ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

PROYECTO DE GRADO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA

ESTUDIO DE LA TRANSMISIÓN DE VOZ Y DATOS EN LA TECNOLOGÍA BLUETOOTH

FABIÁN ANDRÉS VARELA BURBANO

SALGOLQUÍ – ECUADOR

2007

ESTUDIO DE LA TRANSMISIÓN DE VOZ Y DATOS EN LA TECNOLOGÍA BLUETOOTH

CERTIFICACION

Se certifica que el Sr. Fabián Andrés Varela Burbano desarrolló y finalizó el proyecto en su totalidad.

Ing. Román Lara Cueva M.Sc.	**Ing. Julio Larco Bravo M.Sc.**
DIRECTOR	**CODIRECTOR**

RESUMEN

Este estudio pretende dejar esquemas claros y entendibles del proceso de transmisión de voz y de datos mediante la tecnología Bluetooth y el desempeño que puede esperarse al utilizar dispositivos clase 2 en aplicaciones inalámbricas.

Conociendo el rango efectivo teórico de operación de dispositivos Bluetooth clase 2 se intenta identificar el impacto que tiene el incremento de la distancia entre dos nodos Bluetooth (dispositivos BT-1000) en la degradación de una señal de voz originada en un punto y transmitida hacia otro nodo en conexión SCO y utilizando paquetes HV1. Asimismo se busca establecer el porcentaje de error debido a la retransmisión de paquetes DH1 según el incremento de distancia entre nodos con conexiones ACL. Para el análisis de la voz se empleó *Dynamic Time Warping* el cual permite conocer la similitud de dos señales de voz.

Los resultados obtenidos en la prueba de transmisión de datos demuestran que el dispositivo BT-1000 puede soportar enlaces ACL hasta una distancia de 80[m] con línea de vista y 20[m] sin línea de vista entre nodos. Aquí, el número de paquetes retransmitidos llegó a un máximo del 64% (LOS) y 32%(NLOS). En cuanto a la transmisión de voz el dispositivo permite establecer enlaces SCO hasta 80[m] y la degradación de la señal audible llega a un máximo del 8%, pese a que el mensaje todavía puede ser entendido por el receptor.

DEDICATORIA

Este trabajo a mi familia, mi madre Teresita, mi padre Fabián, mi hermana María Fernanda grandes compañeros en toda una vida.

AGRADECIMIENTO

A Dios por la vida, las fuerzas, la salud, por todo lo que tengo y lo que no. A mi familia, por la inspiración, ánimos, y cariño que nunca han faltado. A mis amig@s que preguntaban “¿y ya acabas?” y animaban a seguir avanzando. A mis directores de tesis por su tiempo y ayuda.

PRÓLOGO

Los dispositivos inalámbricos hoy en día resultan de suma importancia y son utilizados por varias industrias para ofrecer servicios o aplicaciones en distintos campos como el industrial, la medicina, automotriz, telecomunicaciones, entre otros. Es por ello, que al considerar las grandes perspectivas de crecimiento que la tecnología inalámbrica ofrece, el análisis de los métodos de conexión empleados para la comunicación de dichos dispositivos como Wi-Fi, GPRS/3G, o Bluetooth y su conocimiento teórico y práctico son de relevancia para el perfil del futuro profesional en la especialización de Telecomunicaciones en cualquier parte del mundo. En particular, el estudio que se presenta detalla la tecnología Bluetooth empleada para la transmisión de información tanto de datos y voz.

Con la utilización de la tecnología inalámbrica se logra liberar cables sin sacrificar posibilidades de conexión, de esta manera las limitaciones de espacio y tiempo desaparecen, lo que conlleva a que un sitio de trabajo pueda ubicarse prácticamente en cualquier lugar. Los dispositivos con tecnología inalámbrica como Bluetooth dan la suficiente movilidad y flexibilidad de red en cualquier entorno de trabajo.

Así también, la tecnología móvil debido a sus altas prestaciones permite mayor libertad en el manejo de recursos en entornos industriales o de oficina, es por ello que la informática móvil ofrece facilidades para la productividad, en cuanto a la facilidad de acceso a información o de comunicación sin necesidad de estar en un sitio fijo, evitando así viajes o interrupciones de un lugar a otro; además de la flexibilidad y rapidez que conllevan a mayor competitividad debido a las rápidas respuestas que se pueden generar en un ambiente totalmente flexible, todas estas características propias que ofrece la solución inalámbrica Bluetooth.

El conocimiento de la tecnología inalámbrica Bluetooth resulta necesario para entender, desarrollar y mejorar aplicaciones que están siendo utilizadas muy ampliamente en todo el mundo. El proyecto expuesto busca dejar conceptos necesarios y las bases fundamentales para el entendimiento del proceso que la

comunicación de voz y datos en la tecnología Bluetooth requieren. Así, con referencias claras y documentadas sobre esta tecnología inalámbrica de actualidad el entendimiento de la misma será posible.

En el estudio presentado, el Capítulo 1 es una introducción a la tecnología Bluetooth, aquí se presentan subtemas como su origen, generalidades, ventajas, problemas, y una variedad de aplicaciones que son posibles mediante Bluetooth. El Capítulo 2 describe la arquitectura de la pila de protocolos empleados en Bluetooth y los enlaces que resultan necesarios establecer la conexión entre nodos Bluetooth, aquí se mencionan dos partes fundamentales para entender a Bluetooth las cuales son las capas para el manejo del enlace (LM – *Link Manager*) y para el control lógico del enlace (L2CAP – *Logical Link Control and Adaptation Protocol*). De igual manera, todo dispositivo que disponga de Bluetooth necesariamente identificará los perfiles que dispone, la ejecución de estos perfiles significará el soporte o no de servicios como Headset para la comunicación de voz mediante auricular, OBEX o FTP para el intercambio de archivos, etc. Estos perfiles se presentan en el Capítulo 3. La cuarta parte del estudio consiste en realizar pruebas de transmisión de datos y voz con los equipos de Bluetooth BT-1000 disponibles. Aquí se presenta un análisis de la influencia de la variación de la distancia entre los dos nodos Bluetooth y su impacto en la degradación de la voz en enlaces SCO y en el número de retransmisiones de paquetes (en enlaces ACL) que suceden, dichas retransmisiones serán interpretadas con porcentajes de error. Finalmente, conclusiones y recomendaciones obtenidas a lo largo de la realización del estudio son presentadas en el Capítulo 5.

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS

CAPÍTULO 1

CAPÍTULO 2

CAPÍTULO 3

CAPÍTULO 4

ANEXOS

ÍNDICE DE FIGURAS

CAPÍTULO 1

CAPÍTULO 2

CAPÍTULO 3

Capítulo 4

ANEXOS

GLOSARIO DE TÉRMINOS Y LISTA DE ACRÓNIMOS Y ABREVIACIONES

LISTA DE ACRÓNIMOS Y ABREVIACIONES

ACK	Acknowledge	Confirmar recepción
ACL	Asynchronous Conectionless	Sin Conexión Asíncrona
AG	Audio Gateway	Entrada de Audio
AM_ADDR	Active Member Address	Dirección de Miembro Activo
ARQ	Automatic Repeat Request	Petición de Repetición Automática
ARQN	Automatic Repeat Request Number	Número de Petición de Repetición Automática
BB	Baseband	Banda Base
BCC	Bluetooth Control Center	Centro de Control Bluetooth
BD	Bluetooth Device	Dispositivo Bluetooth
BD_ADDR	Bluetooth Device Address	Dirección del Dispositivo BT
BIN	Binary	Binario
CAC	Channel Access Code	Código de Acceso al Canal
CBSD	Continuous Variable Slope Delta Modulation	Modulación Delta de Pendiente Continuamente Variable
CC	Call Control	Control de Llamada
CID	Channel Identifier	Identificador de Canal
CO	Connection Oriented	Orientado a la Conexión
CoD	Class of Device	Clase de Dispositivo
CRC	Cyclic Redundancy Check	Chequeo por Redundancia Cíclico
DAC	Device Access Code	Código de Acceso al Dev
Dev	Device	Dispositivo
DSSS	Direct Sequence Spread Spectrum	Espectro Ensanchado con Secuencia Directa
DT	Data Terminal	Terminal de Datos

FDD	Frequency Division Duplex	Dúplex por División de Frecuencia
FEC	Forward Error Correction	Corrección de Error hacia Delante
FHS	Frequency Hopping Synchronization	Sincronización por Salto de Frecuencia
FHSS	Frequency-Hopping Spread Spectrum	Espectro Ensanchado con Salto de Frecuencia
FTP	File Transfer Profile	Perfil de Transferencia de Archivos
FTP	File Transfer Protocol	Protocolo de Transferencia de Archivos
GAP	Generic Access Profile	Perfil de Acceso Genérico
GOEP	Generic Object Exchange Profile	Perfil Genérico de Intercambio de Objetos
GW	Gateway	Entrada
HCI	Host Controller Interface	Interface Controladora de Host
HEC	Header Error Check	Chequeo de Error de Cabecera
HS	Headset Profile	Perfil de Auricular
I/O	Input / Output	Entrada / Salida
IAC	Inquiry Access Code	Código de Acceso de Indagación
ICP	Intercom Profile	Perfil de Intercomunicador
IP	Internet Protocol	Protocolo Internet
IR	Infrared	Infrarrojo
IrDA	Infrared Data Association	Asociación de Datos por IR
ISM	Industry, Science and Medical	Industria, Ciencia y Medicina
L2CAP	Logical Link Control & Adaptation Protocol	Protocolo de Control y Adaptación del Enlace Lógico
LC	Link Control	Control del Enlace
LM	Link Manager	Administración del Enlace

LMP	Link Manager Protocol	Protocolo de Administración del Enlace
LocDev	Local Device	Dispositivo Local
LT_ADDR	Logical Trasport Address	Dirección Lógica de Transporte
MTU	Maximum Transmission Unit	Unidad de Transmisión Máxima
NAK	No Acknowledge	No Confirma recepción
OBEX	Object Exchange Protocol	Protocolo de Intercambio de Objetos
OPP	Object Push Profile	Perfil de Empuje de Objetos
PC	Personal Computer	Computador Personal
PCM	Pulse Code Modulation	Modulación por Código de Pulsos
PAN	Personal Area Network	Red de Área Personal
PDU	Protocol Data Unit	Unidad de Datos de Procolo
PIN	Personal Identification Number	Número de Identificación Personal
PPP	Point-to-Point Protocol	Protocolo Punto a Punto
PSM	Protocol / Service Multiplexer	Multiplexador de Protocolo/Servicio
QoS	Quality of Service	Calidad de Servicio
RemDev	Remote Device	Dispositivo Remoto
RFCOMM	Serial Cable Emulation Protocol	Protocolo de Emulación de Cable Serial
SCO	Synchronous Connection-Oriented	Orientado a la Conexión Síncrona
SD	Service Discovery	Descubrimiento de Servicio
SDAP	Service Discovery Application Profile	Perfil de Aplicación de Descubrimiento de Servicio
SDDB	Service Discovery Data Base	Base de Datos de SD
SDP	Service Discovery Profile	Perfil de Descubrimiento de Servicio
SDP	Service Discovery Protocol	Protocolo de Descubrimiento

		de Servicio
SEQN	Sequence Number	Número de Secuencia
Sep	Serial Port	Puerto Serial
SIG	Special Interest Group	Grupo de Interés Especial
SPP	Serial Port Profile	Perfil de Puerto Serial
SrvDscApp	Service Discovery Application	Aplicación de Descubrimiento de Servicio
SS	Spread Spectrum	Espectro Ensanchado
TCP	Transport Control Protocol	Protocolo de Control de Transporte
TCS	Telephony Control Specification	Especificación de Control de Telefonía
TDM	Time Division Multiplexing	Multiplexación por División en el Tiempo
TDMA	Time Division Multiple Access	Acceso Múltiple por División en el Tiempo
TL	Terminal	Terminal
UA	User channel Asynchronous	Canal de Usuario Asíncrono
UDP	User Data Protocolo	Protocolo de Datos de Usuario
UI	User Interface	Interface de Usuario
UI	User channel Isochronous	Canal de usuario Isócrono
US	User channel Synchronous	Canal de Usuario Síncrono
WAE	Wireless Application Environment	Ambiente de Aplicación Inalámbrica
WAP	Wireless Application Protocol	Protocolo de Aplicación Inalámbrica
WTA	Wireless Telephony Application	Aplicación de Telefonía Inalámbrica

GLOSARIO DE TÉRMINOS

Active Mode. Un dispositivo Bluetooth está en modo activo cuando participa activamente en el canal. Aquí, con ayuda de la sincronización del dispositivo maestro se puede mantener a los esclavos en estado activo.

AM_ADDR. La dirección de miembro activo está conformada por 3 bits únicamente cuando un esclavo se encuentra activo en el canal. Ocasionalmente se la conoce como dirección MAC de un dispositivo Bluetooth.

ARQN (Automatic Repeat Request Number). El número de petición para la repetición automática se compone de 1 bit que es encargado de informar de una transferencia exitosa o no de paquetes con CRC.

Autenticación. Proceso de verificación de quién se encuentra en el otro lado del enlace. La autenticación se puede dar por el proceso de emparejamiento (ingresando un PIN) o por una clave de enlace ya existente (generada por 2 dispositivos una vez ingresado el PIN por primera vez).

AUX. Paquete parecido a un DH1 para enlaces ACL que puede llevar hasta 30 bytes de información pero no cuenta con un código CRC.

Banda base. Es encargada de definir las especificaciones del procesamiento digital de señales que se realizan en hardware.

BD_ADDR (Bluetooth Device Address). Dirección única de 48 bits del transceiver ubicado en un dispositivo Bluetooth.

Bluetooth Device Class. Este parámetro es conocido mediante el proceso de descubrimiento e indica el tipo de dispositivo y servicios que soporta.
Cabecera de paquete. Tiene un tamaño total de 54 bits que contienen información del control del enlace, se divide en 6 campos (AM_ADDR, TYPE, ARQN, SEQN, y HEC).

CDMA (Code Division Multiple Access). Tecnología que utiliza espectro ensanchado para manejar comunicaciones de radio, asigna un código a cada llamada (que es conocido por el teléfono celular y por la radio base), y permite la agrupación de múltiples llamadas en una sola frecuencia.

Channel (Canal). Conexión lógica en el nivel de L2CAP entre dos dispositivos que corren una misma aplicación o algún protocolo de alguna capa superior.

Clase de Servicio. Define los atributos contenidos en un registro de servicio para describir una clase de servicio soportado por el dispositivo.

CRC (Cyclic Redundancy Check). Es un código de 16 bits agregado a un paquete para poder determinar si existe o no errores.

DAC (Device Access Code). El código de acceso del dispositivo es derivado de su BD_ADDR y se lo utiliza en el procedimiento de paging (scan y response).

DCID (Destination Channel Identifier). Representa el punto final (dispositivo receptor del mensaje) del canal establecido en transmisiones L2CAP.

Descubrimiento de nombre. Proceso que involucra pedir y recibir el nombre de un dispositivo.

DH (Data High Rate). Categoría de paquetes Bluetooth para datos en enlaces ACL que permiten altas tasas de transmisión mediante la reducción de chequeo de errores. Los paquetes DH son similares a los DM1 pero no cuentan con FEC.

DH1. Paquetes DH que pueden llevar hasta 28 bytes de información en un slot de tiempo.

DH3. Paquetes DH que pueden llevar hasta 185 bytes de información en un máximo de 3 slots de tiempo.

DH5. Paquetes DH que pueden llevar hasta 341 bytes de información en un máximo de 5 slots de tiempo.

Dispositivo conocido. Dispositivo Bluetooth del cual se conoce por lo menos su BD_ADDR.

Dispositivo Maestro. Dispositivo que con su reloj y secuencia de saltos ayuda a la sincronización de los dispositivos esclavos en una piconet.

DLCI (Data Link Connection Identifier). Identificador de 6 bits que representa la conexión entre un cliente y la aplicación del servidor. El DLCI se lo utiliza en la capa RFCOMM.

DM (Data Medium Rate). Paquete de datos para enlaces ACL que permite medianas tasas de transmisión, contiene un código CRC de 16 bits, y hace uso de FEC de 2/3.

DM1. Paquete DM que únicamente lleva datos de información que no superan los 18 bytes en un único slot de tiempo.

DM3. Paquete DM que puede llevar hasta 123 bytes de información en un máximo de 3 slots de tiempo.

DM5. Paquete DM que puede llevar hasta 226 bytes de información en un máximo de 5 slots de tiempo.

DV (Data Voice). Paquete para voz y datos en enlaces SCO, su campo total se divide en dos partes, para la voz no se utiliza FEC y se destina 80 bits con posibilidad de retransmisión, mientras que para datos se asignan 150 bits, cuenta con FEC de 2/3, y la retransmisión se la realiza si es necesario.

Enlace Físico. Se logra cuando un dispositivo Bluetooth está sincronizado y cumpliendo con una secuencia de saltos de RF. Es una asociación en el nivel de banda base entre dos dispositivos establecidos después del paging, donde se

tiene una secuencia de slots de transmisión en el canal físico entre maestro y esclavo.

HV (High quality Voice). Paquete de voz para enlaces SCO que no cuenta con CRC ni cargas para cabeceras.

HV1. Paquete HV que lleva hasta 10 bytes de información que son protegidos con un FEC de 1/3.

HV2. Paquete HV que lleva hasta 20 bytes de información que son protegidos con un FEC de 2/3.

HV3. Paquete HV que lleva hasta 30 bytes de información que no son protegidos con ningún FEC.

KEY (Clave). Clave de autenticación utilizada para establecer enlaces entre dispositivos Bluetooth.

Llave de enlace. Llave de autenticación para establecer un enlace entre dos dispositivos Bluetooth.

Modo Hold. Permite que dispositivos sincronizados en una piconet puedan permanecer en modo de ahorro de energía debido a que solo un cronómetro interno estará corriendo; así, su ciclo de trabajo se reduce. El modo hold tiene una eficiencia media de ahorro de energía de todos los modos (sniff, hold, y park).

Modo Sniff. Permite que dispositivos sincronizados en una piconet puedan permanecer en modo de ahorro de energía mediante la reducción de su ciclo de trabajo (esclavos escuchan a la piconet con una tasa menor). En el modo sniff se tiene el ciclo de trabajo más alto de todos los modos para ahorro de energía (sniff, hold, y park).

Modo Visible. Cualquier dispositivo Bluetooth se encuentra en modo visible cuando los demás dispositivos Bluetooth pueden detectarlo.

Non-Connectable Device. Dispositivo que no responde al procedimiento de paging se dice que no es conectable.

Non-Discoverable Device. Dispositivo no responde al procedimiento de inquiry se dice que no puede ser descubierto.

Null Packet. Paquete de 126 bits (con una cabecera de paquete y el CAC únicamente) utilizado para devolver información del enlace para la fuente.
Perfil. Servicio o aplicación definida que un dispositivo Bluetooth ofrece o está en capacidad de utilizar.

Piconet. Red compuesta por un dispositivo maestro y hasta siete dispositivos esclavos que cuentan con la tecnología Bluetooth.

Scatternet. Dos o más piconets independientes y sin sincronismo forman una scatternet.

SEQN (Sequence Number). Provee un esquema de numeración secuencial para el ordenamiento de cadenas de paquetes de datos.

Slot de tiempo. Espacio de tiempo que en Bluetooth dura 625us en el cual se puede enviar un paquete desde un dispositivo Bluetooth a otro.

CAPITULO 1

INTRODUCCIÓN A LA TECNOLOGÍA BLUETOOTH

1.1. ORIGEN DE BLUETOOTH

El origen de la tecnología Bluetooth ocurrió en el año de 1994, en el cual la compañía de telecomunicaciones Ericsson empezó sus estudios e investigaciones de factibilidad acerca de un nuevo interfaz que requiera baja potencia para interconectar vía radio dispositivos, como teléfonos móviles y accesorios, con el fin de eliminar cables [1].

Conforme este proyecto investigativo avanzaba se fue haciendo claro que este tipo de enlace podía ser utilizado en un gran número de aplicaciones en distancias cortas, así, grandes empresas como IBM, Intel, Nokia, Toshiba, Motorola, 3Com y más de 4.000 empresas hasta el 2005 han unido esfuerzos en un mismo grupo de trabajo para encontrar soluciones y crear el estándar que hoy se conoce como Bluetooth [5].

En julio de 1999, el gran grupo de trabajo publicó la primera especificación de Bluetooth 1.0, la cual contó con dos principales documentos del núcleo y el perfil de Bluetooth, los cuales mencionan diseños, componentes y especificaciones para la interoperabilidad.

1.2. LA TECNOLOGÍA BLUETOOTH

Ante la creciente demanda de movilidad, Bluetooth es una tecnología inalámbrica para la comunicación en distancias cortas que busca eliminar el cableado de conexiones entre dispositivos electrónicos, tanto portátiles como fijos, manteniendo altos niveles de seguridad. Esta tecnología inalámbrica cuenta con características que ofrecen fiabilidad, bajo consumo de energía, tamaño pequeño de dispositivos y mínimo coste [6]. De esta manera, mediante la especificación Bluetooth, varios dispositivos pueden conectarse y comunicarse entre sí debido a la gran capacidad de ésta tecnología para gestionar simultáneamente transmisiones tanto de voz como de datos mientras se encuentren en el mismo radio de acción.

Las especificaciones Bluetooth comprenden un sistema integral de hardware, software y requerimientos de interoperabilidad. La tecnología inalámbrica Bluetooth hace uso de la banda de frecuencias no licenciadas para la industria, ciencia y medicina (ISM), concretamente entre 2.4 y 2.485 GHz, por lo que no requiere de pagos extras para utilizar el espectro, sino que el único gasto está vinculado directamente al dispositivo. Es así que su disponibilidad se encuentra en todo el mundo y dicho estándar está en continuo estudio y desarrollo para reforzar sus características.

Debido al gran número de aplicaciones con Bluetooth, ésta tecnología es una de las líderes en el mercado, es por ello que alrededor de 500 millones de unidades fueron distribuidas sobre diversos sectores industriales hasta finales del 2005 [7]. La necesidad de enlazar tanto dispositivos móviles como fijos en una misma red hace de Bluetooth una solución innovadora y atractiva para proveer aplicaciones con la ventaja de reemplazar cables por equipos que brindan simplicidad mediante su portabilidad, ligereza, bajo costo de los chips y de gran campo aplicativo.

Los radio enlaces con Bluetooth resultan robustos debido a la utilización de saltos de frecuencia y Spread Spectrum (SS) para evitar interferencias y desvanecimiento de señales. Con el esparcimiento de la

potencia de la señal sobre una banda más amplia de frecuencias y los saltos de una frecuencia a otra, las comunicaciones de voz y datos resultan más seguras. Los dispositivos de SS tienen alrededor de 1600 saltos por segundo sobre 79 frecuencias y siempre antes de iniciar la comunicación se estable la secuencia de saltos para los dispositivos involucrados.

La seguridad es un aspecto que Bluetooth considera, por ello las emisiones de radiación, que generalmente son de 1mW, no superan los límites establecidos para teléfonos inalámbricos de la industria, además de ser inofensivas para la gente en general y no causan efectos perjudiciales ni a las comunicaciones públicas o privadas, ni a los equipos de telecomunicaciones involucrados. También en seguridad, además de ser extremadamente difícil la intercepción de señales debido al salto de frecuencia, existe un mecanismo de autenticación el cual no permite el acceso de datos o funciones críticas que pueden ser causantes de daños al sistema, así mismo la encriptación de datos es llevada a cabo en la transmisión para garantizar privacidad en el enlace. Si se requiere mayor seguridad se puede sesionar con un generador de clave, el cual cada cierto tiempo e incluso para el ingreso a la red pedirá la clave de autorización.

1.3. VENTAJAS DE BLUETOOTH

La especificación Bluetooth es una tecnología global de bajo costo para comunicaciones inalámbricas entre dispositivos fijos o móviles que funcionan bajo esta tecnología. La conexión entre dispositivos electrónicos con Bluetooth se establece de manera simple, es decir al encenderlos por su propia cuenta tratarán de interconectarse. De esta manera se evitan cables, puertos periféricos o antenas relativamente grandes para lograr conexión. De hecho, la eliminación de cables da mayor seguridad en ambientes de trabajo puesto que se evita posibles desconexiones por imprevistos o descuidos, además de hacerlos confortables y adecuados para un buen desempeño.

Dependiendo de las aplicaciones que se quieran brindar, existen dispositivos con tecnología Bluetooth con la capacidad de cubrir distancias de por lo menos 10 metros hasta 100 metros, lo cual resulta un beneficio en caso de querer lograr una mayor cobertura.

La tecnología Bluetooth también permite la comunicación rápida y efectiva entre dispositivos tan pronto se encuentren en los rangos de operación, evitando así a usuarios el tener que acceder a aplicaciones o presionar botones con el fin de iniciar el proceso de conexión. Estos dispositivos, a diferencia de los infrarrojos (IR), no requieren ningún tipo de configuración ni tampoco línea de vista para lograr comunicación.

Otro aspecto importante de Bluetooth es que su protocolo es estándar a nivel mundial, lo cual significa que dispositivos una vez adquiridos funcionan en cualquier parte del mundo y con el mismo desempeño.

1.4. REDES DE ÁREA PERSONAL

Una piconet pertenece a una red de área personal o PAN en la que dispositivos Bluetooth se encuentran formándola. Una piconet está compuesta por lo menos de 2 dispositivos y puede llegar a tener un máximo de 8, donde se establece uno y solamente un maestro, mientras que los dispositivos restantes son llamados esclavos.

El grupo de trabajo de la IEEE 802.15 es el encargado de desarrollar estándares para la interoperabilidad de dispositivos involucrados en comunicaciones en redes PAN o inalámbricas de corto alcance. El objetivo principal es crear un estándar global para la coexistencia de numerosas aplicaciones en redes PAN orientadas a un mercado que crece cada día más a nivel mundial.

1.5. TOPOLOGÍA DE BLUETOOTH

Conociendo que la especificación Bluetooth soporta conexiones punto-punto o punto-multipunto, varias piconet pueden formarse en un ambiente común y no interferirse una a otra debido a la sincronización diferente que deben guardar. Varias piconet pueden ser enlazadas debido a que dispositivos Bluetooth pueden pertenecer a dos o más piconets al utilizar múltiplexación por división de tiempo TDM siempre y cuando el dispositivo perteneciente a ambas redes esté activo únicamente en una de ellas a la vez, a esta nueva red formada por varias piconets se conoce como *scatternet*, donde cada piconet es diferenciada debido a un distinto salto de frecuencia y sincronización.

Cada piconet, como se dijo anteriormente, puede tener un máximo de 8 dispositivos punto-punto, donde tan solo uno de ellos será el maestro, la Figura 1.1 presenta los esquemas mencionados.

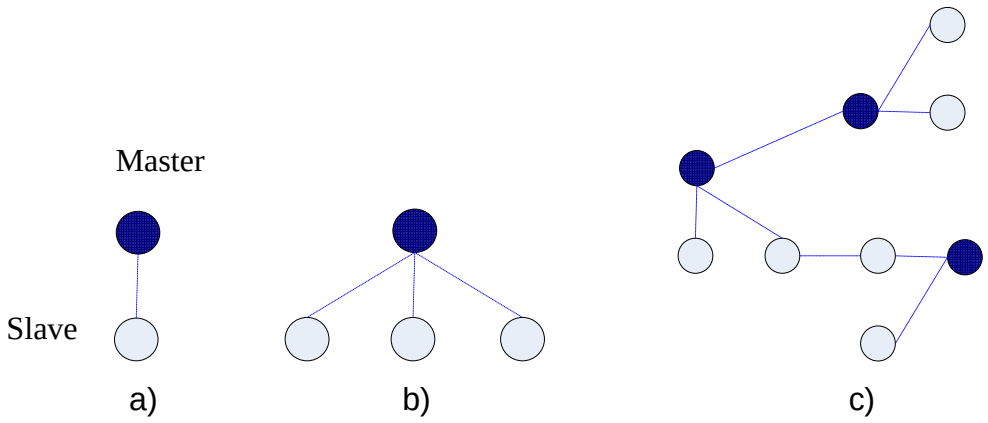

Figura. 1. 1. Posibles topologías en una red Bluetooth a) Piconet único esclavo. b) Piconet múltiples esclavos. c) Scatternet

1.6. QUÉ SE PUEDE HACER CON BLUETOOTH

La tecnología Bluetooth permite la conexión de varios dispositivos de telecomunicaciones y computación de manera sencilla y rápida, la cual evita el uso de cables, periféricos, traslado de equipos de un lugar a otro y asignación de espacios fijos. Bluetooth, además de permitir conexiones ad hoc de manera rápida, también elimina múltiples equipos y cables con tan solo un enlace de radio. Las aplicaciones que se pueden lograr son innumerables, entre ellas se puede mencionar teléfonos 3-en-1, conexión a Internet, dispositivos manos libres para atender llamadas, computadoras portátiles como teléfono, funciones de impresión y fax, escritorios inalámbricos, entre otras.

Las presentaciones con la instalación de cables entre el proyector y la computadora no serán necesarias si se enciende un dispositivo Bluetooth y se espera pocos segundos para permitir la adquisición de parámetros para establecer la conexión mediante el enlace de radio. Así mismo se pueden realizar conferencias o presentaciones en distintas habitaciones con la utilización de monitores de computadores comunes sin necesidad de contar con una pantalla de proyección, iluminación especial y espacios para los asistentes.

La sincronización de datos es otra aplicación de Bluetooth puesto que facilitaría el almacenamiento de información, así, en una oficina se puede trabajar en una laptop y automáticamente cada cambio o información guarda pasaría al dispositivo móvil (como palmtops) sin necesidad de realizar ninguna operación. De igual manera, al llegar a casa y encender el computador automáticamente se bajará la información recientemente agregada al dispositivo móvil sin comandos explícitos siempre y cuando se mantengan ambos equipos a una distancia o rango permitido.

La sincronización remota es posible y utilizable, de hecho se puede obtener acceso a Internet desde una palmtop o laptop mediante el teléfono

celular que se encuentra en un portafolio cercano sin necesidad de conectarlos físicamente.

La impresión es otra área beneficiada con Bluetooth puesto que se puede imprimir remotamente desde un computador e incluso enviar fotografías desde una cámara digital directamente hacia la impresora. También, se podría enviar fotografías desde una cámara digital al teléfono celular y luego hacia una dirección electrónica.

La tecnología de Bluetooth también ayuda al comercio electrónico, por ello clientes de supermercados o tiendas no tendrán que llevar consigo tarjetas de crédito o dinero en efectivo debido a que los celulares tendrán tarjetas inteligentes con toda la información personal que se requiere y podrán tener una cuenta de dinero electrónico. De hecho ya existe una billetera inalámbrica de Ericsson la cual utiliza la tecnología Bluetooth y varias tarjetas inteligentes, así la billetera se puede utilizar para compras en cualquier lugar o incluso por Internet, para lo cual dependiendo del tipo de transacción se utilizará la tarjeta inteligente destinada a dicha función. De esta manera inalámbricamente se podría promocionar entradas o avisar si todas están vendidas, comprarlas y recibirlas electrónicamente vía Bluetooth utilizando los terminales de puntos de venta de cines o teatros.

1.7. PROBLEMAS CON BLUETOOTH

Pese a que Bluetooth puede ser escalada para soportar conexiones con dispositivos a 100 metros de distancia si se utiliza el rango de frecuencia de 5 GHz, la tecnología Bluetooth tiene una gran limitación en cuanto a su tasa de datos de 1 Mbps que maneja. Otro inconveniente es la utilización de la banda no licenciada ISM debido a que en algunos países se requiere autorización del gobierno para operar sobre dichas frecuencias. De igual manera, el costo de los chips todavía no es lo suficientemente económico para implementarlos masivamente en dispositivos portátiles, móviles o fijos.

1.8. CONCEPTOS BÁSICOS

1.8.1. Spread Spectrum. La tecnología Bluetooth hace uso de la técnica de codificación digital de spread spectrum (SS), la cual consiste en tomar una señal de banda angosta y ensancharla o esparcirla en una porción de banda de frecuencias más ancha, la figura 2.1 presenta lo mencionado. Con esto se logra tener señales más robustas y resistentes a interferencia además de obtener mayor seguridad para evitar intercepciones. Además de emplear SS, Bluetooth también utiliza TDD (Time Division Duplexing) derivada de TDMA (Time Division Multiple Access) para asignar slots de tiempo para la comunicación de voz y datos.

El llevar a cabo el ensanchamiento de espectro no solo requiere mayor ancho de banda, sino también aumentar el número de bits a transmitirse. En este proceso tanto transmisor como receptor funcionan bajo el mismo código, con la diferencia que el receptor es encargado de transformar la señal a su forma original.

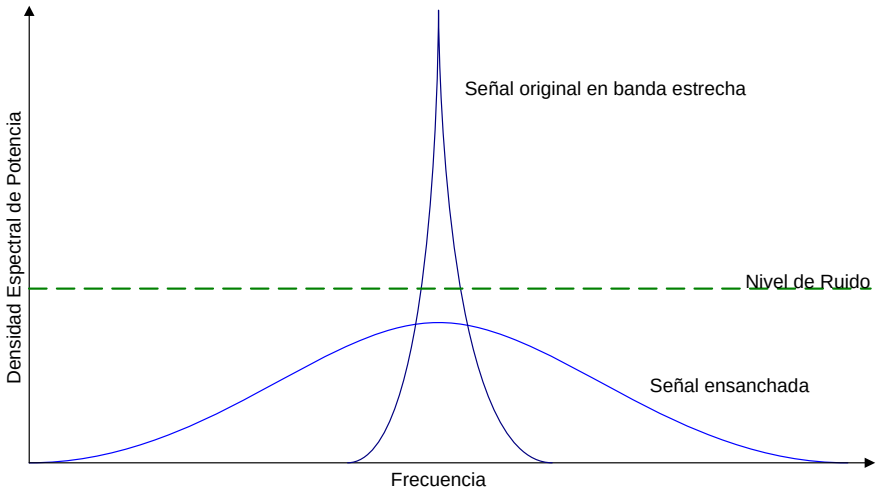

Figura. 1. 2. Comparación de señal banda estrecha con señal en SS

La principal ventaja que ofrecen las ondas de radio con SS es la facilidad de manipularse para obtener propagaciones adecuadas sin importar la interferencia electromagnética en el medio. En SS existen dos modos de operación, mediante saltos de frecuencia, que es utilizado en Bluetooth, o por secuencia directa.

Saltos de Frecuencia, en este sistema la portadora transmitida cambia constantemente de frecuencia de manera pseudoaleatoria y con una tasa de saltos de acuerdo al código de secuencia. En Bluetooth se establece una tasa 1600 saltos por segundo a lo largo de 79 frecuencias. Por otro lado, el receptor se encarga de encontrar esta secuencia y generar una señal de frecuencia intermedia constante, por lo que solo dicho receptor puede entender los datos que están siendo transmitidos. Cada unidad de Bluetooth posee un reloj interno que determina tiempos y saltos de su transceiver. Debido a que en una piconet el maestro fija dichos parámetros, los mismos son desplazados, almacenados y actualizados periódicamente en cada unidad.

Secuencia Directa, este sistema es el más utilizado y conocido debido a que permite crear y operar múltiples redes en una misma área. La secuencia directa consiste en ensanchar la energía de radio en una porción mayor a la necesitada normalmente para los datos, ello se logra dividiendo un bit en varios sub-bits llamados chips, lo cual provoca aumento en la tasa de modulación. La ventaja de aumentar los chips es que son proporcionales a la inmunidad de la señal a interferencias. Por otro lado, el receptor es encargado de reconstruir la señal original multiplicando la señal recibida por una generada localmente e impuesta por la secuencia de código.

1.8.2. Time Division Duplexing. Bluetooth utiliza Duplexación por División de Tiempo (TDD) la cual permite que sistemas puedan comunicarse bidireccionalmente al mismo tiempo, lo que es indispensable para comunicaciones de voz y la demanda de carga y descarga de información de manera asíncrona además de facilitar la transferencia de archivos entre

dispositivos portátiles u ordenadores y facilitar el proceso de sincronización. TDD fue creado con el fin de transportar señales digitales de datos y utiliza únicamente un solo canal y una antena para enviar y recibir información, lo cual es posible dividiendo y asignando tiempos para cada función.

Por lo mencionado, pese a que TDD no es completamente full duplex, se acerca mucho a ello. De hecho en Bluetooth el intercalar envío y recepción ocurre tan rápidamente que una conversación se puede establecer sin que los oyentes puedan percibir dichos cambios como espacios sin habla. En Bluetooth cada canal es dividido en slots de longitud de tiempo de 625 milisegundos en donde maestro y esclavos transmiten alternadamente, pudiendo únicamente iniciar la transmisión en slots pares e impares respectivamente.

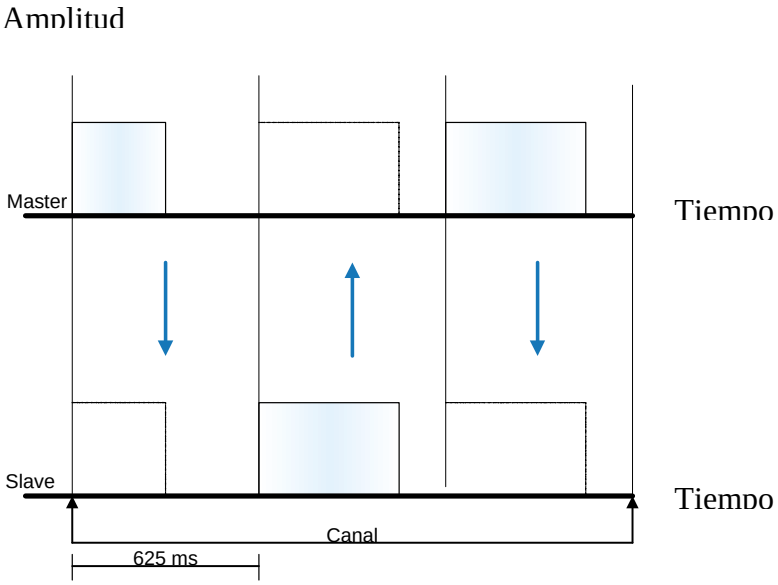

Figura. 1. 3. Esquema de TDD utilizado en la tecnología Bluetooth

En TDD, a diferencia de FDD (*Frequency Division Duplex*) que tiene anchos de banda fijos, los anchos de banda pueden ser asignados de acuerdo a necesidades que la demanda requiera y garantizar dicha espacios a aplicaciones específicas. La Figura 1.4 presenta el esquema de paquetes utilizando un slot y múltiples slots.

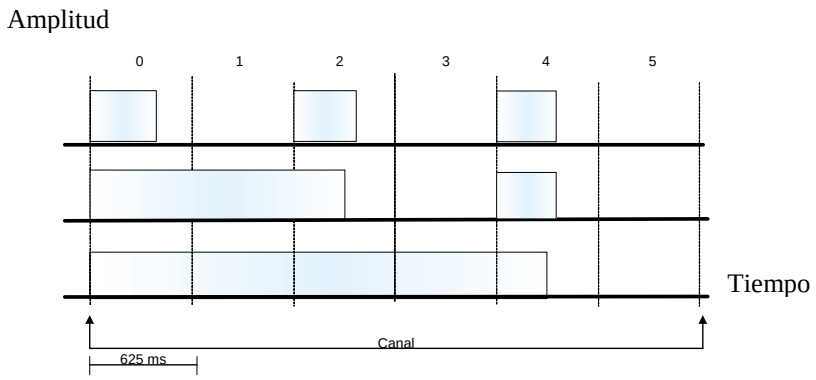

Figura. 1. 4. Esquema de saltos en paquetes con uno y múltiples slots

1.8.3. Enlaces Físicos. Dos tipos de enlaces se pueden tener entre maestro y esclavos en una piconet, los cuales son *orientados a la conexión* y *sin conexión*. Los primeros garantizan que la información transmitida llegue en el orden adecuado y para ello es necesario establecer primeramente una sesión (como en el caso de *ATM*[1] o frame relay[2]), mientras que en los enlaces sin conexión simplemente se envían paquetes hacia el receptor sin considerar una sesión previa, el orden de llegada de los datos, ni la llegada de los mismos (como ocurre con IP). En Bluetooth estos tipos de enlaces son ***SCO*** *(Synchronous Connection-Oriented*) y ***ACL*** (*Asynchronous Connectionless*).

El enlace ***SCO*** es un enlace punto a punto y simétrico entre el maestro y uno o varios esclavos en una piconet, para asegurar dicho enlace se fija determinados slots de tiempo en intervalos regulares, por lo que es considerado como una conexión con conmutación de circuitos y utilizable para comunicaciones de voz (información sensitiva al tiempo) sin existir retransmisión de paquetes. El maestro, quien es encargado de establecer el enlace SCO mediante un mensaje de setup que indica tiempos y slots asignados, puede tener un máximo de tres enlaces SCO entre uno o varios esclavos en una piconet. De igual manera cada esclavo

[1] Modo de Transferencia Asíncrona o **Asynchronous Transfer Mode**

[2] Técnica de comunicación mediante retransmisión de tramas (conmutación de paquetes), permite transmitir tramas de tamaños variables

puede tener un máximo de tres enlaces SCO si provienen del mismo maestro o dos enlaces SCO si provienen de varios maestros.

El enlace ***ACL*** es un enlace punto a multipunto entre maestro y todos los esclavos de una piconet y utiliza los slots de tiempo no reservados para enlaces SCO. Aquí, la conexión que se establece es mediante conmutación de paquetes, la retransmisión de paquetes es válida y únicamente puede existir un enlace ACL en cada esclavo.

1.9. PAQUETES BLUETOOTH

Debido a los dos tipos de enlaces existentes, también se encuentran dos tipos de paquetes disponibles para SCO y ACL. Para la voz se utilizan paquetes SCO, los cuales son enrrutados hacia los puertos I/O de voz pese a no contar con mecanismos para corregir errores. Por otro lado, los paquetes ACL si cuentan con mecanismos para la corrección de errores (FEC) y pueden llevar información tanto de datos de interés para el usuario como de datos de control. El paquete general de Bluetooth se presenta en la Figura 1.5.

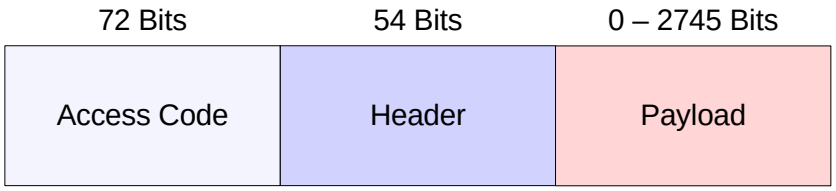

Figura. 1. 5. Formato general de un paquete Bluetooth

El ***código de acceso*** se emplea para la señalización y existen tres tipos de código:

- *Código de acceso al canal (CAC)*, encargado de identificar una piconet.

- *Código de acceso al dispositivo (DAC)*, utilizada para procedimientos de señalización como paging.
- *Código de acceso de indagación (IAC)*, existen dos tipos de IAC el general, utilizado para descubrir unidades Bluetooth, y el dedicado que se emplea en grupos específicos de unidades Bluetooth para descubrir únicamente este tipo de dispositivos comunes.

La ***cabecera*** o ***header*** cuenta con información de control y su utilización no es necesaria. En caso de utilizarla se incluyen seis campos:

- *Dirección de miembro activo,* los 3 bits ayudan a reconocer los miembros activos en una piconet. La dirección 000 es definida para broadcast.
- *Tipo*, los 4 bits especifican el tipo de paquete (ACL o SCO) y el número de slots que se ocupará dicho paquete.
- *Flujo*, este bit se activa cuando el buffer del receptor en un enlace ACL está saturado, así se detiene la transmisión momentáneamente pese a que paquetes SCO todavía pueden recibirse.
- *Petición de repetición automática*, este bit es utilizado para comunicar al transmisor que la transferencia de información ocurrió sin inconvenientes. Así, la bandera ACK activa indica una respuesta positiva y la bandera NAK lo contrario.
- *Número de secuencia*, este bit asigna un esquema de numeración secuencial con el fin de ordenar los paquetes según fueron transmitidos.
- *Cabecera de control de errores*, 8 bits se utilizan para controlar la cabecera y garantizar integridad.

La última parte del formato de paquetes Bluetooth es el ***payload***, el cual cuenta con dos campos según sea el enlace establecido, es decir un síncrono (voz) u otro asíncrono (datos). Este último segmento está dividido en tres partes:

- *Cabecera de carga*, puede ser de 1 o 2 bytes y únicamente existe en paquetes de datos. Esta cabecera menciona el número de bytes del cuerpo de carga, controlar el flujo y especifica el canal lógico a utilizarse.
- *Cuerpo de carga*, puede tener una longitud de 0 a 2721 bits, es encargada de incluir la información del usuario.
- *CRC*, 16 bits de código de redundancia calculados antes de la transmisión.

Existen 3 tipos de paquetes definidos para enlaces SCO (HV1, HV2, y HV3) y un híbrido que tiene la capacidad de llevar tanto datos como voz (DV - Data Voice). Los paquetes HV (High-quality Voice) son paquetes utilizados específicamente para el habla y debido a que la retransmisión no es posible en estos casos, este tipo de paquetes no cuentan con CRC ni utilizan cargas para cabeceras. A continuación se define los paquetes mencionados [8]. La Tabla 1.1 presenta un resumen de las características de los paquetes para SCO.

Tabla. 1. 1. Tipos de paquetes para SCO [9]

Tipo	Cabecera (bytes)	User Payload (bytes)	FEC	CRC	Máx. Tasa Simétrica (Kbps)
HV1	No	10	1/3	No	64.0
HV2	No	20	2/3	No	64.0
HV3	No	30	No	No	64.0
DV	1	10 + (0-9 D)	2/3	Si (Datos)	64.0 + 57.6 D

- *HV1*, paquete HV que puede llevar hasta 10 bytes de información que son protegidos con un FEC de 1/3 y no cuenta con CRC. Este tipo de paquete ocupa dos slots de tiempo y debe ser enviado cada 2 slots de

tiempo, además puede soportar 1.25ms (2x625us) de habla a 64Kbps. La longitud del payload es de 240 bits al igual que HV2 y HV3.

- *HV2,* paquete HV que puede llevar hasta 20 bytes de información que son protegidos con un FEC de 2/3. Este tipo de paquete debe ser enviado cada 4 slots de tiempo.
- *HV3,* paquete HV que puede llevar hasta 30 bytes de información que no son protegidos con ningún FEC. Este tipo de paquetes deben enviarse cada 6 slots de tiempo.
- *DV,* paquete combinado para voz y datos en el cual se asigna 80 bits para el campo de voz (no hay posibilidad de retransmisión) y 150 bits para el campo de datos que cuenta con FEC de 2/3 y la retransmisión si es posible.

Además de los paquetes HV y DV existen también 7 tipos de paquetes definidos para enlaces ACL, 3 paquetes DM (Data Medium Rate) para tasas medias de transmisión; 3 paquetes DH (Data High Rate) para tasas altas de transmisión; y un paquete AUX. La Tabla 1.2 presenta un resumen de las características de los paquetes para ACL.

Tabla. 1. 2. Tipos de paquetes para ACL [9]

Tipo	FEC	Cabecera (bytes)	User Payload (bytes)	CRC	Máx. Tasa Simétrica (Kbps)	Máx. Tasa Asimétrica	
						Forward	Reverse
DM1	2/3	1	0 – 17	Si	108.8	108.8	108.8
DM3	2/3	2	0 – 121	Si	258.1	387.2	54.4
DM5	2/3	2	0 – 224	Si	286.7	477.8	36.3
DH1	No	1	0 – 27	Si	172.8	172.8	172.8
DH3	No	2	0 – 183	Si	390.4	585.6	86.4
DH5	No	2	0 – 339	Si	433.9	**723.2**	57.6
AUX	Opcional	1	0 - 29	Opcional	185.6	185.6	185.6

Los paquetes DM1, DM3, y DM5 llevan 18, 123, y 226 bytes de información (incluido cabeceras) y ocupan un máximo de 1, 3, y 5 slots de tiempo respectivamente [10], además estos paquetes son codificados con un FEC de 2/3 (información y CRC) y cuentan con un código CRC de 16 bits por lo que la retransmisión si es posible [8].

Por otro lado, los paquetes DH son parecidos a los DM con la excepción de que los bytes de información no son codificados con FEC. Aquí existen 3 tipos de paquetes DH1, DH3, y DH5 que llevan 28, y 185, 341 bytes de información (incluido cabeceras) y ocupan 1, 3, y 5 slots de tiempo respectivamente.

Asimismo existe también el paquete AUX, el cual es parecido a un DH1 y puede llevar hasta 30 bytes de información pero generalmente no cuenta con un código CRC.

1.10. CANALES LÓGICOS

En Bluetooth existen canales lógicos destinados para el control e información de los dispositivos participantes. Estos canales se establecen sobre los enlaces ACL o SCO y cada uno de ellos cumple funciones dependiendo del tipo de enlace físico. En Bluetooth existen cinco tipos de enlaces lógicos y se mencionan a continuación:

- LC (Enlace de Control).
- LM (Enlace Administrador).
- UA (Usuario Asíncrono).
- UI (Usuario Isócrono).
- US (Usuario Síncrono).

1.11. ARQUITECTURA CLIENTE – SERVIDOR

La arquitectura cliente-servidor tiene su origen debido a la necesidad de interactuar usuarios de computadores con bases de datos, de esta forma un servidor puede brindar información requerida de manera controlada, íntegra y eficiente a través de programas, además de ofrecer la ventaja de escalabilidad en el sistema.

En Bluetooth, y particularmente en una piconet establecida, la arquitectura es comparable con la de cliente-servidor, donde los clientes (esclavos) serán los usuarios individuales o dispositivos Bluetooth que realizan tareas diferentes y requieren distinto tipo de información del servidor. En cambio, el servidor es comparado con el nodo maestro en una piconet, realiza las funciones de administración, es capaz de soportar múltiples usuarios y brindar el acceso o la información que cada usuario requiere.

Los dispositivos Bluetooth pueden conocer su identidad o rol en la red mediante el servicio de descubrimiento. El protocolo **SDP *(Service Discovery Protocol)*** asigna los roles a los dispositivos Bluetooth y así son separados en dispositivos locales o dispositivos remotos. Los primeros cuentan con la aplicación de descubrimiento y son encargados de buscar, descubrir y presentar los resultados de los dispositivos encontrados. La participación de los dispositivos remotos consiste en responder las indagaciones que realizan los dispositivos locales.

CAPITULO 2

ARQUITECTURA DE LA PILA DE PROTOCOLOS Y ENLACES EN BLUETOOTH

2.1. PROTOCOLOS DE BLUETOOTH

La especificación Bluetooth cuenta con una serie de protocolos que son utilizados dependiendo de la aplicación a realizarse; aunque sin importar de la aplicación, la pila de protocolos hace uso de la capa física y de enlace de datos. La Figura 2.1 presenta los protocolos involucrados en la especificación Bluetooth 1.0, donde algunos se utilizan en distintas plataformas pero particularmente LMP y L2CAP pertenecen a la tecnología Bluetooth [1].

Debido a que Bluetooth cuenta con protocolos que funcionan en otras plataformas, la creación de protocolos de aplicación sobre los ya existentes resulta posible, de esta manera, se puede ampliar el número de aplicaciones disponibles.

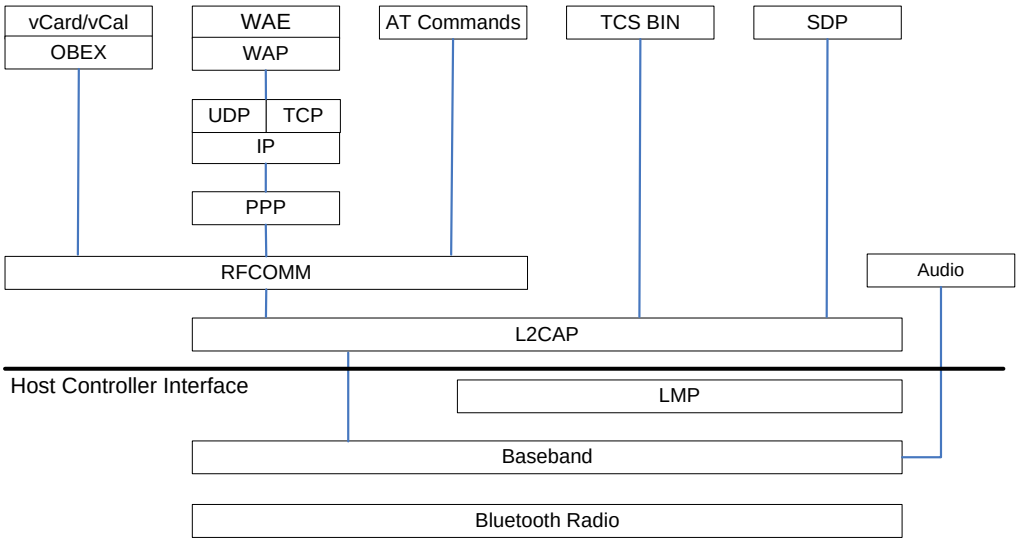

Figura. 2. 1. Pila de Protocolos Bluetooth Especificación 1.0

En Bluetooth existen cuatro capas con protocolos específicos en cada una de ellas como lo indica la Figura 2.2. Los protocolos que son esenciales para la mayor parte de dispositivos Bluetooth son los del núcleo de Bluetooth y el de radio de Bluetooth. A continuación se presenta un resumen de los protocolos existentes.

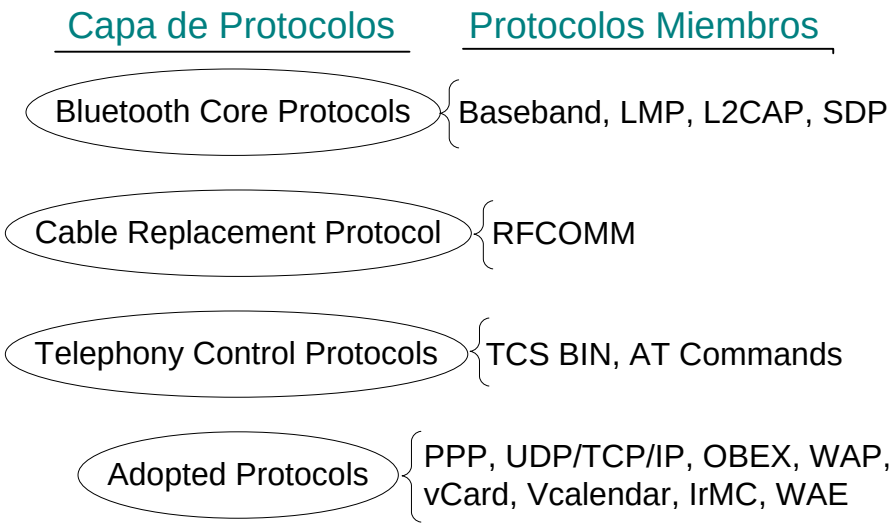

Figura. 2. 2. Las capas en Bluetooth y sus Protocolos

Mencionando los protocolos del núcleo de Bluetooth encontramos al protocolo ***Baseband*** (BB) el cual permite los dos tipos de enlaces físicos con los respectivos paquetes entre los dispositivos Bluetooth en una piconet. Además abriendo únicamente un enlace de audio, en enlaces SCO los paquetes con datos de audio son enviados directamente en banda base entre los dispositivos Bluetooth. Asimismo, el protocolo de banda base cuenta con un chequeo de redundancia cíclico *CRC* de 16 bits para controlar la integridad en paquetes ACL (que tienen secuencia) y brinda conexiones fiables mediante el mecanismo *ARQ* (*Automatic Repeat Request*).

El ***LMP*** es responsable del establecimiento del enlace y el control en dispositivos Bluetooth. Controla los ciclos de trabajo, los modos de potencia y los estados de conexión entre dispositivos en una piconet. Este protocolo es utilizado para la seguridad (autenticación y encriptación) y tiene mayor prioridad que los datos de un usuario.

El ***L2CAP*** se utiliza únicamente en enlaces ACL, permite la transmisión y recepción de paquetes de datos con una tasa de hasta 64 Kbps en aplicaciones, contienes QoS puesto que siempre se dará prioridad a los datos del usuario y es encargado de realizar la segmentación y la unificación de paquetes.

El protocolo ***SDP*** permite descubrir servicios o características de servicios que están disponibles en dispositivos Bluetooth vecinos.

El protocolo de reemplazo de cables es el ***RFCOMM***, el cual permite emular un puerto serial RS-232 y sus señales de control y datos en bandabase con el protocolo L2CAP. El protocolo RFCOMM es encargado de la conexión directa entre dispositivos Bluetooth (configuración directa) o de la conexión de un dispositivo y el módem de red en caso de tener una configuración con conexión vía Bluetooth por un extremo y una red cableada en otro.

El protocolo para el control de telefonía es el ***TCS-BIN***, el cual permite el establecimiento de llamadas telefónicas mediante señales de control y operaciones en bits. Este protocolo se ejecuta sobre la capa L2CAP y menciona los procedimientos para el manejo de grupos de dispositivos Bluetooth.

Existen protocolos ya existentes y utilizados en varias aplicaciones que han sido adoptados por Bluetooth con el fin de permitir la interoperabilidad y facilitar el uso de antiguas o desarrollo de nuevas aplicaciones en base a esta tecnología. Estos protocolos se encuentran en las capas más altas y entre ellos se menciona a ***PPP***, que especifica la manera en que datagramas IP son trasmitidos en un enlace punto a punto.

Otros protocolos adoptados son el ***TCP/UDP/IP*** que son empleados para la comunicación en Internet. El protocolo de transferencia ***OBEX*** que está basado en un modelo cliente-servidor, cuenta con tres aplicaciones basadas en él (SYNC, FTP y OPP) y especifica los protocolos de comunicación y los objetos de datos que se requieren para el intercambio de objetos. El protocolo ***WAP*** permite el acceso a Internet, enviar y leer el contenido o mensajes en dispositivos inalámbricos que cuenten con pantallas, por lo que utiliza estándares como HTML, HTTP, TLS y TCP para redes móviles. El protocolo ***WAE*** proporciona el entorno con WWW (*World Wide Web*) y tecnologías móviles contando con lenguajes de WML (*Wireless Markup Language* – parecido a HTML pero realizado para terminales móviles), WMLScript (basado en JavaScript) y aplicaciones de telefonía inalámbrica (*WTA*) que son extensiones específicas que permiten controlar y acceder a aplicaciones de agenda de teléfonos o calendario.

2.2. MANEJO DE LOS ENLACES

La entidad encargada de descubrir otros dispositivos de Bluetooth que se encuentren en el perímetro de acción es el Link Manager (LM) y la

información intercambiada punto a punto entre los dispositivos se realiza con el protocolo LMP (*Link Manager Protocol*).

El LM al ser encargado de descubrir y comunicarse con otros Link Mangers de otros dispositivos también realiza el establecimiento del enlace, la autenticación, encriptación, configuración y la negociación de los tamaños de paquetes en banda base. Para todas estas funciones el LM hace uso de los servicios del Link Control (LC - controlador del enlace).

El LC realiza el procedimiento de autenticación, fija el tipo de enlace, asigna un tipo de frame o cuadro a ser utilizado, detecta errores, realiza retransmisiones y pone en estado de espera a los dispositivos que no tienen que estar activos en ciertas transmisiones.

El LMP es un protocolo orientado a paquetes que se emplea para el control del establecimiento lógico del enlace, la creación de nuevos enlaces lógicos y la confidencialidad. Estos paquetes son llamados PDU (*Protocol Data Unit*) y son de tamaño limitado para garantizar que entren en un time slot [11]. La Figura 2.3 presenta la posición del LMP en Bluetooth.

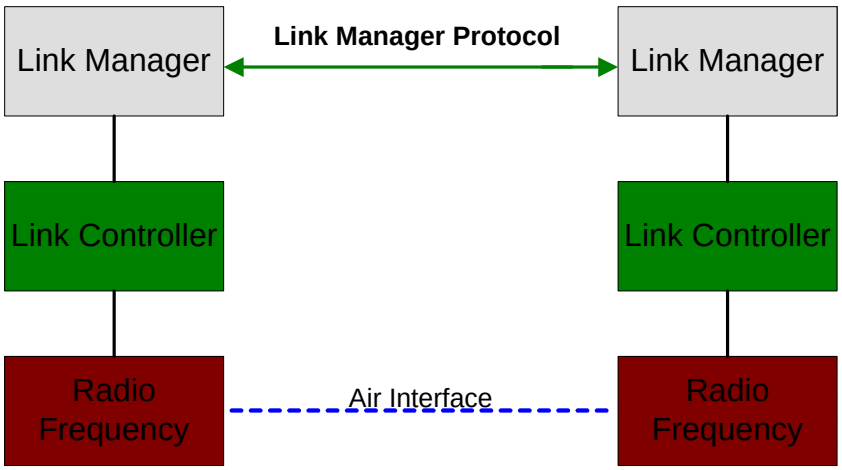

Figura. 2. 3. Posición de LMP en la Bluetooth

2.2.1. Tipos de PDU. Cada PDU cuenta con un op code de 7 bits y puede ser obligatorio u opcional. La Tabla 2.1 presenta los 57 PDUs disponibles en la especificación Bluetooth, donde las funciones pueden ser obligatorias (M) u opcionales (O).

Tabla. 2. 1. PDUs existentes en LMP [12]

Tipo	PDU LMP	Bytes	OP code	Sentido	Contenido
Respuesta General (M)	LMP_accepted	2	3	M a S	Op code
	LMP_not_accepted	3	4	M - S	Op code Razón
Autentificación (M)	LMP_au_rand	17	11	M - S	Número random
	LMP_sres	5	12	M - S	Respuesta de autentificación
Emparejamiento Pairing (M)	LMP_in_rand	17	8	M - S	Número random
	LMP_au_rand	17	11	M - S	Número random
	LMP_sres	5	12	M - S	Respuesta de autentificación
	LMP_comb_key	17	9	M - S	Número random
	LMP_unit_key	17	10	M - S	Llave
Cambio de Clave del Enlace (M)	LMP_comb_key	17	9	M - S	Número random
	LMP_unit_key	17	10	M - S	Llave
Cambio de Clave Actual del Enlace (M)	LMP_temp_rand	17	13	M a S	Número random
	LMP_temp_key	17	14	M a S	Llave
	LMP_use_semi_permanent_key	1	50	M a S	
Encripción (O)	LMP_encryption_mode_req	2	15	M - S	Modo encriptación
	LMP_encryption_key_size_req	2	16	M - S	Tamaño de llave
	LMP_start_encryption_req	17	17	M a S	Número random
	LMP_stop_encryption_req	1	18	M a S	

Tipo	PDU LMP	Bytes	OP code	Sentido	Contenido
Petición de ajuste (offset) del reloj (M)	LMP_clkoffset_req	1	5	M a S	
	LMP_clkoffset_res	3	6	S a M	Offset del clock
Petición de ajuste de slot (O)	LMP_slot_offset	9	52	M - S	Offset del slot Dirección del dispositivo BT
Petición de la información de precisión de la sincronización (M)	LMP_timing_accuracy_req	1	47	M - S	
	LMP_timing_accuracy_res	1	48	M - S	Dirección
Versión del LMP (M)	LMP_version_req	6	37	M - S	Número de versión***** ID de la compañía****** Número de la Sub-versión
	LMP_version_res	6	38	M - S	Número de versión ID de la compañía Número de la Sub-versión
Características soportadas (M)	LMP_features_req	9	39	M - S	Características
	LMP_features_res	9	40	M - S	Características
Cambio de rol mestro-esclavo (O)	LMP_switch_req	1	19	M - S	
	LMP_slot_offset	9	52	M - S	Offset del slot Dirección del dispositivo BT
Petición de nombre (M)	LMP_name_req	2	1	M - S	Nombre del offset
	LMP_name_res	17	2	M - S	Nombre del offset Nombre de longitud Nombre fragmento
Separación (M)	LMP_detach	2	7	M - S	Razón

Tipo	PDU LMP	Bytes	OP code	Sentido	Contenido
Modo Hold (O)	LMP_hold	3	20	M - S	Tiempo de Hold
	LMP_hold_req	3	21	M - S	Tiempo de Hold
Modo Sniff (O)	LMP_sniff	10	22	M a S	Banderas de control Intento de sniff Interrupción de sniff
	LMP_sniff_req	10	23	M - S	Banderas de control Intento de sniff Interrupción de sniff
	LMP_unsniff_req	1	24	M - S	
Modo Park (O)	LMP_park_req	1	25	M - S	
	LMP_unpark_PM_ADDR_req	varía	30	M a S	Banderas de control Direcciones de miembros activos y parqueados
	LMP_unpark_BD_ADDR_req	varía	29	M a S	Banderas de control Dirección de miembro activo y del dispositivo BT
	LMP_set_broadcast_scan_window	4 o 6	27	M a S	Banderas de control Ventana examinar broadcast
	LMP_modify_beacon	11 o 13	28	M a S	Control sincronismo
	LMP_park	17	26	M a S	Banderas de control
Control de potencia (O)	LMP_incr_power_req	2	31	M - S	
	LMP_decr_power_req	2	32	M - S	
	LMP_max_power	1	33	M - S	
	LMP_min_power	1	34	M - S	

Tipo	PDU LMP	Bytes	OP code	Sentido	Contenido
Cambio de la calidad manejada en el canal (O)	LMP_auto_rate	1	35	M - S	
	LMP_preferred_rate	2	36	M - S	Tasa de datos
Calidad de servicio (M)	LMP_quality_of_service	4	41	M a S	Intervalo encuenta
	LMP_quality_of_service_req	4	42	M - S	Intervalo encuenta
Enlaces SCO (O)	LMP_SCO_link_req	7	43	M - S	Manejo de SCO Banderas de control Paquete SCO Modo aire
	LMP_remove_SCO_link_req	3	44	M - S	Manejo SCO Razón
Control de paquetes multi-slot (M)	LMP_max_slot	2	45	M a S	Máximo de slots
	LMP_max_slot_req	2	46	S a M	Máximo de slots
Esquema de paging (O)	LMP_page_mode_req	3	53	M - S	Esquema paging Parámetros
	LMP_page_scan_mode_req	3	54	M - S	Esquema paging Parámetros
Supervisión del enlace (M)	LMP_supervision_timeout	3	55	M - S	Supervisión de interrupción
Establecimiento de la conexión (M)	LMP_host_connection_req	1	51	M - S	
	LMP_setup_complete	1	49	M a S	
El modo Prueba (M)	LMP_test_active	1	56	M a S	
	LMP_test_control	1	57	M a S	
Manejo de errores (M)	LMP_not_accepted	3	4	M - S	Op code Razón

***Número de versión**

Tabla. 2. 2. Especificación LMP en Bluetooth y su código

Código	Versión LMP
0	Bluetooth LMP Especificación 1.0
1	Bluetooth LMP Especificación 1.1
2	Bluetooth LMP Especificación 1.2
3-255	Reservados

**** ID de la compañía**

Tabla. 2. 3. ID de compañías y su código en Bluetooth

Código	Compañía
0	Ericsson Mobile Communications
1	Nokia Mobile Phones
2	Intel Corp.
3	IBM Corp.
4	Toshiba Corp.
5	3Com
5-65534	Reservados

2.3. CONTROL LÓGICO DEL ENLACE

El protocolo para el control lógico y adaptación del enlace es llamado L2CAP (***Logical Link Control and Adaptation Protocol***). Este protocolo se encuentra sobre el protocolo de banda base y es utilizado para servicios

de datos orientados y no orientados a la conexión. Entre las funciones del protocolo L2CAP se encuentran la segmentación y unificación de paquetes, la múltiplexación para protocolos de más alto nivel, brindar calidad de servicio QoS a la información entre puntos terminales Bluetooth, y permitir a protocolos de alto nivel la transmisión y recepción de paquetes L2CAP con longitud máxima de 64Kilobytes.

El protocolo L2CAP se define únicamente para enlaces ACL (máximo uno entre dos dispositivos), es capaz de reconocer protocolos superiores como TCS-BIN, SDP, RFCOMM, entre otros, en el momento de la múltiplexación, y hace uso de verificadores ubicados en la capa de banda base para que información transmitida sea protegida. La Figura 3.4 presenta la ubicación de L2CAP en arquitectura de capas de Bluetooth [1].

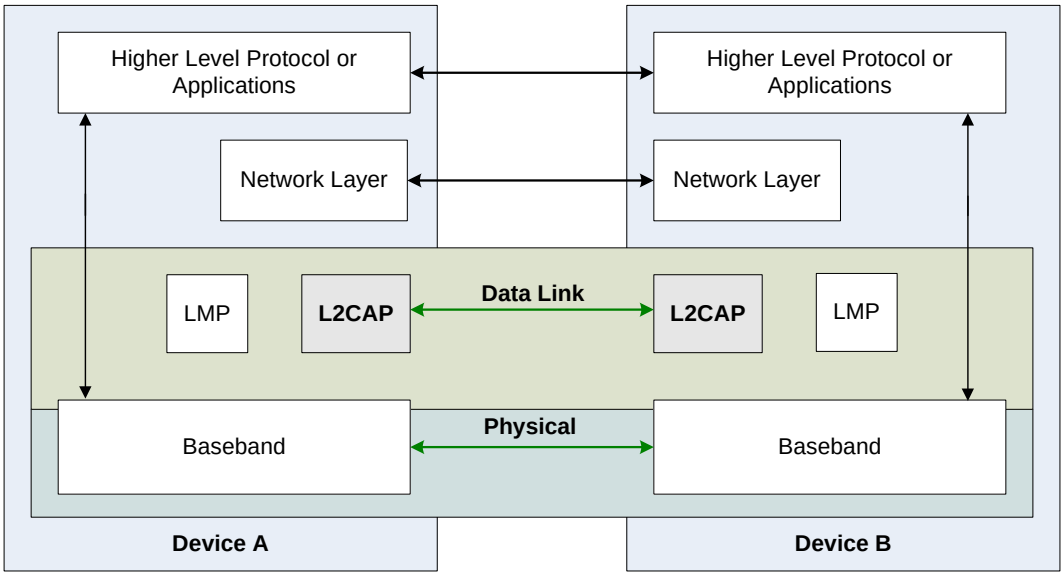

Figura. 2. 4. Ubicación de L2CAP en la capa de enlace de datos

En la utilización de L2CAP se debe considerar dos situaciones con respecto al audio. La primera es cuando se requiere calidad de voz en los canales de comunicación, que sucede en aplicaciones audio y telefonía, aquí por utilizar enlaces SCO en banda base la capa L2CAP no es utilizada. Por otra parte, la segunda consideración puede darse en la

telefonía IP, en la cual se cuenta con paquetes de datos de audio que son considerados datos y hacen uso de protocolos de comunicación sobre L2CAP.

Algunas características sobre L2CAP también son el no brindar un canal fiable para multicast, no asegura la integridad de la información puesto que no cuenta con checksums, no realiza retransmisiones y no maneja un nombre específico para un grupo.

En el protocolo L2CAP existen identificadores de canales (***CID – Channel Identifier***) que pueden ser reservados o no. Como ejemplo, identificadores reservados se tiene para aplicaciones particulares de L2CAP en el caso del tráfico de datos que ingresa sin conexión (CID unidireccional), o para la señalización (CID bidirecional), con la cual se negocia cambios de características de canales de datos que son orientados a la conexión.

Las implementaciones L2CAP, como forma de brindar consistencia, hacen uso de la longitud de las cabeceras de paquetes L2CAP para verificar que los paquetes concuerden en longitud, en caso de no hacerlo, los paquetes serán desechados (si no se requiere fiabilidad de canal), o en el caso de requerir fiabilidad de canal, se informará a la capa superior de la posibilidad de tener un canal no fiable (dependiendo del valor del *flush timeout*).

En la implementación de la capa L2CAP se suscitan eventos y acciones para que se de la comunicación, entre ellos se encuentran Peticiones (Req) y Confirmaciones (Cfm) en la parte del cliente e Indicaciones (Ind) y Respuestas (Rsp) en la parte del servidor. Aquí, las respuestas que requieren de un largo procesamiento son llamadas Pendientes (Pnd). Las notificaciones de Confirmaciones y Respuestas se asumen positivamente, mientras que si resultan negativas se agregará un sufijo de negación de la siguiente manera: L2CAP_ConnectConfNeg. La Figura 2.5 indica los eventos y acciones que ocurren entre capas [1].

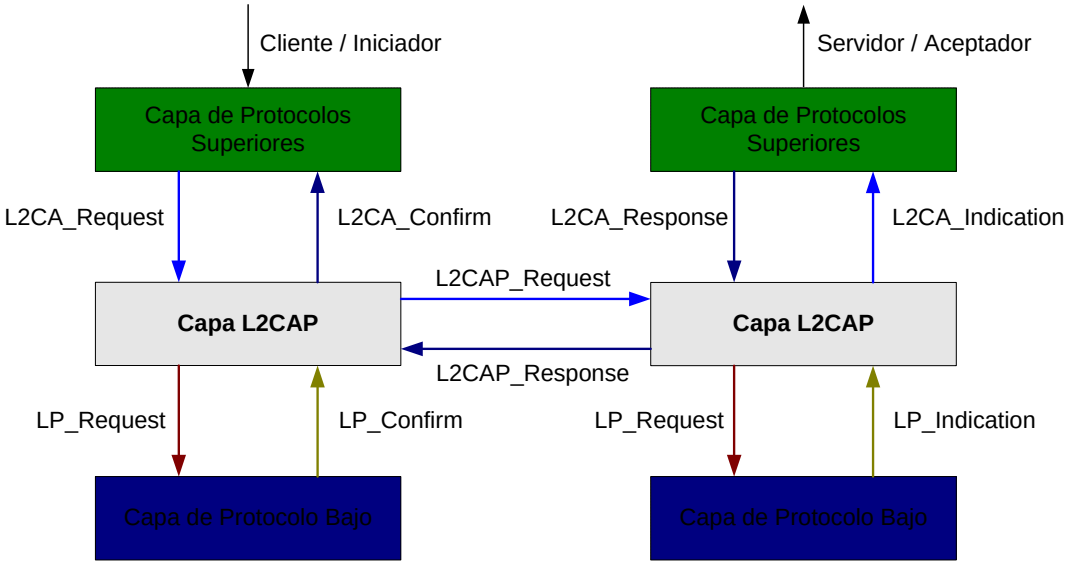

Figura. 2. 5. Eventos y acciones en diferentes capas

Los eventos y acciones se clasifican de distinta manera, la Tabla 2.4 presenta esta clasificación.

Tabla. 2. 4. Acciones y eventos que ocurren [1]

E V E N T O S		
Eventos de Capas Bajas a L2CAP	LP_ConnectCfm	Confirma la petición de establecer conexión con la capa más baja
	LP_ConnectCfmNeg	Confirma no poder establecer la conexión con la capa más baja
	LP_ConnectInd	Indica que el protocolo de abajo ha establecido la conexión
	LP_DisconnectInd	Indica que el protocolo de abajo ha sido apagado por comandos LMP o eventos de interrupción
	LP_QoSCfm	Confirma la petición para una calidad de servicio dada
	LP_QoSCfmNeg	Confirma no poder brindar una calidad de servicio dada
	LP_QoSViolationInd	Indica que el protocolo más bajo detectó una violación del acuerdo de QoS especificado en LP_QoSReq
Eventos de Datos L2CAP a L2CAP	L2CAP_Data	Es el único mensaje que se intercambia entre entidades L2CAP para indicar que un paquete de datos ha sido recibido
Eventos de Señalización L2CAP a L2CAP	L2CAP_ConnectReq	Un paquete de petición de conexión ha sido recibido
	L2CAP_ConnectRsp	Un paquete de respuesta de conexión ha sido recibido positivamente y la conexión fue establecida
	L2CAP_ConnectRspPnd	Un paquete de respuesta de conexión ha sido recibido e indicando que el terminal remoto recibió la petición y lo está procesando
	L2CAP_ConnectRspNeg	Un paquete de respuesta de conexión se recibió e indica que no se estableció la conexión
	L2CAP_ConfigReq	Un paquete de petición de configuración se recibió indicando que el terminal remoto quiere negociar parámetros del canal
	L2CAP_ConfigRsp	Un paquete de respuesta de configuración se recibió indicando que el terminal remoto está de acuerdo con todos los parámetros siendo negociados
	L2CAP_ConfigRspNeg	Un paquete de respuesta de configuración se recibió indicando que el terminal remoto no está de acuerdo con los parámetros recibidos en el paquete de respuesta

E V E N T O S		
Eventos de señalización L2CAP a L2CAP	L2CAP_DisconnectReq	Un paquete de petición de desconexión se recibió indicando que el canal debe empezar el proceso de desconexión. Seguido a esto, una entidad de L2CAP retorna su CID local al grupo de CIDs no asiganados
	L2CAP_DisconnectRsp	Un paquete de respuesta de desconexión se recibió. Seguido a esto, se libera el CID local al grupo de CIDs no asignados
Eventos de Capas Altas a L2CAP	L2CA_ConnectReq	Una petición de la capa superior para crear un canal para un dispositivo remoto.
	L2CA_ConnectRsp	Una respuesta de la capa superior a la indicación de petición de conexión del equipo remoto
	L2CA_ConnectRspNeg	Una respuesta negativa de la capa superior a la petición de conexión del equipo remoto
	L2CA_ConfigReq	Una petición de la capa superior para (re)configurar el canal
	L2CA_ConfigRsp	Una respuesta de la capa superior a la indicación de petición de (re)configurar el canal
	L2CA_ConfigRspNeg	Una respuesta negativa de la capa superior a la indicación de petición de (re)configurar el canal
	L2CA_DisconnectReq	Una petición de la capa superior para la inmediata desconexión de un canal
	L2CA_DisconnectRsp	Una respuesta de la capa superior a la indicación de petición de desconexión
	L2CA_DataRead	Una petición de la capa superior para la transferencia de datos recibidos de la entidad L2CAP a la capa superior
	L2CA_DataWrite	Una petición de la capa superior para la transferencia de datos de la capa superior hacia la entidad L2CAP para transmisión sobre un canal abierto
Eventos de Timer	RTX	Utilizado para terminar el canal cuando el terminal remoto no responde peticiones de señalización. Se activa cuando una petición de señalización es enviada y se desactiva cuando la respuesta es recibida. Así, si el cronómetro expira se puede retransmitir un duplicado y aumentando o no el tiempo de espera o desactivar el canal especificado en la petición

E V E N T O S		
Eventos del Timer	ERTX	Utilizado en lugar del cronómetro RTX cuando se sospecha que el terminal remoto realiza un procesamiento adicional de la señal de petición. Se activa cuando el terminal remoto responde que una petición está pendiente (L2CAP_ConnectRspPnd) y se desactiva cuando se recibe una respuesta formal o el enlace físico es perdido
A C C I O N E S		
Acciones de L2CAP a Capas Bajas	LP_ConnectReq	L2CAP pide al protocolo más bajo crear una conexión. En caso de no existir enlace físico hacia el dispositivo remoto, éste mensaje se envía al protocolo más bajo para establecer la conexión física. Continuando con el proceso, la capa baja retorna una confirmación o negación de la petición (L2CAP_ConnectCfm o L2CAP_ConnectCfmNeg)
	LP_QoSReq	L2CAP pide al protocolo más bajo fijar un partámetro de QoS. Siguiendo el proceso, la capa baja retorna un L2CAP_QoSCfm o L2CAP_QoSCfmNeg para indicar que si la petición fue aceptada o rechazada
	LP_ConnectRsp	Indica una respuesta positiva aceptando la petición de la indicación de conexión previa
	LP_ConnectRspNeg	Indica una respuesta negativa a la petición de la indicación de conexión previa
Acciones de Señalización L2CAP a L2CAP	Las mismas acciones de L2CAP a Capas Bajas pero haciendo referencia a la trasmisión de mensajes, mas no a la recepción	
Acciones de Datos de L2CAP a L2CAP	Similar al evento de datos L2CAP a L2CAP, pero ahora se refiere a la transmisión de datos mas no a la recepcción	

A C C I O N E S		
Acciones de L2CAP a Capas Altas	L2CA_ConnectInd	Indica que una petición de conexión ha sido recibida de un dispositivo remoto
	L2CA_ConnectCfm	Confirma que una petición de conexión ha sido aceptada
	L2CA_ConnectCfmNeg	Confirmación negativa para la petición de conexión
	L2CA_ConnectPnd	Confirma que una respuesta de conexión ha sido recibida del dispositivo remoto
	L2CA_ConfigInd	Indica que una petición de configuración ha sido recibida de un dispositivo remoto
	L2CA_ConfigCfm	Confirma que una petición de configuración ha sido aceptada
	L2CA_ConfigCfmNeg	Confirmación negativa a una petición de configuración
	L2CA_DisconnectInd	Indica que una petición de desconexión ha sido recibida de un dispositivo remoto o ha sido desactivado por no responder peticiones de señalización
	L2CA_DisconnectCfm	Confirma que una petición de desconexión ha sido procesada por el dispositivo remoto
	L2CA_TimeOutInd	Indica que un cronómetro RTX o ERTX ha expirado
	L2CA_QoSViolationInd	Indica que un acuerdo de QoS ha sido violado

De igual manera, existen también Estados Operacionales del Canal que se presentan cuando el canal está siendo configurado, la Tabla 2.5 muestra lo mencionado.

En el Anexo 1 se presenta los eventos y las acciones que suceden en un estado particular del canal.

Tabla. 2. 5. Estados operacionales del canal

ESTADOS OPERACIONALES DEL CANAL	
CLOSED	En este estado no existe canal asociado a un CID. Es el único estado donde no existe enlace de conexión
W4_L2CAP_CONNECT_RSP	El CID representa un terminal local y un mensaje L2CAP_ConnectReq ha sido enviado con referencia a este punto, esperando así a un L2CAP_ConnectRsp
W4_L2CA_CONNECT_RSP	El terminal remoto existe y un L2CAP_ConnectReq ha sido recibido por la entidad local L2CAP, por lo que envía un L2CA_ConnectInd y espera la respuesta
CONFIG	La conexión ya fue establecida pero se sigue negociando los parámetros del canal. Aquí, todo el tráfico de entrada y salida es suspendido, se deben emplear mensajes de L2CAP_ConfigReq en ambos terminales, y para pasar al estado OPEN se requiere que ambos estén listos
OPEN	La conexión ya fue establecida y configurada, por lo que el flujo de datos puede proceder
W4_L2CAP_DISCONNECT_RSP	La conexión se está cerrando y un mensaje de L2CAP_DisconnectReq fue enviado, esperando así una respuesta
W4_L2CA_DISCONNECT_RSP	La conexión en el terminal remoto se está cerrando, se recibió un mensaje L2CAP_DisconnectReq, un L2CA_DisconnectInd fue enviado a la capa superior para conocer qué CID se está cerrando y se espera una respuesta para luego responder al terminal remoto

Pese a que L2CAP está basada en paquetes, la utilización de canales es necesaria para la representación del flujo de datos. Al existir dos tipos de servicios de datos, orientados y no orientados a la conexión, también se

guarda un formato para los canales mencionados. La Figura 2.6 y Figura 2.7 presentan el formato de los paquetes L2CAP orientados y no orientados a la conexión respectivamente.

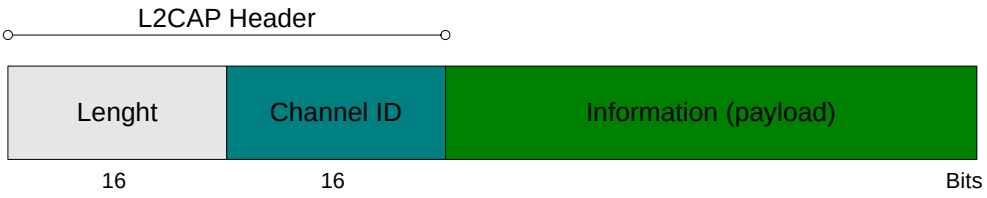

Figura. 2. 6. Formato de paquete L2CAP orientado a la conexión

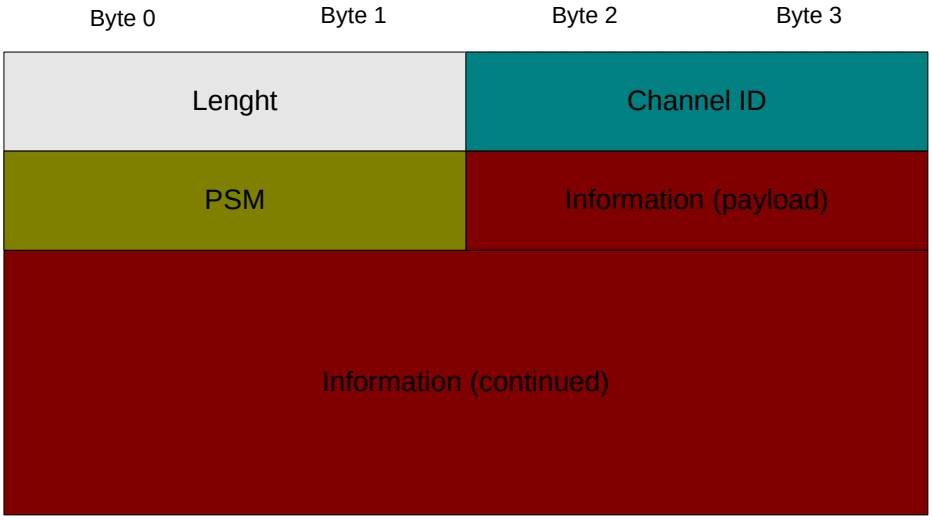

Figura. 2. 7. Formato de paquete L2CAP no orientado a la conexión

Para ambas figuras, en el campo de *Lenght* se especifica la longitud del payload y del campo PSM (***Protocol/Service Multiplexer***, en caso de existir), *Channel ID* menciona el terminal del canal de destino del paquete o el grupo de destino del paquete, PSM que menciona protocolos y valores empleados junto a SDP y el campo de Information contiene la carga de información.

En Bluetooth también se puede enviar múltiples **comandos de señalización** (conexión, configuración, desconexión, echo, o información), de los 11 existentes, en un solo paquete L2CAP. La implementación

L2CAP permitirá identificar la BD_ADDR del dispositivo que envió el comando. Cada comando cuenta con un código propio que pasa de una entidad L2CAP a otra y puede ser de Request o Response únicamente. Asimismo, todos los comandos de señalización, ya sean ubicados en uno o varios paquetes, serán enviados al CID 0x0001 o canal de señalización. Además de la señalización, existen paquetes de opción para fijar o negociar requerimientos de conexión. Son 4 **opciones de parámetros de configuración** (MTU, *Flush TimeO*ut, *QoS*, y *Request/Response Path*) para ser negociados entre entidades L2CAP.

CAPÍTULO 3

PERFILES GENERALES Y ESPECÍFICOS DE BLUETOOTH

Los perfiles de Bluetooth pueden ser considerados como cortes verticales de la pila de protocolos existentes. Los perfiles definen protocolos, características, procedimientos y mensajes específicos que soportan particulares modelos utilizados, por esta razón existen opciones y parámetros que deben cumplirse al momento de utilizar los diferentes perfiles. Con ayuda de perfiles definidos el uso de un servicio puede lograrse sin inconvenientes entre dispositivos de diferentes fabricantes.

En Bluetooth existen perfiles generales y específicos, pero cada uno de ellos debe ser implementado de tal manera que cuente con las características necesarias para que dicha implementación funcione de la misma forma en cualquier dispositivo sin importar su fabricación. Se puede considerar cuatro perfiles generales que se emplean en varios modelos utilizados, como son GAP (***Generic Access Profile*)**, SPP (***Serial Port Profile***), SDAP (***Service Discovery Application Profile***), y GOEP (***Generic Object Exchange Profile***). De igual manera existen otros perfiles para diferentes modelos de uso, como son el ICP (***Intercom Profile***), HS (***Headset Profile***), FTP (***File Transfer Profile***), OPush u OPP (***Object Push Profile***), o Sync (***Synchronization Profile***), y cada uno de ellos es dependiente de cierta forma de algún perfil general e inclusive un mismo perfil general puede estar relacionado con otro, la Figura 3.1 presenta la relación entre el perfil de acceso genérico y otros perfiles de Bluetooth. A continuación se presentan perfiles generales y específicos empleados para la transmisión de voz y de objetos de forma detallada.

3.1. GENERIC ACCESS PROFILE

La razón de existir el perfil de acceso genérico es para describir la forma de uso de las capas bajas de la pila de protocolos de Bluetooth (LM y LMP) para especificar procedimientos generales para la localización (identidad, nombre, y capacidades básicas) de dispositivos Bluetooth y el manejo del enlace para la correspondiente conexión de los mismos. Este perfil también hace mención a capas superiores como L2CAP, RFCOMM y OBEX para brindar alternativas de seguridad a los procedimientos.

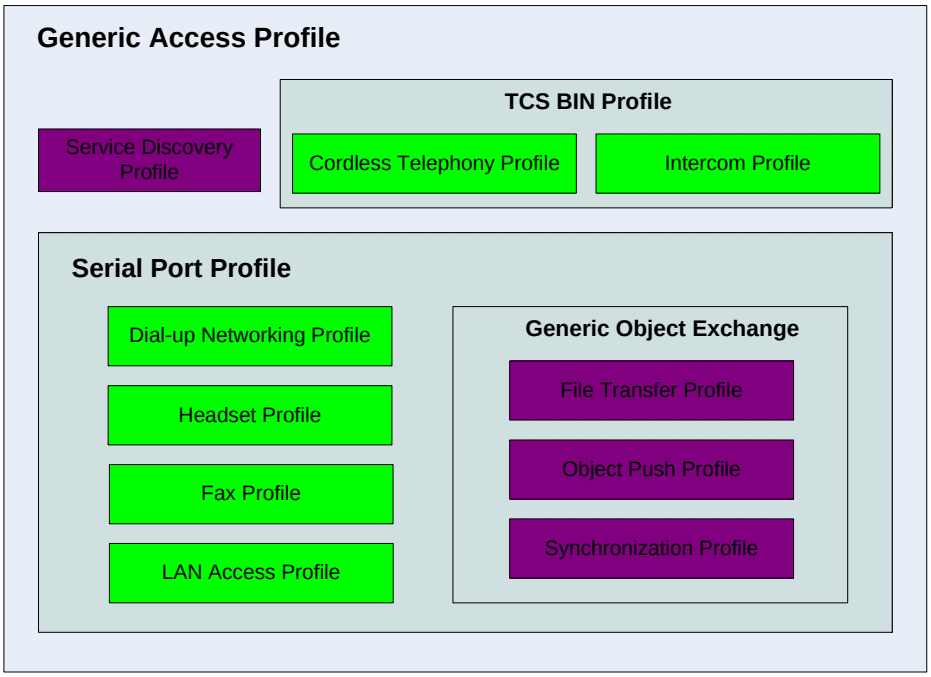

Figura. 3. 1. Relación del perfil de acceso genérico con otros perfiles de Bluetooth

Entre los parámetros básicos que un dispositivo Bluetooth debe soportar se encuentra:

- *Device Names*, que permite dar un nombre con hasta 248 bytes a un dispositivo.
- *Bluetooth PIN (personal identification number)*, que ayuda en el proceso de autenticación.

- *Class Device,* que indica el tipo de dispositivo que es y los tipos de servicios que soporta.
- *Discovery Modes*, existe dos modos, el modo *non-discoverable* que impide respuestas a cualquier petición de dispositivos Bluetooth y el modo *discoverable* que se subdivide en limitado y general, el primero está sujeto a periodos de tiempo específicos, mientras que el general se encuentra listo para responder en todo momento, excepto cuando se está realizando alguna otra actividad en banda base.
- *Pairing Modes*, son esenciales para crear el enlace común que será utilizado. Existen dos modos, el *Pairable* y el *Non-Pairable*, en el primero se acepta la creación de vínculos que fue iniciada por un dispositivo remoto, mientras que en el segundo modo no.
- *Security Modes*, existen tres tipos de seguridad, 1, 2, y 3. En seguridad 1, nunca se inicia un procedimiento de seguridad; en seguridad 2 no inicia un procedimiento seguro hasta que se recibe o inicia el procedimiento para establece un canal; mientras que en seguridad 3 se inicia procedimientos seguros antes de enviarse el mensaje de fijación de enlace completo (LMP_link_setup_complete). Un ejemplo de seguridad 3 sería permitir comunicación únicamente con dispositivos pareados previamente [1].

En el perfil GAP existe el procedimiento *Device Discovery* que brinda al dispositivo iniciador la información de la dirección, el nombre, la clase y el reloj de los dispositivos que se encuentran en modo *Discoverable*. Toda esta información es requerida para iniciar procedimientos de establecimiento del enlace (físico tipo ACL), del canal (enlace lógico), y de la conexión entre dos dispositivos.

3.2. SERIAL PORT PROFILE

El perfil de Puerto Serial se encuentra en el perfil de Acceso Genérico y su función es setear los dispositivos Bluetooth con el fin de poder emular

puertos y cables seriales para poder obtener un canal orientado a la conexión que podrá soportar una tasa de transmisión de hasta 128Kbps (paquetes de un slot). Para ello se utiliza el protocolo de transporte RFCOMM que emula puertos RS-232 y corre sobre un canal L2CAP. En cuanto a la seguridad las características de implementación de autorización, autenticación y encriptación son opcionales. La figura 3.2 presenta un modelo para emular una conexión vía cable serial.

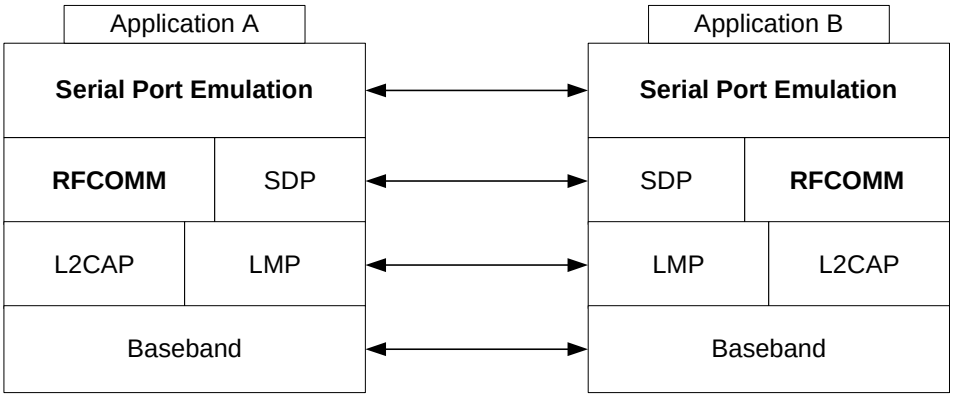

Figura. 3. 2. Modelo para emular una conexión vía cable serial [1]

Son tres los requerimientos que se deben cumplir para poder establecer una conexión vía cable serial emulada entre dos dispositivos se mencionan en los literales siguientes y de manera detallada en el Anexo 2.

a. Establecer el enlace y setear la conexión serial virtual.
b. Aceptar el enlace y establecer la conexión serial virtual.
c. Registrar el servicio en la base de datos local de SDP.

3.2.1. Requerimientos de Interoperabilidad de RFCOMM. Todos los dispositivos Bluetooth deben informar de todo cambio que se de en las señales de control RS-232 con el Comando de Estado del MODEM, aunque no es necesario utilizar todas las señales de control, sí es obligatorio hacer uso de la señal de control de flujo en la implementación.

También, este perfil permite hacer negociación del puerto remoto en cuanto a la tasa de baudios o paridad.

3.2.2. Requerimientos de Interoperabilidad de L2CAP. Debido a que el perfil de puerto serial únicamente permite canales orientados a la conexión, broadcast no podrá ser utilizado, aunque no existen restricciones para el uso de canales no orientados a la conexión junto a otros perfiles. Aquí, el único requisito para la señalización L2CAP es que el dispositivo iniciador realiza la petición de conexión L2CAP y ejecute el perfil.

3.2.3. Opciones de Configuración. Entre las opciones de configuración disponibles en este perfil se menciona a MTU, cuya restricción de tamaño dependerá de L2CAP pero deberá tener un mínimo de 48 bytes; el tiempo de vacío *(Flush Timeout)*, seteado por omisión en 0xFFFF; y QoS, la cual debido a que L2CAP soporta servicios del "mayor esfuerzo" es opcional y en caso de emplearse se requerirá mencionar los parámetros de variación de retardo, ancho de banda pico y latencia.

En cuanto a los requisitos de interoperabilidad con SDP, no existe ninguno, aunque en la base de datos del dispositivo remoto existirán entradas relacionadas al Puerto Serial como *ServiceClassIDList*, *ProtocolDescriptorList*, y *ServiceName*.

3.3. SERVICE DISCOVERY APPLICATION PROFILE (SDAP)

El perfil SDAP es el encargado de encontrar servicios disponibles de las unidades Bluetooth aledañas, este descubrimiento de servicios se lo realiza mediante preguntas de acuerdo a la clase de servicio, sus atributos, o mediante browsing de servicios.

Para todos estos procedimientos de descubrimiento de servicios, el perfil SDAP hace uso de la aplicación de Servicio de Descubrimiento de Aplicación de Usuario (*SrvDscApp – Service Discovery Application*), la cual

es interfaz con el SDP del cliente para recibir respuestas y enviar preguntas de servicios de servidores SDP de dispositivos remotos, además de ser requisito en todo dispositivo Bluetooth para la localización de servicios [13]. La Figura 3.3 presenta la pila de protocolos que se emplean en el perfil SDAP, nótese que SDP requiere de un servicio de transporte orientado a la conexión en L2CAP y utiliza un enlace ACL con banda base para luego poder ser transmitidos los PDUs de SDP.

Los únicos requerimientos para la utilización de este perfil son que los dispositivos deben haber sido encendidos e inicializados previamente y que se haya creado un enlace legítimo de Bluetooth. Aquí, el perfil no menciona requerimiento alguno para que el dispositivo iniciador sea el maestro y los dispositivos remotos sus esclavos. La Figura 3.4 presenta los procedimientos que se realizan para la utilización de este perfil.

Entre las opciones de configuración disponibles para este perfil se menciona a MTU, cuya restricción de tamaño dependerá de L2CAP pero deberá tener un mínimo de 48 bytes; el tiempo de vacío (*Flush Timeout*), seteado por omisión en 0xFFFF; y QoS, la cual debido a que L2CAP soporta servicios del “mayor esfuerzo” es opcional y en caso de emplearse se requerirá mencionar los parámetros de variación de retardo, ancho de banda pico y latencia.

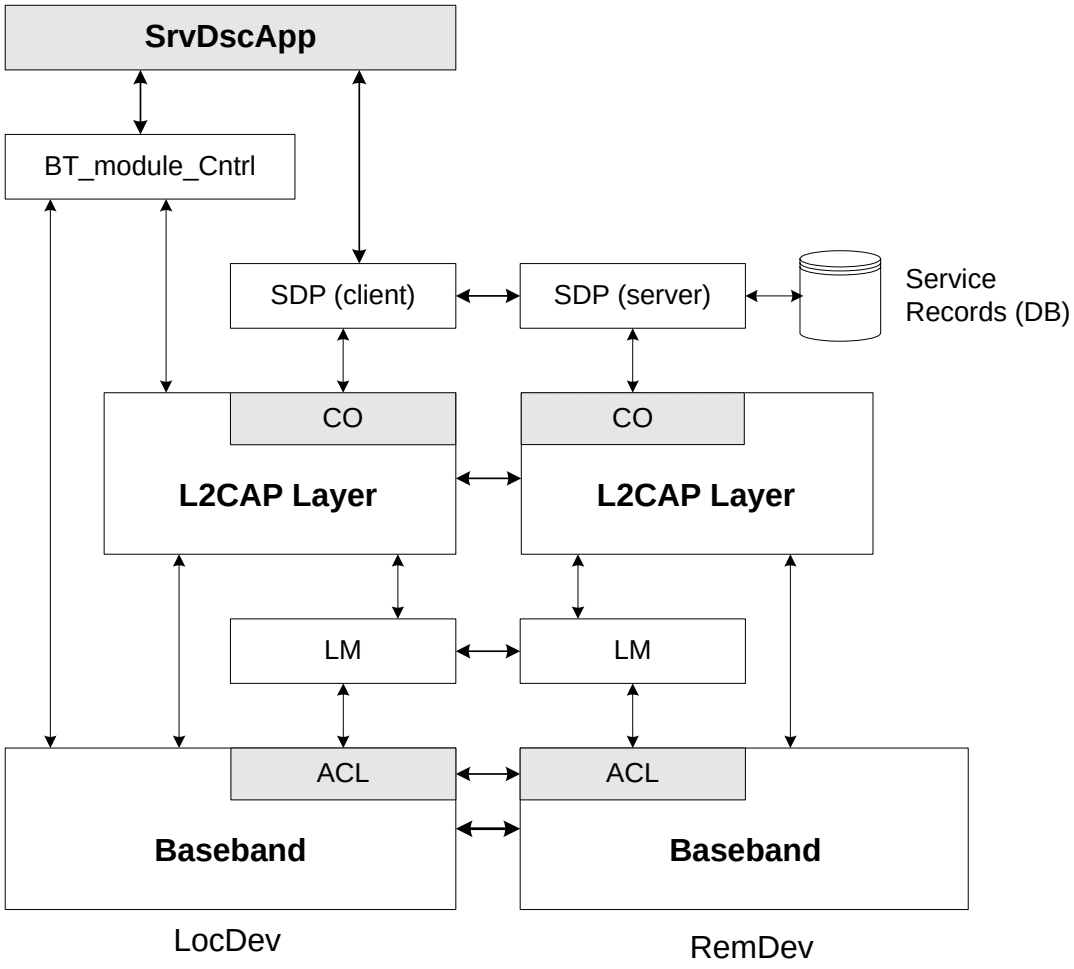

Figura. 3. 3. Pila de Protocolos Bluetooth en el SDAP [2]

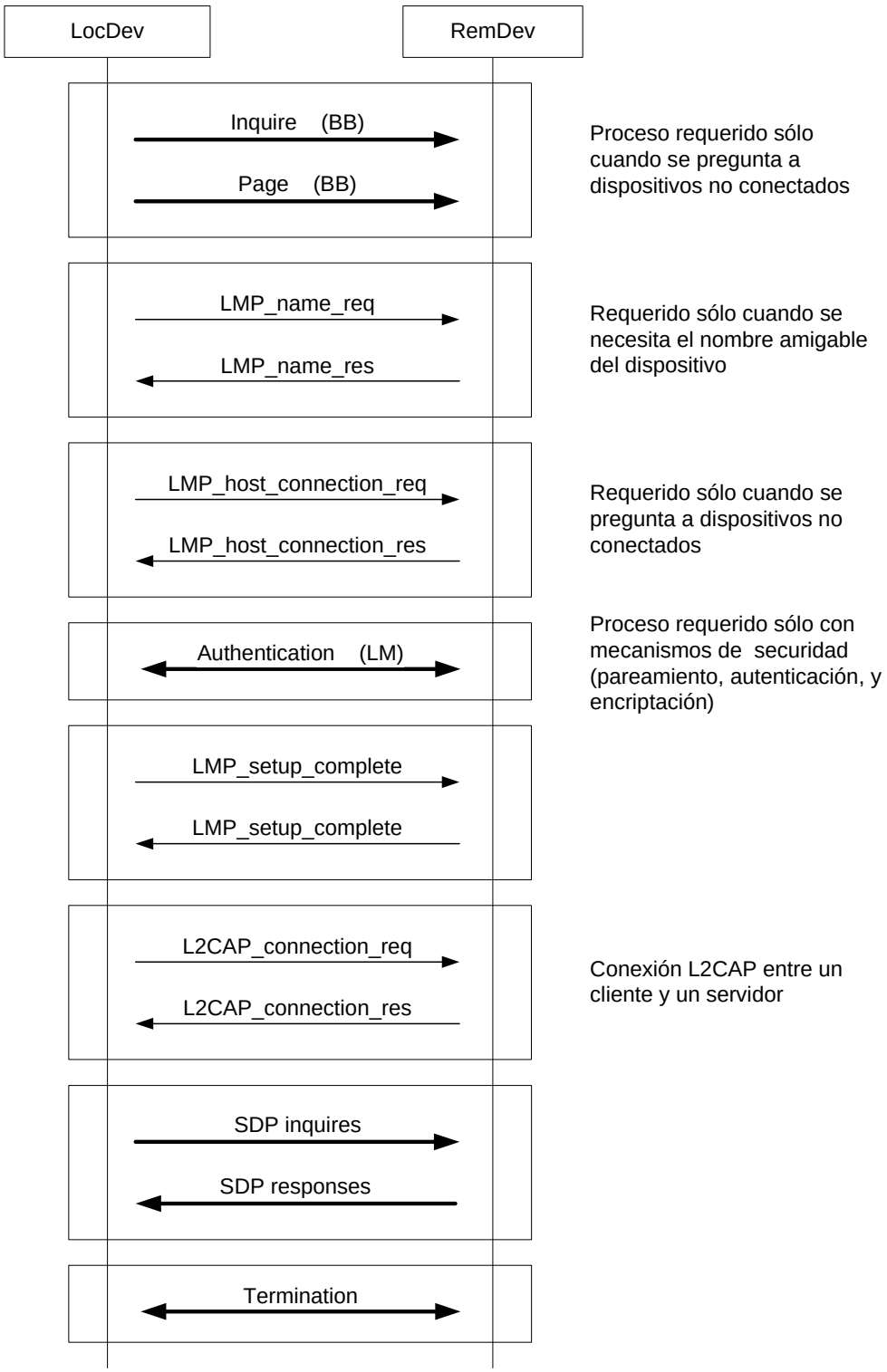

Figura. 3. 4. Procedimientos realizados en Bluetooth para utilizar el SDAP [2]

3.4. GENERIC OBJECT EXCHANGE PROFILE (GOEP)

El Perfil Genérico de Intercambio de Objetos (GOEP), como su nombre lo indica, define la manera en que los dispositivos Bluetooth deberán soportar modelos específicos con protocolos, procedimientos para el intercambio de objetos, y requerimientos necesarios de las capas inferiores (Bandabase y LMP). Con el uso de estos modelos se derivan a su vez nuevos perfiles asociados a GOEP que hacen uso del protocolo de intercambio de objetos OBEX y se presentan en la Figura 3.5, entre ellos se encuentran el Perfil de Transferencia de Archivos (FTP), el Perfil de Carga de Objetos (*OPP – Object Push Profile*) y el Perfil de Sincronización (SP – *Synchronization Profile*).

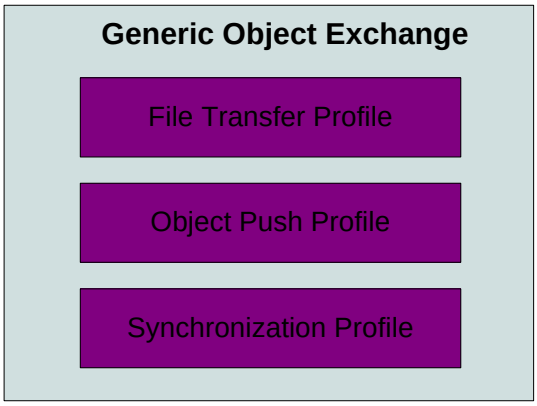

Figura. 3. 5. Perfiles asociados a GOEP

Debido a que el protocolo OBEX se basa para modelos cliente-servidor y facilita aplicaciones de carga y descarga de información o archivos, es decir se realiza “push” (el cliente envía objetos) y “pull” (cliente recupera objetos), los dispositivos más comunes que utilizan los perfiles mencionados anteriormente son PCs, PDAs, y teléfonos móviles que cuentan con la tecnología Bluetooth [2]. La Figura 3.6 presenta el modelo de protocolos utilizados en el perfil GOEP.

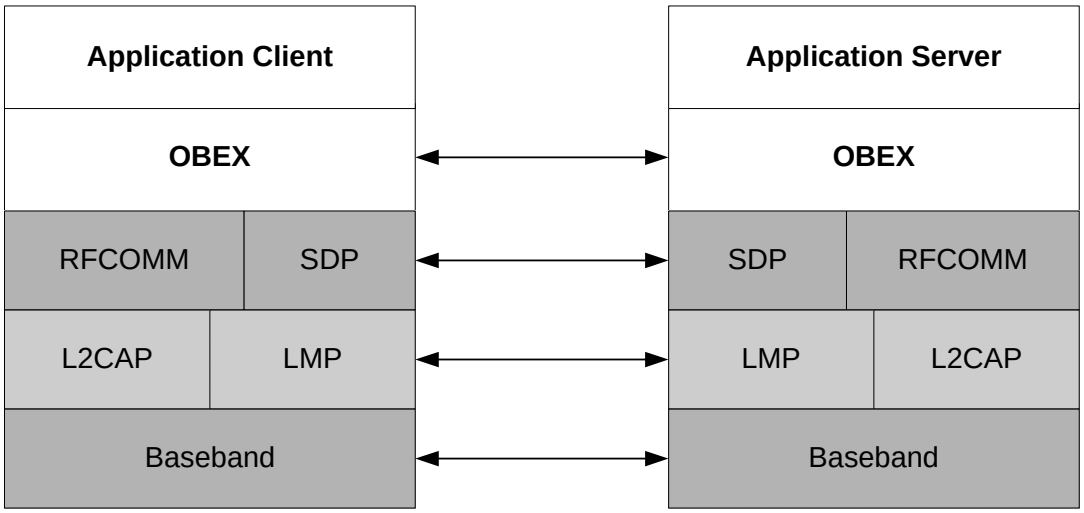

Figura. 3. 6. Modelo de protocolos de GOEP

Existen dos restricciones para el uso de este perfil y ambas están relacionadas con el servidor, la primera menciona que el servidor debe estar en modo *discoverable* y *connectable* para poder ser descubierto y posteriormente establecerse un enlace; la segunda restricción se refiere a que únicamente se podrá ofrecer servicios a un solo cliente puesto que el perfil está basado para configuraciones punto a punto, pero dicho impedimento puede ser modificado durante la implementación.

En este perfil, previo a que un servidor sea utilizado por un cliente, se debe realizar varios procedimientos como el de *bonding* (o atadura), el cual obliga al usuario a activarlo manualmente e ingresar un código PIN tanto por parte del cliente como del servidor; la inicialización de OBEX, cuya autenticación se realiza con el ingreso de una contraseña OBEX, idéntica o diferente al código PIN, que servirá para futuras autenticaciones (si la opción de autenticación OBEX está habilitada) posterior al establecimiento de conexiones OBEX; proveer mecanismos de seguridad; y el establecimiento del enlace y del canal de acuerdo a los procedimientos de GAP [2]. Una vez realizados los procedimientos mencionados, las opciones ofrecidas por este perfil son tres:

Tabla. 3. 1. Opciones ofrecidas por GOEP

No.	Opción
1	Establecer una sesión de intercambio de objetos. (Con autenticación o sin ella – Anexo 3)
2	Poner un objeto de datos (del cliente al servidor).
3	Retirar un objeto de datos (del servidor al cliente).

Dependiendo del perfil de aplicación que hará uso de GOEP, se deberán especificar no solo las operaciones necesitadas del protocolo OBEX sino también el tipo de cabeceras deberán ser soportadas para el correcto funcionamiento de los mismos. Entre las operaciones que OBEX ofrece se encuentran la de *Connect, Disconnect, Put, Get, Abort, y SetPath*. Aquí, se deberá definir un tiempo adecuado (30s) [2] entre petición y respuesta pese a que la especificación no menciona ninguno.

3.5. FILE TRANSFER PROFILE (FTP)

El perfil FTP, que utiliza como base a GOEP para definir requerimientos de interoperabilidad para protocolos utilizados en aplicaciones de intercambio de objetos, menciona los protocolos y procedimientos necesarios para la ejecución de aplicaciones que emplean el modelo de transferencia de archivos. Mediante este perfil se puede lograr la transferencia de objetos de datos desde un dispositivo Bluetooth a otro, abrir y hacer búsqueda de archivos y carpetas que se encuentran en un dispositivo Bluetooth y navegar de acuerdo a la jerarquía de capetas que éste contenga, así también se puede manipular objetos disponibles, es decir se podría crear nuevas carpetas e inclusive borrar objetos. Aquí es importante mencionar que la especificación Bluetooth del perfil de transferencia de archivos no define requerimientos de Banda base, LMP,

L2CAP, o RFCOMM puesto que ellos se mencionan en GOEP [2]. La Figura 3.7 presenta la pila de protocolos para FTP.

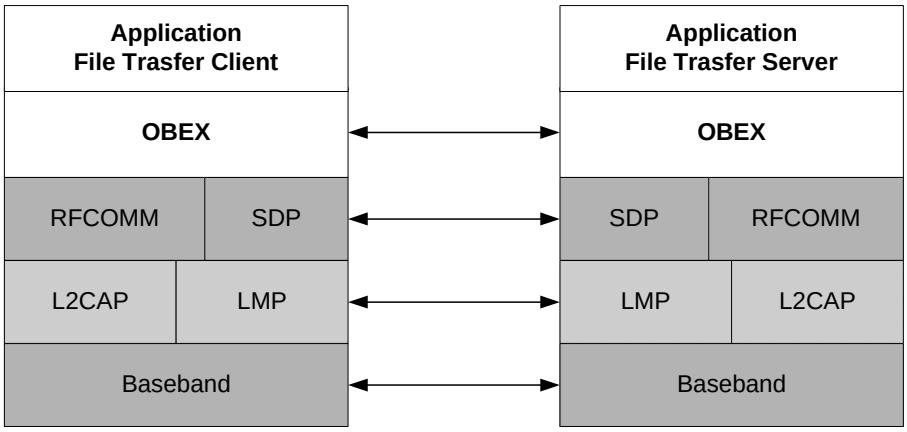

Figura. 3. 7. Modelo de protocolos para FTP

Como se observa en la Figura 3.7 existen dos partes la del cliente y del servidor. El cliente será encargado de interpretar o entender la lista de objetos que el servidor proveerá como respuesta a una petición de listado de objetos. El servidor por su parte deberá contar con una carpeta raíz que deberá ser pública y en la cual se podrá ofrecer los archivos o carpetas a compartir, así también el servidor puede establecer carpetas de solo lectura para restringir la creación o eliminación de archivos.

Para que el cliente pueda iniciar operaciones de transferencia de archivos con el servidor se requiere que el servidor se encuentre primeramente en modo *File Transfer*, haber seteado éste dispositivo en modo de *descubrimiento limitado*, y asegurar que el bit de transferencia de objetos sea seteado en CoD (*Class of Device*). Una vez realizados los pasos mencionados anteriormente se podrá ejecutar todas las funciones disponibles en las aplicaciones de trasferencia de archivos; es decir, *Push object, Pull Object, Delete Object, Create Folder, Select Server,* y *Navigate folders* [2].

El usuario cliente al escoger la función *Select Server* y seleccionar la aplicación para la transferencia de archivos se realizará el procedimiento de indagación o *inquiry* para conocer los dispositivos servidores que se encuentran disponibles. Como se dijo anteriormente, por parte del servidor se debe garantizar que el usuario del mismo fije al dispositivo en modo *File Transfer* para que éste pueda ser presentado al usuario cliente en una lista de servidores disponibles. En este punto, el cliente podrá escoger el servidor al cual se desea conectar y dependiendo de su necesidad de autenticación, de enlace y/o de OBEX, se pedirá al servidor ingresar una y/o dos contraseñas. Una vez escogido el servidor y fijada la conexión con el mismo, el cliente podrá observar la carpeta raíz del servidor [2].

Con respecto a la capa de aplicación, se destacan tres campos:

- *Folder Browsing*, cuya sesión empieza con la conexión del cliente al servidor y presentando el contenido de su carpeta raíz. En toda sesión de browsing el servidor es capaz de compartir distintas carpetas a diferentes usuarios, por lo que su acceso dependerá únicamente de la aceptación a peticiones GET *folder object* por parte del usuarios para ver, poner o quitar archivos en cualquier carpeta del servidor.

  La conexión cliente – servidor se la realiza mediante el comando OBEX CONNECT en el cual se especifica la cabecera de destino mediante la fijación del UUID del *Browsing Folder* que es F9EC7BC4-953C-11D2-984E-525400DC9E09 y es enviado de forma binaria (16 bytes) con el byte más significativo al principio.

  Para bajar contenido de la carpeta raíz del servidor se lo hace mediante GET, cuyo Tipo de Cabecera debe ser seteada de tipo MIME y la cabecera de CONNECT ID con el valor que es devuelto en la operación de conexión.

  Debido a que una sesión de *browsing* involucra desplegar contenidos, acceder a carpetas, y definir nuevas carpetas como

localidades actuales, el comando OBEX SETPATH es empleado para fijar una carpeta como actual.

- *Object Transfer*, para que un objeto sea enviado desde el cliente hacia el servidor se utiliza el comando OBEX PUT, mientras que si se envía desde el servidor hacia el cliente se utilizará OBEX GET.

  Para enviar una carpeta desde el cliente hacia el servidor se debe crear una nueva carpeta en la carpeta actual del servidor utilizando SETPATH y utilizar el comando PUT por parte del cliente para agregar cada archivo a la carpeta del servidor.

  Para recibir una carpeta desde el servidor hacia el cliente, se debe seleccionar la carpeta en la cual se va a recibir dicho objeto mediante SETPATH, obtener el contenido de la carpeta mediante el comando GET, y finalmente todos los archivos mediante el comando GET por cada archivo.

- *Object Manipulation*, el cliente puede crear y borrar archivos o carpetas que se encuentran en el lado del servidor. Para borrar un archivo se lo realiza mediante PUT con el nombre del archivo en la cabecera *Name* y dejando a la cabecera de cuerpo vacía. El mismo procedimiento se realiza para borrar carpetas vacías. Por otro lado, para crear una carpeta se lo hace mediante SETPATH con el nombre de la carpeta a crearse en la cabecera *Name*.

Para que todas las operaciones mencionadas anteriormente en los tres campos de aplicación puedan ejecutarse correctamente es indispensable y requerimiento obligado que tanto cliente como servidor soporten las siguientes operaciones de OBEX: *Connect, Disconnect, Put, Get, Abort,* y *SetPath*. Así también se deben soportar las siguientes cabeceras de OBEX: *Name, Type, Lenght, Target, Body, End of Body, Who, Connection ID, Authenticate Challenge,* y *Authenticate Response*.

3.6. INTERCOM PROFILE

El perfil de intercomunicador define los protocolos y procedimientos requeridos para poder establecer una comunicación de voz directamente entre dos dispositivos Bluetooth que pertenezcan a la misma red sin la necesidad de utilizar una red telefónica pública; esto permitirá, por ejemplo, la interconexión y comunicación de dos teléfonos en una misma oficina [14]. Esta utilidad normalmente es conocida por el usuario como "*walkie-talkie*".

En cuanto a la pila del perfil de intercomunicación se establece una entidad para el control de llamada (CC – *Call Control*), la cual se encargará de desconectar o conectar las rutas para el habla mediante el control de sincronización del habla (a); así también este perfil permite el envío de mensajes TCS en canales L2CAP orientados a la conexión (b) y mediante CC, que controla directamente LM, se controla el establecimiento o liberación de los enlaces SCO (c) [2]. La Figura 3.8 presenta la pila del perfil de intercomunicador.

En este perfil se requiere que el iniciador cuente con la dirección Bluetooth del dispositivo aceptador (utilizando descubrimiento de dispositivos descrito en GAP) para establecer una llamada de intercomunicador, pero no menciona obligatoriedad de seguridades (autenticación y habilitación de encriptación). El Anexo 4 presenta la señalización y procedimientos obligados para el establecimiento de llamadas de intercomunicación y su anulación (*clearing*).

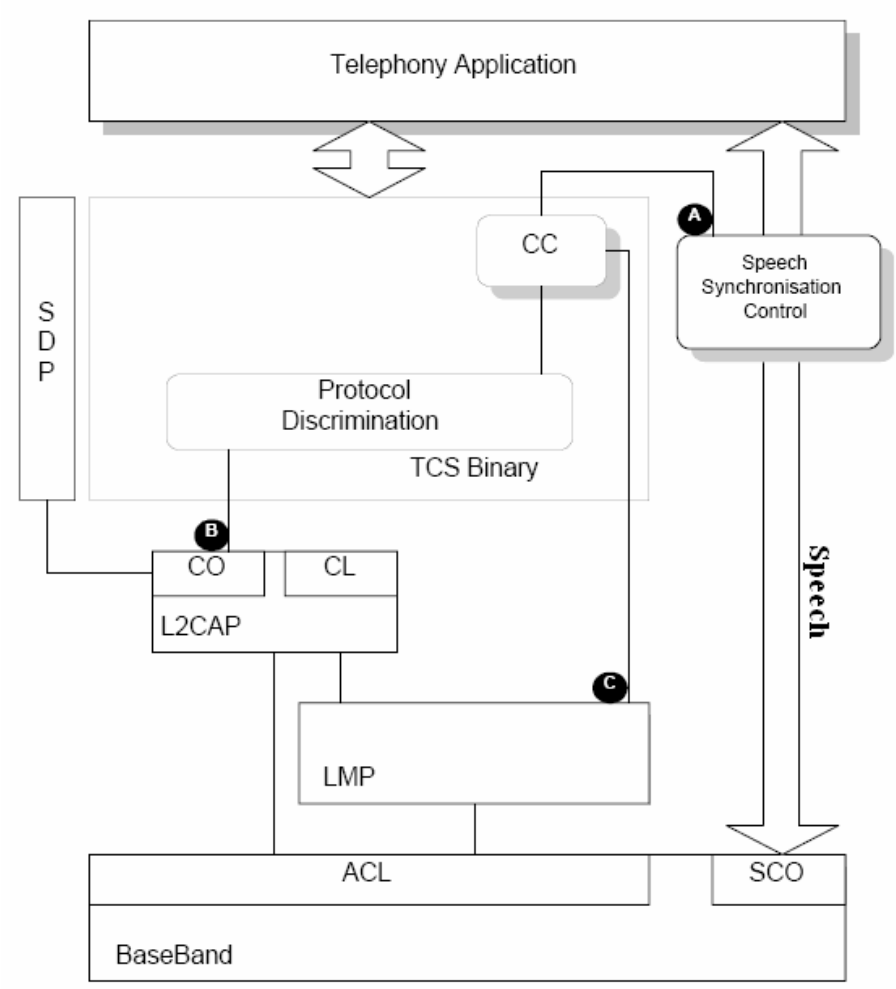

Figura. 3. 8. Pila del perfil Intercomunicador

El perfil de intercomunicador requiere, de manera obligada, de varios mensajes para el control de telefonía (TCS Bin) como son Alerting, *Connect, Connect Acknowledge, Disconnect, Release, Release Complete,* y *Setup*. De igual manera se necesita elementos de información como Tipo de mensaje (Message Type), Capacidad de soporte (*Bearer Capability*), Clase de llamada (*Call Class*), Causa (*Cause*), y Manejo de SCO (SCO handle) que se utilizarán en estos mensajes. Así también se especifica como obligatorio el elemento de información de Clase de llamada (*Call class*), el cual debe ser *Intercom call*.

Otras capacidades que deben ser soportadas con respecto a LMP son *Authentication, Pairing, Change link key, Change the current link key, Clock*

offset request, LMP version, Supported features, Name request, Detach, QoS, Link supervision, y *Connection establishment*. De igual manera, los requerimientos de interoperabilidad con SDP que deben ser soportados y se mencionan en el registro de servicios son *ServiceClassIDList, Protocol Descriptor List*, y en el campo de *BluetoothProfileDescriptorList* a Profile0 y a Param0, que indican los perfiles soportados y la versión del perfil respectivamente. En cuanto al principal requerimiento con respecto a LC para el control del enlace se debe soportar a CVSD (*Continuosly Variable Slope Delta Modulation* – Modulación Delta de Pendiente Continuamente Variable).

Entre las opciones de configuración disponibles en referencia a L2CAP para este perfil se menciona a MTU, cuya restricción de tamaño dependerá de L2CAP pero deberá tener un mínimo de 48 bytes; el tiempo de vacío (*Flush Timeout*), seteado por omisión en 0xFFFF; y QoS, la cual debido a que L2CAP soporta servicios del “mayor esfuerzo” es opcional y en caso de emplearse se requerirá mencionar los parámetros de variación de retardo, ancho de banda pico y latencia.

Por otra parte, los requerimientos que deben ser soportador en GOEP son el *Non-discoverable* mode y *General discoverable mode* en *Discoverability Modes*, y *Connectable mode* en *Connectability modes*.

CAPÍTULO 4

PRUEBAS DEL EQUIPO DE BLUETOOTH BT-1000

El equipo de desarrollo BT-1000 disponible en el laboratorio de SAT está compuesto por dos dispositivos BlueSEM II de Samsung Electronics que pertenecen a la clase 2 y satisfacen a la especificación Bluetooth versión 1.1. Cada dispositivo BT-1000 cuenta con diferentes etapas dedicadas a interfaces que permitirán establecer la comunicación entre el PC, que correrá la aplicación de voz o datos, y el dispositivo BT-1000. Así mismo se cuenta con una etapa dedicada para el CODEC (PCM), el cual convertirá señales analógicas de voz a datos digitales y por lo tanto será utilizado en sesiones de comunicación de voz. La Figura 4.1 presenta al dispositivo BT-1000 y sus partes involucradas se mencionan a continuación.

1) UART Entrada/Salida
2) SPI Entrada/Salida
3) Parallel Entrada/Salida
4) USB Entrada/Salida
5) Módulo Bluetooth
6) PIO
7) Codec
8) Fuente de poder

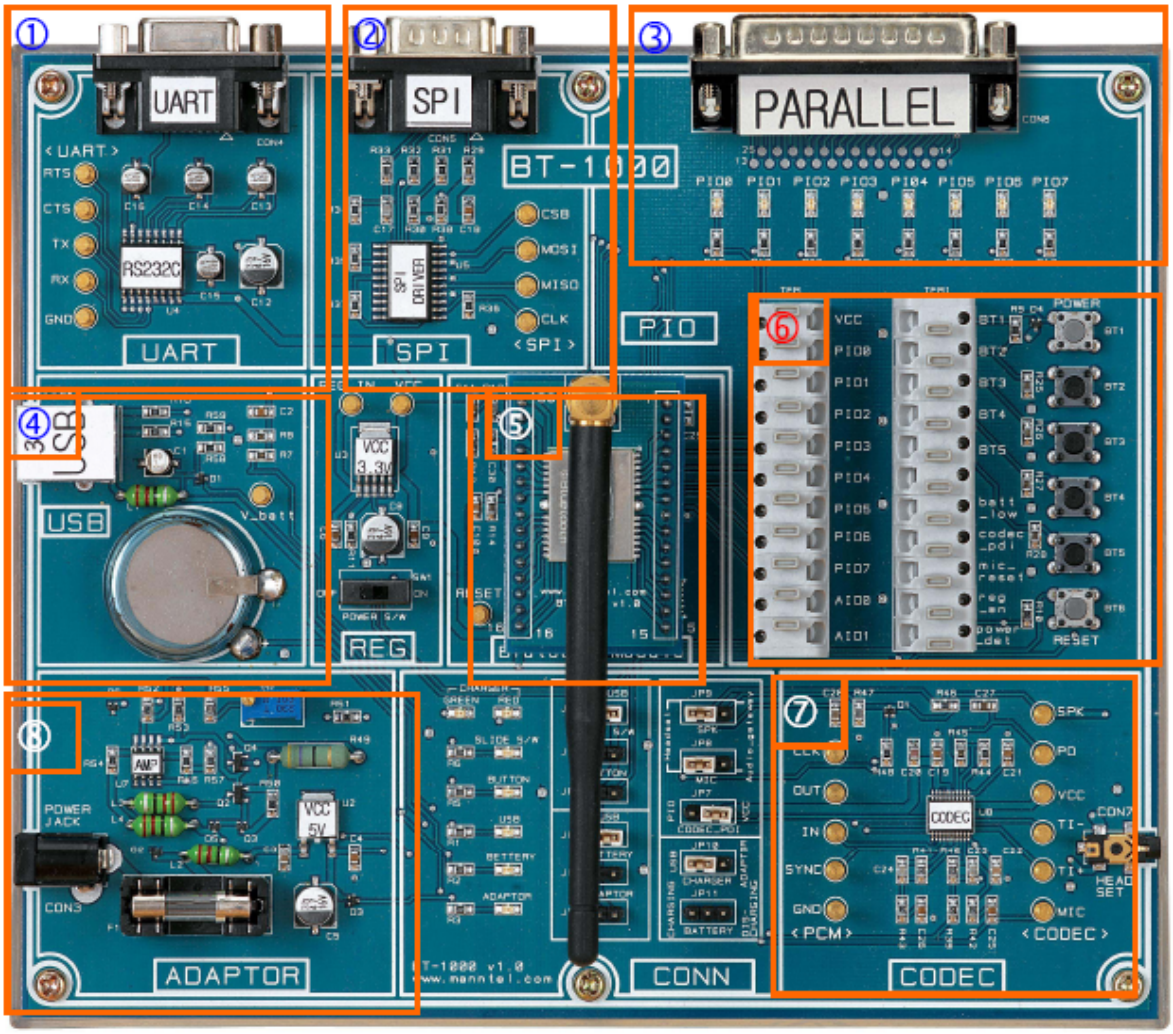

Figura. 4. 1. Partes del dispositivo BT-1000 [3]

El software del Kit de desarrollo BT-1000 provee una carpeta de sencilla instalación que brindará la posibilidad de establecer sesiones FTP para la transferencia de archivos, sesiones de Chat, además de la comunicación de voz en tiempo real. Para ejecutar dichas aplicaciones es necesario contar con dos PC, los dos dispositivos BT-1000 con sus módulos Bluetooth, y cables USB que además de ser interfaces entre PC y dispositivos servirán de alimentación para los mismos [3]. Las pruebas que se mencionarán en este capítulo involucran la aplicación de voz y de FTP. La configuración del sistema se presenta en la Figura 4.2.

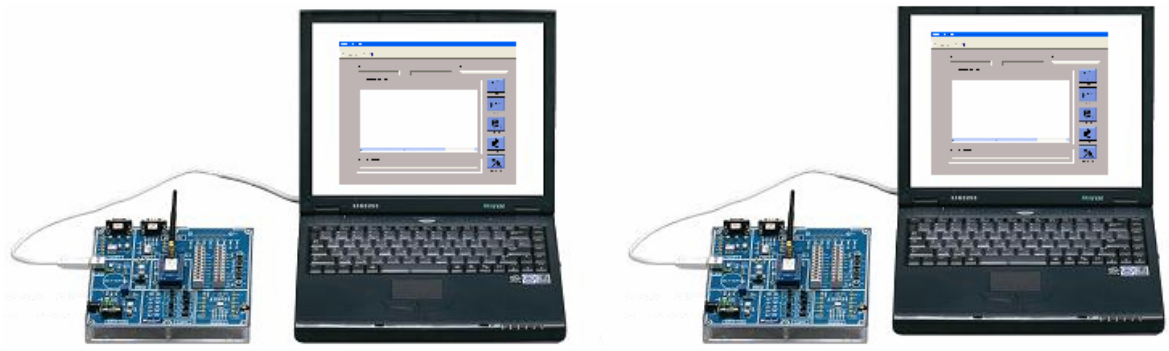

Figura. 4. 2. Configuración del Sistema [3]

4.1. TRANSMISIÓN DE DATOS

La prueba de transmisión de datos se la realizó con dos dispositivos BT-1000 y con ayuda del software disponible en su kit de desarrollo y específicamente con la aplicación de transferencia de archivos (FTP) que permite establecer la sesión de comunicación entre dos puntos distantes mediante un enlace Bluetooth para la transferencia de archivos.

En la ejecución de la aplicación FTP se utiliza un enlace ACL para establecer la comunicación entre los dos dispositivos BT-1000 y para la transmisión de cualquier tipo de archivo se utilizan paquetes de datos tipo DH1. Debido a que los paquetes DH1 permiten retransmisión de los mismos en caso de existir errores y conociendo que un enlace se degrada con el aumento de la distancia entre dos puntos, es decir existirá mayor número de retransmisiones cada vez que se incrementa la distancia, la realización de la prueba tiene como objetivo determinar el número de retransmisiones que se producen debido a la variabilidad de la distancia. De esta manera, conociendo el tamaño del archivo a enviarse, el tamaño establecido para paquetes DH1, y el tiempo que demora llevar un archivo de un punto a otro, el cálculo de paquetes retransmitidos será posible.

Para mayor precisión de la prueba se enviaron ocho veces el mismo archivo desde un computador local hacia otro remoto con incrementos de distancia entre dichos computadores de cinco metros tanto con línea de vista y sin línea de vista. En la prueba con línea de vista se llegó a una distancia máxima de 80 metros, es

decir a los 85 metros fue imposible establecer un enlace ACL puesto que ningún dispositivo BT-1000 podía ser reconocido mediante el procedimiento de *Inquiry*. Por otro lado, en la prueba sin línea de vista que se realizó en el Bloque D de la ESPE con pocas personas circulando en la cuarta planta, la presencia de paredes de 40cm de espesor, y numerosas bancas de madera a una temperatura promedio de 15 °C aproximadamente se logró un alcance máximo de 20 metros, es decir a los 25m de separación ningún dispositivo BT-1000 podía reconocerse mediante el procedimiento de *Inquiry*.

Para establecer la sesión FTP, al igual que una comunicación de voz, entre dispositivos BT-1000 como se explicó en capítulos anteriores es necesario una serie de pasos, como el procedimiento de Inquiry, Paging, el establecimiento de la conexión con capas superiores como sucede con L2CAP, SDP, el servicio de OBEX y finalmente la comunicación o intercambio de datos entre los perfiles FTP de los dos dispositivos Bluetooth. Cada uno de estos pasos se menciona a continuación.

4.1.1. Proceso de inicialización. Procedimiento de inicialización y señalización en la parte del esclavo/maestro para poder escuchar peticiones de Inquiry y Paging. La Figura 4.3 presenta los pasos necesarios.

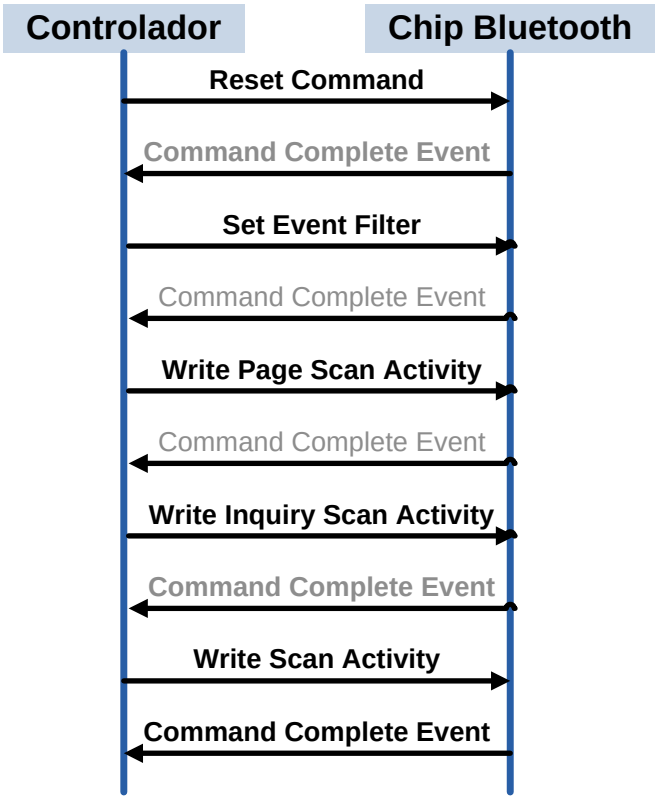

Figura. 4. 3. Inicialización y Señalización para la configuración de la comunicación con un esclavo

4.1.2. Conexión. Una vez establecida la inicialización en cada dispositivo Bluetooth involucrado se podrá realizar el proceso de conexión. La Figura 4.4 presenta este procedimiento entre maestro y esclavo.

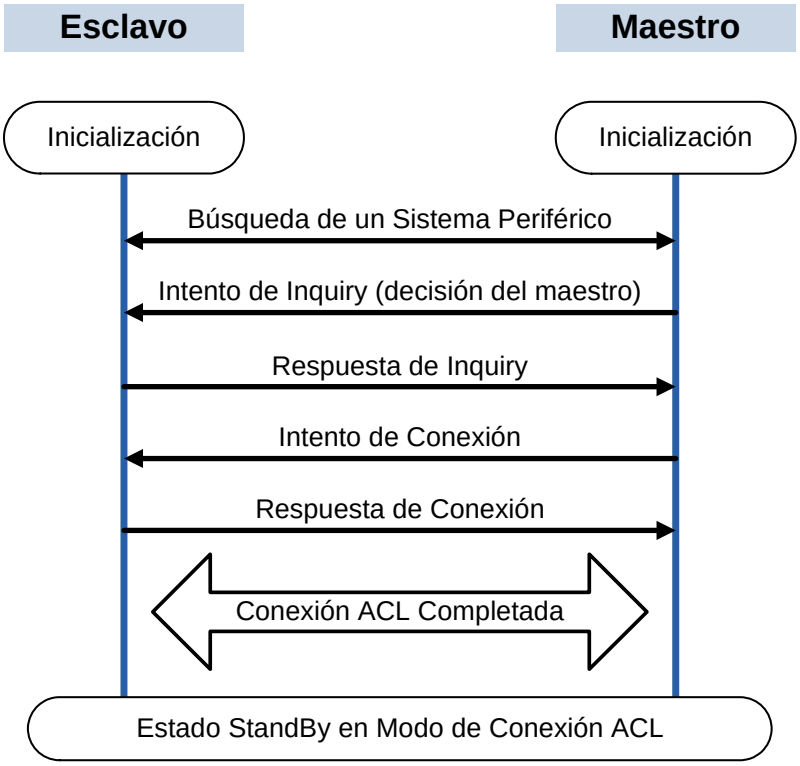

Figura. 4. 4. Proceso de conexión entre Maestro y Esclavo [3]

Inquiry es el proceso en el cual el dispositivo Bluetooth desea descubrir otros dispositivos Bluetooth que se encuentran cercanos, en este proceso únicamente los dispositivos que son fijados en modo "*discoverable*" podrán responder a las peticiones que el dispositivo iniciador está solicitando. En este procedimiento de *inquiry* no se utiliza ninguna de las capas superiores sobre el canal físico. La Figura 4.3 presenta el procedimiento de inicialización y señalización en la parte del esclavo para poder escuchar peticiones de Inquiry y Paging.

Paging (emparejamiento) es el procedimiento de conexión y consiste en que un dispositivo iniciador realice el *page* mientras el segundo dispositivo se encuentre en modo "*connectable*". Este procedimiento se lo realiza de tal manera que únicamente un solo dispositivo Bluetooth podrá responder a la petición de conexión. Aquí, los dispositivos en modo "*connectable*" utilizan un canal físico especial para poder escuchar a todas las peticiones de conexión que se realicen. Este procedimiento de *paging*

se presenta en la Figura 4.4 y se lo detalla en la Figura 4.5 que muestra los pasos para la conexión entre los dos dispositivos Bluetooth.

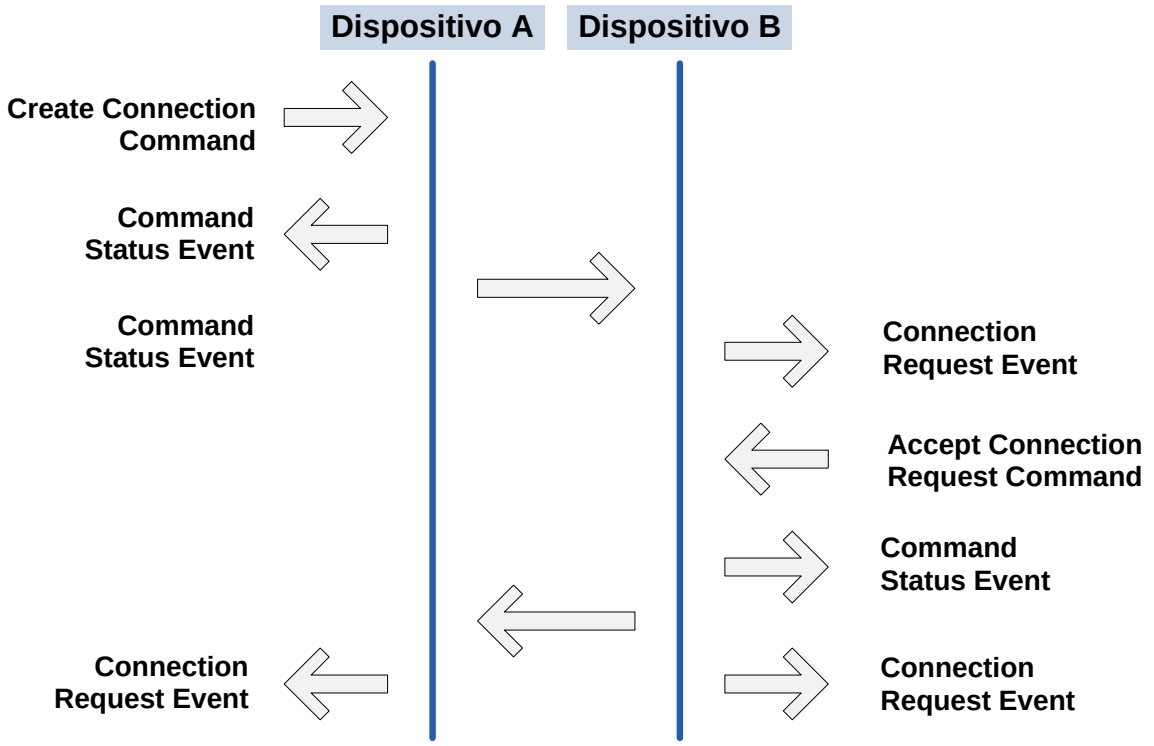

Figura. 4. 5. Pasos para la conexión de dos dispositivo Bluetooth [4]

Una vez terminado este proceso, una conexión ACL entre los dos dispositivos Bluetooth involucrados es establecida y por lo tanto se podrá conocer los servicios que son soportados por cualquier dispositivo. La Figura 4.6 presenta el proceso necesario para solicitar servicios de Esclavo a Maestro.

4.1.3. Búsqueda de servicio entre dispositivos Bluetooth. Una vez establecida la conexión con la capa superior L2CAP y su confirmación, se realiza el procedimiento para encontrar los servicios que cada dispositivo ofrece. Aquí, la capa SDP recibe la petición de un servicio que es comparado con la base de datos que posee, si cuenta con el servicio solicitado se confirmará de su existencia, caso contrario no sucederá nada

hasta que se apague el dispositivo. La Figura 4.6 presenta este proceso de búsqueda.

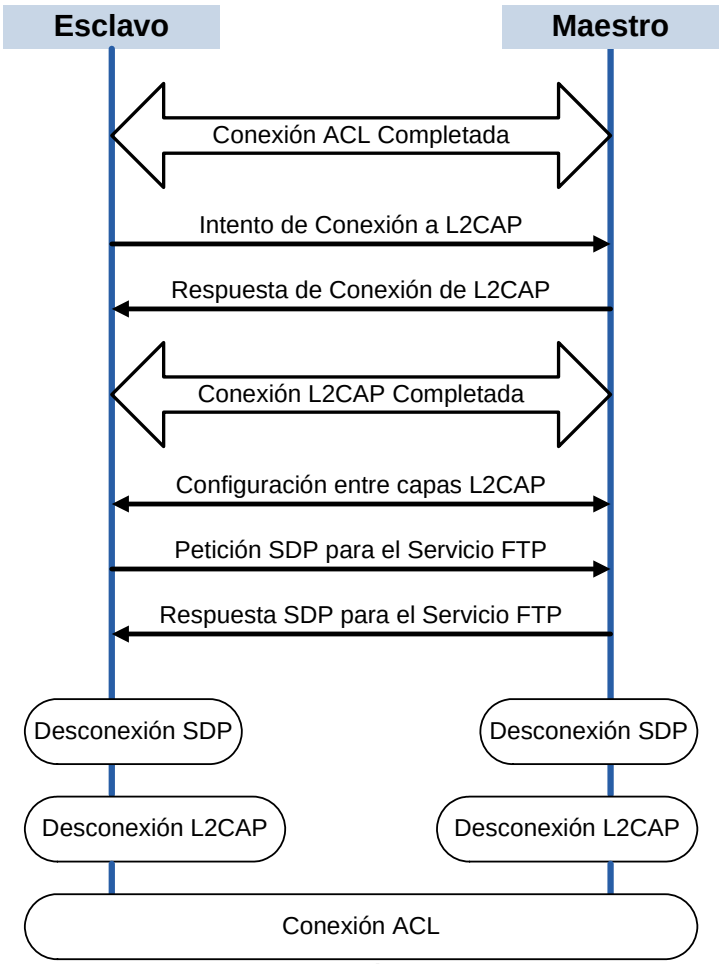

Figura. 4. 6. Proceso de búsqueda de servicios [3]

4.1.4. Procedimiento para la Transferencia de Datos. Para realizar la transferencia de datos vía FTP entre dos dispositivos Bluetooth es necesario establecer conexión con las capas L2CAP y RFCOMM, posterior a estos pasos se ajusta la configuración entre perfiles FTP y en este punto el servidor podrá enviar los datos requeridos por el cliente. La Figura 4.7 presenta estos pasos.

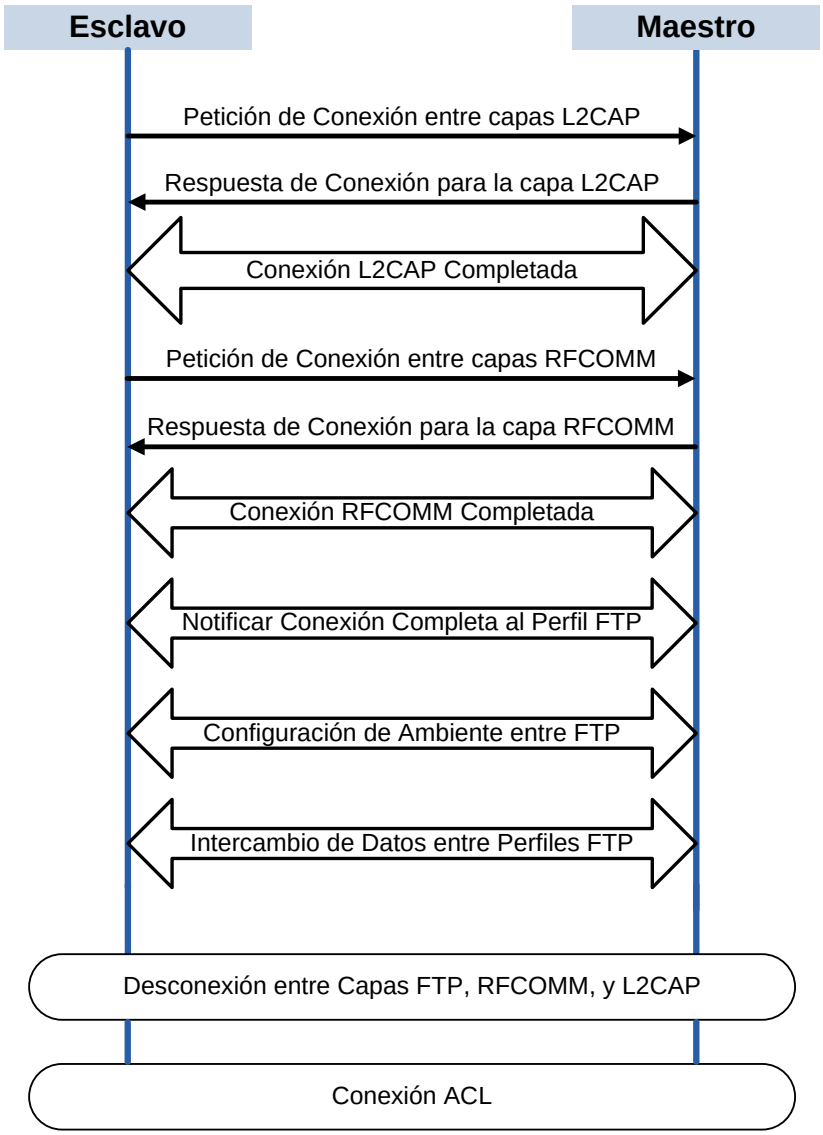

Figura. 4. 7. Procedimiento para la transferencia de Datos [3]

4.1.5. Procedimiento de Desconexión. Para realizar la desconexión tanto para conexiones ACL como SCO el controlador envía un comando de desconexión al módulo Bluetooth. Aquí, en caso de haberse establecido una conexión SCO sobre la conexión ACL, SCO deberá cerrarse primero para posteriormente desconectar la conexión ACL. La Figura 4.8 presenta este proceso.

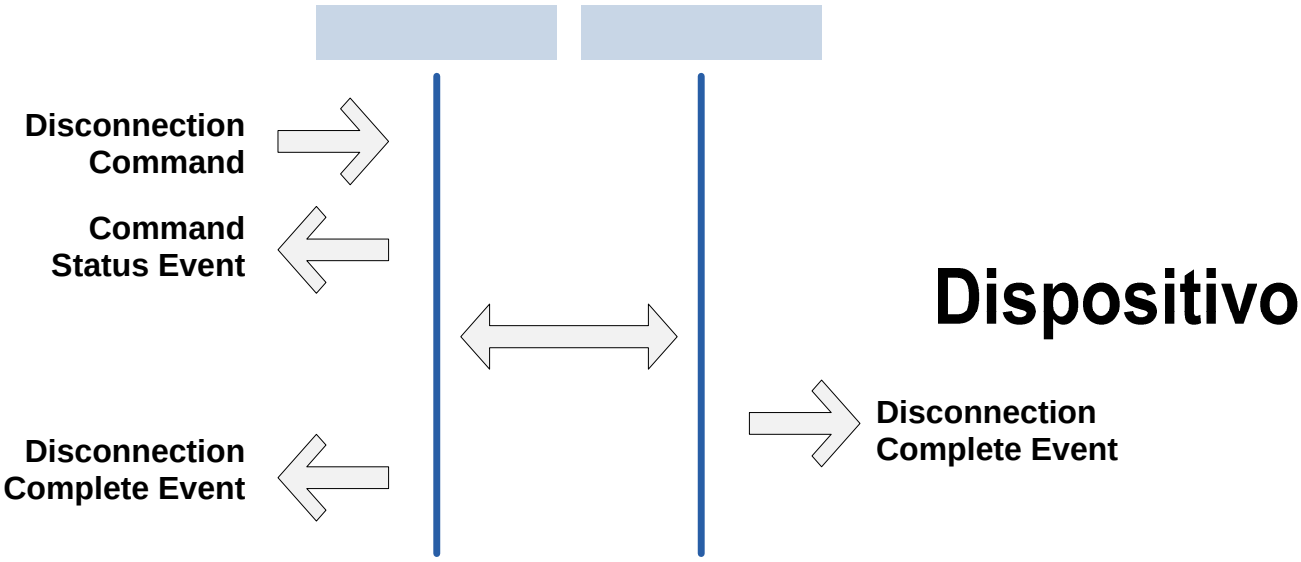

Figura. 4. 8. Procedimiento de Desconexión [4]

Una vez realizados los pasos mencionados anteriormente (4.1.1 – 4.1.5) y utilizando el archivo AntiKnight.rar para la transferencia vía FTP con Bluetooth de un punto a otro y cuyo tamaño es de 26.095 Bytes, se obtuvieron los resultados presentados en la Tabla 4.1 y los tiempos promedios según las ocho muestras tomadas cada cinco metros se los grafica en la Figura 4.10. Con estos datos y conociendo que en la transferencia del archivo se utilizan paquetes DH1, cuyo tamaño se conoce y se presenta en la Figura 4.9, y además sabiendo el tiempo promedio que demora en transferirse el archivo completo en cada punto, se considera el tiempo promedio ocurrido a 20 cm de distancia como base y referencia de cero retransmisiones para los cálculos de paquetes retransmitidos y mencionados en tablas siguientes. Las pruebas realizadas al dispositivo BT-1000 se analizan en las secciones 4.1.6 y 4.1.7, las cuales corresponden a pruebas con línea de vista y sin línea de vista respectivamente.

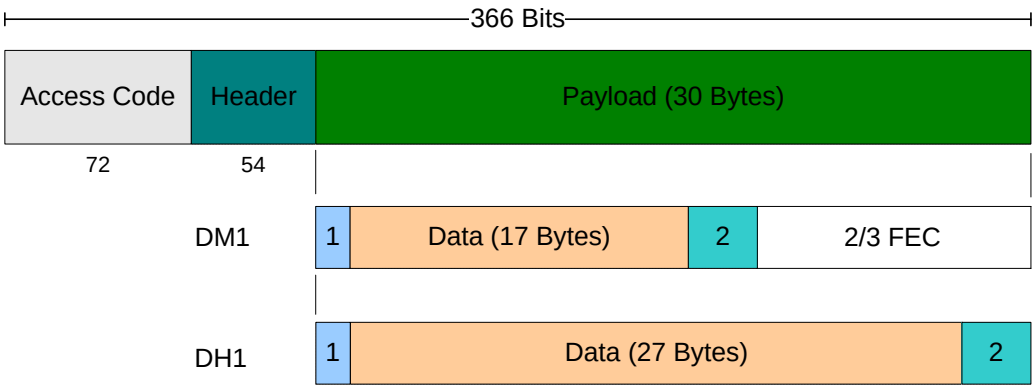

Figura. 4. 9. Paquetes DM1 Y DH1

4.1.6. Prueba con Línea de Vista

Distancia	[m]	0,2	5	10	15	20	25	30	35	40
Tiempo [s]	t1	19,22	18,57	23,39	22,34	22,63	19,30	24,20	20,37	21,25
	t2	19,70	20,36	22,58	23,92	21,38	19,48	24,79	20,77	22,08
	t3	19,31	18,29	22,77	21,27	21,25	20,69	25,57	19,32	21,65
	t4	19,88	19,73	22,96	21,98	22,20	21,64	22,89	20,08	22,17
	t5	19,62	21,21	22,66	20,81	21,73	21,78	22,37	20,75	22,83
	t6	19,35	21,17	22,18	22,70	21,59	22,10	22,50	20,70	22,62
	t7	19,47	21,02	23,11	23,38	19,32	19,40	22,79	21,65	22,72
	t8	19,27	19,58	24,45	22,12	20,10	19,55	22,95	22,09	22,80
	t_Prom	**19,48**	**19,99**	**23,01**	**22,32**	**21,28**	**20,49**	**23,51**	**20,72**	**22,27**
	t_Prom_Tx	**9,74**	**10,00**	**11,51**	**11,16**	**10,64**	**10,25**	**11,75**	**10,36**	**11,13**

Distancia	[m]	45	50	55	60	65	70	75	80
Tiempo [s]	t1	25,24	22,49	22,80	22,45	27,44	26,67	21,92	33,07
	t2	30,17	22,32	22,00	22,26	28,90	27,05	24,06	31,22
	t3	30,54	22,48	22,30	22,08	29,43	24,12	23,65	33,18
	t4	30,29	22,72	22,10	19,43	29,64	23,52	23,75	31,82
	t5	30,00	22,78	22,60	19,91	28,15	22,94	23,45	31,95
	t6	28,84	22,85	22,67	20,15	28,43	22,85	24,86	31,22
	t7	33,49	22,65	22,70	21,05	28,18	22,63	26,11	32,43
	t8	32,33	22,88	22,65	20,76	28,23	23,49	26,28	31,26
	t_Prom	**30,11**	**22,65**	**22,48**	**21,01**	**28,55**	**24,16**	**24,26**	**32,02**
	t_Prom_Tx	**15,06**	**11,32**	**11,24**	**10,51**	**14,28**	**12,08**	**12,13**	**16,01**

Tabla. 4. 1. Valores del tiempo de transferencia según la variación de la distancia con línea de vista

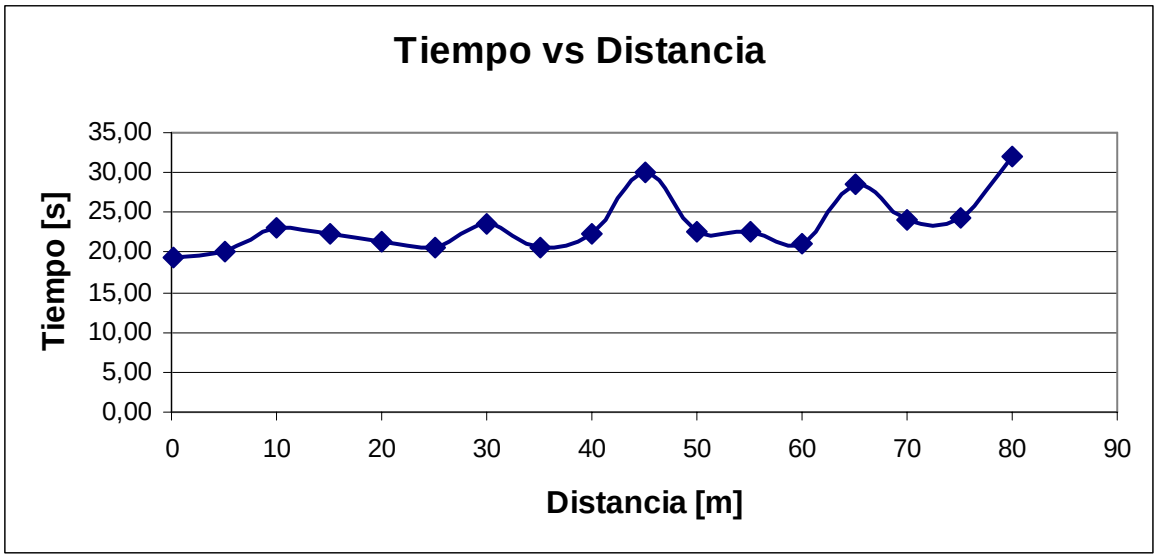

Figura. 4. 10. Tiempos de transferencia según la variación de la distancia con línea de vista

Obteniendo estos tiempos (t_Prom: promedio de tiempo de las 8 muestras, t_Prom_Tx: t_Prom/2) y debido a que los paquetes DH1 ocupan un slot de tiempo y el siguiente slot de tiempo es dedicado para uso específico del esclavo podemos determinar el tiempo de transmisión (Tiempo promedio / 2). La Figura 4.11 presenta dichos tiempos.

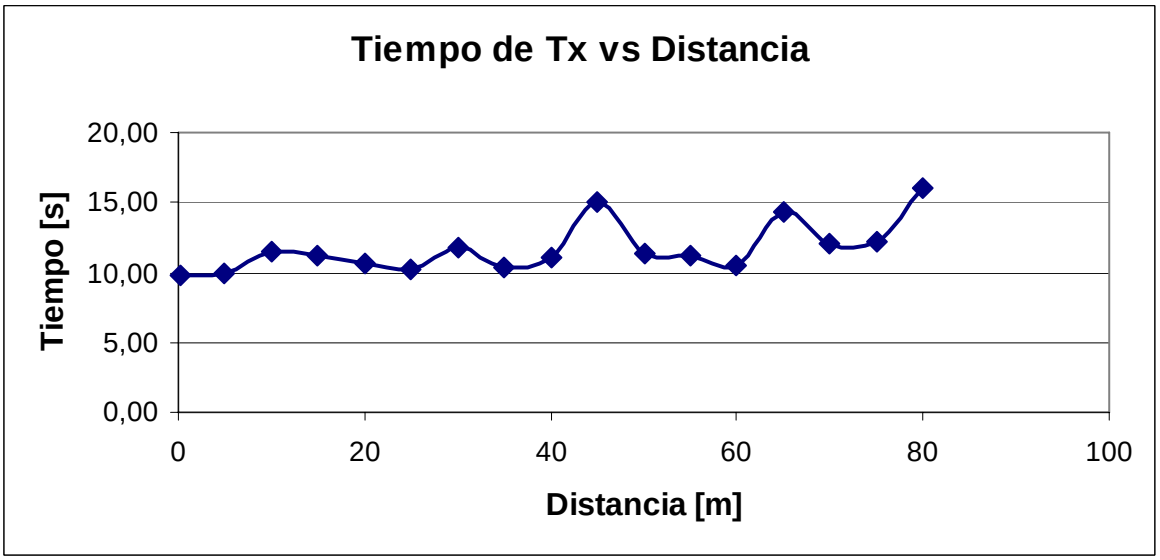

Figura. 4. 11. Tiempo de transmisión según la variación de la distancia con línea de vista

Si dividimos por partes al paquete DH1 podremos conocer el tamaño en bits destinado para datos del usuario, de igual manera si multiplicamos el tamaño del archivo transferido (en Bytes) por 8 obtendremos su tamaño en Bits y por lo tanto si dividimos este valor para los bits dedicados para datos del usuario en cada paquete se conocerá el número de paquetes que deberán ser transmitidos. Este paso se lo presenta en la Tabla 4.2 y 4.3.

Tabla. 4. 2. División por secciones del paquete DH1

Sección	Bits
Access Code	72
Header	54
Payload	8 (header)
	216 (data)
	16 (CRC)
Total	366
Paquete sin Datos	366 – 216 = 150

Tabla. 4. 3. Número de paquetes DH1 a transmitirse según el tamaño del archivo a enviarse

AntiKnight.rar	26095	Bytes
	208760	Bits
Paquetes a Tx	$\frac{208760}{216} = 967$	Paquetes DH1

De esta manera, si multiplicamos el número de paquetes por el tamaño de cada paquete DH1 (367 bits) se podrá conocer el número de bits totales que serán enviados desde el transmisor hacia al receptor en el proceso de transferencia del archivo AntiKnight.rar. Así, se podrá calcular el *Throughput*[3] que ocurre con el incremento de la distancia, aunque el *Throughput* real será aquel calculado a la distancia 20cm debido a que se considera ideal y será tomado como base inicial para conocer el

[3] Throughput se refiere al volumen total de información que pasa a través de un sistema en un tiempo determinado, generalmente se lo mide en bits por segundo (bps)

incremento de bits ocasionado por las retransmisiones que ocurrirán según la variación de la distancia, la Tabla 4.4 presenta estos cálculos y la variación del *Throughput* se la grafica en la Figura 4.12.

Número de bits totales a transmitirse = 967 [paquetes] * 366 [bits]

Número de bits totales a transmitirse =354 [Kbits]

Tabla. 4. 4. Cálculo del Throughtput y su variación con respecto al valor base con línea de vista

Distancia [m]	Tiempo Tx [s]	Throughput [Kbps]	Diferencia con respecto a la Base [Kbps]
0,2	9,74	36	0,00
5	10,00	35	1
10	11,51	31	6
15	11,16	32	5
20	10,64	33	3
25	10,25	35	2
30	11,76	30	6
35	10,36	34	2
40	11,14	32	5
45	15,06	24	13
50	11,33	31	5
55	11,24	31	5
60	10,51	34	3
65	14,28	35	12
70	12,08	29	7
75	12,13	29	7
80	16,01	22	14

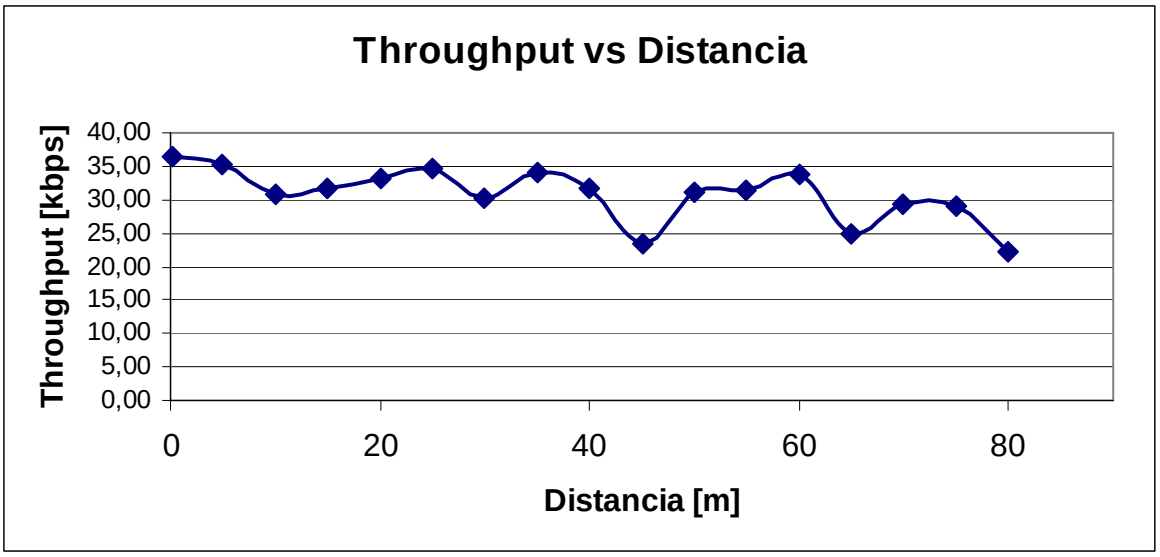

Figura. 4. 12. Variación del Throughput según la distancia con línea de vista

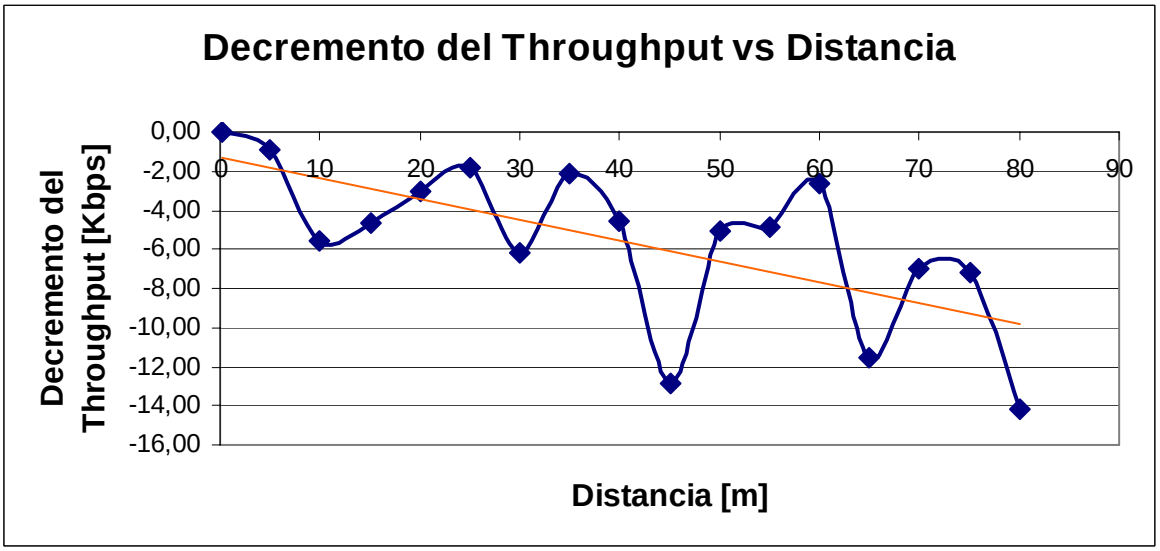

Figura. 4. 13. Decremento del Throughput según la distancia con línea de vista

Tomando en consideración que no existen retransmisiones a 20cm de distancia entre los dos dispositivos BT-1000 se procede a obtener la diferencia de tiempos según la variación de la distancia y por lo tanto sabiendo el número de bits que se transmiten por segundo (según el *throughput* ideal) se podrá conocer el número de bits y por lo tanto de

paquetes que deben ser retransmitidos en ese lapso tiempo. La Tabla 4.5 presenta los cálculos mencionados anteriormente.

$$\begin{array}{ll} 1[seg] & 36317[Bits] \\ (Tiempo_i - Tiempo\,base) & Bits\,reTx \end{array}$$

$$Bits\,reTx = \frac{a[s]\cdot 36317[bits]}{1[s]}$$

$$Paquetes\ reTx = \frac{a[bits]}{366[bits]}$$

$$Paquetes\,Totales = Número\ paquetes\,inicial + paquetes\,reTx$$

Tabla. 4. 5. Cálculos del número de bits, paquetes retransmitidos y paquetes totales retransmitidos con línea de vista

Distancia [m]	Diferencia de Tiempo [s]	Bits reTx	Paquetes reTx	Paquetes Totales Tx
0,20	0,00	0,00	0,00	966,48
5,00	0,25	9261	26	992
10,00	1,77	64100	175	1142
15,00	1,42	51571	141	1108
20,00	0,90	32686	90	1056
25,00	0,50	18340	51	1017
30,00	2,02	73180	200	1167
35,00	0,62	22517	62	1028
40,00	1,40	50663	139	1105
45,00	5,32	193027	528	1494
50,00	1,59	57563	158	1124
55,00	1,50	54476	149	1116
60,00	0,77	27783	76	1043
65,00	4,54	164700	450	1417
70,00	2,34	84983	233	1199
75,00	2,39	86799	238	1204
80,00	6,27	227711	623	1589

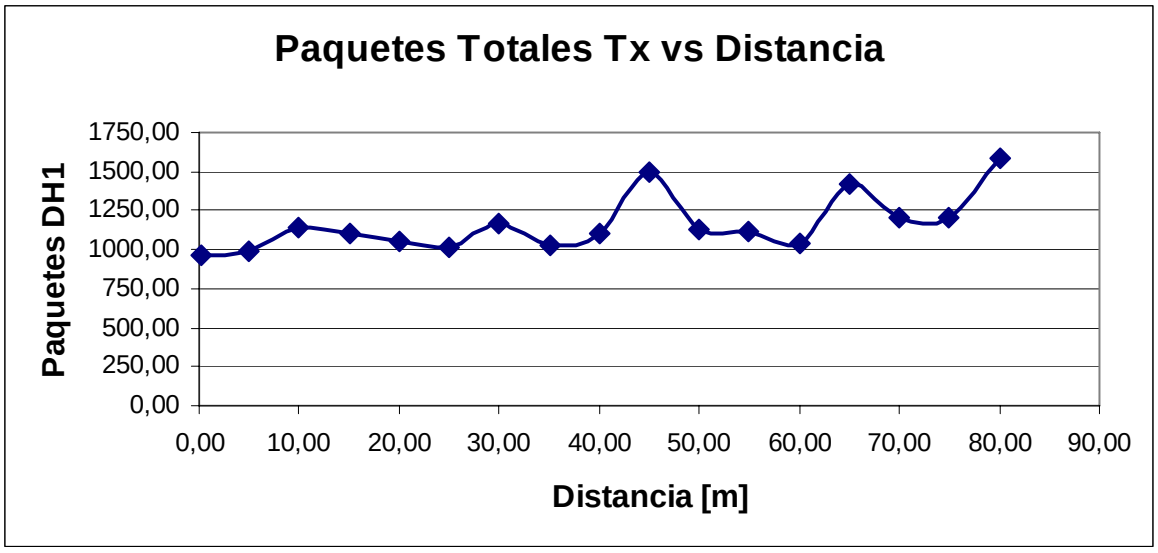

Figura. 4. 14. Número de paquetes totales transmitidos según la distancia con línea de vista

Finalmente, si se conoce el número de paquetes totales base (cero retransmisiones) y el número de paquetes totales según la variabilidad de la distancia, se puede obtener el error porcentual que se produce al incrementar la distancia. La Tabla 4.6 presenta los cálculos de este error porcentual y su gráfica se observa en la Figura 4.15.

$$Paquetes\,Totales\,Base = 966$$

$$Error\,\% = \frac{Paquetes\,Totales - Paquetes\,Totales\,Base}{Paquetes\,Totales\,Base} * 100$$

Tabla. 4. 6. Cálculo del porcentaje de error según la distancia

Distancia [m]	Error %
0,20	0,00
5,00	2,62
10,00	18,12
15,00	14,58
20,00	9,24
25,00	5,18
30,00	20,69
35,00	6,37
40,00	14,32
45,00	54,57
50,00	16,27
55,00	15,40
60,00	7,85
65,00	46,56
70,00	24,02
75,00	24,54
80,00	64,37

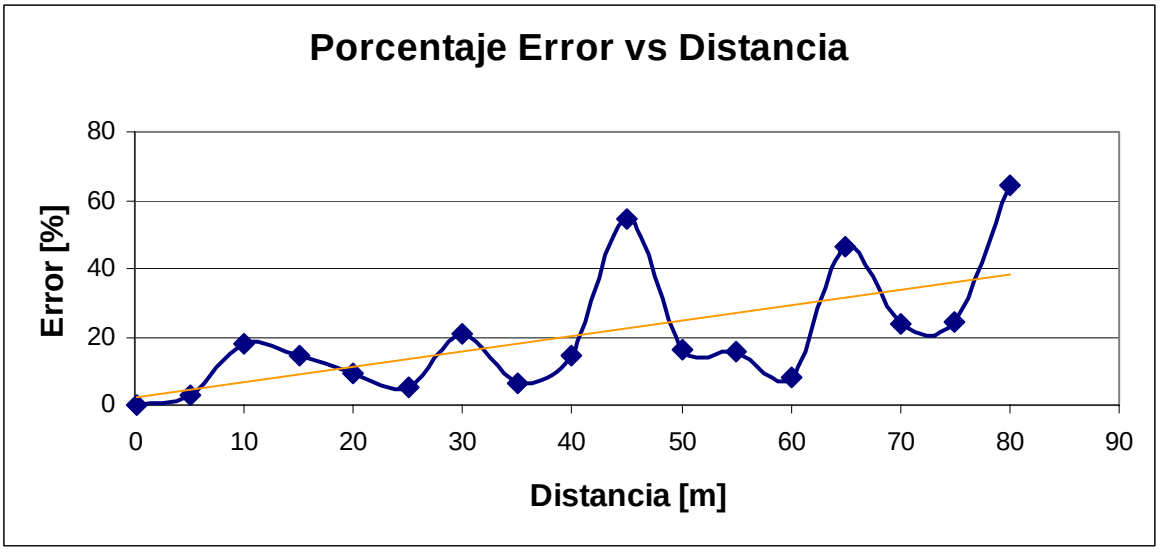

Figura. 4. 15. Porcentaje de error según la distancia con línea de vista

Asimismo, podemos calcular el *Goodput*[4] ideal que ocurre en esta transferencia.

$$Goodput = \frac{Tamaño\ Archivo\left[Bytes\right] \cdot 8\left[\frac{bits}{Byte}\right]}{\frac{19,48}{2}\left[s\right]}$$

$$Goodput = \frac{26096\left[bits\right]}{9,74\left[s\right]}$$

$$Goodput = 21\left[Kbps\right]$$

Prueba sin Línea de Vista. Siguiendo el mismo procedimiento de cálculos mencionado en el paso 4.1.6 se obtienen las siguientes tablas y gráficas según los datos tomados en las pruebas sin línea de vista.

Tabla. 4. 7. Valores del tiempo de transferencia según la variación de la distancia sin línea de vista

Distancia	**[m]**	**0,2**	**5**	**10**	**15**	**20**
Tiempo [s]	**t1**	19,42	20,10	22,98	22,2	24,95
	t2	19,68	20,16	20,74	22,32	29,13
	t3	19,71	21,33	20,42	22,00	26,76
	t4	19,55	18,93	21,08	22,14	23,15
	t5	19,87	19,21	21,31	21,51	26,50
	t6	19,52	18,59	20,86	22,08	25,10
	t7	19,64	19,63	21,37	23,68	23,75
	t8	19,75	19,74	21,51	22,59	26,57
	t_Prom	**19,64**	**19,71**	**21,28**	**22,33**	**25,74**
	T_Prom_Tx	**9,82**	**9,86**	**10,64**	**11,17**	**12,87**

t_Prom: tiempo promedio de las 8 muestras

T_Prom_Tx: tiempo de transmisión promedio (t_Prom/2)

[4] Se refiere a la tasa de transmisión perteneciente a los datos del usuario únicamente, generalmente se lo mide en bits por segundo (bps)

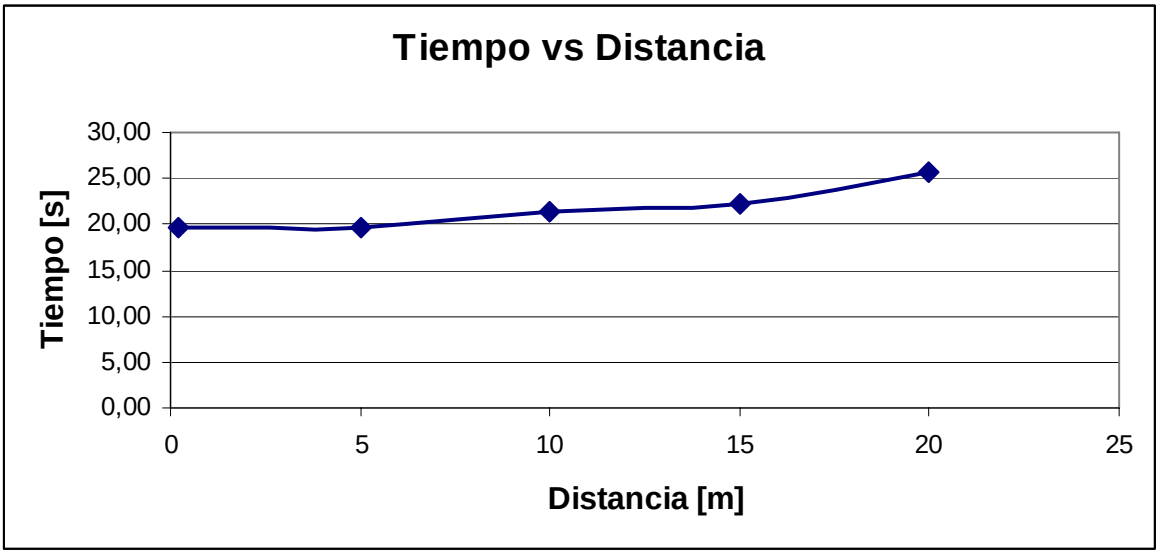

Figura. 4. 16. Tiempos de transferencia según la variación de la distancia sin línea de vista

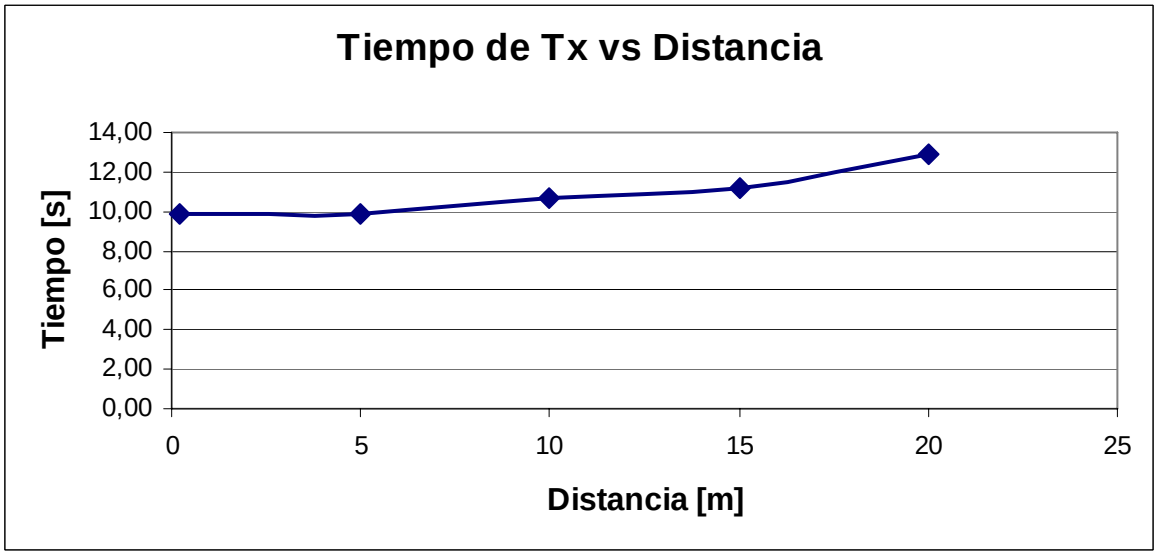

Figura. 4. 17. Tiempos de transmisión según la variación de la distancia sin línea de vista

Tabla. 4. 8. Cálculo del Throughtput y su variación con respecto al valor base (LOS) sin línea de vista

Distancia [m]	Tiempo [s]	Throughput [Kbps]	Diferencia con respecto a la Base [Kbps]
(LOS) 0,2	9,74	36	0
(NLOS) 0,2	9,82	36	0
5	9,86	36	0
10	10,64	33	3
15	11,17	32	5
20	12,87	27	9

(LOS): Tiempo promedio de transmisión obtenido con línea de vista entre nodos

(NLOS): Tiempo promedio de transmisión obtenido sin línea de vista entre nodos

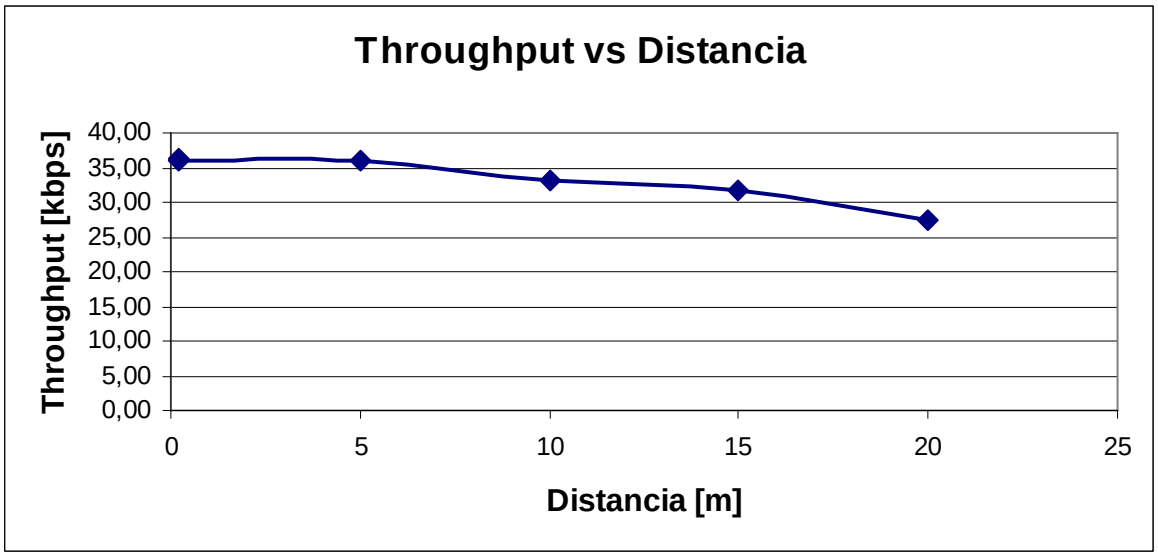

Figura. 4. 18. Variación del Throughput según la distancia sin línea de vista

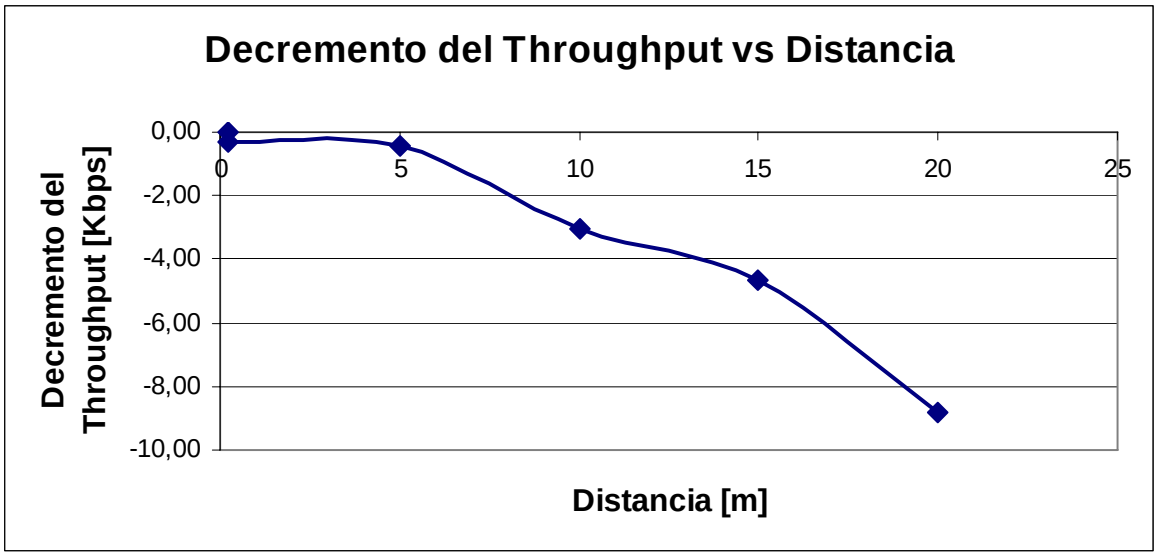

Figura. 4. 19. Decremento del Throughput según la distancia sin línea de vista

Tabla. 4. 9. Cálculos del número de bits, paquetes retransmitidos y paquetes totales retransmitidos sin línea de vista

Distancia [m]	Diferencia de Tiempo [s]	Bits reTx	Paquetes reTx	Paquetes Totales Tx
(LOS) 0,20	0,00	0,00	0,00	967
(NLOS) 0,20	0,08	2905	8	975
5,00	0,12	4177	12	978
10,00	0,90	32686	90	1056
15,00	1,43	51752	142	1108
20,00	3,13	113674	311	1278

Tabla. 4. 10. Cálculo del porcentaje de error según la distancia sin línea de vista

Distancia [m]	Error %
0,20	0,00
0,20	0,82
5,00	1,18
10,00	9,24
15,00	14,63
20,00	32,14

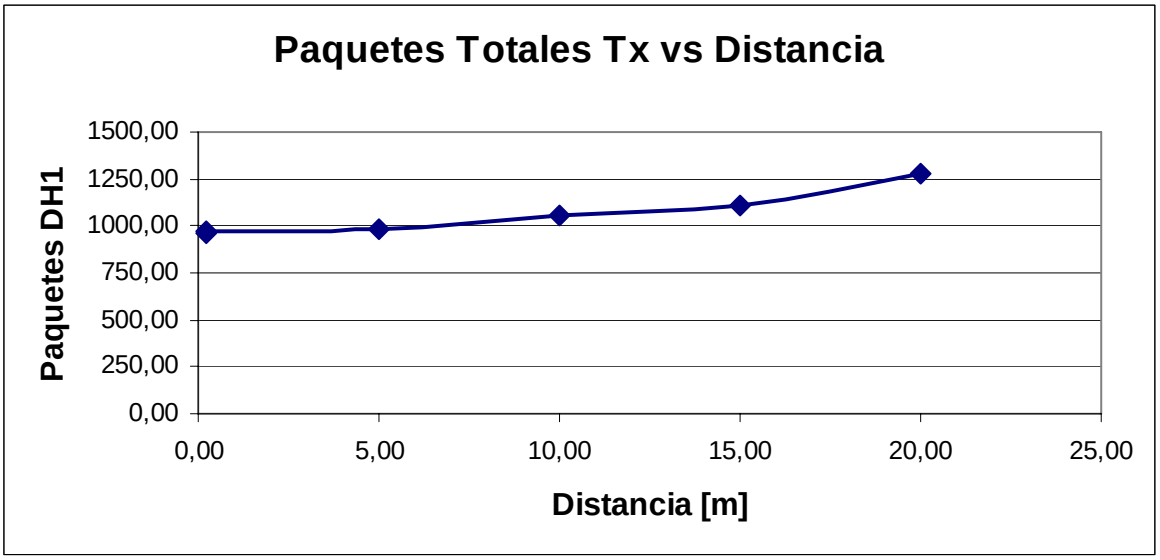

Figura. 4. 20. Número de paquetes totales transmitidos según la distancia sin línea de vista

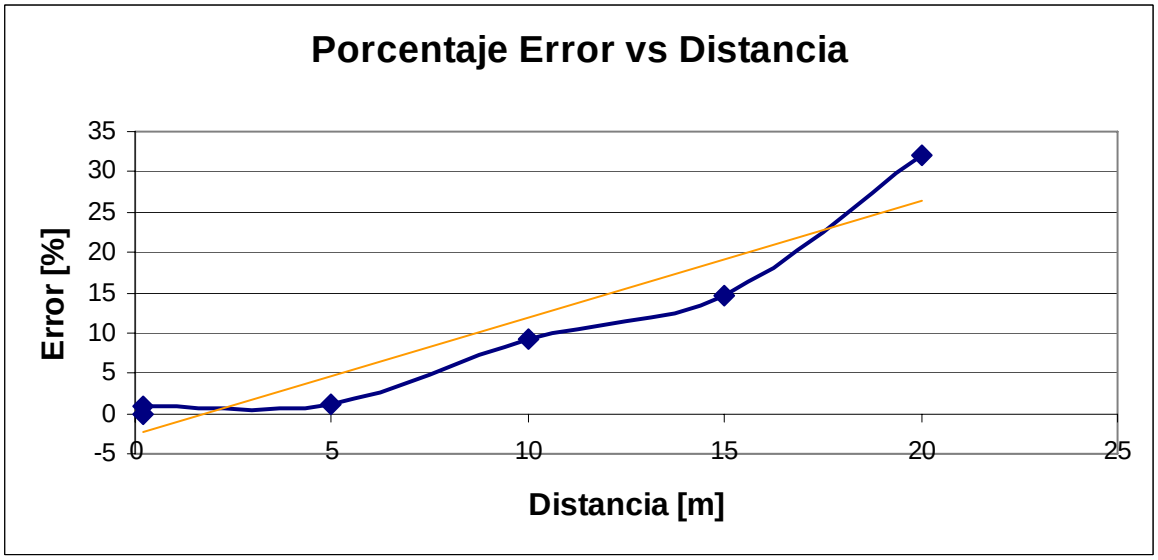

Figura. 4. 21. Porcentaje de error según la distancia sin línea de vista

4.1.7. Problemas en la transmisión de datos. La utilización del dispositivo BT-1000 para la transmisión de datos con la aplicación FTP no presentó ningún problema considerando que es un dispositivo de clase 2 y se estima

un buen desempeño con otros nodos siempre que estos se encuentren en un radio de 10m de distancia. De esta forma tanto con y sin línea de vista se tiene una diferencia de error menor al 10% con respecto al tiempo que lleva transmitir la misma información en la condición más óptima (línea de vista a 20cm de separación entre nodos).

Es importante resaltar que los porcentajes de error obtenidos en la práctica, además de ser paquetes retransmitidos, significarán una demora de corta duración que para el usuario que está utilizando la aplicación será transparente; es decir, no la percibirá.

4.2. TRANSMISIÓN DE VOZ

La prueba de transmisión de voz se la realizó con dos dispositivos BT-1000 del laboratorio de SAT de la ESPE. Con ayuda del software disponible en el kit de desarrollo BT-1000 y específicamente con la aplicación de voz (*Voice*) se puede establecer la comunicación entre dos puntos remotos mediante un enlace Bluetooth. El dispositivo BT-1000 cuenta con la especificación Bluetooth v1.1.

En la ejecución de la aplicación *Voice* se establece una conexión SCO sobre la conexión ACL existente y se utiliza un enlace SCO, donde tanto la velocidad de transmisión y recepción permanecen constantes y cualquier paquete que se encuentre con error no será retransmitido y será procesado tal y como llega a su destino. La comunicación *full-duplex* establecida entre los dos dispositivos BT-1000 se la realiza a una velocidad de 64Kbps con el uso de paquetes HV1 que cuentan con FEC de 1/3 y no hacen uso de CRC. La Figura 4.22 presenta la estructura del paquete HV1.

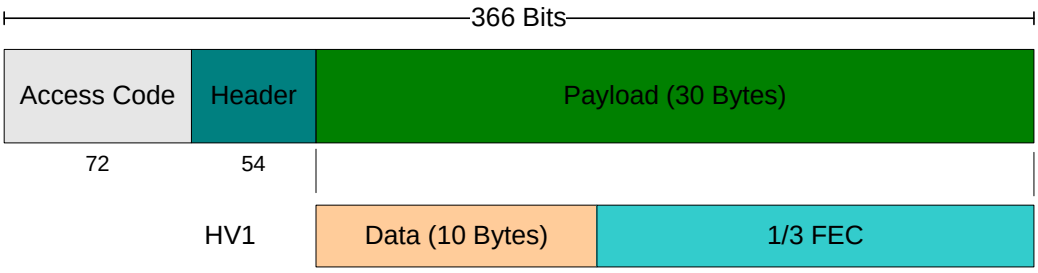

Figura. 4. 22. Paquete HV1 utilizado para la comunicación de voz

La realización de la prueba tiene como objetivo observar la degradación de la voz en un enlace Bluetooth con utilización de paquetes HV1 según el aumento de la distancia entre dos puntos. Para ello se cuenta con dos dispositivos BT-1000 que son alimentados mediante puerto USB, dos computadoras portátiles instaladas *Adobe Audition 2.0*, y dos cables de audio estéreo machos en ambos extremos respectivamente. El primer cable se conectara a la línea de salida del computador1 y al *jack* de *headset* (CON7) del BT-1000 número uno y el segundo cable de audio se conecta desde el *jack* de *headset* (CON7) del segundo dispositivo BT-1000 a la línea de entrada del computador2.

La aplicación *Voice* que permite la comunicación de voz en tiempo real implementa el perfil *Headset* con el cual se controlará la entrada y salida de audio y además la configuración de comunicación de voz. Como se explicó en capítulos anteriores, para establecer la comunicación entre dos dispositivos Bluetooth es necesario una serie de pasos, como *Inquiry*, *Paging* (establecimiento de la conexión), conexión con capas superiores, y finalmente la comunicación. Estos pasos se los presentó en las secciones 4.1.1 y 4.1.2. Debido a que el paquete HV1 utilizado en esta conexión cuenta con 10Bytes (80 bits) dedicados para datos del usuario y cada paquete es enviado cada slot de tiempo, efectivamente se verifica que se cuenta con 64000 Kbps para la comunicación de voz, esto se lo realiza considerando la frecuencia de salto utilizada, es decir 1600 saltos por segundo y tomando en cuenta que la mitad de los saltos serán empleados por el maestro y la otra mitad por el esclavo.

$$Tasa\ de\ Tx\ de\ Datos = 10[Bytes] \cdot \frac{8[bits]}{1[Byte]} \cdot \frac{1600[saltos]}{1[s]} \cdot \frac{1}{2}$$

$$Tasa\ de\ Tx\ de\ Datos = 64000\,[bps]$$

$$Tasa\ de\ Tx\ de\ Datos = 64\,[Kbps]$$

Debido a que la aplicación de voz implementa el perfil *Headset*, el cual es dependiente de los perfiles GAP y SPP mencionados en la sección 3.1 y 3.2 respectivamente, una serie de pasos son necesarios para permitir la comunicación *Full-Duplex* requerida entre el dispositivo *Audio-Gateway* (utilizado para E/S del audio, normalmente computadores personales o teléfonos celulares) y el dispositivo *Headset* (utilizado como E/S de audio del dispositivo remoto). De acuerdo a estos dos dispositivos, el *Audio-Gateway* es el encargado de controlar el establecimiento y liberación del enlace SCO, mientras que el *Headset* se encargará directamente de conectar o desconectar las cadenas de audio internas según se encuentre establecido o liberado el enlace SCO respectivamente. Las Figuras 4.23 y 4.24 presentan a dispositivos *Audio-Gateway* y *Headset*, y el modelo de protocolos involucrados en este perfil.

Figura. 4. 23. Dispositivos de Audio Gateway y Headset

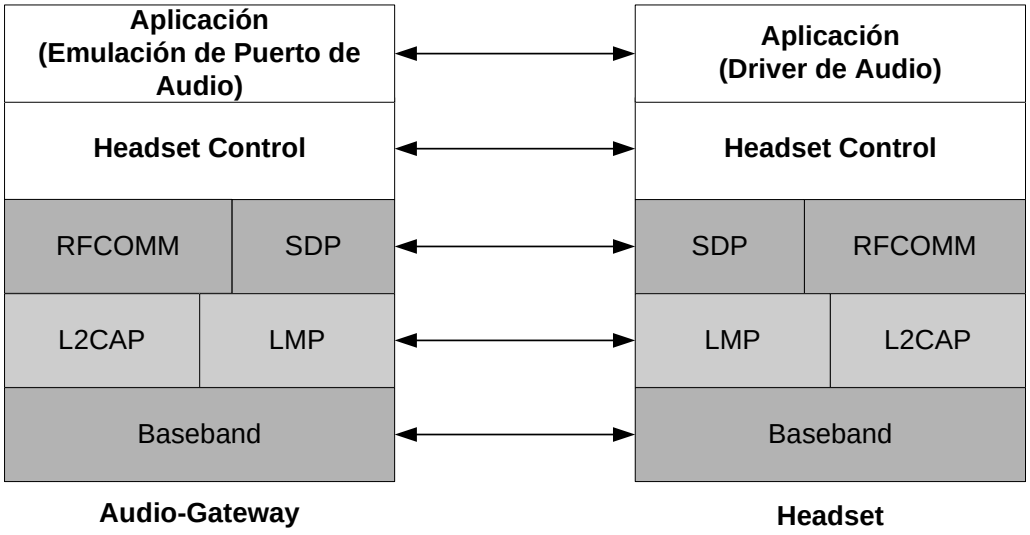

Figura. 4. 24. Modelo de protocolos involucrados en el perfil Headset

De la Figura 4.24, como se mencionó en capítulos anteriores, las capas de *Baseband*, LMP, y L2CAP corresponden a protocolos Bluetooth de capa 1 y 2 según el sistema OSI, SDP es el protocolo para el descubrimiento de servicios Bluetooth, y *Headset* Control es la entidad responsable específicamente del control de señalización (basado en comandos AT). Por otro lado, la Emulación de Puerto de Audio se refiere a la entidad encargada de emular la aplicación en dispositivos *Audio-Gateway*, mientras que el *Driver* de Audio es el software en el dispositivo Headset que permitir la aplicación. Aquí es necesario mencionar que únicamente una conexión entre *Headset* y *Audio-Gateway* es soportada a la vez y que el uso de la modulación CVSD (*Continuously Variable Slope Delta*) es obligatorio para dar como resultado un audio monofónico, cuya degradación de audio no es perceptible en condiciones normales.

El proceso de conexión entre el *Audio-Gateway* (AG) y el *Headset* (HS) se lo realiza con ayuda del *Headset* Control (Figura 4.24); aquí, el AG inicia la conexión (ACL) y una vez establecida se procede a enviar un código no solicitado de RING hacia el usuario como resultado o alerta de dicha conexión establecida, posterior a ello, AG realiza la inicialización de conexión SCO y nuevamente envía un RING al usuario para informarle de la creación del enlace SCO y esperará un comando

AT+CKPD=200 (generado por el usuario con la pulsación de un botón en el *Headset*) que mencionará su aceptación, en este punto AG enviará el comando OK y establecerá el enlace SCO. Es necesario mencionar que el último RING generado se lo repetirá periódicamente hasta que se reciba el comando AT+CKPD que indicará la aceptación para el establecimiento de la conexión. La Figura 4.25 presenta este procedimiento de conexión iniciado por el AG.

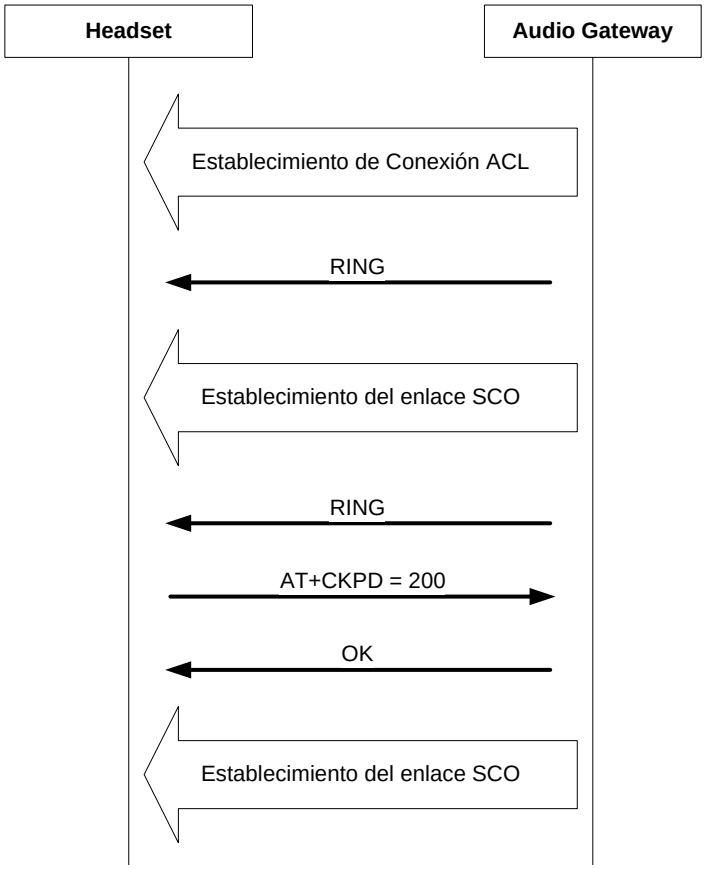

Figura. 4. 25. Establecimiento de conexión para audio iniciado por AG

Finalmente, después de lograr la comunicación de audio, la liberación de la conexión de audio puede iniciarse tanto por el HS o por el AG; en el HS se lo realiza con la pulsación del botón que indica terminación de la conexión o de la llamada y en el AG por intervención del usuario o por acciones internas. Las Figuras 4.26 y 4.27 presentan la liberación de la conexión iniciada por el HS y el AG respectivamente.

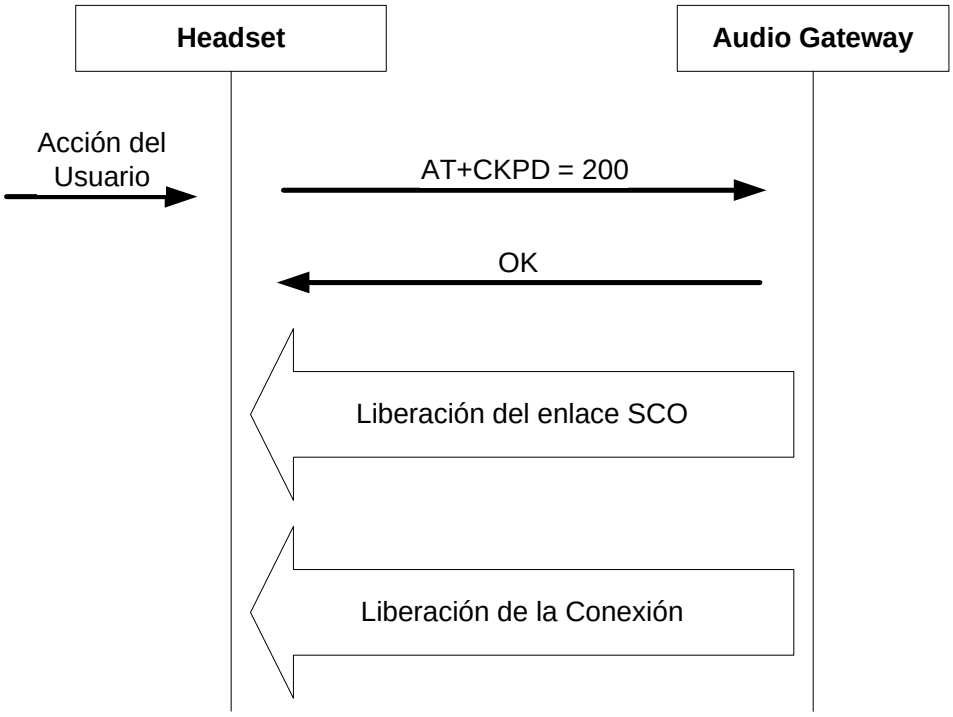

Figura. 4. 26. Liberación de la conexión de Audio iniciada por el HS

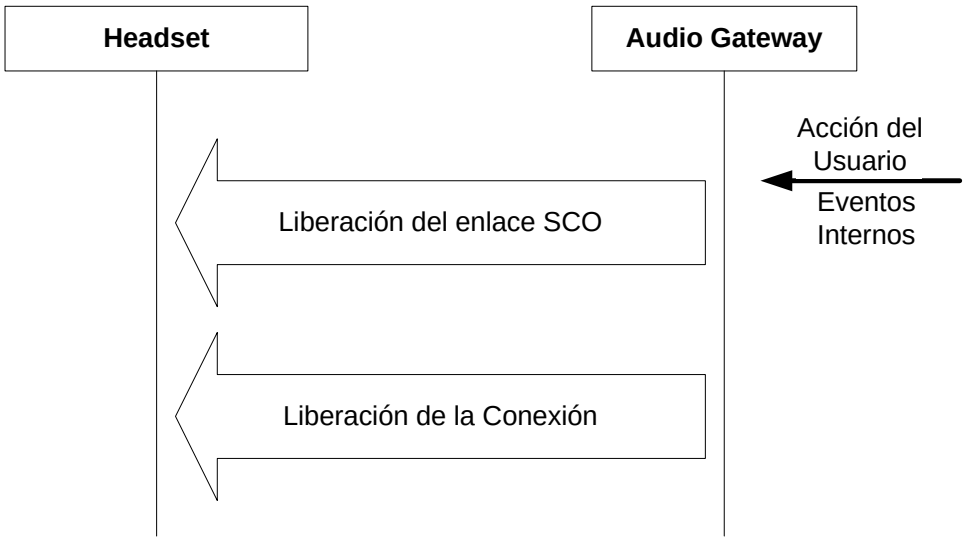

Figura. 4. 27. Liberación de la conexión de Audio iniciada por el AG

Para evaluar la degradación de la voz se compararon una señal fuente definida como referencia y grabada previamente con la herramienta de *AdobeAudition 2.0* a 44.100 Hz y con 16 bits por muestra y enviada desde el dispositivo iniciador; y la señal que llega en el punto de destino igualmente

grabada a 44.100 Hz y con 16 bits por muestra. Con ayuda del programa dtw.m [15] cuya implementación en Matlab está disponible en Internet y pertenece a la universidad de Columbia es posible comparar estas dos señales y encontrar su similaridad de acuerdo a un análisis vectorial. El programa dtw.m presenta como resultado dos gráficas que tanto su eje 'x' como 'y' son dependientes de las muestras tomadas de las dos señales ingresadas y a su vez de su duración. Además, dtw.m después de realizar todo su procesamiento entrega un resultado que representa el número de muestras no coincidentes en su análisis, es así que si este valor es 0 significará que existe coincidencia total y si el valor crece será lo contrario. La Figura 4.28 es el resultado de utilizar dtw.m para comparar dos señales periódicas idénticas y por lo tanto de igual duración. Como se observa, en la Figura de la derecha podemos notar que los cuadros sobre y bajo la línea roja son simétricos por lo que ambas señales serán idénticas.

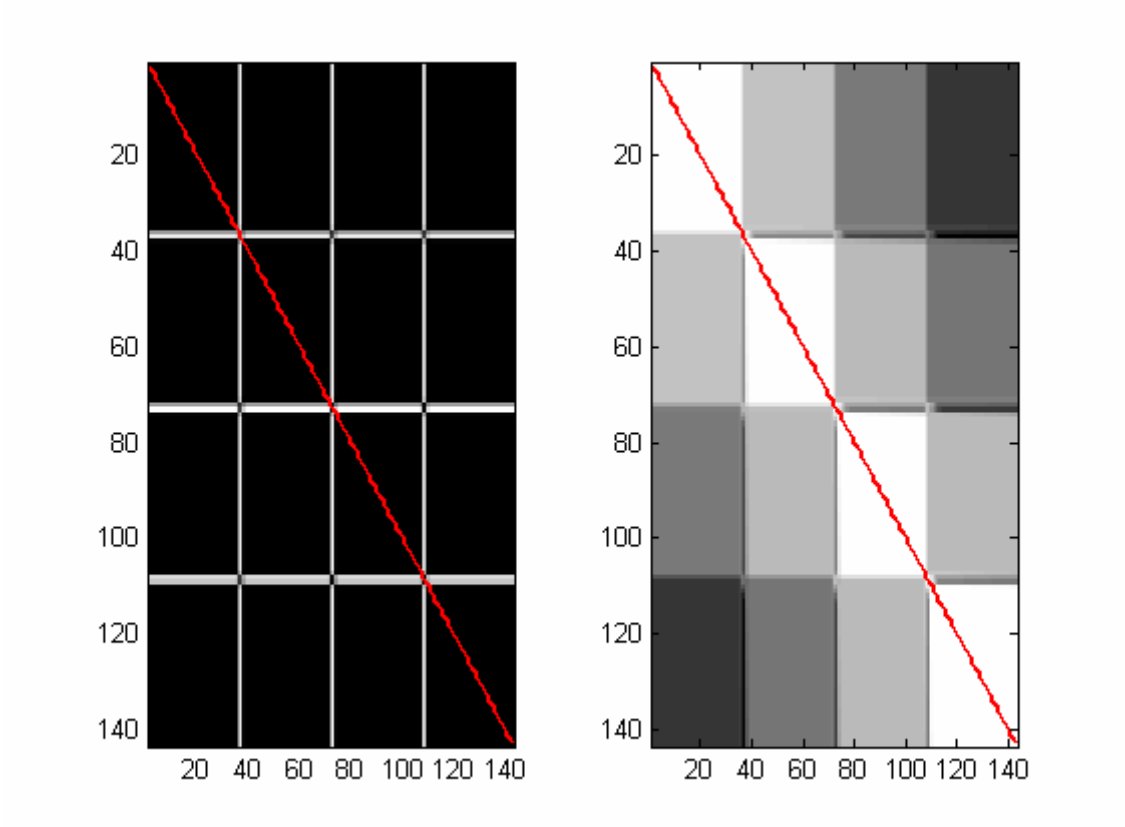

Figura. 4. 28. Comparación de dos señales periódicas e idénticas

El valor numérico entregado en el ejemplo del gráfico anterior es $3.8858x10^{-15}$, es decir 0, o total coincidencia. La Figura 4.29 es un ejemplo de la comparación de dos señales totalmente diferentes mediante dtw.m.

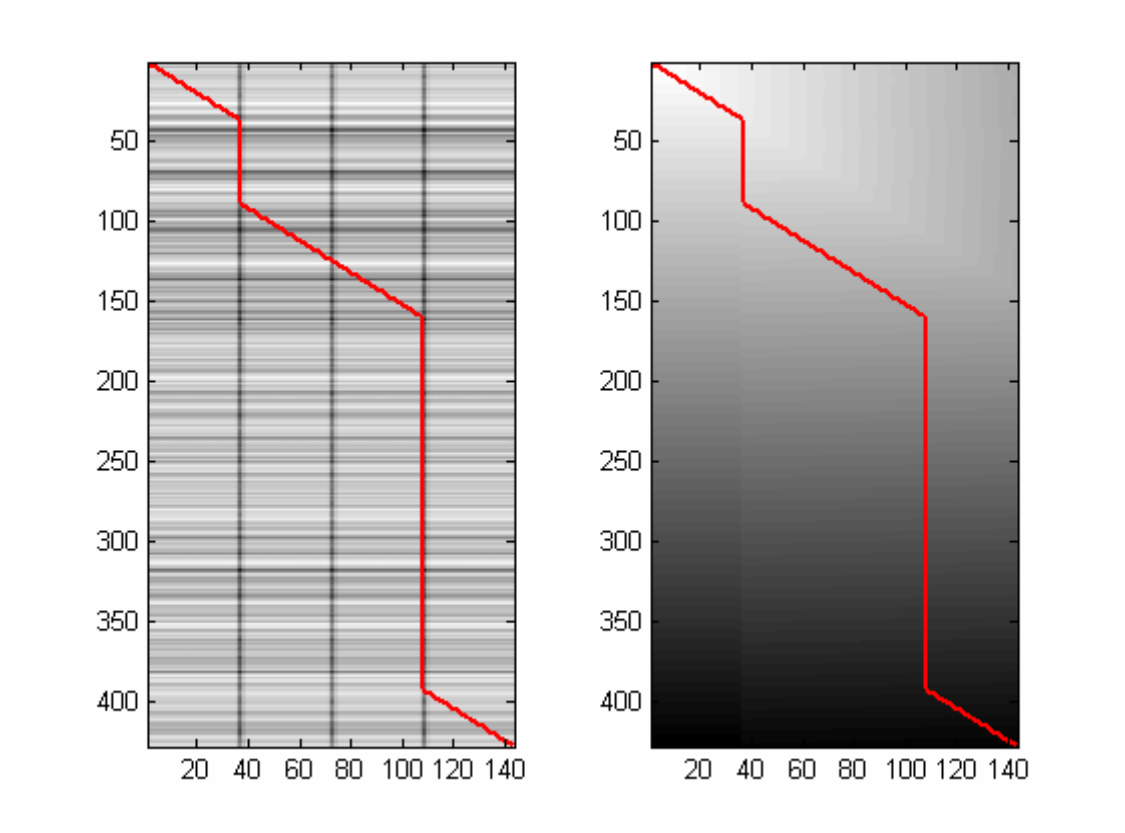

Figura. 4. 29. Comparación de dos señales totalmente distintas en duración y contenido

El valor numérico entregado en el ejemplo del gráfico 4.29 es de 398.8203, es decir mucho mayor a 0, o sin coincidencia en absoluto. Para obtener la degradación de la voz en un enlace Bluetooth según la variación de la distancia y para mayor precisión de la prueba se tomaron cinco muestras de voz cada cinco metros y se las procesó cada una de ellas en el programa dtw.m para posteriormente obtener un promedio de todas. La Tabla 4.11 presenta los resultados obtenidos y graficados en la Figura 4.30. En la prueba realizada se observó que la conexión SCO establecido para la comunicación de voz pudo soportar hasta una distancia de 80 metros entre los dos equipos BT-1000, pasada esta distancia resultó imposible realizar el proceso de Inquiry y por lo tanto establecer conexión.

Tabla. 4. 11. Resultados del valor numérico obtenido después de utilizar dtw.m con cada muestra

Distancia [m]		5,00	10,00	15,00	20,00	25,00	30,00	35,00	40,00
Muestra	1	6,20	9,37	9,33	19,85	10,07	17,69	7,54	29,36
	2	8,83	9,34	6,56	19,36	12,45	13,16	7,23	19,09
	3	0,22	9,06	8,36	19,31	19,49	15,40	8,98	18,58
	4	3,54	9,01	8,54	18,02	15,54	13,04	10,88	19,10
	5	1,43	6,79	9,35	19,03	10,67	11,40	9,35	18,23
	Promedio	4,04	8,71	8,43	19,11	13,64	14,14	8,80	20,87

Distancia [m]		45,00	50,00	55,00	60,00	65,00	70,00	75,00	80,00
Muestra	1	11,66	15,53	16,07	26,18	24,78	18,87	35,04	29,71
	2	10,07	11,12	13,62	28,54	24,74	23,63	35,60	27,38
	3	12,15	16,25	10,44	29,14	24,87	18,76	35,39	31,08
	4	10,26	14,16	16,30	33,45	27,03	25,42	36,45	35,47
	5	13,91	17,19	18,74	33,80	24,65	20,41	42,52	30,87
	Promedio	11,61	14,85	15,03	30,22	25,21	21,42	37,00	30,90

Promedio: Valor promedio de los valores entregados por dtw de las 5 muestras

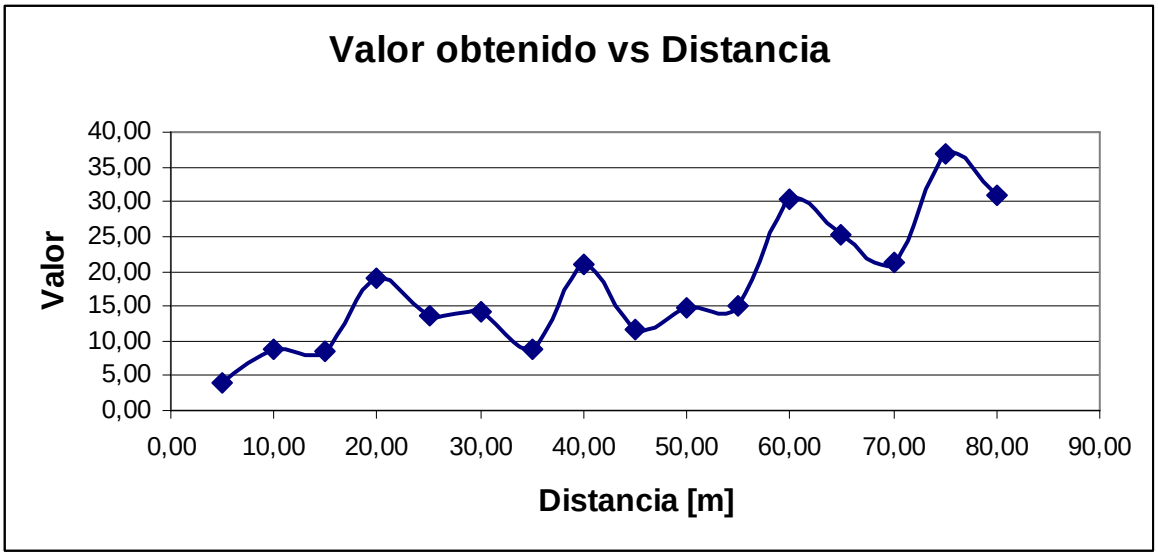

Figura. 4. 30. Valores de no coincidencia promedios obtenidos con dtw.m según la variación de la distancia

Para tener un patrón real de la degradación de la señal base (archivo prueba.wav) se la juntó con una señal completamente diferente con ayuda de *Adobe Audition 2.0*, de esta forma con el programa dtw.m se comparó la señal original con la mezcla de la señal original y una señal completamente distinta. La Figura 4.31 presenta la señal prueba.wav y tono.wav (señal completamente diferente a prueba.wav pero de igual duración). La Tabla 4.12 presenta estos resultados y se exponen en la Figura 4.33.

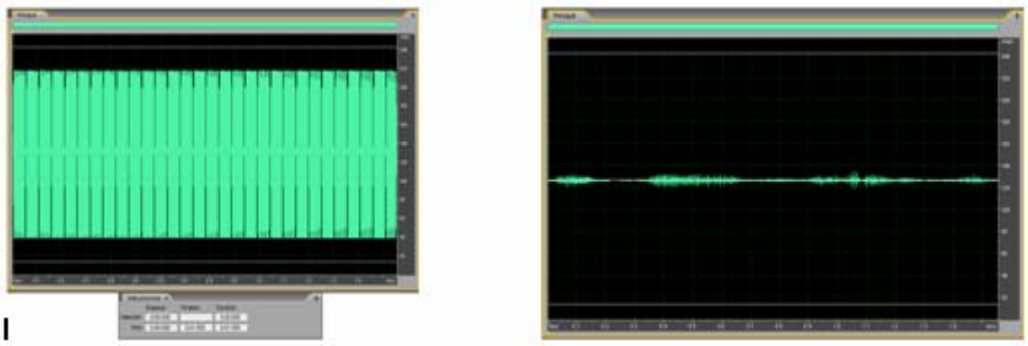

Figura. 4. 31. Señal generada completamente diferente a señal prueba.wav

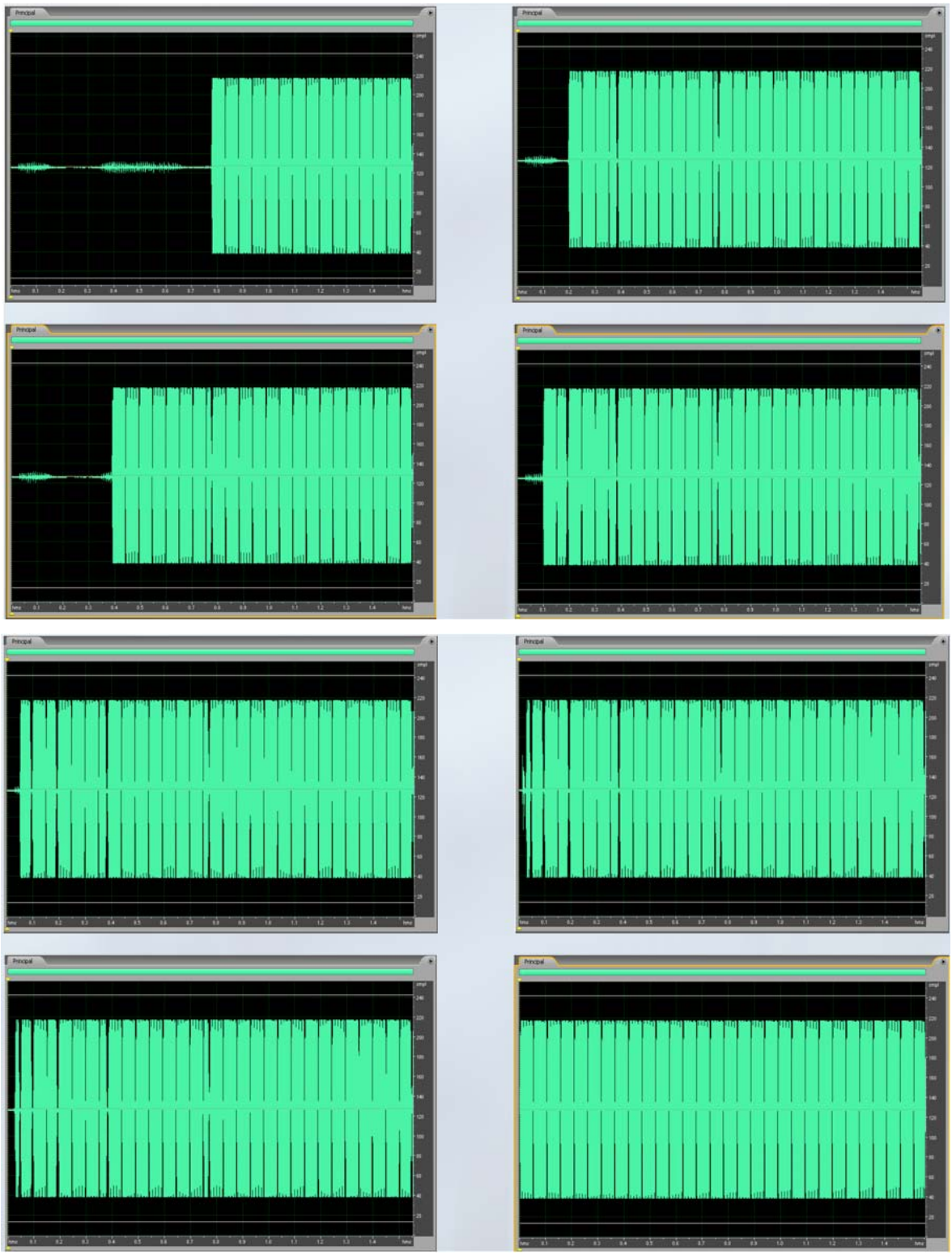

Figura. 4. 32. Mezcla de señal completamente diferente a prueba.wav y prueba

Tabla. 4. 12. Interpretación del valor numérico obtenido y su relación de igualdad según la variación de la señal original

Señal Original (%)	Señal Diferente (%)	Según los valores Obtenidos		
		Valor obtenido	Desigualdad	Igualdad
100	0	0	0	100
50	50	152,81	33,00	67,00
25	75	233,06	50,32	49,68
12,5	87,5	265,71	57,37	42,63
6,25	93,75	290,34	62,69	37,31
3,125	96,875	296,76	64,08	35,92
1,5625	98,4375	300,82	64,95	35,05
0,78125	99,21875	304,58	65,77	34,23
0,390625	99,609375	461,00	99,54	0,46
0	100	463,13	100,00	0,00

*Tanto al momento de comparar la señal original con respecto a la señal completamente diferente o viceversa se obtiene el mismo resultado (463)

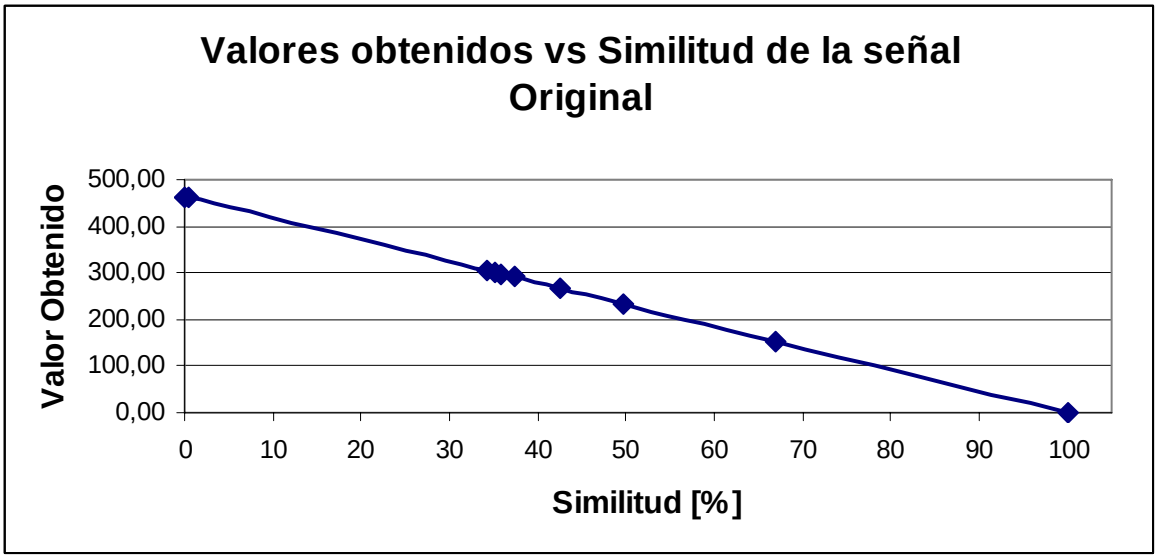

Figura. 4. 33. Número de no coincidencias según la similitud a la señal original

Utilizando esta referencia lineal se puede comparar o relacionar el valor promedio de muestras obtenido según la variación de la distancia y obtener el porcentaje de similitud con la señal original, o el equivalente a la degradación de la señal. La Figura 4.34 presenta la similitud de las muestras promedio obtenidas según la variación de la distancia con la señal original.

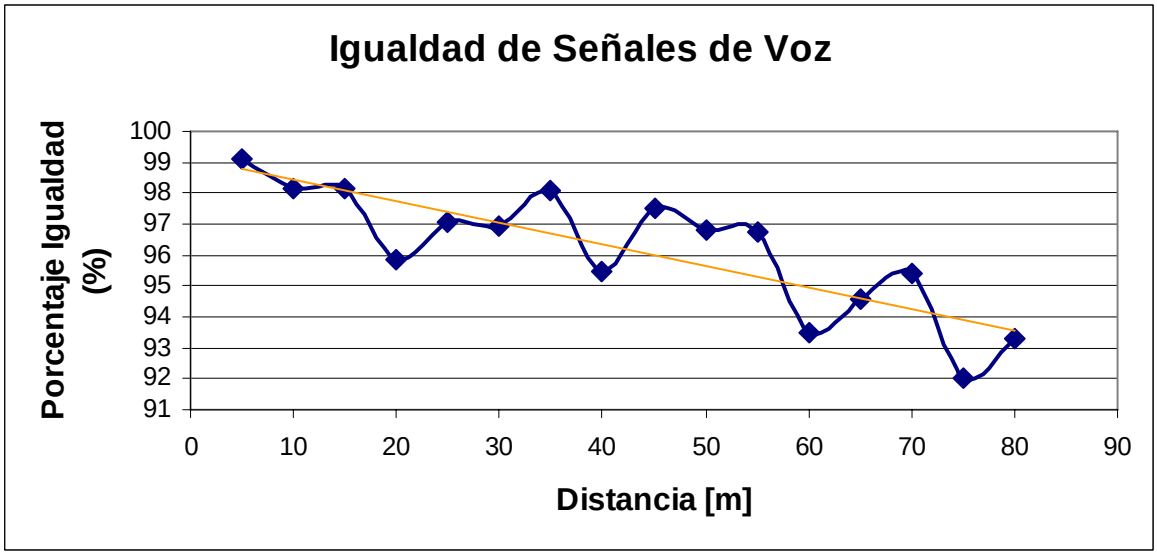

Figura. 4. 34. Similitud de las muestras promedio obtenidas según la variación de la distancia con la señal original

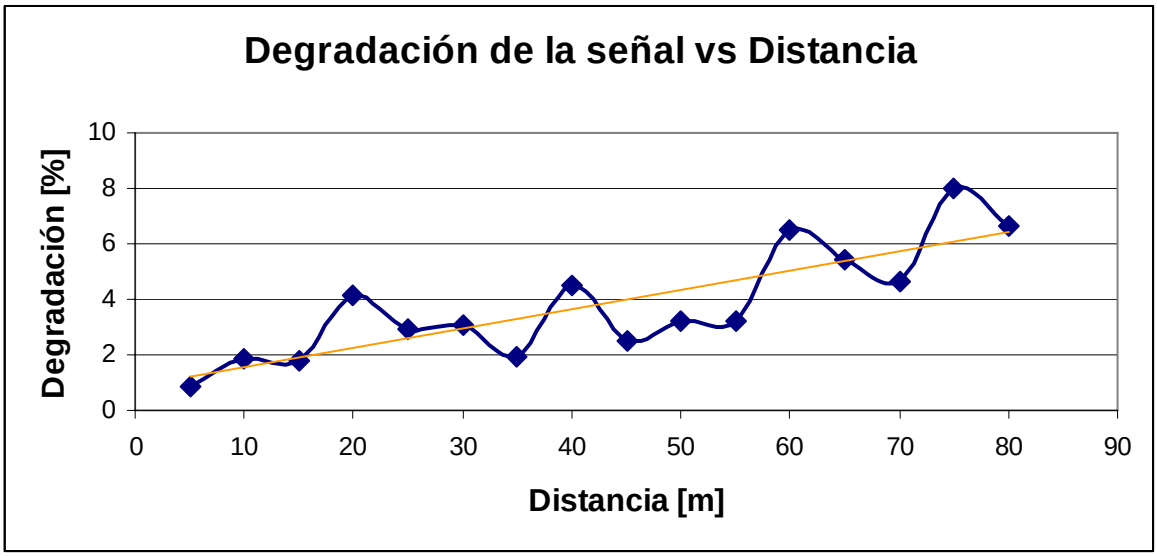

Figura. 4. 35. Degradación de la señal original según la variación de la distancia

4.2.1. Problemas en la transmisión de voz. La utilización de los dispositivos BT-1000 para la transmisión de voz mediante su aplicación *Voice* no presentó ningún problema considerando que dichos dispositivos pertenecen a clase 2, la cual establece dispositivos con radio efectivo de 10-100m como máximo, de tal forma y tomando en cuenta este parámetro la degradación de la voz a esta distancia entre nodos se mantuvo baja, llegando a una similitud del 98.12% entre la señal original y la recibida, cuya degradación no se percibe por el oído humano. La degradación más importante de la señal de voz ocurrió a partir de los 55m de separación entre nodos, a partir de esta distancia todas las muestras presentan un ruido más acentuado y fácilmente reconocible por el oído humano.

Es importante resaltar que los porcentajes de error obtenidos en la práctica, además de ser errores de bits en cada paquete HV1 transmitido, no significará que el usuario que utiliza la aplicación de voz no podrá recibir el mensaje, sino que con el aumento de la distancia entre nodos, la señal audible transmitida irá perdiendo claridad; es decir, cada vez aumentará el ruido en la voz del hablante.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

Un sistema Bluetooth está compuesto principalmente de tres partes que son el elemento de RF, el controlador del enlace de banda base, y el software para la administración del enlace. Para el funcionamiento de todos ellos se dispone de una memoria Flash en la cual se almacena todo el software necesario para su funcionamiento.

Al configurar un teléfono móvil, una PDA, un computador personal, o cualquier dispositivo que cuente con Bluetooth, en modo detectable (detectable mode) se permite a dicho dispositivo enviar señales que indican su estado de disponibilidad para inicializar procedimientos de pairing (emparejamiento o conexión) con cualquier otro dispositivo y así posteriormente establecer un enlace para la transferencia de información.

Cualquier dispositivo Bluetooth que se encuentre en modo detectable puede ser víctima del robo de su PIN (*Personal Identification Number*), que es usado para el proceso de autenticación, y así correr el riesgo de compartir información personal y del sistema sin su autorización con el dispositivo atacante. Por ello, es importante configurar cualquier dispositivo Bluetooth en modo no detectable con el fin de mantenerlo seguro y únicamente configurarlo en modo detectable cuando se desea realizar una transferencia de información. Así, es recomendable no guardar información sensible en un dispositivo Bluetooth puesto que puede ser utilizada por un intruso sin la autorización del usuario.

Habilitar la opción de autenticación OBEX brinda mayor seguridad al dispositivo Bluetooth puesto que obliga al usuario, además de ingresar su PIN manualmente en el procedimiento de atadura (bonding), a ingresar su contraseña OBEX que será utilizada para futuras autenticaciones. Así, serán dos contraseñas las que el usuario tendrá que ingresar para poder realizar transferencia de datos con otro dispositivo.

Escoger un código PIN lo suficientemente complejo con el fin de brindar mayor seguridad al dispositivo. Se recomienda que el PIN esté compuesto de por lo menos 5 dígitos para garantizar mayor seguridad.

El consumo de energía de un dispositivo Bluetooth depende de la clase a la que pertenece, de esta forma se requerirá mayor potencia y por lo tanto mayor consumo de energía en dispositivos clase 1, pero con la ventaja de cubrir mayores áreas y menor consumo por parte de dispositivos clase 2.

Bluetooth ofrece tres modos de seguridad de los cuales el modo 1 no cuenta con seguridad alguna por lo que autenticación ni autorización son realizadas, el modo 2 brinda seguridad, en cuanto a la restricción o no del uso de servicios, posterior a establecerse una conexión (requiere autenticación pero no autorización), y el modo 3 que resulta el más seguro y necesita de autenticación y autorización en todas las solicitudes antes de establecerse la conexión.

El uso de salto de frecuencia y *spread spectrum* (FHSS – *Frequency Hopping Spread Spectrum*) brindan mayor confiabilidad en la comunicación puesto que cualquier dispositivo que trate de escuchar la señal transmitida entre dos puntos no contará con la secuencia de saltos necesaria para entender la información transferida y por lo tanto dicha señal no será entendible. Además de ello, expandir la señal con SS (espectro ensanchado) la vuelve altamente resistente al ruido, a la interferencia, y resultaría difícil de interceptar.

Bluetooth al trabajar en la banda de 2.4GHz (ISM) y específicamente en el rango de frecuencias 2400-2483.2 [MHz], en el Ecuador no se requiere de pagos por el uso del espectro, por lo que puede ser utilizado sin permiso alguno.

Un dispositivo maestro es capaz de mantener hasta tres enlaces SCO con un esclavo o hasta un enlace SCO con tres dispositivos diferentes. De igual manera puede soportar hasta siete enlaces ACL con siete esclavos distintos, o un enlace SCO y otro ACL al mismo tiempo.

Para la transmisión de paquetes de voz o datos, el dispositivo maestro siempre utilizará los slots pares, mientras que los impares son destinados para la transmisión de los esclavos.

La velocidad de transmisión/recepción puede ser simétrica o asimétrica, y depende del tipo de paquete que se utiliza en la comunicación. De esta manera los paquetes que utilizan mayor número de slots de tiempo (DH3, DH5) para su transmisión podrán tener mayor espacio para transferir los datos propios del usuario, lo que aumentaría la velocidad de transmisión pero a su vez reduciría la velocidad de transmisión del segundo dispositivo Bluetooth involucrado. Asimismo, la utilización o no de FEC puede disminuir o aumentar la tasa de transmisión respectivamente. La máxima tasa de transmisión que puede suceder es de 723.2 Kbps (asimétricamente) en caso de utilizar paquetes DH5 que ocupan 5 slots de tiempo y no cuentan con FEC, por otro lado la velocidad de recepción se reducirá a 57.6Kbps.

Todo dispositivo Bluetooth es único, puesto que cuenta con un identificador de 48 bits (BD_ADDR) que nunca se repite y se compone por una parte del identificador del fabricante y por otra del número de dispositivo. Aquí es posible hacer una analogía con las direcciones MAC (*Media Access Control* - Dirección para el control de acceso al medio) utilizadas en diferentes tecnologías como Ethernet, ATM, o 802.11.

Bluetooth al utilizar TDD (*Time Division Duplex*) permite que la velocidad establecida para la comunicación de voz con enlaces SCO punto a punto siempre permanezca constante pero en caso de existir algún error en un paquete, dicho paquete no será retransmitido. Debido a que la velocidad de transmisión/recepción es fijada a 64Kbps la comunicación *Full-Duplex* es

soportada y se garantiza el entendimiento audible de la voz tanto para emisor como para receptor.

Si bien es cierto que los modos de ahorro de energía (*Sniff, Hold, y Park*) ayudan a disminuir el ciclo de trabajo de los dispositivos Bluetooth y extender la vida de sus baterías, es recomendable emplear fuentes de poder para alimentar a dispositivos que vayan a utilizarse en aplicaciones que requieran el funcionamiento de Bluetooth durante largos periodos de tiempo, caso contrario se requerirá del cambio de baterías de manera constante.

Toda piconet está sujeta y utilizará la secuencia de saltos de frecuencia que el maestro establezca, por lo tanto diferentes piconets obligadamente contarán con diferente secuencia de saltos. La secuencia de saltos es dependiente del reloj y del ID propio de cada dispositivo Bluetooth y para mantenerla se realizan actualizaciones periódicas desde el maestro hacia todos los esclavos.

El Maestro es encargado de decidir el orden de comunicación y la duración de la misma según el número de esclavos y el tipo de conexión establecida con cada uno de ellos.

El dispositivo BT-1000 al pertenecer a dispositivos clase 1 se estimaría que su radio de cobertura se encuentre alrededor de los 10-100m, aunque en la práctica es posible alcanzar una cobertura menor ya sea con línea de vista (80m tanto con enlaces ACL y SCO) o sin línea de vista 20m con enlaces ACL.

En conexiones ACL establecidas para la transferencia de archivos (FTP) el número de paquetes que requieren retransmitirse incrementan según el aumento de la distancia entre dos nodos.

En conexiones ACL establecidas para la transferencia de archivos (aplicación FTP) con el dispositivo BT-1000 se observó que cada 10m de aumento de la distancia entre los dos nodos el porcentaje de error incrementa un 4.75%; es decir, el 4.75% de los paquetes totales que se transmiten normalmente en condición de línea de vista y separación mínima entre dispositivos (20cm) es

aumentado en el proceso de transmisión. Mientras que en conexiones ACL sin línea de vista cada 5m de distancia que se aumenta entre dos nodos Bluetooth el porcentaje aumenta un 6% aproximadamente y su distancia máxima para soportar el enlace ACL se reduce ¼ de la distancia que soportaría con línea de vista. Así, en la prueba de datos sin línea de vista el radio disminuyó de 80m a 20m.

En el enlace SCO establecido para la comunicación de voz entre dispositivos BT-1000 se observó una degradación del 0.77% cada 10m de aumento entre los dos nodos con respecto a la señal original. En esta prueba se ratificó que el radio máximo que soportan los dispositivos BT-1000 es de 80m con línea de vista.

Buenos resultados se obtienen en la transmisión de datos con el dispositivo BT-1000 manteniendo línea de vista entre antenas y operando a una distancia de separación máxima de hasta 20m entre nodos (9.24% de error). A partir de esta distancia el número de retransmisiones podrá aumentar considerablemente incluso hasta un 64% más del número de paquetes necesarios en condiciones ideales. Para el caso de transmisiones sin línea de vista entre antenas la separación máxima y adecuada entre nodos se reduce a 10m (9.24% de error).

Buenos resultados se obtienen en la transmisión de voz con el dispositivo BT-1000 manteniendo línea de vista entre antenas y operando a una distancia efectiva de separación máxima de hasta 55m entre nodos (96.75% de aceptación). A partir de esta distancia la degradación de la voz se acentúa y pese a que el audio es todavía entendible por el oído, existe mayor ruido en la señal.

Existe una variedad de aplicaciones que utilizan o podrían utilizar la tecnología Bluetooth como alternativa para transmitir o recibir información, entre ellas cualquier aplicación que utilice sensores de movimiento, humedad, calor, humo, podrían juntarse a Bluetooth y trabajar inalámbricamente. Asimismo aplicaciones para el control de abertura de puertas, encendido de luces, controlar motores a pasos, entre otras.

ANEXOS

ANEXO 1

EVENTOS Y ACCIONES QUE SUCEDEN EN UN ESTADO PARTICULAR DEL CANAL

Tabla. A1.1 Eventos y acciones que suceden en un canal particular del canal

EVENTO	ESTADO ACTUAL	ACCIÓN	NUEVO ESTADO
LP_ConnectCfm	CLOSED	Señala el enlace físico como activo e inicia la conexión L2CAP	CLOSED
LP_ConnectCfmNeg	CLOSED	Señala el enlace físico como inactivo y suspende toda petición de conexión enviando el mensaje L2CA_ConnectCfmNeg a la capa superior	CLOSED
LP_ConnectInd	CLOSED	Señala el enlace como activo	CLOSED
LP_DisconnectInd	CLOSED	Señala el enlace como inactivo	CLOSED
LP_DisconnectInd	Cualquiera pero CLOSED	Envía a la capa superior el mensaje L2CA_DisconnectInd	CLOSED
LP_QoSViolationInd	Cualquiera pero OPEN	Descartado	N/C
LP_QoSViolationInd	OPEN	Envía a la capa superior el mensaje L2CA_QoSViolationInd. Si se tiene garantía de nivel de servicio, se termina el canal	OPEN o W4_L2CA_DISCONNECT_RSP
L2CAP_ConnectReq	CLOSED (CID dinámicamente asignado del grupo de libres)	Envía a la capa superior el mensaje L2CA_ConnectInd o envía al punto L2CAP_ConnectRspPnd	W4_L2CA_CONNECT_RSP
L2CAP_ConnectRsp	W4_L2CAP_CONNECT_RSP	Envía a la capa superior el mensaje L2CA_ConnectCfm y deshabilita el timer RTX	CONFIG

EVENTO	ESTADO ACTUAL	ACCIÓN	NUEVO ESTADO
L2CAP_ConnectRspPnd	W4_L2CAP_CONNECT_RSP	Envía a la capa superior el mensaje L2CA_ConnectPnd, deshabilita el timer RTX y activa el ERTX	N/C
L2CAP_ConnectRspNeg	W4_L2CAP_CONNECT_RSP	Envía a la capa superior el mensaje L2CA_ConnectCfmNeg, regresa el CID al grupo de libres y deshabilita los timers RTX y ERTX	CLOSED
L2CAP_ConfigReq	CLOSED	Envía al punto el mensaje L2CAP_ConfigRspNeg	N/C
L2CAP_ConfigReq	CONFIG	Envía a la capa superior el mensaje L2CA_ConfigInd	N/C
L2CAP_ConfigReq	OPEN	Suspende la transmisión de datos en un punto conveniente y envía a la capa superior el mensaje L2CA_ConfigInd	CONFIG
L2CAP_ConfigRsp	CONFIG	Envía a la capa superior el mensaje L2CA_ConfigCfm y deshabilita el timer RTX	N/C o OPEN
L2CAP_ConfigRspNeg	CONFIG	Envía a la capa superior el mensaje L2CA_ConfigCfmNeg y deshabilita el timer RTX	N/C

L2CAP_DisconnectReq	CLOSED	Envía al punto el mensaje L2CAP_DisconnectRsp	N/C
EVENTO	**ESTADO ACTUAL**	**ACCIÓN**	**NUEVO ESTADO**
L2CAP_DisconnectReq	Cualquiera pero CLOSED	Envía a la capa superior el mensaje L2CAP_DisconnectInd	W4_L2CA_DISCONNECT_RSP
L2CAP_DisconnectRsp	W4_L2CAP_DISCONNECT_RSP	Envía a la capa superior el mensaje L2CA_DisconnectCfm y deshabilita el timer RTX	CLOSED
L2CAP_Data	OPEN o CONFIG	Si se completa la recepción del paquete L2CAP, se envía a la capa superior el mensaje L2CA_Read para confirmar	N/C
L2CA_ConnectReq	CLOSED (CID dinámicamente asignado del grupo de libres)	Envía al punto el mensaje L2CAP_ConnectReq e inicia el timer RTX	W4_L2CAP_CONNECT_RSP
L2CA_ConnectRsp	W4_L2CA_CONNECT_RSP	Envía al punto el mensaje L2CAP_ConnectRsp	CONFIG
L2CA_ConnectRspNeg	W4_L2CA_CONNECT_RSP	Envía al punto el mensaje L2CAP_ConnectRspNeg y regresa el CID al grupo de libres	CLOSED
L2CA_ConfigReq	CLOSED	Envía a la capa superior el mensaje L2CA_ConfigCfmNeg	N/C
L2CA_ConfigReq	CONFIG	Envía al punto el mensaje L2CAP_ConfigReq e inicia el timer RTX	N/C

EVENTO	ESTADO ACTUAL	ACCIÓN	NUEVO ESTADO
L2CA_ConfigReq	OPEN	Suspende la transmisión de datos en un punto conveniente, envía al punto el mensaje L2CAP_ConfigReq e inicia el timer RTX	CONFIG
L2CA_ConfigRsp	CONFIG	Envía al punto el mensaje L2CAP_ConfigRsp	N/C o OPEN
L2CA_ConfigRspNeg	CONFIG	Se envía al punto el mensaje L2CAP_ConfgRspNeg	N/C
L2CA_DisconnectReq	OPEN o CONFIG	Se envía al punto el mensaje L2CAP_DisconnectReq e inicia el timer RTX	W4_L2CAP_DISCONNECT_RSP
L2CA_DisconnectRsp	W4_L2CA_DISCONNECT_RSP	Se envía al punto el mensaje L2CAP_DisconnectRsp y retorna el CID al grupo delibres	CLOSED
L2CA_DataRead	OPEN	Si se completa la carga, se transfiere la carga al InBuffer	OPEN
L2CA_DataWrite	OPEN	Se envía al punto el mensaje L2CAP_Data	OPEN
Timer_RTX	Cualquiera	Se envía a la capa superior el mensaje L2CA_TimeOutInd. Si expira el tiempo, se retorna el CID al grupo de libres	CLOSED
Timer_ERTX	Cualquiera	Se envía a la capa superior el mensaje L2CA_TimeOutInd. Si expira el tiempo, se retorna el CID al grupo de libres	CLOSED

Tabla. A1. 1. Eventos y acciones que suceden en un estado particular del canal

ANEXO 2

PROCEDIMIENTOS PARA EL NIVEL DE APLICACIÓN

En cuanto a los procedimientos para el nivel de aplicación [28], son tres los requerimientos que se deben cumplir para poder establecer una conexión vía cable serial emulada entre dos dispositivos.

a. Establecer el Enlace y Setear la Conexión Serial Virtual

i. Utilizar SDP para identificar el número del canal del servidor RFCOMM de la aplicación en el dispositivo remoto. En caso de contar con la opción de búsqueda y de conocer el servicio a contactar, se puede únicamente escoger el parámetro asociado con el servicio (Service Class ID).
ii. Se puede habilitar la opción de encriptación o la autenticación del dispositivo remoto.
iii. Realizar la petición de un nuevo canal L2CAP para la entidad RFCOMM remota.
iv. Iniciar una sesión RFCOMM en el canal L2CAP.
v. Iniciar una nueva conexión de enlace de datos en la sesión RFCOMM utilizando el número del canal del servidor mencionado en el paso i.

b. Aceptar el Enlace y Establecer la Conexión Serial Virtual

i. Si se requiere se puede proveer autenticación y después habilitar encriptación.
ii. Aceptar la indicación de establecimiento del nuevo canal de L2CAP.
iii. Aceptar el establecimiento de la sesión RFCOMM en ese canal.
iv. Aceptar la nueva conexión de enlace de datos en la sesión RFCOMM.

c. Registrar el Servicio en la Base de Datos Local de SDP

i. Todos los servicios o aplicaciones que utilizan RFCOMM deben tener un registro de servicio SDP, el cual incluye los parámetros necesarios para alcanzar dicho servicio o aplicación.

*Las llamadas a la base de datos y las respuestas a consultas SDP se realizan mediante una aplicación de ayuda, la cual ayuda al usuario en el proceso de configurar el puerto.

ANEXO 3

ESTABLECIMIENTO DE UNA SESIÓN OBEX

El establecimiento de una sesión OBEX se puede dar de dos maneras[5], con autenticación o sin ella.

a. SESIÓN OBEX SIN AUTENTICACIÓN

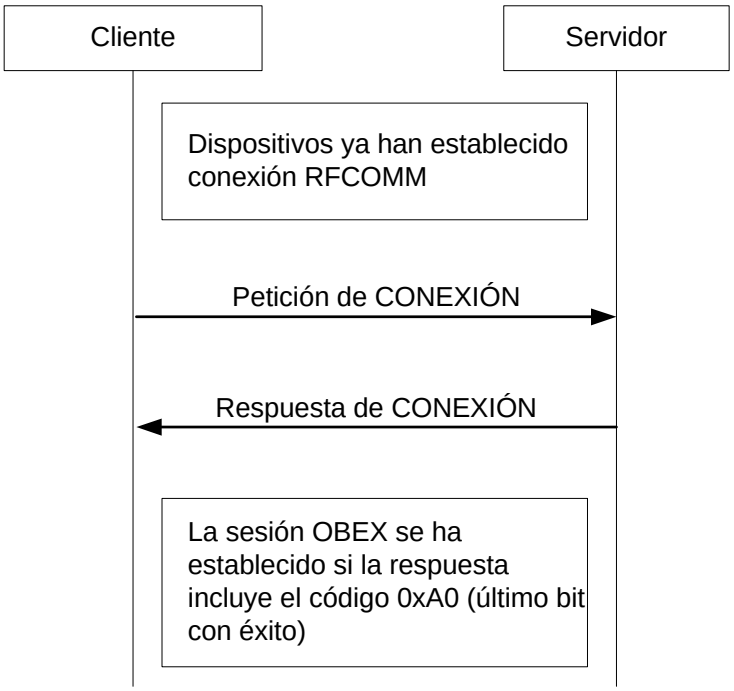

Figura. A3. 1. Proceso de establecimiento de sesión OBEX sin autenticación

En la petición de CONEXIÓN los campos que se deben mencionar son: el Opcode para la conexión (cuyo valor es 0x80), la longitud del paquete (Connect Packet Lenght), el número de versión de OBEX (OBEX Version Number), las banderas (Flags), y la longitud máxima del paquete OBEX (Max OBEX Packet Lenght), cuyos valores varían. Además, después de este último campo se especifica la cabecera Target, cuyo uso dependerá de los perfiles de aplicación.

[5] Fuente: Specifications of the Bluetooth System. Version 1.1. February 22 2001. Generic Object Exchange Profile.

Por otro lado, en la respuesta de CONEXIÓN los campos que se deben mencionar son: el código de respuesta para la petición de conexión (Response code for CONNECT request), la longitud del paquete de respuesta de conexión (Connect Response Packet Lenght), el número de versión OBEX (OBEX Version Number), las banderas (Flags), y la longitud máxima del paquete OBEX (Max OBEX Packet Lenght), cuyos valores varían. Además, después de este último campo y si en la petición de conexión se utilizó la cabecera Target se especificarán las cabeceras ConnectionID (indica la conexión para el servicio específico) y Who (cuyo valor es igual al de Target).

b. SESIÓN OBEX CON AUTENTICACIÓN

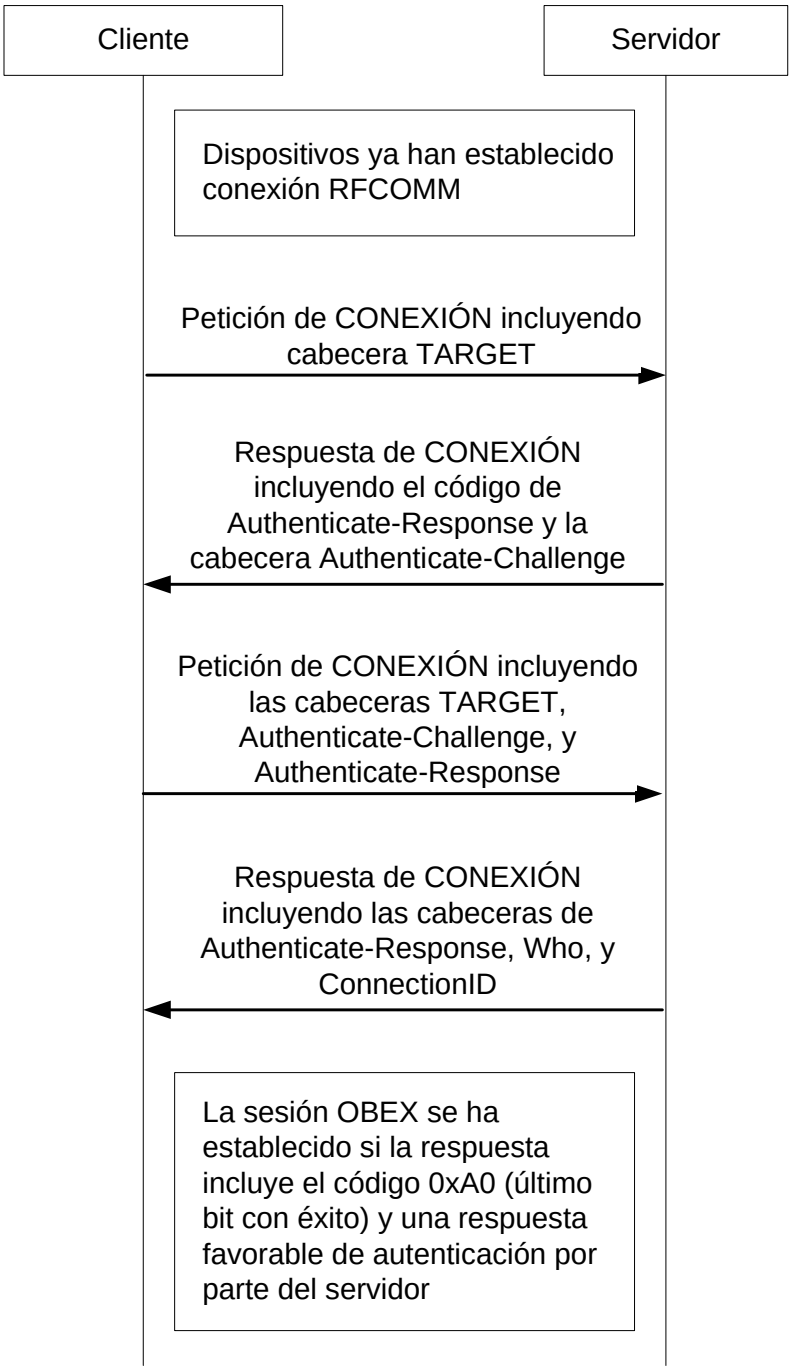

Figura. A3. 2. Proceso de establecimiento de sesión OBEX con autenticación

En la primera petición de CONEXIÓN los campos que se deben mencionar son los mismos utilizados en la sesión OBEX sin autenticación y corre el mismo principio con respecto a la cabecera Target.

En la primera respuesta de CONEXIÓN se mencionan los mismos campos utilizados en la sesión OBEX sin autenticación pero se agrega la cabecera Authenticate Challenge (lleva una cadena de dígitos).

En la segunda petición de CONEXIÓN los campos que se deben mencionar son los mismos utilizados en la primera petición de conexión y obligadamente agregar las cabeceras Authenticate Challenge (lleva la cadena de desafío del servidor) y Authenticate Response (lleva la cadena de respuesta para el servidor).

En la segunda respuesta de CONEXIÓN se mencionan los mismos campos utilizados en la primera respuesta de conexión y solamente especifica las cabeceras de Who (cuyo valor es igual al de Target) y Authenticate Response (como respuesta lleva la cadena de desafío del cliente).

ANEXO 4

SEÑALIZACIÓN EN EL ESTABLECIMIENTO Y LA LIBERACIÓN DE LLAMADAS INTERCOM

Para el establecimiento de una llamada de intercomunicación se realiza una petición de llamada (Call request) posterior al establecimiento de una conexión L2CAP orientada a la conexión, se espera por su confirmación (Call confirmation), una vez recibida la confirmación se establece la llamada de conexión (Call connection) – para la cual el establecimiento SCO debe iniciarse antes de enviar CONNECT – y en este punto las rutas del habla se conectarán al recibir CONNECT o CONNECT ACKNOWLEGDE. [6]

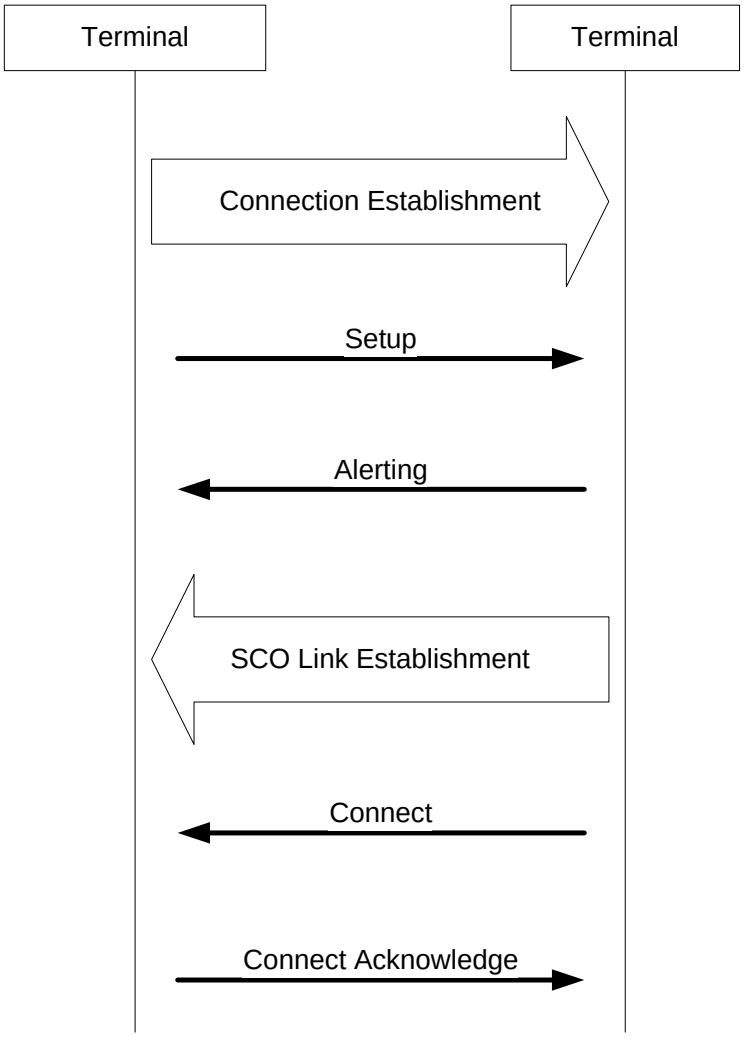

Figura. A4. 1. Establecimiento de llamada en Intercom

La Figura A4.1 presenta la señalización y procedimientos obligados para el establecimiento de llamadas de intercomunicación.

[6] Fuente: Specifications of the Bluetooth System. Version 1.1. February 22 2001. Intercom Profile. Signaling Flows.

Por otro parte, la Figura A.4.2 presenta los pasos para liberar la llamada de intercomunicación.

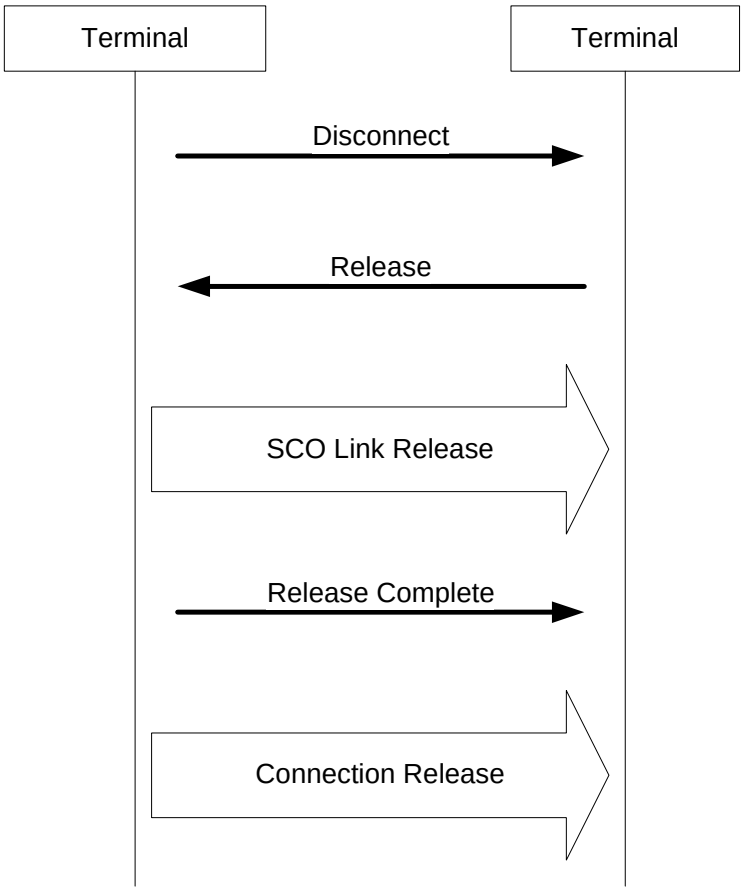

Figura. A4. 2. Liberación (clearing) de la llamada de Intercom

REFERENCIAS BIBLIOGRÁFICAS

[1] MULLER, Nathan, **Bluetooth Demystified,** 1ra Edición, McGraw-Hill Telecom, 8/Septiembre/2000.

[2] VARIOS AUTORES, **Specifications of the Bluetooth System**, vol. 2, Version 1.1., Bluetooth SIG, February 22 2001.

[3] MAN&TEL CO, **BT-1000 User Manual**, First Edition, Korea 2004.

[4] PARK, Sung-Jim y JUNG Kwang-Wook, **Wireless Embedded Training System,** First Edition, Man&Tel Co, Korea 2005.

[5] http://www.unescap.org/stat/gc/gcnl/gcnl14.asp, Bluetooth more versatile than infrared.

[6] http://whitepapers.zdnet.co.uk/0,1000000651,260150570p,00.htm, Mobile Working

[7] http://www.bluetooth.com/Bluetooth/Press/SIG/Bluetooth_Shipments_Climb_to_Five_Million_Per_Week.htm, Entregas de Bluetooth ascienden a cinco millones por semana. Mayo 24, 2005.

[8] http://www.cs.ucla.edu/NRL/wireless/uploads/rohit-voice.pdf, Carrying Voice over ACL Links.

[9] http://etd.adm.unipi.it/theses/available/etd-06112004-110434/unrestricted/02-Chapter-two.pdf, Bluetooth Baseband Specifications.

[10] http://www.ieee802.org/15/pub/2001/Nov01/01316r3P802-15_TG2-MAC-Scheduling-Mechanism-0926.doc, IEEE P802.15.

[11] http://www.commsdesign.com/main/2000/05/0005stand.htm, LMP protocol basics.

[12] Toolkit.http://clusterfie.epn.edu.ec/ibernal/html/CURSOS/AbrilAgosto06/Inalambricas/CLASES/BluetoothParteII.pdf, Especificación de la Administración del Enlace.

[13] http://www.swedetrack.com/usbluetooth_1.pdf, Using Bluetooth to Communicate with Automatically Controlled Vehicles.

[14] http://www.bluetooth.com/Bluetooth/Learn/Works/Profiles_Overview.htm, Bluetooth Wireless Technology Profiles.

[15] http://labrosa.ee.columbia.edu/matlab/dtw/. Dynamic Time Warp (DTW) in Matlab.

[16] http://www.bluetooth.com/bluetooth/, Bluetooth Technology.

[17] http://grouper.ieee.org/groups/802/15/pub/2000/May00/00013r9P802-15_WG-Final-SIGnal-Issue-No-5.pdf, The Best Innovative Technology.

[18] http://spanish.bluetooth.com/Bluetooth/Learn/Works/Data_Transport_Architecture.htm, Arquitectura del Transporte de Datos.

[19] http://www.tdx.cesca.es/TESIS_UV/AVAILABLE/TDX-0514104-141205//mu%F1oz.pdf, Arquitectura Abierta Escalable para Monitorización Domiciliaria: Aplicación a Pacientes con Patologías Cardiacas.

[20] http://medien.informatik.uni-ulm.de/~frank/research/hicss36.pdf, Bluetooth-based Ad-Hoc Networks for Voice Transmission.

[21] http://www.seas.upenn.edu/~swati/challenge_v4.pdf, Bluetooth Technology Key Challenges and Initial Research.

[22] http://www.eetasia.com/ARTICLES/2002JAN/2002JAN16_NTEK_RFD_CT_TA.PDF, Making the Bluetooth Application Connection.

[23] https://portal.fucapi.br/tec/imagens/revistas/ed002_016_028.pdf, Computación Móvil: Nuevas Oportunidades y Nuevos Desafíos.

[24] http://citeseer.ist.psu.edu/cache/papers/cs/25600/http:zSzzSzwww.markus-jakobsson.comzSzbluetoothzSzbluetooth.pdf/jakobsson01security.pdf, Security Weaknesses in Bluetooth.

[25] http://gospel.endorasoft.es/bluetooth/especificacion-bluetooth/arquitectura-de-protocolo/index.html, Arquitectura de Protocolos Bluetooth.

[26] http://www.teleco.com.br/es/tutoriais/es_tutorialbluetooth/pagina_1.asp, Bluetooth.

[27] http://msdn2.microsoft.com/EN-US/library/aa938549.aspx, Supporting Bluetooth Event Notifications.

[28] K5 – Serial Port Profile. Application Layer. Procedure Overview. <http://www.palowireless.com/infotooth/tutorial/k5_spp.asp>

FECHA DE ENTREGA

El proyecto fue entregado al Departamento de Eléctrica y Electrónica y reposa en la Escuela Politécnica del Ejército.

Sangolquí 19 de diciembre de 2007

ELABORADO POR:

_____

Fabián Andrés Varela Burbano
170939691-3

AUTORIDAD :

_____

Dr. Gonzalo Olmedo Cifuentes
Coordinador de la Carrera de Ingeniería en Electrónica y Telecomunicaciones