload FHS

El dispositivo sending de FHS pondrá los bits SR según en la tabla 6.4.

| SR bit format $b_1b_0$ | SR mode |
|---|---|
| 00 | R0 |
| 01 | R1 |
| 10 | R2 |
| 11 | reserved |

Tabla 6.4. Contenido del campo de SR

El dispositivo sending de FHS pondrá los bits al modo page scan según la tabla 6.5

| Bit format $b_2b_1b_0$ | Page scan mode |
|---|---|
| 000 | Mandatory scan mode |
| 001 | Reserved for future use |
| 010 | Reserved for future use |
| 011 | Reserved for future use |
| 100 | Reserved for future use |
| 101 | Reserved for future use |
| 110 | Reserved for future use |
| 111 | Reserved for future use |

Tabla 6.5. Contenido del campo del modo page scan

LAP, UAP, y NAP forman juntos los 48-bit del Dispositivo de dirección Bluetooth y del dispositivo enviado del paquete FHS. Usando los bits de paridad y el LAP, el receptor puede construir directamente el canal de código de acceso y el envió del Paquete FHS.

Al inicializar el HEC y CRC para el paquete de FHS de inquiry response, el UAP será el DCI.

### 6.5.1.5 Paquete DM1

DM1 es parte del segmento 1 para soportar el control de mensajes en cualquier transporte lógico que permite al paquete (vea tabla 6.2). Sin embargo, él también puede llevar datos de usuario regularmente. Entonces el paquete DM1 puede considerarse como un Paquete de ACL, esto será discutido en la Sección 6.5.4.

### 6.5.2 Paquetes SCO

HV y DV son paquetes usados en el transporte lógico síncrono SCO. Los paquetes HV no incluyen un CRC y no serán retransmitidos. Los paquetes de DV incluyen un CRC en la sección de datos, pero no en la sección de datos síncronos. En la sección de paquetes de datos de DV serán retransmitidos. Los paquetes SCO rutean al puerto de I/O síncrono. Cuatro paquetes se permiten en el transporte lógico SCO. HV1, HV2, HV3 y DV. Estos paquetes se usan típicamente para transmisión de 64kb/s pero puede usarse para los datos síncronos transparentes.

#### 6.5.2.1 Paquete HV1

El paquete HV1, paquete de10 bytes de información. Los bytes son protegidos con un rate de 1/3 FEC. Ningún CRC está presente. El payload tiene una longitud fija de 240 bits. No hay ningún payload presente en el header.

#### 6.5.2.2 Paquete HV2

El paquete HV2 paquete tiene 20 bytes de información. Los bytes son protegidos con un rate de 2/3 FEC. Ningún CRC está presente. El payload tiene una longitud fija de 240 bits. No Hay ningún payload presente en el header.

#### 6.5.2.3 Paquete HV3

El paquete HV3 paquete tiene 30 bytes de información. Los bytes no están protegidos por FEC. Ningún CRC está presente. El payload tiene una longitud fija de 240 bits. No hay ningún payload presente en el header.

#### 6.5.2.4 Paquete DV

El paquete DV es una combinación de paquete de datos-voz. El paquete DV sólo será usado en lugar de un paquete HV1. El payload es dividido dentro de un campo de voz de 80 bits y un campo de datos que contienen 150 bits, vea Figura 6.9. El campo de la voz no esta protegido por FEC. El campo de datos tiene entre 1 y 10 bytes de información (incluyendo el 1-byte del header del payload) incluyendo 16-bit de CRC. El campo de datos es codificado con un rate de 2/3 FEC. Desde que el paquete de DV tiene que ser enviado regularmente a intervalos debido a su contenido síncrono, abajo se lista los tipos de paquete SCO. Los campos de voz y datos serán tratados por separado. El campo de la voz será manejado de la misma manera como los datos SCO normalmente y nunca serán retransmitidos; es decir, el campo de la voz siempre será nuevo. El campo de los datos se verifica para los errores y debe ser retransmitido si es necesario. Cuando el campo de los datos asíncronos en el paquete DV no ha sido acknowledged,

antes de que transporte lógico SCO se termine, el campo de los datos asíncronos será retransmitido en un paquete DM1.

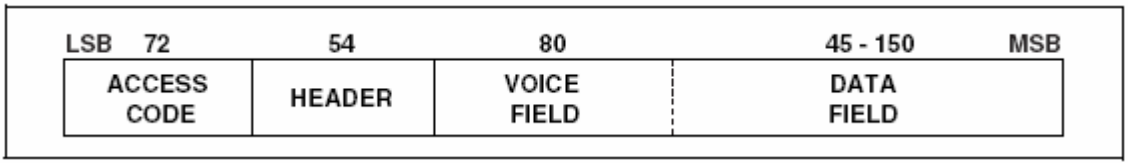



Figura 6.9: Formato de paquete DV

## 6.5.3 Paquetes eSCO

Se usan paquetes EV en el transporte lógico síncrono eSCO. Los paquetes pueden incluir y aplicar un CRC y retransmitir sin ningún **acknowledgetment,** la recepción apropiada se recibe dentro de la retransmisión de window. Los paquetes eSCO puede rutearse al puerto de I/O síncrono. Tres paquetes eSCO han sido definidos. Los paquetes del eSCO pueden usarse para 64kb/s de transmisión speech y para datos transparentes a 64kb/s y otras rates.

### 6.5.3.1 Paquete EV3

El paquete EV3 tiene entre 1 y 30 bytes de información más 16-bit de código CRC código. Los bytes no son protegidos por FEC. El paquete EV3 puede cubrir a un solo time slot. No hay presente ningún payload header. La longitud del payload es fija durante el LMP el arreglo eSCO es fijo hasta que el enlace es removido o re-negociado.

### 6.5.3.2 Paquete EV4

El paquete EV4 tiene entre 1 y 120 bytes de información más 16-bit de código CRC. El paquete EV4 puede cubrir a tres time spot. La ventaja es que la información de los bits CRC son codificado con una proporción 2/3 FEC. No hay ningún payload header presente. La longitud del payload es fija durante el LMP, el arreglo eSCO es fijo hasta que el enlace es removido o re-negociado.

### 6.5.3.3 Paquete EV5

El paquete EV5 tiene entre 1 y 180 vides de información más 16-bit de código CRC. Los bytes no son protegidos por FEC. El paquete EV5 puede cubrir a tres time slot. No hay ningún payload header presente. La longitud del payload es fija durante el LMP el arreglo eSCO es fijo hasta que el enlace es removido o re-negociado.

### 6.5.4 paquetes de ACL

Se usan paquetes de transporte lógico ACL asíncrono. La información puede ser llevada por un usuario de datos del usuario o control de datos.

#### 6.5.4.1 Paquete DM1

El paquete DM1 sólo lleva información de datos. El payload tiene entre 1 y 18 bytes de información (incluso el 1-byte header del payload) más 16-bit de código CRC. El paquete DM1 ocupa un solo time slot. La información más los bits CRC son codificados con un rate de 2/3 FEC. El header del payload en el paquete DM1 es1 byte largo, vea Figura 6.10. El indicador de longitud en el header del payload especifica el número de bytes del usuario (excluyendo el header del payload y el código CRC).

#### 6.5.4.2 Paquete DH1

Este paquete es similar al paquete DM1, sólo que la información en el payload FEC esta codificada. Como resultado, el paquete DH1 tiene entre 1 y 28 bytes de información (incluso el 1-byte del header del payload) más 16-bit código CRC, el paquete DH1 ocupa un solo time slot.

#### 6.5.4.3 Paquete DM3

El paquete DM3 puede ocupar tres time slot. El payload tiene entre 2 y 123 bytes de información (incluso el 2-byte del header del payload) más 16-bit de código CRC. La información más los bits CRC son codificados con un rate de 2/3 FEC. El header del payload en el paquete DM3 es 2 byte largos, vea Figura 6.10 El indicador de longitud en el header del payload específica el número de bytes del usuario (excluyendo el header del payload y el código CRC).

#### 6.5.4.4 Paquete DH3

Este paquete es similar al DM3 paquete, sólo que la información en el payload en el FEC no esta en código. Como resultado, el DH3 paquete tiene entre 2 y 185 bytes de información (incluso el 2-byte del header del payload) más 16-bit de código CRC. El paquete DH3 puede ocupar tres time slot.

#### 6.5.4.5 Paquete DM5

El paquete DM5 puede ocupar a time slot. El payload tiene entre 2 y 226 bytes de información (incluso el 2-byte del header del payload) más 16-bit de código CRC. El header del payload en el paquete DM5 paquete es de 2 bytes largos. La

información más los bits CRC son codificados con un rate de 2/3 FEC. El header del payload en el paquete DM5 es 2 byte largos, vea Figura 6.10. El indicador de longitud en el header del payload especifica el número de bytes del usuario (excluyendo el header del payload y el código CRC).

### 6.5.4.6 Paquete DH5

Este paquete es similar al DM5, sólo que la información en el payload en el FEC no esta en código. Como resultado, el paquete DH5 tiene entre 2 y 341 bytes de información (incluyendo el 2-byte header del payload) más 16-bit de código CRC. El paquete DH5 puede ocupar cinco time slot.

### 6.5.4.7 Paquete AUX1

Este paquete se parece a un paquete DH1 no tiene ningún código de CRC. El paquete AUX1 tiene entre 1 y 30 bytes de información (incluso el 1-byte del header del payload). El paquete AUX1 ocupa un solo time slot. El paquete AUX1 no será usado para los enlaces lógicos ACL-U o ACL-C. Un paquete AUX1 paquete desecharse.

## 6.6 Formato PAYLOAD

En el payload, dos campos son distinguidos: el campo de los datos síncronos y el campo de los datos asíncronos. Los paquetes ACL sólo tienen el campo de los datos asíncronos en el SCO y los paquetes eSCO. Sólo tienen el campo de los datos síncronos con la excepción de los paquetes de DV que tienen ambos.

### 6.6.1 Campo de los Datos Síncronos

En SCO, el campo de los datos síncronos tiene una longitud fija y sólo consiste de la porción de entidad de datos síncronos. Ningún header del payload está presente. En eSCO, el campo de los datos síncrono consiste en dos segmentos: una entidad síncrono de los datos y un código de CRC. Ningún header del payload está presente.

### I. La entidad de los Datos Síncronos

Para HV y paquetes DV, la longitud de entidad de datos síncronos es fija. Para paquetes EV la longitud de la entidad de datos síncronos es negociada durante el arreglo LMP de eSCO. Una vez negociado, la longitud síncrona de la entidad de datos se queda constante a menos que se renegocie. La longitud de la entidad de datos síncrona puede ser diferente para cada dirección del transporte lógico eSCO.

## II. Código CRC

El 16-bit CRC en el payload se genera como se especifica en la Sección 7.1. El 8-bit UAP del master es acostumbrado a iniciar el generador de CRC.

## 6.6.2 Campo de los Datos Asíncronos

Los paquetes ACL tienen un campo de los datos asíncronos que consiste en dos o tres segmentos: Un header del payload, una entidad del payload, y posiblemente un código de CRC (los AUX1 el paquete no lleva un código de CRC).

## 1. Payload header

El payload header es mucho tiempo uno o dos bytes. Los paquetes en e el segmento uno y dos tienen un 1-byte del payload header; los paquetes en los segmentos tres y cuatro tienen un 2-byte payload header. El payload header específica el enlace lógico (2-bit Indicación de LLID), el control de flujo en los canales lógicos (1-bit indicación de FLUJO), y tiene un indicador de longitud de payload (5 bits y 9 bits para 1-byte y 2 -header de payload de byte, respectivamente.) En el caso de un 2-byte payload header, el indicador de longitud está extendido por cuatro bits en el próximo byte. Permaneciendo cuatro bits del segundo byte son reservados para el uso futuro y se pondrán a cero. Los formatos del 1-byte y el 2-byte headers del payload se muestran en Figura 6.10 en y Figura 6.11.

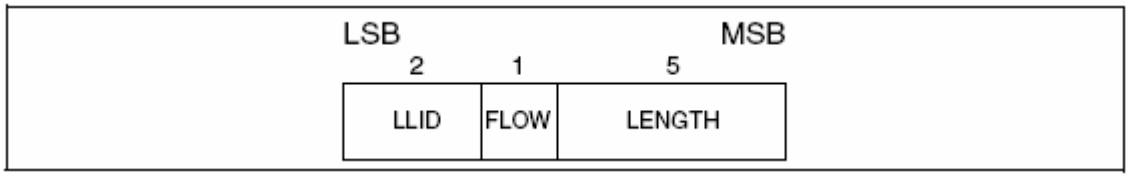



Figura 6.10: Payload header formato para un solo-slot paquetes ACL.

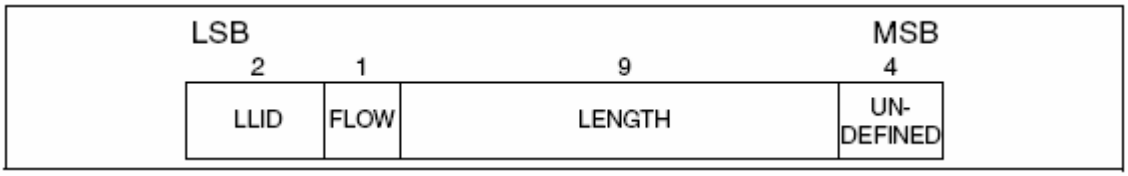



Figura 6.11: el payload header formato para multi-slots paquetes ACL.

El campo LLID se transmitirá primero, el campo de longitud al último. En la tabla 6.6, se listan más detalles sobre el contenido del campo de LLID.

| LLID code $b_1b_0$ | Logical Link | Information |
|---|---|---|
| 00 | NA | undefined |
| 01 | ACL-U | Continuation fragment of an L2CAP message |
| 10 | ACL-U | Start of an L2CAP message or no fragmentation |
| 11 | ACL-C | LMP message |

Tabla 6.6: Enlace Lógico LLID y contenido de campo

Un mensaje L2CAP puede fragmentarse en varios paquetes. Código “10” será usado para un paquete ACL-U que lleva el primer fragmento de semejante mensaje; código “01”se usarán para continuar fragmentos. Si no hay ninguna fragmentación, el código 10 se usará para cada paquete. Código 11 se usarán para los mensajes de LMP. Código 00 es reservado para el uso futuro.

El indicador de flujo en el payload se usa para controlar el flujo de los niveles L2CAP. Se usa para controlar el flujo del enlace lógico. FLOW=1 como medios FLOW-ON (VA) y FLOW=0 medios FLOW-OFF de (STOP). Después de una nueva conexión se tiene establecido que el indicador de flujo se pondrá en GO. Cuando un dispositivo recibe un payload header con el bit de flujo se pondrá en STOP, detendrá la transmisión de paquetes ACL antes de una cantidad adicional de datos del payload enviados. Esta cantidad se define como el retraso de control de flujo, expresado como un número, de bytes. El retraso corto de control de flujo, en el buffering del otro dispositivo deberá indicar esta función. El retraso de control de flujo no excederá 1792 bytes (7 × 256 bytes). Para permitir a los dispositivos perfeccionar la selección de la longitud del paquete y espacio del buffer, el retraso de control de flujo de una aplicación dada, se proporcionará en el mensaje de LMP_features_res.

Si un paquete que contiene el bit del flujo del payload el STOP es recibido, con un paquete header valido, pero con un payload malo, el bit del control del flujo del payload se ignorará. Se recibirán los baseband que ACK contuvo en el header del paquete y un paquete ACL que puede transmitirse. Cada ocurrencia de esta situación permite enviar un paquete de ACL extenso a pesar de la demanda de control de flujo enviándose vía bit de control de flujo header-payload. Se recomienda que dispositivos que usan el payload header de flujo de control de bit deben asegurar que en ninguna situación. Se envían paquetes de ACL hasta el payload de flujo control de bit sino se ha recibido correctamente.

Esto puede ser logrado simultáneamente encendiendo el flujo de bit en el header del paquete y guardándolo mas adelante hasta que un ACK recibe anteriormente (ARQN=1).

Éste será típicamente sólo un round trip time. Desde que les faltan un payload CRC, no deben usarse paquetes AUX1 con un payload de control de flujo de STOP.

El controlador de recurso de baseband es responsable de poner y procesar el control de flujo en el payload header. El bit de control de flujo se llevará a cabo en tiempo-real a nivel del paquete por el controlador de enlace vía bit de control de flujo del header. (Ver Sección 6.4.3). Con el bit de control de flujo del payload, el tráfico extremo remoto puede controlarse. Se permitía generar y enviar un paquete ACL con cero de longitud de payload en estado de flujo independiente de L2CAP start-fragment e indicaciones de continuación-fragmento (LLID=10 y LLID=01) también retenga su significado cuando la longitud del payload es igual a cero (es decir un vacío en el start-fragment no se enviará en el medio de un paquete ACL-U de transmisión continua. siempre es seguro enviar un paquete de ACL con length=0 y LLID=01. El bit de flujo del payload tiene su propio significado para cada enlace lógico (ACL-U o ACL-C), tabla 6.7. En el enlace lógico ACL-C, no se aplica el control de flujo y el bit de flujo del payload siempre se pondrán a uno.

| LLID code $b_1b_0$ | Usage and semantics of the ACL payload header FLOW bit |
|---|---|
| 00 | Not defined, reserved for future use. |
| 01 or 10 | Flow control of the ACL-U channel (L2CAP messages) |
| 11 | Always set FLOW=1 on transmission and ignore the bit on reception |

*Table 6.7: Use of payload header flow bit on the logical links.*

En el indicador de longitud se pondrá al número de bytes (ejemplo 8-bit palabras) en el payload que excluye el payload header y el código de CRC; es decir la entidad payload. Con referencia a la Figura 6.10 y Figura 6.11, el MSB de el campo de longitud en un header 1-byte es el último (right-most) bit en el payload header; el MSB del campo de longitud en un header 2-bytes el cuarto bit (de salida) del segundo byte en el payload header.

## 2. l Entidad de Payload

La entidad del payload incluye la información del usuario y determina la efectividad throughput del usuario. Se indica la longitud de la entidad del payload en la longitud del campo del payload header.

## 3. CRC Generación de Código

El código de 16-bit de chequeo de redundancia cíclica en el payload se genera como es especificado en la Sección 7.1. Antes de determinar CRC codifique, un valor de 8-bit se usa para inicializar el generador de CRC. Para codificar CRC en los paquetes FHS es enviado en un sub-estado master response, el UAP del slave es usado. Para el paquete FHS enviado en sub-estado de inquirí response, el DCI (vea la Sección 1.2.1) se usa. Para todos los otros paquetes, el UAP del Master es usado.

## 6.7 Resumen de Paquete

Un resumen de los paquetes y sus características se muestran a continuación.

| Type | Payload (bytes) | FEC | CRC | Symmetric Max. Rate | Asymmetric Max. Rate |
|---|---|---|---|---|---|
| ID | na | na | na | na | na |
| NULL | na | na | na | na | na |
| POLL | na | na | na | na | na |
| FHS | 18 | 2/3 | yes | na | na |

*Tabla 6.8: Link control packets*

| Type | Payload Header (bytes) | Payload (bytes) | FEC | CRC | Symmetric Max. Rate (kb/s) | Asymmetric Max. Rate (kb/s) | |
|---|---|---|---|---|---|---|---|
| | | | | | | Forward | Reverse |
| DM1 | 1 | 0-17 | 2/3 | yes | 108.8 | 108.8 | 108.8 |
| DH1 | 1 | 0-27 | no | yes | 172.8 | 172.8 | 172.8 |
| DM3 | 2 | 0-121 | 2/3 | yes | 258.1 | 387.2 | 54.4 |
| DH3 | 2 | 0-183 | no | yes | 390.4 | 585.6 | 86.4 |
| DM5 | 2 | 0-224 | 2/3 | yes | 286.7 | 477.8 | 36.3 |
| DH5 | 2 | 0-339 | no | yes | 433.9 | 723.2 | 57.6 |
| AUX1 | 1 | 0-29 | no | no | 185.6 | 185.6 | 185.6 |

*Table 6.9: ACL packets*

| Type | Payload Header (bytes) | Payload (bytes) | FEC | CRC | Symmetric Max. Rate (kb/s) |
|---|---|---|---|---|---|
| HV1 | na | 10 | 1/3 | no | 64.0 |
| HV2 | na | 20 | 2/3 | no | 64.0 |
| HV3 | na | 30 | no | no | 64.0 |
| DV[1] | 1 D | 10+(0-9) D | 2/3 D | yes D | 64.0+57.6 D |
| EV3 | na | 1-30 | No | Yes | 96 |
| EV4 | na | 1-120 | 2/3 | Yes | 192 |
| EV5 | na | 1-180 | No | Yes | 288 |

*Table 6.10: Synchronous packets*

## 7 Proceso de Bitstream

Los dispositivos Bluetooth usarán el bitstream, que procesa esquemas que son definidos en las secciones siguientes. Antes de que el payload se envíe sobre el área de interface, algunos bits realizan ejecuciones en el transmisor para aumentar fiabilidad y seguridad. Un HEC es agregado al header del paquete, los bits del header son scrambreld con una palabra en blanco y el FEC codificado se aplica. En el receptor, los procesos inversos se llevan a cabo. Figura 7.1 se muestras los procesos llevados fuera del paquete header para que ambos transmitan y reciban a la vez todos los procesos del header de bit son obligatorios.

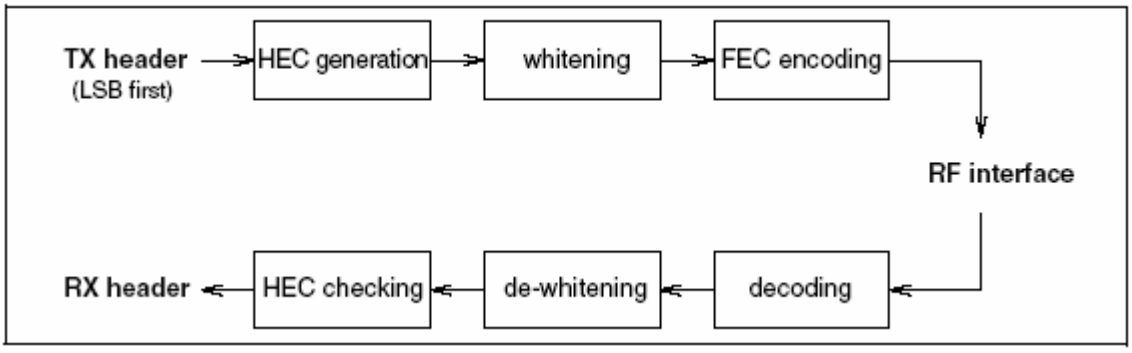



Figura 7.1: Procesos del header de bit

En la Figura 7.2 se muestran los procesos en los que pueden llevarse a cabo el Payload. Además de los procesos definidos para el header del paquete, encryption, puede aplicarse en el payload. Sólo blanqueando y des-blanqueando, como se explicó en la Sección 7.2, es obligatorio para cada payload; todos los

otros procesos son optativos y depende del tipo de paquete (vea Sección 6.6) y si la encryption se habilita. En Figura 7.2. Optativo los procesos son indicados por los bloques subrayados.

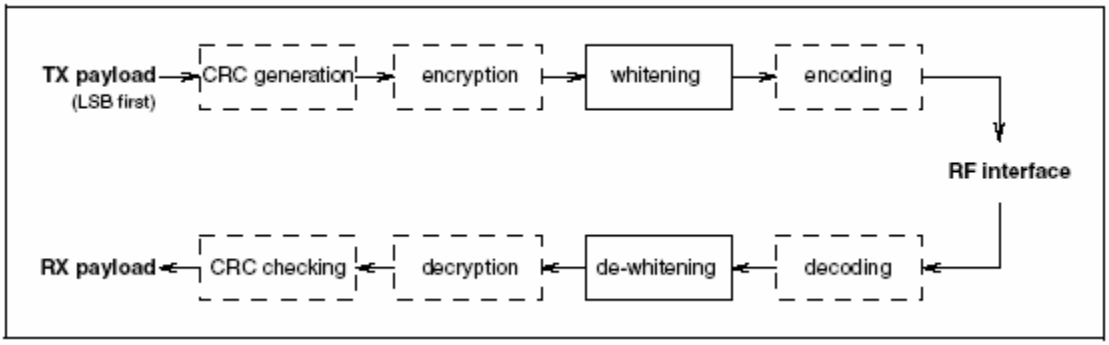



Figura 7.2: Procesos del payload de bit

## 7.1 Verificación de Error

El paquete puede verificarse para errores o para la mala entrega que usa el canal de código de acceso, el HEC en el header, y el CRC en el payload. El paquete de recepción, el código de acceso se verifican primero. Desde los 64-bit word de la sincronización en el canal de código de acceso se derivan los 24-bit del master LAP, esto se verifica si el LAP es correcto, y previene al receptor aceptar un paquete de otro piconet (con tal de que el campo del LAP del BD_ADDR del master sea diferente).Normalmente se inicializan los cómputos de HEC y CRC con el UAP del master. Aunque el código de acceso puede ser el mismo para dos piconets, los valores de UAP diferentes causarán que el HEC y CRC típicamente puedan fallar. Sin embargo, hay una excepción donde ningún UAP común está disponible en el transmisor y receptor. Éste es el caso cuando se generan el HEC y CRC para el paquete FHS en el subestado de inquiry response. En este caso el valor de DCI se usará.

La generación y verificación del HEC y CRC están resumidas en la Figura 7.5 y en la Figura 7.8. Antes de calcular el HEC o CRC, el cambio se registra en los generadores de HEC/CRC se inicializará con los 8-bit UAP (o DCI) el valor. Entonces el header y la información del payload se cambiarán en los HEC y generadores de CRC, respectivamente (con el LSB primero).

### 7.1.1 Generación de HEC

El HEC que LFSR genera se ve en la figura 7.3. El generador del polinomio es polinomio. $g(D) = (D+1)(D^7+D^4+D^3+D^2+1) = D^8+D^7+D^5+D^2+D+1$ Inicialmente este circuito se pre-cargará con los 8-bit UAP tal que el LSB del UAP (denotó $UAP_0$) va al left-most el elemento de registro de cambio, y, UAP7 van a el right-most del elemento. El estado inicial del HEC LFSR se ve en la Figura 7.4. Entonces los datos se cambiarán con el interruptor que S puso en posición 1 cuando el último bit de los

datos ha sido clocked en el LFSR, el interruptor S se pondrá en posición 2, y, el HEC puede leerse fuera del registro. Los bits de LFSR se leerán fuera del derecho a la izquierda (es decir, el bit en posición 7 es el primero para ser transmitido, seguido por el momento en posición 6, etc.).

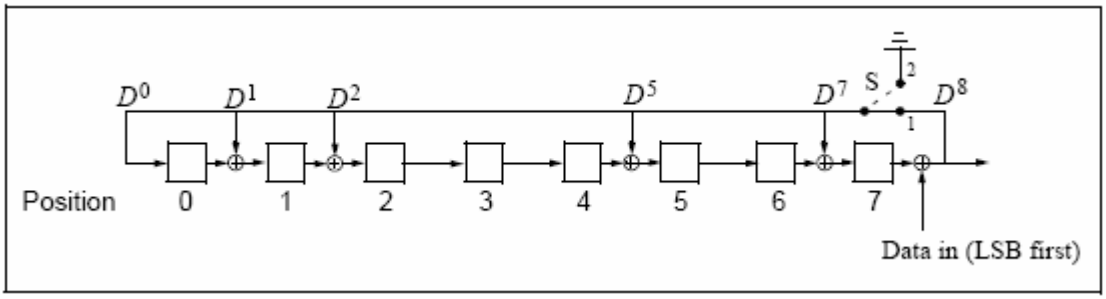



Figura 7.3: El circuito de LFSR que genera el HEC.

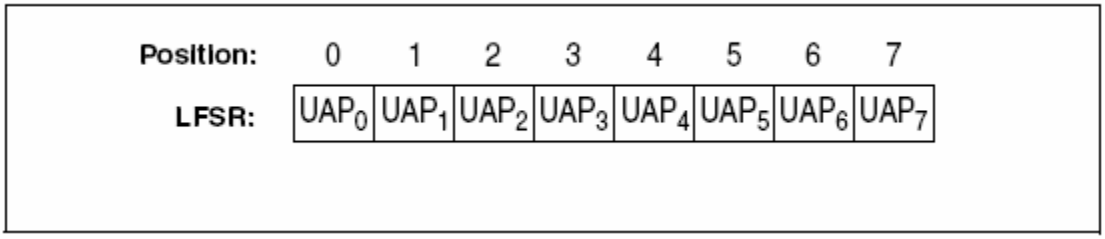



Figura 7.4: El estado Inicial del HEC del circuito generador.

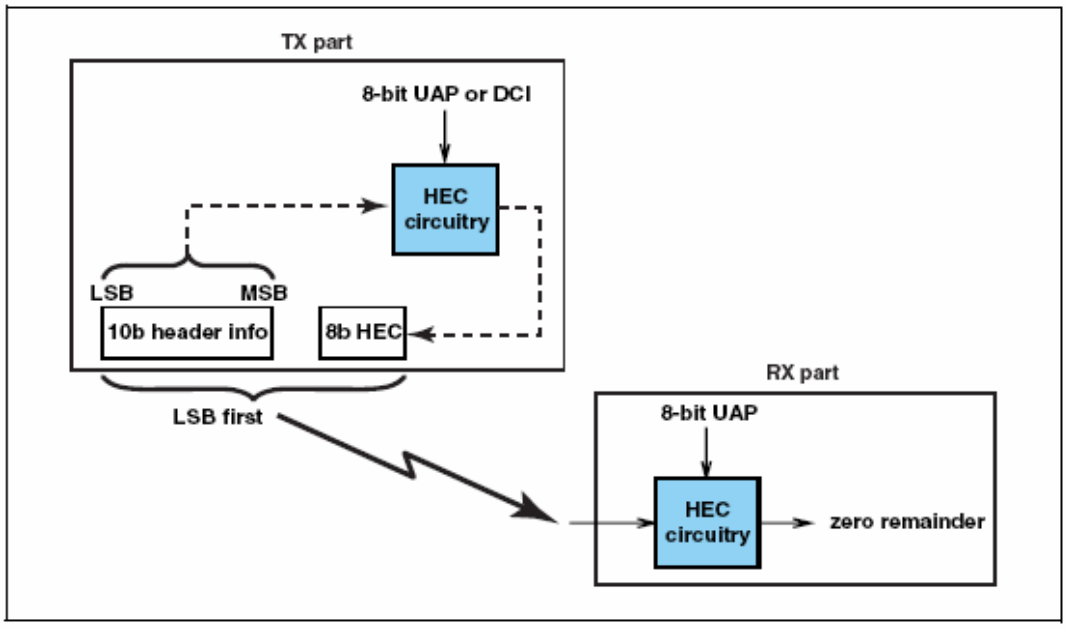



Figura 7.5: la generación y verificación del HEC.

### 7.1.2 Generación de CRC

El bit 16 LFSR para el CRC se construye semejantemente al HEC que usa el Generador de CRC-CCITT polinómico (es decir 210041 en representación octal) (vea Figura 7.6 en la página 120). Para este caso, los 8 left-most se cargarán bits inicialmente con los 8-bit UAP ($UAP_0$ a la izquierda y $UAP_7$ al derecho) mientras los 8 right-most se restablecerán para poner a cero. El estado inicial del bit16 que LFSR especifica en la Figura 7.7 en la página 120. El interruptor S se pondrá en posición 1 mientras los datos se cambien. Después de que el último bit ha entrado al LFSR, el interruptor se pondrá en posición 2, y, los contenidos del registro serán transmitidos, de derecha a izquierda (es decir, empezando con posición 15, posición 14, etc.).

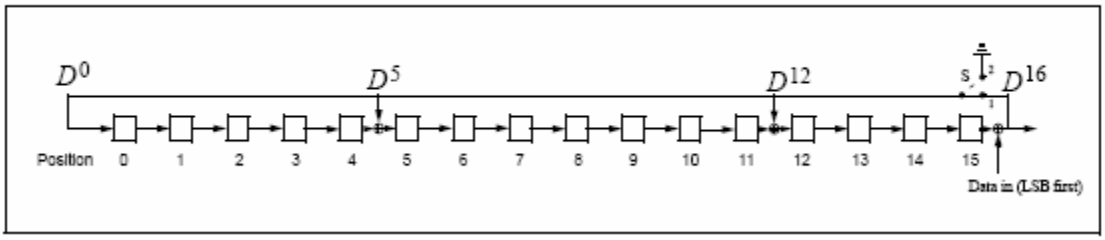



Figura 7.6. El circuito de LFSR que genera el CRC.

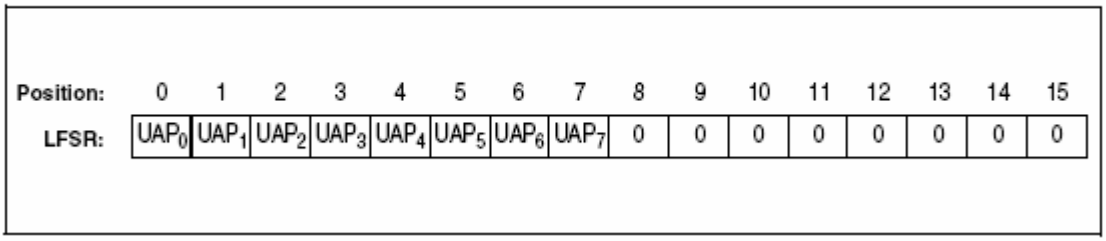



Figura 7.7. Estado Inicial del CRC circuito generador.

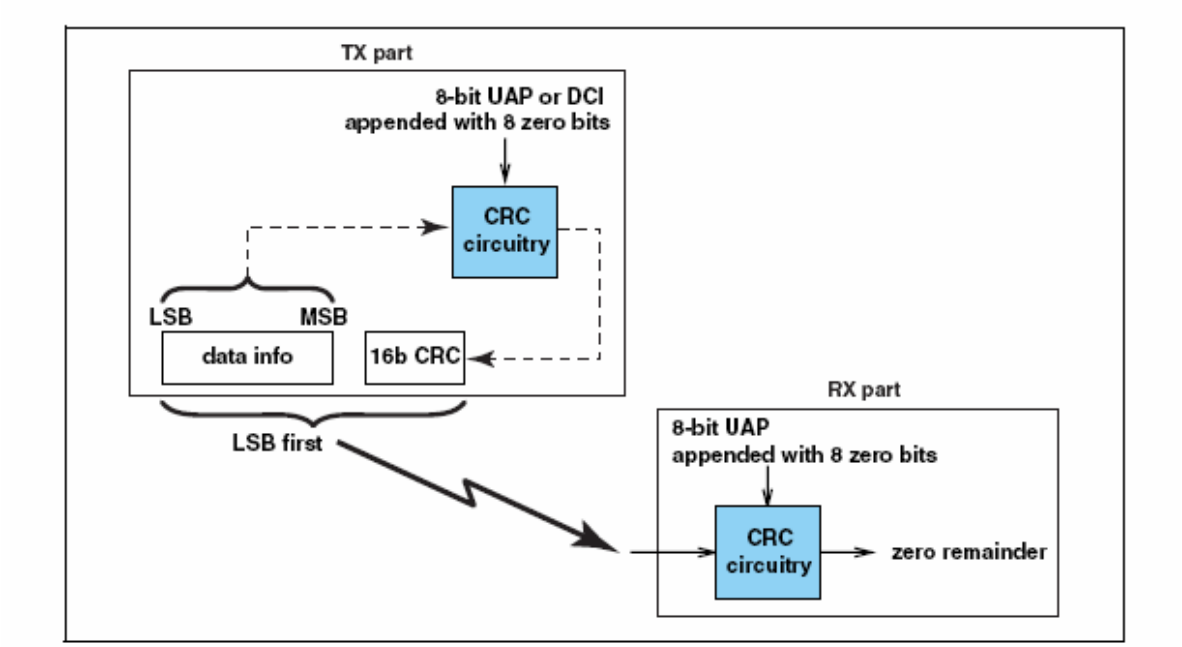



Figura 7.8. La generación verificación de CRC.

## 7.2 Datos whitening

Antes de la transmisión, el header y los payload se correrán con un datos whitening word para aleatorizar los datos de modelos redundantes y para minimizar el paquete DC. El descrambling se realizará antes de poner la codificación de FEC.
En el receptor, los datos recibidos estarán descrambled que usan el mismo whitening word generado en el destinatario. Los descrambling se realizarán después de la codificación del FEC.

Whitening word palabra se genera con el polinomio $g(D) = D^7 + D^4 + 1$ (es decir, 221 en representación del octal) y será como consecuencia X0Red con el header y el payload. Whitening word se genera con el cambio de la regeneración lineal registro mostrado en la Figura 7.9 en la página 121. Antes de cada transmisión, el cambio del registro se inicializará con una porción del master del reloj Bluetooth, $CLK_{6-1}$, es extendido con un MSB de valor uno. Esta inicialización se llevará fuera con $CLK_1$ escriben la posición 0, CLK2 escribe la posición 1, etc. Una excepción es el paquete FHS envía durante page response e inquiry, donde la inicialización de el registro whitening se llevará a cabo diferente. En lugar del reloj del master, la entrada –X usada en inquirí o page response (dependiendo de estado actual) la rutina se usará, vea tabla 2.2. El 5-bit valor se extenderá con dos MSBs de valor 1. Durante la inicialización del registro, el LSB de X (es decir, $X_0$) será escrito para posicionar 0, se escribirá X1 posicionar 1, etc.,

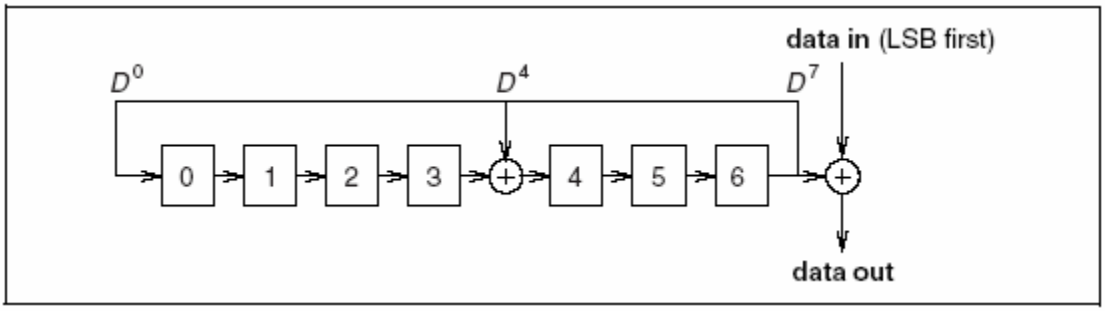



Figura 7.9. Datos whitening LFSR.

Después de la inicialización, el header del paquete y el payload (incluso el CRC) es whitened. Los payload whitening continuarán en el estado whitening LFSR que tenía la finalidad del HEC. No habrá ninguna re-inicialización del registro de cambio entre el header del paquete y payload. El primer bit de "los datos en" la sucesión será los LSB del header del paquete.

## 7.3 Corrección del Error

Hay tres esquemas de corrección de error definidos para Bluetooth:

- 1/3 rate FEC
- 2/3 rate FEC
- ARQ esquema para los datos

El propósito del esquema de FEC en el payload de los datos es reducir el número de retransmisiones. Sin embargo, en un ambiente error-libre razonable, FEC da innecesariamente la reducción del throughput. Por consiguiente, se han guardado definiciones del paquete Sección 6 en la página 97 del uso flexible de FEC en el payload y no, produciendo los DM y paquetes DH para el transporte lógico ACL, paquetes HV para el transporte lógico SCO, y paquetes EV para el transporte lógico eSCO. El header del paquete siempre es protegido por una 1/3 rate FEC desde que contiene importante información del enlace y se diseña para resistir más errores de bit.

La corrección se mide para enmascarar errores en el decodificador de la voz, no es incluido en esta sección. Esta materia se discute en la Sección 9.3.

## 7.4 Código de FEC: Rate 1/3

Una repetición 3-times simple que el código FEC usa para el header. La repetición del código es levada a cabo repitiendo cada bit tres times, vea la ilustración en la Figura 7.10. El 3-times código de repetición se usa para el header entero así como para el campo de los datos síncronos en el paquete HV1.

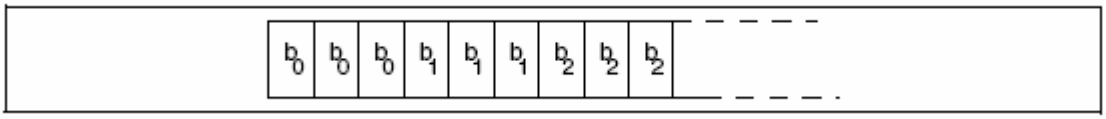



Figura 7.10. Esquema de codificación del bit-repetición.

## 7.5 Código de FEC: Rate 2/3

El otro esquema FEC es un (15,10) cortó código Hamming. El generador del polinomio es. $g(D) = (D+1)(D^4+D+1)$. Esto corresponde a 65 en notación octal. El LFSR que genera este código se ve en la Figura 7.11. Inicialmente todos los registros ponen los elementos a cero. Los 10 bits de información se alimentan secuencialmente en el LFSR con los interruptores S1 y S2 colocados en la posición 1. Entonces, después de la entrada final del bit, los interruptores S1 y S2 son fijos en posición 2, y los cinco bits de paridad se cambian afuera. Los bits de paridad se añaden a los bits de información. Como consecuencia, cada bloque de 10 bits de información se coloca en un código 15 codeword del bit. Este código puede corregir solo todos los errores y descubre todos los errores dobles en cada codeword. Estos 2/3 rate FEC es usado en los paquetes de DM, en el campo de los datos del paquete DV, en el paquete FHS, en el paquete HV2, y en el paquete EV4. Desde que el encoder opera con segmentos de información de longitud 10, se añadirán bits de la cola con un valor cero después del CRC los bits para traer el número total de bits igualan a un múltiplo de 10. El número de bits de la cola serán posibles añadirlos eso logra esto (es decir, en el intervalo 0...9). Éstos van detrás de los bits no son incluido en el indicador de longitud del payload para los paquetes de ACL o en el campo de longitud de payload del arreglo del eSCO del comando LMP.

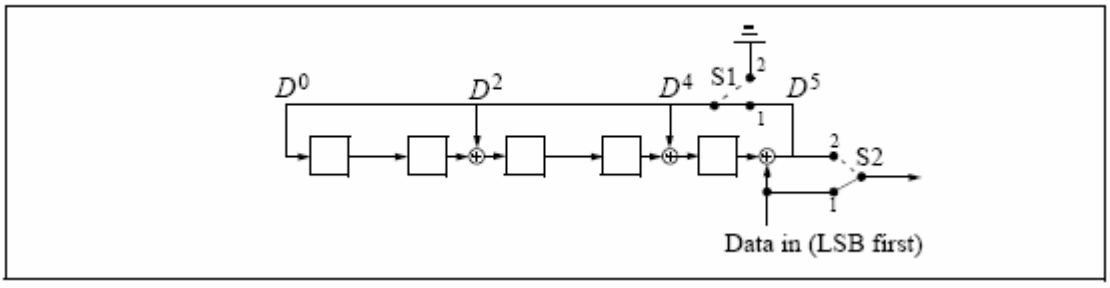



Figura 7.11: LFSR que genera (15,10) cortó código Hamming.

## 7.6 Esquema ARQ

Con un esquema automático y repetición de la demanda, DM, DH y el campo de los datos de paquetes DV, se transmitirán paquetes EV hasta que el **acknowledgement** de una recepción exitosa es devuelto por el destinatario (o la interrupción se exceda). La información del **acknowledgement** será incluida en el header del paquete del retorno. El esquema de ARQ sólo se usa en el payload del

paquete y sólo en paquetes que tienen un CRC. El header del paquete y los payload de los datos síncronos de HV y paquetes de DV no son protegidos por el esquema de ARQ.

### 7.6.1 Unnumbered ARQ

Bluetooth usa un rápido, esquema de **acknowledgement** unnumbered. Un ACK (ARQN=1) o un NAK (ARQN=0) devuelve previamente una contestación al recibimiento de un paquete recibido. El slave responderá en slot de slave-a-master directamente siguiendo el slot master-a-slave a menos que el slave tenga compromisos del scatternet en ese timeslot; el master responderá al próximo evento dirigiéndose al mismo slave (el master se puede haber dirigido a otros slaves entre el último paquete recibido considerando la contestación del slave y el master a este paquete) por lo menos para tener éxito, el HEC debe pasar por una recepción del paquete. Además, el CRC debe pasar si esta presente.

En el primer paquete POLL a la salida de una nueva conexión (como resultado de un page, page scan, role switch o unpark) el master inicializará el bit ARQN a NAK. El paquete de la contestación enviado por el slave también tendrá que el bit ARQN colocado a NAK. Los paquetes subsecuentes usarán las reglas siguientes. El valor inicial del enlace eSCO del master ARQN a la estructuración será NAK.

El bit ARQ sólo será afectado por paquetes de los datos que contienen CRC y vacío de slots. Como es mostrado en la Figura 7.12. En la recepción exitosa de un paquete CRC, el bit de ARQN se pondrá a ACK. Si, en cualquiera recibe slots en el trabajo como slave, o, en uno recibe slots en el master la transmisión siguiente de un paquete en, uno, de estos eventos aplica:

1. Ningún código de acceso es detectado
2. El HEC falla
3. El CRC falla

Entonces el bit de ARQN se colocar a NAK. En eSCO el bit de ARQN puede colocarse a ACK igualan cuando CRC en un paquete de EV le faltado o así do inhabilitado entrega paquetes erróneos.

Paquetes que tienen HEC correcto se dirigen a otros slaves, o otros paquetes DH, DM, DV o paquetes de EV, no afectarán el bit ARQN, excepto como se notó en la Sección 7.6.2.2 en la página 128. En estos casos el bit de ARQN estará saliendo como era anterior a la recepción del paquete. Para los paquetes de ACL, si un paquete de CRC con un header correcto el mismo SEQN tiene como el CRC previamente recibido el paquete, el bit de ARQN se pondrá a ACK y el payload se ignorará sin verificar el CRC. Para los paquetes del eSCO, el SEQN no usará Bitstream Processing al determinar el ARQN. Si un paquete del eSCO se ha recibido con éxito dentro de la ventana del eSCO las recepciones subsecuentes dentro de la ventana del eSCO se ignorarán. Al final la ventana del eSCO, el

ARQN del master estará reteniendo la primera transmisión del master-a-slave en la próxima ventana del eSCO. El bit ARQ en el paquete de FHS no es significante. Los contenidos del bit ARQN en el paquete de FHS no se verificarán. Transmitirá y se inspeccionarán errores de paquetes que usa CRC, pero ningún esquema ARQ se aplicará. Transmitirá y nunca habrá **acknowledgement** en los paquetes.

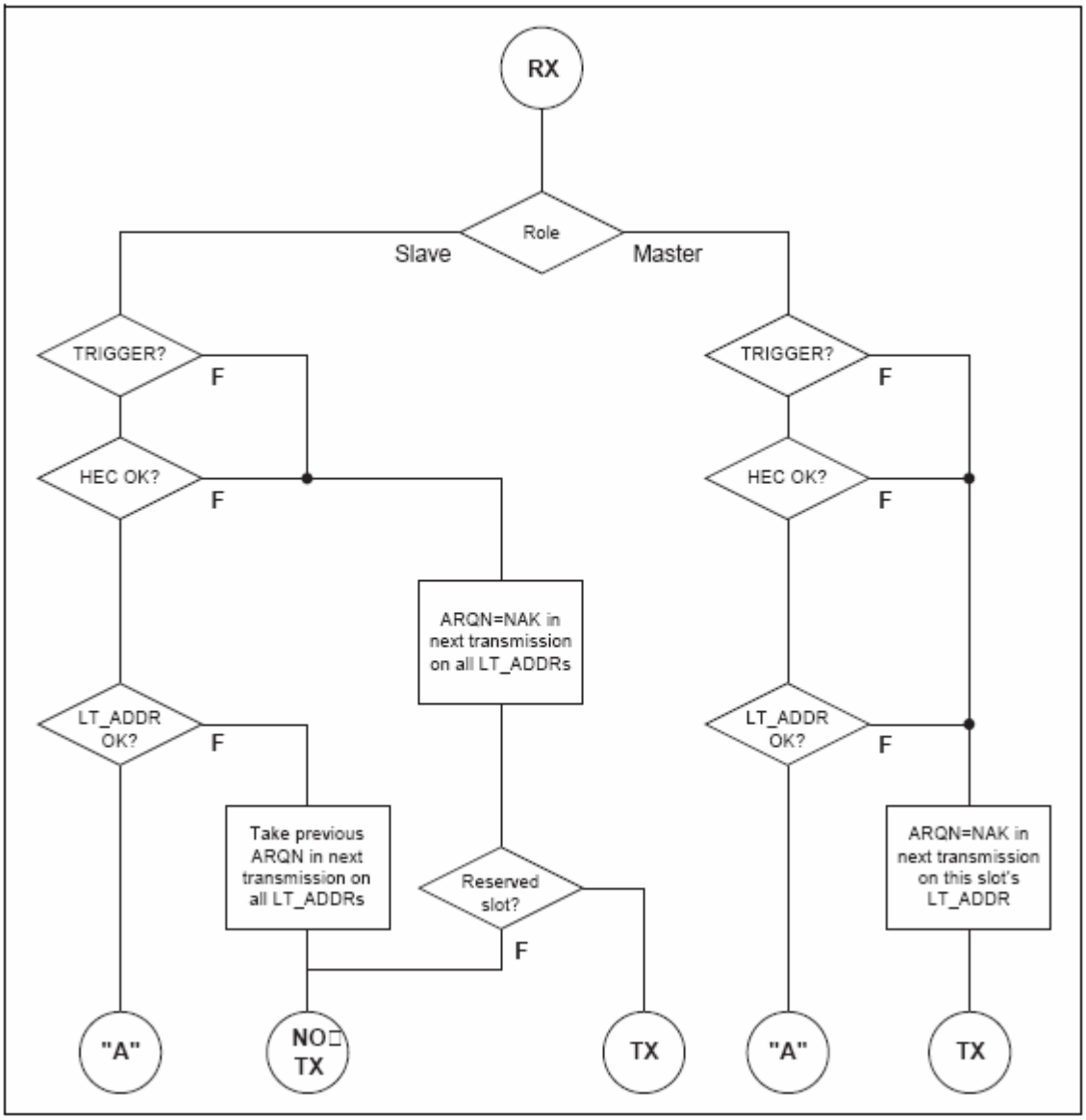



*Figura 7.12: Stage 1 of the receive protocol for determining the ARQN bit.*

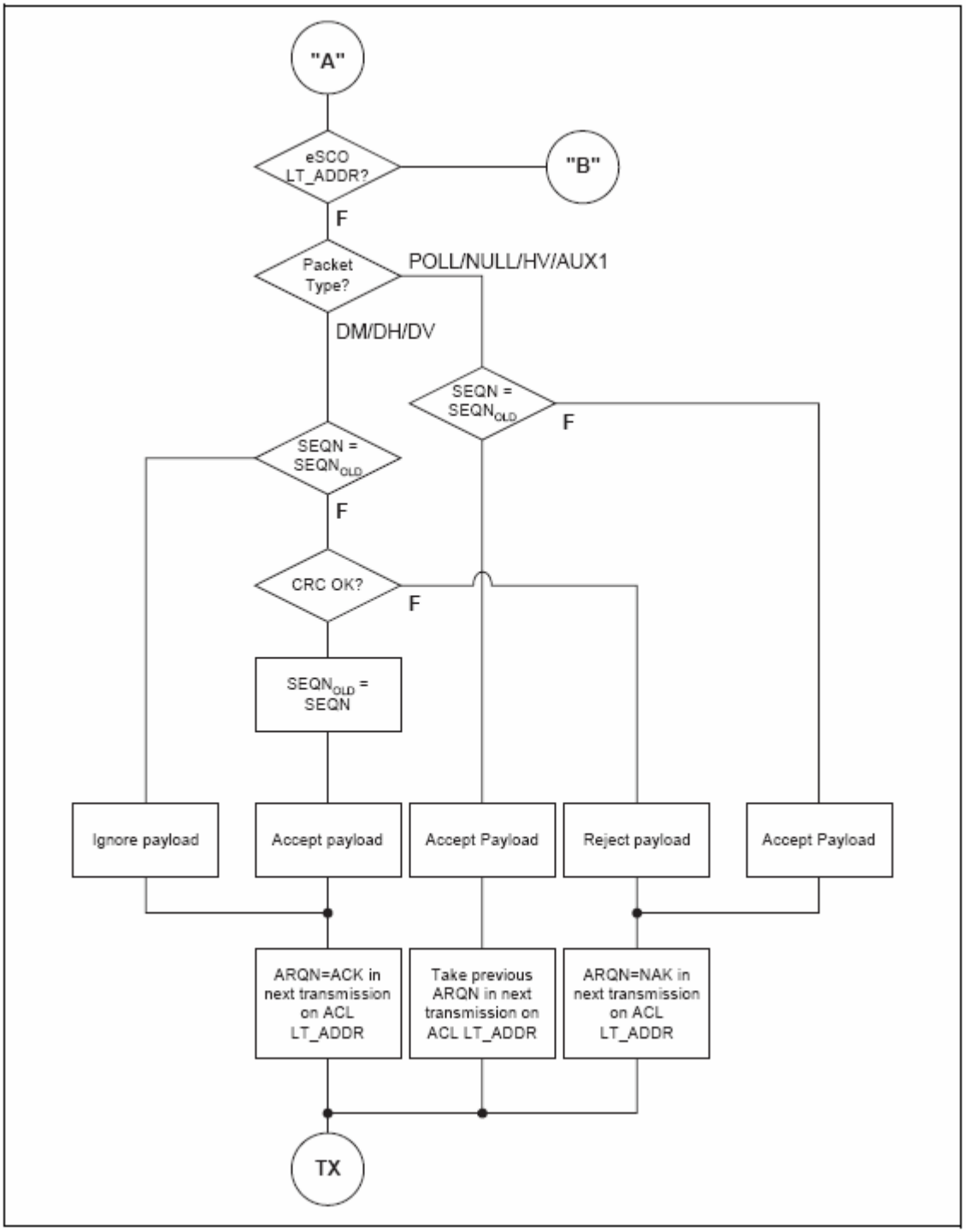



*Figura 7.13: Stage 2 (ACL) of the receive protocol for determining the ARQN bit.*

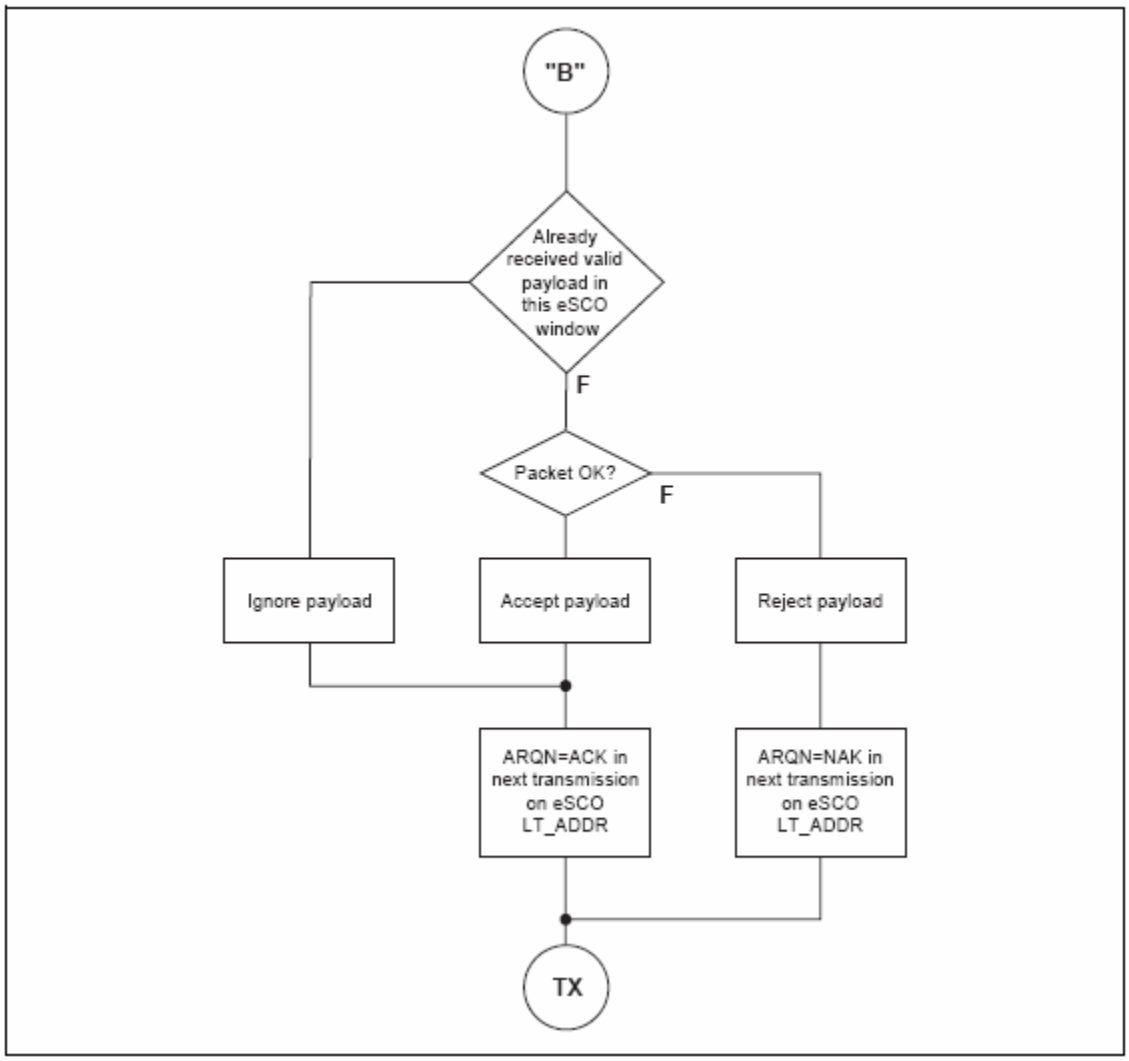



*Figura 7.14: Stage 2 (eSCO) of the receive protocol for determining the ARQN bit.*

## 7.6.2 Retransmisión Filtrada

Los payload de los datos se transmitirán hasta que un positivo **acknowledgement** es recibido o un timeout es exceded. Un retransmisión se llevará a cabo porque la propia transmisión del paquete falló, o porque el **acknowledgement** transmitido en el retorno del paquete del retorno ha fallado (nota que el último tiene un fracaso más bajo de probabilidad de que el header es más extensamente codificado). En el último caso, sigue recibiendo el mismo payload una y otra vez. Para filtrarse fuera de las retransmisiones en el destinatario, el bit de SEQN está presente en el header. Normalmente, este bit se alterna para cada nuevo CRC de datos de transmisión del payload. En caso de un retransmisión, este bit no se cambiará de destino así puede comparar que el bit SEQN con el valor de SEQN anterior. Si es diferente, el nuevo payload de los datos ha llegado; por otra parte el mismo payload de los datos puede ser ignorado. Se transferirán sólo nuevos payloads de

los datos al controlador de recursos Baseband note que CRC datos payloads sólo pueden ser llevados por DM, DH, DV o paquetes de EV.

### 7.6.2.1 inicialización SEQN a la Salida de una Nueva Conexión

El bit SEQN del primer CRC de los datos de paquete a la salida de una conexión (como un resultado de page, page scan, role switch o unpark) en ambos el master y el slave se colocaran en 1. Los paquetes subsecuentes usarán las reglas en las secciones siguientes.

### 7.6.2.2 ACL y SCO Retransmisión filtrada

El bit SEQN sólo será afectado por los datos de paquetes CRC como se muestra en la Figura 7.15. Se invertirá cada tiempo cuando se envía un nuevo paquete de datos CRC. El paquete de datos será retransmitido con el mismo numero SEQN hasta un ACK, es recibido o el paquete se vacía. Cuando un ACK se recibe, nuevos payload pueden ser enviados y en esa transmisión el bit de SEQN se invertirá. Si un dispositivo decide vaciar (vea Sección 7.6.3 en la página 130), y no ha recibido un **acknowledgement** para el paquete actual, reemplazará el paquete actual con un ACL-U continuo del paquete con el mismo número de la sucesión como la continuación del paquete y cero de longitud. Si se reemplaza el paquete actual de esta manera no debe seguir transmitiendo el próximo paquete hasta que haya recibido un ack.

Si el slave recibe otros paquetes mas DH, DM, DV o EV con el bit SEQN invertido este será recibido exitosamente después del header en alguna LT_ADDR, pondrá el bit ARQN a NAK hasta un DH, DM, DV o el paquete EV es con recibido exitosamente.

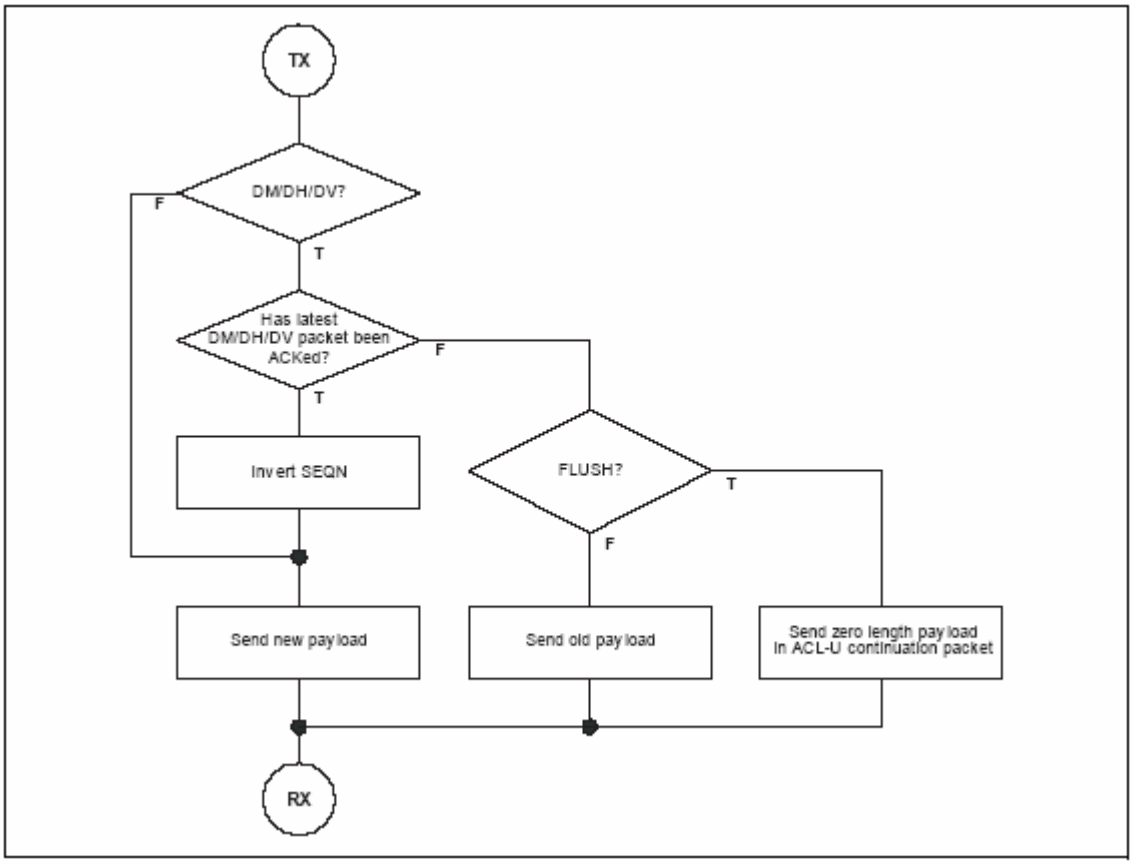



Figura 7.15: Transmisión filtrada para los paquetes con CRC.

### 7.6.2.3 Retransmisión Filtrada en eSCO

En eSCO, el bit de SEQN será toggled en cada ventana del eSCO. El valor deberá ser constante para la duración de la ventana del eSCO. El valor inicial de SEQN será cero. Para una ventana del eSCO dada el valor de SEQN será constante.

### 7.6.2.4 Retransmisión Filtrada FHS

El bit SEQN en el paquete FHS no es significante. Este bit puede colocarse en cualquier valor. Los contenidos del bit SEQN en el paquete de FHS no se verificarán.

### 7.6.2.5 Retransmisión Filtrada en Paquetes sin el CRC

Durante la transmisión de paquetes sin un CRC el bit SEQN permanecerá igual que como estaba en el paquete anterior.

### 7.6.3 Payloads Vacíos Flushing

En ACL, el esquema de ARQ puede causar retraso inconstante subsecuentemente en el flujo de tráfico se insertan retransmisiones para asegurar el traslado de los datos error-libre. Con toda seguridad al enlace de comunicación, sólo una cantidad limitada de retraso se permite. Retransmisiones a un cierto límite al que los payload actuales se ignorarán.
Este traslado de los datos se indica como tráfico isócrono. Esto significa que el proceso de retransmisión debe ser de orden continuo para con los próximos datos del payload. Abortando el esquema de retransmisiones cumplido por flushing el dato anterior esta obligando al control de enlace a tomar los próximos datos a cambio.

Resultados flushing en pérdida de porciones restantes de un mensaje L2CAP.Por consiguiente, el paquete que sigue el flush tendrá una indicación de paquete de salida de LLID = 10 en el payload header para el próximo mensaje L2CAP. Esto informa el destino del flush. (Vea Sección 6.6). Flushing necesariamente no resulta un cambio en el valor del bit SEQN , vea la sección anterior.

La Interrupción de flush define el periodo máximo después de que todos los segmentos del paquete ACL-U se vacía del controlador del buffer. La Interrupción de flush se debe a la salida del Primer segmento del paquete ACL-U se guarda en controlador del buffer Después de que la interrupción de flush ha expirado el control de enlace puede continuar con las transmisiones del procedimiento descrito en la Sección 7.6.2.2 en la página 128, sin embargo el controlador de recursos de baseband no continuará la transmisión del paquete de ACL-U al control de enlace. Si el controlador de recursos de baseband tiene segmentos extensos del paquete están en la cola para la transmisión el controlador anulará los segmentos restantes del paquete ACL-U de la cola. En caso de que el paquete completo de ACL-U no se guardó todavía en el controlador del buffer, cualquier continuación segmentada, recibida por el transporte lógico ACL se flushed vaciará, hasta que un primer segmento se reciba. Cuando el paquete ACL-U se ha flushed vaciado completamente el control de enlace continuará la transmisión luego del paquete ACL-U para trasporte lógico ACL. La Interrupción de flush predefinida será infinito, es decir las re-transmisiones se llevan a cabo hasta la pérdida del enlace físico cuando esto ocurre es llamado un “canal fiable” Todos los dispositivos apoyarán la Interrupción de flush predefinida. En eSCO, se flushed vaciarán paquetes automáticamente al final de la ventana del eSCO.

### 7.6.4 Consideraciones del Multi-Slave

En un piconet con transportes lógicos múltiples, el master llevará a cabo el protocolo ARQ Independientemente en cada transporte lógico.

## 7.6.5 Paquetes de la Transmisión

Paquetes de transmisión son paquetes transmitidos por el master a todos los slaves simultáneamente. Si están usándose secuencias del múltiples hop cada transmisión sólo puede ser recibido por algunos de los slaves. En este caso el master debe repetir la transmisión en cada sucesión del hop. Un paquete de la transmisión será Bitstream Processing. Indicado por todos los ceros LT_ADDR (nota; el paquete FHS es el único paquete qué puede tener todos los ceros LT_ADDR pero no puede ser un paquete de la transmisión). En la Transmisión no se reconocerán paquetes (por lo menos no al nivel de LC).

Puesto que no se reconocen mensajes de la transmisión, cada paquete de la transmisión es transmitido a un número fijo de tiempos por lo menos. Un paquete de la transmisión debe ser $N_{BC,}$ el tiempo antes de la próxima transmisión del paquete es igual a la transmisión del mensaje que se transmite, vea la Figura 7.16. Opcionalmente, una transmisión del paquete puede transmitirse $N_{BC}$ + 1 veces. Nota: $N_{BC}$=1 medios cada uno de los paquetes de la transmisión sólo debe enviarse una vez, pero opcionalmente puede enviarse dos veces. Sin embargo, la información de la transmisión tiempo-crítico puede abortar la transmisión continua de tren

Por ejemplo, mensajes del unpark enviados a los casos de beacon pueden hacer esto, vea la Sección 8.9.5. Si están usándose secuencia del hop múltiples, el master puede transmitir adelante del secuencias hop diferentes en cualquier orden, proporcionando esa transmisión de nuevo el paquete de la transmisión no se empezará hasta que todas las transmisiones de cualquier paquete anterior se ha completado en todas las secuencias del hop.

La transmisión de un solo paquete de la transmisión puede entrelazarse entre las secuencias del hop para minimizar el tiempo total para transmitir un paquete. El master tiene la opción de transmitir sólo $N_{BC}$ cronometra en canales comunes a todas las secuencias del hop.

Transmitir paquetes con un CRC tendrá su propio número de secuencia. El SEQN del primer paquete de la transmisión con un CRC se pondrá a SEQN = 1 por el master y se invertirá después de esto para cada nuevo paquete de la transmisión con CRC. Transmitirá paquetes sin un CRC no tiene influencia en el número de la secuencia. El slave aceptará el SEQN del primer paquete de la transmisión en el que se recibe una conexión y verificará para el cambio en SEQN para la transmisión subsiguiente de paquetes. No hay ningún reconocimiento de mensajes de la transmisión subsecuentemente y ninguna indicación de fin de paquete, es importante recibir los paquetes de la salida correctamente. Asegurar esto, repeticiones de los paquetes de la transmisión que son paquetes de salida L2CAP y los paquetes de LMP no se filtrarán fuera. Estos paquetes se indicarán por LLID=1X en el payload header como se explicó en la sección 6.6.

Sólo repeticiones de los paquetes de continuación L2CAP se filtrarán fuera.

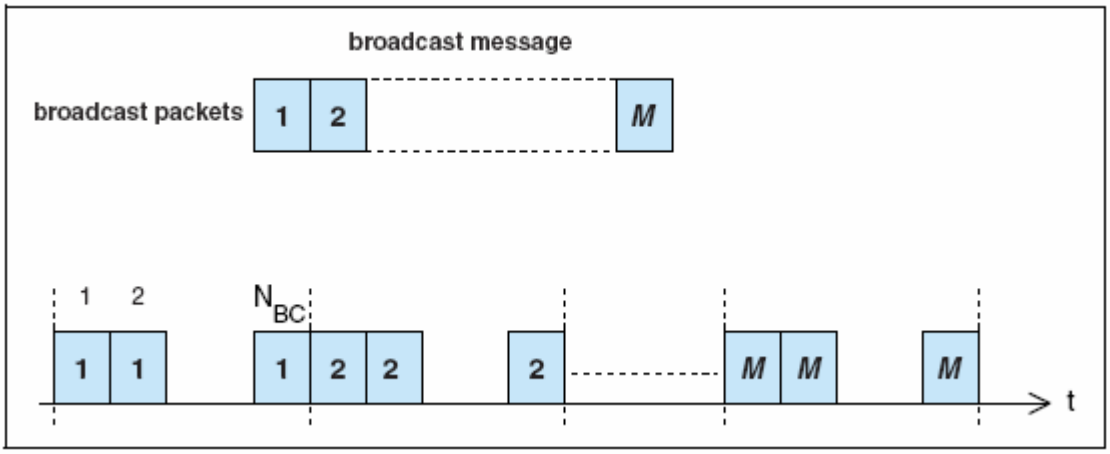



Figura 7.16: Esquema de repetición de Transmisión

# 8 Operación del Controlador de Enlace

Esta sección describe cómo un piconet se establece y cómo los dispositivos pueden ser agregados y fuera del piconet. Varios estados definen el funcionamiento dispositivos para apoyar estas funciones. Además, el funcionamiento de varios piconets con uno o los miembros más comunes, el scatternet, se discute.

## 8.1 Apreciación Global de Estados

Figura 8.1 muestra un diagrama que ilustra los diferentes estados usados en el controlador de enlace. Hay tres estados del controlador. STANDBY, CONNECTION, y PARK además hay siete subestados, page, page scan, inquiry inquirí response master response, slave response, e inquiry response. Los subestados son estados interinos que se usan para establecer conexiones y habilitan el dispositivo discovery. Para mover de un estado o a otro subestado, se usan cualquier orden del control de enlace, o las señales dentro del control de enlace se usan (como la señalización trigger del correlator y la interrupción de señal).

.

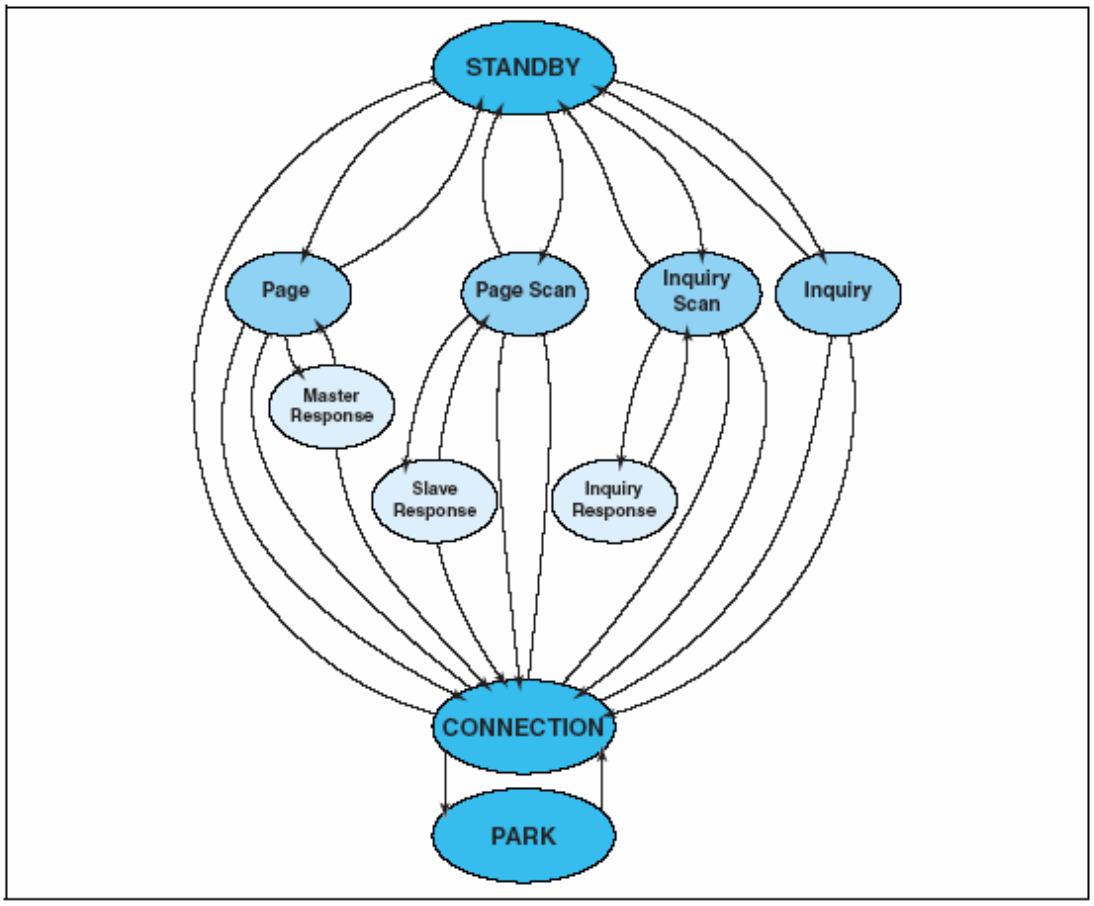



Figura 8.1: El diagrama de estados del control de enlace.

## 8.2 Estado STANDBY

El estado STANDBY es el estado predefinido en el dispositivo. En este estado, el dispositivo puede estar en un modo de bajo-poder. Sólo el reloj nativo está corriendo a la exactitud del LPO (o mejor).
El controlador puede dejar el estado STANDBY para page o mensajes inquiry o para page e inquiry itself.

## 8.3 Establecimiento de Conexión de Subestados

Para establecer nuevas conexiones en el procedimiento paging se usa. Sólo el dispositivo Bluetooth de dirección se exige preparar una conexión. **Acknowledgement** sobre el reloj, obteniendo el procedimiento inquiry (vea la Sección 8.4) o de una conexión anterior con este dispositivo, el modo page scan de el otro dispositivo acelerará el procedimiento del arreglo. Un dispositivo que establece una conexión lleva a cabo un procedimiento de page y se volverá automáticamente el master de la conexión.

### 8.3.1 Subestado Page Scan

En el subestado page scan, es un dispositivo que puede configurarse para usar la norma de interlaced procedimiento scanning. Durante la norma scan, el dispositivo listen determina la duración del scan window $T_{w\_page\_scan}$ (11.25ms default, vea HCI [la Parte E] Sección 7.3.2), mientras el interlaced scan se realiza como dos de espaldas examina de Tw_page_scan. Si el intervalo scan no es menor dos veces el scan window, entonces interlaced scanning no se usará. Durante cada scan window, el dispositivo escuchará a una sola frecuencia del hop, su correlator, emparejando a su código de acceso de dispositivo (DAC). El scan window será bastante largo para examinar 16 frecuencias de page completamente.

Cuando un dispositivo entra en subestado page scan, seleccionará la frecuencia scan según la secuencia de hopping de page, determinada por el dispositivo Bluetooth y dispositivo address vea la Sección 2.6.4.1 en la página 79. La fase en el la secuencia será determinada por $CLKN_{16-12}$ del reloj nativo del dispositivo; eso es, cada 1.28s una frecuencia diferente es seleccionada.

En el caso de una norma scan, si el correlator excede el threshold (umbral) del trigger durante page scan, el dispositivo entrará en el subestado de slave response descrito en Sección 8.3.3.1 en la página 140 El dispositivo scan también puede usar interlaced scan En este caso, si el correlator no excede el trigger el umbral durante el primero scan que un segundo tiempo scan es usando la fase en la secuencia determinada por $[CLKN_{16-12} + 16]$ mod 32. Si en este segundo scan el correlator excede el umbral del tigger el dispositivo entrará en el subestado slave response usa $[CLKN_{16-12} + 16]$ mod 32 como el frozen CLKN * en el cálculo para Xprs ([79]), vea Sección 2.6.4.3 en la página 80 para detalles. Si el correlator no excede el umbral del trigger durante un modo scan normal o estado STANDBY o estado de CONNECTION. Page scan pueden entrar en subestado del estado STANDBY y el estado CONNECTION. En el estado STANDBY una conexión se ha establecido y el dispositivo puede usar toda la capacidad de llevar a cabo page scan. Antes de entrar en el subestado page scan de la CONEXION el dispositivo debe reservar tanta capacidad como sea posible para scanning.

Si el dispositivo puede colocar conexiones de ACL en HOLD, PARK o SNIFF, vea Sección 8.8 adelante la página 165 y Sección 8.9 en página la165. Las conexiones síncronas no deben ser interrumpidas por page scan, aunque las retransmisiones del eSCO deben hacer una pausa durante scan. Page scan puede interrumpirse por el reservado de slots síncronos que deberán tener prioridad más alta que page scan. Deben usarse paquetes SCO requiriendo la menor cantidad de capacidad (paquetes HV3). El scan window se aumentará para minimizar el retraso del arreglo. Si un el transporte lógico SCO es presente usando paquetes HV3 y $T_{SCO}$=6 slots o en el transporte lógico eSCO lógico es presente usando paquetes EV3 y $T_{ESCO}$=6 slots, en total scan window $T_{w\ page\ scan}$ de por lo menos 36 slots (22.5ms) se recomienda; si dos enlaces SCO son presentes usando paquetes

HV3 y $T_{SCO}$=6 slots o dos enlaces del eSCO son presentes usando paquetes EV3 paquetes y $T_{ESCO}$=6 slots, en total scan window por lo menos de los 54 slots (33.75ms) se recomienda. Examine intervalo $T_{page\ scan}$ se define como el intervalo entre los principios de dos page scan consecutivas. Una distinción se hace entre el caso donde el intervalo scan es igual al scan window $T_{w\ page\ scan}$ (scan constante), el intervalo scanning es máximo 1.28s, o el intervalo scanning es máximo 2.56s. Estos tres casos determinarán la conducta del dispositivo de paging; es decir, si el dispositivo paging usará R0, R1 o R2, también vea Sección 8.3.2 en la página 136. Tabla 8.1 ilustra la relación entre $T_{page\ scan}$ y modos R0, R1 y R2. Aunque en scanning R0 el modo es constante, scanning pueden ser interrumpido, por ejemplo por slots síncronos reservados. La información del intervalo scan es incluido en el campo de SR en el paquete FHS.

| SR mode | $T_{page\ scan}$ |
|---|---|
| R0 | ≤ 1.28s and = $T_{w\ page\ scan}$ |
| R1 | ≤ 1.28s |
| R2 | ≤ 2.56s |
| Reserved | - |

Tabla 8.1: la Relación entre intervalo scan, y modos paging R0, R1 y R2.

## 8.3.2 Subestado Page

El substate page es usado por el master (fuente) para activar y conectar a un slave (destino) en el subastado page scan. El master intenta coincidir con el slave scan activando por transmisión repetida el mensaje paging consiste del dispositivo del código de acceso del slave (DAC) en diferentes canales hop.
Desde que los relojes Bluetooth en el master y el slave no se sincronizan, el master no sabe exactamente cuando el slave se wake-up en la frecuencia hop. Por consiguiente, transmite un tren de mensajes page scan. En diferentes frecuencias hop y listen entre los intervalos transmite hasta que recibe respuesta del slave.

El procedimiento page en el master consiste en varios pasos. Primero, el host comunica BD_ADDR del slave al controlador. Este BD_ADDR será usado por el master para determinar la secuencia hopping de page, vea Sección 2.6.4.2. El BD_ADDR del slave será usado para determinar secuencia hopping de page, vea Sección 2.6.4.2. Para la fase en la secuencia, el master usará una estimación del reloj del slave. Por ejemplo, esta estimación puede derivarse del timing de información que se intercambió durante el último enlace con este dispositivo particular (qué podría actuar como master todo el tiempo), o de un procedimiento inquiry con esta estimación CLKE de el reloj Bluetooth del slave, el master puede predecir en el canal hop la salida del slave en page scanning.

La estimación del reloj de Bluetooth en el slave puede estar completamente equivocada. Aunque el master y el slave usan la misma secuencia hopping, ellos usan fases diferentes en la secuencia y nunca podría seleccionar la misma frecuencia.

Para compensar las tendencias del reloj, el master enviará su mensaje de page durante un intervalo de tiempo corto en varios wake-up frecuencias. Transmitirá también en frecuencias del hop sólo antes y después de la actual, la frecuencia hop durante cada time slot de TX, el master transmitirá secuencialmente en dos diferentes frecuencias hop en el time slot de RX siguiente, el receptor listen secuencialmente a dos RX correspondientes hops por el paquete ID. Los hops de RX serán seleccionados según la secuencia hopping de page response.
La secuencia hopping de page response se relaciona estrictamente a la secuencia hopping de page para cada page hop es correspondiente a cada hop page response. El RX/TX timing en el subestado de la página se describe en la Sección 2.2.5 en la página 60, también vea la figura 2.7 en la página 65. En el próximo slot de TX, transmitirá en dos frecuencias hop diferentes del anterior. Nota: El incremento del hop rate 3200 hops/s. Con el incremento del hop rate como se describió anteriormente, el transmisor puede cubrir 16 frecuencias hop diferentes en 16 slots o 10 ms. La secuencia hopping de page es dividida en fuera de dos trenes de paging A y B de 16 frecuencias. El tren A incluye las 16 frecuencias hop que rodean el current, f (k) frecuencia de hop mencionada, donde k es determinada por el reloj estimación $CLKE_{16-12}$. El primer tren consiste en hops.

F (k-8), f (k-7),..., f (k),..., f (k+7)

Cuando la diferencia entre los relojes Bluetooth del master y el slave están entre -8x1.28 s y +7x1.28 s, una de las frecuencias usadas por el master, será la frecuencia hop a la que el slave listen. Desde que el master no sabe cuando el slave entrará en el subestado page scan, el master tiende a repetir este tren A $N_{page}$ times o hasta que se obtenga una respuesta, quienquiera es más corto. Si el intervalo scan del slave corresponde a R1, el número de la repetición es menor a128; si el intervalo scan del slave corresponde a R2 o si el master no tiene previamente leído el modo de SR del slave, el número de la repetición es por lo menos 256. Si el amo no ha leído el modo de SR del slave previamente usará $N_{page}$ >= 256. Note que $CLKE_{16-12}$ cambia cada 1.28 s; por consiguiente, cada 1.28 s, los trenes incluirán frecuencias diferentes en el set page hopping.

Cuando la diferencia entre los relojes Bluetooth del master y el slave es menor de -8x1.28 s o mayor que +7x1.28 s, siguen siendo 16 hops los que se usan para formar el nuevo tren 10 ms B. El segundo tren consiste en hops (k-16), f (k-15),..., f (k-9), f (k+8)..., f (k+15)

El Tren B se repetirá durante tiempos de $N_{page}$. Si ninguna respuesta se obtiene, el tren A se probará en tiempos Npage nuevamente. Usando alternadamente el tren A y el tren B será constante hasta que una respuesta se reciba o pageTO en

la interrupción se excederá. Si una respuesta es devuelta por el slave, el dispositivo del master entrara en el subestado response

Los subestados page pueden entrar desde el estado STANDBY o CONNECTION, en el estado STANDBY, ninguna conexión se ha establecido y el dispositivo puede usar toda la capacidad para llevar a cabo page. Antes de entrar al subestado paging desde el estado CONNECTION, el dispositivo debe librar tanta capacidad como sea posible para scanning. Para asegurar esto, se recomienda que las conexiones de ACL estén en HOLD o PARK. Sin embargo, las conexiones síncronas no serán perturbadas por page. Esto significa que page se interrumpirá por los SCO reservados y slots eSCO que tienen prioridad más alta que page. Para obtener capacidad para paging, se recomienda usar los paquetes SCO que usan la menor cantidad de capacidad de los (paquetes HV3). Si los enlaces SCO y eSCO están presentes, el número repeticiones $N_{page}$ de un solo tren aumentará, vea tabla 8.2. Aquí se asume que los paquetes HV3 son usados con un intervalo $T_{SCO}$=6 slots o paquetes EV3 se usan con un intervalo de $T_{ESCO}$=6 slots que corresponden a un enlace sincrono de64 kb/s.

.
La construcción del tren de page será independiente de la presencia de los enlaces síncronos es decir, se envían paquetes síncronos en los slots reservados pero no afecta las frecuencias hop usadas en los slots no reservados, vea Figura 8.2 en la página 138.

| SR mode | no synchronous link | one synchronous link (HV3) | two synchronous links (HV3) |
|---|---|---|---|
| R0 | $N_{page} \geq 1$ | $N_{page} \geq 2$ | $N_{page} \geq 3$ |
| R1 | $N_{page} \geq 128$ | $N_{page} \geq 256$ | $N_{page} \geq 384$ |
| R2 | $N_{page} \geq 256$ | $N_{page} \geq 512$ | $N_{page} \geq 768$ |

Figura 8.2: lL página Convencional (un), página mientras un presente del eslabón síncrono (b), página mientras dos presente de los eslabones síncrono (c).

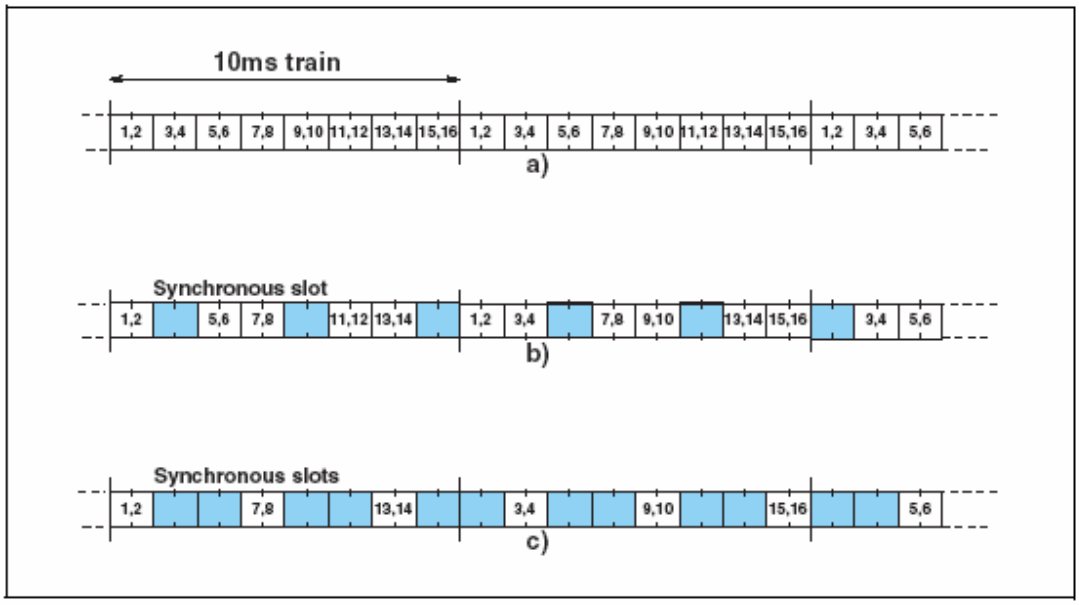



Tabla 8.2 La Relación entre la repetición del tren, y compaginando modos R0, R1 y R2 cuando los eslabones síncronos están presentes.

## 8.3.3 Subestado Page Response

Cuando un mensaje page es recibido con éxito por el slave, hay una Sincronización de FH entre el master y el slave. El master y el slave entra en un subestado de response para intercambiar información vital para continuar el arreglo de conexión. Es importante para la conexión del piconet que ambos dispositivos usen el mismo código de acceso de canal, usen el mismo canal de secuencia hop, y sus relojes estén sincronizados. Estos parámetros serán derivados del dispositivo del master.
El dispositivo que inicializa la conexión (salida paging) se define como el dispositivo master (qué sólo es válido durante el tiempo que el piconet existe). El código de acceso de canal y canal de secuencia hop se derivará del dispositivo de dirección Bluetooth (BD_ADDR) del master. El timing será determinado por el reloj del master. Un desplazamiento debe ser agregado al reloj nativo del slave para sincronizar el reloj del slave temporalmente al reloj del master. Los parámetros start-up se transmiten desde master al slave. El mensaje entre el master y el slave al start-up se especifica en esta sección.

El mensaje inicial entre el master y el slave se muestra en la tabla 8.3 en la página 139 y en Figura 8.3 en la página 140 y Figura 8.4 en la página 140. En las dos frecuencias de las figuras f (k), f(k+1), etc. son las frecuencias para la secuencia page hopping determinada por el BD_ADDR del slave. Las frecuencias f'(k de frecuencias), f'(k+1), etc. Son las frecuencias del page_response correspondientes (slave-a-master). Las frecuencias g (m) pertenecen al canal básico de secuencia hopping.

| Step | Message | Packet Type | Direction | Hopping Sequence | Access Code and Clock |
|---|---|---|---|---|---|
| 1 | Page | ID | Master to slave | Page | Slave |
| 2 | First slave page response | ID | Slave to master | Page response | Slave |
| 3 | Master page response | FHS | Master to slave | Page | Slave |
| 4 | Second slave page response | ID | Slave to master | Page response | Slave |
| 5 | 1st packet master | POLL | Master to slave | Channel | Master |
| 6 | 1st packet slave | Any type | Slave to master | Channel | Master |

Tabla 8.3: El messaging Inicial durante salida-a.

En el paso 1 (ver tabla 8.3 en página 139), el dispositivo del master está en el subestado page y el dispositivo del slave en el subastado page scan, en este paso se asume que el mensaje page es enviado por el master y el slave. Al recibir el mensaje page, el slave entra en slave response en el paso 2. El master espera para una respuesta del slave y cuando este llega al paso 2, entrará master response en el paso 3. Nota eso es durante el intercambio del mensaje inicial, todo los parámetros se derivan del dispositivo address del slave, y que sólo el page hopping y la secuencia page response son usados (esto también se deriva del dispositivo address del slave). Nota cuando el master y slave entran en los estados de response, su entrada al reloj page y secuencia page response estado frozen esto se describe en la Sección 2.6.4.3.

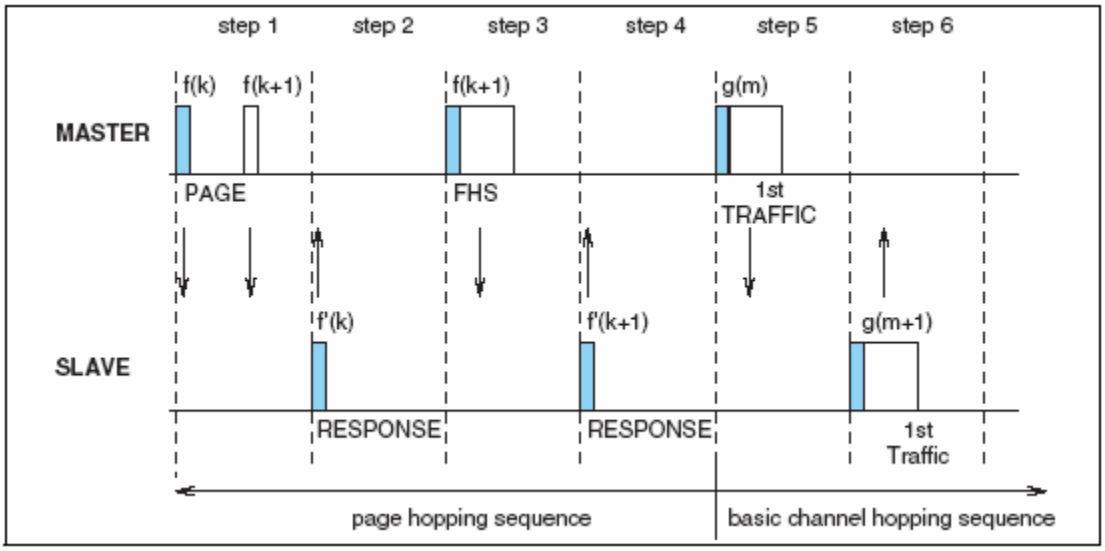



Figura 8.3: Messaging a la conexión inicial cuando el slave responde para primer mensaje paging.

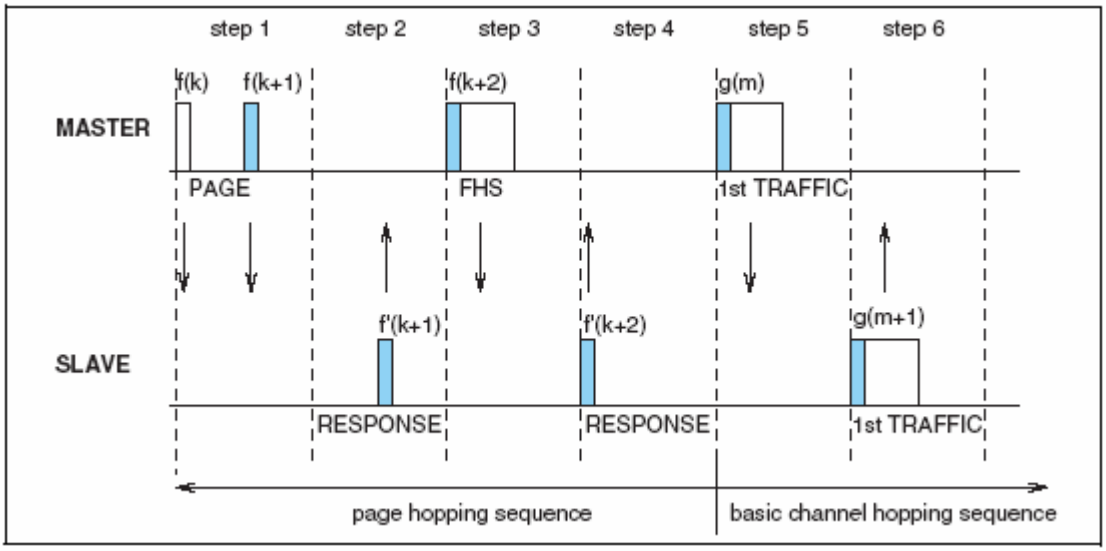



Figura 8.4: Messaging a la conexión inicial cuando el slave responde para secundar mensaje de page.

### 8.3.3.1 Subestado Slave Response

Después de haber recibido el mensaje de page en el paso 1, el dispositivo del slave transmitirá un mensaje page response, (el código de acceso del dispositivo del slave) en el paso 2. Este mensaje response será el código de acceso del dispositivo del slave. El slave transmita este response a 625 µs después del inicio del mensaje page response y la frecuencia hop response que corresponde a la frecuencia hop en el que el mensaje page fue recibido. La transmisión del slave es por consiguiente tiempo alineado a la transmisión del master. Durante el mensaje inicial, el slave todavía usará la secuencia page response para devolver información al master. El reloj de entrada CLKN16-12 será frozen al valor al que tenía al tiempo que el mensaje page fue recibido. Después de haber enviado el mensaje response, el receptor del slave se activará 312.5 µs después de la salida del mensaje response y esperará la llegada de un paquete FHS. Nota un paquete FHS puede llegar 312.5 µs después de la llegada de el mensaje page mostrado en la Figura 8.4, y no después de 625 µs como normalmente es el caso en el piconet canal físico. El RX/TX timing. Más detalles pueden encontrarse en Sección 2.4.4.

Si el arreglo falla antes que se alcance el estado de CONEXION, el siguiente procedimiento se llevará a cabo. El slave atenderá mientras que ningún paquete FHS se recibe hasta que el pagerespond TO se excede. Cada 1.25 ms, sin embargo, este seleccionara la próxima frecuencia hop master-a-slave según la secuencia hop page. Si nada se recibe después del pagerespTO, el slave regresa al subestado page scan para un periodo scan. La longitud de un periodo scan depende del slot reservado síncrono presente. Si ningún mensaje page es recibido

durante este periodo scan adicional, el slave reasumirá scanning en su intervalo regular scan y volverá al estado que seria prioridad al primer estado page scan.

Si un paquete FHS es recibido por el slave en el subestado slave response, el slave regresara un mensaje slave page response en el paso 4 la recepción acknowledge del paquete FHS. Esta respuesta usará la secuencia page response hopping. La transmisión del paquete slave page response basado en la recepción del paquete FHS. Entonces el slave cambiará al canal master el código de acceso y el reloj como recibió del paquete FHS. Sólo se transfieren 26 MSBs del reloj del master: el timing serán tal que $CLK_1$ y $CLK_0$ ambos están en cero en el momento que el paquete FHS se recibió como la transmisión master en solo los slot pares. El offset entre el reloj del master y el reloj del slave se determinará del reloj del master en el paquete FHS y reportado al del slave.

Finalmente, el slave entra en el estado de CONEXION paso 5. Desde este momento, el slave usara el reloj del master y el BD_ADDR del master determinara el canal básico de secuencia hopping y el código de acceso del canal. El slave debe usar el LT_ADDR en el payload de FHS como el LT_ADDR primario en el estado de CONEXION. El modo de conexión empezará con un paquete POLL transmitido por el master. El slave puede responder con cualquier tipo de paquete.

Si la el paquete POLL no es recibido por el slave, o el paquete response no se recibe por el master dentro de la nueva conexión TO el numero de slots después del reconocimiento del paquete FHS el master y el slave regresaran a los subastados page y page scan , respectivamente. Vea Sección 8.5.

### 8.3.3.2 Subestado Master Response

Cuando el master ha recibido un mensaje page response en el paso 2, él, entra en la rutina master response. Congelara la entrada del reloj actual a la selección del esquema page hop. El master transmitirá un paquete FHS entonces en el paso 3 que contiene el tiempo-real del reloj del master Bluetooth, el BD_ADDR master, los BCH bit de paridad, y la clase de dispositivos. El paquete FHS contiene toda la información para construir el código de acceso del canal sin requerir una derivación matemática de Bluetooth master del dispositivo address. El campo LT_ADDR en el paquete header del paquetes FHS en el subestado master response se pondrán en all-zero. El paquete FHS se transmitirá al comienzo del slot master-a-slave que sigue al slot de respuesta.
El paquete FHS llevará los all-zeros LT_ADDR. El timing TX del FHS no esta basado en la recepción del paquete response del slave. El paquete FHS puede enviarse 312.5 µs por consiguiente después de la recepción del paquete response como se muestra en la Figura 8.4 página 140 y no 625 µs después del paquete response como es usual en el piconet del canal físico RX/TX timing, vea también Sección 2.4.4.

Después de que master ha enviado su paquete FHS, esperará por una segunda mensaje slave page response en el paso 4 que reconoce la recepción del paquete FHS. Esta respuesta será el código de acceso del dispositivo slave. Si ninguna respuesta es recibida, el master debe retransmitir el paquete FHS con un reloj actualizado y todavía usando los parámetros del slave. Debe retransmitir el paquete FHS con el reloj actualizado hasta un segundo mensaje slave page response recibido, o la interrupción de pageRespTO excedido. En el último caso, el master regresar al subastado page y enviará un mensaje del error al Baseband Resource Manager. Durante la retransmisiones del paquete FHS, el master usará la secuencia page hopping.

Si el slave response es recibido, el master cambiará a usar los parámetros master así que usará el código de acceso de canal y el reloj del master. Los bits del reloj más bajos $CLK_0$ y $CLK_1$ se restablecerán para poner a cero a la salida la transmisión del paquete FHS que no es incluido en el paquete FHS. Finalmente, el master entra en el estado de CONEXION en el paso 5. El master BD_ADDR se usará para cambiar a una nueva secuencia hopping, el canal básico de secuencia hopping.

El canal básico de secuencia hopping usa todos los 79 canales hop en una forma pseudorandom también vea Sección 2.6.4.7 en página 82. El master deberá enviar su primer paquete de tráfico en un hop determinado con los nuevos (master) parámetros. Este primer paquete será un paquete POLL. Vea Sección 8.5 en página 148. Este paquete se enviará dentro una nueva conexión TO número de slots después de la recepción del FHS paquete reconocimiento. El slave puede responder con cualquier tipo de paquete. Si el paquete PULL no es recibido por el slave o la VOTACION la contestación del paquete no es recibida por el amo dentro del número de la nueva CONEXION de hendeduras, el amo y el slave volverán para compaginar y la página examina substates, respectivamente.

# PARTE C
# LINK MANAGER PROTOCOL (LMP)

El Link Manager Protocol (Protocolo de Director de Enlace) lleva acabo el arreglo de la conexión, la autenticación, configuración de conexión y otros protocolos. Descubre otros LM´s remotos y se comunica con ellos vía Link Manager Protocol (LMP). Para realizar su papel de proveedor de este servicio, el LM utiliza los servicios fundamentales del Link Controller (LC).

El Protocolo del Director de Enlace consiste esencialmente en varios Protocol Data Units PDU (Unidades de Datos del protocolo), que se envían de un dispositivo a otro, determinado por el AM_ADDR en el header del paquete. LM PDUs siempre se envían como paquetes de slot-simple y el header de payload son por lo tanto de un byte.

Los paquetes DM1 se utilizan para transportar LM PDUs excepto cuando una conexión SCO se presenta utilizando paquetes HV1 y la longitud del contenido es menor de 9 byte. En este caso los paquetes DV son utilizados.

## 1 Sincronización

Los mensajes de LMP son llevados en ACL-C logical link, el cual no garantiza un tiempo de entrega ni de reconocer paquetes (acknowledge packets). Los procedimientos de LMP toman en cuenta esto al sincronizar los cambios del estado en los dos dispositivos. Por ejemplo, los criterios que definen esto especifican cuando una dirección lógica del transporte (LT_ADDR) puede volverse a emplear después que llega a ser disponible debido a un dispositivo que sale del piconet o entrar en el estado park. Otros procedimientos de LMP, tal como hold o role switch incluye el reloj Bluetooth como un parámetro para definir un punto fijo de sincronización. Las transiciones en y fuera del modo sniff son protegidos con un modo de transición.

## 2 LMP PDUs

En la siguiente lista, están los tipos de PDU disponibles, su función y operación. Estos PDU´s son: Obligatorios M (deben ser sostenido), u Opcional O (opcionalmente sostenido)

## 2.1 Respuesta General

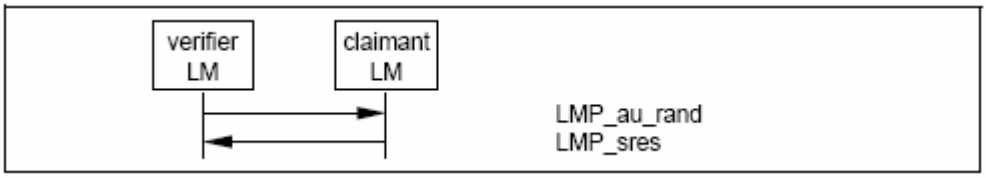
**(M) LMP_accepted, LMP_not_acecepted**

Este PDU es utilizado como mensajes de respuesta a otro PDU en diferentes procedimientos, conteniendo el opcode[5] del mensaje que es respondido a.

## 2.2 Autenticación

### (M) LMP_Au_rand, LMP_sres

El procedimiento de la autenticación se basa en un esquema challenge-response (desafió-respuesta). La verificadora envía un LMP_Au_rand PDU que contiene un número aleatorio (challenge) al claimant (demandante). El claimant calcula una respuesta, que es una función del challenge, el claimant BD_ADDR y una llave secreta (secret key). La respuesta es devuelta a la verificadora, que verifica si la respuesta es correcta o no. Un cálculo exitoso de la respuesta de la autenticación requiere que dos dispositivos compartan una llave secreta. El master y el slave pueden ser verificadoras.




Authentication. Claimant tiene un link key.

## 2.3 Pairing

### (M) LMP_en_rand, LMP_Au_rand, LMP_sres, LMP_comb_key, LMP_unit_key

Cuándo dos dispositivos no tienen una llave de enlace común (link key) una llave de inicialización (K init) es creada basada en un PIN y un número aleatorio. Cuándo ambos dispositivos han calculado el init K la llave de conexión se crea, y finalmente una autenticación mutua se hace. El procedimiento de pairing comienza cuando un dispositivo envía un LMP_en_rand; este dispositivo es referido como "iniciando LM " o el "iniciador" y el otro dispositivo es referido como " LM respondiendo " o el "contestador".

[5] Opcode
En informática , un Opcode es la porción de una instrucción en terminología de informática que especifica la operación que se realizará. El término es una abreviatura del código de Op. Sys. del eration . Su especificación y formato son presentados en la arquitectura del sistema de instrucción (ISA) del componente -- normalmente un CPU del hardware , pero posiblemente una unidad especializada. Algunas operaciones tienen *operandos* implícitos, o de hecho ningunos. Algún ISAs tiene instrucciones con los campos definidos para los opcodes y los operandos, mientras que otros tienen una estructura más complicada y más ad hoc.
Los operandos sobre los cuales los opcodes funcionan pueden depender de la arquitectura de la CPU, concite en registros, valores en memoria , valores, puertos de I/O, etc. Las operaciones que un opcode puede especificar pueden incluir aritméticas, operaciones lógicas, y control de programa.
Opcodes se puede también encontrar en los códigos del octeto interpretados por un intérprete de código (o máquina virtual). En éstos, la arquitectura del sistema de instrucción se crea para ser interpretada por el software, en un dispositivo de hardware. A menudo, los intérpretes de código del octeto funcionan con los tipos y las operaciones de alto nivel de datos que en un sistema de instrucción del hardware, pero se construyen a lo largo de líneas similares.

## 2.4 Cambio del Link Key.

### (M) LMP_comb_key

Si la llave de enlace se deriva de la combinación de llaves y el enlace actual es la llave semi-permanente de la conexión, la llave del enlace se puede cambiar. Si la llave del enlace es una llave de unidad, las unidades deben pasar por el procedimiento pairing para cambiar la llave de enlace. El contenido de LMP_comb_key es protegida por un bitwise XOR [6]con la llave actual del enlace.

| M/O | PDU | Contents |
|---|---|---|
| M | LMP_comb_key | random number |

Tabla 2.4 PDUs usado para cambiar el link key.

## 2.5 Cambia la Llave Actual de la Conexión

### (M) LMP_temp_rand , LMP_temp_key , LMP_use_semi_permanent_key

La llave actual de enlace puede ser una llave semi-permanente de la conexión o una llave temporal de la llave de enlace. Se puede cambiar temporalmente, pero el cambio es válido sólo para la sesión. Cambiar a una llave temporal de enlace es necesario si el piconet soporta una transmisión encriptada.

| M/O | PDU | Contents |
|---|---|---|
| O(23) | LMP_temp_rand | random number |
| O(23) | LMP_temp_key | key |
| O(23) | LMP_use_semi_permanent_key | - |

Table 2.5 PDUs usado para cambiar la actual link key.

## 2.6 Encryption

### (O) LMP_encryption_mode_req , LMP_encryption_key_size_req , LMP_start_encryption_req , LMP_stop_encryption_req

Si por lo menos una autenticación se ha realizado se puede utilizar encriptación. Si el master quiere que todos los slave en el piconet utilicen los mismos parámetros de encriptación deben emitir una llave temporal (master K) y hacer de esta llave, la

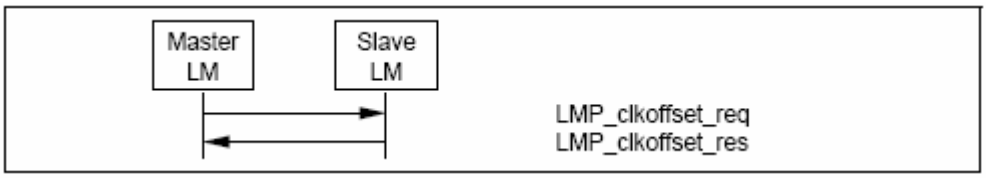
[6] En un programa, una operación bitwise opera en uno de los dos bit patterns o en numeros binarios en el nivel de bits individuales.
Un bitwise XOR toma dos bits patterns de longitud igual y realiza la operación lógica XOR en cada par de bits correspondientes. El resultado en cada posición es 1 si los dos bits son diferentes, y 0 si son los mismos.

llave actual del enlace para todos los slaves en el piconet, antes de que empiece la encriptación. Esto es necesario si la transmisión de paquetes se debe encriptar.

## 2.7 Clock Offset Request

### (M) LMP_clkoffset_req, LMP_clkoffset_res

Cuándo un slave recibe el paquete FHS, la diferencia se computa entre su propio reloj y el reloj master incluidos en el payload del paquete FHS. El clock offset se actualiza también cada vez que se recibe un paquete del master. El master puede solicitar este Clock Offset en cualquier momento durante la conexión. Salvando este Clock Offset, el master sabe en qué canal RF el slave alerta a page scan después que ha dejado el piconet. Esto se puede utilizar para acelerar el tiempo de paging la próxima vez que el mismo dispositivo se pagine.



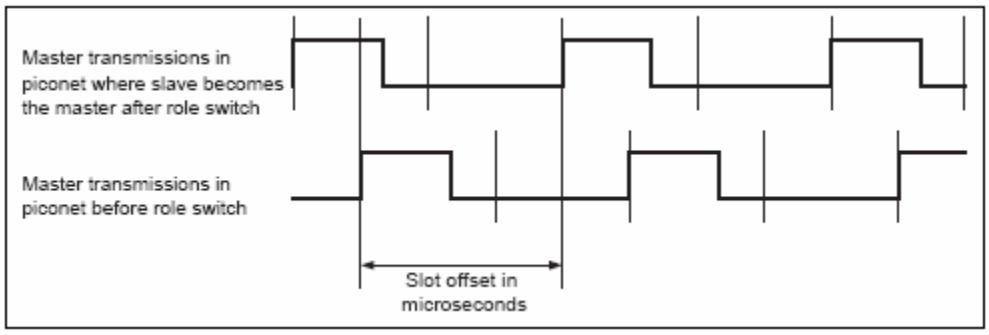

*Clock offset requested.*

## 2.8 Slot Offset Request

### (O) LMP_slot_offset

Con LMP_slot_offset la información acerca de la diferencia entre el slot boundaries de piconets diferentes, se transmite. Este PDU lleva los parámetros slot offset y BD_ADDR. El slot_offset es la sustracción en tiempo real del comienzo del master's TX slot en el piconet donde el PDU es transmitido en tiempo real del comienzo del master's TX slot en el piconet donde el dispositivo BD_ADDR es un modulo master 1250.

Antes de hacer un switch master-slave, este PDU se transmitirá del dispositivo que llega a ser master en el procedimiento switch. El PDU puede ser también útil en comunicaciones inter-piconet.




*2.8: Slot offset para role switch.*

## 2.9 Timing Accuracy Information Request (Pedido de Información de Certeza de Tiempo)

### (O) LMP_timing_accuracy_req , LMP_timing_accuracy_res

LMP soporta pedidos para la certeza de tiempo. Esta información se puede utilizar para minimizar el scan window para un tiempo hold dado cuando al volver a hold y para extender el tiempo hold máximo. Se puede utilizar también para minimizar el scan window cuando scanning para el slot modo sniff o paquetes beacon el modo park. Los parámetros de certeza de tiempo regresados son derivados del largo plazo medido en el ppm y la inestabilidad a largo plazo medido en µs del reloj utilizado durante el modo hold, sniff y park. Estos parámetros se fijan para cierto dispositivo y deben ser idénticos cuando es solicitado varias veces.

## 2.10 LMP Version

### (M) LMP_version_req , LMP_version_res

La capa de LMP soporta request para la versión del protocolo LM. El dispositivo solicitado enviara una respuesta con tres parámetros: VersNr, CompId y Sub-Versnr. VersNr especifica la versión que el dispositivo soporta, de la especificación Bluetooth de LMP. CompId se utiliza para rastrear los posibles problemas con las capas más bajas Bluetooth. Todas las compañías que crean una implementación única del Link Manager tendrán su propio CompId. La misma compañía es también responsable de la administración y la conservación del SubVersNr. Se recomienda que cada compañía tuviera un SubVersNr único para cada implementación RF/BB/LM.

## 2.11 Características Soportadas.

### (M) LMP_features_req , LMP_features_res

El Bluetooth radio y el Link Controller puede soportar sólo un subconjunto de los tipos de paquete y características descritas en la Especificación Baseband y en la Especificación Radio. Un dispositivo no puede enviar ningún paquete que no sea ID, FHS, NULL, POLL, DM1 o DH1 antes de estar enterado de las características soportadas por el otro dispositivo. Después que las características solicitadas se han llevado a cabo, la intersección de los tipos de paquetes soportados para ambos lados también se puede transmitir. Siempre que una solicitud se publica, debe ser compatible con las características soportadas por el otro dispositivo. Por ejemplo al establecer un enlace SCO el iniciador no puede proponer utilizar paquetes HV3 si ese tipo paquete no es soportado por el otro dispositivo.

## 2.12 Switch de Master-Slave Role

### (O) LMP_switch_req , LMP_slot_offset

Desde que el dispositivo paging llegue siempre a ser el master del piconet, es necesario en ocasiones un switch del role master-slave. Suponga que el dispositivo A es slave y el dispositivo B master. El dispositivo que inicia el switch completa la transmisión del mensaje actual L2CAP y entonces manda LMP_switch_req. Nota: en un slave iniciado master-slave switch, el slave (A) enviara primero LMP_slot_offset, después LMP_switch. En un master iniciado master-slave switch, the master (B) enviara primero LMP_switch, antes de recibir LMP_slot_offset del slave (A). Si el switch acepta, el otro dispositivo completa la transmisión del mensaje L2CAP actual y entonces responde con un LMP_accepted. El procedimiento de swith entonces toma lugar, y después el dispositivo A es master y el dispositivo B es slave.

## 2.13 Name Request (Solicitud de nombre).

### (M) LMP_name_req , LMP_name_res

LMP soporta solicitudes de nombre a otro dispositivo Bluetooth. El nombre es un nombre fácil de manejar asociado con el dispositivo Bluetooth y consiste en 248 byte máximo codificados según el Standard UTF-8. El nombre se fragmenta sobre uno o más paquetes DM1.

## 2.14 Detach (Separa)

### (M) LMP_detach

La conexión entre dos dispositivos Bluetooth puede ser cerrada en cualquier momento por el master o el slave. Un parámetro de razón se incluye en el mensaje para informar a la otra parte del por qué se cierra la conexión.

## 2.15 Hold Mode

### (O) LMP_hold , LMP_hold_req

El enlace ACL de una conexión entre dos dispositivos Bluetooth se puede colocar en el modo Hold para un tiempo hold especificado. Durante este tiempo ningún paquete ACL se transmitirá del master. El modo hold entra típicamente cuando no hay necesidad de enviar los datos para un tiempo relativamente largo. El transceptor entonces se puede estar apagado para salvar potencia. Pero el modo hold se puede utilizar también si un dispositivo quiere descubrir o ser descubierto por otros dispositivos Bluetooth, o tiene necesidad de unir otros piconets. Lo qué un dispositivo hace realmente durante el tiempo hold no es controlado por el mensaje hold, pero está arriba de cada dispositivo para decidir.

## 2.16 Sniff Mode

**(O) LMP_sniff_req , LMP_unsniff_req**

Para entrar en el modo Sniff, el master y el slave negocian un sniff intervalo, sniff T y sniff offset, sniff D, el cual especifica el tiempo de sniff slots. El offset determina el tiempo del primer sniff slots; después de que sniff slots siguen periódicamente con sniff interval T sniff. Cuándo la conexión está en el modo sniff el master puede sólo empezar una transmisión sniff slot. Dos parámetros controlan la actividad que atiende el slave. El parámetro sniff attempt determina para cuántos slots el slave debe atender, empezando por el sniff slot, incluso si no recibe un paquete con su propia dirección AM. El parámetro sniff timeout determina para cuántos slots adicionales el slave debe atender si continúa recibiendo sólo paquetes con su propia dirección AM.

## 2.17 Park Mode

**(O) LMP_park_req , LMP_unpark_PM_ADDR_req , LMP_unpark_BD_ADDR_req , LMP_set_broadcast_scan_window , LMP_modify_beacon**

Si un slave no necesita tomar parte en el canal, pero no obstante debe ser sincronizado FH-synchronized, se puede colocar en el modo park. En este modo el dispositivo renuncia a su AM_ADDR pero se re-sincroniza al canal levantándose en los instantes beacon, separados por el intervalo beacon. El intervalo beacon, beacon offset y la bandera indican cómo el primer instante beacon es calculado y determinado. Después que estos instantes, beacon siguen periódicamente en el intervalo beacon predeterminado. En el instante beacon que el slave park puede ser activado otra vez por el master, el master puede cambiar los parámetros del modo park, transmitir información o permitir que los slaves parked soliciten el acceso al canal.

Estos PDUs son los unicos PDUs que puede ser enviados a un slave en el modo park y solamente estos PDUs pueden ser transmitidos. Cuándo un slave es colocado en el modo park se le asigna un PM_ADDR único, que puede ser utilizado por el master en el unpark de este slave.

## 2.18 Power Control

**(O) LMP_incr_power_req , LMP_decr_power_req , LMP_max_power , LMP_min_power**

Si el valor de RSSI difiere demasiado del valor preferido de un dispositivo Bluetooth, puede solicitar un aumento o disminución de potencia de otro dispositivo TX. Tras la recepción de este mensaje, el poder de salida se aumenta o disminuye en un paso. En el lado del master el poder TX es completamente

independiente para diferentes slaves; la solicitud de un slave puede tomar efecto sólo en el poder TX del master para ese mismo slave Los request de ajuste del poder se pueden hacer en cualquier momento siguiendo el procedimiento paging baseband exitoso.

## 2.19 Channel Quality-Driven Change (entre DM and DH)

### (O) LMP_auto_rate , LMP_preferred_rate

El rendimiento de transferencia de los datos para un tipo de paquete dado, depende de la calidad del canal RF. Las medidas de calidad en el receptor de un dispositivo se pueden utilizar para controlar dinámicamente el tipo de paquete transmitido del dispositivo remoto, para la optimización del rendimiento de transferencia de datos. Si un dispositivo A necesita que el dispositivo remoto B tenga este control envía LMP_auto_rate una vez. El dispositivo B entonces puede devolver LMP_preferred_rate al dispositivo A siempre que desee cambiar el tipo de paquetes que A transmite.

Este PDU tiene un parámetro que determina la codificación elegida (con o sin 2/3FEC) y el tamaño preferido (en slots) de los paquetes. El dispositivo A no es requerido para cambiar el tipo de paquete especificado por este parámetro y nunca puede enviar un paquete que sea más grande que el número de slots máximo permitido incluso si el tamaño elegido es más grande que este valor.

## 2.20 Quality of Service

### (M) LMP_quality_of_service , LMP_quality_of_service_req

El LM proporciona las capacidades de Quality of Service (Calidad de Servicio). Un intervalo Poll, que se define como el tiempo máximo entre transmisiones subsiguientes del master a un slave particular, es utilizado para soportar la asignación de ancho de banda y el control latente. Además, el master y el slave negocian el número de repeticiones para la transmisión de paquetes (NBC).

## 2.21 SCO Links

### (O) LMP_SCO_link_req , LMP_remove_SCO_link_req

Cuándo una conexión se ha establecido entre dos dispositivos Bluetooth la conexión consiste en un enlace ACL. Uno o más enlaces SCO se pueden establecer entonces. El enlace SCO reserva slots separados por el intervalo SCO, T sco. El primer slot reservado para el enlace SCO es definido por $T_{sco}$ y SCO delay, $D_{sco}$. Después estos slots de SCO siguen periódicamente con el intervalo SCO. Cada enlace SCO se distingue de todos los otros enlaces SCO por SCO handle.

## 2.22 Control de Multi-Slot Packets

### (M) LMP_max_slot , LMP_max_slot_req

El número de slots utilizados por un dispositivo se puede limitar. Un dispositivo permite al dispositivo remoto utilizar un número máximo de slots enviando el PDU LMP_max_slot que proporcionar los slots máximos como parámetro. Cada dispositivo puede solicitar un número máximo de slots a utilizar, enviando el PDU LMP_Max_slot_req que proporciona el parámetro de los slots máximos. Después de una conexión nueva, como resultado de page, page scan, master-slave switch or unpark, el valor por default es de 1 slot. Dos PDUs se utilizan para el control de paquetes multi-slot. Estos PDUs pueden ser enviados en cualquier momento después de que el arreglo de la conexión este completa.

## 2.23 Esquema Paging

### (O) LMP_page_mode_req , LMP_page_scan_mode_req

Además del esquema obligatorio de Paging, el sistema Bluetooth define un esquema opcional Paging. LMP proporciona un medio para negociar el esquema paging, el cual deberá ser utilizado la próxima vez que una unidad este en paged.

## 2.24 Link Supervision

### (M) LMP_supervision_timeout

Cada conexión Bluetooth tiene un reloj que se utiliza para la supervisión de la conexión. Este reloj se utiliza para discernir la pérdida de la conexión causada por dispositivos que mudan del rango, un dispositivo power-down(bajo poder), u otros casos de falla semejantes. Un procedimiento de LMP es utilizado para poner el valor del tiempo muerto de supervisión.

## 2.25 Connection Establishment

### (M) LMP_host_connection_req , LMP_setup_complete

Cuándo el dispositivo paging desea crear una conexión que implica capas encima de LM, envía LMP_host_connection_req. Cuándo en el otro lado recibe este mensaje, el host es informado acerca de la conexión entrante. El dispositivo remoto puede aceptar o rechazar el pedido de conexión enviando LMP_accepted o LMP_no_accepted.

Si LMP_host_connection_req se acepta, los procedimientos de la seguridad de LMP (pairing, authentication and encryption) pueden ser invocados. Cuándo un dispositivo no iniciará algún procedimiento de seguridad durante el establecimiento de la conexión envía LMP_setup_complete. Cuándo ambos dispositivos han

enviado LMP_setup_complete, el primer paquete en un canal lógico diferente de LMP, entonces puede ser transmitido.

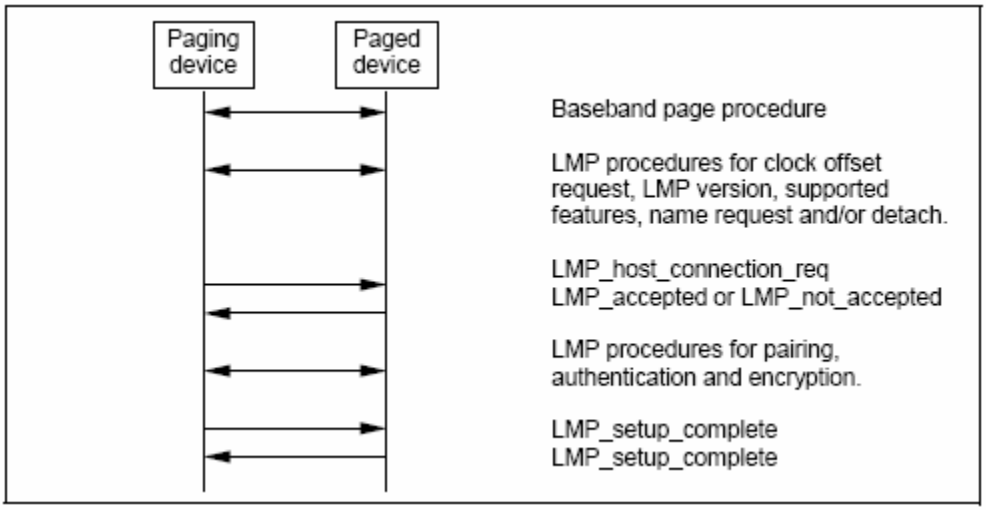



*Figura 2.25 Connection establishment.*

## 2.26 Test Mode

### (M) LMP_test_activate , LMP_test_control

LMP tiene PDUs para soportar diferentes modos de la prueba Bluetooth, que se utilizan para la certificación y la conformidad de prueba de la Radio y Baseband Bluetooth.

La activación se puede llevar a cabo localmente (vía una interfase HW o el SW), o utilizar la interfase air.

- Para la activación sobre la interfase air, entrando en el test mode se permitirá localmente para seguridad y razones de tipo de aprobación. La implementación de este no esta sujeta a la estandardización. El probador manda una orden de LMP que forzará el DUT para entrar en el modo de. El DUT terminará toda operación normal antes de entrar en el modo de prueba. El DUT volverá un LMP_Accepted en la recepción de una orden de activación. LMP_no_accepted será vuelto si el DUT no se permite localmente.
- Si la activación se realiza utilizando localmente una interfase HW o el SW, el DUT terminará toda operación normal antes de entrar en el modo de prueba. Hasta que una conexión al verificador exista, el dispositivo realizará un page scan e inquiry scan. Se recomienda prolongar la actividad scan.

| LMP PDU | PDU number | Possible Direction | Contents | Position in Payload |
|---|---|---|---|---|
| LMP_test_activate | 56 | m → s | | |
| LMP_test_control | 57 | m → s | test scenario<br>hopping mode<br>TX frequency<br>RX frequency<br>power control mode<br>poll period<br>packet type<br>length of test data | 2<br>3<br>4<br>5<br>6<br>7<br>8<br>9-10 |
| LMP_detach | 7 | m → s | | |
| LMP_accepted | 3 | m ← s | | |
| LMP_not_accepted | 4 | m ← s | | |

*Tabla 2.26 LMP messages used for Test Mode*

El control PDU se utiliza tanto para el transmisor como para loop back tests. Las siguientes restricciones aplican para los parámetros de los escenarios:

| Parameter | Restrictions Transmitter Test | Restrictions Loopback Test |
|---|---|---|
| TX frequency | 0 ≤ k ≤ 93 | 0 ≤ k ≤ 78 |
| RX frequency | same as TX frequency | 0 ≤ k ≤ 78 |
| Poll period | | not applicable (set to 0) |
| Length of test sequence | depends on packet type:<br>DH1: <= 27 bytes<br>DH3: <= 183 bytes<br>DH5: <= 339 bytes<br>AUX1: <= 29 bytes<br>HV3: = 30 bytes<br>EV3: <= 30 bytesEV5: <= 180 bytes | For ACL and SCO packets: not applicable (set to 0)<br><br>For eSCO packets:<br>EV3: <= 1-30 bytes<br>EV4: <= 1-120 bytes<br>EV5: <= 1-180 bytes |

*Tabla 2.26 Restrictions for Parameters used in LMP_Test_Control PDU*

## 2.27 Error Handling

### (M) LMP_not_accepted

Si el Link Manager recibe un PDU con un opcode irreconocible, responde con LMP_not_accepted con el código de razón LMP PDU *desconocido.* El parámetro

de opcode que resuena es el opcode irreconocible. Si el Link Manager recibe un PDU con parámetros inválidos, responde con LMP_no_accepted con el código de razón *parámetros inválidos* de LMP. Si el tiempo de respuesta máximo se excede o si un enlace es perdido se detecta la parte que espera por la respuesta, y concluirá que el procedimiento ha terminado sin éxito.

Los mensajes erróneos LMP pueden ser causados por errores en el canal o errores sistemáticos en el lado de transmisión. Para detectar el último caso, el LM debe monitorear el número de mensajes erróneos y desconectar si excede un umbral de tiempo, la cual es una implementación-dependiente.

# PARTE E
# HOST CONTROLLER INTERFACE (HCI)

El HCI proporciona un comando de interface al controlador de la baseband y al Link Manager, el acceso al estado del hardware y a los registros de control. Esencialmente esta interface proporciona un método uniforme para acceder a las capacidades baseband Bluetooth. HCI existe a través de 3 secciones, Host - Transport Layer - Host Controller. Cada una de estas secciones tiene un papel diferente en el sistema de HCI.

## 1 Entidades Funcionales HCI

El HCI se separa funcionalmente en 3 partes:

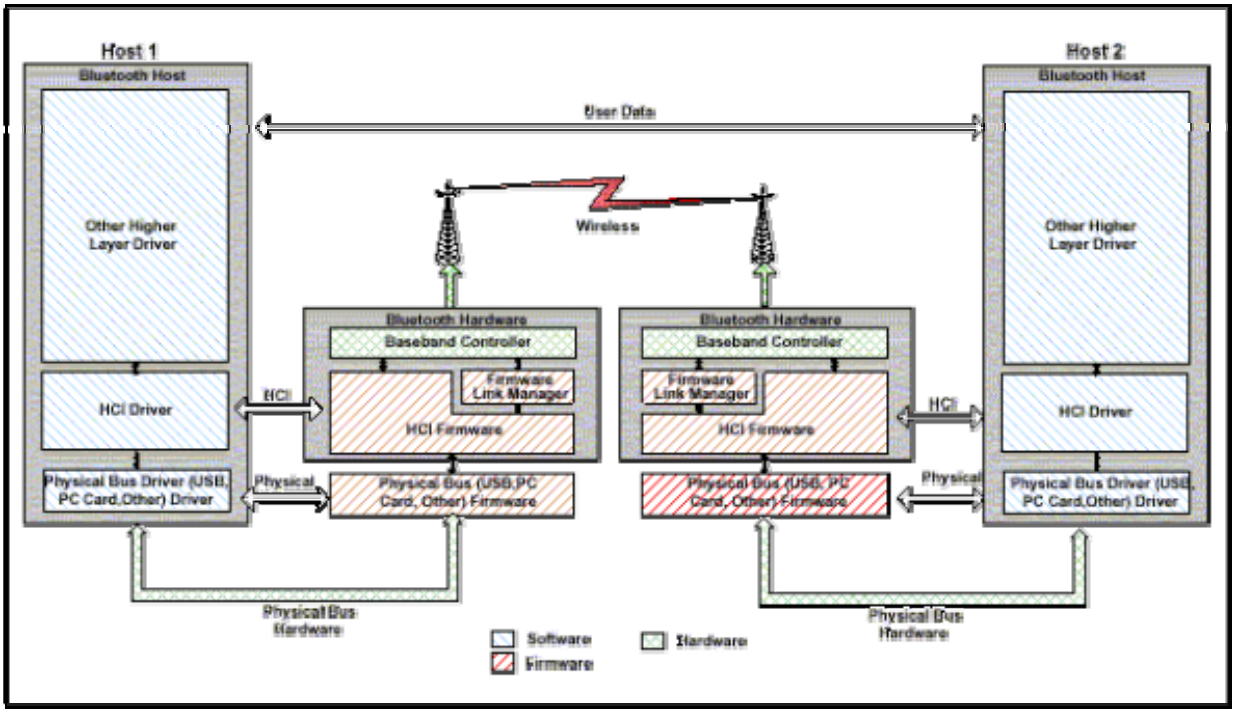



*Figura 1.2: End to End Overview of Lower Software Layers to Transfer Data*

La figura 1,2, ilustra el path de una transferencia de datos de un dispositivo a otro. El HCI driver en el intercambio de datos en e Host y comandos con el firmware HCI en el hardware Bluetooth. La Host Control Transport Layer (bús físico) driver proporciona ambas capas de HCI con la habilidad de cambiar información uno con el otro.

El Host recibe las notificaciones asíncronas de eventos independientes de HCI del cual el Host Control Transport Layer es utilizado. Los eventos HCI se utilizan para

notificar al Host cuando algo ocurre. Cuándo el Host descubre que un acontecimiento ha ocurrido, entonces, analizará sintácticamente el paquete recibido, del evento para determinar que evento ocurrió.

## 1.1 HCI Firmware (ubicación: Host Controller)

El HCI Firmware, se localiza en el Host controller, ( el actual dispositivo hardware Bluetooth). El HCI Firmware aplica los comandos HCI para el hardware Bluetooth accediendo a los comandos Baseband, comandos link manager, registros de estado de hardware, control de registros, y registros Events. El término Host controller significa dispositivo Bluetooth HCI- enabled (HCI-permitido)

## 1.2 HCI Driver (ubicación: Host)

El Drive HCI, el cual se localiza en el Host (entidad del software). El Host recibirá las notificaciones asíncronos HCI events, estos se utilizan para notificar al Host cuando algo ocurre. Cuándo el Host descubre que un Event ha ocurrido entonces analizará sintácticamente el paquete event recibido para determinar que event ocurrió. El termino Host significa Unidad HCI- enabled (HCI-permitido) de la unidad de Software.

## 2 Host Controller Transport Layer (ubicación: Intermediate Layers)

El HCI Driver and Firmware se comunican vía la Host Controller Transport Layer (Capa del Transporte del Host Controller), es decir una definición de las varias capas que pueden existir entre el HCI drive en el sistema del Host y el HCI Firmware en el hardware de Bluetooth. Estas capas intermedias, el Host Controller Transport Layer, debe proporcionar la habilidad de transferir los datos sin el conocimiento profundo de los datos que se están transfiriendo. Varias Capas diferentes del Host Controller se pueden utilizar, de las cuáles 3 han sido definidas inicialmente para Bluetooth**: USB**, **UART y RS232.** El Host debe recibir las notificaciones asíncronas de HCI events independientes de las cuales se utiliza Host Controller Transport Layer.

## 3 Comandos HCI

El HCI proporciona un método uniforme de comandos de acceso a las capacidades del hardware de Bluetooth. Los comandos de HCI Link provee al Host con la habilidad de controlar la conexión de la capa de enlace de con otros dispositivos Bluetooth. Estos comandos implican típicamente al Link Manager (LM ) cambiar comandos LMP con dispositivos remotos Bluetooth. Los comandos HCI Policy se utilizan para afectar el comportamiento del LM local y remoto. Estos comandos Policy provee al Host con métodos de influencia cómo el LM administra al piconet. El *Host Controller and Baseband commands*, *Informational commands* ,

and *Status commands*, proporcionan al Host acceso a varios registros en el Host Controller.

Los commands y events que se envían entre el Host y el Controller

| | |
|---|---|
| Generic Events | The generic events can occur due to multiple commands, or events that can occur at any time. |
| Device Setup | The device setup commands are used to place the Controller into a known state. |
| Controller Flow Control | The controller flow control commands and events are used to control data flow from the Host to the controller. |
| Controller Information | The controller information commands allow the Host to discover local information about the device. |
| Controller Configuration | The controller configuration commands and events allow the global configuration parameters to be configured. |
| Device Discovery | The device discovery commands and events allow a device to discover other devices in the surrounding area. |
| Connection Setup | The connection setup commands and events allow a device to make a connection to another device. |
| Remote Information | The remote information commands and events allow information about a remote device's configuration to be discovered. |
| Synchronous Connections | The synchronous connection commands and events allow synchronous connections to be created |
| Connection State | The connection state commands and events allow the configuration of a link, especially for low power operation. |
| Piconet Structure | The piconet structure commands and events allow the discovery and reconfiguration of piconet. |
| Quality of Service | The quality of service commands and events allow quality of service parameters to be specified. |
| Physical Links | The physical link commands and events allow the configuration of a physical link. |
| Host Flow Control | The Host flow control commands and events allow flow control to be used towards the Host. |
| Link Information | The link information commands and events allow information about a link to be read. |
| Authentication and Encryption | The authentication and encryption commands and events allow authentication of a remote device and then encryption of the link. |
| Testing | The testing commands and events allow a device to be placed into test mode. |

*Tabla 3.1: Overview of commands and events*

## 3.1 HCI Events

Un número de diferentes Events se definen para la capa de HCI. Los Events proporcionan un método para devolver los parámetros y los datos asociados a

cada Events. 32 HCI Events diferentes se han implementado hasta ahora, ellos recorren *Inquiry Complete Event* a *Page Scan Repetition Mode Change Even*.

## 3.2 Autenticación y Encriptación

El grupo de la autenticación y encriptación de comandos y eventos permite la autenticación de un dispositivo remoto y la encriptación de la conexión a uno o más dispositivos remotos**.**

## 3.3 TESTING

El grupo de comandos y eventos testing permite a un dispositivo ser colocado en un modo testing especial.

# 4 Flow Control

El Flow Control se utiliza en la dirección del Host al Host Controller para evitar llenar los buffers de datos del Host Controller con datos de ACL destinados para un dispositivo remoto (conexión handle) que no responde. Es el Host el que maneja los buffers del Host Controller.

## 4.1 Desconexión Behaviour

Cuándo el Host recibe un evento Desconexión Completa, el Host asumirá que todos Paquetes no reconocidos de Datos de HCI han sido mandados al Controller para regresar el Connection Handle que ha sido limpiado, y que los buffers correspondientes de datos se han liberado. El Host no tiene que notificar al Controller acerca de esto en varios eventos Completed Packets.

## 4.2 Códigos de Error HCI

Si un error ocurre para un comando, para el cual un evento Command Complete es regresado, el campo de los parámetros de regreso no puede contener todos los parámetros del regreso especificado para el comando. El parámetro del Status, que explica la razón del error y que es el primer parámetro de regreso, siempre será devuelto. Si hay un parámetro Connection Handle o un derecho del parámetro BD_ADDR después del parámetro Status, este parámetro también regresara para que el Host pueda identificar en cuál caso de un comando la Command Complete event pertenece. En este caso, el Connection Handle o el parámetro BD_ADDR tendrán exactamente el mismo valor como en el parámetro comando correspondiente. Es la implementación específica si más parámetros se regresan en caso de un error.

Muchos códigos de error se han definido para la capa de HCI. Cuándo un comando falla, los códigos de Error se emiten para indicar la razón del error.

Treinta y cinco códigos de error HCI se tienen hasta ahora definidos, del *Unknown HCI Command* to *LMP PDU Not Allowe.*

## 5 Cambio de información HCI-Specific

La Capa del Host Controller Transport Layer proporciona el cambio transparente de información HCI-Specific. Estos mecanismos de transporte proveen la habilidad al Host para enviar comandos HCI, datos ACL , y datos SCO al Host Controller. Estos mecanismos de transporte proporcionan además la habilidad para que el Host reciba eventos HCI, datos ACL, y datos SCO del Host Controller. Desde Host Controller Transport Layer proporciona el cambio transparente de información de HCI-Specific, y la especificación de HCI especifica el formato de los eventos de los comandos, y cambio de datos entre el Host y el Host Controller.

## 6 Comandos de Control de Enlace (Link Control Commands)

El Link Control Commands permite al Host Controller controlar las conexiones a otros dispositivos Bluetooth. Cuándo el Link Control Commands se utiliza, el Link Manager (LM) controla los piconets y scatternets Bluetooth se establecen y son soportados. Estos comandos ordenan al LM crear y modificar las conexiones de capa de enlace con dispositivos Bluetooth remotos, realizan Inquiries de otros dispositivos Bluetooth en el rango y en otros comandos LMP.

### 6.1 Link Policy Commands

El Link Policy Commands proporciona los métodos para que el Host afecte en cómo el Link Manager maneje el piconet. Cuándo Link Policy Commands se utiliza, el Link Manager aun controla los piconets y scatternets Bluetooth se establecen y son soportados, dependiendo de los parámetros ajustables de Policy. Estos comandos de Policy modifican el comportamiento de Link Manager que puede tener como resultado los cambios a las conexiones de la capa de enlace con dispositivos Bluetooth remotos.

### 6.2 Host Controller & Baseband Commands

El Host Controller & Baseband Commands proporcionan el acceso y el control a varias capacidades del hardware Bluetooth. Estos parámetros proporcionan el control de dispositivos Bluetooth y las capacidades del Host Controller, Link Manager y Baseband. El dispositivo Host puede utilizar estos comandos para modificar el comportamiento del dispositivo local.

### 6.3 Parámetros de Información

Los Parámetros de información son fijados por el fabricante del hardware de Bluetooth. Estos parámetros proporcionan información acerca del dispositivo

Bluetooth y las capacidades del Host Controller, Link Manager, y Baseband. El dispositivo Host no puede modificar cualquiera de estos parámetros.

## 6.4 Parámetros de Posición

El Host Controller modifica todos los parámetros de posición. Estos parámetros proporcionan información acerca del estado actual del Host Controller, Link Manager y Baseband. El dispositivo Host no puede modificar cualquiera de estos parámetros de otra manera que reponer ciertos parámetros específicos.

## 6.5 Testing Commands

El testing Commands se utiliza para proporcionar la habilidad de probar varias funcionalidades del hardware Bluetooth. Estos comandos proporcionan la habilidad de inducir condiciones para la prueba.

# 7 Delimitacion - Bluetooth Host Controller Transport Layers

## 7.1 UART Transport Layer

El objetivo de la Capa HCI UART Transport Layer es hacer posible utilizar el HCI Bluetooth sobre una interface serial entre dos UARTs en la misma PBC. El HCI UART Transport Layer asume que la comunicación de UART es libre de errores de línea. Eventos y paquetes de datos fluyen por esta capa, pero la capa no los decodifica.

## 7.2 RS232 Transport Layer

El objetivo de la Capa HCI RS232 Transport Layer deberá hacer posible utilizar el HCI Bluetooth sobre una interface física RS232 Host Bluetooth y el Host Controller Bluetooth. Eventos y paquetes de datos fluyen por esta capa, pero la capa no los decodifica.

## 7.3 USB Transport Layer

El objetivo de Universal Serial Bus (USB) Transport Layer está en hacer uso de un hardware USB interface para hardware de Bluetooth (la cual se puede personalizar de una o dos maneras: como un dongle[7] USB, o integrado en el motherboard de un PC notebook). Una clase de código se utilizará para especificar a todos los dispositivos de USB Bluetooth. Permite también al HCI commands para diferenciarse de los comandos USB a través del punto final del control.

[7] Dongle: Dispositivo de seguridad para software.

# Capitulo III
# Core System Package
# [Host volume]

# PARTE A
# L2CAP Logical Link Control and Adaptation Protocol

El Control Lógico de Conexión y Protocolo de Adaptación (L2CAP) es la capa sobre el Protocolo Baseband y reside en los datos de enlace de capa. L2CAP proporciona servicios de datos connection-oriented[1] y connectionless[2] a la capa superior del protocolo con la capacidad de multiplexión del protocolo, con la operación de segmentación y ensamble, con las abstracciones del grupo. L2CAP permite protocolos de más alto nivel y aplicaciones para transmitir y recibir paquetes de datos L2CAP de hasta 64 kilo bites de longitud.

Dos tipos de enlace se soportan en la capa Baseband: Enlace Synchronous Connection-Oriented (SCO) y Enlace Asynchronous Connection-Less (ACL). Enlace SCO soporta tráfico de voz en tiempo real usando el ancho de banda reservado. El enlace ACL soporta mejor el effort traffic. Las especificaciones de L2CAP se definen sólo para enlaces ACL y no son planeadas para soportar enlaces SCO.

## 1 Requisitos Funcionales L2CAP

L2CAP soporta varios requisitos importantes del protocolo:

### 1.1 Protocol Multiplexing

L2CAP debe soportar el protocolo multiplexing porque el Protocolo Baseband no soporta cualquier “tipo”, campo que identifica la capa más alta del protocolo siendo multiplexado encima de el, L2CAP debe ser capaz de distinguirse entre protocolos superiores de tal capa, como el Service Discovery Protocol, RFCOMM, y el Control de Telefonía.

[1] Connection-oriented: transferencia de datos que requiere que se establezca un circuito virtual.
[2] Connectionless: Termino utilizado para describir la transferencia de datos sin la existencia de un circuito virtual.

## 1.2 Segmentation & Reassembly

Comparado a otros medios físicos alambrados, los paquetes de datos definidos por el Protocolo Baseband se limitan en tamaño. Exportar una unidad máxima de transmisión (MTU) es asociado con el payload más grande de Baseband (341 byte para paquetes DH5) limita el uso eficiente del ancho de banda para protocolos en la capa más alta que se diseñan para utilizar paquetes más grandes. Los paquetes L2CAP grandes se deben dividir en múltiples paquetes pequeños Baseband antes de su transmisión en el aire. Asimismo, múltiples paquetes recibidos Baseband pueden ser devueltos en un solo paquete L2CAP más grande, que sigue un chequeo sencillo de integridad. La funcionalidad, Segmentation and Reassembly (SAR) es absolutamente necesaria para soportar protocolos que utilizan paquetes más grandes que los soportados por Baseband.

## 1.3 Quality of Service

El proceso de establecimiento de conexión L2CAP permite el cambio de información con respecto a Quality of Service (QoS) entre dos unidades de Bluetooth. Cada implementación L2CAP debe controlar los recursos utilizados por el protocolo y asegura que los contratos QoS se cumplan.

## 1.4 Grupos

Muchos protocolos incluyen el concepto de un grupo de direcciones. El Protocolo Baseband soporta el concepto de piconet, un grupo de dispositivos síncronos hopping juntos utilizando el mismo reloj. La abstracción del grupo L2CAP permite que las implementaciones tracen eficientemente los grupos del protocolo en los piconets. Sin una abstracción del grupo, el nivel más alto del protocolo necesitaría ser expuesto al Protocolo Baseband y la funcionalidad del Link Manager para manejar eficientemente los grupos.

# 2 Operación General de L2CAP

La capa L2CAP se basa alrededor del concepto de “*canales ".* Cada uno de los end-points de un canal L2CAP es referido por un *channel identifier.*

## 2.1 Channel Identifiers (identificador de Canal).

Channel Identifiers (CIDs) son los nombres locales que representan un end-point de un canal lógico en el dispositivo. Las implementaciones son libres de manejar el CIDs de la mejor manera convenida para esta implementación particular, con la provisión que el mismo CID no es reusado como un end-point local de un canal L2CAP para múltiples canales L2CAP simultáneos entre un dispositivo local y algún dispositivo remoto.

La tarea del CID es relativa a un dispositivo particular y un dispositivo puede asignar CIDs independientemente de otros dispositivos (a excepción de ciertos CIDs reservados, tal como la señalización del canal). Así, incluso si el mismo valor del CID a sido asignado a (telemando) los puntos finales del canal por varios dispositivos remotos conectados a un solo dispositivo local, el dispositivo local puede asociarse todavía extraordinariamente a cada CID remoto con un dispositivo diferente.

## 2.2 Operación Entre Dispositivos

Los canales de datos de connection-oriented representan una conexión entre dos dispositivos, donde un CID identifica cada end-point del canal. Los canales de connectionless restringen el flujo de datos en una sola dirección. Estos canales se utilizan para soportar un canal “grupo” donde el CID en el origen representa uno o más dispositivos remotos. Hay también varios CIDs reservados para propósitos especiales. La señalización del canal es un ejemplo de un canal reservado. Este canal se utiliza para crear y establecer los canales de datos de connection-oriented y para negociar cambios en las características de estos canales. El soporte para la señalización de un canal dentro de una entidad L2CAP es obligatorio. Otro CID se reserva para todo el tráfico de datos entrantes connectionless.

## 2.3 Operación Entre Capas

Las implementaciones L2CAP siguen la arquitectura general descrita aquí:

- Las implementaciones L2CAP deben transferir datos entre capas más altas y bajas del protocolo.
- Cada implementación debe soportar también un conjunto de comandos de señalización para el uso entre implementaciones L2CAP.
- Las implementaciones L2CAP se deben preparar también para aceptar ciertos tipos de acontecimientos de capas más bajas y generar eventos en capas superiores. Cómo estos eventos se pasan entre capas es un proceso de implementación-dependiente.

## 2.4 Segmentation & Reassembly

Las operaciones Segmentation & Reassembly (SAR) se utilizan para mejorar la eficiencia soportando una unidad máxima de transmisión (MTU) con un gran tamaño que el paquete más grande Baseband. Esto reduce por encima, separando los paquetes de la red y del transporte usados por protocolos de capa más altas sobre varios paquetes Baseband. Todos los paquetes L2CAP se pueden dividir para la transferencia sobre paquetes Baseband. El protocolo no realiza cualquier operación de Segmentation & Reassembly pero el formato del paquete soporta adaptación en pequeñas tramas físicas.

Una implementación L2CAP expone la salida (es decir, la recepción del Host remoto) el MTU saliente divide paquetes en capas más altas en "**chunks"** que se pueden pasar al Link Manager vía Host Controller Interface (HCI), siempre que uno exista. En el lado de recepción, una implementación L2CAP recibe "**chunks"** del HCI y re-ensambla estos chunks en paquetes L2CAP usando la información proporcionada con el HCI y del packet header.

## 3 L2CAP State Machine

Esta sección describe el canal L2CAP connection-oriented state machine. La sección define los estados, acontecimientos que causan las transiciones de estado, y las acciones a ser realizadas en respuesta a acontecimientos. Este state machine es solamente pertinente a CIDs bidireccional y no es representante del canal de señalización o del canal unidireccional.

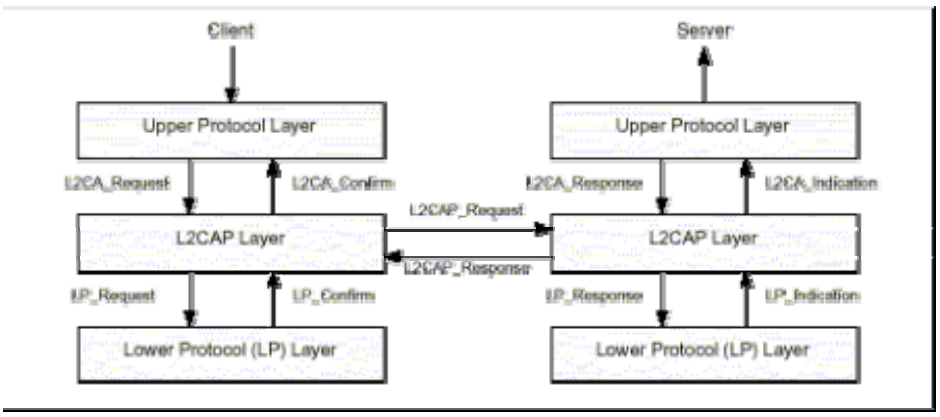



Fig. 3.1* Fuente Del Diagrama: Cortesía de SIG de Bluetooth, Specs L2CAP

La figura 3.1* arriba ilustra los acontecimientos y las acciones realizadas por una implementación de la capa L2CAP. El cliente y el servidor representan simplemente el inicio de la solicitud y el aceptante del pedido respectivamente. Un nivel de aplicación cliente iniciaría y aceptaría peticiones. La convención de nominacion es como sigue.

- La interface entre dos capas (interface vertical) utiliza el prefijo de la capa más baja que ofrece el servicio a la capa más alta, por ejemplo, L2CA.
- La interface entre dos entidades de la misma capa (interface horizontal) utiliza el prefijo del protocolo (agregando una P a la identificación de la capa), por ejemplo, L2CAP.
- Los acontecimientos que vienen de antedicho (comenzando arriba) son llamados Request (Req) y las respuestas correspondientes se llaman Confirmación (Cfm).

- Los acontecimientos que vienen de abajo (empezando abajo) son llamados las Indicaciones (Ind) y las respuestas correspondientes se llaman las Respuestas (Rsp).
- Las respuestas que requieren aún más procesamiento se llama Pendiente (Pnd). La anotación para Confirmación y Respuesta asumen contestaciones positivas. Contestaciones negativas son denotadas por un 'Neg' el sufijo tal como L2CAP_ConnectCfmNeg.

# 4 Otras Características de L2CAP

## 4.1 Formato de Paquete de Datos

L2CAP es un paquete basado en un modelo de comunicación en *el canal* A. Un canal representa un flujo de datos entre entidades L2CAP en dispositivos remotos. Los canales pueden ser connection-oriented o connectionless. Todos campos del paquete utilizan la orden del byte Little Endian.

## 4.2 Señalización

Varios comandos de señalización se pueden pasar entre dos entidades L2CAP en dispositivos remotos. Todos los comandos de señalización son envíadas a CID 0x0001 (el canal de señalización). La implementación L2CAP debe ser capaz de determinar la dirección Bluetooth (BD_ADDR) del dispositivo que envía las órdenes. Los comandos múltiples se pueden enviar en un solo paquete (L2CAP) y los paquetes se envían a CID 0x0001. Los comandos MTU toman la forma de peticiones y respuestas.

## 4.3 Configuración del Parámetro de Opción

Las opciones son un mecanismo para ampliar la capacidad de negociar diversos requisitos de conexión. Las opciones se transmiten en forma de elementos de información comprende un tipo de opción, una longitud de opción, y uno o más campos de datos de opción.

## 4.4 Service Primitivos (Servicios primitivos)

Varios servicios son ofrecidos por L2CAP en términos de Service Primitivos y parámetros. La interface de servicio es requerida para pruebas. Ellos incluyen primitivos a:

- **Connection**: setup , configura , disconnect
- **Data**: read , write
- **Group**: create, close, add member, remove member , get membership
- **Information**: ping, get info, request a call-back at the occurrence of an event
- **Connection-less Traffic**: enable, disable

# PARTE B
# SERVICE DISCOVERY PROTOCOL (SDP)

El protocolo de servicio de descubrimiento (SDP) proporciona los medios para aplicaciones, de descubrir cuales servicios están disponibles y para determinar las características de estos servicios disponibles.

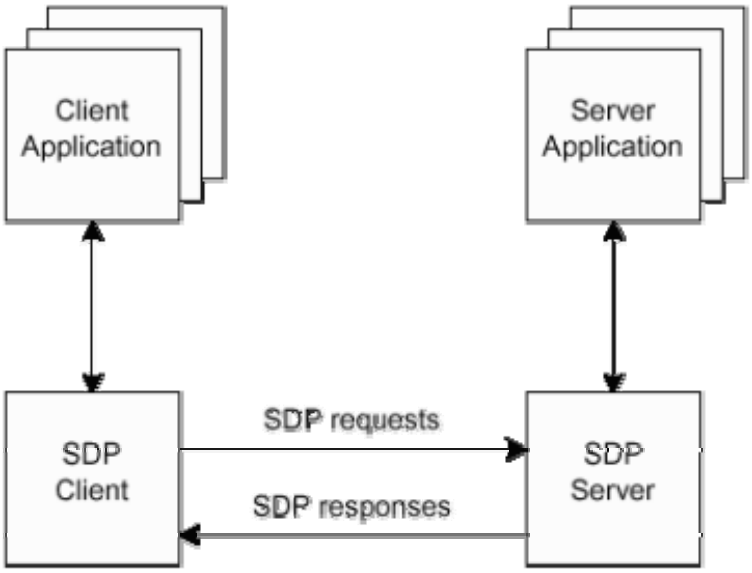
Un protocolo específico del Descubrimiento de Servicio se necesita en el ambiente Bluetooth, como el sistema de los servicios que están disponibles dinámicamente a cambio basados en la proximidad del RF de dispositivos en movimiento, cualitativamente diferente del descubrimiento de servicio en ambientes tradicionales basados en red. El protocolo de descubrimiento de servicio definido en la especificación de Bluetooth se intenta dirigir a las características extraordinarias en ambiente Bluetooth.

## 1 Arreglo del Protocolo SDP

### 1.1 Vista General

SDP es un protocolo sencillo con requisitos mínimos en el transporte subyacente. Puede funcionar sobre un transporte seguro de paquete (o aún no fiables, si el cliente aplica tiempos muertos y repite request que sean necesarios). SDP utiliza un modelo de request/response (pedido/respuesta) donde cada transacción consiste de un request protocol data unit (PDU) y una respuesta PDU. Sin embargo, las peticiones pueden potencialmente ser canalizadas y las respuestas pueden regresar estando fuera de servicio.

En el caso donde SDP se utiliza con el protocolo del transporte Bluetooth L2CAP, sólo un SDP request PDU por conexión da un servidor SDP, que puede sobresalir en un instante dado. Es decir, un cliente debe recibir una respuesta a cada pedido antes de publicar otro pedido en la misma conexión L2CAP. La limitación del SDP a enviar un pedido unacknowledged PDU (no reconocido) proporciona una forma simplificada del control del flujo.




*La Fuente de esquema: SDP Specs, Cortesía de Bluetooth SIG

## 1.2 Formato PDU

Cada SDP PDU consiste en un header PDU seguido por parámetros PDU-específicos. El header contiene tres campos:

- **PDU ID:** el campo identifica el tipo de PDU. Es decir su significado y los parámetros específicos.
- **Transaction ID:** el campo identifica únicamente la petición PDUs y se utiliza para igualar la respuesta PDUs a la petición PDU.
- **ParameterLength** el campo especifica la longitud (en byte) de todos los parámetros contenidos en el PDU.

Los parámetros pueden incluir un parámetro de **estado de continuación**, descrito abajo; los parámetros PDU-ESPECIFICOS para cada de tipo PDU son descritos luego en descripciones separadas de PDU.

## 1.3 Respuestas Parciales y Estado de Continuación

Algunos request SDP pueden requerir respuestas que son más grandes y que puede encajar en una sola respuesta PDU. En este caso, el servidor SDP generara una respuesta parcial junto con un parámetro de Estado de la Continuación. El parámetro de Estado de Continuación puede ser suministrado por el cliente en un pedido subsiguiente para recuperar la próxima porción de la respuesta completa.

## 1.4 Manejo de Error

Cada transacción consiste en un request y una respuesta PDU. Generalmente, cada tipo de request PDU tiene un tipo correspondiente de respuesta PDU. Sin embargo, si el servidor determina que un request se ajusta al formato incorrectamente o por alguna razón el servidor no puede responder con el tipo de PDU apropiado, responderá con un error PDU (*SDP_ErrorResponse).*

# 2 Servicios de SDP

La siguiente sección describe cómo las características individuales (servicios) de los diversos dispositivos que se almacenan.

## 2.1 Registro de Servicio

Un servicio es cualquier entidad que puede proporcionar información, realizar una acción, o controlar un recurso a favor de otra entidad. Un servicio se puede ejecutar como software, hardware, o una combinación de hardware y software. Toda la información acerca de un servicio que sea mantenido por un servidor SDP es contenido dentro de un solo registro de servicio. El registro de servicio consiste enteramente de una lista de atributos de servicio.

## 2.2 Atributo de Servicio

Cada atributo de servicio describe una sola característica de un servicio. Algunos ejemplos de atributos de servicio son *ServiceClassIDList & ProviderName.* Algunas definiciones del atributo son comunes a todos los registros de servicio, pero los proveedores de servicio pueden definir también sus propios atributos de servicio.

Un atributo de servicio consiste en dos componentes: un atributo ID de identificación y un valor del atributo.

- Un **attribute ID** es un entero sin signo de 16-bit que distingue cada atributo de servicio de otros atributo de servicio dentro de un registro de servicio. El attribute ID identifica también la semántica del valor asociado del atributo.
- El **attribute value** es un campo de longitud variable cuyo significado es determinado por el atributo de identificación se asociada a él y por la clase de servicio, del registro de servicio en el que se contiene el atributo. En el Service Discovery Protocol, un valor del atributo se representa como un elemento de datos.

## 2.3 Clase de Servicio

Cada servicio es un caso de una clase de servicio. La definición de clase de servicio proporciona las definiciones de todos los atributos contenidos en los registros de servicio que representan casos de esta clase. Cada definición del

atributo especifica el valor numérico del attribute ID, el uso previsto del valor del atributo, y del formato del valor del atributo. Un registro de servicio contiene los atributos que son específicos a una clase de servicio así como atributos universales que son comunes a todos los servicios.

A cada clase de servicio se asigna una identificación unica, en esta identificación de clase de servicio, se contiene el valor del atributo para el atributo de ServiceClassIDList, y es representada como un UUID. Un UUID es una identificación universal única que esta garantizada para ser unica a través de todo espacio y tiempo. UUIDs se puede crear independientemente en una manera distribuida. No se requiere ningún registro central de UUIDs asignado. Un UUID tiene un valor 128-bit.

## 3 Service Discovery

El punto total del SDP es permitir que los dispositivos de bluetooth descubran lo que otros dispositivos de bluetooth pueden ofrecer (servicios)**.** SDP permite esto en varios medios. El **Searching** significa buscar servicios específicos, mientras que **Browsing** significa mirar para ver qué servicios se están ofreciendo realmente

## 3.1 Searching for Services

La transacción **Searching for Services** permite a un cliente recuperar el registro de servicio encargado de registros particulares de servicio, basados en los valores de atributos contenidos dentro de éstos registros.

La capacidad de búsqueda de los registros de servicio son basados en los valores de atributos arbitrarios no se proporciona. Sino, que se proporciona la capacidad para buscar sólo los atributos cuyo valora son Identificaciones Universalmente Extraordinarias (UUIDs). Los atributos importantes de los servicios que se pueden utilizar para buscar un servicio son representados como UUIDs. El patrón de la búsqueda de servicio se utiliza para localizar el servicio deseado. Un patrón de búsqueda de servicio es una lista UUIDs (atributos de servicio) utilizado para localizar los registros del servicio.

## 3.2 Browsing For Services

Este proceso busca cualquier servicio ofrecido. En SDP, el mecanismo browsing para servicios esta basado en un atributo compartido por todas las clases de servicio. Este atributo se llama el atributo de **BrowseGroupList.** El valor de este atributo contiene una lista UUIDs. Cada UUID representa un grupo browse con que un servicio se puede asociar para el propósito de browsing.

Cuándo un cliente desea buscar un servicio del servidor SDP, crean un patrón de búsqueda del servicio que contiene el UUID que representa la raíz del grupo browse. Todos los servicios en los que se puede hacer el browsing en el nivel

superior se hacen miembros de la raíz del grupo browse teniendo la raíz UUID del grupo browse como valor dentro del atributo BrowseGroupList.

# 4 Representación de Datos

Como se menciono arriba, en el Service Discovery Protocol, un valor del atributo se representa como un elemento de datos. Un elemento de datos es una representación escrita en lenguaje máquina de datos. Consiste en dos campos: un campo header y un campo de datos.

## 4.1 Data Element header field

El campo header se compone de 2 partes, un **Type Descriptor** y un **Size Descriptor**.

- **Type Descriptor**: Un tipo de elementos de datos se representa como Type Descriptor de 5-bit. El Type Descriptor esta contenido en el bit más significativos de los 5 (high-order) del primer byte del data element header.
- **Size Descriptor**: El Size Descriptor es representado como un índice de 3 bits de tamaño, seguido por 0, 8, 16, o 32 bits. El índice del tamaño esta contenido en bit menos significativo (low-order) de 3 bits del primer byte del data element header.

## 4.2 Data Element Data Field

Los datos son una sucesión de los byte cuya longitud se especifica en el descriptor de tamaño y en cuyo significado es (parcialmente) especificado por el descriptor de tipo.

# 5 Service Discovery Background info

## 5.1 Service Discovery

Cuando computar continúa mover a un modelo red-céntrica, encontrando y para utilizar los servicios que pueden estar disponibles en la red llega a ser cada vez más importante. Los servicios pueden incluir imprimiendo, paging, FAX-ing, etcétera, así como varias clases del acceso de información tal como teleconferencia, puentes de red y puntos de acceso, facilidades de eCommerce, etcétera — la mayoría de cualquier tipo de servicio que un servidor o proveedor de Internet quizás ofrezcan.

Además de la necesidad de una manera estándar de descubrir los servicios disponibles, hay otras consideraciones: obtener el acceso a los servicios (encontrar y obtener los protocolos, conseguir los métodos de acceso, los "drivers" y otro códigos necesarios para utilizar el servicio), controlando el acceso a los servicios, anunciando los servicios, eligiendo entre servicios competentes,

facturación de servicios, etcétera. Este problema se reconoce extensamente; muchas compañías, estándares y consorcios lo dirigen en varios niveles de varias maneras. Service Location Protocol (SLP), Jini™, and Salutation ™, para denominar apenas unos pocos, todos tratan un cierto aspecto del descubrimiento del servicio.

## 5.2 Bluetooth Service Discovery

Las direcciones del descubrimiento de servicio de Service Discovery Protocol (SDP) Bluetooth son especificadas para el ambiente de Bluetooth. Se optimiza para la naturaleza altamente dinámica de las comunicaciones de Bluetooth. El SDP se centra sobre todo en descubrir los servicios disponibles o a través de los dispositivos de Bluetooth. SDP no define los métodos para conseguir acceso a los servicios; una vez que los servicios se descubran con el SDP, pueden ser alcanzados de varias maneras, dependiendo del servicio. Esto quizás incluya el uso de otro descubrimiento de servicio y mecanismos de acceso tales como los mencionado arriba; SDP proporciona un medio a otros protocolos para ser utilizados junto con el ambiente donde esto pueden ser factible. Mientras SDP puede coexistir con otros protocolos de descubrimiento de servicio, no los requiere. En ambientes Bluetooth, los servicios se pueden descubrir utilizando SDP y pueden ser accesados utilizando otros protocolos definidos por Bluetooth.

# 6 Perfil del Servicio de Descubrimiento (Aplicación)

Este perfil define los rasgos y procedimientos para una aplicación en un dispositivo Bluetooth descubrir servicios registrados en otros dispositivos Bluetooth y recuperar cualquiera información disponible deseada y pertinente a estos servicios.

Esencialmente, el perfil de Descubrimiento de Servicio define los protocolos y procedimientos que serán usados por una aplicación de descubrimiento de servicio en un dispositivo para localizar servicios en otros dispositivos Bluetooth-habilitados que usan el Protocolo de Descubrimiento de Servicio Bluetooth (SDP).

## 6.1 Descripción del perfil

### 6.1.1 Introducción

El perfil de Descubrimiento de Servicio define los protocolos y procedimientos que serán usados por una aplicación de descubrimiento de servicio en un dispositivo para localizar servicios en otros dispositivos Bluetooth-habilitados que usan el Protocolo de Descubrimiento de Servicio Bluetooth (SDP).

Con respecto a este perfil, la aplicación de descubrimiento de servicio es una aplicación usuario-iniciada específica. En este aspecto, este perfil está en contraste con otros perfiles donde repara interacciones del descubrimiento entre dos entidades de SDP en dos dispositivos Bluetooth-habilitados sea el resultado de la necesidad de habilitar un servicio de transporte particular (ej. RFCOMM, etc.), o un escenario particular de uso (traslado de archivo de ej., telefonía inalámbrica, LAN AP, etc.) encima de estos dos dispositivos. Las interacciones del descubrimiento de servicio de la última clase se pueden encontrar dentro de los documentos apropiados del perfil del escenario de uso Bluetooth.

El propósito principal de este perfil es describir el uso de las capas más bajas del protocolo Bluetooth (LC y LMP). Para describir las alternativas relacionadas en seguridad, también es incluida las capas más altas (L2CAP, RFCOMM y OBEX.).

### 6.1.2 Pila del Perfil

La aplicación del usuario del descubrimiento de servicio (SrvDscApp) en una interfase de dispositivo local (LocDev) con el cliente Bluetooth SDP para envíar inquiries de servicio y recibir inquiries de respuestas SDP de dispositivos remotos (RemDevs).

SDP usa la conexión-orientada (CO), el servicio de transporte en L2CAP, qué a su vez usa la baseband en conexiones asíncronas (ACL) los enlaces finalmente llevan el SDP PDUs sobre el aire. El descubrimiento de servicio es relacionado esencialmente a descubrir dispositivos, y a realizar inquiries y pages. Así, la interfase Srvd-scApp con la baseband vía BT_module_Cntrl instruye el módulo Bluetooth cuando entra en varios modos de búsqueda de la operación.

### 6.1.3 Configuraciones y funciones

Las siguientes funciones definen este perfil:

- **El dispositivo Local (LocDev)**

Dispositivo que comienza el procedimiento del descubrimiento de servicio . Un LocDev debe contener por lo menos la parte del cliente de la arquitectura SDP Bluetooth. Un LocDev contiene la aplicación del descubrimiento de servicio (SrvDscApp) usado por un usuario al comenzar a descubrir y desplegar los resultados de estos.

- **Device Remoto(s) (RemDev(s)):**

Es cualquier dispositivo que participa en el proceso del servicio de descubrimiento respondiendo al servicio inquiries generado por un LocDev. Un RemDev debe por lo menos contener la parte del servidor de la arquitectura SDP Bluetooth. Un RemDev contiene un servicio de base de datos la cual consulta el servidor SDP para crear respuestas a la solicitud del descubrimiento de servicio

El LocDev o RemDev asignados a un dispositivo no son permanentes ni exclusivos. Un RemDev también puede tener un SrvDscApp instalado en él así como un cliente SDP, y un LocDev puede tener un servidor SDP. En conjunto con cada dispositivo tienen un SrvDscApp instalado, un SDP-cliente, y un SDP-servidor, la asignación de dispositivos a los roles anteriores es relativo a cada transacción SDP individual (y relacionado). Así, un dispositivo podría ser un LocDev para una transacción de SDP particular, mientras en el mismo momento sea un RemDev para otra transacción SDP.

### 6.1.4 Requisitos y escenarios de usuario.

Los escenarios cubiertos por este perfil son lo siguiente:

- Search para los servicios por clase de servicio,
- Search para los servicios por atributos de servicio, y
- Service browsing.

Los dos primeros casos representan servicios específicos de búsqueda de conocidos y como parte de la pregunta de usuario "¿Es un servicio A, o es el servicio A con características B y C, disponible?" El último caso representa una búsqueda de servicio general que es una respuesta a la pregunta de usuario "¿Qué servicios están disponibles?".

El usuario Bluetooth debe en principio poder conectar un dispositivo Bluetooth a cualquier otro dispositivo Bluetooth. Aun cuando los dos dispositivos conectados no comparten ninguna aplicación en común, debe ser posible al usuario encontrar esto fuera de usar las capacidades básicas Bluetooth.

### 6.1.5 Principios del perfil

Antes que cualquiera de los dos dispositivos Bluetooth-provisto pueda comunicarse entre sí se puede necesitar lo siguiente:

- Los dispositivos necesitan estar accionados e inicializados. La inicialización puede requerir y proporcionar un PIN para la creación de una llave de enlace, para la autorización del dispositivo y la encriptación de datos.
- Tiene que ser creada una conexión Bluetooth, la cual puede requerir el descubrimiento de otro dispositivo BD_ADDR vía un proceso inquiry y paging de otro dispositivo.

Mientras pueda parecer natural para considerar un LocDev que funcione como Bluetooth master y RemDev(s) como slave(s) Bluetooth, no hay ningún requisito tal impuesto en los dispositivos que participan en este perfil. El descubrimiento de servicio como presenta en este documento puede ser iniciado por el master o un dispositivo del slave en cualquier punto para que estos dispositivos sean miembros del mismo piconet. También, un slave en un piconet puede comenzar el

descubrimiento de servicio posiblemente en un nuevo piconet, de tal manera al master del piconet original que será indisponible (posiblemente entrando en el modo operacional hold) para una cantidad dada de tiempo.

### 6.1.6 Conformidad

Si la conformidad a este perfil se exige, toda la capacidad indicada obligatoria para este perfil se apoyará de la manera especificada (proceso-obligatorio). Esto también aplica a todas las capacidades optativas y condicionales para las que el soporte se indica.

## 6.2 Aspectos de la Interfase de usuario

### 6.2.1 Pairing

Ningún requisito particular con respecto a pairing es impuesto por este perfil. Pairing puede o no realizarse. Siempre que un LocDev realice un descubrimiento de servicio todavía como "desconectado" RemDev(s), será la responsabilidad del SrvDscApp permitir pairing antes de la conexión, o para evitar que cualquier dispositivo pueda requerir pairing primero. Este perfil se enfoca solo en realizar servicios de descubrimiento siempre que el LocDev pueda establecer un enlace baseband legítimo y útil con RemDev(s).

### 6.2.2 Selección de modo

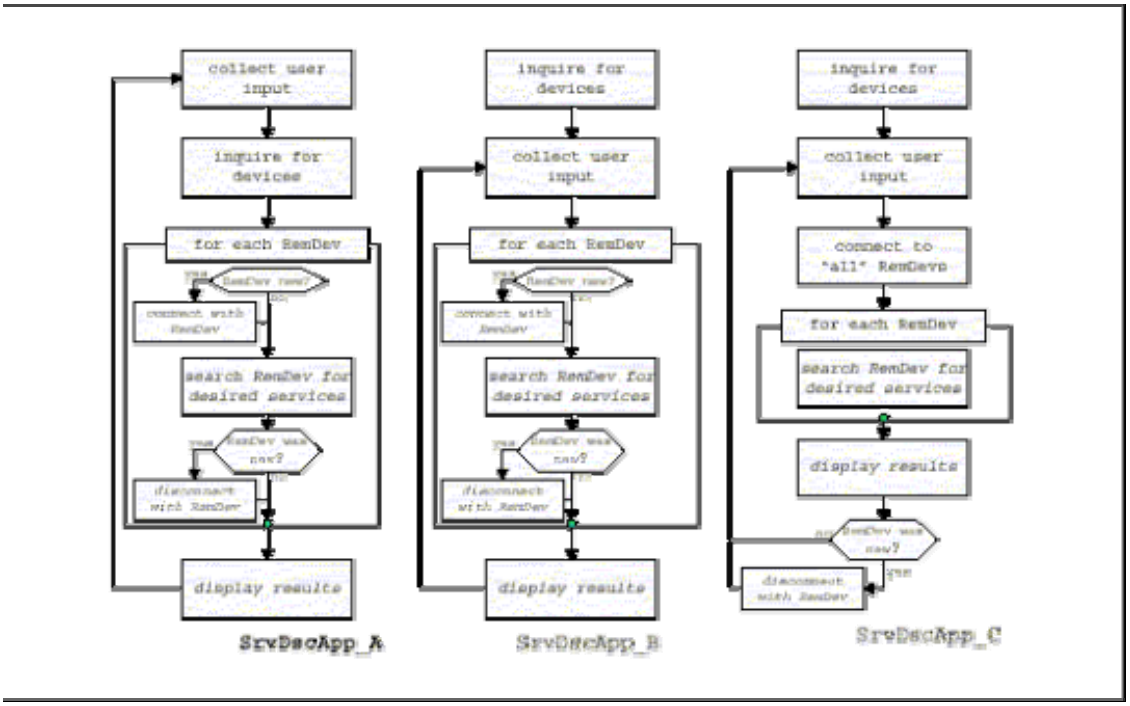
Este perfil asume que, bajo la guía de SrvDscApp, el LocDev podrá entrar en estado inquiry y/o en paging. A su vez un RemDev con servicios que quiere hacer disponible a otros dispositivos (por ejemplo una impresora una LAN DAP, un gateway PSTN, etc.) será capaz de entrar en los estados inquiry scan y page scan.

Puesto que el SrvDscApp puede también realizar inquiry contra el ya conectado RemDevs, no es obligatorio según el perfil, siempre un LocDev será el master de la conexión con un RemDev. De igual forma, un RemDev no siempre será el slave de una conexión con un LocDev.

## 6.3 Capa de Aplicación

### 6.3.1 Aplicación Descubrimiento de Servicio

En esta subdivisión, se presenta la estructura operacional del SrvDscApp, la figura muestras las posibles alternativas para un SrvDscApp.




Las alternativas de SrvDscApp mostradas arriba, (las cuales no son exhaustivas por cualquier medios), logran los mismos objetivos pero con caminos diferentes para ello. En la primera alternativa (SrvDscApp_A), el SrvDscApp en un LocDev pregunta a su usuario para que le proporcione información para la búsqueda de servicio deseado.
Siguiendo esto, el SrvDscApp busca dispositivos, vía procedimiento inquiry Bluetooth. Para cada dispositivo encontrado, el LocDev se conectará a él, realizará cualquier estructuración del enlace necesaria, y le preguntara por los servicios deseados. En la segunda alternativa (SrvDscApp_ B), el inquiry de dispositivos se hace antes de collecting user input para la búsqueda del servicio.

En las primeras dos alternativas, page, la creación de enlace, y el descubrimiento de servicio se hace secuencialmente en la base de RemDev; es decir, el LocDev no compagina a cualquiera RemDev nuevo antes de completar la búsqueda de servicio con un RemDev anteriores y (si necesario) se desconectando de él. En la última alternativa (SrvDscApp_C), el LocDev, bajo el control del SrvDscApp, compaginará primero todos los RemDev, entonces creará enlaces con todos estos dispositivos (hasta un máximo de 7 a la vez), y entonces preguntara a todos los dispositivos conectados por los servicios deseados.

Cuando un LocDev realiza una búsqueda de descubrimiento de servicio, esto hace con tres tipos diferentes de RemDevs:

**1. Dispositivos de confianza:**
Éstos son dispositivos con que actualmente no es conectado el LocDev pero el dispositivo LocDev ya tiene una relación confiada establecida.

**2. Dispositivos desconocido (nuevos):**
Éstos son dispositivos de incertidumbre que no se conectan actualmente con el LocDev.

**3. Dispositivos conectados:**
Éstos son dispositivos que ya se conectan al LocDev.

Para descubrir el tipo 1 o 2 RemDevs, el SrvDscApp necesita activar los procesos inquiry y/o page Bluetooth. Para el tipo 3 RemDevs, se necesitan los últimos procesos. Para realizar su tarea, SrvDscApp necesita tener acceso al BD_ADDR de los dispositivos en la inmediación de un LocDev, no importa si estos dispositivos se han localizado vía proceso inquiry Bluetooth o ya se han conectado al LocDev. Así, BT_module_Cntr en un LocDev mantendrá la lista de dispositivos en la inmediación del LocDev y esta lista será útil al SrvDscApp.

### 6.3.2 Service Primitivos Abstraction (Servicios de Abstracción Primitivos)

Esta sección describe brevemente la funcionalidad de un SrvDscApp. Esta funcionalidad se presenta en la forma de servicios de abstracciones primitivas que mantienen una estructura formal describiendo las expectativas del usuario de un SrvDscApp. Se asume que el stack Bluetooth fundamental puede conocer los objetivos de éstos servicios de abstracciones primitivas directamente o indirectamente. La sintaxis exacta y semántica del servicio de las abstracciones primitivas (o simplemente "primitivos de servicio") puede ser la plataforma-dependiente (ej. un sistema operativo, una plataforma de hardware, como un PDA, una computadora del cuaderno, un teléfono celular, etc.) y está más allá del alcance de este perfil. Sin embargo, se espera que la funcionalidad de estos primitivos esté disponible al SrvDscApp para alcanzar su tarea.

La siguiente tabla contiene un conjunto mínimo de servicios primitivos que soportan a SrvDscApp. Los Primitivos de bajo nivel bajo-nivelado como el openSearch (.) o closeSearch (.) no se muestra y se asume que son parte de la aplicación primitiva mostrada siempre que es necesario. Las diferentes aplicaciones del stack Bluetooth debe por (mínimo) habilitar las funciones que estos servicios primitivos proporcionan.

Por ejemplo, el serviceSearch de servicio, los permisos primitivos, los funcionamientos idénticos múltiples son manejados en seguida. Una aplicación de la pila que exige a una aplicación lograr esta función por iterating a través del uno-a-un-tiempo de los funcionamientos idénticos múltiples será considerada como habilitar la función de este servicio primitivo. Adicionalmente los servicios primitivos pueden preverse relacionando a las operaciones puramente locales como el registro de servicio, pero estos, están fuera del alcance de este perfil

| | |
|---|---|
| **serviceBrowse** | Búsqueda de servicios, que pertenece a la lista de servicios browseGroup en los dispositivos de la lista RemDevs; la búsqueda puede calificarse aun mas con una lista de parámetros de RemDev Relation. |
| **serviceSearch** | Una búsqueda si los dispositivos se listaron en la lista de servicios de soporte RemDevs de la lista solicitada de servicios; cada servicio debe tener un modelo de búsqueda de servicio que es un que es un súper conjunto de searchPattern; para cada servicio se recuperan los valores de los atributos contenidos en el attributeLis correspondiente. |
| **RemDev** | Una búsqueda para RemDev en la zona de un LocDev. |
| **terminatePrimitive** | una terminación de acciones ejecutadas como resultado de invocar los servicios primitivos identificados por el primitiveHandle |

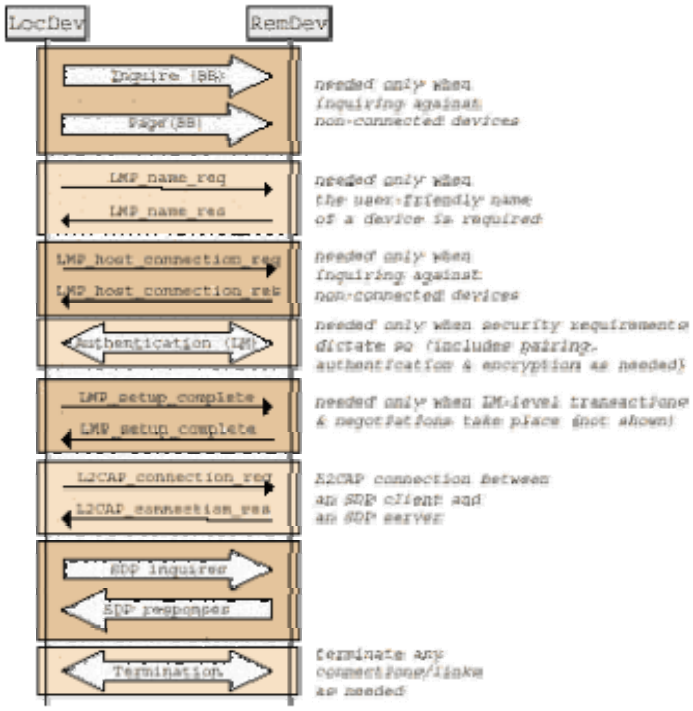
.

Los servicios primitivos anteriores regresan la información solicitada, siempre que es encontrada. Basado en esta forma éstos servicios primitivos son soportados por aplicaciones stack Bluetooth, los resultados de una búsqueda pueden volver directamente por la función correspondiente, o por un indicador en una estructura de datos con toda la información pertinente. Alternativamente, una aplicación snack Bluetooth puede tener medios diferentes para proporcionar los resultados de una búsqueda.

### 6.3.3 Mapas de Sucesión de Mensaje

Este perfil se preocupa por tres procedimientos Bluetooth distintos. Descubrimiento del dispositivo, descubrimiento de nombre de dispositivo, descubrimiento de servicio. Note que cada uno de estos procedimientos no evita cualquier otro; ej. Para conectar a un RemDev, con un LocDev puede tener que descubrirlo primero, y también puede pedir su nombre.

La figura resume el intercambio del mensaje clave “fases” encontrado durante la ejecución de este perfil. No todos los procedimientos están presentes en todo momento, y no siempre todos los dispositivos necesitan pasar por estos procedimientos. Por ejemplo, si la autenticación no se requiere, la fase de autenticación en la figura no se ejecutará. Si el SrvDsvApp necesita inquirir para los servicios específicos en un RemDev con los que se conecta actualmente el LocDev, no pueden ejecutarse inquiry y page. En la figura, las condiciones bajo, que las fases particulares se ejecuten o no también, son proporcionadas.

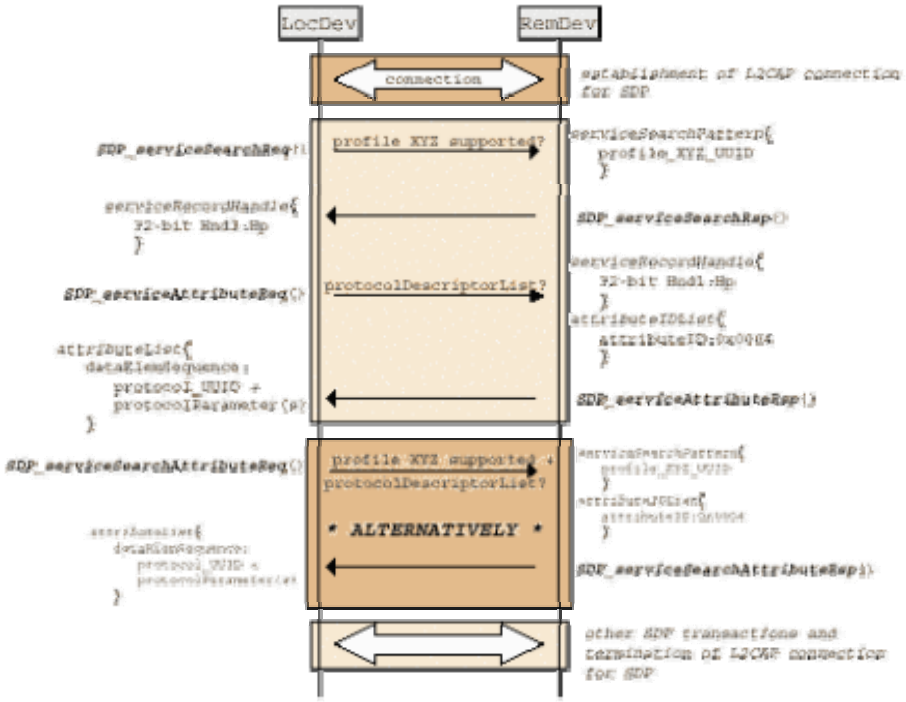



## 6.4 Descubrimiento de Servicio

La aplicación de descubrimiento de servicio no hace uso de SDP como un medio de acceso a un servicio, sino como un medio para informar al usuario de un LocDev sobre los servicios que están disponible del dispositivo (y posiblemente vía) RemDev(s). Aplicaciones BT-conscientes que corren en un dispositivo local también pueden usar los procedimientos descritos en esto y las secciones siguientes para recuperar cualquier información pertinente que facilitará la aplicación accediendo un servicio deseado en un dispositivo remoto.

### 6.4.1 Un Ejemplo del Intercambio de SDP PDU

La figura muestra dos ejemplos de intercambios de PDU SDP. En particular, muestra que PDU intercambian secuencias de inquiry y recuperación cualquier información pertinente a un perfil de Bluetooth particular.




Se muestran dos alternativas utilizando diferentes SDP PDUs para recuperar la misma información final - los protocolos DescriptorList atribuido de dispositivos que soportan un perfil Bluetooth específico. Con la primera alternativa, la información deseada se derivada en dos pasos.

1. El LocDev envía un SDP_serviceSearchReq PDU que contiene un modelo de búsqueda de servicio compuesto del UUID asociado al perfil deseado. El perfil deseado (perfil “XYZ”) es identificado por su UUID, denotado en la figura como “el profile_XYZ_UUID”. En su repuesta PDU, el servidor de SDP devuelve uno o más de los 32-bit de registro de servicio cuyos archivos de servicio correspondientes contienen el”profile_XYZ_UUID' UUID”. En la figura, sólo se muestra, denotado como “el prHndl”.

2. El LocDev entonces entra en prHndl en un SDP_serviceAttribute PDU junto con uno o más atributos IDs. En este ejemplo, el atributo de interés está que el protocolDescriptorList cuyo atributo ID es 0x0004. El servidor SDP entonces, en su respuesta, ingresa a la lista de protocolo requerida.

En el evento que ningún registro de servicio contiene el modelo de búsqueda de servicio, se encuentra en el servidor SDP, el SDP_serviceSearchResp PDU contendrá un serviceRecordHandleList vacío y un parámetro del totalServiceRecordCount puesto en su valor mínimo. Si los atributos deseados no existen en el servidor de SDP, el SDP_serviceAttributeResp PDU contendrá un attributeList vacío y un parámetro del attributeListByteCount puestos a su valor mínimo.

Con la segunda alternativa, los atributos deseados se recuperan en un paso:

1. El LocDev envía un SDP_serviceSearchAttributeReq PDU donde ambos son incluidos (modelo de búsqueda de servicio: profile_XYZ_UUID) y el atributo(s) deseado se proporcionan (atributo ID: 0x0004). En su repuesta el servidor SDP proporcionará el atributo(s) solicitado del registro de servicio esto empareja el modelo de búsqueda de servicio.

En caso de que ningún registro de servicio que contiene el modelo de búsqueda de servicio deseado y/o el atributo deseado se encuentra en el servidor SDP, el SDP_serviceSearchAttributeResp PDU que contendrá un attributeLists vacío y un parámetro del attributeListsByteCount puesto en su valor mínimo.

## 6.5 L2CAP

### 6.5.1 Tipos de Canal

En este perfil, se usarán solo canales conexión-orientada. En particular, ninguna transmisión L2CAP será usada para este perfil.

### 6.5.2 Señalización

Para el propósito de recuperar información de SDP sólo un LocDev puede iniciar un pedido de la conexión L2CAP y publicar un pedido de la conexión L2CAP PDU

Con el propósito de recuperar información SDP-related, sólo un LocDev puede comenzar una conexión L2CAP y puede emitir una L2CAP demanda de conexión PDU; (aunque hay excepciones). Igualmente con las terminaciones correspondientes de conexión L2CAP. De otra manera que SDAP no impone ninguna restricción ni requisitos adicionales en la señalización de L2CAP.

En el campo PSM del paquete de Connection Request, el valor 0x0001 (vea sección “Signalling” de la capa L2CAP) se usará para indicar pedido para la creación de una conexión L2CAP para acceder a la capa de SDP.

### 6.5.3 Opciones de Configuración.

Esta sección describe el uso de las opciones de configuración en el perfil de descubrimiento de servicio.

#### 6.5.3.1 Unidad de Transmisión de máximo (MTU)

Para el uso eficaz de los recursos de comunicación, el MTU se seleccionará tan grande como sea posible, mientras respetando cualquier constreñimiento físico impuesto por los dispositivos involucrados, y la necesidad que estos dispositivos continúan cumpliendo cualquier contrato ya concordado sobre QoS con otros

dispositivos y/o aplicaciones. Se espera que durante la vida de una conexión L2CAP para las transacciones de SDP (también llamado la “el sesión” de SDP) entre dos dispositivos, cualquiera de estos dispositivos puede comprometerse en una conexión L2CAP con otro dispositivo y/o aplicación.

Si esta nueva conexión tiene “non-default” requisitos de QoS, el MTU para la sesión referida de SDP permite ser renegociado durante la vida de esta sesión de SDP, para acomodar las limitaciones de QoS de la nueva conexión L2CAP

### 6.5.3.2 Flush Time-out

Las transacciones SDP se llevan encima de un L2CAP cauce fiable. El rubor tiempo-fuera valor se pondrá a su valor 0xFFFF predefinido. Las transacciones de SDP se llevan sobre un canal seguro L2CAP. El valor Flush Time-out será puesto a un valor predefinido 0xFFFF.

### 6.5.3.3 Calidad de Servicio (Quality o Service)

El uso de Quality o Service (QoS) y la negociación de QoS son optativos. Si QoS es negociado, las escenas predefinidas serán usadas. En particular, se tratará tráfico de SDP como un servicio best-effort tipo de tráfico.

## 6.5.4 Transacciones SDP y Vida de Conexión L2CAP

Mientras, en general, las transacciones de SDP comprenden una sucesión de demanda-y-respuesta de servicio que PDU intercambia, el propio SDP constituye un servicio de datagrama sin conexión, en que ninguna conexión SDP-level se forma antes de cualquier SDP en el intercambio de PDU. SDP delega la creación de conexiones en su nombre a la capa L2CAP. Es así la responsabilidad de SDP o, más correctamente, de la capa de SDP – solicitar a la capa L2CAP “Derribar” estas conexiones en su beneficio también.

Desde que los servidores de SDP son considerados sin estado, “derribando” una conexión L2CAP después que una demanda de servicio PDU se envía (como un verdadero servicio sin conexión puede implicar) será perjudicial a la transacción de SDP. Es más, la pena significativa del desempeño se tendrá que pagar si, para cada transmisión de SDP PDU, una nueva conexión L2CAP deberá ser creada. Así, las conexiones L2CAP para transacciones de SDP durarán más que la transmisión de un solo SDP PDU.

Una sesión de SDP entre un cliente SDP y un servidor SDP representa el intervalo de tiempo que el cliente y el servidor tiene la misma conexión L2CAP continuamente presente. Una mínima transacción de SDP representará un solo intercambio de una demanda de transmisión SDP PDU de un cliente SDP a un servidor de SDP, y la transmisión de una respuesta correspondiente de SDP PDU del servidor de SDP al cliente de SDP. Con respecto a este perfil, bajo las

condiciones operacionales normales, la duración mínima de una sesión de SDP será la duración de una transacción SDP mínima.

Una sesión de SDP puede durar menos que el mínimo requirió en caso de errores irrecuperable (procesados o ligdos) en capas debajo de SDP en el LocDev y RemDev, o en la capa de SDP y el servicio registra la base de datos en el RemDev. Una sesión de SDP puede ser interrumpida también por la intervención del usuario que puede terminar la sesión de SDP antes de la terminación de una transacción SDP.

## 6.6 Link Manager

### 6.6.1 Vista General de Capacidad.

En esta sección se muestra que las características de LMP son obligatorias para soportar, al este perfil del descubrimiento de servicio, que es opcional y que se excluye. La razón para excluir las características es que estos pueden degradar la operación de dispositivos en el caso de uso. Por lo tanto, estas características nunca serán activadas por una unidad activa en este caso.

El tráfico generado durante las interacciones de descubrimiento de servicio no tiene ningún requisito particular de QoS. Como tal, ninguna provisión particular del enlace Bluetooth se exige para apoyar este perfil.

### 6.6.2 Comportamiento de Error

Si una unidad trata de utilizar una característica obligatoria, y las otras unidades de respuesta no son soportadas, la unidad que inicia mandará un PDU LMP_detach "la característica no apoyada de LMP."

Si una unidad intenta usar un rasgo obligatorio, y las otras respuestas de la unidad que no son soportadas, la unidad comenzara enviando LMP_detach PDU con “Sin ofrecer apoyos LMP."

Una unidad siempre podrá manejar el rechazo de la demanda para un rasgo optativo.

### 6.6.3 Link Policy

No hay ningún rol master–slave fijo para la ejecución de este perfil.
Este perfil no indica ningún requisito de cuál modo de baja potencia utilizar, ni cuando utilizarlo. Depende del Link Manager de cada dispositivo decidir y solicitar características especiales de la conexión

## 6.7 Link Control

Para las próximas cuatro subdivisiones, se asume que un LocDev deberá realizar las búsquedas del servicio con RemDevs originalmente desconectado. Así necesita inquiry y page (o sólo page) para estos RemDevs. Ninguno de las siguientes cuatro subdivisiones aplican siempre que un LocDev realiza las búsquedas de servicio con el que RemDevs ya se conecta.

### 6.7.1 Inquiry

Siempre instruido por el SrvDscApp, el LocDev informara a su baseband a entrar en el estado inquiry. La entrada en este estado puede o no puede ser inmediato, sin embargo, dependiendo de los requisitos de QoS de cualquiera conexión que ya exista y continué.

El usuario del SrvDscApp podrá poner el criterio para la duración de un inquiry. No obstante, el tiempo de la residencia real en el estado inquiry debe obedecer la recomendación cedida en la sección Baseband Bluetooth. Cuando se invoca inquiry en un LocDev, el procedimiento de inquiry general se utilizará utilizando un GIAC.

### 6.7.2 Inquiry Scan

Los dispositivos que operan en un modo descubrible de funcionamiento, podría ser descubierto por un inquiry enviado por otros dispositivos. Para ser descubierto por un inquirí resultado de una acción SrvDscApp, un RemDev debe entrar en inquiry scan usando el GIAC;

### 6.7.3 Paging

Siempre que el SrvDscApp necesita conectar a un RemDev específico para preguntar acerca de sus registros de servicio, el LocDev aconsejará a su baseband entrar en el estado paging. La entrada en este estado puede o no ser inmediato, sin embargo, dependiendo de los requisitos de QoS de cualquiera conexión que ya exista y continué

Dependiendo de la clase paging (R0, R1, o R2) indicado por un dispositivo de RemDev, el LocDev por consiguiente estará en paging.

### 6.7.4 Page Scan

Los dispositivos que operan en un modo conectado de operación, podrían establecer las conexiones Bluetooth con otros dispositivos de pages enviados por estos otros dispositivos. Para establecer una conexión con un RemDev, un

LocDev debe envíar un page que resulta de una acción SrvDscApp que utiliza el RemDev de 48 bits BD_ADDR.

### 6.7.5 Error Behaviour

Puesto que la mayoría de las características en el nivel de LC tienen que ser activadas por procedimientos LMP, los errores se congelaran generalmente en esa capa. Sin embargo, hay algunos procedimientos de LC que son independientes de la capa de LMP, tal como inquiry y paging. El mal uso de cosas así ofrece es difícil o a veces imposible. Hay que ningún mecanismo definido para descubrir o prevenir tal uso impropio.

# PARTE C
# GENERIC ACCESS PROFILE

## 1 Descripción del Perfil

### 1.1 Pila del perfil

El propósito principal de este perfil es describir el uso de las capas más bajas del protocolo Bluetooth (LC y LMP). Describir la seguridad de alternativas relacionadas, también incluidas en las capas más altas (L2CAP, RFCOMM y OBEX).

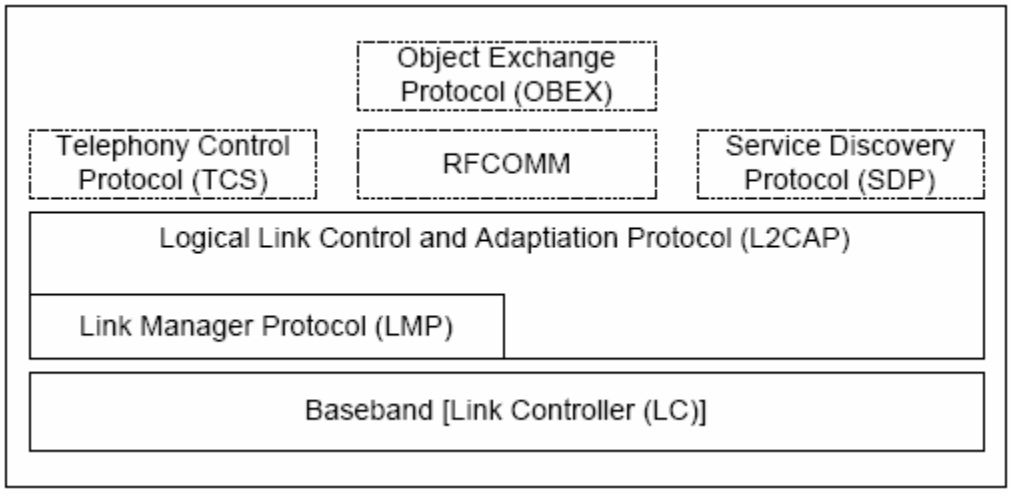



*Figure 2.1: Pila del perfil cubierto por este perfil.*

### 1.2 Configuración y Función

Para las descripciones en este perfil de las funciones que puede tomar los dos dispositivos involucrados en una comunicación Bluetooth, se usara la notación genérica de A-party (el dispositivo ***paging*** en caso del establecimiento del enlace, o ***iniciador*** en caso de otro procedimiento en un enlace establecido) y el B-party (dispositivo paging o aceptador). El A-party es el procedimiento dado para comenzar el establecimiento del enlace físico o comenzar una transferencia en un enlace existente.

El iniciador y el aceptador generalmente operan los procedimientos genéricos según el perfil al que se refieren.

## 1.3 Requerimientos y Escenarios del Usuario

El usuario Bluetooth debe como principio poder conectar un dispositivo Bluetooth a cualquier otro dispositivo Bluetooth. Aun cuando los dos dispositivos conectados no compartan ninguna aplicación común, debe ser posible para el usuario aplicar esto, fuera de utilizar las capacidades básicas Bluetooth.

## 1.4 Principios del Perfil

Este perfil:

- Requerimientos de estado en nombres, valores y esquemas de codificación usados para los nombres de los parámetros y procedimientos experimentados en el nivel interfase de usuario.

- Define los modos de funcionamiento que no son servicios - o perfil-específico, pero genérico a todos los perfiles.

- Define los procedimientos generales que pueden usarse para descubrir las identidades, nombres y capacidades básicas de otros dispositivos Bluetooth que están en un modo donde ellos pueden ser descubiertos. Sólo los procedimientos donde ningún canal o conexión establecida es usada y se especifica.

- Define el procedimiento general para crear un enlace entre los dispositivos Bluetooth.

- Describes los procedimientos generales que pueden usarse para establecer las conexiones a otros dispositivos Bluetooth.

## 1.5 Conformidad

Dispositivos Bluetooth que no conforman cualquier otro perfil Bluetooth conformarán este perfil para asegurar interoperabilidad básica y co-existencia.

# 2 Aspectos de Interface de Usuario

El perfil de acceso general especifica los términos genéricos que deben usarse en el nivel de interfase de usuario.

## 2.1 Representación de Parámetro Bluetooth

**Bluetooth Device Address (BD_ADDR):**

El BD_ADDR es una dirección de 48 bits (12 caracteres hexadecimales) de un dispositivo Bluetooth usada para que identifique un dispositivo remoto durante el procedimiento de descubrimiento de dispositivo.

**Bluetooth Device Name (User Friendly Name):**

El dispositivo nombre de Bluetooth es una serie del carácter devueltos en LMP_name_res como una respuesta a un LMP_name_req. Puede depender de 248 caracteres, aunque no debe asumirse que un dispositivo remoto puede manejar más de los primeros 40 caracteres.

**Bluetooth Pass-Key (PIN Number):**

El PIN Bluetooth se usa para autenticar dos dispositivos Bluetooth (esto no antes de cambiar keys de enlace) el uno al otro y crear una relación confiada entre ellos. Esto incluye guardar una llave de enlace común para la futura autentificación.

**Bluetooth Class of Device (Device Type)**

Class of Device es un parámetro recibido durante el procedimiento de descubrimiento de dispositivo, indicando el tipo de dispositivo y qué tipos de servicios soporta.

## 2.2 Pairing

Se definen dos procedimientos que hacen el uso del procedimiento pairing definidos en el nivel LMP.

1. El usuario comienza el procedimiento de conexión y registra el passkey con el propósito explícito de crear la conexión entre dos dispositivos Bluetooth, o

2. El usuario pide introducir el passkey durante el establecimiento, desde el procedimiento los dispositivos de antemano no comparten una llave de enlace común.

Se dice que el usuario realiza en el primer caso, conexión (registrando el passkey) y en el segundo caso el usuario autentifica usando el passkey.

# 3 Modos

## 3.1 Modos de Detección (Discoverability Modes)

Con respecto a inquiry, un dispositivo Bluetooth estará en modo **non-discoverable** o en un modo **discoverable** (descubrible). (El dispositivo estará dado en uno, y sólo en un modo discoverability.)

| Procedure | Ref. | Support |
|---|---|---|
| Discoverability modes: | 4.1 | |
| Non-discoverable mode | | C1 |
| Limited discoverable mode | | O |
| General discoverable mode | | O |
| Connectability modes: | 4.1.3.3 | |
| Non-connectable mode | | O |
| Connectable mode | | M |
| Pairing modes: | 4.2.2.2 | |
| Non-pairable mode | | O |
| Pairable mode | | C2 |
| C1: If limited discoverable mode is supported, non-discoverable mode is mandatory, otherwise optional. | | |
| C2: If the bonding procedure is supported, support for pairable mode is mandatory, otherwise optional. | | |

Tabla 3 Requisitos Conformance relacionados a los modos definidos en esta sección

Existen dos modos discoverable y son definidos y llamados aquí como el **Limited discoverable Mode**(modo de detección limitado) y el **General discoverable Mode** (modo de detección general)

## Non-discoverable Mode:

- Cuando un dispositivo Bluetooth está en este modo nunca entrará en el estado de INQUIRY_RESPONSE.
- El dispositivo es no-detectable.

## Limited discoverable Mode:

- Este modo debe ser usado por dispositivos que necesitan sólo ser descubiertos por un periodo limitado, durante condiciones temporales o para un evento específico. El propósito es responder a un dispositivo que hace un inquiry limitado.

- El dispositivo es detectable.

## General discoverable Mode:

- El modo General discoverable usado por dispositivos que necesitan ser continuamente detectados o en ninguna condición específica. El propósito es responder a un dispositivo que hace un inquiry general (inquiry que usa el GIAC).
- El dispositivo es detectable.

## 3.2 Modos de Conectibilidad

Con respecto a paging, un dispositivo Bluetooth estará en modo **no-connectable** o en modo **connectable**. Cuando un dispositivo Bluetooth esta en el modo no-connectable no responde a paging. Y cuando un dispositivo Bluetooth está en modo connectable este responde a paging.

**Modo Non- connectable:**

- Cuando un dispositivo Bluetooth está en este modo nunca entrará en el estado de PAGE_SCAN.

- El dispositivo esta no-conectado.

**Modo Connectable:**

- Cuando un dispositivo Bluetooth está en modo connectable entrará en el estado de PAGE_SCAN periódicamente.

- El dispositivo esta conectado.

| Scenario | Page Scan Interval | Page Scan Window | Scan Type |
|---|---|---|---|
| R0 (1.28s) | $T_{GAP}(107)$ | $T_{GAP}(107)$ | Normal scan |
| Fast R1 (100ms) | $T_{GAP}(106)$ | $T_{GAP}(101)$ | Interlaced scan |
| Medium R1 (1.28s) | $T_{GAP}(107)$ | $T_{GAP}(101)$ | Interlaced scan |
| Slow R1 (1.28s) | $T_{GAP}(107)$ | $T_{GAP}(101)$ | Normal scan |
| Fast R2 (2.56s) | $T_{GAP}(108)$ | $T_{GAP}(101)$ | Interlaced scan |
| Slow R2 (2.56s) | $T_{GAP}(108)$ | $T_{GAP}(101)$ | Normal scan |

Tabla 3.2 Los parámetros page scan para escenarios de velocidad de conexión

## 3.3 Modo Pairing

Con respecto a pairing, un dispositivo Bluetooth estará en el modo no-pairable o en pairable. En el modo pairable el dispositivo Bluetooth acepta pairing - es decir la creación de conexiones - comenzando por el dispositivo remoto, y en el modo no-pairable modo no lo hace.

**Modo Non-pairable:**

- Cuando un dispositivo Bluetooth esta en el modo no-pairable responderá a un LMP_in_rand recibido con LMP_not_accepted con el razonamiento pairing not allowed.

- El Dispositivo esta “no-conectado” o en “modo no-conectado” o “no acepta conexiones.”

**Modo Pairable**

- Cuando un dispositivo Bluetooth está en este modo responderá a un LMP_in_rand recibido con LMP_accepted (o con LMP_in_rand si tiene un PIN fijo).

- El Dispositivo esta “conectado” o en “modo conectado” o “o acepta una conexión".

## 4 Aspectos de seguridad

La seguridad se garantiza en el perfil de acceso general por dos métodos, un proceso de la autenticación y una elección de modos de seguridad.

| | Procedure | Ref. | Support |
|---|---|---|---|
| 1 | Authentication | 5.1 | C1 |
| 2 | Security modes | 5.2 | |
| | Security mode 1 | | O |
| | Security mode 2 | | C2 |
| | Security mode 3 | | C2 |
| C1: If security mode 1 is the only security mode that is supported, support for authentication is optional, otherwise mandatory. (Note: support for LMP-authentication and LMP-pairing is mandatory according [2] independent of which security mode that is used.) | | | |
| C2: If secure communication is supported, then support for at least one of Security mode 2 or Security mode 3 is mandatory. | | | |

Tabla : Conformance requirements related to the generic authentication procedure and the security modes defined in this section

## 4.1 Autenticación

El procedimiento de autenticación genenarl describe cómo se usan los procedimientos LMP-autenticación y LMP-pairing, cuando inicia la autenticación a través de un dispositivo Bluetooth hacia otro, dependiendo si una llave de enlace existe o no y si pairing se permite o no.
Nota: El dispositivo que comienza la autenticación tiene que estar en seguridad modo 2 o en seguridad modo 3.

## 4.2 Modos de seguridad

Seguridad Modo 1

Cuando un dispositivo Bluetooth está en seguridad modo 1 nunca iniciara algún procedimiento de seguridad (es decir, nunca enviará LMP_au_rand, LMP_in_rand o LMP_encryption_mode_req).

Seguridad Modo 2

Cuando un dispositivo Bluetooth está en seguridad modo 2 no comenzará ningún procedimiento de seguridad antes de recibir una solicitud de establecimiento de canal (L2CAP_ConnectReq) o que este procedimiento allá comenzado por sí mismo.

Seguridad Modo 3
Cuando un dispositivo Bluetooth está en seguridad modo 3 comenzará el procedimiento de seguridad antes del envío de LMP_link_setup_complete.

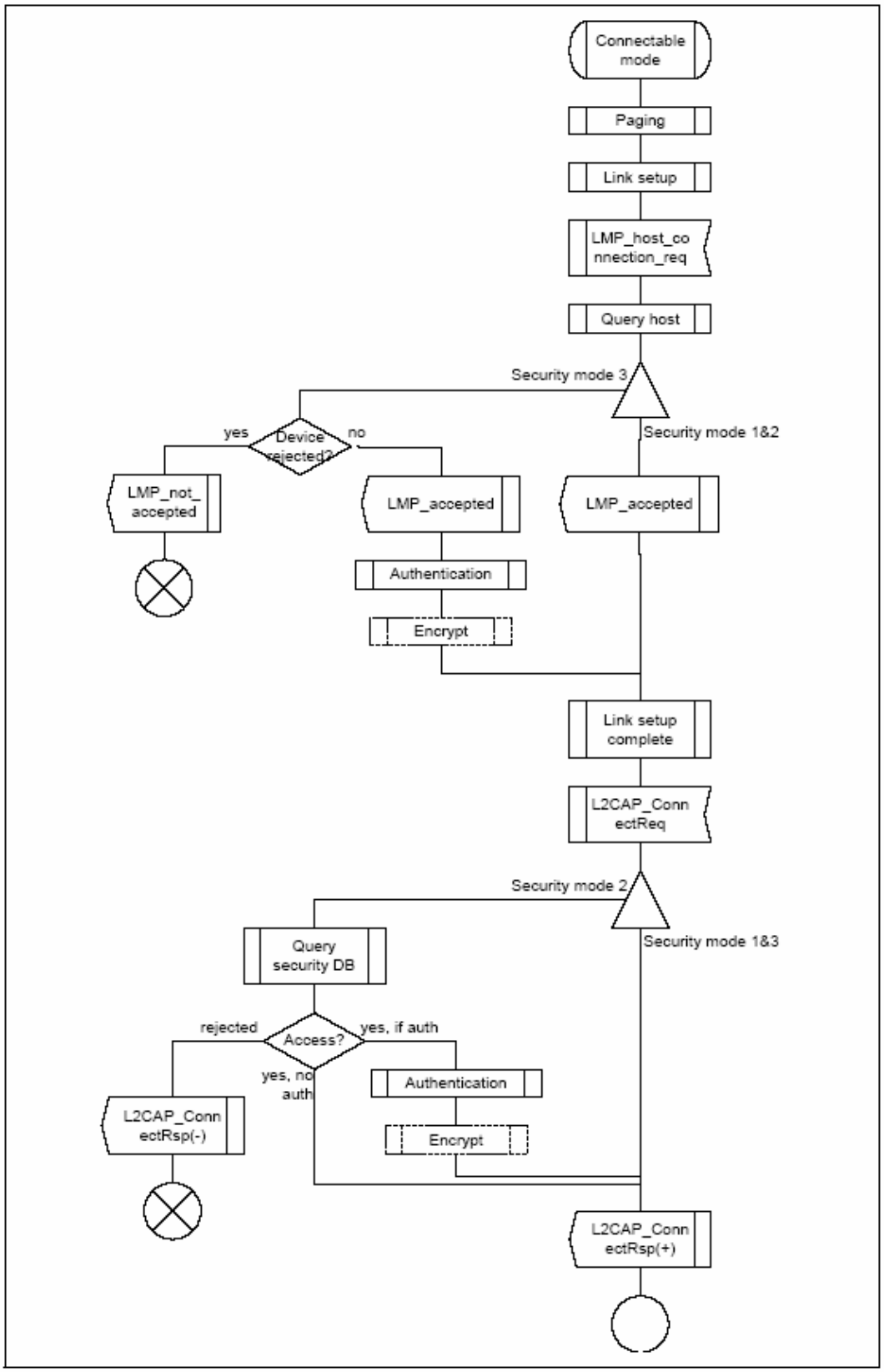



Figura 5.2: Ilustración del establecimiento de canal usando diferentes modos de seguridad.

# 5 Procedimientos del Modo Idle

Los procedimientos inquiry y discovery descritos aquí sólo son aplicables al dispositivo que los comienza (A).

## 5.1 Inquiry General

- El propósito de inquiry general es proporcionar el inicio con el dispositivo de dirección Bluetooth, reloj, la Clase de Dispositivo y el usó page scan modo de dispositivos descubribles generales (es decir dispositivos que están en rango con respecto al iniciador y son escaneados por los mensajes inquiry con el Código de Acceso General Inquiry). También se descubrirán dispositivos en modo descubrible limitado usando inquiry general.

- Inquiry general debe ser usado por dispositivos que necesitan descubrir dispositivos que son descubribles continuamente o en ninguna condición específica.

- Para recibir repuesta de inquiry, los dispositivos remotos en rango tienen que estar descubribles (limitado o general).

## 5.2 Inquiry limitado

- El propósito de inquiry general es proporcionar el inicio con el dispositivo de dirección Bluetooth, reloj, la Clase de Dispositivo y el usó page scan modo de dispositivos descubribles limitados. Los últimos dispositivos son dispositivos que están en rango con respecto al iniciador, y puede escanear los mensajes inquiry con Código de Acceso Limitado Inquiry (LIAC), además de ser escaneado por los mensajes inquiry Código de Acceso General Inquiry.

- Inquiry limitado debe ser usado por dispositivos que necesitan descubrir dispositivos que sólo se hacen descubrible por un periodo limitado de tiempo, durante las condiciones temporales o para un evento específico. Puesto que no se garantiza que el dispositivo descubrible scanee para el LIAC, el dispositivo de inicio puede escoger cualquier procedimiento inquiery (general o limitado).

- Para recibir respuesta inquiry, los dispositivos remotos en rango tienen que estar en descoverable limitado.

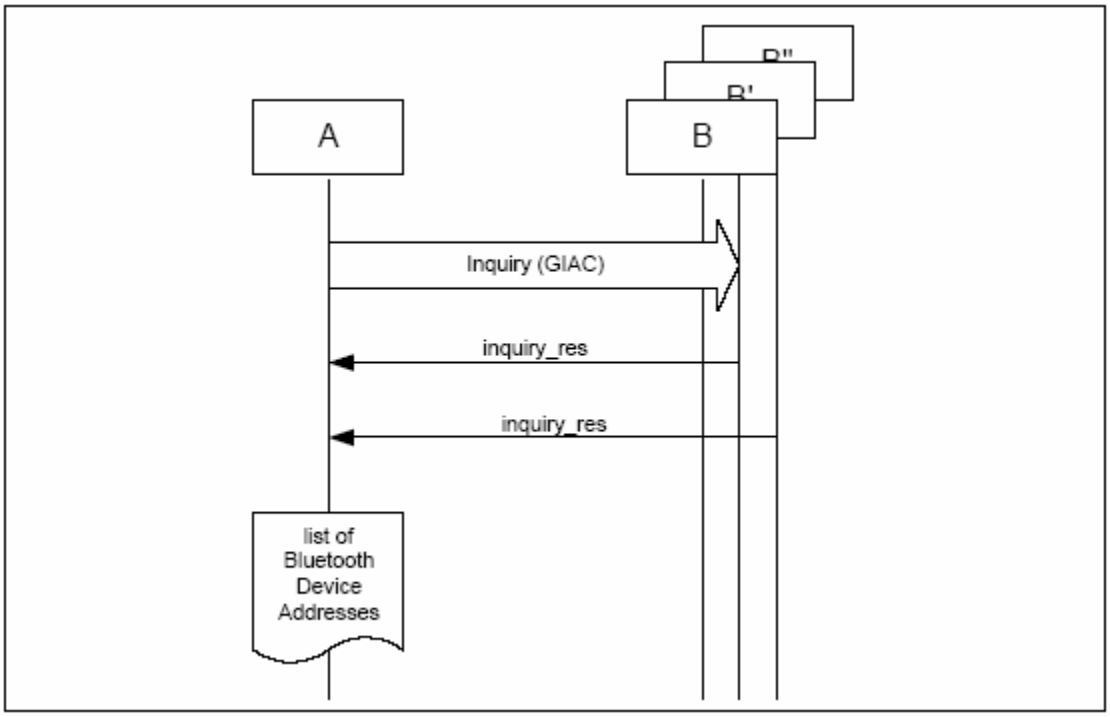



Figure 5.2: General inquiry, donde B es un dispositivo en modo non-discoverable, B´ es un dispositivo en modo limited discoverable mode y B" es un dispositivo en modo general discoverable mode. (Note todos lo s dispositivos discoverable son descubiertos usando general inquiry, independent del cual el modo discoverable esta dentro.)

## 5.3 Descubrimiento del Nombre

El propósito de descubrir el nombre es proporcionar el inicio con el Dispositivo Name Bluetooth, de dispositivos conectados (es decir los dispositivos en rango que responderá a paging), puede hacerse de 2 maneras:

**Name request**

- Procedimiento para recuperar el Dispositivo Name Bluetooth de un dispositivo conectado Bluetooth. No es necesario realizar el procedimiento de establecimiento de enlace full solo para conseguir el nombre de otro dispositivo.

**Name discovery**

- Procedimiento para recuperar el Dispositivo Name Bluetooth de un dispositivo conectado Bluetooth realizando por name request hacia los dispositivos conocidos (es decir dispositivos Bluetooth para los cuales el Dispositivo Addresses Bluetooth están disponibles).

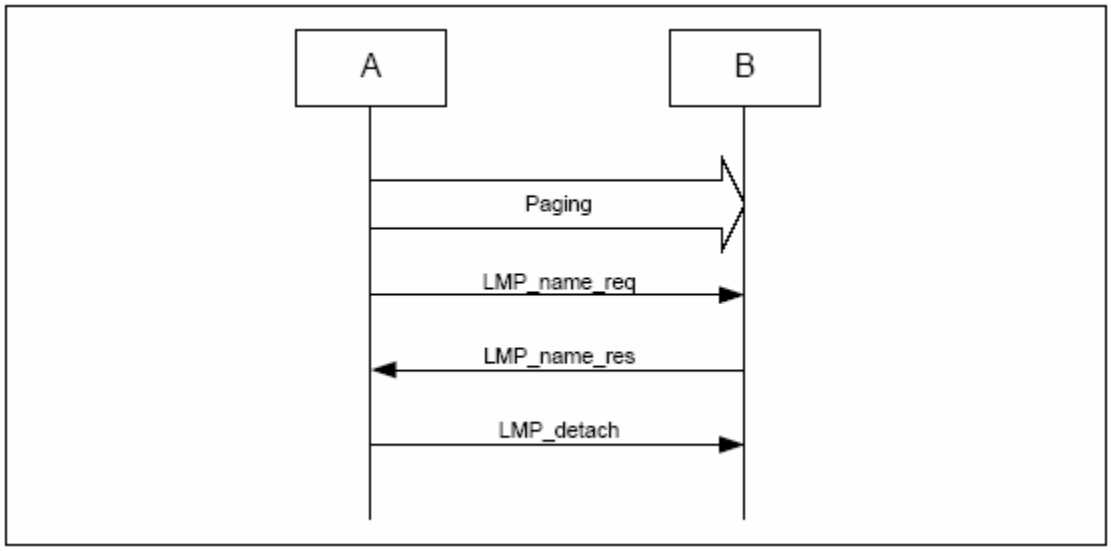



Figura 5.3: Name request procedure.

## 5.4 Descubrimiento del Dispositivo

- El propósito del descubrimiento del dispositivo es proporcionarle la Dirección Bluetooth al iniciador, reloj, Clase de Dispositivo, el uso del modo page scan y el dispositivo name Bluetooth de dispositivos descubribles.
- Dispositivos descubiertos durante el descubrimiento de dispositivo, deben ser ambos, descubribles y conectados.

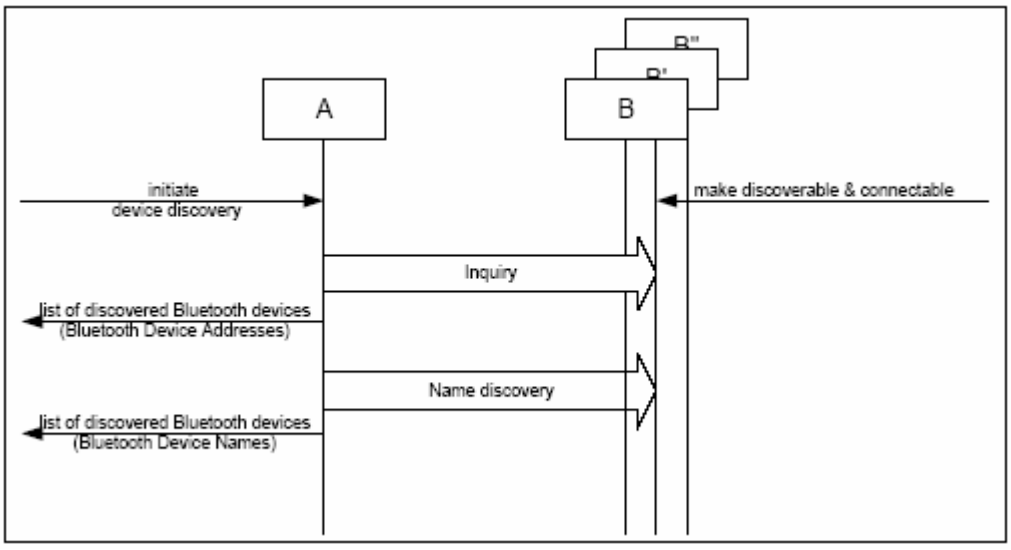



Figura 5.4: Device discovery procedure.

## 5.5 Bonding

- El propósito de Bonding es crear una relación entre dos dispositivos Bluetooth basado en una llave de enlace común (a bond). La llave de enlace se crea y se intercambia (pairing) durante el procedimiento bonding y se espera que sea guardado por ambos dispositivos Bluetooth, para ser usado para una autenticación futura.
- Antes de bonding se puede iniciar, el dispositivo de inicio (A) debe saber el Código de Acceso del Dispositivo del dispositivo para unirse.

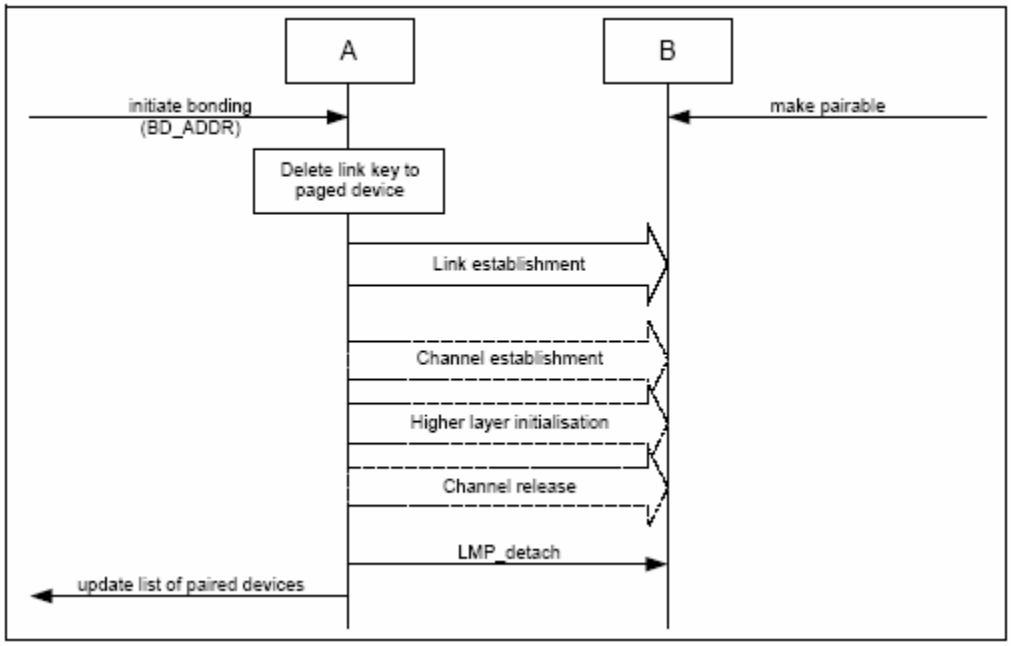



Figura 5.5: Descripción General de bonding es como el procedimiento de establecimiento de conexión ejecutado bajo condiciones específicas en ambos dispositivos, seguido por un proceso más alto opcional de inilization de capa.

# 6 Procedimientos de Establecimiento.

Antes de que se inicie los procedimientos de establecimiento, la información proporcionada durante el descubrimiento de dispositivo (en el paquete de FHS de la respuesta inquiry o en name request) tiene que estar disponible en el dispositivo que inicia. Esta información es:

- El Dispositivo de Dirección Bluetooth (BD_ADDR) del cual el Dispositivo de Código de Acceso se genera;
- El reloj del sistema del dispositivo remoto;
- El modo page scan usado por el dispositivo remoto.

## 6.1 Establecimiento de Enlace

El propósito del procedimiento de establecimiento de enlace es establecer un enlace físico (de tipo de ACL) entre dos dispositivos Bluetooth.
Involucra 2 fases típicamente: Paging y Link Setup

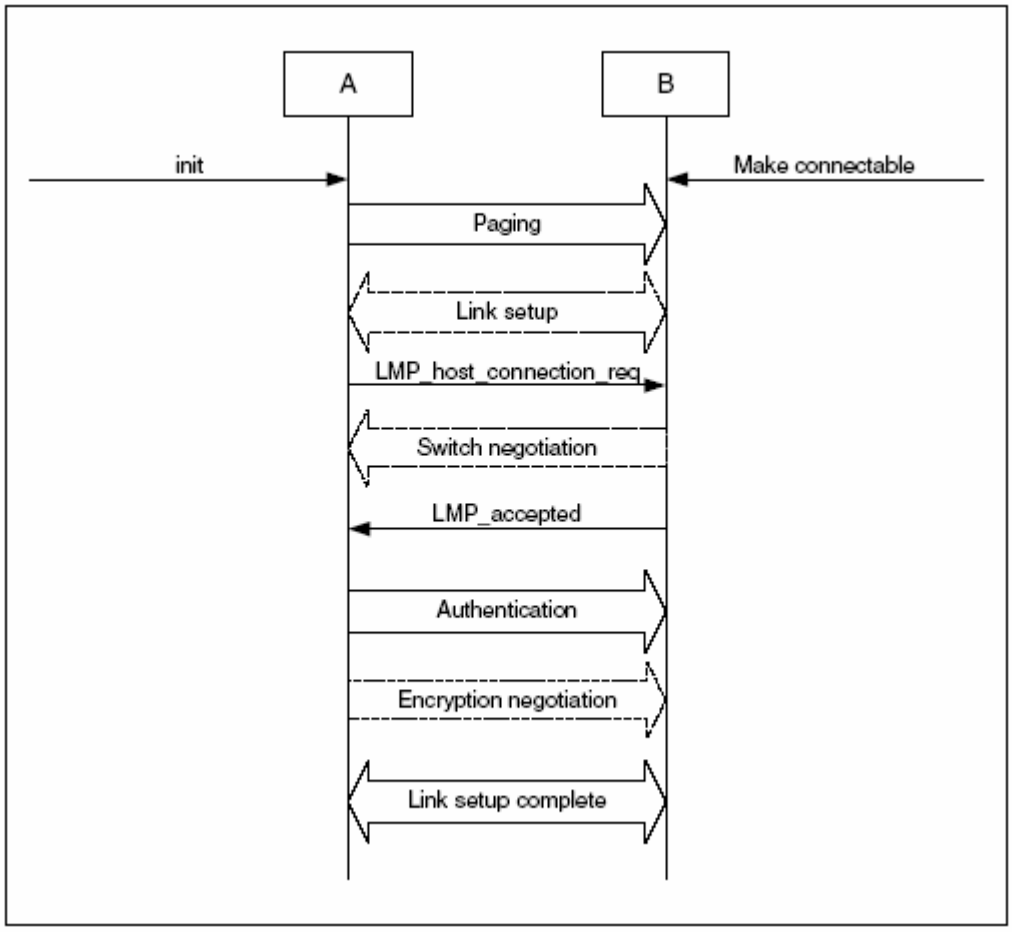



Figura 6.1: Procedimiento de establecimiento de enlace cuando el dispositivo paging (A) está en el modo de la seguridad 3 y el dispositivo paginado (B) está en el modo de la seguridad 1 o 2.

## 6.2 Establecimiento de Canal

El propósito de este es establecer un canal Bluetooth (un enlace lógico) entre dos dispositivos Bluetooth.
El establecimiento del canal empieza después del establecimiento del enlace que se completa cuando el iniciador envía una petición de establecimiento de canal.

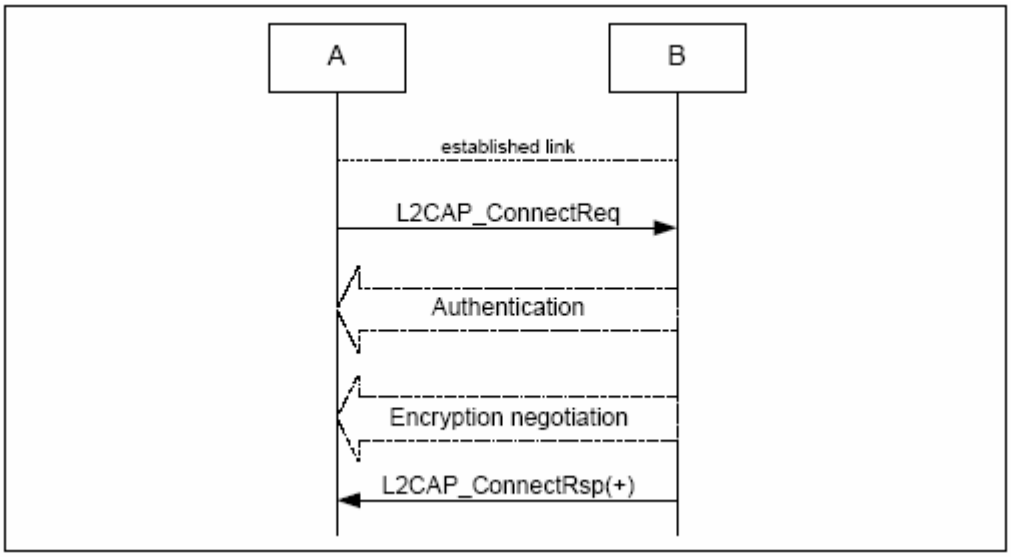



Figura 6.2 :Procedimiento del establecimiento de canal cuando el iniciador (A) está en el modo de la seguridad 3 y el aceptante (B) está en el modo de la seguridad 2.

## 6.3 Establecimiento de Conexión

El propósito es establecer una conexión entre las aplicaciones en dos dispositivos Bluetooth.

## 6.4 Establecimiento de Conexiones Adicionales

Cuando un dispositivo de Bluetooth ha establecido una conexión con otro dispositivo Bluetooth, puede estar disponible para el establecimiento de:

- Una segunda conexión en el mismo canal, y/o
- Un segundo canal en el mismo enlace, y/o
- Un segundo enlace físico

# Capitulo IV
# Implementación de una red Bluetooth

# PARTE A
# PERFIL OBJETIVO DE CAMBIO GENÉRICO

Este perfil define los requisitos para dispositivos Bluetooth necesarios para el soporte, un uso de modelos del perfil objetivo de cambio genérico, son por ejemplo, Sincronización, la Transferencia del Archivo, o el modelo push object Esencialmente, el propósito de este documento será trabajar como un documento genérico de perfil, para todos los perfiles de la aplicación que utilizan el protocolo de OBEX.

## 1 Vista General del Perfil

## 1.1 Reglas/Configuraciones

Las siguientes reglas se definen para este perfil:

- **Servidor –** Esto es el dispositivo que proporciona a un objetivo de cambio y del cuál objetos de datos se pueden pushed y pueden ser pulled, respectivamente.
- **Cliente –** Esto es el dispositivo que puede push o/y pull los objetos de datos al servidor.

## 1.2 Escenarios de Perfil

Los escenarios cubiertos por este perfil son lo Siguientes:

- El uso de un servidor por Cliente para push los datos objeto (objetos)
- El uso de un servidor por Cliente para pull los datos objeto (objetos)

Ciertas restricciones aplican a este perfil, por ejemplo el perfil sólo sostiene configuraciones punto a punto, y la interacción de usuario requiere a colocar al servidor en modos iniciales.

## 1.3 Fundamentos de Perfil

Los fundamentos del perfil, con la cual todos perfiles de la aplicación deben obedecer, será lo Siguiente:

1. Antes un servidor es utilizado con un Cliente por un primer tiempo, un procedimiento que vincula inclusive se puede realizar el paring. Este procedimiento deberá soportar, pero su uso es dependiente en los perfiles de la aplicación. El bonding implica típicamente activado manualmente bonding soporte y entrada a un código de PIN Bluetooth en los teclados de los dispositivos de Cliente y servidor. Este procedimiento quizás se tenga que repetir bajo ciertas circunstancias; por ejemplo, si una llave común de la conexión (es un resultado bonding) es quitado en el dispositivo implicado en el cambio objetivo.
2. Además del nivel de la conexión bonding, un procedimiento de la inicialización de OBEX se puede realizar antes de que el Cliente puede utilizar al servidor para el primer tiempo. La aplicación del perfil utilizando GOEP debe especificar si este procedimiento se debe soportar para proporcionar el nivel de seguridad requerido.
3. La seguridad puede ser proporcionada autenticando la otra parte sobre el establecimiento de la conexión, y la codificación de todos los datos de usuario en el nivel de la conexión. La autenticación y la codificación deben ser sostenidas por los dispositivos; pero si ellos son utilizados depende del perfil de la aplicación que utiliza GOEP.
4. Enlace de conexión y canal se deberán hacer según los procedimientos definidos en GAP
5. No hay reglas fijas de master/esclavo.
6. Este perfil no requiere un modo más bajo para poder ser utilizado..

## 2 Capa de Aplicación

Esta sección describe las capacidades de servicio que pueden ser utilizadas por los perfiles de la aplicación que utiliza GOEP.

## 2.1 Vista General de Características

Hay 3 características que el perfil Objetivo de cambio Genérico que proporciona para los perfiles de aplicación:

- Establecer una Sesión Objetiva de Cambio
- Objeto Pushing de Datos
- Pulling Objeto de Datos

El uso de otras características (por ejemplo. colocando la guía actual) debe ser definido por los perfiles de aplicaciones que lo necesitan.

## 2.2 Establece una Sesión Objetiva del Cambio

Esta característica se utiliza para establecer la sesión objetiva de cambio entre el Cliente y el servidor. Antes una sesión es establecida, los datos de payload no se podrán cambiar entre el Cliente y el servidor.

## 2.3 Objeto Pushing de Datos

Si las necesidades de datos para ser transferidas del Cliente a servidor, entonces estas características son utilizadas.

## 2.4 Pulling Objeto de Datos

Si los datos necesitan ser transferidos de servidor a cliente, entonces esta característica se utiliza.

# 3 Requisitos de Interoperabilidad de OBEX

## 3.1 Operaciones de OBEX Utilizadas

Hay 6 operaciones de OBEX que son especificadas por el protocolo de OBEX: **Conecta**, **Desconecta**, **GET**, **Aborta** & **SetPath.**

La aplicación de perfil utilizada GOEP debe especificar cuál operaciones se deberá soportar para proporcionar la funcionalidad definida en los perfiles de aplicación.

## 3.2 Inicialización OBEX

Si la autenticación de OBEX se soporta y es utilizada por el servidor y el cliente, la inicialización para esta autenticación se debe hacer antes de la primera conexión de OBEX se podrá establecer. La inicialización se puede hacer en tiempo, antes de la primera conexión de OBEX. La inicialización de la autenticación de OBEX requiere la intervención del usuario en el dispositivo del cliente y el dispositivo del servidor.

La autenticación se hace utilizando una contraseña de OBEX, que puede ser igual que un código de PIN Bluetooth en el nivel de la conexión. Incluso si el usuario utiliza el mismo código para la autenticación de la conexión y la autenticación de OBEX, el usuario debe entrar a estos códigos separadamente. Después que entrar la contraseña de OBEX en el cliente y el servidor, la contraseña de OBEX se almacena en el cliente y el servidor, y se puede utilizar en el futuro para autenticar al cliente y al servidor. Cuándo una conexión de OBEX se establece, los dispositivos deben autenticar uno al otro, si la autenticación de OBEX lo permite.

## 3.3 Establecimiento de Sesión OBEX

Para objetivo de cambio, la conexión de OBEX se puede hacer con o sin la autenticación de OBEX. Todos los perfiles de la aplicación que utiliza GOEP deben soportar una sesión de OBEX sin la autenticación.

## 3.4 Pushing Servicio de Datos

El objeto de datos (objetos) es pushing al servidor que utiliza la operación PUT ldel protocolo de OBEX. Los datos se pueden mandar en uno o más paquetes OBEX.

## 3.5 Pulling Servicio de Datos

El objeto de datos (objetos) es el pulled del servidor que utiliza el GET de operación del protocolo OBEX. Los datos se pueden mandar en uno o más paquetes de OBEX.

# 4 Requisitos de Interoperabilidad de Perfil de Puerto en Serie

Este perfil requiere la conformidad a los requisitos del protocolo del Perfil del Puerto en serie (SPP) [12]. Para los propósitos de leer el SPP, el servidor siempre será considerado para ser Dispositivo B y el cliente siempre será considerados para ser Dispositivo A.

Ningunas adiciones a los requisitos de la interoperabilidad de SPP indicaron para el L2CAP, RFCOMM, SDP & el uso del Director de Conexión se requieren. Para capas de LM en este perfil, los modos discoverable Limitados se deben utilizar; pero, si el dispositivo del servidor para alguna razón (por ejemplo. la falta de un usuario suficiente comunica) las necesidades para ser visible siempre, el modo discoverable General se puede utilizar en lugar. El dispositivo del cliente debe soportar el procedimiento General de inquiry y debe soportar también el procedimiento Limitado inquiry.

# 5 Requisitos Genéricos de Interoperabilidad de Perfil de Acceso

Este perfil requiere la conformidad al Perfil Genérico del Acceso. Definen en detalle los requisitos de soporte con consideraciones a procedimientos y capacidades definidos en el GAP.

# PARTE B
# PERFIL DEL OBJETO PUSH

Este perfil define los requisitos para los protocolos y los procedimientos que serán utilizados por las aplicaciones que proporciona el modelo. El objetivo del uso del push el modelo objetivo del uso del push utiliza el Perfil Objetivo, Genérico y fundamental del Cambio (GOEP) definir los requisitos de la interoperabilidad para los protocolos necesitados por aplicaciones. Los escenarios típicos cubiertos por este perfil son: objeto Push, la Tarjeta pull & el Cambio de Tarjeta, todos los cuales implican el push/pull de objetos de datos entre dispositivos de Bluetooth.

## 1 Perfil General

## 1.1 Reglas/Configuraciones

Las siguientes reglas se definen para este perfil:

- **Push al servidor –** Esto es el dispositivo del servidor que proporciona a un servidor objetivo de cambio. Además de los requisitos de la interoperabilidad definidos en este perfil, el Push del servidor debe conformarse con los requisitos de la interoperabilidad para el servidor del GOEP si no es definido en el contrario.

- **Push a Cliente –** Este es el dispositivo de cliente que pushes y pulls objetos del servidor push. Además de los requisitos de la interoperabilidad definidos en este perfil, el push a cliente debe conformarse con los requisitos de la interoperabilidad para cliente del GOEP, si no es definido al contrario.

## 1.2 Escenarios de Perfil

Los escenarios cubiertos por este perfil son los Siguientes:

- El uso de un cliente push del un objeto a un servidor push. El objeto puede, por ejemplo, ser una tarjeta o una cita.
- El uso de un cliente push del pull a una tarjeta de servidor.
- El uso de un cliente push del para cambiar tarjetas con un push del servidor.

## 1.3 Fundamentos de Perfil

Los fundamentos del perfil, son igual que el GOEP

Enlace de nivel autenticación y codificación plana es obligatorio el soporte y el uso opcional. Bonding es obligatorio soporte y la autenticación es opcional de utilizar. OBEX no se utiliza.

Este perfil no pone bajo obligación al servidor ni al cliente para entrar discoverable ni los modos de connectable automáticamente, incluso si ellos son capaces de hacerlo así.

## 2. Interface de Usuario (Aspectos)

## 2.1. Selección del Modo Servidor Push

El modo objetivo de Cambio afecta al servidor push. Permite a clientes push a push/pull al servidor push. Los clientes push pueden tratar también objetos pull del servidor push en este modo. El servidor push no tiene que soportar la característica de pull, pero debe ser capaz de responder con un mensaje de error apropiado.

## 2.2. Selección de la Función de Clientes Push

Hay tres **funciones** diferentes asociadas con el perfil Objetivo Push:

- Función objetiva push
- Función la Tarjeta pull
- Función de Cambio de Tarjeta

1. La función **Objetiva** push inicia la función en pushes uno o más objetos a un servidor push.
2. La **Tarjeta pull** inicia su función pull en la tarjeta de un servidor push.
3. La función **del Cambio de** la Tarjeta inicia la función que cambia tarjetas con un servidor push.

Las tres funciones deben ser activadas por el usuario. Estas no deben ser realizadas automáticamente sin una interacción de usuario. Cuando el usuario escoge uno de estas funciones, un procedimiento de inquiry se realizará para producir una lista de dispositivos disponibles en la localidad.

## 2.3. Acontecimientos de Uso de Aplicación

Las interacciones del usuario determinan cómo los varios escenarios (objeto push, la Tarjeta pull, el Cambio de Tarjeta) quedan fuera. Los detalles repletos de estos escenarios se muestran en el Perfil Objetivo push.

# 3. Capa de Aplicación

Esta sección describe las capacidades del servicio que pueden ser utilizadas por los perfiles de la aplicación que utiliza GOEP.

## 3.1 Vista General

La función Objetivo es obligatoria en el Cliente push y Servidor push. La Tarjeta pull y las funciones del Cambio de Tarjeta son opcionales. Sin embargo el servidor push debe ser capaz de responder con un código del error en cualquiera pull request, incluso si no soporta esta característica.

## 3.2 Característica del objeto Push

Esta característica permite que un Cliente push mande uno o más objetos a un Servidor push.

- **Formato content:** lograr la aplicación la interoperabilidad plana, formatos content se definen para el objetivo push. Para algunas aplicaciones especificadas lo contento de formatos se recomienda sumamente que un Cliente push no tratara de mandar objetos de un formato que el Servidor push no sostiene.

- **El Procedimiento de la aplicación**: es obligatorio para servidores push ser capaz de recibir múltiples objetos dentro de una conexión de OBEX. No es obligatorio para Clientes push ser capaz de mandar múltiples objetos durante una conexión de OBEX. El Cliente push utiliza un PUT la operación para cada objeto que lo quiere mandar. No es obligatorio soportar mandar ni recibir de múltiples objetos dentro de una sola operación PUT.

## 3.3 Característica de la Tarjeta Pull

Un Cliente push puede suministrar opcionalmente la necesitar funcionalidad de una tarjeta pull de un servidor push. Esto es opcional para el Servidor push para soportar la tarjeta pull la característica. Sin embargo, debe ser capaz de responder para pull request con un mensaje de error.

- La **Tarjeta owner**: Dispositivos que sostienen la tarjeta pull y los servicios del cambio de tarjeta deben almacenar la tarjeta owner en el Defecto de OBEX GET Objeto. Algunos dispositivos (por ejemplo. dispositivos públicos) quizás tenga información en la tarjeta owner que es pertinente al dispositivo rather mas que al dispositivo owner.

## 3.4 Característica de Cambio de Tarjeta

Un Cliente push puede suministrar opcionalmente y necesitar la funcionalidad pull de una tarjeta de un servidor push es opcional para el Servidor push para soportar la tarjeta pull la característica. Sin embargo, debe ser capaz de responder para pull los pedidos con un mensaje de error, esta Característica se parece a la Tarjeta pull la Característica con el ejemplo obvio que las tarjetas de ambos lados se cambian. Vea característica de la tarjeta pull para más información.

# 4 Requisitos de Interoperabilidad de OBEX

## 4.1 Las Operaciones utilizadas por OBEX

Hay 5 operaciones de OBEX que se utilizan en el Perfil Objetivo push: **Conecta**, **Desconecta**, **Put**, **Get** & **Aborta.**

## 4.2 Inicialización de OBEX

Desde que la autenticación de OBEX no es utilizada por este perfil, la inicialización de OBEX no es aplicable.

## 4.3 Establecimiento de Sesión de OBEX

Para el Cambio Objetivo, la conexión de OBEX se puede hacer con o sin la autenticación de OBEX. Todos perfiles de la aplicación que utiliza GOEP deben soportar una sesión de OBEX sin la autenticación.

## 4.4 Datos de Push

Se recomienda sumamente que el Cliente push utilizara el header del tipo objetos pushing al Servidor push.

## 4.5 Datos de Pull

En el Perfil Objetivo push, el Cliente push sólo datos pull desde el Servidor push cuando GET el Default Get objeto (tarjeta owner).

Si no hay Default Get objeto, el Servidor push debe responder con el código de la respuesta del error "NOT FOUND". El Cliente push debe ser capaz de entender este código de la respuesta del error.

El Cliente push debe utilizara el header del Tipo al obtener el Default Get Objeto del Servidor push.

El header del nombre no se utiliza al obtener el Default Get Objeto del Servidor push. Si el Cliente push manda un header no vacío del nombre, el Servidor push debe responder con el código de respuesta "FORBIDDEN".

La nota, el encima de texto contiene extractos de la Especificación de SIG Bluetooth, así como varias interpretaciones del Specs. Para detalles completos de las varias secciones, consulte la Especificación verdadera de Bluetooth

# PARTE C
# PERFIL DE LA TRANSFERENCIA DE ARCHIVO

Este perfil define los requisitos para los protocolos y los procedimientos que serán utilizados por las aplicaciones que proporciona el modelo del uso de la transferencia de archivo el modelo de uso de transferencia de archivo utiliza el Perfil Objetivo, de cambio Genérico (GOEP) definir los requisitos de la interoperabilidad para los protocolos necesitados por aplicaciones. Los escenarios típicos cubrieron por este perfil que implica un dispositivo bluetooth browsing, transfiriendo y en/con objetos que manipula otro dispositivo de Bluetooth.

## 1 Perfil General

## 1.1 Reglas/Configuraciones

Los papeles siguientes se definen para este perfil:

- **El servidor –** El dispositivo de servidor es el dispositivo objetivo remoto Bluetooth que proporciona a un servidor objetivo de cambio y browsing y la capacidad que utiliza el formato de la lista de fólder de OBEX. Además de los requisitos de la interoperabilidad definido en este perfil, el servidor debe conformarse con los requisitos de la interoperabilidad para el servidor GOEP si no es definido en el contrario.

- **El cliente –** El dispositivo de Cliente inicia la operación, que push y Objeto Pulls al servidor. Además de los requisitos de la interoperabilidad definido en este perfil, el Cliente debe conformarse también con los requisitos de la interoperabilidad para el Cliente GOEP si no es definido en el contrario.

## 1.2 Escenarios de Perfil

Los escenarios cubiertos por este perfil son los siguientes:

- El uso del **Cliente para browsing la tienda objetiva del servidor.** Los clientes son requeridos a pull y entender Objetos de la Lista de fólder. Los servidores son requeridos a responder a pedidos para Objetos de Listado de fólder. Los servidores son requeridos a tener una fólder de raíz. Los servidores no son requeridos a tener una jerarquía de fólder debajo del fólder de raíz.
- El uso del **Cliente para transferir objetos a y del servidor.** La transferencia de objetos incluye fólder y archivos. Los clientes deben soportar la habilidad de push o pull los archivos del servidor. Los clientes no son requeridos a push ni pull fólder. Los servidores son requeridos a soportar el push del archivo, pull, o los dos. Los servidores son permitidos para tener folders y archivos de sólo lectura, que significa que ellos pueden restringir los objetivos push. Así, los servidores no son requeridos a soportar el push de folder ni pull.
- El uso del **Cliente para crear folders y borrar objetos** (folders ***y archivos*** ) **en el servidor.** Los clientes no son requeridos a soportar la supresión del folder/archivo ni la creación de folder. Los servidores son permitidos a soportar folders y archivos de sólo lectura, que significa que ellos pueden restringir la supresión del folder/archivo y la creación.

Un dispositivo que adhiere a este perfil debe soportar la capacidad de Cliente, la capacidad de servidor o los dos.

## 1.3 Fundamentos de Perfil

Los fundamentos del perfil, son igual que el GOEP

Soporte para el enlace la autenticación y la codificación plana se requiere pero su uso es opcional. Apoyo para la autenticación de OBEX se requiere, pero su uso es opcional. El apoyo de bonding se requiere pero su uso es opcional.

Este perfil no pone bajo mando al servidor ni a cliente para entrar discoverable ni los modos de connectable automáticamente, incluso si ellos son capaces de realizar así en el lado de Cliente, la intervención de usuario final siempre necesitara iniciar la transferencia del archivo.

## 2 Interface de usuario (Aspectos)

## 2.1. La Selección del Modo de Transferencia de Archivo de los servidores

Los servidores deben ser colocados en el modo de Transferencia de Archivo. Este modo permite a un Cliente a realizar las operaciones de la transferencia del archivo con el servidor. Al entrar este modo, Archiva a servidores de Transferencia deberán poner el dispositivo en el modo Discoverable *Limitado* (ver el Perfil Genérico de Acceso), asegura que el Bit Objetivo de la Transferencia se coloque,

y registre un registro del servicio en el SDDB. Se recomienda que este modo fuera set y unset para la interacción de usuario, cuándo posible.

## 2.2 Selección de la Función de Clientes

Los clientes proporcionan al archivo las funciones de la transferencia al usuario vía un usuario comunican. Un ejemplo de un usuario de transferencia de archivo es comunicar una interface de file-tree browse folders y archivos. Utilizando a tal interface file-tree de sistema, el usuario puede browse y manipular los archivos en otra computadora personal, que aparece en la vista de la red.

Las Aplicaciones de Transferencia del archivo proporcionan las funciones siguientes.

- Seleccionar servidor
- Navegar folders
- Objeto pull
- Objeto pull
- Objeto delate
- Crear fólder

Cuándo el usuario escoge la función Seleccionar servidor, un procedimiento de la inquiry se realizará para producir una lista de dispositivos disponibles en la localidad.

## 2.3 Uso de Aplicación

Las interacciones del usuario determinan cómo los varios escenarios (seleccionar servidor, Transferencia de archivo) play out. Los detalles completos de estos escenarios se encuentran en la Sección 3,3 del Perfil Transferencia de Archivo.

# 3 Capa de Aplicación

Esta sección describe las capacidades del servicio que pueden ser utilizadas por los perfiles de la aplicación que utiliza GOEP.

## 3.1 Vista General de Característica

La función fólder browsing y el objeto de transferencia (la Transferencia de Archivo) es obligatorio en el Cliente de la Transferencia de Archivo y servidor de transferencia de Archivo. La Transferencia Objeto (Transferencia de folder) y Función Objetiva de Manipulación es opcional. Sin embargo el servidor debe ser capaz de responder con un código de error en cualquier request, incluso si no soporta esta característica.

## 3.2 Fólder Browsing

Una sesión de la transferencia de archivo comienza con el Cliente que conecta al servidor pull el contenido del folder raíz del servidor. Cuándo una conexión de OBEX se hace, el servidor empieza con su conjunto actual de folder el folder de la raíz.

Browsing un objeto store implica el contenido de folder que demuestra y coloca el 'folder actual'. La orden de OBEX SETPATH se utiliza para poner el folder actual. Para demostrar una jerarquía de folder que empieza con el folder de raíz, el Cliente debe leer y utilizar el contenido del folder de raíz GET. Entonces debe recuperar el contenido de todo utilizar el subfolder GET. Si la subfolder contiene folders, entonces el Cliente debe recuperar el contenido de estas folders etcétera. Para recuperar el contenido de un folder , el Cliente debe poner el folder actual a el subfolder que utiliza SETPATH , entonces pull utiliza el contenido del subfolder GET.

## 3.3 Transferencia de Objeto

Los objetos son transferidos del Cliente al servidor que utiliza OBEX PUT y los objetos son transferidos del servidor al Cliente que utiliza OBEX GET. Transferir los archivos requiere un solo PUT o GET de la operación por el archivo. Transferir folders requieren transferir todos los artículos almacenados en un folder, inclusive otros folders. El proceso de transferir una folder puede requerir que folders nuevos sean creados. La orden de SETPATH se utiliza para crear folders.

## 3.4 Manipulación Objetiva

Un Cliente puede borrar folders y archivos en un servidor. Puede crear también folders nuevas en un servidor. Un resumen breve de estas funciones se muestra abajo.

- Un archivo es borrado utilizando un PUT, la orden con el nombre del archivo en un name header y ningún body header.
- Un folder vacío es borrado utilizando la orden PUT con el nombre del folder en un name header y ningún body header.
- Un folder no vacío se puede borrar del mismo modo que una folder pero los servidores vacíos no pueden permitir esta operación. Si un servidor se niega a borrar una fólder no vacío debe volver el "la condición fallada" (0xCC) código de respuesta. Este código de respuesta el Cliente lo deberá borrar primero todos los elementos del fólder individualmente antes de borrar el folder.
- Una folder nueva se crea en el fólder actual del servidor utilizando la orden de SETPATH con el nombre del fólder en un name header. Si una folder con ese nombre ya existe, entonces una fólder nueva no se crea. En ambos casos el fólder actual es puesto al fólder nuevo.

# 4 Requisitos de Interoperabilidad de OBEX

## 4.1 Las Operaciones utilizadas por OBEX

Hay 6 operaciones de OBEX que se utilizan en el Perfil Objetivo del Push: **Conecta**, **Desconecta**, **Put**, Get, **Aborta & SetPath.**

## 4.2 Inicialización de OBEX

Los dispositivos que aplica el perfil de Transferencia de Archivo puede utilizar opcionalmente la autenticación de OBEX.

## 4.3 Establecimiento de Sesión OBEX

La conexión de OBEX debe utilizar un conjunto de header de Objetivo al Archivo browsing UUID, (F9EC7BC4-953C-11D2-984E-525400DC9E09). Este UUID se manda en el código binario (16 byte) con 0xF9x enviándolo primero. La autenticación de OBEX se puede utilizar opcionalmente.

## 4.4 Browsing Folders

Browsing folders implican pulling, Folder objetos listing y colocan el folder actual. Navegar una jerarquía de folder requiere foward y backward, cambiando el folder actual. Sobre la terminación OBEX Conecta la operación folder actual de servidor y rota al folder raíz.

- **Pull Lista de Folder de Objeto:** PULL una Lista de folder de objeto utiliza una operación GET. La Conexión identificación y headers de Tipo son obligatorios. Un header del nombre que contiene el nombre del folder se utiliza para pull la lista de un folder. Manda la orden GET sin un header del nombre y se utiliza para pull el contenido del folder actual.
- **Colocación del Folder Actual (Foward)**: Colocando el folder actual requiere la operación de SETPATH.
- **Colocación del Folder Actual (Backward)**: Colocando el fólder back actual requiere la operación de SETPATH.
- **Colocación del Folder Actual (Raíz)**: Colocando el folder actual a la raíz requiere la operación de SETPATH.

## 4.5 Objeto Pushing

Objeto Pushing implica los archivos pushing y a los folders.

- **Archivos Pushing:** Siguen el procedimiento descrito en los Datos que Pushing al servidor GOEP. El header de la Conexión de identificación es obligatorio.

- **Folders Pushing:** Implican folders nuevos que crean a los archivos pushing los archivos. Puede implicar también navegar por la jerarquía de folder.

## 4.6 Objeto Pull

Objeto Pull implica los archivos pulling y los folders.

- **Archivos Pull:** Siguen el procedimiento descrito en los Datos Pull del servidor de GOEP. El header de la conexión de identificación es obligatorio.
- **Folders Pull:** Implican navegar la jerarquía de folder, lista de folder de objeto y los archivos Pull.

## 4.7 Manipulación de Objeto

Manipular objetos incluye objetos que borran y crean folders nuevos. Borran objetos e implican a los archivos que borran los folders.

- **Borrar los Archivos**: Borrando un archivo requiere la operación PUT.
- **Borrar Folders**: Un folder se puede borrar utilizando el mismo procedimiento que se utilizó para borrar un archivo. Borrar un folder no vacío borrará todo su contenido, inclusive a otros folders. Algunos servidores no pueden permitir esta operación y volver a "la Condición Fallada" (0xCC) código de respuesta, indicando que el folder no es vació. En este caso el Cliente necesitará borrar el contenido antes de borrar la folder.

# 5 Descubrimiento de Servicio

La pertenencia del servicio al perfil de la Transferencia del Archivo es un servidor, que permite la transferencia genérica bidireccional del archivo. OBEX se utiliza como un protocolo de sesión para este servicio.

La nota, encima del texto contiene extractos de la Especificación de SIG de Bluetooth, así como varias interpretaciones del Specs. Para detalles completos de las secciones, consulte la Especificación verdadera Bluetooth.

# PARTE D
# PERFIL DE SINCRONIZACIÓN

Este perfil define los requisitos para los protocolos y los procedimientos que serán utilizados por las aplicaciones que proporciona el modelo del uso de sincronización. El de uso de sincronización utilizan el Perfil Objetivo, Genérico y fundamental del Cambio (GOEP) definir los requisitos de la interoperabilidad para los protocolos necesitados por aplicaciones. Los guiones típicos cubrieron por este perfil que implica una computadora que instruye un celular o PDA para cambiar los datos de PIM, o viceversa (un móvil que instruye una computadora para cambiar los datos de PIM), o empezando automáticamente sincronización cuando 2 dispositivos de Bluetooth vienen dentro de la gama.

## 1 Vista General de Perfil

### 1.1 Reglas/Configuraciones

Un ejemplo típico de sincronización es uno en que un celular actúa como a un servidor de IrMC y un cuaderno de computadora personal como un Cliente de IrMC. El Cliente de IrMC (computadora personal) tira los datos de PIM del servidor de IrMC y sincroniza estos datos con datos almacenados en el cliente de IrMC. Después que eso, el cliente de IrMC pone estos datos sincronizados apoyan al servidor de IrMC. nota: La capa de Cliente de IrMC es la entidad que procesa la sincronización según la especificación de IrMC, y el servidor de IrMC es el software de servidor sumiso a la especificación de IrMC.

Los papeles siguientes se definen para este perfil:

- El Servidor de IrMC – Esto es el dispositivo de servidor de IrMC que proporciona a un servidor objetivo del cambio. Típicamente, este dispositivo es un celular o PDA. Además de los requisitos de la interoperabilidad definió en este perfil, el servidor de IrMC debe conformarse con los requisitos de la interoperabilidad para el servidor del GOEP, si no definido al contrario.

- El Cliente de IrMC – Esto es el dispositivo de cliente de IrMC, que contiene un motor de sincronización y tira y pushing los datos de PIM de y al Servidor de IrMC. Generalmente, el dispositivo de Cliente de IrMC es una computadora personal. Porque el Cliente de IrMC debe proporcionar también la funcionalidad para recibir la orden de la inicialización para la sincronización, a veces debe actuar como temporalmente a un servidor.

Además de los requisitos de la interoperabilidad definió en este perfil, el servidor de IrMC debe conformarse con también los requisitos de la interoperabilidad para el servidor y el cliente del GOEP si no definido al contrario.

## 1.2 Escenarios de Perfil

Los escenarios cubiertos por este perfil son:

- **El uso de un Servidor de IrMC por Cliente** de IrMC para los datos pull PIM necesita ser sincronizados del Servidor de IrMC, para sincronizar estos datos con los datos en el Cliente de IrMC, y para pushing estos datos sincronizados apoyan al Servidor de IrMC.
- **El uso de un Cliente de IrMC por un Servidor** de IrMC para iniciar el guión previo mandando una orden de sincronización al Cliente de IrMC.
- **Sincronización automática** iniciada por el cliente de IrMC.

Las restricciones que aplican a este perfil son igual que en el GOEP. Además de estas restricciones, la sincronización de igual a igual no es sostenida por la sincronización de BT.

## 1.3 Fundamentos de Perfil

Los fundamentos del perfil son igual que definido en el GOEP, con la adición de los requisitos que vinculando, liga la autenticación plana, y la codificación (los Fundamentos 1 y 3 en GOEP) siempre debe ser utilizado para este perfil. La autenticación de OBEX (Fundamental 2 en GOEP) como un mecanismo aplicación-plano de seguridad debe ser sostenida por los dispositivos que proporcionan este perfil, pero este perfil no pone bajo el mandato que se debe utilizar.

En este perfil, el Cliente de IrMC y Servidor de IrMC pueden actuar como a un cliente (Servidor de IrMC temporal), pueden iniciar establecimientos de conexión y canal; es decir crea una conexión física entre estos dos dispositivos.

Este perfil no pone bajo el mandato el servidor de IrMC ni a cliente para entrar descubrible ni los modos de connectable automáticamente, incluso si ellos sean capaces de hacer así. Esto significa que la intervención de usuario final se puede necesitar en ambos los dispositivos cuando, por ejemplo, la sincronización se inicia en el dispositivo de cliente de IrMC.

# 2 Comunicación de Usuarios Aspectos

## 2.1 Selección de Modo

Hay dos modos asociados con el perfil de Sincronización.

- **Modo de Sincronización de inicialización**

  En el modo de Sincronización de Inicialización, el Servidor de IrMC está en el Limitado descubrible (o el modo descubrible General), Connectable , y los modos de Pairable. El Cliente de IrMC no entra este modo en este perfil. Se recomienda que el procedimiento Limitado de la Indagación fuera utilizado por el Cliente de IrMC al descubrir al servidor de IrMC.

- **Modo general de Sincronización**

  En el modo General de Sincronización, el dispositivo está en el modo de Connectable. El Cliente de IrMC y el Servidor pueden entrar este modo. Para el Servidor de IrMC, este modo se utiliza cuando el Cliente de IrMC conecta al servidor y empieza la sincronización en los tiempos subsiguientes después deparing. Para el Cliente de IrMC, el modo se utiliza cuando la sincronización es iniciada por el servidor de IrMC.

## 2.2 Uso de Aplicación

Las interacciones del usuario determinan cómo los varios escenarios: Sincronización (first-time/subsequent) y uso de perfil Automático de Sincronización

# 3 Capa de Aplicación

Esta sección describe las capacidades del servicio que pueden ser utilizadas por los perfiles de la aplicación que utiliza GOEP.

## 3.1 Vista General de Características

Es obligatorio para ambos Clientes de IrMC & Servidores, soportar uno o más de los casos siguientes de Sincronización:

1. Sincronización de phonebooks
2. Sincronización de calendarios
3. Sincronización de mensajes
4. Sincronización de notas

Además el Cliente de IrMC debe soportar obligatoriamente la Orden de Sincronización y soportar opcionalmente la Sincronización Automática. El Servidor

de IrMC debe soportar opcionalmente la Orden de Sincronización y soportar obligatoriamente la Sincronización Automática

## 3.2 Característica de Sincronización

El apoyo de Sincronización con el nivel 4 de IrMC en su funcionalidad es obligatorio para ambos Clientes de IrMC y Servidores de IrMC. Los requisitos para la Sincronización de IrMC se definen en el spec de IrMC. La sincronización de Bluetooth debe soportar por lo menos uno de los casos siguientes ( Clases de aplicación):

1. Sincronización de phonebooks
2. Sincronización de calendarios
3. Sincronización de mensajes
4. Sincronización de notas

Para lograr la aplicación la interoperabilidad plana, los formatos se definen para la Sincronización de Bluetooth. Los formatos son dependientes en las clases de la aplicación, que se diseña para los propósitos diferentes. Las clases sostenidas de la aplicación se deben identificar en términos de los datos en el SDDB del Servidor de IrMC

## 3.3 Característica de Orden de Sincronización

Esta característica significa que el dispositivo de cliente de IrMC trabaja temporalmente como un servidor y es capaz de recibir una Orden de Sincronización del servidor de IrMC, que en este caso actúa temporalmente como un cliente. Esta Orden de la Sincronización ordena que el cliente de IrMC para empezar sincronización con el Servidor de IrMC.

Después que mandar la orden de sincronización y obtener la respuesta para lo, el Servidor de IrMC debe terminar la sesión de OBEX y los datos de RFCOMM ligan la conexión. Esta característica debe ser sostenida por el Cliente de IrMC y por puede ser sostenida opcionalmente por el Servidor de IrMC.

## 3.4 Característica Automática de Sincronización

En esta característica, el Cliente de IrMC puede empezar la sincronización cuando el Servidor de IrMC entra en proximidad del RF del Cliente de IrMC. Básicamente, esto significa que, en el nivel de Baseband, el Cliente de IrMC page entra al Servidor de IrMC en intervalos y, cuando encuentra que el Servidor de IrMC está en gama, el Cliente de IrMC puede empezar la sincronización.

El apoyo de esta característica es opcional para el Cliente de IrMC pero es obligatorio para el Servidor de IrMC. Esto significa que el Servidor de IrMC debe

ofrecer una capacidad de colocar el dispositivo de servidor en el modo General de Sincronización para que este no salga de un modo automáticamente.

# 4 Requisitos de Interoperabilidad de OBEX

## 4.1 Las Operaciones de OBEX

Hay 6 operaciones de OBEX que se utilizan en el Perfil Objetivo: **Conecta**, **Desconecta** , **PUT** ,**GET** , **Aborta & SetPath.**

## 4.2 Inicialización de OBEX

**La autenticación** de OBEX debe ser soportada por los dispositivos que se aplican en el perfil de Sincronización.

## 4.3 Establecimiento de de la Sesión de OBEX

El header del Objetivo se debe utilizar cuando el cliente de IrMC establece la conexión. El valor de header de Objetivo es "IRMC-SINCRONIZACION".

## 4.4 Los Datos de Pushing de / Datos/Desconexión

Datos Pushing: Ver la Sección Datos Pushing del Servidor del GOEP

Datos PULL: Ver la Sección Datos PULL del Servidor del GOEP

Desconexión: Ver Perfil verdadero GOEP.

# 5. Descubrimiento de Servicio

Hay dos servicios separados relacionados al perfil de la Sincronización. El primero es el servidor verdadero de sincronización (servidor de IrMC), y el segundo es el servidor de la orden de sincronización (Cliente de IrMC).

La nota, encima de texto contiene extractos de la Especificación de SIG de Bluetooth, así como varias interpretaciones del Specs. Para detalles completos de las secciones, consulte la Especificación verdadera Bluetooth.

# PARTE E
# IMPLEMENTACION DE UNA RED BLUETOOTH

## 1. Introducción

El objetivo es proponer nuevas aplicaciones de Bluetooth que sean distintas a las convencionales -sustitución de cables en periféricos- y a las obvias -soporte de redes ad-hoc, identificación por radiofrecuencia (*Radio Frequency Identi_cation*, RFID)-. Las propuestas se centrarán en dos líneas que son enteramente novedosas: el soporte de localización en interiores para servicios y la telecomunicación múltiple (control de un recurso compartido desde un número dado de terminales).

Han pasado algunas décadas desde la creación de la primera computadora que ocupaba 1.600 metros cuadrados y pesaba unas 30 toneladas hasta la aparición de las PC´s actuales con un peso de hasta 250 gramos y una frecuencia de reloj 10.000 veces mayor. Pero no sólo se ha reducido el tamaño, el consumo y el precio, que ha permitido la popularidad de estos dispositivos, sino que además tienen nuevas funciones inimaginables en tiempos de aquellas primeras computadoras. De las cuales, la principal que esta revolucionando la forma de trabajo es la *movilidad*. Actualmente, se pueden establecer comunicaciones, acceder al correo electrónico, administrar la agenda, o visualizar imágenes desde *terminales móviles*. Hoy en día los pequeños dispositivos portátiles con presencia creciente en el mercado, como los PDAs (*Personal Digital Assistants*) que están convergiendo con otros dispositivos como teléfonos móviles y terminales portátiles industriales, concebidas como ayuda en un principio para recordar citas, números telefónicos, etc. tenían la necesidad de conectarlas físicamente a otros sistemas más potentes, como una PC, para la actualización de la agenda o simplemente para acceder a la red de datos, frente a esto la solución previa fue proporcionarle un puerto infrarrojo (IrDA). Sin embargo, las restricciones de visión directa, alineación y baja velocidad han propiciado que se busquen otras soluciones. Bluetooth es la tecnología inalámbrica mejor posicionada como alternativa.

La tecnología inalámbrica Bluetooth ha sido concebida para dar soporte a aplicaciones tales como la sustitución de cables, y orientada a dimensiones reducidas, bajo consumo y velocidad de transmisión media. Algunos ejemplos pueden ser la telefonía doméstica o la interconexión de periféricos de ordenador. Sin embargo, el potencial de los módems Bluetooth no finaliza ahí. Como toda nueva tecnología, admite usos alternativos, distintos de aquellos para los que fue concebida. Una de las principales motivaciones para descubrir dichos usos son las excelentes previsiones de precio de los módems Bluetooth. Podemos pensar, por

tanto, en el beneficio que aportaría a la industria cualquier implantación masiva que fuese novedosa.

Los dispositivos con comunicación inalámbrica Bluetooth tienen innumerables aplicaciones. En entornos industriales, una persona puede monitorear la línea de producción y su funcionamiento, el operador puede consultar manuales de mantenimiento para reparaciones. Un guardia puede observar las cámaras de vigilancia mientras realiza su ronda, sin necesidad de desplazarse a la sala de monitores. Los ejemplos solo están limitados por la imaginación.

Esta tecnología está siendo integrada en terminales de comercialización masiva, como PDA´s y teléfonos móviles.

Una de las líneas de trabajo más importantes en redes Bluetooth de estos últimos años es interactuar esta tecnología con IEEE 802.11b, una vez que se ha reconocido que se complementan. Como las dos utilizan la banda libre de frecuencias de 2.4 GHz, se puede esperar en un principio que la interferencia mutua puede ser un problema. Uno de los estudios más importantes al respecto en el que se realizan pruebas de coexistencia en diferentes escenarios. De este trabajo se desprende un resultado importante, y es que los módems IEEE 802.11b se ven más afectados por la presencia de los módems Bluetooth que viceversa[1].

Un avance importante para converger estas tecnologías lo dio el IEEE con la creación del Grupo de Trabajo 802.15 (IEEE802.15), que se encarga de la elaboración de estándares de Redes Inalámbricas Personales (*Wireless Personal Area Networks*, WPANs). En especial, el Grupo de Tareas número 2 se encarga de la coexistencia entre redes 802.15 y 802.11. Aparte de los resultados de investigación que hemos mencionado, actualmente existen equipos que integran módems Bluetooth e IEEE 802.11b [2]

## 2. Creación de nuevos Sistemas, con lleva a nuevas necesidades.

Los nuevos sistemas como lo son las redes inalámbricas permiten, entre otras aplicaciones, orientar a los usuarios de un centro comercial, un museo o cualquier superficie dividida en salas o departamentos. El tamaño de la solución varia dependiendo de la aplicación y de la información. Puede tratarse de una tienda, o la sala de un museo, de un objeto en una exposición, o el área temática en una exposición, ferial, etc.

[1] Los estudios relacionados son Enn98, She00, MobCorp01, CR02, SAWN02, Lucent.

[2] Possio. Wireless services gateway PX30. http://www.possio.com, 2003.
Texas Instruments. Wanda PDA.http://focus.ti.com/pdfs/vf/wireless/wanda_031403.pdf, 2003.

Actualmente, el nacimiento del comercio móvil es un caso particular del comercio electrónico, y de los sistemas de información basados en comunicaciones móviles e inalámbricas, que hoy en día ya existen y son tecnologías capaces de dar este soporte. Es indudable que muchos avances tecnológicos han y seguirán impulsados por las necesidades de los individuos y de las empresas. Sin embargo, en otras ocasiones, es el producto quien genera necesidad de consumo en las personas. Otra forma de ver los sistemas de información dependientes del contexto es como una ayuda para eliminar los cientos o miles de artículos de una base de datos que no son importantes para un usuario determinado. Por ejemplo, si estamos en la sección de automóvil de unos grandes almacenes, posiblemente solo necesitaremos información de ese departamento. En el caso más claro de un gran museo, acceder a la página web oficial. Desde una terminal móvil no será de ninguna ayuda, porque el usuario se verá saturado de información y no será capaz de relacionar los datos recibidos.

La tecnologías inalámbricas, como Bluetooth, obedecen a la necesidad de los usuarios, y proponemos una nueva solución basada en la esta tecnología.

Si continúa la tendencia del mercado cada vez más terminales dispondrán de módems Bluetooth, y el usuario no tendrá necesidad de ningún *hardware* adicional para acceder a un sistema de información.

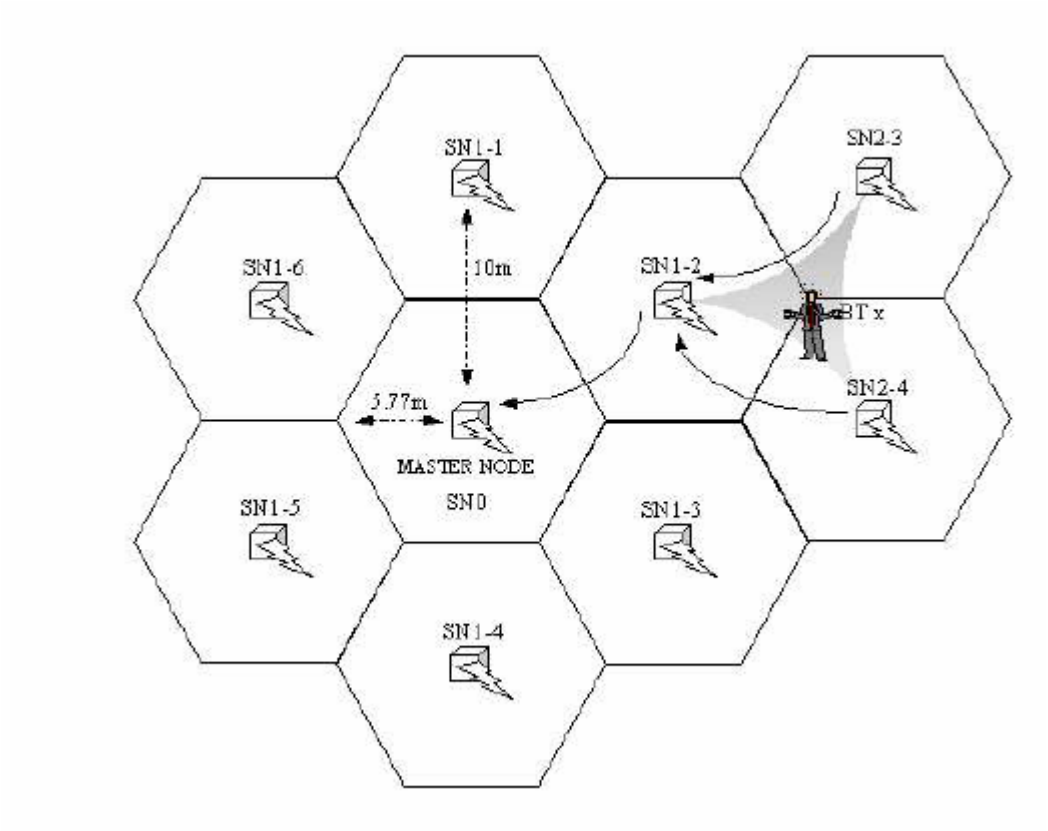
En este capítulo propondremos los mecanismos necesarios para la creación de una red Bluetooth, utilizando para ello módems Bluetooth. Es decir, la misión de la red Bluetooth es habilitar la localización de los dispositivos móviles Bluetooth para determinar los servicios de información que el usuario requiere.

## 3. Entidades de la red

Las terminales móviles se podrán desplazar libremente por una gran superficie cubierta por la red inalámbrica Bluetooth.
Las terminales móviles utilizarán tecnologías comerciales para acceder a la red de datos.
Para la transferencia de datos, los usuarios pueden utilizar cualquier dispositivo con conexión IP: PDAs teléfonos móviles etc.
El servidor central puede *forzar información* hacia las terminales móviles en cualquier instante.

## 3.1. Entidades de la red de localización

A continuación, describiremos las entidades necesarias para construir la red de localización, algunos conceptos básicos y la terminología del capítulo.




En la figura 3.1 se muestra una disposición posible de entidades en un área determinada.

Se puede apreciar que en el centro de cada hexágono imaginario hay un *nodo estático* Bluetooth (*Static Node*) SN*X*-*Y*, donde *X* es el identificador de *capa* concéntrica e *Y* es el índice dentro de la capa. SN*X* _ 1 es el nodo de la capa *X* situado inmediatamente encima del nodo central, y los índices posteriores se incrementan en el sentido de las agujas del reloj. Existe un nodo central (*Master Node*, MN). Como veremos más adelante, en una red Bluetooth, puede existir más de un nodo maestro. Cada nodo estático dependerá de un nodo maestro, pero podrá asociarse dinámicamente a otro distinto si la red así lo requiere (ante fallos, o si se instala un segundo nodo maestro más próximo). El conjunto de nodo maestro y nodos estáticos forman la parte estática o infraestructura de la red Bluetooth. Obviamente, la red también se compone de módems móviles o de usuarios, que serán los dispositivos a detectar y localizar. Nos referiremos a ellos como *BDx*, donde *x* es un índice de dispositivo, *etiquetas*, o *móviles*.

La distancia entre nodos estáticos es fundamental. Debido al alcance nominal de de los módems Bluetooth de clase 2, por ejemplo se ha optado por colocar los nodos estáticos vecinos a esta distancia. Por lo tanto, cada hexágono tiene una área de 86.55 m2. Para darnos una idea de la cantidad de nodo estáticos necesarios, en un recinto de 8,000 m2, se necesitarían 8,000 /86,55 = 93 nodos estáticos. A la vista de los precios de los módems Bluetooth, el costo no resultaría excesivo.

### 3.1.1. Instalación

Para instalar la red Bluetooth es preciso determinar la posición de los nodos estáticos. A fin de minimizar su número, nos interesa alejarlos cuanto sea posible respetando sus limitaciones técnicas. Por otra parte, para minimizar el tamaño de los contextos[3], interesa que cada nodo estático vea el mayor número posible de vecinos. La aproximación de este capítulo mediante una estructura hexagonal es habitual en planificación 2D. En ella, si los nodos estáticos están situados a una distancia suficiente, cada uno de ellos tiene sólo seis vecinos. De esta forma no violamos la restricción básica de Bluetooth que fija un máximo de siete conexiones activas simultáneas. *Los protocolos de este capítulo son independientes de la topología*, y por tanto serían válidos para cualquier topología que cumpla la restricción básica: mallas para 2D, cubos *k*-arios[4] para 3D o disposiciones irregulares dependientes de las condiciones de propagación. La adecuación de la topología a entornos concretos no se considera en esta tesis. En cualquier caso, para minimizar los problemas de absorción, resulta recomendable colocar los nodos estáticos en los techos de los recintos. Determinada la topología, es necesario anotar las direcciones únicas BD_ADDR de cada nodo estático, de forma que su correspondencia con las direcciones topológicas (SN*X*-*Y*) sea conocida. Hecho esto, tan solo resta activar la red Bluetooth. Los nodos maestros asociados a estos servidores desencadenarán una fase de configuración según se describe en la sección 3.2.1, cuya duración será función del número de nodos estáticos y nodos maestros presentes en la red.

### 3.1.2. Zonas de detección

La localización de un usuario se determina en función del conjunto de un nodo estático que lo detecta. Denominamos a esta estrategia *localización cooperativa*, que no ha sido considerada previamente en el contexto de las redes Bluetooth. Debemos enfatizar que Bluetooth se ha utilizado como tecnología de base en trabajos previos. En concreto, la referencia[5] satisface algunas de las condiciones de diseño, y *afirma* que la precisión de Bluetooth es insuficiente para soportar información dependiente del contexto. Considérese por ejemplo el escenario de la figura 3.2. En principio, si nos limitamos a establecer una correspondencia entre un módem y una sala, el usuario podría estar situado en cualquiera de ellas. Con

[3] Contexto: Cualquier información que puede ser utilizada para caracterizar la situación de una entidad. Una entidad es una persona, lugar u objeto relevante para la interacción entre un usuario y una aplicación, incluyendo a los propios usuarios y aplicaciones.

[4] W. Mao y D.M. Nicol. On k-ary n-cubes: theory and applications. NASA CR-194996 ICASE report # 94-88, 1994.

[5] T. Yamasaki, M. Kishimoto, N. Komoda y H. Oiso. An information o_ering system for exhibition explanation by Bluetooth technology. En *Proc. SSGRR 2001, Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, 2001.

nuestra filosofía, basada en la cooperación, se determinaría que el usuario está situado en la intersección de las zonas de detección (área gris). Obsérvese que la mayor parte de la intersección pertenece a la sala donde el usuario está realmente situado. Esto es interesante, porque este escenario en concreto fue utilizado en la referencia[6] como contraejemplo para criticar a la tecnología Bluetooth.
Así, las zonas de detección colaborativa en el escenario ideal de la figura 3.1 se muestran en la figura 3.3

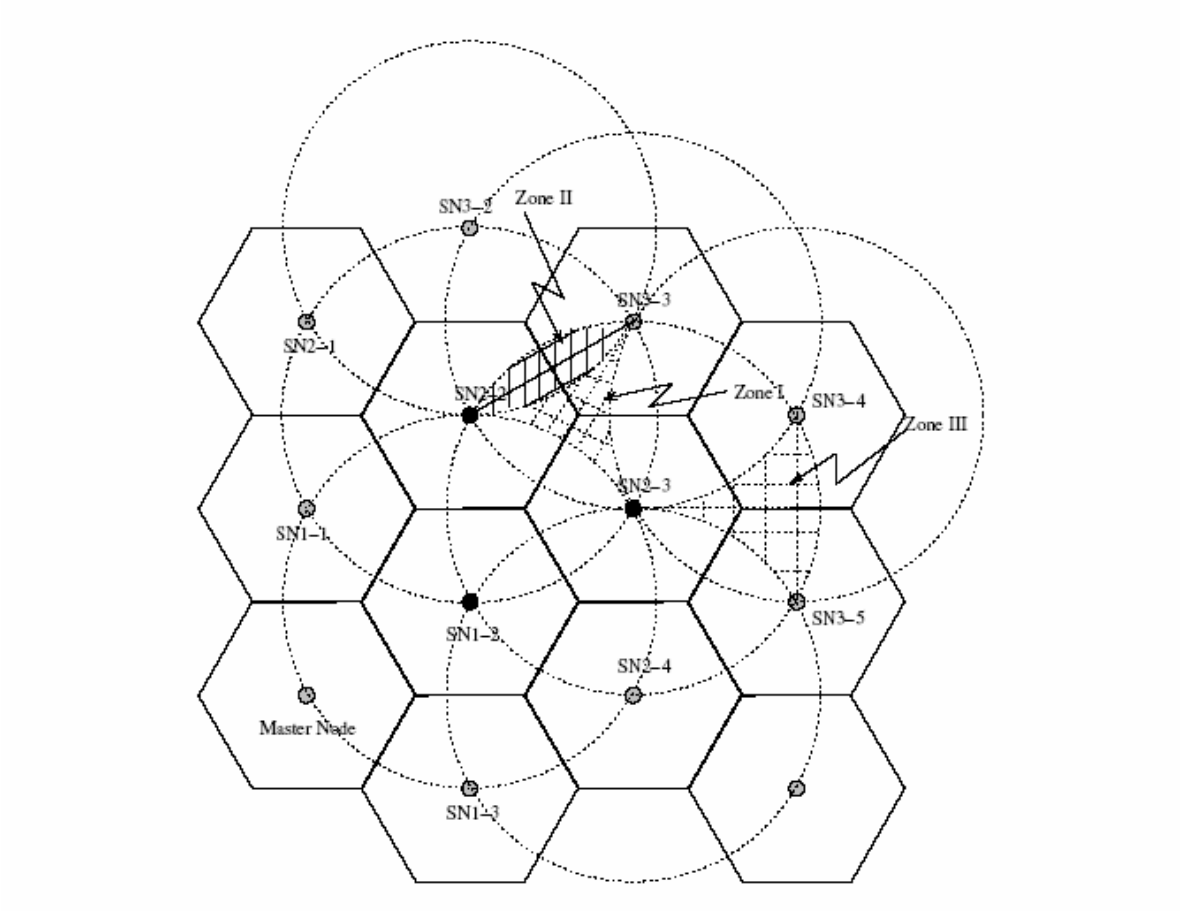



Figura 3.3: Tipos de zonas de localización

Expondremos algunos conceptos básicos, necesarios para entender lo que viene a continuación. Un nodo estático sondea a los móviles que estén en su zona de cobertura (área circundante) mediante ciclos de *inquiry*. Dichos móviles contestarán con su identificación Bluetooth (BD_ADDR*z*). El nodo estático comprobará si tiene que envíar o no estas identificaciones hacia su nodo maestro (como se indica en el apartado 3.2.2, no tiene sentido hacerlo si se trata de móviles detectados previamente). Por lo tanto, el nodo maestro recibe pares que

[6] T. Yamasaki, M. Kishimoto, N. Komoda y H. Oiso. An information o_ering system for exhibition explanation by Bluetooth technology. En *Proc. SSGRR 2001, Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, 2001.

consisten en la dirección de un nodo estático y las direcciones de los móviles detectados por dicho nodo.

Nótese que el servidor de localización recibirá varias tuplas (SN*x* _ *y*,BD_ADDR*z*) es decir, varias entradas para un determinado móvil *z*. Con esta información y el conocimiento de la topología se puede estimar la zona de ubicación de cada móvil en el recinto.

En la figura 3.3 se muestran tres tipos diferentes de zonas, para el caso de estructura hexagonal, dependiendo de los nodos estáticos que cubran dichas zonas. En las zonas de tipo 1 *únicamente* tres nodos estáticos pueden detectar un móvil. Por ejemplo, en la zona 1 marcada en la figura 3.3 se aprecia que sólo los módems SN2-2, SN2-3 y SN3-3 detectarían al móvil. Es interesante resaltar que son tan importantes los nodos que detectan al móvil como los que no lo detectan. En el ejemplo anterior, el no recibir detección de SN3-2, SN3-4 y SN1-2 colabora en de_nir la zona como de tipo I. Las zonas de tipo I tienen un área de 16.12 m2. Si SN3-2 también detecta al móvil, la zona de detección pasaría a ser de tipo II, con un área de 18.12 m2. Las zonas de tipo III corresponden a los casos especiales, que se dan ante caídas de nodos estáticos, errores de detección o en los bordes del recinto.

El ejemplo que mostramos tiene un área de 34.24 m2. Idealmente, existiría además una .zona. IV con precisión exacta, que correspondería al caso en que un móvil está situado exactamente debajo de un nodo estático. En este caso, siete nodos estáticos informan de la detección (el nodo mencionado y sus seis vecinos). El servidor de localización determinaría que la posición del móvil coincide con la del nodo estático central.

Naturalmente, es preciso formalizar el procedimiento utilizado por los servidores de información para situar al usuario en una *malla*, previendo la posibilidad de que se produzcan errores de detección.
Dependiendo del número de nodos que detecten al usuario, y descartando los bordes de la red, podemos considerar los casos siguientes:

- Más de cuatro nodos detectores: El usuario está necesariamente en una zona IV. Su situación coincidirá con la intersección de los rangos de detección (un punto).

- Cuatro: El usuario está en una zona II o en una zona IV. Supongamos que el patrón coincide con una zona II. Si en realidad está en una zona IV, la posición real pertenece a la zona II estimada, puesto que es uno de los vértices de dicha zona.

- Tres: El usuario está en una zona I, II o IV. Supongamos que el usuario está en realidad en una zona IV. En ese caso, la posición real pertenece a cualquier zona estimada, puesto que es uno de sus vértices. Si se estima que el usuario está en una zona I pero en realidad está en una zona II, se tratará de alguna

de las adyacentes a la zona I estimada. En consecuencia, se debe considerar como salvaguarda que el usuario puede estar en cualquiera de ellas, con un área conjunta de 70.48 m2.

- Dos: Si están a una distancia fuera del área de cobertura de un módem Bluetooth, hay constancia de que hay errores de detección y no tiene sentido una equivalencia en zonas. Se sabe, no obstante, que el usuario está en la intersección de dos sectores de detección, con un área de 122.84m2. Curiosamente, si están a distancias mayores, se puede determinar que el usuario está en una zona II, e incluso en una zona IV.

- Uno: De nuevo hay constancia de que hay errores de detección, y no tiene sentido una equivalencia en zonas. El usuario puede estar en cualquier punto del área de cobertura del nodo estático detector, con un área de 314.16 m2. Este caso nos lleva al escenario de la referencia[7]. Sin embargo, es muy poco probable, puesto que implica como mínimo dos errores de detección (cuando el usuario está en realidad en una zona I).

Por lo tanto, razonablemente, en el peor de los casos el máximo tamaño del contexto es de 122.84 m2. En general, sólo los errores *sistemáticos* son preocupantes, ya que en caso de errores esporádicos sucesivas actualizaciones incrementarán la precisión. Nótese que el fallo físico de un nodo aislado provoca errores sistemáticos, pero la Posición de los usuarios es todavía estimable, siempre y cuando la red Bluetooth se reconfigure. Trataremos este aspecto en la sección 3.3.2.

## 3.2. Protocolos de la Red Bluetooth

Describiremos los protocolos diseñados para la auto-configuración de la red Bluetooth y los utilizados para la localización de los móviles y transmisión de dicha localización desde los nodos estáticos hacia sus respectivos nodos maestros.

### 3.2.1. Auto-configuración de la Red Bluetooth

Uno de nuestros principales objetivos es minimizar el trabajo, de forma que la mayor parte del trabajo de configuración recaiga sobre la propia red. La configuración de la red consiste en el establecimiento de las rutas que permiten a cada nodo estático determinar hacia qué enlace debe encaminar sus paquetes de

[7] T. Yamasaki, M. Kishimoto, N. Komoda y H. Oiso. An information o_ering system for exhibition explanation by Bluetooth technology. En *Proc. SSGRR 2001, Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, 2001.

localización en la fase normal de operación. Para ello, cada nodo estático mantiene una tabla donde almacena tuplas (SN*x* _ *y*, distancia). Esta tabla se ordena en orden creciente de distancia. Así, cuando un nodo tenga que envíar un paquete hacia su nodo maestro, elegirá el enlace que lo conecte con el nodo estático especificado en la cima de la tabla.
Ilustraremos la creación de la tabla de encaminamiento con un ejemplo. En la figura 3.4 se muestran los primeros pasos de la configuración.

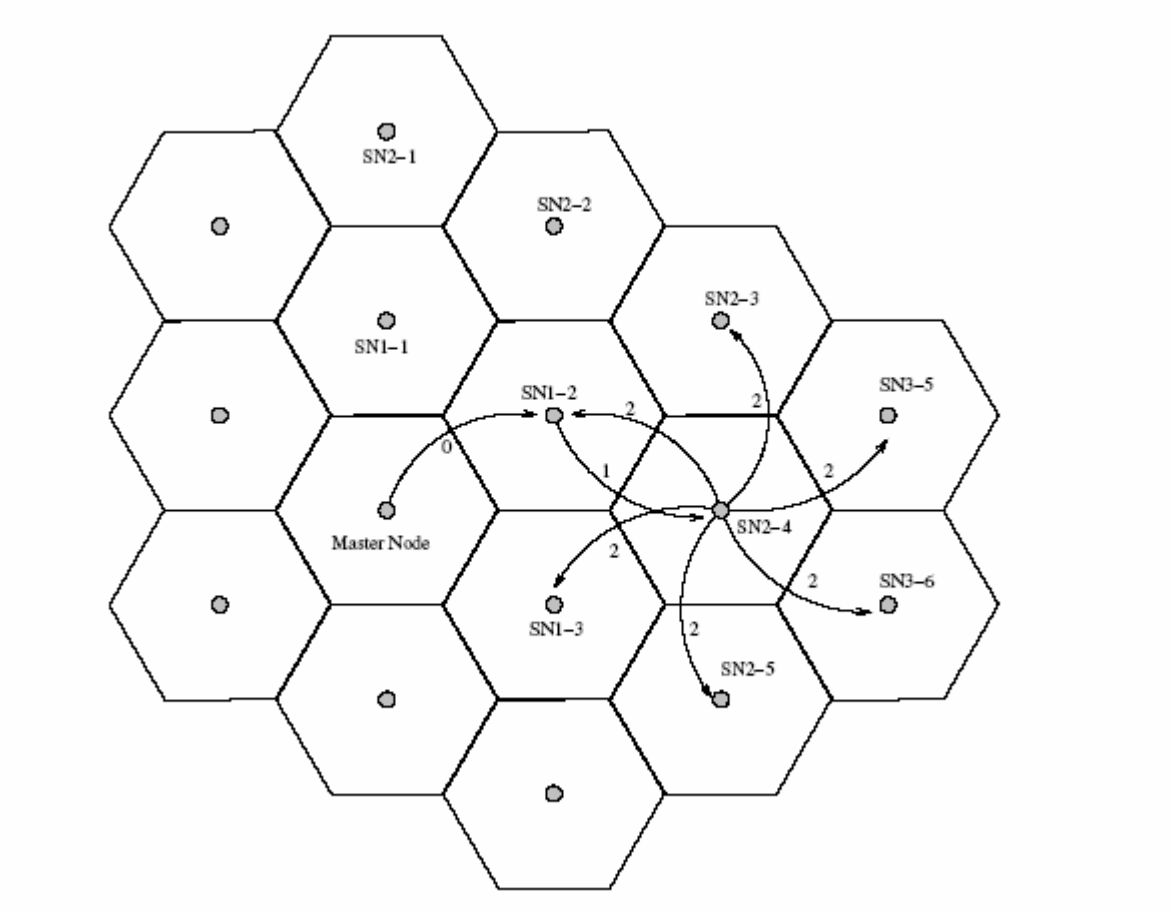



Figura 3.4: Configuración de la Red Bluetooth

La diferencia entre nodo estático y nodo maestro radica sólo en la tabla que deben cargar al activarse. Los nodos maestros inician su tabla con distancia mínima a sí mismos, 0. Los nodos estáticos, por el contrario, inician sus tablas con una distancia infinita (con la representación adecuada). En cuanto se activa cualquier nodo estático, MN o SN, comienza a efectuar los ciclos de *inquiry*. Estos ciclos descubren otros módems, en nuestro caso MNs, SNs o móviles. Cuando un módem en estado *Inquiry Scan* detecta una señal de *inquiry*, debe contestar con un paquete *FHS* que transportará, entre otras cosas, su propia dirección.

Aunque la fase de configuración no persigue detectar los móviles, utiliza los mismos mecanismos que la fase de detección para encontrar los módems Bluetooth de los nodos estáticos. Se establece un intercambio ad-hoc para

establecer que un módem detectado es realmente otro nodo estático y no un dispositivo que no forme parte de la infraestructura de la red Bluetooth (una impresora o un móvil, por ejemplo). Si un nodo estático *a* detecta un *inquiry* de otro nodo *b* y *b* no está en la tabla de encaminamiento de *a*, éste último construye un *paquete de distancia mínima* hacia su nodo maestro y se lo entrega a *b*. Es aquí donde se efectúa el intercambio de comprobación, ya que si *b* no forma parte de la red Bluetooth no entenderá el paquete de distancia y no acusará de recibo. El componente más importante del payload del paquete es la distancia mínima de "*a"* a su nodo maestro. La dirección Bluetooth de *a fi*gura en el propio header del paquete. Cada vez que un nodo estático recibe un paquete de distancia, debe consultar en su tabla de encaminamiento su distancia mínima al nodo maestro y compararla con la recibida. Si la distancia recibida es menor que la mínima actual, se deberá:

1. Modificar la tabla de encaminamiento para reflejar que el mejor camino pasa ahora a través del nodo estático que la transmitió.

2. Notificar la nueva distancia mínima a todos los nodos estáticos contenidos en la tabla[8].

Veamos el ejemplo de la figura 3.4. Supongamos que en un determinado instante se activa el MN (*Master Node* en la figura). Al terminar la iniciación de su tabla de encaminamiento, empieza a emitir y recibir *inquiries*. Por ejemplo, cuando el nodo maestro recibe un *inquiry* de SN1-2, busca la dirección de SN1-2 en su tabla de encaminamiento. Como inicialmente no la encuentra, construye un paquete de distancia (con distancia mínima 0) y se lo entrega a SN1-2. Supongamos que SN1-2 y SN2-4 ya se han descubierto y han intercambiado sus distancias mínimas ( ), lo que parece irrelevante, pero posibilita que cada uno tenga constancia de la existencia del otro. Supongamos también que SN2-4 ya conoce las direcciones de todos sus restantes vecinos. Cuando SN1-2 recibe el paquete de distancia del nodo maestro, compara su distancia mínima con la del paquete. Como la distancia recibida es menor, actualiza su tabla de encaminamiento (con la distancia recibida más 1, debido al salto adicional necesario para llegar hasta él) y construye a su vez un paquete de distancia mínima que enviará a todos los demás módems que le consten, y por tanto al propio nodo maestro y a SN2-4. La tabla 3.1 es una visión de la tabla de encaminamiento de SN2-4 antes y después de la llegada del paquete de distancia mínima.

Cuando el paquete de distancia llega a SN2-4, SN2-4 detecta que la distancia mínima que transporta (1) es menor que la propia, y actualiza su tabla en consecuencia. Como la distancia mínima de SN2-4 cambia, SN2-4 construye un

[8] En una versión previa proponíamos que se enviase una distancia infinita a los nodos circundantes situados a lo largo de un camino de longitud igual a la nueva distancia mínima, emulando el algoritmo *split horizon* [Tan97]. Sin embargo, un estudio con Spim/Promela reveló que pueden producirse lazos ante patrones específicos de fallos en nodos vecinos. Observación de J. Fernández Iglesias, envíada a la conferencia FME 2003.

nuevo paquete de distancia (2) que transmite hacia sus vecinos. Estos cambios se van propagando por toda la red hasta que se alcanza un estado de reposo. Los cambios también se producen ante la caída de un nodo estático. Los vecinos del nodo afectado detectan la caída al realizar *inquiries* que no reciben respuesta, y actualizan sus tablas de encaminamiento en consecuencia. Si el nodo caído está en la ruta de distancia mínima de uno de sus vecinos hacia el nodo maestro, dicho vecino deberá elegir la siguiente alternativa de su tabla y comunicar a su vez el cambio a sus propios vecinos. Si la tabla contiene una alternativa de longitud igual a la distancia (mínima) de la ruta afectada, el vecino simplemente retira al nodo caído de la lista. La capacidad de supervivencia ante fallos de la red se analiza en la sección 3.3.2.

### 3.2.2. Protocolo de Localización

El objetivo principal de la red Bluetooth es realizar el seguimiento de los móviles en el recinto de interés. Para cumplir ese objetivo, todos los nodos estáticos deben emitir periódicamente *inquiry* y recolectar las respuestas (además de realizar ciclos de *inquiry*, los nodos estáticos inicia periódicamente ciclos de transmisión de datos y de *inquiry scan* para ser detectados por otros nodos estáticos).

Cada nodo estático tiene una *caché de localización*, donde se guardan las direcciones Bluetooth de las respuestas. Además de dichas direcciones Bluetooth, se mantiene un campo *detectado* y otro *nuevo*. Estos campos se emplean para minimizar el número de mensajes que se transmiten hacia el nodo maestro, con el fin de no cargar la red con mensajes innecesarios. Cuando se detecta un nuevo móvil, se construye un *paquete de localización*. Este paquete contiene las direcciones Bluetooth del nodo estático detector y el móvil (64+64 bits) y un *byte* de control.
Es necesario añadir la dirección del nodo estático detector, puesto que el paquete recorrerá varios nodos estáticos a lo largo de su ruta, y la posición del móvil se aproximará por la posición del nodo estático detector. Existe la posibilidad de construir paquetes a nivel de aplicación que contengan información de varios móviles, en vez de envíar un paquete por móvil detectado. Con esta estrategia se reduce el retardo medio de transmisión y se aumenta la velocidad de transmisión, ya que los paquetes *DM5* se transmiten a mayor tasa que los *DM1* en escenarios asimétricos como el nuestro (unidireccional)[9].

Por ejemplo, la segunda y tercera columnas de la tabla 3.2 muestran una representación del contenido de un caché de detección al comienzo del ciclo de *inquiry* (en ese momento, todas las columnas se ponen a *NO*). Las columnas cuatro y cinco muestran el estado al finalizar el ciclo de *inquiry*. Durante el ciclo, cuando se recibe un paquete *FHS*, se lee su dirección Bluetooth. Si dicha dirección ya figura en el caché, se pone la columna *Detectado* correspondiente a

[9] M. Leopold. Evaluation of Bluetooth communication: Simulation and experiments. Technical report 02/03. Department of Computer Science, University of Copenhagen.

*SI*. Si la dirección no figura, se activan las columnas *Detectado* y *Nuevo*. Cuando el ciclo finaliza, se construye un paquete de localización para las direcciones cuya columna *Nuevo* esté a *SI* (móviles que han entrado en el rango del nodo estático) y para aquellas cuya columna *Detectado* contenga un *NO* (móviles que abandonan el rango del nodo estático)[10]. La diferencia radica en el *byte* de control asociado a cada móvil, donde se activa o desactiva un bit dependiendo de que se produzca una u otra situación.

En el ejemplo de la tabla 3.2, al finalizar el ciclo de *inquiry*, se construirá un mensaje de nivel de aplicación formado por la dirección del nodo estático y tres tuplas (*control, dirección móvil*). Por ejemplo, la primera de ellas puede tener la dirección de la etiqueta BD-19 y el *nuevo* bit activado en su

| | Antes del *inq.* | | Después del *inq.* | |
|---|---|---|---|---|
| Dir. Bluetooth | Detectado | Nuevo | Detectado | Nuevo |
| BD-3 | NO | NO | SI | NO |
| BD-7 | NO | NO | NO | NO |
| BD-11 | NO | NO | NO | NO |
| BD-13 | NO | NO | SI | NO |
| BD-17 | NO | NO | SI | NO |
| BD-19 | | | SI | SI |

Tabla 3.2: Caché de detección de un SN

Campo de control. Las dos siguientes pueden informar que BD-7 y BD-11 han dejado la zona de detección. La figura 3.5 muestra la posible representación del mensaje construido. En ella, el *nuevo* bit es el primero del campo de control. Las PDUs de la capa L2CAP soportan payload de hasta 65.535 octetos, por lo que el tamaño de nuestro mensaje no está restringido en la práctica.

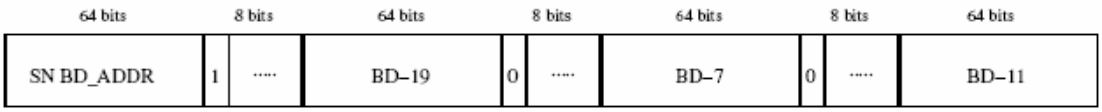



Figura 3.5: Mensaje de localización

Debemos indicar que cualquier módem Bluetooth que se encuentre en el rango de un nodo estático podría responder a los *inquiries*, aunque no sea un móvil. Existen dos casos de este tipo:

[10] Si bien los paquetes de no-detección no son estrictamente necesarios para determinar la localización del móvil, refuerzan la validez, puesto que los nodos estáticos implicados han detectado al móvil en algún momento.

1. El paquete *FHS* de respuesta proveniente de otro nodo estático. Antes de actualizar su caché de detección, el nodo estático receptor comprobará su tabla de encaminamiento. Si la dirección del paquete *FHS* aparece en ella, el paquete es la respuesta de otro nodo estático. De hecho, este tipo de intercambios periódicos es fundamental para el funcionamiento del protocolo de configuración, puesto que implica que el nodo estático emisor sigue activo (cada entrada de la tabla de encaminamiento tendrá un campo adicional que se actualizará cada vez que detectemos el nodo estático correspondiente).

2. El paquete *FHS* de respuesta proviene de un dispositivo extraño. El nodo estático no tiene forma de saberlo, por lo que anota la dirección del paquete en su caché de detección y, cuando termina el ciclo de *inquiry*, envía un paquete de localización hacia su nodo maestro. El servidor de localización asociado al nodo maestro comprobará que la dirección no pertenece a un móvil registrado, por lo que se limitará a descartar el evento. Esto podría parecer un defecto, pero los nodos estáticos detectores generarán un paquete *único* de localización si el dispositivo que emitió el paquete *FHS* no se desplaza (la caché no volverá a registrarlo como *nuevo*). De esta forma, los módems Bluetooth de periféricos de ordenador no suponen un problema.

### 3.3.1. Escalabilidad

Las figuras 3.3 y 3.4 muestran una configuración con un nodo maestro único. Como se ha dicho anteriormente, el protocolo de configuración admite más de un nodo maestro por red Bluetooth. Esto hace que se creen redes Bluetooth independientes sobre una misma región, lo que puede ser aconsejable para zonas muy grandes en las que los paquetes de los nodos estáticos más alejados necesitarían muchos saltos para llegar al nodo maestro. Por consiguiente, se garantiza la escalabilidad de la red Bluetooth siempre y cuando se pueda conectar el número necesario de nodos maestros a los servidores. En las configuraciones donde existan varios nodos maestros se podrá plantear la incorporación de tarjetas inalámbricas de comunicación (por ejemplo, tarjetas IEEE 802.11) que faciliten la interconexión entre los diferentes nodos maestros con el fin de evitar cableado innecesario o en su caso una red LAN.

En el caso de las tarjetas inalámbricas o la red LAN servirían a su vez para establecer la infraestructura de red de datos, utilizada para la transferencia de información contextual desde los servidores de contenidos hacia las terminales de los usuarios. Si se utilizasen tarjetas IEEE 802.11b, se tendría que prestar especial cuidado a las interferencias que provocan en los dispositivos Bluetooth[11], y por tanto sería recomendable seguir las propuestas de coexistencia planteadas en trabajos como “Mobilian Corporation. WiFi (802.11b) and Bluetooth”. Otra posibilidad sería emplear tecnologías que trabajen en frecuencias distintas a las

[11] En realidad, los dispositivos IEEE 802.11b se ven más afectados por la presencia de dispositivos Bluetooth que viceversa [GVDS01].

de Bluetooth (IEEE 802.11a, por ejemplo), para así eliminar cualquier tipo de interferencia. Hoy en día existen equipos duales que integran ambas tecnologías en un sólo dispositivo[12].

El protocolo descrito en la sección 3.2.1 hace que los nodos estáticos se asocien al nodo maestro más próximo, a quien transmitirán sus paquetes de localización. Posteriormente, en caso de fallos, puede ocurrir que las rutas cambien y que un nodo estático se asocie a un nodo maestro distinto al original. Por consiguiente, además de disminuir el número de saltos y la carga por nodo maestro, añadir nuevos nodos maestros mejora la robustez de la red de localización, ya que aumenta el número de rutas posibles desde un nodo estático a cualquier nodo maestro.

En principio, para una estructura hexagonal, una buena solución es agrupar las celdas en macro-celdas hexagonales, en cuya celda central habría un nodo maestro. Esta estrategia reparte el plano por igual entre los nodos maestros. Sin embargo, en un escenario irregular o con obstáculos a la propagación, las macro-celdas podrían tener cualquier forma. El estudio de este tipo de situaciones no se tratara en este trabajo.

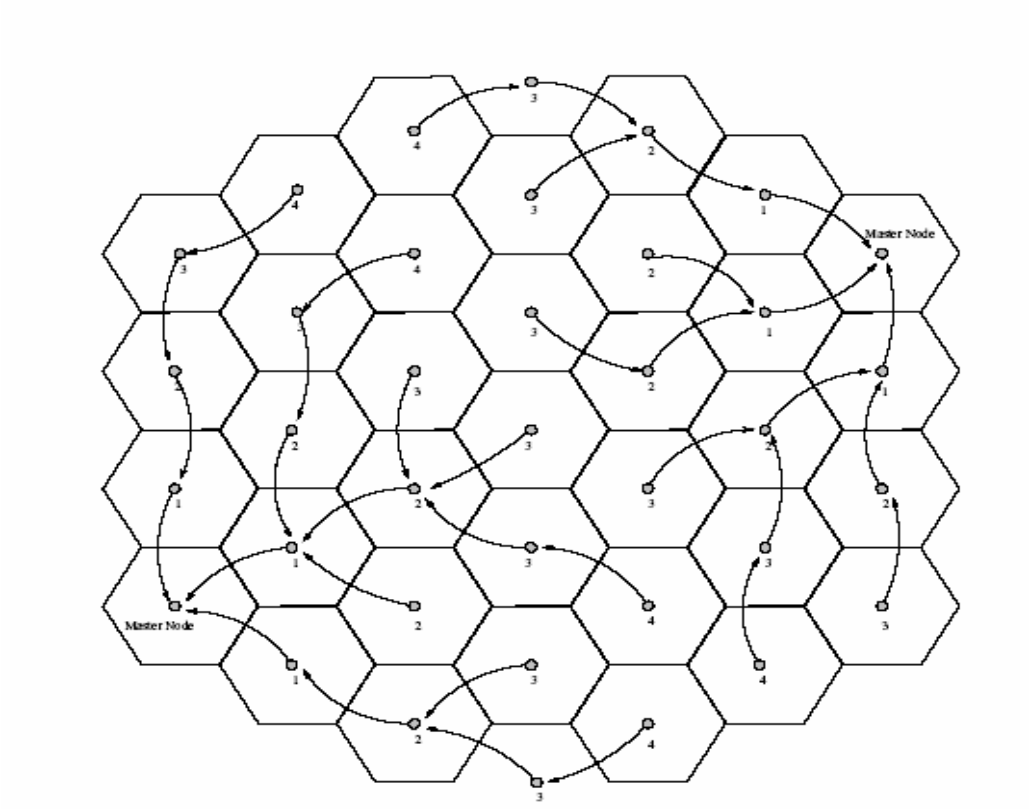



Figura 3.6: Rutas de encaminamiento con dos MNs

[12] Possio. Wireless services gateway PX30. http://www.possio.com, 2003.

Las figuras 3.6 y 3.7 muestran posibles configuraciones en una red Bluetooth de tres capas con dos y cuatro nodos maestros situados en los extremos, respectivamente (macro-celdas de radio de 3-4 celdas).

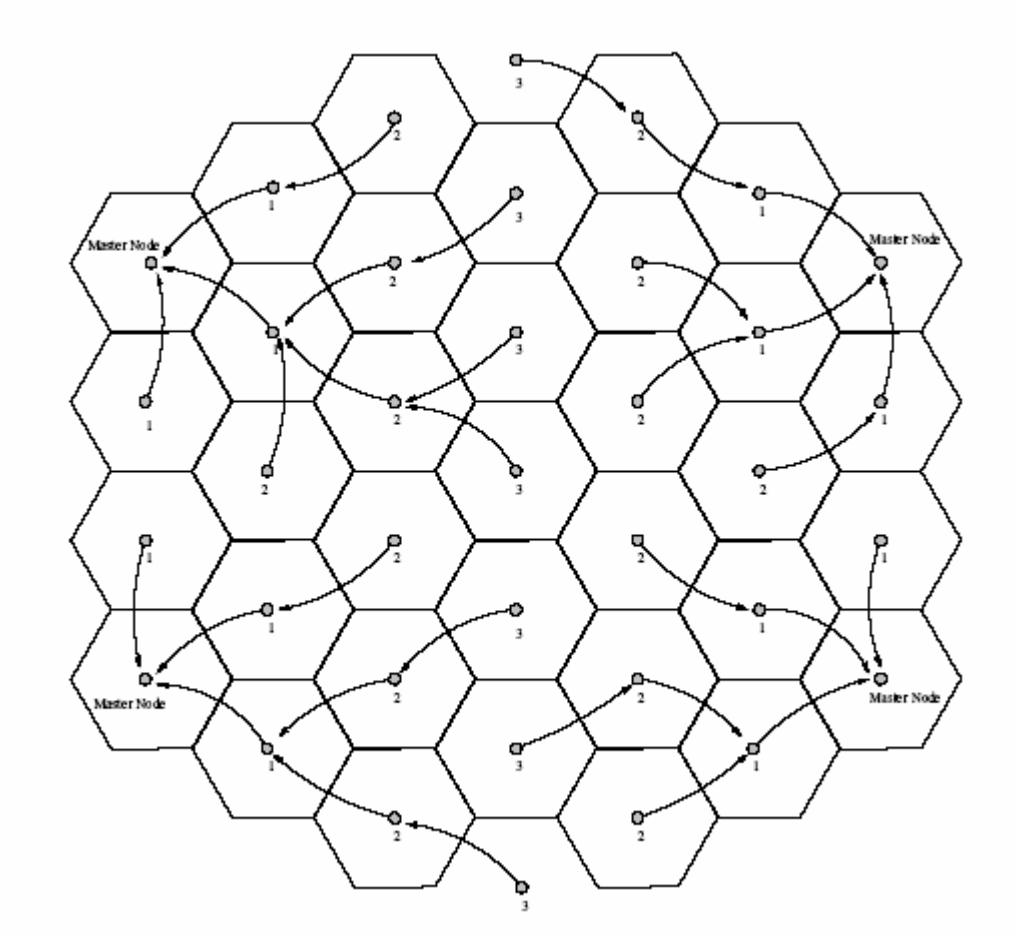



Figura 3.7: Rutas de encaminamiento con cuatro MNs

Es importante resaltar que, si se cambia la variable aleatoria que rige el instante de inicio de cada nodo, el resultado puede ser diferente. La tabla 3.3 muestra el número medio de nodos asociados a cada nodo maestro y la media de las distancias mínimas de los nodos estáticos, a través de las diversas simulaciones.

| MN | $\overline{SN}_{MN}$ | $\overline{d}_{MN}$ |
|---|---|---|
| 1 | 36 | 2.33 |
| 2 | 17.5 | 2.54 + (-0.06,0.09) |
| 4 | 8.25 | 1.71 + (-0.21, 0.38) |

Tabla 3.3: Evaluación de simetría

En la tabla 3.3, *S NMN* es el número medio de los nodos estáticos asociados a un nodo maestro. Nótese que es inversamente proporcional al número de nodos maestros. La columna *dMN* muestra la media de las distancias mínimas de los nodos estáticos y su variación. Es interesante observar que, al añadir un segundo nodo maestro, la distancia mínima aumenta. Si nos fijamos en la figura 3.6, podemos ver que hay nodos estáticos cuyos paquetes de localización necesitan cuatro saltos para llegar a un nodo maestro.

En una configuración con el mismo número de capas pero sólo un nodo maestro, el número máximo de saltos es de cuatro. Podría parecer que un segundo nodo maestro degrada las prestaciones, pero esto es sólo aparente, ya que hacia cada uno de ellos convergerá la mitad del tráfico total de localización, *que no varía*. Sin embargo, al añadir dos nodos maestros más, incluso la distancia mínima disminuye, puesto que no existe ningún nodo estático con distancia mínima mayor que 3, y se reduce el número de nodo estático con esa distancia.

### 3.3.2. Supervivencia

Es importante que la red de localización siga prestando sus servicios ante caídas de nodos estáticos, hasta que se reparen. Por lo tanto, no se debe imponer un costo de mantenimiento elevado, lo que implica que se hable de tardar días en reemplazar un nodo estático averiado. Mientras tanto, el propietario de la red deseará que, aunque el funcionamiento sea sub-óptimo, la localización y seguimiento de móviles se siga realizando.

En consecuencia, exigimos como requisito que la red Bluetooth se pueda reconfigurar ante fallos de nodos estáticos. Creemos que el fallo más probable sin necesidad de reparación inmediata será una caída de un nodo estático individual. Las caídas en *cadenas* de nodos estáticos serán normalmente debidas a fallos de alimentación, que se suelen reparar con prontitud.

Si la reconfiguración ante fallos no se llevase a cabo, se podrían perder paquetes de localización: si un nodo estático estuviese en la cima de su tabla de encaminamiento de un nodo estropeado, los paquetes no llegarían a su destino. Por tanto, es necesario elegir un nuevo nodo estático al cual envíar los paquetes de localización. Como comentamos anteriormente, en una red sometida a fallos con varios nodos maestros, los paquetes de localización emitidos por un nodo estático determinado puede acabar en distintos nodos maestros. Esto no es un problema, puesto que asumimos que todos los nodos maestros están conectados al mismo servidor de localización. Naturalmente, los fallos en dichos servidores o en los propios nodos maestros son inaceptables, ya que inhabilitan la red de localización. Por ello, estas entidades tienen que estar protegidas (mediante redundancia, por ejemplo).

Gracias al protocolo propuesto para la configuración inicial de la red Bluetooth, la reconfiguración tiene lugar de forma casi automática. Como sabemos, los nodos estáticos realizan ciclos periódicos de *inquiry* para propósitos de localización.

Además, conocen la distancia en número de saltos a algún nodo maestro a través de cualquiera de sus vecinos, en una lista en orden ascendente. Un nodo estático detecta que uno de sus vecinos ha caído si dicho vecino ignora repetidamente los *inquiries*. Si se trata del vecino por el que enviaba sus paquetes (primero de la ruta de distancia mínima hacia su nodo maestro), el nodo estático elige el siguiente nodo de su lista de encaminamiento (la siguiente ruta). Evidentemente, en el caso extremo en el que se caigan todos, el nodo estático queda *aislado* de cualquier nodo maestro.

La capacidad de supervivencia se puede evaluar de muchas maneras. Una de ellas es comprobar que cuando un nodo estático se ve afectado por un fallo, puede descubrir una ruta alternativa hacia un nodo estático si dicha ruta existe[13]. Otro criterio más amplio consiste en medir la resistencia ante fallos de la capacidad de detección de móviles. En principio, un número reducido de fallos no debe afectar a la capacidad de detección (aunque si disminuirá la precisión de localización en el área del problema). Según los fallos vayan aumentando, se puede llegar a un punto en el que se creen zonas de sombra en las que los móviles son indetectables.

### 3.3.3. Rendimiento del Ciclo de Inquiry

El rendimiento del ciclo de *inquiry* es fundamental a la hora de determinar la eficiencia de la red Bluetooth. La capacidad de detección de los nodos estáticos puede saturarse si el número de móviles es muy elevado, lo que produciría errores y, por consiguiente, un posible descenso de la precisión.

El estado inquiry permite iniciar la red, reconfigurarla en caso de fallos y detectar los móviles. Cuando un nodo estático realiza *inquiries*, tanto sus vecinos como los móviles circundantes responden con paquetes *FHS*. La recepción de la respuesta no es inmediata, ya que los módems Bluetooth en estado *Inquiry Scan s*ólo escuchan en determinados instantes. Por tanto, algunos móviles podrían no responder a los sondeos de los nodos estáticos si la duración del ciclo de *inquiry* es muy pequeña. De hecho, en Bluetooth se recomienda que dure 10.24 segundos al menos. Esto garantiza que se sondean todas las frecuencias, pero no que se reciban las respuestas de todos los móviles Bluetooth situados en la zona de cobertura. Debemos determinar cuantos móviles quedan sin detectar en media en un ciclo de *inquiry*. Encontramos un error que provocaba una salida anormal de la ejecución cuando se introducían más de ocho dispositivos Bluetooth. El error se debía a que no se reservaba memoria al llegar a ese límite, posiblemente porque

[13] F.J. González-Castaño y J.J. García-Reinoso. Survivable Bluetooth location networks. En *Proc. IEEE International Conference on Communications2003 (ICC 2003)*, Anchorage, EEUU, may. 2003

se suponía que no tenía sentido sobrepasar la cota de conexiones activas. Evidentemente, esto no es realista, porque dicha cota se refiere a conexiones activas *simultáneas.*

Por otro lado, la cobertura del módem Bluetooth de cada nodo estático que es de 10 metros de radio, lo que equivaldría a un círculo con un área de 314.16 $m^2$. Si dividimos esta superficie por el número de usuarios que se pueden detectar sin problemas, a cada uno de ellos le correspondería un espacio de 314,16 / 49 = 6,41$m^2$, es decir, un circulo de únicamente 1,43 m de radio. Evidentemente, en los escenarios de aplicación esto supondría que los recintos están abarrotados, y por tanto creemos que el número de usuarios no está limitado en la práctica.

### 3.3.4. Retardo de transmisión

Por retardo de transmisión nos referimos al tiempo necesario para transmitir los paquetes de localización desde un nodo estático hasta su destino final, el nodo maestro más próximo. Del modelado del comportamiento de los usuarios en un escenario resultan plazos, durante los cuales las estimaciones de posición siguen siendo válidas. Finalizados dichos plazos, el usuario habrá cambiado de contexto y la información proporcionada por los servidores no tendrá ningún valor. El lector reconocerá fácilmente que, por lo tanto, la red de localización puede considerarse un *sistema de tiempo real*.

A fin de comparar los plazos de determinación de posición con las prestaciones de la red Bluetooth, debemos sumar el tiempo de detección de la sección 3.3.3 y el retardo de transmisión. Este retardo depende de varios factores: posición del nodo estático detector en la red de localización, número de nodos maestros en la instalación, carga de la red en cualquier instante, etc.

Por ello, una aproximación analítica no es inmediata. En el peor caso, un nodo tiene que soportar el tráfico de sus seis vecinos (caso del nodo maestro central).

Debemos resaltar que en todos los casos, los módems Bluetooth están siempre en cualquiera de estos tres estados: *inquiry*, *inquiry scan* o transferencia de datos.

Como vimos en la sección 3.2.2, se generan mensajes de nivel de aplicación que transportar información de localización de más de un dispositivo. Un paquete *DM5* puede transportar información de localización de varios móviles. Del payload, 8 *bytes* identifican al nodo estático detector y 24 x 9 *bytes* se utilizan para codificar posiciones: en cada posición, 8 *bytes* codifican la dirección Bluetooth del móvil y el *byte* restante se usa para control (ver figura 3.5). En condiciones normales habrá menos móviles en la vecindad de cualquier nodo estático (más de 13 m2 libres por usuario), y por tanto dicho nodo *sólo tendrá que emitir un paquete de localización por ciclo*.

En el sector de red Bluetooth de la figura 3.10, para la simulación con siete nodos se activan los nodos del 0 al 6; para la simulación con ocho nodos se activan los nodos 0 al 7, y así sucesivamente.

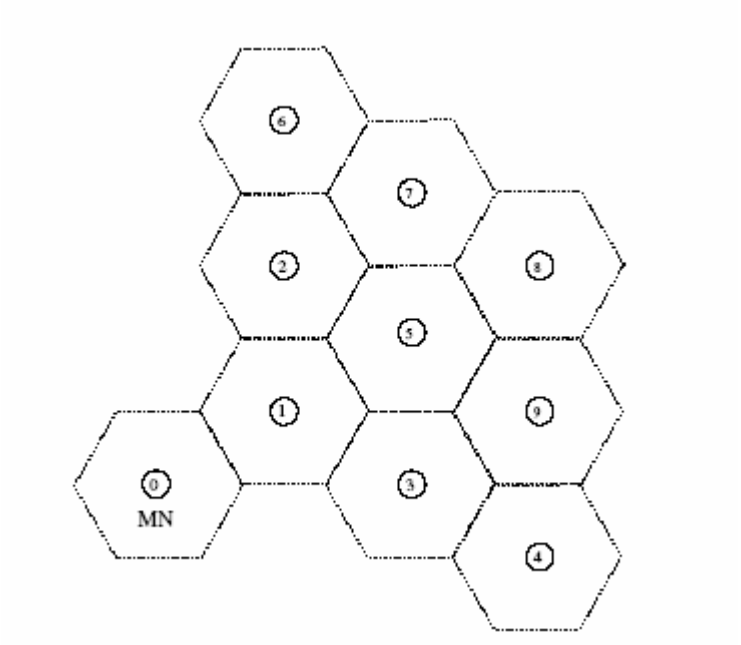



Figura 3.10: Sector de BLN simulado con Blueware

El sector de la red Bluetooth de la figura 3.10 corresponde a la peor configuración que puede soportar un nodo de la primera capa (recibe tráfico de un número máximo de nodos).

El retardo de los paquetes depende del número de nodos considerados. Parecería lógico pensar que los *inquiries* del nodo 7 no repercuten en las transmisiones del Nodo 4 al 0. Debemos recordar que los nodos tienen que responder a esos *inquiries*, por lo que se pierde más tiempo cuantos más nodos realizan esa acción.

Asimismo, pasado un cierto tiempo, las configuraciones con nueve y diez nodos sufren un retardo similar.

En la zona de estabilidad, el retardo es inferior para cualquier configuración.

De estos resultados se deduce que, en un escenario con muchos usuarios, el retardo no sobrepasa los 30 segundos una vez alcanzada la zona de estabilidad. Si añadimos el tiempo de detección de la sección 3.3.3, obtenemos que el plazo mínimo exigible por la aplicación será de unos 35 segundos. Si lo comparamos con el comportamiento de un peatón que hace paradas frecuentes para consultar su PDA (y no puede caminar mientras lee), el plazo parece poco relevante. Además, en el apartado 3.3.1 ya indicamos que es conveniente tener más de un nodo maestro en la red Bluetooth. Refiriéndonos a la tabla 3.3, vemos que para una red de tres capas el número medio de saltos es 1,71 para cuatro nodos

maestros En la figura 3.7, de los 37 nodos estáticos que hay en total, tan sólo siete están a distancia 3 de su nodo maestro.

Un nodo no acepta paquetes de datos mientras realiza un ciclo de *inquiry*, por lo que dichos paquetes deben esperar en la cola del nodo emisor. Para disminuir el retardo se podría recurrir a nodos estáticos con *dos* módems Bluetooth: uno para realizar tareas de *inquiry* y otro dedicado al encaminamiento de los paquetes de localización. Los protocolos variarían muy poco. Básicamente, el módem encargado de realizar los *inquiries* tendría que comunicar a los nodos vecinos la dirección Bluetooth de su compañero además de la propia, para que los paquetes de localización se enviasen al módem dedicado al transporte (existirían dos redes: de localización y de transporte de paquetes de localización). En el caso de optar por esta solución, se podría pensar que la interferencia provocada por dos p*iconets* simultáneamente activas supondría un gran inconveniente. A diferencia de la coexistencia de dispositivos Bluetooth e IEEE 802.11b, en la que la interferencia puede degradar de forma significativa el rendimiento si no se toman las medidas oportunas, la interferencia entre dos *piconets* Bluetooth es inferior debido a la baja probabilidad de que dos integrantes de *piconets* diferentes emitan en la misma frecuencia[14].

### 3.3.5. Estudio con Bluetooth

Como hemos visto anteriormente, Bluetooth nos impone un límite de siete conexiones simultáneas de esclavos activos. Se podría pensar que la única forma de soportar digamos 250 participantes, es crear una gran *scatternet*[15] para este número de dispositivos. Para ello, se podría aplicar un algoritmo de establecimiento de *scatternets* como TSF [16] . Sin embargo, además de la complejidad resultante de un número de nodos tan elevado, un algoritmo de conexión *espontánea* no garantizaría la comunicación entre un nodo cualquiera y la estación base (ver sección 3.3.4).

En consecuencia, hemos estudiado una forma de evitar el límite de siete esclavos activos, y hemos llegado a la conclusión de que, utilizando el modo *park* de Bluetooth, no es necesario que las 250 terminales estén activas simultáneamente En las especificaciones Bluetooth se indica que, cuando un esclavo no necesita participar en su *piconet*, pero quiere seguir sincronizado con el maestro, puede entrar en el modo *park* de bajo consumo y actividad mínima. Este modo prolonga la vida de las baterías, lo que resulta muy beneficioso para la terminal móvil. De todas formas, lo más importante para nosotros es que el número de esclavos en

[14] Bluetooth usa un esquema FHSS donde se realiza un salto de frecuencia para cada transmisión.

[15] Y por tanto implementar el sistema como una red móvil ad-hoc.

[16] T. Godfrey y J. Guttag. A locally coordinated scatternet scheduling algorithm.En *Proc. IEEE Conference on Local Computer Networks*, 2002.

modo *park* únicamente está limitado por el rango de las direcciones Bluetooth y, por consiguiente, no está limitado a efectos de nuestra aplicación, en la práctica.

En “Bluetooth-based high-performance LAN access point incorporating
Múltiple radio unita” se propone el uso del modo *park* para implementar un punto de acceso a Internet desde múltiples terminales (con varios módems actuando como maestros). Sin embargo, no existen propuestas previas en el contexto de las aplicaciones de telecomunicaciones con restricciones de tiempo real.

## 4. Subestado *park* Bluetooth

Cuando uno o más esclavos pasan a modo *park*, el maestro establece un beacon channel (figura 4.1) para:

Transmitir paquetes maestro-esclavo para que los dispositivos en modo *park* se Sincronicen.

Transmitir paquetes para cambiar los parámetros del beacon channel en los esclavos en modo *park*.

Transmitir mensajes *broadcast* con cualquier propósito hacia los esclavos en modo *park*.

*Unpark* uno o más esclavos en modo *park*.

La figura 4.2 muestra un ejemplo de ventanas de acceso (*access windows*) de un beacon channel adaptado a nuestra aplicación (puede existir cualquier número de ventanas de acceso en una slot del beacon channel incluso ninguna).

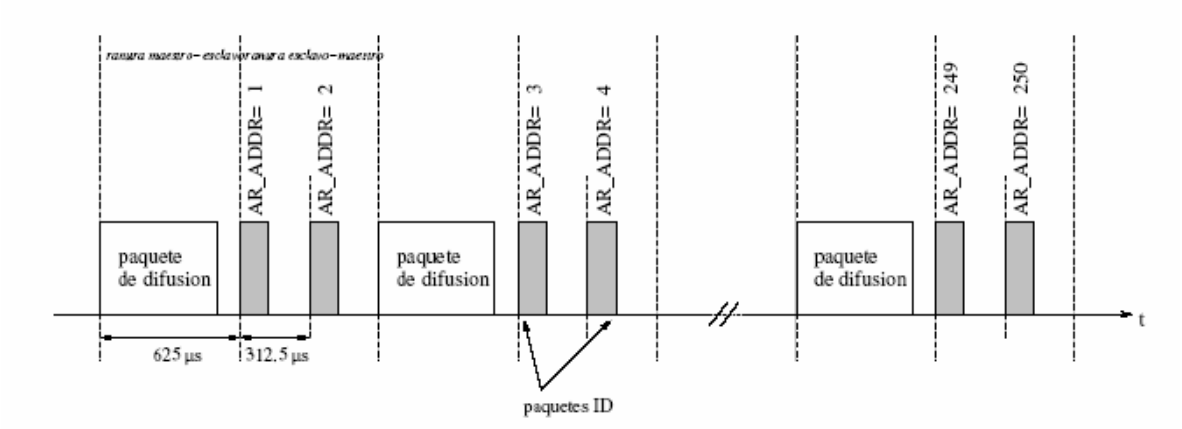



Figura 4.2: Ventanas de acceso (figura tomada de [Bluetooth])

Cada unidad de la ventana de acceso consiste en un slot maestro-esclavo y otro slot con dos accesos esclavo-maestro, potencialmente compartida por varios esclavos. El maestro sondea a los esclavos en el slot maestro-esclavo. Cuando un esclavo entra en modo *park*, el maestro le asigna medio slot para su dirección

AR_ADDR (posiblemente compartida). Un esclavo en modo *park* puede envíar respuestas al maestro en el medio slot que le corresponde, que estará identificada por AR_ADDR. Los esclavos en modo *park* pueden envíar peticiones de tipo *unpark* en las slots temporales que les corresponden.

# 5 Servicios Propuestos para la Arquitectura m-Mall

La arquitectura m-Mall[17] consistía en la combinación de una versión inicial de la red Bluetooth y una serie de servicios asociados dependientes del contexto. La red Bluetooth ha sido suficientemente descrita. Aquí presentamos algunos servicios posibles. Dichos servicios se basan en tres características básicas de m-Mall:

Los usuarios portan un terminal móvil con conexión IP o WAP.
El sistema m-Mall conoce la ubicación de dichos usuarios en tiempo real
El sistema m-Mall posee perfiles de los usuarios para ajustar los servicios ofrecidos.

Los servicios asumen la existencia de perfiles particulares de usuario. El perfil se compone de datos personales, preferencias, especificaciones de terminal, localización, historia de compras y enlaces con otros perfiles (miembros de una familia, por ejemplo).

Hemos clasificado los servicios m-Mall en dos categorías principales dependiendo de quién inicia la comunicación (el usuario o el propio sistema m-Mall). Los anuncios, las notificaciones, etc. son ejemplos de servicios iniciados por el sistema. Los usuarios pueden iniciar servicios tales como guiado por recintos, búsqueda de productos, etc.

## 5.1 Servicios Iniciados por el Sistema

### 5.1.1 Publicidad.

En vez de realizar envíos indiscriminados de publicidad, que pueden cansar a los usuarios y generar rechazo, es posible seleccionar direcciones dependiendo de los perfiles individuales y/o posiciones actuales dentro del recinto. Por ejemplo, se puede utilizar la historia de compras para envíar anuncios en sintonía con las últimas adquisiciones.
También se podría informar de las ofertas especiales de comidas a los usuarios situados en la zona de restaurantes. Estos criterios se pueden combinar con los

[17] J.J. García-Reinoso, J. Vales-Alonso, F.J. González-Castaño, L.E. Anido-Rifón y P.S. Rodríguez-Hernández. A New m-Commerce Concept: m-Mall. *Lecture Notes in Computer Science*, 2232:14.25, 2001.

datos de preferencia de los usuarios (por ejemplo, si el restaurante ofrece comida rápida, china o italiana).

### 5.1.2 Notificaciones.

Se pueden envíar notificaciones generales a las PDAs de los clientes (por ejemplo, hora de cierre del local). Los perfiles y localizaciones se pueden utilizar para filtrar las notificaciones específicas. También se pueden realizar notificaciones dependientes de la localización (por ejemplo, a 10 minutos para cerrar se enviaría sólo a las personas situadas a más de 200 metros de las salidas). Finalmente, se pueden realizar combinaciones de perfiles y localizaciones: se informará al usuario X que una estrella del fútbol firma autógrafos en su entorno actual.

### 5.1.3 Servicios Generales.

Que no encajan en las categorías anteriores. Por ejemplo, balance de crédito, menú al entrar en un restaurante (dependiente de la localización), menú vegetariano al entrar en un restaurante (dependiente de la localización y el perfil), etc.

## 5.2 Servicios Solicitados por el Usuario

### 5.2.1 Servicios de Búsqueda.

Para localizar un producto particular en las tiendas del recinto. Las búsquedas pueden ser generales o dependientes del perfil y/o la localización. Una búsqueda basada en el perfil utiliza los datos de preferencia del cliente (por ejemplo, las búsquedas de libros del usuario X mostrarán solo los relacionados con arte, pintura y ciencia ficción, etc.). Las búsquedas dependientes de la localización utilizarán la posición actual de los consumidores (por ejemplo, las búsquedas de restaurantes sólo mostrarán los que se encuentren cerca del usuario). Finalmente, se pueden hacer filtrados múltiples, que den como resultado los restaurantes de comida china, rápida o italiana que se encuentren cerca del usuario X. Figura A.2

### 5.2.2 Sistemas de Compra/Reserva.

Se podrán realizar compras o reservas desde los dispositivos móviles de los clientes. Se pueden incluir sistemas de pre-pago para garantizar las compras.
Para aquellos productos que impongan un tiempo de espera (petición de mesa en restaurantes, por ejemplo), m-Mall podría incluir un sistema de reserva. El sistema generaría las notificaciones oportunas: la mesa está lista, la película comenzará en cinco minutos., etc.

### 5.2.3 Servicios Guiado.

El sistema ofrece un servicio de guiado desde la posición actual del usuario hasta la tienda elegida.
Puede ser activado por una compra de entrada de cine o por una reserva de restaurante, por ejemplo.

## 5.3 Prototipo m-Mall

La arquitectura m-Mall sigue un modelo de tres etapas. (1) La etapa *back-end* almacena todos los datos del sistema: datos de localización dinámica, información de compras, ofertas, perfiles de usuario, etc. (2) La etapa *business logic* se responsabiliza de alimentar a la etapa *back-end* y utiliza sus datos para ofrecer los servicios finales. (3) La etapa *front-end* recibe datos de (2) y los presenta en las interfaces de usuario.

La figura A.2 muestra la arquitectura propuesta. El soporte *middleware* utilizado en New m-Commerce Concept: m-Mall. *Lecture Notes in Computer Science*,
es CORBA OMG. Common Object Request Broker Architecture, debido a su capacidad probada para manejar aplicaciones heterogéneas y a la propia experiencia del grupo investigador.

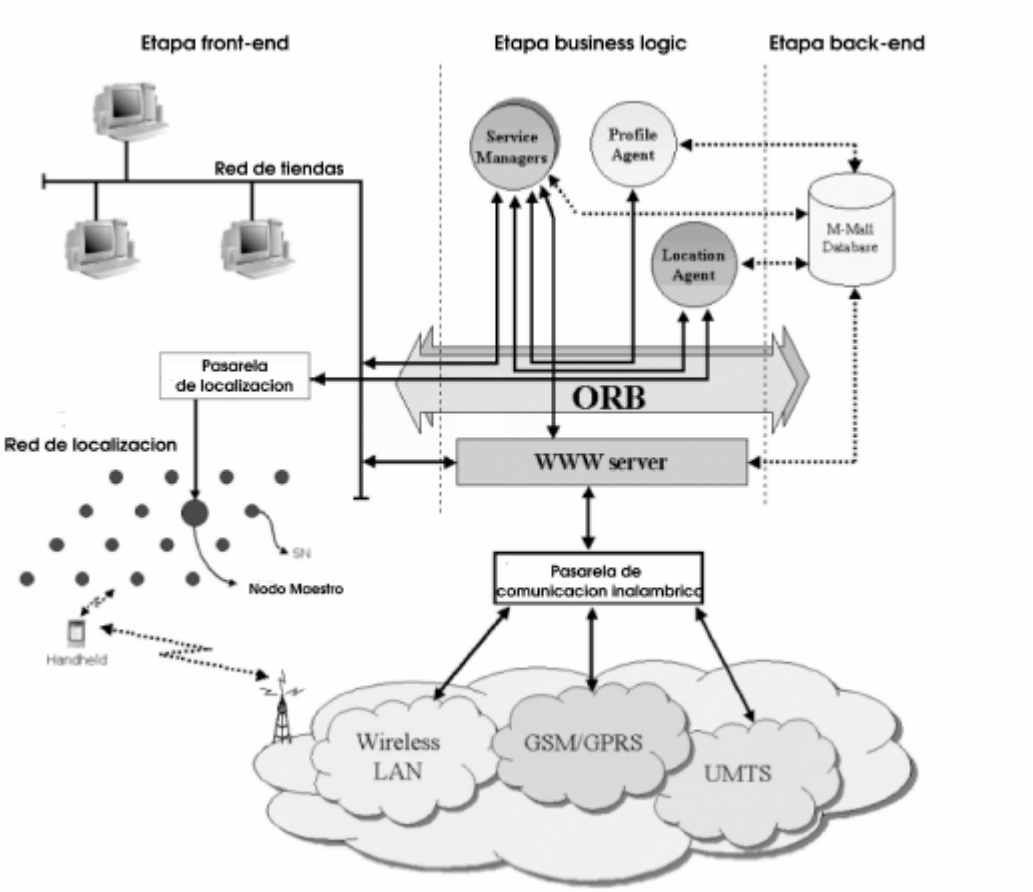


Figura A.2: Arquitectura m-Mall

Las principales entidades m-Mall en la etapa *business logic* son:

El *pro_le agent* administra datos específicos de los usuarios. Los usuarios suministran sus preferencias a través de una interfaz Web. El servidor m-Mall incorpora gradualmente las historias a la información de preferencias, de acuerdo con las acciones del cliente, compras, etc. Los servicios m-Mall solicitan información de perfil al *pro_le agent* a través de la interfaz de servicio pre-definida.

*Location agent*. La red Bluetooth envía información de localización a este agente a través de una pasarela dedicada. El agente realiza la correspondencia entre los datos de localización y la información dependiente de dicha localización. Cuando un usuario entra en una nueva celda de contexto m-Mall (las celdas de la red Bluetooth y las tiendas pueden no corresponder exactamente), el agente de localización envía un evento al agente *service manager* mediante el servicio de eventos CORBA.

Los agentes *service manager* son el corazón del sistema m-Mall. Existe un agente diferente por cada servicio. Cada agente podrá utilizar datos de perfiles o información de localización para aumentar sus capacidades de servicio.

Tanto los accesos de clientes desde sus dispositivos moviles como los accesos de las tiendas desde sus ordenadores son posibles gracias a la combinación de servicios basados en HTTP y las aplicaciones CORBA antes mencionados. Se requerirá por tanto un servidor HTTP y una pasarela para acceder a los objetos CORBA. La infraestructura de datos para los terminales de los usuarios se integra a través de una pasarela dedicada, que estará conectada a la etapa *business logic*.

La interfaz de usuario de la etapa *front-end* es compatible con navegadores Web o *applets* Java en las terminales de usuario, y con *software* propietario en las aplicaciones de las tiendas.

# CONCLUSIONES

Bluetooth es una tecnología inalámbrica concebida para sustituir cables. En algún momento se consideró que se podría utilizar para implementar LANs, pero rápidamente se descartó la idea debido a los problemas de escalabilidad. Sin embargo, para aplicaciones con bajas tasas de transferencia como la propuesta, Bluetooth es una excelente tecnología por sus características de costo y consumo. Además, que comienza a estar disponible en todo tipo de terminales móviles comerciales. El beneficio, por lo tanto, es de una facilidad existente utilizándola para un uso distinto del original.

Como se muestra en este analisis la tecnologia Bluetooth propone nuevas aplicaciones, distintas de las convencionales -sustitución de cables en periféricos- y las obvias -soporte de redes ad-hoc, identificación RF activa, adecuando la infraestructura a las condiciones de propagación de los recintos. Por consiguiente, la elección de una tecnología inalámbrica debe basarse en la comparación de precio, tamaño y consumo, lo que posiblemente beneficia en la actualidad a Bluetooth.

El desarrollo de Aplicaciones móviles permite estar a la vanguardia en comunicaciones y sistemas de información, para incrementar las capacidades y mercados de nuevos clientes, mejorar la calidad de servicios, aumentar la productividad de los empleados, toma de decisiones con mayor rapidez y eliminación de la incertidumbre del cliente. Minimiza los costos de comunicación para el acceso remoto a información.

La aplicación permite el acceso desde un dispositivo Bluetooth a toda la información ya sea parcial o total, de uso personal que quiera movilizar, tal como agendas de clientes, catálogos de productos y precios, lista de teléfonos de empleados o amigos o cualquier información textual.
Con el creciente desarrollo de dispositivitos móviles y tecnologías Bluetooth, ha sido posible perfeccionar la movilidad de estos usuarios, de manera que ya no están obligados a realizar una actividad en un lugar fijo. El desarrollo de estas aplicaciones beneficia a los usuarios ya que les permite la perfección en la movilidad Facilitando el intercambio de información y mensajes en tiempo real como principal razón

Actualmente ya existe una amplia oferta de dispositivos móviles en el mercado y los fabricantes distribuyen con nuevos modelos constantemente. Como resultado de esta gran variedad de dispositivos, se deben afrontar problemas cada vez que se desarrolla una aplicación Web móvil.

Existen una infinidad de productos bajo la tecnología Bluetooth en el mercado, desde tarjetas de acceso, ratones para PC, teclados, antenas, software para

desarrollo, etc. Al adquirir cualquier producto, hay que tener muy en cuenta que la compañía fabricante pertenezca o esté certificada por el Bluetooth SIG.

La solución móvil está mostrando sus beneficios para la gestión de las empresas en la mejora de la productividad, y por supuesto en la creación de nuevos servicios.

Si bien es cierto que la tecnología crece, este medio de comunicación debe enfocar sus esfuerzos a mejorar la interoperatibilidad, el soporte de velocidades más altas y aspectos relacionados con la seguridad.

La ratificación de Bluetooth por la IEEE en el mes de marzo del 2002 es un triunfo importante para el Bluetooth SIG, el cual ha luchado por mucho tiempo por ser un protagonista de los estándares de redes inalámbricas en el mundo. Esta ratificación le dará a Bluetooth más credibilidad y permitirá más desarrollos en la industria encaminados a satisfacer necesidades más específicas de los usuarios finales a un bajo costo y buen desempeño.

# Nomenclatura

Los siguientes términos provienen de la especificación Bluetooth 1.2

AD HOC NETWORK: Una red típica creada en una manera espontánea.
Una red ad hoc no requiere ninguna infraestructura formal y es limitado en la extensión temporal y espacial.

Active Slave Broadcast (ASB): El Slave transmisión activa, es el transporte lógico que transporta al usuario en el tráfico de L2CAP a todos los dispositivos activos en el piconet.

Beacon Train: Un modelo de slots reservados dentro de un canal físico Piconet básico o adaptado. Las transmisiones empiezan en estos slots que son usualmente resincronizados en los dispositivos almacenados.

Bits de paridad: Este campo de 34-bit contiene los bits de paridad que forman la primera parte de la sincronización de la palabra del código de acceso del dispositivo que envía el paquete FHS. Estos bits se derivan de LAP.

Bluetooth: Es un enlace de comunicación inalámbrico, funciona dentro de la banda de ISM, usa una frecuencia de salto transeptora 2.4 gHz sin necesidad de licencia. Admite AV de tiempo real y comunicación de datos entre host de Bluetooth. El protocolo de enlace está basado en los slots.

Bluetooth Baseband: La parte del sistema Bluetooth que especifica o implementa el acceso mediano y los procedimientos de capa física para apoyar el intercambio de la voz en tiempo real, flujos de información de datos, y conexión en red ad hoc entre dispositivos Bluetooth.

Bluetooth Clock: Reloj interno Bluetooth de 28 bit controla a un subsistema Bluetooth. El valor de este reloj define la numeración del slot y regula varios canales físicos cada 312.5 μs

Bluetooth Controller: El sub- sistema contiene Bluetooth RF, Baseband,
Resource controller, Link Manager, Device manager y un Bluetooth HCI.

Bluetooth Device: Un dispositivo de comunicaciones inalámbricas de corto alcance usado en el sistema Bluetooth.

Bluetooth Device Address: Dirección de 48bit de Bluetooth usado para identificar cada dispositivo Bluetooth.

BD_ADDR: Bluetooth Device Address, BD_ADDR, es usada para que identifique un dispositivo Bluetooth.

Bluetooth HCI: Provee una interfaz de comando al controlador de baseband y el Link manager y acceso a estado de equipo físico y a registros de control. Esta interfaz provee un método uniforme de acceder a la capacidad baseband Bluetooth.

Bluetooth Host: Dispositivo de computación, periférico, teléfono celular, punto de acceso a red de PSTN o a LAN,s etc. Un anfitrión de Bluetooth fija a un controlador Bluetooth que puede comunicarse con otros bluetooth host adjuntados a sus controladores Bluetooth.

Channel: Canal físico o un canal de L2CAP, dependiendo del contexto.

Clases de Dispositivos: Este campo de 24-bit contiene la clase del dispositivo que envía el Paquete de FHS. El campo se define en Bluetooth Assigned Numbers. (https://www.bluetooth.org/foundry/assignnumb/document / assigned_numbers).

CLK $_{27\text{-}2}$ Este campo de 26-bit contiene el valor del reloj nativo del dispositivo que envía el paquete FHS, comprueba al principio de la transmisión el código de acceso de este paquete FHS. El valor de este reloj tiene una resolución de 1.25ms (intervalo de dos-slots). Para cada nueva transmisión, este campo se pone al día para que refleje el valor del reloj con precisión en tiempo-real.

Connect (to service): Establecimiento de una conexión (al servicio) a un servicio. Si no esta listo, también incluye el establecimiento de un enlace físico, transporte lógico, enlace lógico y canal L2CAP.

Connectable device: Es la extensión que atiende periódicamente el dispositivo Bluetooth, page explora al canal físico y responde a page en ese canal.

Connected device: Dos dispositivos Bluetooth conectados en el mismo piconet y con un enlace físico entre ellos.

Conneting: Conecta una fase de la comunicación entre dispositivos, cuándo una conexión entre ellos está siendo establecida. (Conecta la sigue fase después del establecimiento del enlace, y la fase es completada.)

Connection: Conexión entre dos aplicaciones semejantes o en una capa mas alta de protocolos correlacionados en un canal de L2CAP.

Connection establishment: Establecimiento de conexión con un procedimiento para crear una conexión correlacionada en un canal.

Coverage Area: Área donde dos dispositivos Bluetooth pueden cambiar mensajes con la calidad y rendimiento aceptable.

Creation of a secure connection: Creación de una conexión segura con procedimiento de establecer una conexión, incluye autentificación y encriptación.

Creation of a trusted relationship: Procedimiento donde el dispositivo remoto está marcado como un dispositivo de confianza. Esto incluye guardar una llave de enlace común para la futura autentificación y el paring (si la llave de enlace no esta disponible).

Device discovery: Procedimiento de descubrimiento de dispositivo para recuperar la dirección del dispositivo Bluetooth, reloj, clase de dispositivo y modo page scan y los dispositivos discoverable.

Discoverable device: En el rango que atiende periódicamente un inquiry scan en el canal físico y responderá un inquiry sobre este canal. El dispositivo normalmente esta conectactado.

Inquiring device: Dispositivo que lleva el procedimiento de inquiry (investigación).

Inquiry: Procedimiento de investigación donde un dispositivo Bluetooth transmite mensajes inquiry y está atento a las respuestas para descubrir a otros dispositivos Bluetooth que están dentro el Área de cobertura.

Inquiry scan: Procedimiento donde un dispositivo Bluetooth está atento al mensaje inquiry recibidos en un inquiry scan en el canal físico.

Isochronous: data: Información de datos isócrona en un flujo donde cada entidad de información en el flujo es obligado por una relación de tiempo para entidades anteriores y sucesivas.

Known device: Dispositivo para almacenar por lo menos el BD_ADDR.

LAP: Este campo de 24-bit contiene la parte del dispositivo de dirección más baja que envía el paquete FHS.

L2CAP channel: Conexión lógica entre dos dispositivos en el nivel L2CAP, soporta una aplicación sencilla o en un protocolo de capa más alta.

L2CAP Channel establishment: Procedimiento para establecer una conexión lógica en el nivel de L2CAP.

Link establishment: Procedimiento para establecer la conexión predefinida ACL y la jerarquía de conexiones y canales entre dispositivos.

Link: Enlace lógico.

Link key: Llave confidencial conocida por dos dispositivos y usado para autentificar cada dispositivo de otros.

LMP authentcation: Un LMP dirige el procedimiento para verificar la identidad de un dispositivo remoto.

LMP pairing: Procedimiento que autentifica dos dispositivos y crear una llave de enlace común, puede ser usado como base, relación de confianza o una (sencilla) conexión segura.

Logical Channel: Idéntico a un canal de L2CAP, pero desaprobado a causa de una alternativa en Bluetooth 1.1

Logical Link: El nivel más bajo de la arquitectura usado para brindar independencia en el servicio de transporte de datos a clientes del sistema Bluetooth.

Logical transport: Usado en Bluetooth para representar comúnmente enlaces lógicos diferentes atribuibles al reconocimiento compartido de protocolo e identificadores de enlace.

LT_ADDR: Este campo de 3-bit contiene la dirección de transporte lógico que el receptor debe usar si el paquete FHS se usa en connection setup o role Switch. Un slave responde a un master y un dispositivo que responde a un mensaje de inquirí request, incluirá un all-ZERO en el campo de LT_ADDR enviados en el paquete FHS.

Modo page scan: Este campo de 3-bit indicará como se usado el modo scan por default para el envió del paquete de FHS.

Name discovery: Procedimiento para recuperar el nombre de fácil manejo (el nombre del dispositivo Bluetooth) de un dispositivo conectado.

NAP: Este campo de 16-bit campo contiene la parte del dispositivo de dirección no-significante que envía el paquete FHS (también vea LAP, UAP, y NAP).

Packet: Formato del conjunto de bits que son transmitidos en un canal físico.

Page: Fase inicial del procedimiento de conexión dónde un dispositivo transmite un tren de mensajes page hasta que una respuesta se reciba del dispositivo de destino o un tiempo muerto ocurra.

Page scan: Procedimiento donde un dispositivo está atento a mensajes recibidos de page scan en el canal físico.

Paging device: Dispositivo Bluetooth que está llevando el procedimiento de page.

Paired device: Dispositivo Bluetooth con el que una llave de enlace se intercambio (o antes del establecimiento de conexión fue requerido o durante la fase de conexión)

Parked device: Dispositivo que opera en un piconet básico del modo que es sincronizado al maestro pero ha renunciado al transporte lógico default ACL.

Physical Channel: Caracterizado por la ocupación sincronizada de una secuencia de portadoras de RF por uno o más dispositivos. Varios tipos de canal físicos que existen con características definidas para propósitos diferentes.

Physical link: Conexión nivel-Baseband que establece enlace físico entre dos dispositivos usando paging.

Piconet: Colección de dispositivos ocupando una parte del canal físico donde uno de los dispositivos es el Piconet Master y los otros dispositivos están conectados con él.

Piconet Physical Channel: Canal que es dividido (time slots) en el cuál cada slots está relacionado a un salto de frecuencia en RF. Los saltos consecutivos corresponden a los diferentes saltos de frecuencias RF y ocurren en una tasa de salto usual de 1600 saltos/s. Estos saltos consecutivos persiguen una secuencia seudo-aleatoria de saltos, a través de 79 canales de RF.

Piconet Master: Dispositivo en un piconet de quien el Bluetooth Clock y Bluetooth Address son usados para definir las características físicas del canal de piconet.

Piconet Slave: Cualquier dispositivo en un piconet que no es el Piconet Master, pero está conectado con este.

PIN: Un número fácil de manejar que se puede utilizar para autenticar las conexiones a un dispositivo antes de que el corte allá tomado lugar.

PMP: Participante en Piconets Múltiples. Dispositivo que simultáneamente con un miembro de más de un piconet, utiliza la división de tiempo multiplexada (TDM), para intercalar su actividad sobre cada canal físico piconet.

The Parked Slave Broadcast:The Parked Slave Broadcast es usado para comunicaciones entre el Master y los dispositivos parked.

Scatternet: Dos o más piconets que incluyen uno o mas dispositivos que actúan como PMPs.

Service Layer Protocol: Protocolo de capa que usa un canal de L2CAP para transportar PDUs.

Service Discovery: Procedimientos querying and browsing para servicios ofrecidos por o a través de otro dispositivo Bluetooth.

Silent device: Dispositivo Bluetooth que aparece como silencio en un dispositivo remoto si este no responde a una petición hecha por un dispositivo remoto.

SR: Este campo de 2-bit es el campo de scan repetition e indica el intervalo entre dos page scan, windows consecutivas.

UAP: Este campo de 8-bit contiene la parte superior del dispositivo de dirección que envía el paquete FHS.

Undefined: Este campo de 2-bit es reservado para un uso futuro y será para poner a ZERO.

Unknown Device: Dispositivo Bluetooth para el cual no hay información (Bluetooth Device Address llave de enlace u otro) almacenada.

# Bibliografía

Bluetooth SIG. Bluetooth specifications version 1.2, noviembre 2003. http://www.bluetooth.com

Ericsson. http://www.ericsson.com

F.J. González-Castaño y J.J. García-Reinoso. Bluetooth location networks. En *Proc. IEEE Global Telecommunications Conference 2002 (Globecom'02)*, Taipei, Taiwan, nov. 2002.

F.J. González-Castaño y J.J. García-Reinoso. Survivable Bluetooth location networks. En *Proc. IEEE International Conference on Communications 2003 (ICC 2003)*, Anchorage, EEUU, may. 2003.

F.J. González-Castaño, J.J. García-Reinoso, F. Gil-Castiñeira, E. Costa-Montenegro y J.M. Pousada-Carballo. Bluetooth-assisted contextawareness in educational data networks. *Computers & Education*.

J.J. García-Reinoso, F.J. González-Castaño y F. Gil-Castiñeira. Bluetooth/ IEEE 802.11 Real-Time Mobile Auctions. *Wireless Personal Communications*.

IEEE. 802.11 wireless local area networks. http://grouper.ieee.org/groups/802/11/, 2003.

IEEE. 802.15 working group for WPAN. http://grouper.ieee.org/groups/802/15/, 2003.

Nokia http://www.nokia.com

Possio. Wireless services gateway PX30. http://www.possio.com

Texas Instruments. Wanda PDA.

Toshiba http://www.toshiba.com

T. Godfrey y J. Guttag. A locally coordinated scatternet scheduling algorithm.En *Proc. IEEE Conference on Local Computer Networks*, 2002

T. Yamasaki, M. Kishimoto, N. Komoda y H. Oiso. An information o_ering system for exhibition explanation by Bluetooth technology. En *Proc. SSGRR 2001, Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, 2001

W. Mao y D.M. Nicol. On k-ary n-cubes: theory and applications. N A S A CR-194996 ICASE report # 94-88, 1994.