CONSTRUCCIÓN DE UNA ARQUITECTURA PARA LA INTEROPERABILIDAD DE LA TECNOLOGÍA BLUETOOTH CON REDES IP CABLEADAS PARA TRANSPORTE DE VOZ

JUAN PAULO GUZMÁN FLÓREZ
DARYAN FRANCISCO REINOSO ROJAS

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Popayán
2007

CONSTRUCCIÓN DE UNA ARQUITECTURA PARA LA INTEROPERABILIDAD DE LA TECNOLOGÍA BLUETOOTH CON REDES IP CABLEADAS PARA TRANSPORTE DE VOZ

JUAN PAULO GUZMÁN FLÓREZ
DARYAN FRANCISCO REINOSO ROJAS

Trabajo de grado presentado como requisito para obtener el título de Ingeniero en Electrónica y Telecomunicaciones

Director
Ingeniero JAVIER ALEXANDER HURTADO GUACA

UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE TELEMÁTICA
POPAYÁN
2007

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

ACRÓNIMOS

A2DP	Advanced Audio Distribution Profile
ACK	Acknowledge
ACL	Asynchronous Connection-Less
ACL-C	ACL Control
ACL-U	ACL User
ACM	Abstract Control Model
ADPCM	Adaptative Differential Pulse Code Modulation
API	Application Program Interface
ARP	Address Resolution Protocol
ARQ	Automatic Repeat Request
ASB	Active Slave Broadcast
ASB-U	ASB User
ATM	Asynchronous Transfer Mode
AVDTP	Audio/Video Distribution Transport Protocol
BD_ADDR	Bluetooth Device Address
BNEP	Bluetooth Network Encapsulation Protocol
BW	Bandwidth
CAC	Channel Access Code
CBR	Constant Bit Rate
CELP	Code Excited Linear Prediction Compression
CID	Channel Identifier
CMU	Central Michigan University
CODEC	Codificador-Decodificador
CQB	Class Based Queuing
CQDDR	Channel Quality Driven Data Rate
CRC	Cyclic Redundancy error Check
cRTP	Compressed Real Time Protocol
CS-ACELP	Conjugate Structure - Algebraic Code Excited Linear Prediction
CSMA/CD	Carrier Sense Multiple Access / Collision Detection
DAC	Device Access Code
DH	Data-High Rate Data packet type for high rate data
D-ITG	Distributed Internet Traffic Generator
DLL	Data Link Layer
DM	Data - Medium Rate Data packet type for medium rate
DSP	Digital Signal Processor
EDR	Enhanced Data Rate
eSCO	extended Synchronous Connection
eSCO-S	Stream eSCO (unframed)

FEC	Forward Error Correcting
GAP	General Access Profile
GFSK	Gaussian Frequency Shift Keying
GN	Group Ad-hoc Network
GOEP	Generic Object Exchange Profile
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
HCI	Host Controller Interface
HCRP	Hardcopy Cable Replacement Profile
HEC	Header Error Check
HIDP	Human Interface Device Profile
Home PNA	Home Phoneline Networking Alliance
HP	Hewlett-Packard
IAC	Inquiry Access Code
IAX	Inter-Asterisk eXchange
IEEE	Institute of Electrical and Electronics Engineers
IFQ	Interface Queue
IP	Internet Protocol
IPv6	Internet Protocol version 6
IPX	Internet Packet Exchange
ISDN	Integrated Services Digital Network
ITU-T	International Telecommunication Union – Telecommunication
L2CAP	Logical Link Control and Adaptation Protocol
LAN	Local Area Network
LAP	Lower Address Part
LLC	Logical Link Control
LMP	Link Manager Protocol
LPC	Linear Predictive Coding
LT_ADDR	Logical Transport ADDRess
MAC	Medium Access Control
ME	Management Entity
MGCP	Media Gateway Control Protocol
MOS	Mean Opinion Score
MP-MLQ	Multipulse, Multilevel Quantization
MRG	Metrics for the Evaluation of Congestion Control Mechanisms
MTU	Maximum Transfer Unit
Nam	Network Animator
NAP	Network Access Point
NAP	Non-significant Address Part
NAT	Network Address Translation
NS-2	Network Simulator-2

OBEX	OBject eXchange
OOP	Object Oriented Programming
OS	Operating System
OSI	Open Systems Interconnect
PAN	Personal Area Networking
PANU	PAN User
PCM	Pulse Code Modulation
PDA	Personal Digital Assistant
PDU	Protocol Data Unit a message
PHY	Physic
PPP	Point to Point Protocol
PSB	Parked Slave Broadcast
PSB-U	PSB User
PSQM	Perceptual Speech Quality Measurement
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RED	Random Earty Detection
RF	Radio Frequency
RFC	Request For Comments
RFCOMM	Radio Frequency Communication
RTP	Real Time Protocol
SAR	Segmentation and Reassembly
SCO	Synchronous Connection-Oriented
SCO-S	Stream SCO (unframed)
SDAP	Service Discovery Application Profile
SDP	Service Discovery Protocol
SEQN	Sequential Numbering
SIG	Special Interest Group
SIP	Session Initiation Protocol
SPP	Serial Port Profile
TCL	Tool Command Language
TCP	Transmission Control Protocol
TCS Bin	Telephony Control protocol Binary
UAC	User Agent Client
UAP	Upper Address Part
UAS	User Agent Server
UCBT	University of Cincinnati - BlueTooth
UDP	User Datagram Protocol
USB	Universal Serial Bus
VAD	Voice Activity Detection
VBR	Variable Bit Rate

VoIP	Voice over Internet Protocol
WAP	Wireless Application Protocol
WHCM	Wireless Handset Control Model
WLAN	Wireless Local Area Network

RESUMEN

Las comunicaciones inalámbricas, – dentro de las que se encuentran las redes de área personal Bluetooth –, han evolucionado rápidamente durante los últimos años, de igual forma como la interconexión de éstas con otro tipo de redes ha experimentado un importante avance en investigación y desarrollo.

Por otra parte las redes que operan bajo el estándar IEEE 802.3 conocidas ampliamente como redes *Ethernet* son una de las más populares redes IP difundidas en el mercado debido principalmente a su confiabilidad y simplicidad de instalación. Dichas redes han demostrado ser lo suficientemente robustas para la prestación de diversas clases de servicios soportados sobre éstas, dentro de los que se encuentran los relacionados con VoIP.

VoIP es una tecnología que permite a los usuarios realizar llamadas telefónicas utilizando cualquier red IP en lugar de las tradicionales líneas telefónicas o las actuales redes celulares GSM. Los beneficios de VoIP están normalmente divididos en dos categorías: reducción de costos, y la simplificación y convergencia de infraestructura. Es por esto que cada vez más empresas que se dedican a la prestación de servicios de VoIP se preocupan y orientan sus soluciones hacia un ambiente de interoperabilidad completo que permita a sus clientes operar como entidades integradas altamente eficaces.

Por lo anterior, en el presente trabajo de grado se estudian los criterios y requerimientos tecnológicos necesarios para la construcción de una arquitectura que permita la interoperabilidad de Bluetooth con redes *Ethernet* como base para la prestación de servicios de VoIP.

INTRODUCCION

Las comunicaciones inalámbricas, – dentro de las que se encuentran las redes de área personal–, han evolucionado rápidamente durante los últimos años, de igual forma como la interconexión de éstas con otro tipo de redes ha experimentado un importante avance en investigación y desarrollo. En febrero de 1998, los líderes de la telefonía móvil y la computación (Ericsson, Nokia, IBM, Toshiba, e Intel) formaron un grupo de interés especial (SIG) para crear una interfaz de radio estándar llamada Bluetooth. Debido a las características de servicio tales como "bajo costo, bajo consumo de potencia y reemplazo de cables", la tecnología Bluetooth se ha convertido en la solución líder en redes de área personal y comunicaciones inalámbricas de corto rango. Un gran número de dispositivos electrónicos tales como impresoras, teléfonos celulares, laptops, PDAs, etc. están siendo construidos con chips Bluetooth incorporados.

Por otra parte el protocolo de Internet IP está diseñado para uso en sistemas interconectados de redes de conmutación de paquetes. IP provee bloques de transmisión llamados datagramas desde las fuentes hasta los destinos (llamados *hosts*) identificados por direcciones de longitud fija. IP también provee, si es necesario, fragmentación y reensamblado de paquetes largos, para transmisión de "paquetes pequeños" a través de la red.

El protocolo IP está específicamente limitado para proveer exclusivamente funciones necesarias para entregar paquetes de bits (datagramas IP) desde la fuente hasta el destino sobre un sistema de redes interconectado. No existen mecanismos para garantizar confiabilidad de datos extremo a extremo, control de flujo, secuenciamiento, u otros servicios encontrados en otros protocolos de conexión.

Las redes que operan bajo el estándar IEEE 802.3 conocidas ampliamente como redes Ethernet son una de las más populares redes IP difundidas en el mercado debido principalmente a su confiabilidad y simplicidad de instalación. Dichas redes han demostrado ser lo suficientemente robustas para la prestación de diversas clases de servicios soportados sobre éstas, dentro de los que se encuentran los relacionados con VoIP. Por este motivo, las redes Ethernet han servido como una alternativa idónea para dar solución a las comunicaciones de las empresas. Es sabido que la principal forma de comunicación rápida dentro de una organización son las llamadas telefónicas, las cuales están migrando rápidamente hacia la tecnología de VoIP.

Voz sobre protocolo IP es una tecnología que permite a los usuarios realizar llamadas telefónicas utilizando cualquier red IP en lugar de las tradicionales líneas telefónicas o las actuales redes celulares GSM. Los beneficios de VoIP están normalmente divididas en dos categorías: reducción de costos, y la simplificación y convergencia de infraestructura. Es por esto que cada vez más empresas que se dedican a la prestación de servicios de VoIP se preocupan y orientan sus soluciones hacia un ambiente de interoperabilidad completo que permita a sus clientes operar como entidades integradas altamente eficaces, mostrando que el gran número de soluciones para la diversidad de escenarios existentes, como por ejemplo sistemas de comunicaciones para hospitales, sector empresarial, sector educativo, sector productivo etc., deben estar enfocadas hacia la

interoperabilidad de las redes IP existentes con las nuevas plataformas tecnológicas adquiridas.

En general, las nuevas tecnologías de red que han surgido en los últimos años se han desarrollado con la filosofía de interoperabilidad con las tradicionales redes Ethernet pero principalmente con el protocolo IP con el ánimo de generar un ambiente de convergencia tanto de servicios como de infraestructura de red.

Consecuentemente al aprovechar las redes IP ya existentes dentro de muchas de las instituciones del sector empresarial actual, se establece una reducción de los costos de red y de gestión de las redes a nivel global[1], al mismo tiempo que se constituyen parámetros y criterios que ayuden a fortalecer la base técnica que soporta la interoperabilidad de las mencionadas redes inalámbricas como tecnologías de acceso o de "último kilómetro" con las redes IP cableadas tradicionales funcionando como "*backbone*"[2] en esta clase de sistemas, aprovechando las ventajas de cobertura y flexibilidad que son propias de una tecnología inalámbrica.

Al establecer los requerimientos necesarios referentes a la prestación de servicios de voz en tiempo real bajo condiciones específicas de alta movilidad y rápida localización de personal, se logra establecer una base que en conjunto con los criterios de QoS a tener en cuenta para la prestación de servicios de voz sobre redes de datos, soportan la definición de una arquitectura para la interoperabilidad de la tecnología Bluetooth con redes cableadas Ethernet que puede ser validada utilizando herramientas de simulación o prototipos funcionales permitiendo finalmente la implementación de servicios de voz sobre ésta.

Igualmente con la construcción de una arquitectura basada en criterios de interoperabilidad y su respectiva validación, es posible formular recomendaciones que guíen la implementación de servicios de voz soportados sobre la propia arquitectura.

Así en el presente trabajo de grado se aborda los temas necesarios para la consecución de un objetivo final claro estructurados de la siguiente manera:

En el capítulo 1 se realiza un estudio de la tecnología inalámbrica Bluetooth, las redes de datos IP cableadas tradicionales Ethernet, especificando aspectos como características de la tecnología Bluetooth, perfiles básicos del estándar y se da una primera aproximación a un escenario de interoperabilidad.

En el capítulo 2 se definen criterios y parámetros de interoperabilidad necesarios para la prestación de servicios de voz en ambientes de alta movilidad y rápida localización de personal tales como las características particulares en el manejo de la voz sobre redes de datos. Adicionalmente se describen claramente los perfiles y protocolos que fundamentan la construcción de una arquitectura que incorpore la interoperabilidad de la tecnología Bluetooth con redes IP cableadas tradicionales como soporte para la prestación de servicios de voz en tiempo real.

[1] Global, en el sentido de la totalidad de las redes existentes dentro de la institución.
[2] Backbone: red de mayor nivel, que conecta redes de menor nivel en la jerarquía. Se prefiere el uso de este término en inglés.

En el capítulo 3 se explica en detalle los aspectos concernientes a los requerimientos tecnológicos necesarios para el transporte de voz en ambientes de interoperabilidad en especial los de Calidad de Servicio ofrecidos por la tecnología Bluetooth, que soportan la definición de una arquitectura que permita la implementación de soluciones de voz.

En el capítulo 4, se define la arquitectura concebida a partir de los criterios y parámetros evaluados en el capítulo 2 así como también de los requerimientos tecnológicos abordados en el capítulo 3 finalizando con una explicación de alcances y limitaciones de la arquitectura diseñada.

En el capítulo 5 se identifican las herramientas de simulación para la tecnología Bluetooth y *stack*[3] de protocolos Bluetooth, se justifica la selección de la herramienta NS-2 y el paquete UCBT para la validación de la arquitectura. Finalmente se detallan los resultados obtenidos en el proceso de validación, se explica los resultados obtenidos con la construcción de un prototipo funcional específicamente para brindar el soporte de servicios de voz en tiempo real.

[3] *Stack*: pila de protocolos, se prefiere el uso de este término en inglés.

CAPÍTULO 1 ESTUDIO DE LA INTEROPERABILIDAD DE LA TECNOLOGÍA BLUETOOTH Y LAS REDES IP CABLEADAS ETHERNET.

1.1 DESCRIPCIÓN GENERAL DE LA TECNOLOGÍA INALÁMBRICA BLUETOOTH.

Bluetooth es un sistema de radio que opera en la banda de frecuencia libre de *2.4 GHz,* banda de frecuencia disponible en la mayor parte del mundo. En Colombia el Ministerio de Comunicaciones reglamenta su uso así: "Las bandas, 902 - 924 MHz, 2.400 – 2.483,5 MHz y 5.725 – 5.850 MHz se atribuyen a título secundario, conforme con la resolución 3382 de 15 de diciembre de 1995, para los sistemas de espectro ensanchado" [2]. Esto se ve reflejado directamente sobre el mercado de los dispositivos móviles, que experimenta una gran penetración de la tecnología Bluetooth gracias a la compatibilidad con las regulaciones internacionales existentes.

Bluetooth utiliza 79 canales de radio frecuencia con un ancho de banda de 1 MHz cada uno y una rata máxima de símbolos de 1 MSímbolo/s. Después de que cada paquete es enviado en una determinada frecuencia de transmisión, ésta cambia a otra de las 79 frecuencias. El rango típico de operación de *Bluetooth* es menor a 10 metros, sin embargo se pueden alcanzar distancias de hasta 100 metros con el uso de amplificadores (Clase 1). Como se puede observar en la Figura 1.1, la comunicación sobre Bluetooth se divide en varias capas. A continuación se presenta una breve descripción de algunas de ellas y se tratarán en mayor detalle en secciones posteriores [3].

La capa de comunicación más baja es llamada banda base**.** Esta capa implementa el canal físico real. Emplea una secuencia aleatoria de saltos a través de 79 frecuencias de radio diferentes. Los paquetes son enviados sobre el canal físico, donde cada uno es enviado en una frecuencia de salto diferente. La Banda Base controla la sincronización de las unidades Bluetooth y la secuencia de saltos en frecuencia, además es la responsable de la información para el control de enlace a bajo nivel como el reconocimiento, control de flujo y caracterización de carga útil y soporta dos tipos de enlace: *asíncrono no orientado a la conexión* (ACL, *Asynchronous Connection-Less*), para datos y *síncrono orientado a la conexión* (SCO, *Synchronous Connection-Oriented*), principalmente para audio. Los dos pueden ser multiplexados para usar el mismo enlace de radio frecuencia (*RF, Radio Frequency).* Usando ancho de banda reservado, los enlaces SCO soportan tráfico de voz en tiempo real.

El *Protocolo de Gestión de Enlace (*LMP, *Link Manager Protocol)* es el responsable de la autenticación, cifrado, control y configuración del enlace. El LMP también se encarga del manejo de los modos y consumo de potencia, además soporta los procedimientos necesarios para establecer un enlace SCO.

El *Interfaz del Controlador de Enlace (*HCI, *Host Controller Interface)* brinda un método de interfaz uniforme para acceder a los recursos de hardware de Bluetooth*.* Éste contiene una interfaz de comando para el *controlador banda base,* la *gestión del enlace* y para acceder al hardware.

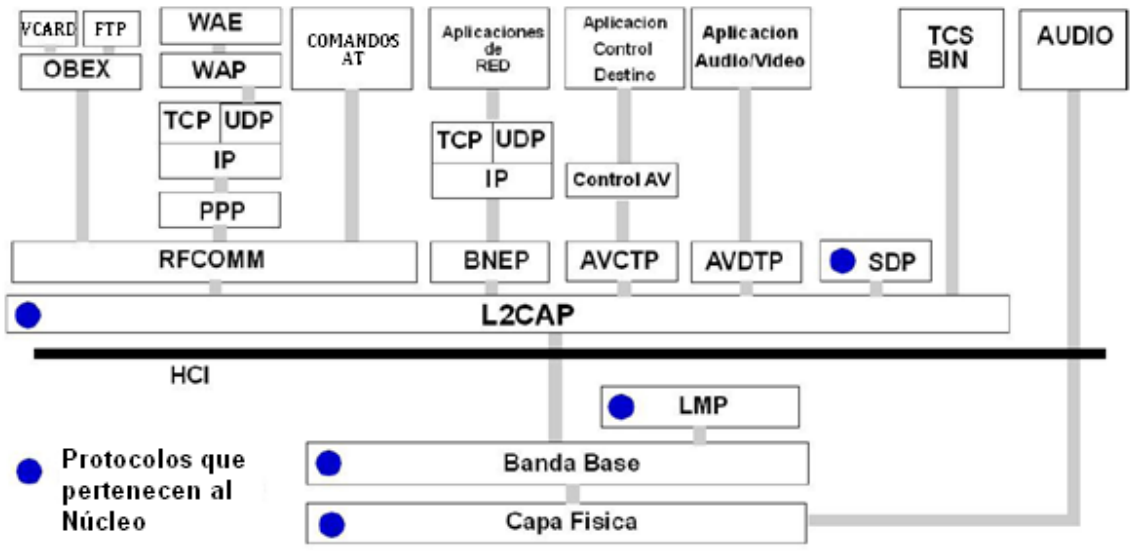


Figura 1.1 *Stack* de Protocolos de Bluetooth [3].

El *Protocolo de Control y Adaptación de Enlace Lógico (L2CAP, Logical Link Control and Adaptation Protocol),* corresponde a la capa de enlace de datos. Ésta brinda servicios de datos orientados y no orientados a la conexión a capas superiores. L2CAP multiplexa los protocolos de capas superiores con el fin de enviar varios protocolos sobre un canal banda base. Con el fin de manipular paquetes de capas superiores más grandes que el máximo tamaño del paquete banda base, L2CAP los segmenta en varios paquetes banda base. La capa L2CAP del receptor reensambla los paquetes banda base en paquetes más grandes para la capa superior. La conexión L2CAP permite el intercambio de información referente a la calidad de la conexión, además maneja grupos, de tal manera que varios dispositivos pueden comunicarse entre sí. El *Protocolo de Descubrimiento de Servicio (SDP, Service Discovery Protocol)* define como actúa una aplicación de un cliente *Bluetooth* para descubrir servicios disponibles de servidores Bluetooth, además, proporciona un método para determinar las características de dichos servicios.

El protocolo RFCOMM (*Radio Frequency Communication*) ofrece emulación de puertos seriales sobre el protocolo L2CAP. RFCOMM emula señales de control y datos RS-232 sobre la banda base Bluetooth. Éste ofrece capacidades de transporte a servicios de capas superiores (por ejemplo OBEX, *OBject eXchange)* que usan una línea serial como mecanismo de transporte. RFCOMM soporta dos tipos de comunicación, directa entre dispositivos actuando como *endpoints* y dispositivo-modem**-**dispositivo**,** además tiene un esquema para emulación de *null modem***.** El TCS binario (*Telephony Control protocol Binary*) es un protocolo que define la señalización de control de llamadas, para el establecimiento y liberación de una conversación o una llamada de datos entre unidades Bluetooth. Además, éste ofrece funcionalidad para intercambiar información de señalización no relacionada con el progreso de llamadas.

La capa de *Audio* es una capa especial, usada sólo para enviar audio sobre Bluetooth. Las transmisiones de audio pueden ser ejecutadas entre una o más unidades usando muchos modelos diferentes. Los datos de audio no pasan a través de la capa L2CAP, pero sí directamente después de abrir un enlace y un establecimiento directo entre dos unidades Bluetooth.

Protocolos Específicos

- **Control de telefonía - Comandos AT.** Bluetooth soporta un número de comandos AT para el control de telefonía a través de emulación de puerto serial (RFCOMM).

- **PPP.** El PPP (*Point to Point Protocol*) es un protocolo orientado a paquetes y por lo tanto debe usar su mecanismo serial para convertir un torrente de paquetes de datos en una corriente de datos seriales. Este protocolo corre sobre RFCOMM para lograr las conexiones punto-a-punto.

- **Protocolos UDP/TCP** - **IP.** Los estándares UDP (*User Datagram Protocol*)/TCP (*Transmission Control Protocol*) e IP (*Internet Protocol*) permiten a las unidades Bluetooth conectarse, por ejemplo a Internet, a través de otras unidades conectadas. Por lo tanto, la unidad puede actuar como un puente para Internet. La configuración TCP/IP/PPP está disponible como un transporte para WAP (*Wireless Application Protocol*).

- **WAP** o Protocolo de Aplicación Inalámbrica. WAP es una especificación de protocolo inalámbrica que trabaja con una amplia variedad de tecnologías de red inalámbricas conectando dispositivos móviles a Internet. Bluetooth puede ser usado como portador para ofrecer el transporte de datos entre el cliente WAP y su servidor de WAP adyacente. Además, las capacidades de red de Bluetooth dan a un cliente WAP posibilidades únicas en cuanto a movilidad comparado con otros portadores WAP.

- **Protocolo OBEX.** OBEX es un protocolo opcional de nivel de aplicación diseñado para permitir a las unidades Bluetooth soportar comunicación infrarroja para intercambiar una gran variedad de datos y comandos. Éste usa un modelo cliente-servidor y es independiente del mecanismo de transporte y del API (*Application Program Interface*) de transporte. OBEX usa RFCOMM como principal capa de transporte.

La Figura 1.2 muestra una comparación del *stack* Bluetooth con el modelo de referencia estándar OSI (*Open System Interconnection*)**,** para *stacks* de protocolos de comunicaciones. A pesar de que Bluetooth no concuerda exactamente con el modelo, esta comparación es muy útil para relacionar las diferentes partes del *stack Bluetooth* con las capas del modelo OSI. Dado que el modelo de referencia es un *stack* ideal y bien particionado, la comparación sirve para resaltar la división de funciones en el *stack* Bluetooth.

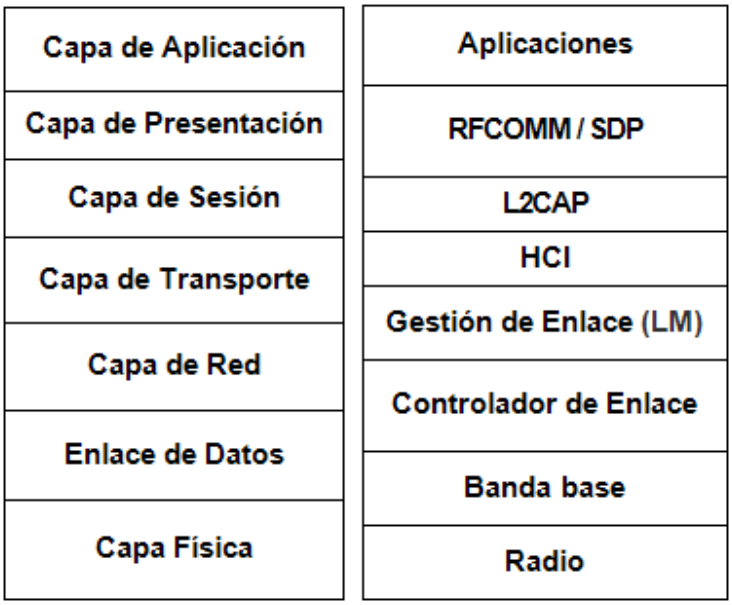


Figura 1.2 Modelo de Referencia OSI y Bluetooth [3].

1.1.1 BANDA BASE

1.1.1.1 Descripción general.

Bluetooth soporta un canal de datos asíncrono de hasta tres canales de voz simultáneos. El canal asíncrono soporta comunicación simétrica y asimétrica. En la comunicación asimétrica pueden ser enviados 723.3 Kbps desde el servidor y 57.6 Kbps hacia el servidor, mientras que en la comunicación simétrica pueden ser enviados 433 Kbps en ambas direcciones.

La tecnología Bluetooth conforma redes *ad-hoc*, es decir que no necesita ninguna infraestructura para su creación y están pensadas para la entrada y salida constante de dispositivos, bajo esta filosofía Bluetooth puede formar dos tipos de red, la más sencilla y fundamental es la *piconet* que consiste en la comunicación entre un maestro y hasta 7 esclavos de forma simultánea y hasta 256 en un estado de relativa inactividad llamado *parked*, cualquier dispositivo puede ser maestro pero teniendo en cuenta que solo debe existir uno por *piconet*. Las *Scatternet* están formadas a partir de de la superposición de varias *piconet*, pues un dispositivo puede ser esclavo de una *piconet* a la vez que es maestro de otra, o puede ser esclavo de varias *piconet* [3]. En la Figura 1.3 se muestra algunos ejemplos de los tipos de redes descritas, en donde la línea punteada que representa el radio de la conexión depende de la potencia del dispositivo, la cual puede variar desde 10m hasta 100m.

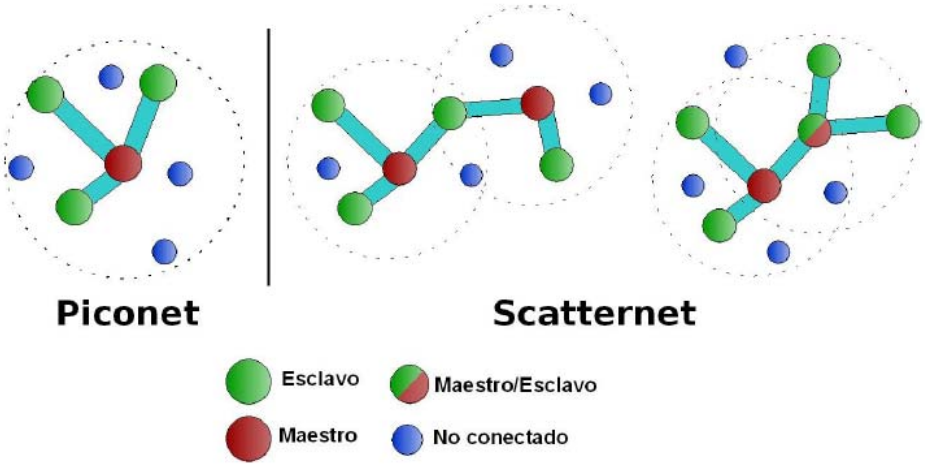


Figura 1.3 Tipos de Redes de Bluetooth [3].

1.1.1.2 Canal físico.

El canal físico contiene 79 frecuencias de radio diferentes, las cuales son accedidas de acuerdo a una secuencia de saltos aleatoria. La rata de saltos estándar es de 1600 *saltos/seg.* El canal está dividido en *timeslots* (ranuras de tiempo), cada *slot* (ranura) corresponde a una frecuencia de salto y tiene una longitud de 625 useg. Cada secuencia de salto en una *piconet* está determinada por la dirección del maestro de la *piconet.* Todos los dispositivos conectados a la *piconet* están sincronizados con el canal en salto y tiempo. En una transmisión, cada paquete debe estar alineado con el inicio de un *slot* y puede tener una duración de hasta cinco *timeslots***.** Durante la transmisión de un paquete la frecuencia es fija. Para evitar fallas en la transmisión, el maestro inicia enviando en los *timeslots* pares y los esclavos en los *timeslots* impares [3]. En la Figura 1.4 se puede observar este esquema de transmisión.

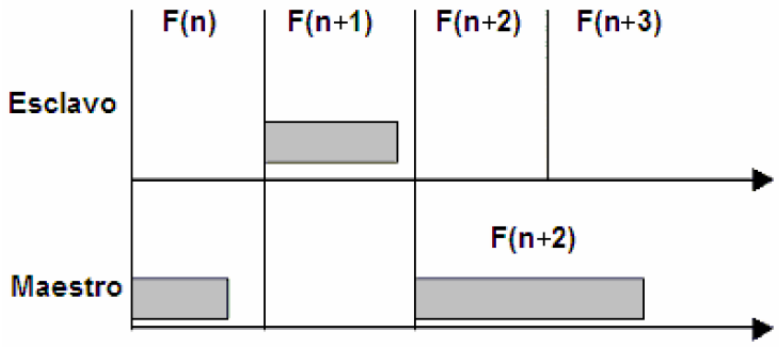


Figura 1.4 Transmisión en una *Piconet* [3].

1.1.1.3 Enlace físico.

La comunicación sobre Bluetooth es perfecta para enlaces SCO o enlaces ACL. El enlace SCO es una conexión simétrica *punto-a-punto* entre el maestro y un esclavo específico. Para lograr la comunicación, el enlace SCO reserva *slots* en intervalos regulares en la iniciación, por esto el enlace puede ser considerado como una conexión de conmutación de circuitos. El enlace ACL es un enlace *punto-a-multipunto* entre el maestro y uno o más esclavos activos en la *piconet.* Este enlace de comunicación es un tipo de conexión de conmutación de paquetes. Todos los paquetes son retransmitidos para asegurar la integridad de los datos. El maestro puede enviar mensajes *broadcast* (de difusión) a todos los esclavos conectados dejando vacía la dirección del paquete, así todos los esclavos leerán el paquete [3].

1.1.1.4 Paquetes.

Los datos enviados sobre el canal de la *piconet* son convertidos en paquetes, éstos son enviados y el receptor los recibe iniciando por el *bit* menos significativo. Como se observa en la Figura 1.5, el formato del paquete general consta de tres campos: código de acceso, cabecera y carga útil [3].

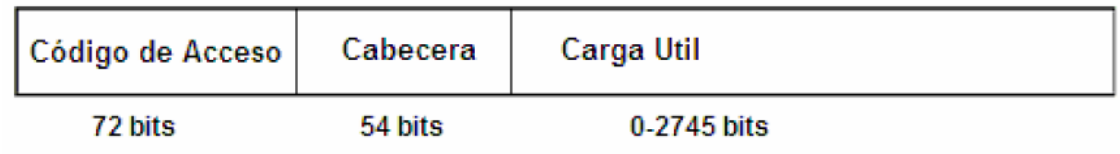


Figura 1.5 Formato del Paquete General [3].

- **Código de acceso.** Es usado para sincronización e identificación. Todos los paquetes comunes que son enviados sobre el canal de la *piconet* están precedidos del mismo código de acceso al canal. Existen tres tipos diferentes de código de acceso:

  > **Código de acceso al canal *(CAC, Channel Access Code)***- Para identificar los paquetes sobre el canal de la *piconet.*

  > **Código de acceso de dispositivo *(DAC, Device Access Code)* -** Para procedimientos de señalización especiales, *paging* (servicio para transferencia de señalización o información en un sentido), entre otros.

  > **Código de Acceso de Búsqueda *(IAC, Inquiry Access Code)*** - llamado *IAC general* cuando se quiere descubrir a otras unidades Bluetooth dentro del rango, o *IAC dedicado* cuando se desea descubrir unidades de un tipo específico.

- **Cabecera de paquete.** Como se observa en la Figura 1.6, la *cabecera de paquete* consta de seis campos:

> **Dirección de transporte lógico *(LT_ADDR, Logical Transport ADDRess)*** - Campo de 3 bits que contiene la dirección de transporte lógica para el paquete. Este campo indica el esclavo destino para paquetes en una transmisión maestro a esclavo e indica la dirección del esclavo fuente para una transmisión esclavo maestro.

> **Tipo** - El campo tipo de 4 bits específica el tipo de paquete que está siendo usado. Existen 16 tipos diferentes de paquetes. La interpretación del campo tipo depende de la dirección de transporte lógico en el paquete *(LT_ADDR)*. Primero, debe ser determinado si el paquete es enviado sobre un enlace lógico SCO, eSCO (e*xtended Synchronous Connection Oriented*) o ACL. Segundo, debe ser determinado si la tasa de datos mejorada ha sido habilitada para el transporte lógico. Adicionalmente este campo determina cuantos *slots* ocupa el paquete transmitido.
En el capítulo 3 se dará una descripción de los tipos paquetes involucrados en la transmisión de datos para brindar soporte a comunicaciones de VoIP.

> **Flujo** - El *bit* de control de flujo de paquetes en conexiones lógicas ACL es usado para notificar al emisor cuándo el buffer del receptor está lleno. Cuando el buffer del receptor está lleno una indicación *stop (flujo=0)* es retornada para detener la transmisión de datos temporalmente. En este estado solo los paquetes ACL son afectados. Cuando el buffer del receptor puede aceptar datos una indicación *go (flujo=1)* es retornada.

> **ARQN –** Un bit de indicación de reconocimiento que es usado para informar a la fuente de una transferencia exitosa del *payload* del paquete, es sacado en base al CRC (*Cyclic Redundancy error Check*) del *payload*.

> **SEQN (*Sequential Numbering*) –** Este campo de un solo bit provee un esquema de enumeramiento secuencial para ordenar el flujo de paquetes de datos.

> **HEC (*Header Error Check*)** – Cada cabecera tiene un HEC para verificar la integridad de ésta.

Direcc. LT	Tipo	Flujo	ARQN	SEQN	HEC

Figura 1.6 Formato de Cabecera del Paquete [3].

- **Carga útil.** La carga útil de un paquete puede ser dividida en dos campos:

> **Campo de Voz** - Consta de datos de voz de longitud fija y existe en paquetes de alta calidad de voz y paquetes combinados de datos-voz. No es necesaria ninguna cabecera de carga útil.

> **Campo de Datos** - Consta de tres partes, cabecera de carga útil, datos de carga útil, y código *CRC.*

1.1.1.5 Corrección de errores.

En una comunicación *Bluetooth* existen varios esquemas diferentes de corrección de errores [3]:

- En la cabecera, cada *bit* es repetido tres veces.

- En la carga útil se usa un esquema de *código Hamming.* Los *bits* de información son agrupados en secuencias de 10 *bits,* éstos son enviados como 15 *bits* y el algoritmo corrige todos los errores de un *bit* y detecta los errores de dos *bits.*

- Para garantizar una recepción correcta, todos los paquetes de datos son retransmitidos hasta que el emisor reciba una confirmación. La confirmación es enviada en la cabecera de los paquetes retornados.

- Los paquetes *broadcast* son paquetes transmitidos desde el maestro a todos los esclavos. No hay posibilidad de usar confirmación para esta comunicación, sin embargo, para incrementar la posibilidad de recibir correctamente un paquete, cada *bit* en el paquete es repetido un número fijo de veces.

- El chequeo de redundancia cíclica *(*CRC*)* se usa para detectar errores en la cabecera. La suma de comprobación CRC está contenida en el campo HEC de la cabecera de paquete. Los chequeos de redundancia cíclica también se aplican sobre la carga útil en la mayoría de los paquetes.

- Para asegurar que no desaparezcan paquetes completos, *Bluetooth* usa números de secuencia. Actualmente sólo se usa un número de secuencia de un *bit.*

1.1.1.6 Transmisión/Recepción.

Como se mencionó en secciones anteriores, el maestro de la *piconet* empieza enviando en *timeslots* pares y el esclavo en los impares. Solamente el último esclavo direccionado está autorizado para enviar en el *timeslot* de los esclavos. Esto no causa problemas ya que el maestro siempre está inicializando todas las conexiones y transmisiones nuevas. Cada esclavo espera las oportunidades de conexión dadas por el maestro. Los paquetes pueden ser más grandes que un *timeslot***,** debido a esto el maestro puede continuar enviando en los *timeslots* impares y viceversa. El sistema de reloj del maestro sincroniza a toda la *piconet.* El maestro nunca ajusta su sistema de reloj durante la existencia de una *piconet,* son los esclavos quienes adaptan sus relojes con un *offset* de tiempo con el fin de igualarse con el reloj del maestro. Este *offset* es actualizado cada vez que es recibido un paquete desde el maestro.

1.1.1.7 Control de Canal.

El control de canal describe cómo se establece el canal de una *piconet* y cómo las unidades pueden ser adicionadas o liberadas en la *piconet.* La dirección del maestro determina la

secuencia de saltos y el código de acceso al canal. La fase de la *piconet* está determinada por el sistema de reloj del maestro. Por definición, la unidad Bluetooth que inicia la conexión representa al maestro.

En Bluetooth, la capa de control de enlace se divide en dos estados principales: *standby y conexión.* Además existen siete sub-estados: *page*, *page scan*, *inquiry*, *inquiry scan*, respuesta de maestro, respuesta de esclavo y respuesta a *inquiry*. Los sub-estados son usados para agregar nuevos esclavos a una *piconet* [3]. Para moverse de un estado a otro se usan comandos de capas más altas o señales internas.

En Bluetooth se define un procedimiento de búsqueda que se usa en aplicaciones donde la dirección del dispositivo de destino es desconocida para la fuente. Esto puede ser usado para descubrir qué otras unidades Bluetooth están dentro del rango. Durante un sub-estado de *inquiry* o búsqueda, la unidad de descubrimiento recoge la dirección del dispositivo y el reloj de todas las unidades que respondan al mensaje de búsqueda, entonces la unidad puede iniciar una conexión con alguna de las unidades descubiertas.

El mensaje de búsqueda difundido por la fuente no contiene información de ella, sin embargo, puede indicar qué clase de dispositivos deberían responder. Una unidad que permita ser descubierta, regularmente entra en un sub-estado de *inquiry scan* para responder a los mensajes de búsqueda.

Existen dos formas de detectar otras unidades. La primera, detecta todas las otras unidades en el rango de cobertura, y la segunda, detecta un tipo específico de unidades. Los esclavos que se encuentran en el sub-estado de *page scan*, escuchan esperando su propio código de acceso de dispositivo. El maestro en el sub-estado *page* activa y conecta a un esclavo.
El maestro trata de capturar al esclavo transmitiendo repetidamente el código de acceso de dispositivo en diferentes canales de salto. Debido a que los relojes del maestro y del esclavo no están sincronizados, el maestro no sabe exactamente cuándo y en qué frecuencia de salto se activará el esclavo.

Después de haber recibido su propio código de acceso de dispositivo, el esclavo transmite un mensaje de respuesta. Este mensaje de respuesta es simplemente el código de acceso de dispositivo del esclavo. Cuando el maestro ha recibido este paquete, envía un paquete de control con información acerca de su reloj, dirección, clase de dispositivo, etc. El esclavo responde con un nuevo mensaje donde envía su dirección. Si el maestro no obtiene esta respuesta en un determinado tiempo, él reenvía el paquete de control. Si el esclavo excede el tiempo de espera, entonces retorna al sub-estado de *page scan*. Si es el maestro quien lo excede, entonces retorna al sub-estado de *page* e informa a las capas superiores.

Cuando se establece la conexión, la comunicación inicia con un paquete de sondeo desde el maestro hacia el esclavo. Como respuesta se envía un nuevo paquete de sondeo y de esta forma se verifica que la secuencia de salto y la sincronización sean correctas. La Figura 1.7 muestra la inicialización de la comunicación sobre el nivel *banda base.*

Cada *transceiver* (receptor-transmisor) Bluetooth tiene una única dirección de dispositivo de 48 bits asignada, la cual está dividida en tres campos: campo LAP (*Lower Address Part*), campo UAP (*Upper Address Part*) y campo NAP (*Non-significant Address Part*) [3]. Los campos LAP *y* UAP forman la parte significativa del código de acceso. En la Figura 1.8 se

puede observar el formato de la dirección para un dispositivo Bluetooth. La dirección del dispositivo es conocida públicamente y puede ser obtenida a través de una rutina *inquiry***.**

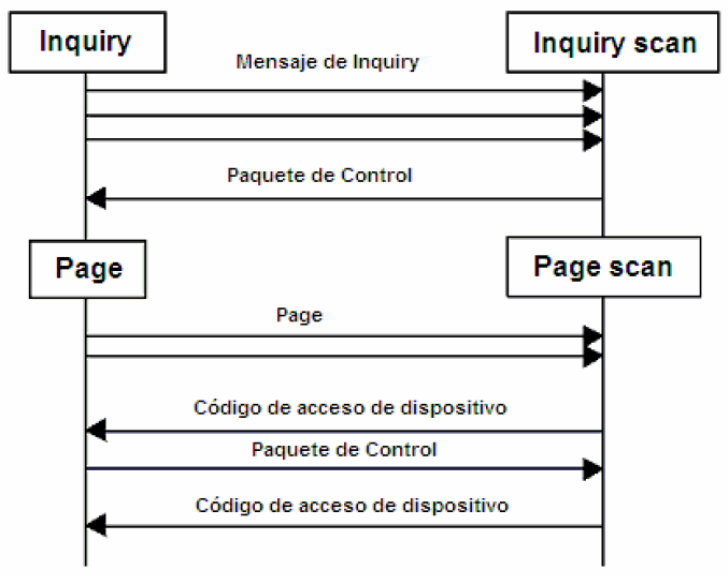


Figura 1.7 Iniciación de una comunicación sobre niveles de Banda Base [3].

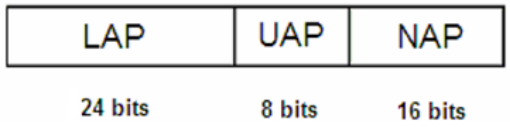


Figura 1.8 Dirección de dispositivo Bluetooth [3].

1.1.1.8 Seguridad en Bluetooth.

Con el fin de brindar protección y confidencialidad a la información, el sistema debe ofrecer medidas de seguridad en las dos capas, la de aplicación y de enlace. Todas las unidades Bluetooth tienen implementadas las mismas rutinas de autenticación y cifrado. En la capa de enlace, estas rutinas constan de cuatro entidades diferentes: una única dirección pública, dos llaves secretas y un número aleatorio el cual es diferente para cada transacción. Solamente es cifrada la carga útil. El código de acceso y la cabecera de paquete nunca son cifrados.

Cada tipo de unidad Bluetooth tiene una rutina de autenticación común. El maestro genera un número aleatorio y lo envía al esclavo, el esclavo usa este número y su propia identidad para calcular el número de autenticación. Luego, este número es enviado al maestro quien hace el mismo cálculo. Si los dos números generados son iguales entonces la autenticación es concedida. El cifrado, frecuentemente está restringida por leyes de varios países. Para evitar estas limitaciones, en Bluetooth la llave de cifrado no es estática, ésta es deducida de la llave de autenticación cada vez que se activa [3].

1.1.2 PROTOCOLO DE GESTIÓN DE ENLACE (LMP)

En el protocolo de gestión de enlace, LMP, se usan mensajes asociados con el establecimiento, seguridad y control. Los mensajes son enviados en la carga útil y no en los mensajes de datos de L2CAP. Los mensajes LMP son separados de los demás por medio de un valor reservado en uno de los campos de la cabecera de carga útil. Todos los mensajes LMP son extraídos e interpretados por la capa LMP del receptor, esto significa que ningún mensaje es enviado a capas superiores. Los mensajes LMP tienen mayor prioridad que los datos de usuario, esto significa que si la gestión de enlace necesita enviar un mensaje, éste no debe ser retrasado por otro tráfico. Solamente las retransmisiones de los paquetes del nivel de banda base pueden retrasar los mensajes LMP. Además, éstos no necesitan rutinas de reconocimiento ya que la capa banda base asegura un enlace confiable [3]. El protocolo de gestión enlace soporta mensajes para:

- Autenticación.
- Paridad.
- Cifrado.
- Temporización y sincronización.
- Versión y características.
- *Switch* para desempeño como maestro o esclavo dependiendo de si el dispositivo es quien inicia (maestro) o no (esclavo) el enlace con otro dispositivo.
- Petición de nombre.
- Desconexión.
- Modo *hold***:** el maestro ordena al esclavo entrar en este estado para ahorro de potencia.
- Modo *sniff***:** para envío de mensajes en time slots específicos.
- Modo *park***:** para que el esclavo permanezca inactivo pero sincronizado en la *piconet.*
- Enlaces SCO.
- Control de paquetes *multi-slot.*
- Supervisión de enlace.

1.1.2.1 Establecimiento de conexión.

Después del procedimiento *paging***,** el maestro debe encuestar al esclavo enviando paquetes de sondeo. El otro lado recibe este mensaje y lo acepta o rechaza, si es aceptado, la comunicación incluyendo las capas superiores están disponibles (ver Figura 1.9).

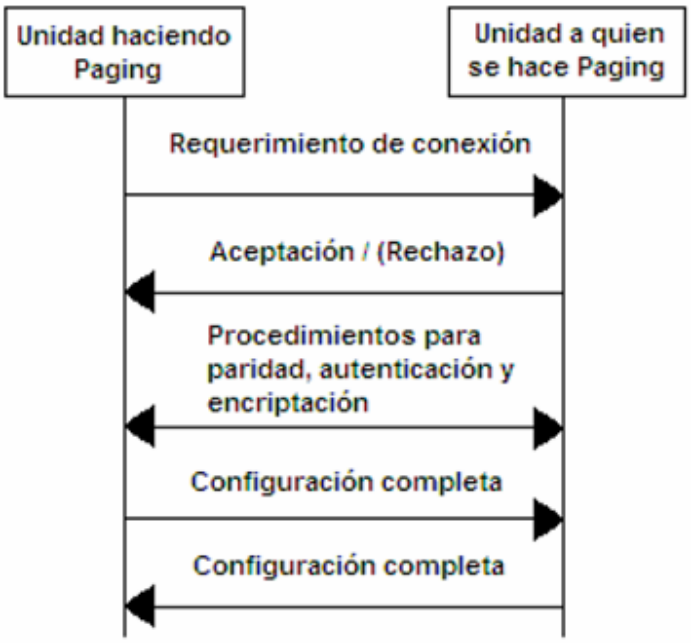


Figura 1.9 Establecimiento de la Conexión [3].

1.1.3 PROTOCOLO DE CONTROL Y ADAPTACIÓN DE ENLACE LÓGICO (L2CAP)

L2CAP se encuentra sobre el *protocolo de gestión de enlace (*LMP*)* y reside en la capa de enlace de datos. L2CAP permite a protocolos de niveles superiores y a aplicaciones la transmisión y recepción de paquetes de datos L2CAP de hasta 64 kilobytes, con capacidad de multiplexación de protocolo, operación de segmentación y reensamblado, y abstracción de grupos. Para cumplir sus funciones, L2CAP espera que la *banda base* suministre paquetes de datos en *full dúplex***,** que realice el chequeo de integridad de los datos y que reenvíe los datos hasta que hayan sido reconocidos satisfactoriamente. Las capas superiores que se comunican con L2CAP son por ejemplo el *protocolo de descubrimiento de servicio (*SDP*), el* RFCOMM *y el control de telefonía (*TCS*)* [3].

1.1.3.1 Canales.

L2CAP está basado en el concepto de canales. Se asocia un identificador de canal, CID (*Channel Identifier*), a cada uno de los *endpoints* de un canal L2CAP. Los CIDs están divididos en dos grupos, uno con identificadores reservados para funciones L2CAP y otro con identificadores libres para implementaciones particulares. Los canales de datos orientados a la conexión representan una conexión entre dos dispositivos, donde un CID identifica cada *endpoint* del canal. Los canales no orientados a la conexión limitan el flujo de datos a una sola dirección. La señalización de canal es un ejemplo de un canal reservado.

Este canal es usado para crear y establecer canales de datos orientados a la conexión y para negociar cambios en las características de esos canales.

1.1.3.2 Operaciones entre Capas.

Las implementaciones L2CAP deben transferir datos entre protocolos de capas superiores e inferiores.

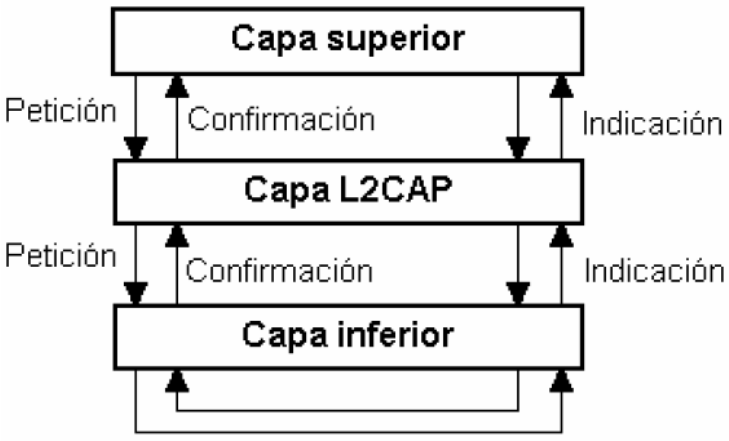


Figura 1.10 Arquitectura L2CAP [3].

Cada implementación debe soportar un grupo de comandos de señalización, además, debe ser capaz de aceptar ciertos tipos de eventos de capas inferiores y generar eventos para capas superiores. En la Figura 1.10 se muestra esta arquitectura.

1.1.3.3 Segmentación y Reensamblado.

Los paquetes de datos definidos por el protocolo banda base están limitados en tamaño. Los paquetes L2CAP grandes deben ser segmentados en varios paquetes banda base más pequeños antes de transmitirse y luego deben ser enviados a la gestión de enlace. En el receptor los pequeños paquetes recibidos de la banda base son reensamblados en paquetes L2CAP más grandes. Varios paquetes banda base recibidos pueden ser reensamblados en un solo paquete L2CAP seguido de un simple chequeo de integridad. La segmentación y reensamblado, SAR (*Segmentation and Reassembly*), funcionalmente es absolutamente necesaria para soportar protocolos usando paquetes más grandes que los soportados por la banda base. La Figura 1.11 muestra la segmentación L2CAP.

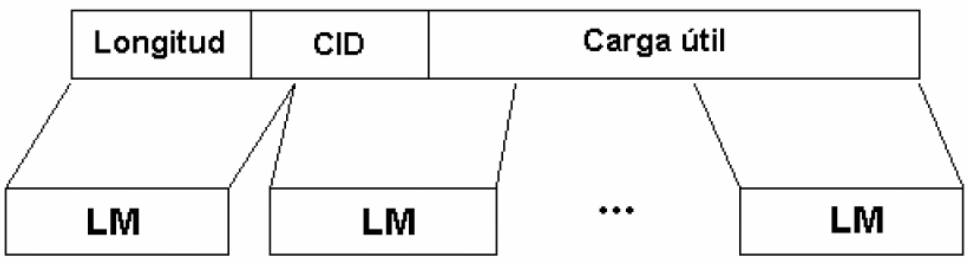


Figura 1.11 Segmentación L2CAP [3].

1.1.3.4 Eventos.

Todos los mensajes y *timeouts* que entran en la capa L2CAP, son llamados eventos. Los eventos se encuentran divididos en cinco categorías: indicaciones y confirmaciones de capas inferiores, peticiones de señal y respuestas de capas L2CAP, datos de capas L2CAP, peticiones y respuestas de capas superiores, y eventos causados por expiraciones de tiempo.

1.1.3.5 Acciones.

Todos los mensajes y *timeouts* enviados desde la capa L2CAP son llamados acciones (en el lado del receptor estas acciones son llamadas eventos). Las acciones se encuentran divididas en cinco categorías: peticiones y respuestas a capas inferiores, peticiones y respuestas a capas L2CAP, datos a capas L2CAP, indicaciones a capas superiores, y configuración de *timers***.**

1.1.3.6 Formato del paquete de datos.

L2CAP está basado en paquetes pero sigue un modelo de comunicación basado en canales. Un canal representa un flujo de datos entre entidades L2CAP en dispositivos remotos. Los canales pueden ser o no orientados a la conexión. Como se puede observar en la Figura 1.12, los paquetes de canal orientado a la conexión están divididos en tres campos: longitud de la información, identificador de canal, e información.

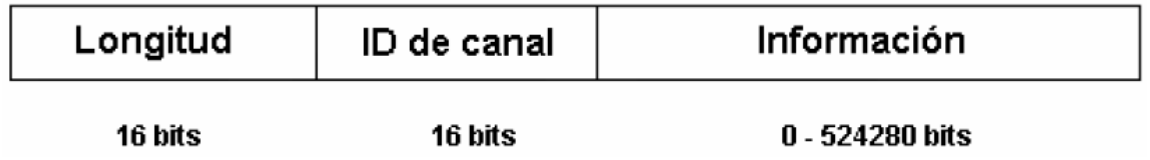


Figura 1.12 Paquete L2CAP [3].

Los paquetes de canal de datos no orientados a la conexión son iguales a los paquetes orientados a la conexión pero adicionalmente incluyen un campo con información multiplexada de protocolo y servicio.

1.1.3.7 Calidad de servicio (QoS, Quality of Service).

La capa L2CAP transporta la información de calidad de servicio (QoS) a través de los canales y brinda control de admisión para evitar que canales adicionales violen contratos de calidad de servicio existentes.

Algunos esclavos pueden requerir un alto rendimiento o una respuesta rápida. Antes de que un esclavo con grandes peticiones sea conectado a una *piconet,* el esclavo trata de obtener una garantía a sus demandas. Puede solicitar una determinada rata de transmisión, tamaño del buffer de tráfico, ancho de banda, tiempo de recuperación de datos, etc. Por lo tanto, antes de que el maestro conecte a un nuevo esclavo o actualice la configuración de calidad, debe chequear si posee *timeslots* y otros recursos libres.

1.1.4 PROTOCOLO DE DESCUBRIMIENTO DE SERVICIO (SDP)

El protocolo de descubrimiento de servicio, SDP, brinda a las aplicaciones recursos para descubrir qué servicios están disponibles y determinar las características de dichos servicios.

1.1.4.1 Descripción General.

Un servicio es una entidad que puede brindar información, ejecutar una acción o controlar un recurso a nombre de otra entidad. El SDP ofrece a los clientes la facilidad de averiguar sobre servicios que sean requeridos, basándose en la clase de servicio o propiedades específicas de estos servicios. Para hacer más fácil la búsqueda, el SDP la habilita sin un previo conocimiento de las características específicas de los servicios. Las unidades Bluetooth que usan el SDP pueden ser vistas como un servidor y un cliente. El servidor posee los servicios y el cliente es quien desea acceder a ellos. En el SDP esto es posible ya que el cliente envía una petición al servidor y éste responde con un mensaje. El SDP solamente soporta el descubrimiento del servicio, no la llamada del servicio [3].

1.1.4.2 Registros de servicio.

Los registros de servicio contienen propiedades que describen un servicio determinado. Cada propiedad de un registro de servicio consta de dos partes, un identificador de propiedad y un valor de propiedad. El identificador de propiedad es un número único de 16 bits que distingue cada propiedad de servicio de otro dentro de un registro. El valor de propiedad es un campo de longitud variable que contiene la información.

1.1.4.3 El protocolo.

El protocolo de descubrimiento de servicio (SDP) usa un modelo petición/respuesta. A continuación se detallas el tipo de mensajes utilizados por el protocolo:

- **Petición de búsqueda de servicio:** se genera por el cliente para localizar registros de servicio que concuerden con un patrón de búsqueda dado como parámetro. Aquí el servidor examina los registros en su base de datos y responde con una *respuesta a búsqueda de* servicio*.*

- **Respuesta a búsqueda de servicio:** se genera por el servidor después de recibir una *petición de búsqueda de servicio* válida.

- **Petición de propiedad de servicio:** una vez el cliente ya ha recibido los servicios deseados, puede obtener mayor información de uno de ellos dando como parámetros el registro de servicio y una lista de propiedades deseadas.

- **Respuesta a propiedad de servicio:** El SDP genera una respuesta a una *petición de propiedad de servicio.* Ésta contiene una lista de propiedades del registro requerido.

- **Petición de búsqueda y propiedad de servicio:** se suministran un patrón de servicio con servicios deseados y una lista de propiedades deseadas que concuerden con la búsqueda.

- **Respuesta de búsqueda y propiedad de servicio:** como resultado se puede obtener una lista de servicios que concuerden con un patrón dado y las propiedades deseadas de estos servicios.

1.1.5 RFCOMM

El protocolo RFCOMM brinda emulación de puertos seriales sobre el protocolo L2CAP. La capa RFCOMM es una simple capa de transporte provista adicionalmente de emulación de circuitos de puerto serial RS-232. El protocolo RFCOMM soporta hasta 60 puertos emulados simultáneamente. Dos unidades Bluetooth que usen RFCOMM en su comunicación pueden abrir varios puertos seriales emulados, los cuales son multiplexados entre sí. La Figura 1.13 muestra el esquema de emulación para varios puertos seriales.

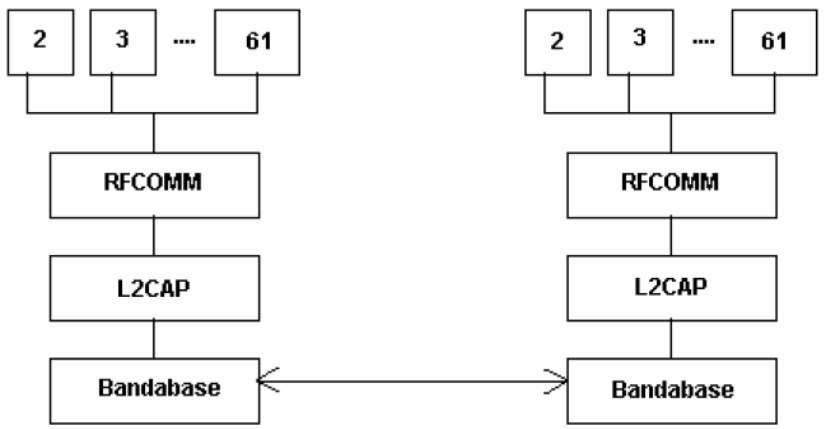


Figura 1.13 Varios puertos seriales emulados mediante RFCOMM [3].

Muchas aplicaciones hacen uso de puertos seriales. El RFCOMM está orientado a hacer más flexibles estos dispositivos, soportando fácil adaptación de comunicación Bluetooth. Un ejemplo de una aplicación de comunicación serial es el protocolo *punto-a-punto (*PPP*).* El RFCOMM tiene construido un esquema para emulación de *null modem* y usa a L2CAP para cumplir con el control de flujo requerido por alguna aplicación [3].

1.1.6 PERFILES BLUETOOTH

El estándar Bluetooth fue creado para ser usado por un gran número de fabricantes e implementado en áreas ilimitadas. Para asegurar que todos los dispositivos que usen Bluetooth sean compatibles entre sí son necesarios esquemas estándar de comunicación en las principales áreas. Para evitar diferentes interpretaciones del estándar Bluetooth acerca de cómo un tipo específico de aplicación debería ser implementado, el Bluetooth SIG (*Special Interest Group*), ha definido modelos de usuario y perfiles de protocolo. Un perfil define una selección de mensajes y procedimientos de las especificaciones Bluetooth y ofrece una descripción clara de la interfaz de aire para servicios específicos. Un perfil puede ser descrito como una "rebanada" completa del *stack* de protocolo.

Existen cuatro perfiles generales definidos, en los cuales están basados directamente algunos de los modelos de usuario más importantes y sus perfiles. Estos cuatro modelos son Perfil Genérico de Acceso (GAP, *General Access Profile*), Perfil de Puerto Serial, Perfil de Aplicación de Descubrimiento de Servicio (SDAP, *Service Discovery Application Profile*) y Perfil Genérico de Intercambio de Objetos (GOEP, *Generic Object Exchange Profile*).

A continuación se hace una breve descripción de estos y algunos otros perfiles *Bluetooth.* La Figura 1.14 muestra el esquema de los perfiles Bluetooth. En ella se puede observar la jerarquía de los perfiles, como por ejemplo que todos los perfiles están contenidos en el *Perfil Genérico de Acceso (*GAP*).*

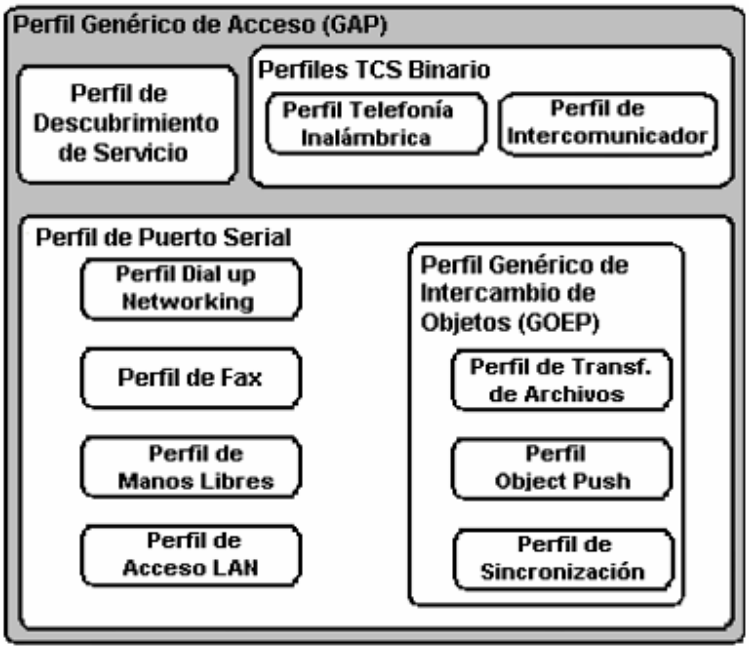


Figura 1.14 Los Perfiles Bluetooth [4].

1.1.6.1 Perfil Genérico de Acceso (GAP).

Este perfil define los procedimientos generales para el descubrimiento y establecimiento de conexión entre dispositivos Bluetooth. El GAP maneja el descubrimiento y establecimiento entre unidades que no están conectadas y asegura que cualquier par de unidades *Bluetooth,* sin importar su fabricante o aplicación, puedan intercambiar información a través de *Bluetooth* para descubrir qué tipo de aplicaciones soportan las unidades.

1.1.6.2 Perfil de Puerto Serial.

Este perfil define los requerimientos para dispositivos Bluetooth, necesarios para establecer una conexión de cable serial emulada usando RFCOMM entre dos dispositivos similares. Este perfil solamente requiere soporte para paquetes de un slot**.** Esto significa que pueden ser usadas ratas de datos de hasta 128 Kbps. El soporte para ratas más altas es opcional. RFCOMM es usado para transportar los datos de usuario, señales de control de modem y comandos de configuración. El perfil de puerto serial es dependiente del GAP.

1.1.6.3 Perfil de Aplicación de Descubrimiento de Servicio (SDAP).

Este perfil define los protocolos y procedimientos para una aplicación en un dispositivo *Bluetooth* donde se desea descubrir y recuperar información relacionada con servicios localizados en otros dispositivos. El SDAP es dependiente del GAP.

1.1.6.4 Perfil Genérico de Intercambio de Objetos (GOEP).

Este perfil define protocolos y procedimientos usados por aplicaciones para ofrecer características de intercambio de objetos. Los usos pueden ser, por ejemplo, sincronización, transferencia de archivos o modelo *Object Push***.** Los dispositivos más comunes que usan este modelo son agendas electrónicas, *PDAs,* teléfonos celulares y teléfonos móviles. El GOEP es dependiente del *perfil de puerto serial.*

1.1.6.5 Perfil de Telefonía Inalámbrica.

Este perfil define como un teléfono móvil puede ser usado para acceder a un servicio de telefonía de red fija a través de una estación base. Es usado para telefonía inalámbrica de hogares u oficinas pequeñas. El perfil incluye llamadas a través de una estación base, haciendo llamadas de intercomunicación directa entre dos terminales y accediendo adicionalmente a redes externas. Es usado por dispositivos que implementan el llamado "teléfono 3-en-1".

1.1.6.6 Perfil de Intercomunicador.

Este perfil define usos de teléfonos móviles los cuales establecen enlaces de conversación directa entre dos dispositivos. El enlace directo es establecido usando señalización de telefonía sobre Bluetooth*.* Los teléfonos móviles que usan enlaces directos funcionan como *walkie-talkies***.**

1.1.6.7 Perfil de Manos Libres.

Este perfil define los requerimientos, para dispositivos Bluetooth*,* necesarios para soportar el uso de manos libres. En este caso el dispositivo puede ser usado como unidad de audio inalámbrico de entrada/salida. El perfil soporta comunicación segura y no segura.

1.1.6.8 Perfil Dial-up Networking.

Este perfil define los protocolos y procedimientos que deben ser usados por dispositivos que implementen el uso del modelo llamado *Puente Internet.* Este perfil es aplicado cuando un teléfono celular o *modem* es usado como un *modem* inalámbrico.

1.1.6.9 Perfil de Fax.

Este perfil define los protocolos y procedimientos que deben ser usados por dispositivos que implementen el uso de fax. En el perfil un teléfono celular puede ser usado como un fax inalámbrico.

1.1.6.10 Perfil de Acceso LAN.

Este perfil define el acceso a una red de área local, LAN (*Local Area Network*), usando el protocolo punto-a-punto, PPP, sobre RFCOMM. PPP es ampliamente usado para lograr acceder a redes soportando varios protocolos de red. El perfil soporta acceso LAN para un dispositivo Bluetooth sencillo, acceso LAN para varios dispositivos Bluetooth y PC-a-PC (usando interconexión PPP con emulación de cable serial).

1.1.6.11 Perfil Object Push.

Este perfil define protocolos y procedimientos usados en el modelo *object push***.** Este perfil usa el GOEP*.* En el modelo *object push* hay procedimientos para introducir en el *inbox***,** sacar e intercambiar objetos con otro dispositivo *Bluetooth.*

1.1.6.12 Perfil de Transferencia de Archivos.

Este perfil define protocolos y procedimientos usados en el modelo de transferencia de archivos. El perfil usa el GOEP*.* En el modelo de transferencia de archivos hay procedimientos para chequear un grupo de objetos de otro dispositivo *Bluetooth,* transferir objetos entre dos dispositivos y manipular objetos de otro dispositivo. Los objetos podrían ser archivos o fólderes de un grupo de objetos tal como un sistema de archivos.

1.1.6.13 Perfil de Sincronización.

Este perfil define protocolos y procedimientos usados en el modelo de sincronización. Éste usa el GOEP*.* El modelo soporta intercambios de información, por ejemplo para sincronizar calendarios de diferentes dispositivos [4].

1.2 DESCRIPCIÓN GENERAL DE LA TECNOLOGÍA ETHERNET

1.2.1 ASPECTOS GENERALES

El estándar *Ethernet* IEEE (*Institute of Electrical and Electronics Engineers*) 802.3 es un estándar ampliamente usado en redes LAN. Existen muchas aplicaciones software que son desarrolladas para ser usadas en este tipo de redes, adicionalmente al gran conocimiento acerca de estas redes que se encuentra en el dominio de los administradores y desarrolladores de redes. Otra de las ventajas mostradas por estas redes, es la completa compatibilidad y uso generalizado del *stack* TCP/IP sobre *Ethernet*. Esta sección describe el estándar *Ethernet* brevemente para tener una visión general del panorama de interoperabilidad de redes.

Este estándar internacional está concebido para abarcar varios tipos de medios y técnicas para una variedad de tasas de señal. *Ethernet* se encuentra ubicado dentro del nivel de acceso a red del stack de protocolos TCP/IP como se ilustra en la Figura 1.15.

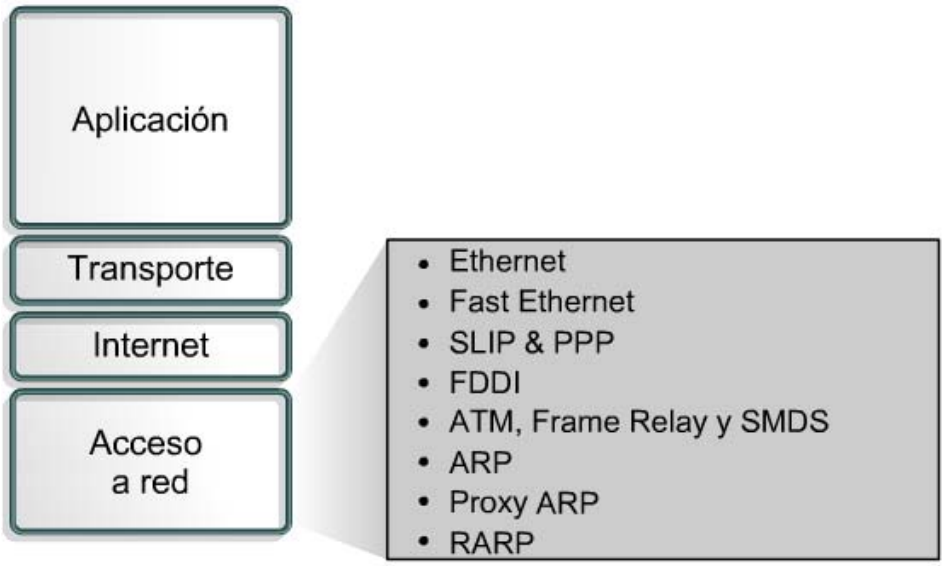


Figura 1.15 Protocolos de acceso de Red del *stack* de protocolos TCP/IP [5].

1.2.2 MÉTODO DE CONTROL DE ACCESO AL MEDIO

El método de control de acceso al medio (MAC, *Medium Access Control*) usado en *Ethernet* es llamado Acceso Múltiple de Detección de Portadora/Detección de Colisión (CSMA/CD). El esquema MAC es bastante simple y fue desarrollado en base a un viejo método conocido como ALOHA. En la Figura 1.16, se muestra una típica configuración *Ethernet*, conocida como bus LAN. El estándar soporta muchas otras topologías pero ésta es la más simple y acorde para el propósito de explicar el método CSMA/CD (*Carrier Sense Multiple Access / Collision Detection*). Todas las estaciones comparten el medio que interconecta físicamente los elementos de la red el cual consiste usualmente de cable coaxial o par trenzado de cobre. Cuando una estación A desea enviar un paquete, ésta "escucha" al medio para determinar si éste se encuentra libre, en tal caso la estación empieza la transmisión. Si la estación B, C o la D tienen datos para ser enviados éstas tendrán que esperar durante la transmisión de la estación A. Una colisión ocurrirá cuando dos o más estaciones empiecen a transmitir casi simultáneamente. Cuando una colisión es detectada, las estaciones implicadas abortan sus transmisiones y las retoman luego de un tiempo determinado aleatoriamente y entonces retransmiten sus tramas utilizando el procedimiento mencionado. Las estaciones que no se encuentran transmitiendo escuchan el tráfico de la red para determinar aquellos paquetes que se encuentran dirigidos hacia ellos.

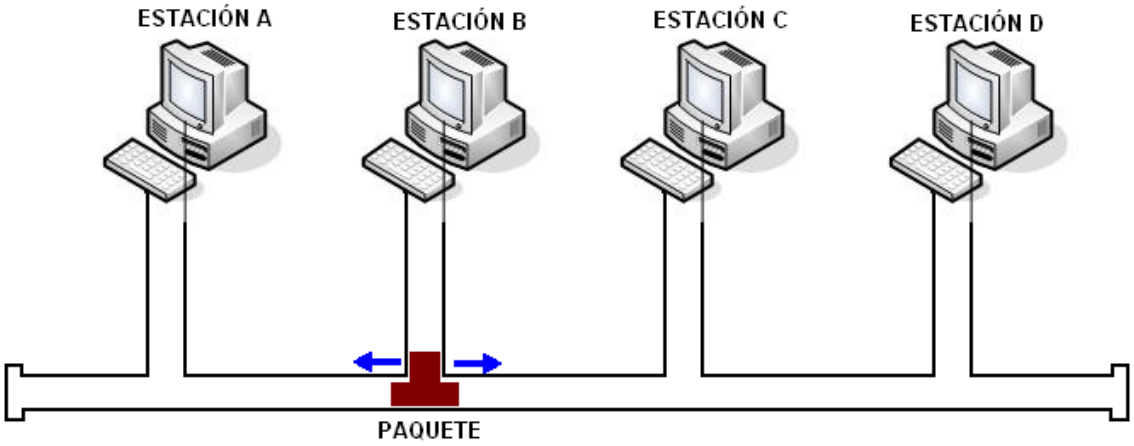


Figura 1.16 LAN *Ethernet* [5].

1.2.3 DIRECCIONAMIENTO

En una red *Ethernet*, todos los dispositivos tienen una dirección única de 6 bytes, la cual es llamada dirección MAC. Para poder asegurar la unicidad de todas y cada una de las direcciones MAC, éstas son administradas por la IEEE. Adicionalmente a las direcciones regulares de los dispositivos, también existen un tipo de direcciones especiales llamadas direcciones *multicast*. Todas las estaciones en la red reciben las tramas con una dirección *multicast* en su campo de dirección de destino. En el stack de protocolos TCP/IP esta característica es usada para el protocolo ARP (*Address Resolution Protocol*), el cual mapea direcciones físicas a direcciones IP.

1.2.4 FORMATO DE TRAMA

El formato de trama especificado en el estándar IEEE 802.3 se muestra en Figura 1.17. El primer campo es la dirección de destino seguido por la dirección de origen, ambos con tamaño de 6 bytes. El tercer campo es el de tipo/longitud el cual dependiendo de su valor numérico debe ser interpretado dependiendo de la longitud de los datos de carga o del tipo de protocolo de red. El campo de carga útil o *payload* que siguen al campo de tipo/longitud puede tener un tamaño máximo de 1500 bytes. Finalmente están el preámbulo y los campos de verificación de secuencia de trama, los cuales son usados a nivel físico.

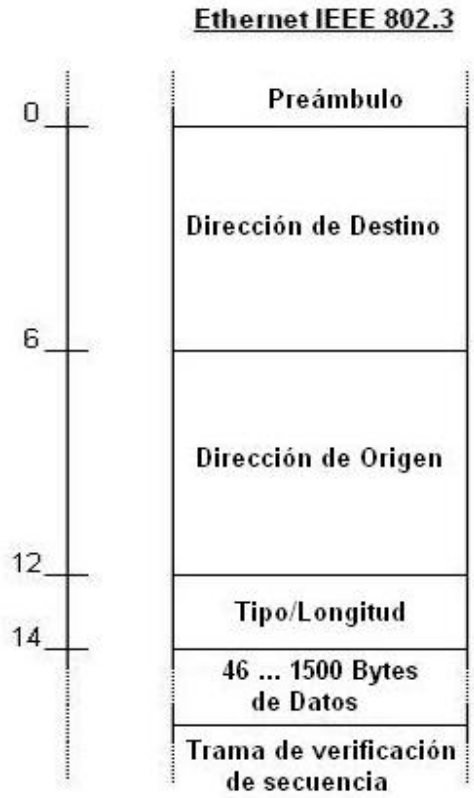


Figura 1.17 Formato de trama *Ethernet* [5].

Debido al funcionamiento de TCP/IP la información pasa a través de un proceso de empaquetamiento que da como resultado paquetes que contienen encabezados de los protocolos que participan en el proceso de comunicación. De ahí que la forma final de la trama obtenida utilizando *Ethernet* como interfaz de red sea la mostrada en la Figura 1.18. Una descripción más detallada del proceso de empaquetamiento de información de aplicaciones específicas tales como VoIP (*Voice over Internet Protocol*) es dada en el capítulo 2 del presente documento.

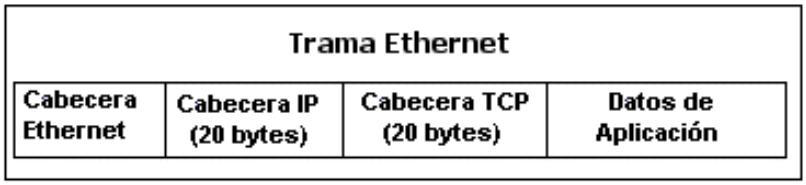


Figura 1.18 Trama final en sistemas TCP/IP [5].

1.3 PRIMERA APROXIMACIÓN A UN ESCENARIO DE INTEROPERABILIDAD

En las anteriores secciones se detalló los aspectos generales y la funcionalidad de las tecnologías Bluetooth y *Ethernet*, sobre las cuales reside el enfoque del presente proyecto de investigación.

Para lograr un panorama de interoperabilidad y poder definir una arquitectura que soporte servicios de Voz sobre IP en tiempo real que se adapte a este panorama, se debe establecer las similitudes y diferencias entre ambas tecnologías.

Inicialmente se debe considerar la inminente diferencia en la naturaleza del medio de propagación de las señales que representan el flujo de datos en ambas tecnologías. Por una parte la utilización de un medio cableado para la conectividad en la tecnología *Ethernet* comparado con la comunicación vía interfaz aire propia de Bluetooth, presenta un desafío de conectividad. Por lo tanto es necesaria la definición de un *Puente* o elemento de interconexión que permita abarcar la brecha existente entre ambas tecnologías en lo que se refiere a medios de propagación diferentes.

Por otra parte, el direccionamiento utilizado por cada una de las tecnologías es un factor que influye sobre la forma de interconexión y reenvío exitosos de paquetes hacia su destino específico, así que debe ser muy tenido en cuenta a la hora de la definición del elemento de conectividad.

Por último las velocidades de transmisión propias de cada tecnología han de ser evaluadas; y contrastadas con las requeridas en sistemas de transmisión de paquetes de voz sobre redes de datos, es decir definir el ancho de banda necesario y suficiente para la prestación de servicios de voz en ambientes de alta movilidad.

CAPÍTULO 2 DEFINICIÓN Y EVALUACIÓN DE LOS CRITERIOS Y PARÁMETROS DE INTEROPERABILIDAD SOBRE AMBIENTES DE PRESTACIÓN DE SERVICIOS DE VOZ

En este capítulo se establecerán una serie de criterios y parámetros de interoperabilidad para la prestación de servicios de voz en base a un estudio de las características de VoIP y perfiles de interconexión propios de Bluetooth que permitan prestar servicios con requerimientos de alta movilidad y rápida localización de usuarios.

2.1 DEFINICIÓN DE LA CONECTIVIDAD DE LOS USUARIOS MÓVILES

Lo primero que se debe definir es la forma como los usuarios accederán al servicio y las tecnologías que soportan dicha conectividad. Para esto se debe especificar y evaluar las capacidades brindadas por los componentes individuales de la tecnología Bluetooth por separado.

2.1.1 ESTABLECIMIENTO DE CONEXIONES BLUETOOTH CON CAPACIDADES IP.

Los clientes móviles deberían escanear regularmente dispositivos Bluetooth los cuales ofrezcan sus servicios como *gateways* IP. Actualmente existen dos maneras de llevar a cabo IP sobre Bluetooth. Una de ellas es crear una línea serial virtual, llamada RFCOMM y entonces crear una conexión PPP sobre esta. La forma más reciente es llamada comúnmente como PAN (descrita en el perfil PAN, *Personal Area Networking* de Bluetooth) y usa directamente la capa Bluetooth -a través del protocolo BNEP (*Bluetooth Network Encapsulation Protocol*)-, lo cual resulta en un encabezado más bajo por paquete final; esto conduce a pensar que ésta es la forma más efectiva para crear enlaces Bluetooth con capacidades IP. La inclusión del perfil PAN dentro de la especificación Bluetooth, ha hecho que algunas de las implementaciones del *stack* Bluetooth incorporen un soporte funcional del perfil PAN.

A continuación se da una breve descripción de las implementaciones más comunes del *stack* Bluetooth en Linux –tales como *BlueZ* el cual es la pila oficial de Linux y donde existe una mayor comunidad desarrolladora, por lo que actualmente es la mejor opción a seguir para trabajar con Bluetooth en este sistema operativo. Está integrada en el núcleo por lo que el despliegue de aplicaciones basadas en *BlueZ* es más sencillo.

2.1.1.1 BlueZ [6]

BlueZ es una pila de protocolos que comenzó su desarrollo como código abierto en Mayo de 2001 y que, un mes después de su publicación por parte de *Max Krasnyansky* en *Qualcomm*, fue adoptada por *Linus Torvalds* como la pila Bluetooth de Linux, lo que significó que desde el *kernel* 2.4.6, *Linux* pasó a tener su propia pila Bluetooth. *Marcel Holtmann* se está encargando del mantenimiento de la pila en el *kernel* 2.6 de Linux.

En Abril de 2005 recibió la Certificación Bluetooth del SIG de Bluetooth, y ya existen cinco productos bajo esta certificación.

Actualmente soporta los perfiles:

- HIDP (*Human Interface Device Profile*)
- A2DP (*Advanced Audio Distribution Profile*)

  Utiliza el protocolo de transporte de distribución Audio/Video (AVDTP, *Audio/Video Distribution Transport Protocol*) por debajo para llevar a cabo la comunicación.

- HCRP (*Hardcopy Cable Replacement Profile*)
- SDAP
- GAP
- DUN
- FAX
- PAN (Contiene un demonio PAN completamente funcional, llamado *pand*.)
- PUSH
- SYNC
- FTP
- Acceso LAN
- CIP
- HCRP

2.1.1.2 Axis OpenBT [6]

Fue la primera implementación para Linux, vio la luz en Abril de 1999. Creada originalmente como parte del soporte Linux del punto de acceso de la compañía AXIS, los perfiles Bluetooth que soporta esta pila son:

• Acceso LAN
• Dial Up
• Comunicaciones Serie

El 14 de Abril de 2005 *Anders Torbjorn* Johansson anunció el final del desarrollo de esta pila y redirigió a todos los que aún la usaban a *BlueZ*.

2.1.1.3 Nokia Affix Bluetooth Stack [6]

Es la pila alternativa a *BlueZ* que tiene algún viso de poder hacerle la competencia. Se comenzó a desarrollar antes de que se publicara *BlueZ* lo que puede explicar su existencia a pesar de *BlueZ*. Dispone de una documentación completa tanto para usar la pila como para programar aplicaciones sobre ella.

Soporta los perfiles Bluetooth:

- GAP
- SDP
- *Serial Port Profile*
- *DialUp Networking Profile*
- Acceso LAN
- OBEX
- PAN
- FAX
- HIDP

2.2 CONCEPTOS BÁSICOS PARA LA IMPLEMENTACIÓN DE UNA RED DE ÁREA PERSONAL BLUETOOTH

La especificación Bluetooth define perfiles o esquemas estándar de los escenarios de desarrollo para las aplicaciones Bluetooth, como fueron mencionados anteriormente. Uno de estos escenarios corresponde al perfil de red de área personal o perfil PAN. El perfil PAN brinda capacidades de red a los dispositivos Bluetooth para lo cual utiliza el Protocolo de Encapsulamiento de Red BNEP, de gran importancia ya que encapsula los paquetes provenientes de varios protocolos de red (IPv4, IPv6 (*Internet Protocol version 6*) e IPX (*Internet Packet Exchange*) entre otros) y los transporta directamente sobre la capa de protocolo L2CAP de Bluetooth, haciendo posible que la red Bluetooth se comporte y forme parte de una red TCP/IP logrando un escenario de interoperabilidad con redes cableadas perteneciente al Nivel de transporte del red del *stack*.

2.2.1 PROTOCOLO DE ENCAPSULAMIENTO DE RED (BNEP)

El protocolo de encapsulamiento de red de Bluetooth encapsula los paquetes de los protocolos de red más utilizados, para ser transportados sobre la capa L2CAP. BNEP soporta los mismos protocolos de red soportados por el estándar IEEE 802.3 para encapsulamiento *Ethernet (*IPv4, IPv6 , IPX entre otros*).* BNEP requiere además un formato de encapsulamiento que optimice los bits de cabecera de tal manera que se ofrezca un manejo eficiente del ancho de banda. Los siguientes apartes son tomados de la especificación del protocolo BNEP publicada por el Bluetooth SIG [6].

2.2.1.1 Consideraciones

1. Este protocolo está implementado usando canales *L2CAP* orientados a conexión.
2. Bluetooth se considera un medio de transmisión al mismo nivel OSI de *Ethernet, Token Ring,* ATM (*Asynchronous Transfer Mode*), etc.
3. Se considera a L2CAP como la capa de control de acceso al medio MAC de *Bluetooth.*
4. BNEP especifica una mínima MTU (*Maximum Transfer Unit*) para L2CAP de1691 bytes[4].

[4] La mínima MTU (Máxima Unidad de Transmisión) de 1691 se seleccionó con base en el máximo paquete de carga útil de Ethernet

5. Se deben aplicar a *Bluetooth* las reglas de conectividad y las topologías de red definidas por el estándar IEEE 802.3 [9] [10]
6. El espacio de dirección Bluetooth BD_ADDR (*Bluetooth Device Address*) [3] es administrado por IEEE, y es asignado desde el espacio de dirección de *Ethernet*. De esta manera es posible construir un punto de acceso de red Bluetooth como un puente entre los dispositivos Bluetooth y una red *Ethernet*.

La Figura 2.1 muestra la ubicación del BNEP dentro de la torre de protocolos de Bluetooth.

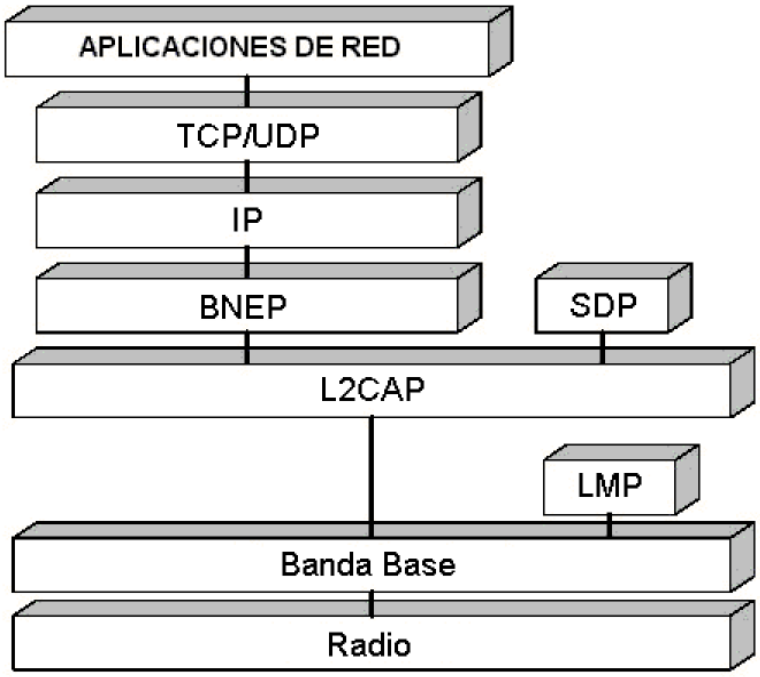


Figura 2.1 Ubicación de BNEP dentro del *Stack* de Protocolos Bluetooth [8].

2.2.1.2 Orden de los Bytes y Valores numéricos

Todos los valores contenidos en esta sección se presentan en notación hexadecimal. Los campos que contienen múltiples bytes (bits) se muestran con los bytes (bits) más significativos hacia la izquierda y los menos significativos hacia la derecha. Los bytes de los encabezados de BNEP se disponen en el formato estándar de red *Bíg Endían,* en donde los bytes más significativos se transmiten antes que los menos significativos. El byte *O* es el más significativo.

(1500 bytes más 15 bytes de encabezado de BNEP y otros de extensión). 1691=5*339 (Tamaño de un DH5) - 4 (Encabezado de L2CAP).

2.2.1.3 Encapsulamiento de Paquetes

La Figura 2.2 muestra la manera como BNEP remueve los encabezados de un paquete *Ethernet y* los reemplaza por un encabezado BNEP*.* El paquete resultante (Carga útil de *Ethernet* y el encabezado de BNEP*)* es encapsulado en un paquete L2CAP y enviado sobre Bluetooth*.*

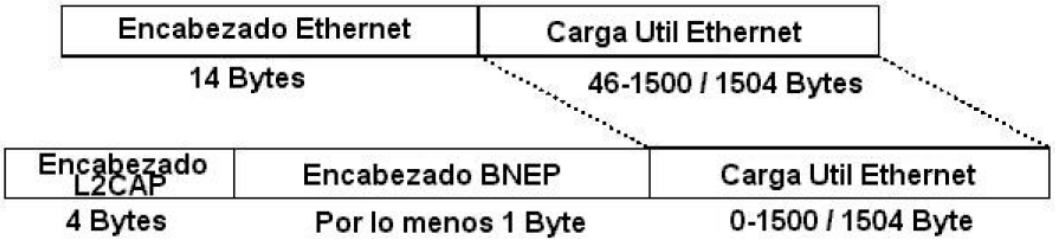


Figura 2.2 Encapsulamiento de un Paquete *Ethernet* en un Paquete L2CAP [8].

BNEP es usado para transportar sobre Bluetooth tanto paquetes de datos como paquetes de control, de esta manera brinda a los dispositivos Bluetooth capacidades de red similares a las ofrecidas por *Ethernet*.

2.2.1.4 Formato de los Encabezados BNEP

Todos los encabezados de BNEP siguen el formato que se muestra en la Figura 2.3.

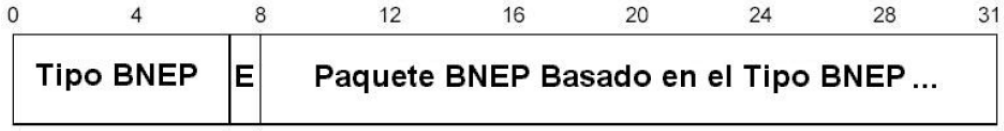


Figura 2.3 Formato de los Encabezados BNEP [8].

Tipo de BNEP: Este campo tiene un tamaño de siete bits e identifica el tipo de encabezado BNEP que contiene el paquete. Los posibles valores que puede tomar y la descripción de los mismos se muestran en la Tabla 2.1.

Valor	*Tipo de Paquete BNEP*
0x00	BNEP_GENERAL_ETHERNET
0x01	BNEP_CONTROL
0x02	BNEP_COMPRESSED_ETHERNET
0x03	BNEP_COMPRESSED_ETHERNET_SOURCE_ONLY
0x04	BNEP_COMPRESSED_ETHERNET_DEST_ONLY
Ox05-Ox7E	Reservado para un futuro uso
Ox7F	Reservado para los Paquetes *LLC* de 802.2 por *IEEE* 802.15.1 WG

Tabla 2.1 Valores para el campo BNEP *Type* el cual define el tipo de paquete BNEP [8].

- **Bandera de Extensión (E)** Corresponde a un bit de extensión que indica si existe uno o más encabezados de extensión entre el encabezado de BNEP y la carga útil. Si toma un valor de 0x1, entonces uno o más encabezados de extensión se ubican antes de la carga útil. Si el valor que tiene es 0x0, la carga útil sigue después del encabezado BNEP.

- **Paquete BNEP** Depende del valor que se haya consignado en el campo del Tipo de BNEP.

2.2.1.5 Tipo de Paquete BNEP_GENERAL_ETHERNET

El paquete general de *Ethernet* para BNEP se debe usar para transportar paquetes *Ethernet* desde y hacia redes Bluetooth. Como se puede apreciar en la Figura 2.4 el paquete está conformado por una dirección destino, una dirección origen y el tipo de protocolo de red contenido en la carga útil (IPv4, IPv6, etc). Cualquiera de las direcciones ya sea origen o destino puede corresponder a una dirección *Ethernet* IEEE, si el origen o destino es un dispositivo IEEE y no un dispositivo Bluetooth.

Figura 2.4 Formato del Encabezado para un paquete *BNEP_GENERAL_ETHERNET* [8].

2.2.1.6 Tipo de Paquete BNEP_CONTROL

El paquete de *control de BNEP* es utilizado para intercambiar información de control. En este tipo de paquete, toda la información de control está contenida en el encabezado del BNEP_CONTROL de tal manera que el campo de carga útil no contiene información alguna. Por el momento hay siete tipos de paquetes de *control BNEP.* El tipo de paquete de control es definido por el valor que se consigne en el campo *tipo de control de BNEP;* no se entra en detalle sobre cada tipo de paquete de control ya que esto no está dentro de los alcances del trabajo de grado. La Figura 2.5 muestra el formato para el encabezado de un paquete BNEP CONTROL*.*

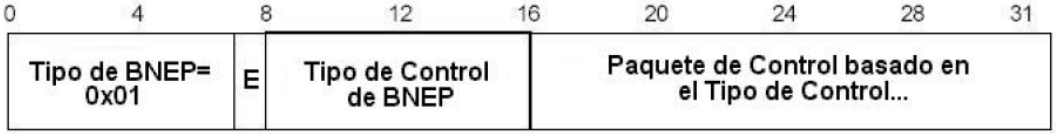


Figura 2.5 Formato del Encabezado para un paquete *BNEP_CONTROL* [8].

2.2.1.7 Tipo de Paquete BNEP_COMPRESSED_ETHERNET

El paquete *Ethernet* comprimido de *BNEP* se usa para transportar paquetes *Ethernet* hacia o desde dispositivos que tienen una conexión directa al nivel de la capa *L2CAP* usando *BNEP.* Debido a la existencia de una conexión *L2CAP* entre los dos dispositivos *Bluetooth* no es necesario incluir dentro del paquete las direcciones de origen y destino. Se debe anotar que las direcciones de *multicast o broadcast* no deben ser comprimidas. La Figura 2.6 muestra el formato de un paquete BNEP_COMPRESSED_ETHERNET*.*

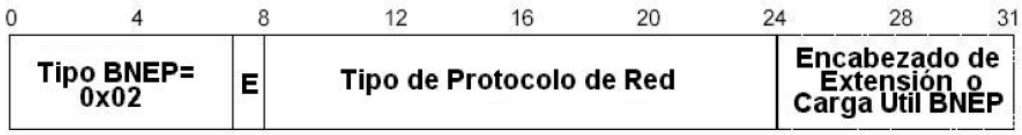


Figura 2.6 Formato del Encabezado para un paquete *BNEP_COMPRESSED_ETHERNET* [8].

2.2.1.8 Tipo de Paquete BNEP_COMPRESSED_ETHERNET_SOURCE_ONLY

Este paquete *Ethernet* comprimido se usa para transportar paquetes *Ethernet* hacia un dispositivo el cual siempre será el destino final para todos los paquetes. Por esta razón los dispositivos no necesitan incluir la dirección destino en los paquetes siendo esta la misma dirección correspondiente al canal *L2CAP* sobre el cual se envían los paquetes. Se debe anotar que las direcciones de *multicast o broadcast* no deben ser comprimidas.

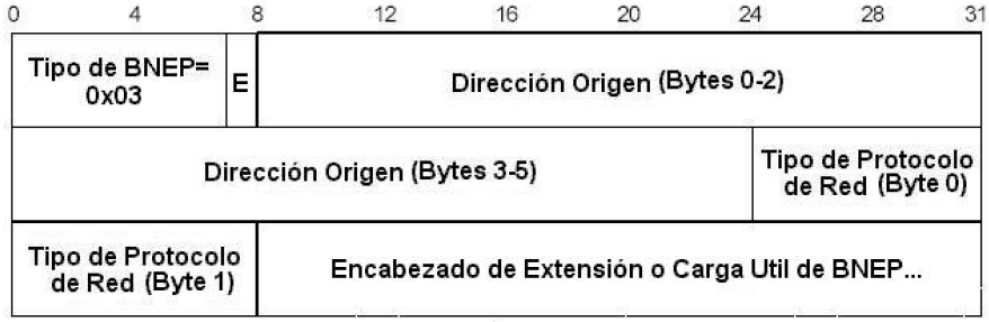


Figura 2.7 Formato del Encabezado para un paquete *BNEP_COMPRESSED_ETHERNET_SOURCE_ONLY* [8].

2.2.1.9 Tipo de Paquete BNEP_COMPRESSED_ETHERNET_DEST_ONLY

Este paquete comprimido *Ethernet* es usado para transportar paquetes desde un dispositivo el cual es la fuente del paquete. De esta manera los dispositivos no necesitan incluir la dirección de la fuente del paquete ya que esta fuente puede determinarse a partir de la conexión *L2CAP.*

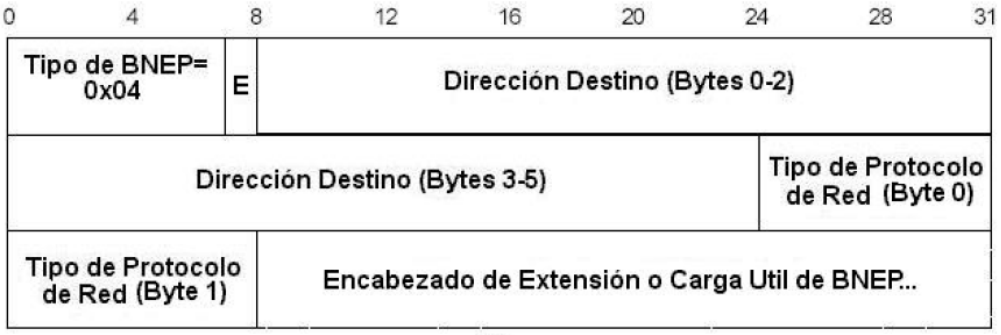


Figura 2.8 Formato del Encabezado para un paquete *BNEP_COMPRESSED_ETHERNET_DEST_ONLY* [8].

El siguiente es un ejemplo en el cual un paquete IP es enviado usando BNEP. Un paquete IPv4 es enviado desde un dispositivo con una dirección IEEE de 48 bits (00:AA:00:55:44:33) hacia un dispositivo con una dirección *Bluetooth* (00:30:67:45:67:89). En este caso el paquete BNEP utilizado es del tipo *BNEP_GENERAL_ETHERNET* como se ilustra en la Figura 2.9. [8]

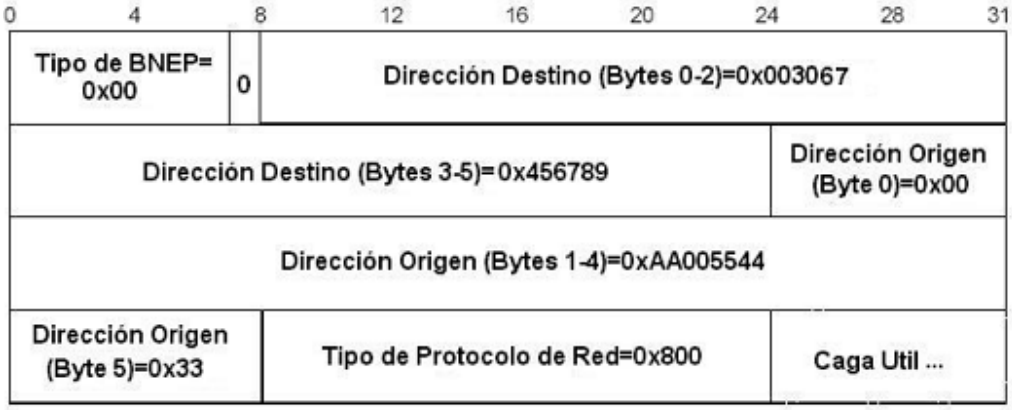


Figura 2.9 Ejemplo del envío de un paquete IPv4 usando BNEP entre un dispositivo con dirección IEEE y otro con dirección *Bluetooth* [8].

2.2.2 PERFIL DE RED DE ÁREA PERSONAL

El perfil PAN describe cómo usar el protocolo BNEP para brindar capacidades de red a los dispositivos Bluetooth. El perfil PAN presenta los siguientes requerimientos funcionales:

- Define una red IP ad-hoc, dinámica y personal
- Debe ser independiente del sistema operativo, lenguaje y dispositivo
- Brinda soporte para los protocolos de red más comunes como IPv4 e IPv6.
- Brinda soporte para puntos de acceso en donde la red puede ser una LAN corporativa, GSM (*Global System for Mobile communications*) u otro tipo de red de datos.
- Debe acomodarse a los recursos reducidos disponibles en los dispositivos pequeños respecto a memoria, capacidad de procesamiento y uso de interfaces.

2.2.2.1 Consideraciones

- El perfil PAN debe soportar *IPv4 e IPv6.* Los otros protocolos pueden estar o no habilitados.
- En una red generalizada, la trayectoria del tráfico originado desde un dispositivo hacia otro puede estar conformada por uno o varios medios de transporte, por ejemplo, Bluetooth*, Ethernet, Token Ring,* PSTN (*Public Switched Telephone Network*), ISDN *(Integrated Services Digital Network*), ATM, GSM, etc.

Son tres escenarios propuestos para el *perfil PAN: Puntos de acceso a una* red (NAP, Network Access Point), *Grupo de red Ad-hoc* (GN, *Group Ad-hoc Networks*) y conexiones Usuario PAN-Usuario PAN (PANU-PANU, PAN User-PAN User)**.** Cada uno de estos escenarios define el rol y el servicio que deben asumir los dispositivos involucrados en una de estas arquitecturas.

2.2.2.2 Puntos de Acceso a una red (NAP)

Un punto de acceso a una red (NAP) corresponde a una unidad que contiene uno o más dispositivos Bluetooth y actúa como *puente, proxy o enrutador,* entre una red Bluetooth y otro tipo de tecnología de red *(Ethernet, GSM, ISDN, Home PNA (Home Phoneline Networking Alliance), Cable Módem y Celular).* La Figura 2.10 ilustra la arquitectura para dos puntos de acceso a red.

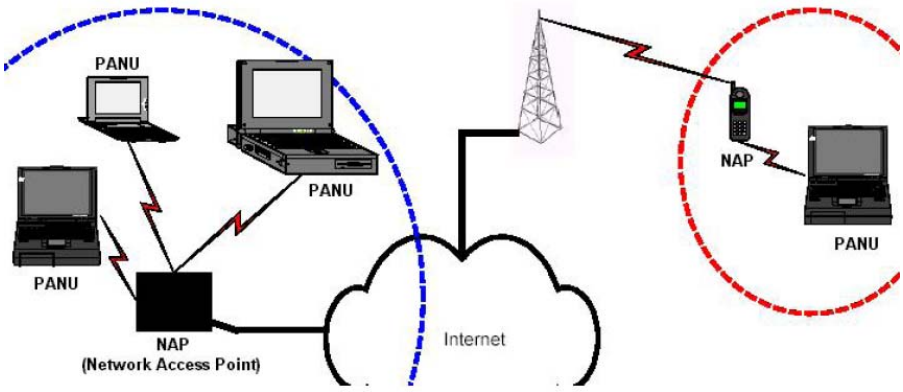


Figura 2.10 Dos Puntos de Acceso a Red [7].

Se debe hacer claridad que el esquema mostrado en la Figura 2.10 no representa la arquitectura del punto de acceso a la red, sino un diagrama de cómo está ubicado el punto de acceso dentro de un sistema. La arquitectura se planteará más adelante y evidenciará su estructura interna.

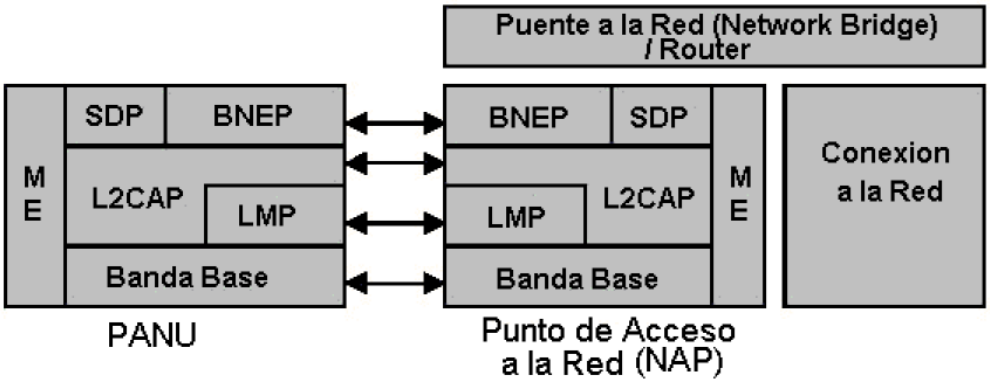


Figura 2.11 *Stack* para el rol NAP [7].

Para la Fase I del perfil PAN, el dispositivo que soporta el servicio NAP debe cumplir con las características de un "Puente *Ethernet*" para soportar de esta manera los servicios de red. El dispositivo NAP distribuye los paquetes *Ethernet* entre los dispositivos Bluetooth conectados o PANU. Los NAP pueden requerir características adicionales en los casos en que el puente sea hacia otro tipo de redes, por ejemplo GPRS (*General Packet Radio Service*). La Figura 2.11 muestra la interacción de las capas de protocolo del modelo Bluetooth para el rol NAP donde ambos tipos de dispositivos intercambian datos a través del protocolo BNEP como se explicó en la sección 2.2.1 del presente capítulo. Los protocolos SDP pertenecientes a cada tipo de dispositivo interactúan de la forma como se explicó en la sección 1.1.4 del capítulo 1.

ME (*Management Entity*) es la Entidad de Gestión la cual coordina los procedimientos durante la iniciación, configuración y gestión de la conexión.

2.2.2.3 Grupo de Red Ad-hoc

La *versión 1.0 del Perfil PAN* [5], especifica el escenario para una red personal ad-hoc el cual consiste en una simple *piconet* con conexiones entre dos o más *dispositivos Bluetooth.* Un maestro y un máximo de siete esclavos conforman esta red. El límite de siete esclavos se debe al esquema activo de direccionamiento de Bluetooth [3]. La Figura 2.12 es un esquema para una red Ad-hoc.

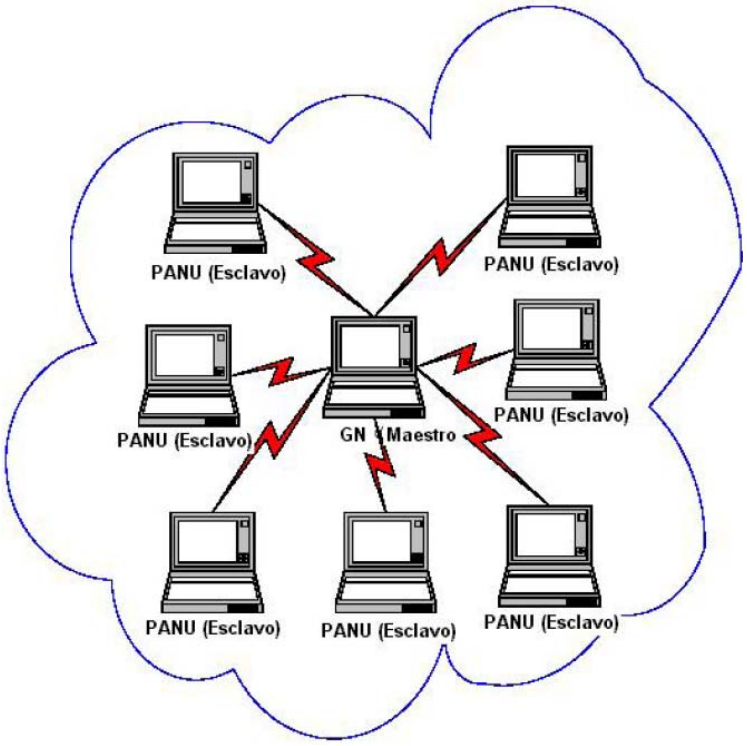


Figura 2.12 Esquema de un grupo de red Ad-Hoc (GN) [7].

Un grupo de red ad-hoc se establece para que un conjunto de dispositivos conforme una red temporal e intercambien información. El dispositivo cuyo rol es GN soporta el servicio *GN.* La Figura 2.13 ilustra a nivel de las capas de protocolo un enlace entre un GN y un PANU.

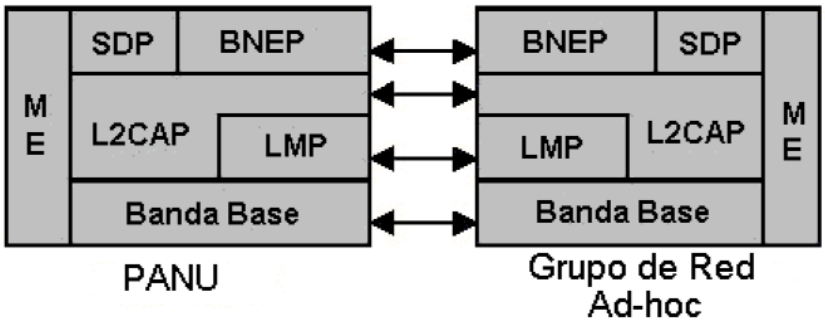


Figura 2.13 *Stack* para un enlace en una red Ad-Hoc Bluetooth [7].

2.2.2.4 PANU-PANU

En este escenario se establece una conexión punto a punto entre dos usuarios de PAN *(PANU),* permitiéndose así una comunicación directa entre ellos. El PANU es el dispositivo Bluetooth que usa los servicios *NAP* o *GN.* El PANU asume el rol de cliente de los roles NAP o GN [7].

2.3 FACTORES QUE AFECTAN LA CALIDAD DE LA VOZ SOBRE REDES DE PAQUETES

En esta sección se identifican factores que afectan las comunicaciones de voz en redes IP con el objetivo de deducir los parámetros y criterios a tomar en cuenta para la definición de la arquitectura.

2.3.1 Factor de compresión

Para poder transmitir la voz a través de una red de datos, es necesario realizar previamente un proceso de digitalización. En telefonía clásica, éste proceso se realiza utilizando CODECs, obteniendo una señal digital de 64 Kbps. Este proceso, se realiza de acuerdo a la recomendación G.711 de la ITU-T (*International Telecommunication Union – Telecommunication*) [11]. Sin embargo, cuando se dispone de velocidades de red reducidas, es conveniente tratar de minimizar el ancho de banda requerido por las señales de voz.

Igualmente, los algoritmos de compresión ayudan a optimizar la infraestructura de red de forma que se llegue a obtener tanta capacidad como sea posible, pero los algoritmos de

compresión involucran una compensación entre eficiencia y encabezado (información adjunta para garantizar un control de errores) que se debe considerar.

Algoritmo	Descripción
G.711	Codificación de Audio a 64 k bit/s (Ley µy Ley A)
G.722	7 kHz Velocidad a 48, 56 y 64K bit/s (voz hi-fi)
G.723.1	Velocidad de tasa dual a 6.4 (basada en MP-MLQ, *Multipulse, Multilevel Quantization*) y (5.3 k bit/s basada en CELP, *Code Excited Linear Prediction Compression*)
G.726:	Codificación de ADPCM (*Adaptative Differential Pulse Code Modulation*) a 40, 32, 24, 16 Kbps;
G.728	Variación de bajo retardo de 16 Kbps de una compresión de voz CELP.
G.729 Annex A	Voz codificada a 8 k bit/s (CS-ACELP, *Conjugate Structure - Algebraic Code Excited Linear Prediction). Reduce complejidad.*
G.729 Annex B	Voz codificada a 8 k bit/s (CS-ACELP*). Reducción de silencios.*
G.729 Annex AB	Voz codificada a 8 k bit/s(CS-ACELP*). Reducción de silencios y complejidad.*

Tabla 2.2 Descripción de los algoritmos estandarizados para la compresión de voz [14].

Para la compresión de la voz se han desarrollado varias recomendaciones, que reducen la velocidad de transmisión requerida, a expensas de degradar la calidad de ésta. En la Tabla 2.2 se resume las recomendaciones de la ITU-T respecto a los algoritmos estandarizados de compresión de voz

Mientras G.711 es el CODEC (*Codificador-Decodificador*) digital del flujo principal para servicios de voz de alta calidad, un número más de CODECs eficientes se usan para aplicaciones de celular y voz. En una red IP, los CODECs de voz se ubican en paquetes de muestras de voz con una duración de 5, 10 o 20 ms, y estas muestras se encapsulan en un paquete de VoIP.

Otro método de compresión utilizado a menudo es la Modulación de Pulsos Codificados Diferencial Adaptativa (ADPCM*)*. Un ejemplo común de la utilización de ADPCM es la ITU-T G.726, codifica utilizando muestras de 4 bits, lo que da una velocidad de transmisión de 32Kbps. A diferencia de la PCM (*Pulse Code Modulation*), los 4 bits no codifican directamente la amplitud de la voz, sino que codifican las diferencias de la amplitud.

PCM y ADPCM son ejemplos de codificación por forma de ondas, técnicas de compresión que explotan las características redundantes de la forma de onda.

En los últimos años se han desarrollado nuevas técnicas que emplean procedimientos de procesamiento de señales que comprimen la voz enviando sólo información paramétrica simplificada sobre la vibración y modulación de la voz original, necesitando menos ancho de banda para transmitir esa información. Estas técnicas se pueden agrupar

generalmente como CODECs de origen, e incluyen variaciones como la codificación con predicción lineal (LPC, *Linear Predictive Coding*), la compresión de predicción lineal con excitación por código (CELP) y la MP-MLQ [11] [12] [13] [16].

2.3.2 Pérdida de paquetes

A diferencia de las redes telefónicas, donde para cada conversación se establece un vínculo estable y seguro, las redes de datos admiten la pérdida de paquetes, más aún cuando se trata de redes inalámbricas. Esto está previsto en los protocolos seguros de alto nivel (protocolos orientados a la conexión), y en caso de que ocurra, los paquetes son reenviados. En los protocolos diseñados para tráfico de tiempo real generalmente no se recibe confirmaciones de recepción de paquetes, ya que si el canal es suficientemente seguro, estas confirmaciones cargan inútilmente al mismo. En aplicaciones de voz, el audio es encapsulado en paquetes y enviado, sin confirmación de recepción de cada paquete así que si el porcentaje de pérdida es pequeño, la degradación de la voz también lo es. Los porcentajes de perdida admisibles dependen de otros factores, como por ejemplo la demora de transmisión y el factor de compresión de la voz. Existen técnicas para hacer menos sensible la degradación de calidad en la voz frente a la pérdida de paquetes. La más sencilla consiste en simplemente repetir el último paquete recibido (utilización de una implementación G.729 [13]), sin embargo si se perdieran múltiples paquetes de forma consecutiva, la estrategia de ocultación se ejecuta solo una vez hasta que se reciba otro paquete. De igual manera los paquetes que llegan a destiempo o fuera de orden también se consideran como paquetes perdidos.

Debido a la estrategia de ocultación de G.729, de modo empírico se puede decir que G.729 tolera hasta un cinco por ciento de pérdida de paquetes como media a lo largo de toda una conversación [13].

2.3.3 Retardo

El retardo o latencia en VoIP es el tiempo que tarda la voz en salir de la boca de una persona que está hablando y en llegar al oído de la persona receptora o que está escuchando.

Este es un factor muy importante en la percepción de la calidad de la voz presentada por un sistema. El retardo total está determinado por varios factores, entre los que se encuentran: retardo debido a los algoritmos de compresión, retardo de procesamiento y retardo de propagación [15].

2.3.3.1 Retardo debido a los algoritmos de compresión

En forma genérica, cuanto mayor es la compresión, más retardo producido en el proceso se introduce, debido a que en general los CODECs requieren más tiempo para codificar

cada muestra. En la Tabla 2.3 se muestran los retardos introducidos típicos de los CODECs más utilizados.

Algoritmo de muestreo/compresión	Retardo típico introducido
G.711 (64 Kbps)	125 µs
G.728 (16 Kbps)	2.5 ms
G.729 (8 Kbps)	10 ms
G.723 (5.3 o 6.4 Kbps)	30 ms

Tabla 2.3 Retardo típico introducido por los CODECs [15].

Por ejemplo en un producto de VoIP, el procesador digital de señal DSP (*Digital Signal Processor*), genera una muestra de voz cada 10ms cuando se utiliza el CODEC G.729. Dos de esas muestras de voz (ambas con 10ms de retardo) se colocan dentro de un paquete. El retardo de paquetes, es por lo tanto de 20ms. Cuando se utiliza el CODEC G.729 se produce un *look-ahead*[5] inicial de 10ms, lo que supone un retardo de 25ms para la primera trama de voz.

2.3.3.2 Retardo de procesamiento

Es el tiempo involucrado en el procesamiento de la voz para la implementación de los protocolos. Dependen de los procesadores utilizados.

Una red basada en paquetes sufre retardos por otras razones. Dos de estas razones son el tiempo que se necesita para mover un paquete hasta la cola de salida (conmutación de paquetes) y el retardo de la gestión de colas.

Cuando los paquetes se guardan en una cola debido a la congestión en una interfaz *outbound* (de salida), el resultado es un retardo en la gestión de colas. Este tipo de retardos ocurre cuando se envían más paquetes que los que la interfaz puede manejar en un intervalo de tiempo dado.

El retardo en la gestión de colas, es otra causa que se le suma al retardo. Este debe estar por debajo de 10ms [15] siempre que se pueda, utilizando cualquier método de gestión de colas que sea óptimo para la red.

En una red no administrada y congestionada, el retardo en la gestión de colas puede agregar más de dos segundos de retardo (o provocar que el paquete se caiga). Este largo periodo de retardo es inaceptable en casi todas las redes de voz. El retardo en la gestión de colas es solo un componente del retardo de extremo a extremo, el cual también se ve afectado por la fluctuación de fase.

[5] *look-ahead* : retardo que representa la cantidad de datos requeridos en la próxima trama para comprimir la trama actual.

2.3.3.3 Retardo de propagación

Una red de fibra óptica alrededor del mundo (21.000 m) induce un retardo en un solo sentido de unos 70 milisegundos (70ms). Aunque este retardo es casi imperceptible al oído humano, la sumatoria de retardos de diversa índole, pueden provocar una degradación apreciable de la voz. Normalmente las redes inalámbricas introducen un retardo de propagación mayor que las redes cableadas, lo que se deduce por la falta de medio físico de transmisión como tal.

Adicionalmente al retardo por la velocidad de transmisión de los enlaces, la congestión, y las demoras de los equipos de red (*routers*, *gateways*, etc.) introducen retardo que se consideran retardo de propagación de una red de telecomunicaciones.

Las demoras no afectan directamente la calidad de la voz, sino la calidad de la conversación. Hasta 100 ms son generalmente tolerados, casi sin percepción de los interlocutores. Entre 100 y 200 ms las demoras son notadas. Al acercarse a los 300 ms de demora, la conversación se vuelve poco natural. Pasando los 300 ms la demora se torna crítica, haciendo muy dificultosa la conversación.

La recomendación G.114 de la ITU-T (*One-way Transmission Time*) especifica que para una buena calidad de voz no debe darse un retardo mayor de 150ms en un solo sentido, de extremo a extremo, como muestra en la Figura 2.14.

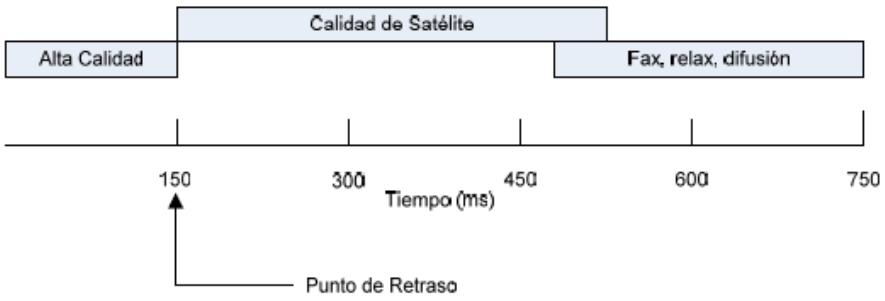


Figura 2.14 Retardo de extremo a extremo [15].

Como se puede observar en la Figura 2.14, algunas formas de retardo son más largas, e incluso superan los valores indicados en la recomendación antes citada, sin embargo, se aceptan debido a que no existe otra alternativa, por ejemplo, en la transmisión por satélite se tarda aproximadamente 250 ms para que una transmisión alcance el satélite y otros 250ms para volver a la tierra. Esto provoca un retardo total de 500ms. A pesar que la recomendación de la ITU-T afirma que esto está fuera de lo aceptable para la calidad de la voz, muchas conversaciones tienen lugar cada día sobre enlaces de satélite. De esta manera, la calidad de voz viene a menudo definida como *lo que los usuarios aceptan y utilizan* [15].

2.3.4 Fluctuación de fase (*Jitter*)

Dicho de manera sencilla, la fluctuación de fase (*jitter*) es la variación del tiempo de llegada de un paquete. Cuando está en un entorno de voz por paquetes, el remitente espera transmitir de forma fiable paquetes de voz en un intervalo regular (por ejemplo, enviar una trama cada 20ms). Esos paquetes de voz se pueden retrasar por toda la red de paquetes y no llegar con el mismo intervalo de tiempo regular a la estación receptora (por ejemplo, puede que no sean recibidos cada 20ms) [15].

La diferencia entre cuando se esperaba recibir el paquete y cuando se recibe en realidad es lo que se llama la fluctuación de fase.

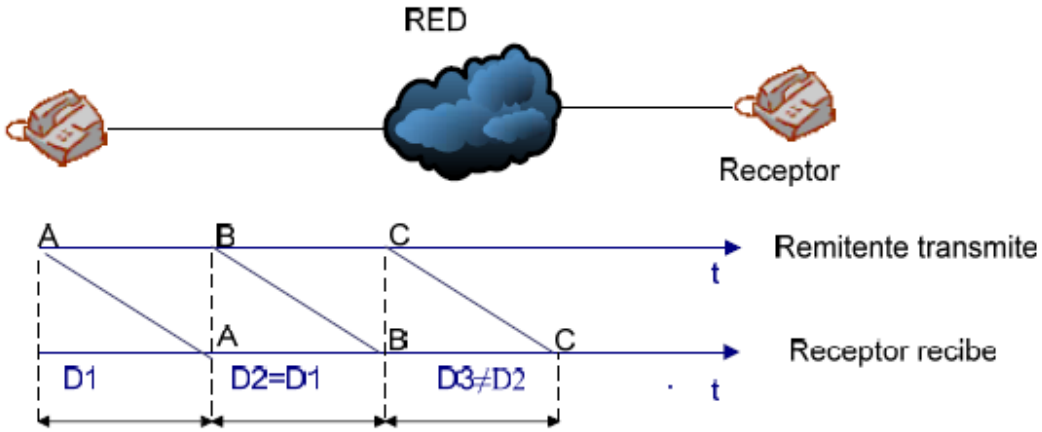


Figura 2.15 Fluctuación de Fase [15].

En la Figura 2.15 se puede ver que el tiempo que se tarda en enviar y recibir paquetes A y B es el mismo (D1=D2). El paquete C tiene un retardo en la red y se recibe después de la hora a la que se esperaba. Es por esto que es necesario un búfer de fluctuación de fase que oculte el retardo.

Es importante resaltar que la fluctuación de fase y el retardo total no son la misma cosa, a pesar de que tener muchas fluctuaciones de fase en una red de paquetes puede incrementar el retardo total en la red. Esto se debe a que cuanta más fluctuación de fase haya, mas necesitará ser compensado el búfer de fluctuación de fase por la impredecible naturaleza de la red de paquetes.

Si la red de datos está bien construida y se toman las precauciones apropiadas, la fluctuación de fase es normalmente un problema menor y el búfer de fluctuación de fase no contribuye significativamente el retardo total de extremo a extremo.

Las *timestamps* (marcas de tiempo) de RTP (*Real Time Protocol*) se utilizan dentro de algunas implementaciones para determinar qué nivel de fluctuación de fase existe dentro de la red. El búfer de fluctuación de fase se considera como una cola dinámica, o estática.

En el primer caso, esta cola puede crecer o disminuir exponencialmente dependiendo del tiempo de los paquetes RTP entre llegada y llegada. En el segundo caso ésta tiene un valor fijo, es decir que el búfer almacena una porción *limitada* de la conversación. La mayoría de los fabricantes eligen utilizar búferes de fluctuación de fase estática, sin embargo, esto obliga a que el búfer sea demasiado grande o demasiado pequeño, y por lo tanto, que la calidad de audio se recienta debido a la pérdida de paquetes o a un retraso excesivo. Algunas empresas, entre las que se cuenta Cisco, utilizan un búfer de fluctuación de fase que se incrementa o disminuye dinámicamente dependiendo de la variación de retraso entre llegadas de los últimos paquetes, lo cual evita los inconvenientes mencionados anteriormente para las colas estáticas [15].

2.3.5 Nota Media de Opinión (MOS, *Mean Opinion Score*):

Hay dos formas de probar la calidad de la voz: subjetiva y objetivamente. Los humanos realizan pruebas de calidad de voz subjetivas, mientras que las computadoras realizan pruebas de voz objetivas. Los CODECs se han desarrollado y armonizado sobre la base de medidas de calidad de voz. Las medidas estándar de calidad objetiva, como una total distorsión armónica y relación señal a ruido no se corresponden muy bien con una percepción de calidad de voz humana, lo que al final es la meta de la mayoría de las técnicas de compresión de voz.

Una referencia subjetiva común para cuantificar el rendimiento del CODEC de voz es lo que se llama la nota media de opinión (MOS). Las pruebas MOS se dan a un grupo de oyentes. Como la calidad de voz y sonido es subjetiva para los oyentes en general, es importante obtener una amplia gama de oyentes y material cuando se realiza una prueba MOS. Los oyentes otorgan a cada muestra de materia de voz una puntuación entre 1 (malo) y 5 (excelente). Se saca luego una media para obtener la puntación media de la opinión.

La comprobación MOS se utiliza también para comparar como funciona un CODEC determinado bajo circunstancias distintas, incluidos diferentes niveles de ruidos de fondo, múltiples codificaciones y decodificaciones, etc. En la Tabla 2.5, se pueden observar el valor MOS para los CODECs más comunes.

Medición de la calidad de voz según la percepción: Aunque la puntuación MOS es un método subjetivo para determinar la calidad de la voz, no es el único método para hacerlo. La ITU-T ha sacado la recomendación P.861, que cubre las maneras con las que se puede determinar objetivamente la calidad de voz utilizando la medición de esta según la percepción (PSQM, *Perceptual Speech Quality Measurement*).

PSQM tiene muchos inconvenientes cuando se utiliza con CODECs de voz (*vocoders*). Uno de estos inconvenientes es que lo que la “máquina” o PSQM “oye” no es lo que percibe el oído humano. En otros términos, una persona puede engañar al oído humano al percibir una voz de mayor calidad, pero una computadora no puede. PSQM fue

desarrollado para oír deterioros provocados por la compresión y descompresión y no por la pérdida de paquetes o la fluctuación de fase.

2.3.6 Eco

Un efecto secundario, generado por los retardos elevados, es el eco. El eco se debe a que parte de la energía de audio enviada es devuelta por el receptor. En los sistemas telefónicos este efecto no tiene mayor importancia, ya que los retardos o demoras son despreciables, y por lo tanto, el eco no es percibido como tal. Cuando el retardo de extremo a extremo comienza a aumentar, el efecto del eco comienza a percibirse.

El eco es un fenómeno que en una conversación puede ir desde lo ligeramente molesto hasta lo insoportable, provocando que la conversación sea inteligible. Oír la propia voz en el auricular mientras se está hablando es común y tranquilizador para la persona que está hablando, pero oír la propia voz después de un retardo de unos 25ms puede provocar interrupciones y romper la cadencia de la conversación.

El eco tiene dos inconvenientes: puede ser alto y puede ser largo. Entre más alto y largo sea, más incomodo resultará. En las actuales redes basadas en paquetes, se pueden construir canceladores de eco en CODECs de velocidad de transmisión baja y hacerlos funcionar en cada DSP. En las implementaciones de algunos fabricantes, la cancelación del eco se hace en el software; esta práctica reduce drásticamente los beneficios de la cancelación del eco. El dispositivo a través del cual está hablando un usuario guarda una imagen inversa de las palabras del usuario durante un cierto tiempo. Es lo que se llama voz inversa (*inverse speech* [-G]). Este cancelador de eco oye el sonido que viene del otro usuario y sustrae el –G para eliminar todo el eco. Los canceladores de eco están limitados por la cantidad total de tiempo que esperan a que llegue la palabra reflejada, un fenómeno conocido como *echo tail (eco posterior)*.

2.3.7 Encapsulamiento de VoIP

En un principio las palabras de voz provenientes del cliente de VoIP, se convierten en la carga útil del protocolo RTP, éste a su vez es encapsulado en un paquete UDP en el cual se especifican los puertos de fuente y destino. Este paquete UDP se convierte en la carga útil del paquete IPv4 o IPv6 (dependiendo de la versión de IP con la que se trabaja). Para IPv4, el encabezado es de 40 bytes y para IPv6, este encabezado aumentará a 60 bytes [14]. Este último paquete IP, es transportado por alguna tecnología de nivel 2 como *Ethernet* (802.3), WLAN (*Wireless Local Area Network*) o un paquete Bluetooth encapsulado de manera tal que pueda viajar a través de redes *Ethernet* (enfoque del proyecto).

VoIP tiene un aspecto de sobrecarga importante a mencionar cuando altos niveles de compresión de voz se aplican a paquetes de corta duración. El sacrificio entre la sobrecarga y la duración del paquete se muestra en la Tabla 2.4, por ejemplo, se puede observar que se obtiene una mejor eficiencia cuando se envían varias palabras G.711 en el paquete de voz.

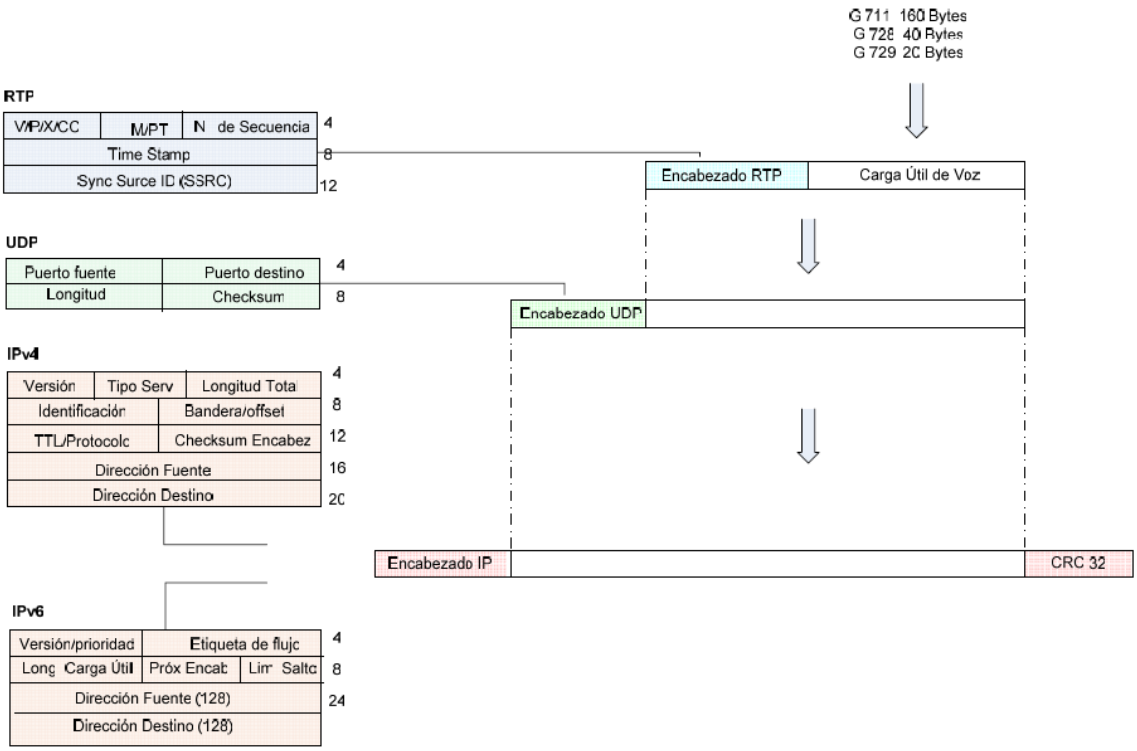


Figura 2.16 Encapsulamiento de VoIP IPv4 e IPv6 [14].

	Duración de la trama del paquete de voz (ms)			
CODEC Utilizado	**5**	**10**	**20**	**40**
IPv4 G.711	47,6%	64,5%	78,4%	87,9%
IPv6 G.711	38,5%	55,6%	71,4%	93,3%
IPv4 G.726	31,3%	47,6%	64,5%	78,4%
IPv6 G.726	23,8%	38,5%	55,6%	71,4%
IPv4 G.729	10,2%	18,5%	31,3%	47,6%
IPv6 G.729	7,2%	13,5%	23,8%	38,5%

Tabla 2.4 Eficiencia: sobrecarga vs duración del paquete [16].

En la Tabla 2.5, se listan los requerimientos en una vía para los CODECs típicos de voz usados en VoIP. Así mismo se listan otros factores como la puntuación MOS que se explicó anteriormente.

Métodos de Compresión	Velocidad de bit (Kbps)	Tamaño de muestra (ms)	Puntuación MOS	Ancho de banda típico IP (en un solo sentido)
G.711 PCM	64	0.125	4.1	80Kbps
G.726 ADPCM	32	0.125	3.85	48Kbps
G.728 LD-CELP (Predicción Lineal con excitación por código de bajo retardo)	15	0.625	3.61	32Kbps
G.729 CS-ACELP (Predicción lineal con excitación por código algebraico de estructura conjugada)	8	10	3.92	23.7Kbps
G.729 (A) CS-ACELP	8	10	3.7	24Kbps
G.723.1 MP-MLQ	6.3	30	3.9	17.07Kbps
G.723.1 ACELP	5.3	30	3.65	16.27Kbps

Tabla 2.5 Características relacionadas con los CODECs [16].

En la Tabla 2.5 se observan dos aspectos determinantes: La puntuación MOS que indica el entendimiento de la conversación y el ancho de banda típico que consume cada uno de los métodos de compresión. Haciendo un análisis, se tiene que el CODEC G.711 es el mejor en cuanto a calidad de voz se refiere y el CODEC G.723.1 es el que menos ancho de banda consume, sin embargo se tiene que el CODEC G.729 es un punto intermedio entre ambos, entregando una calidad de voz bastante buena y un ancho de banda considerablemente menor que el CODEC G.711.

2.3.8 Ancho de banda

Hay algunos aspectos a tener en cuenta para la definición del ancho de banda requerido. Las consideraciones discutidas a continuación afectarán el ancho de banda de las redes de voz:

- ✓ CODECs de voz
- ✓ Muestreos
- ✓ Detección de actividad de voz
- ✓ Compresión del encabezado RTP

***CODECs de voz*:** Como se aclaró anteriormente cada CODEC tiene diferentes tasas de bits y algunos son más complejos que otros (Tabla 2.5). Los CODECs impactan el ancho de banda requerido porque ellos determinan el tamaño de la carga útil de los paquetes transferidos sobre el tramo IP de una llamada. Si se incrementa la carga útil, se reduce el número de paquetes enviados, por lo tanto se decrementa la necesidad de ancho de banda necesitada por la reducción del número de cabeceras requeridas para la llamada.

El número de muestras por paquete es otro factor para determinar el ancho de banda de una llamada de voz. El CODEC define el tamaño de la muestra, pero el total de muestras puestas en un paquete determina qué cantidad de paquetes son enviados por segundo. De tal forma que, el número de muestras incluidas en un paquete afecta el conjunto de ancho de banda de una llamada [17].

Por ejemplo:

Una muestra de G.711 de 10 ms es de 80 bytes por muestra. Una llamada de solo una muestra por paquete debería ser así:

80 bytes + 20 bytes IP + 12 UDP + 8 RTP = 120 bytes por paquete
*120 bytes por paquete*100 pps =(12000*8 bits)/1000 =96 Kbps por llamada.*
La misma llamada usando 2 muestras de 10 ms por paquete debería mostrar lo siguiente:

*(80 bytes*2 muestras)+20 bytes IP+12 UDP+ 8RTP = 200 bytes por paquete*
*(200 bytes por paq.) * (50 pps) = (10000 * 8 bits)/1000 = 80 Kbps por llamada (este valor se tiene en cuenta en la Tabla 2.5).*

Nota: En los cálculos anteriores no se tomaron en cuenta los encabezados de nivel 2 dentro de los que se pueden considerar los encabezados Ehternet y BNEP.

El resultado muestra que hay una diferencia de 16Kbps entre las dos llamadas. Cambiando el número de muestras por paquete, definitivamente se puede cambiar la cantidad de ancho de banda que una llamada usa, sin embargo, cuando se incrementa el número de muestras por paquete, también se incrementa la cantidad de retardo en cada llamada puesto que los recursos DSP, los cuales manejan cada llamada, deben almacenar las muestras por un periodo de tiempo mayor. Se debe tener en cuenta esto cuando se diseña una red de voz.

Detección de actividad de voz (VAD, Voice Activity Detection): Las conversaciones típicas de voz pueden contener entre un 35% y 50% de silencio. Con las redes tradicionales, basadas en circuitos, todas las llamadas de voz usan un ancho de banda fijo de 64 Kbps sin importar cuánto silencio haya en éstas. Con las redes de VoIP, todas las conversaciones y silencios son paquetizados. La VAD envía paquetes RTP solamente cuando la voz es detectada. Para el planeamiento de ancho de banda de VoIP, se asume que VAD reduce el ancho de banda hasta en un 35%. Aunque este valor puede ser menor, este provee una estimación aproximada para varios dialectos y patrones de lenguaje.

Los CODECs G.729 Anexo-B y G.723.1 Anexo-A incluyen una función VAD integrada, pero aparte de eso tienen un desempeño igual al G.729 y G.723.1, respectivamente.

Compresión del Encabezado RTP: Como se explicó anteriormente (ver Figura 2.16) todos los paquetes de VoIP tiene dos componentes: muestras de voz y encabezados IP/UDP/RTP. Aunque las muestras de voz son comprimidas por el procesador digital de señales (DSP) y varia en tamaño dependiendo del CODEC que use, los encabezados siempre son de 40 bytes para IPv4 o de 60 bytes para IPv6. Cuando se comparan los 20 bytes de muestras de voz en una llamada con el CODEC G.729, los encabezados toman

una considerable cantidad de sobrecarga (el doble). Usando la compresión de encabezado RTP (cRTP, *Compressed Real Time Protocol*), estos encabezados pueden ser comprimidos a 2 o a 4 bytes. Esta compresión puede ofrecer un ahorro de ancho de banda sustancial. Por ejemplo una llamada de VoIP consume 24 Kbps sin cRTP, pero sólo 12 Kbps con cRTP activada. El tipo de código, las muestras por paquete, VAD, y cRTP afectan, de una u otra manera. En cada caso, hay una transacción entre la calidad de voz y el ancho de banda. La Tabla 2.6 muestra la utilización del ancho de banda para varios escenarios. La eficiencia VAD se asume de un 50% para observar los efectos que induce. Como se vio anteriormente los silencios son relativos al tipo de conversación que se esté llevando a cabo, es decir si se trata de una conversación fluida los silencios pueden ser pocos [17].

En la Tabla 2.6 se relacionan los efectos del tamaño de la carga útil en los requerimientos de ancho de banda de varios de los CODECs.

2.4 CRITERIOS DEFINIDOS

En el presente capítulo se dio una descripción detallada del perfil PAN el cual en cierta forma da capacidades *Ethernet* a los dispositivos Bluetooth. Se explicó a profundidad el protocolo BNEP que soporta dicho perfil y se plantearon los escenarios de aplicabilidad de éste donde se plantea un diagrama de cómo estarían ubicados los puntos de acceso en sistemas basados en la arquitectura.

Así mismo, se estudiaron las implementaciones del *stack* Bluetooth en el sistema operativo Linux, para la especificación del mencionado puente de interconexión en la arquitectura.

Adicionalmente se estudiaron cuidadosamente los factores que afectan la calidad de la voz sobre redes de paquetes, definiendo los criterios que se deben de tener en cuenta para comunicaciones de voz sobre dichas redes.

Así, finalmente los criterios a tener en cuenta para la construcción de la arquitectura son:

- Los perfiles y protocolos necesarios tales como PAN y BNEP para lograr la interconexión de dispositivos Bluetooth con dispositivos pertenecientes a redes cableadas *Ethernet*.
- Factores que afectan la calidad de la voz en redes de paquetes.
- Implementaciones del *stack* Bluetooth en el sistema operativo Linux.

CODEC/ Algoritmo	Ancho de banda de Voz (Kbps)	Tamaño de Trama (bytes)	Paquetes por segundo	Encabezado IP/UDP/RTP (bytes)	Encabezado CRTP (bytes)	L2	Encabezado capa 2(bytes)	Ancho de banda total sin VAD (Kbps)	Ancho de banda total con VAD (Kbps)
G.711	64	80	50	40	-	Ether	14	85.6	42.8
G.711	64	80	50	-	2	Ether	14	70.4	35.2
G.711	64	80	50	40	-	PPP	6	82.4	41.2
G.711	64	80	50	-	2	PPP	6	67.2	33.6
G.711	64	80	50	40	-	FR	4	81.6	40.8
G.711	64	80	50	-	2	FR	4	66.4	33.2
G.711	64	80	100	40	-	Ether	14	107.2	53.6
G.711	64	80	100	-	2	Ether	14	76.8	38.4
G.711	64	80	100	40	-	PPP	6	100.8	50.4
G.711	64	80	100	-	2	PPP	6	70.4	35.2
G.711	64	80	100	40	-	FR	4	99.2	49.6
G.711	64	80	100	-	2	FR	4	68.8	34.4
G.729	8	10	50	40	-	Ether	14	29.6	14.8
G.729	8	10	50	-	2	Ether	14	14.4	7.2
G.729	8	10	50	40	-	PPP	6	26.4	13.2
G.729	8	10	50	-	2	PPP	6	11.2	5.6
G.729	8	10	50	40	-	FR	4	25.6	12.8
G.729	8	10	50	-	2	FR	4	10.4	5.2
G.729	8	10	33	40	-	Ether	14	22.4	11.2
G.729	8	10	33	-	2	Ether	14	12.3	6.1
G.729	8	10	33	40	-	PPP	6	20.3	10.1
G.729	8	10	33	-	2	PPP	6	10.1	5.1
G.729	8	10	33	40	-	FR	4	19.7	9.9
G.729	8	10	33	-	2	FR	4	9.6	4.8
G.723.1	6.3	30	26	40	-	Ether	14	17.6	8.8
G.723.1	6.3	30	26	-	2	Ether	14	9.7	4.8
G.723.1	6.3	30	26	40	-	PPP	6	16.0	8.0
G.723.1	6.3	30	26	-	2	PPP	6	8.0	4.0
G.723.1	6.3	30	26	40	-	FR	4	15.5	7.8

Tabla 2.6 Efectos del tamaño de carga útil en los requerimientos de ancho de banda [18].

CAPÍTULO 3 DEFINICIÓN DE LOS REQUERIMIENTOS TECNOLÓGICOS NECESARIOS PARA EL TRANSPORTE DE VOZ EN AMBIENTES DE INTEROPERABILIDAD DE LA TECNOLOGÍA BLUETOOTH CON REDES IP CABLEADAS ETHERNET.

En los capítulos anteriores se definieron el perfil y el protocolo respectivo de soporte necesarios para proveer capacidades IP a las conexiones Bluetooth. Se definió el perfil PAN sobre el protocolo BNEP como la opción más idónea para este fin. En este capítulo se estudiarán y definirán las capacidades de la tecnología Bluetooth para prestar servicios de VoIP en base a lo definido en los capítulos anteriores. También se especificarán los protocolos subyacentes sobre los que se soporta el protocolo BNEP, así como las particularidades de la Calidad del Servicio (QoS) que éstos ofrecen.

3.1 CONSIDERACIONES DE REDES BLUETOOTH CON CAPACIDADES IP

En la prestación de servicios de voz debe tenerse en cuenta que Bluetooth puede prestar dicho servicio empaquetando la información codificada de voz o generando flujos (*streams*) de audio sobre enlaces síncronos, pero sólo a nivel L2CAP es gestionado el empaquetamiento de servicios de niveles superiores como telefonía IP, soportando simultáneamente capas de mayor jerarquía en la pila de protocolos (*stack*) (Figura 3.1).

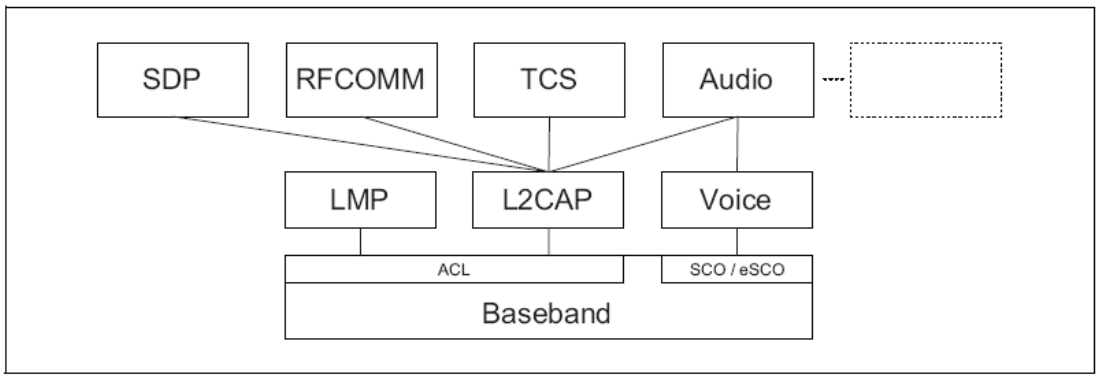


Figura 3.1 Flujos de datos L2CAP en la arquitectura de protocolos Bluetooth [3].

Como se menciono en el capítulo 1, los requerimientos funcionales de la capa L2CAP incluyen multiplexación de protocolos y canales, segmentación y reensamblado, control de flujo por canal, control de errores y gestión de grupos. L2CAP soporta el protocolo de control de enlace e interfaces con otros protocolos de comunicación tales como SDP, RFCOMM, TCS y BNEP.
Además de lo anterior, L2CAP brinda la posibilidad del préstamo de QoS. El núcleo (*Core*) de la especificación *Bluetooth* 1.1 incorpora un número de mecanismos que pueden

afectar factores como la QoS según el tipo de aplicación Bluetooth. La especificación 1.2 agrega algunas nuevas características que brindan más opciones, aclarando también en mayor profundidad la relación de QoS y la capa L2CAP [19].

Los parámetros que afectan el uso de QoS no se enfocan en una sola capa de la pila de protocolos Bluetooth, pero dicha calidad se puede ver afectada por el tipo de conexión y demás parámetros que intervengan. Además, para conexiones ACL hay un factor importante que es la QoS a nivel de la capa L2CAP. Por lo tanto, es necesario establecer el nivel de QoS acorde a la aplicación en términos de la capa L2CAP. Está opción describe un flujo similar al detallado en el RFC (*Request For Comments*) 1363[6].

3.2 CANALES BLUETOOTH

Una radiocomunicación *Bluetooth* puede utilizar dos tipos de conexiones: orientada a conexión y no orientada a conexión. Dentro de los canales orientados a conexión existen 2 tipos de *flujos:* asincrónicos y sincrónicos.

En los canales orientados a conexión Bluetooth, los datos asincrónicos son mapeados por la capa L2CAP y llevados a tipos de paquetes ACL. Estas conexiones (Figura 3.2) son confiables, lo cual significa que estos paquetes serán confirmados si son recibidos, retransmitidos si no es recibida la confirmación y un error será reportado si la retransmisión finalmente falla. Los enlaces ACL son comparables al protocolo TCP de la *pila* TCP/IP.

Las transmisiones ASB (*Active Slave Broadcast*) y PSB (*Parked Slave Broadcast*) (Figura 3.2) son no confiables. En lo absoluto, no hay manera para que el transmisor conozca si el paquete llegó o no a su destino, debido a que este tipo de paquetes no implementan acuse de recibo, permitiendo que sólo el maestro envíe *broadcast* en una *piconet*.

Las conexiones ACL, ASB y PSB son entramadas, lo cual significa que es posible enviar datos más grandes que el tamaño de un paquete propiamente dicho, puesto que los datos serán distribuidos en múltiples paquetes. Estos también son asíncronos, lo cual significa básicamente que el ancho de banda no será reservado para estos.

Los datos sincrónicos son mapeados a paquetes SCO (opcionalmente eSCO en la especificación Bluetooth 1.2). Las conexiones SCO y eSCO son síncronas (Figura 3.2), no confiables y no susceptibles de ser empaquetados. En el establecimiento de la conexión son reservadas ranuras de tiempo para estos paquetes, y ningún otro dispositivo podrá enviar datos durante una ranura de tiempo reservada. Estas conexiones son principalmente usadas para enviar datos de audio en formato PCM a los sistemas de habla manos-libres y auricular [3].

[6] Ingeniería en Internet, “Una Especificación de Flujo Propuesto”, RFC 1363, Septiembre 1992.

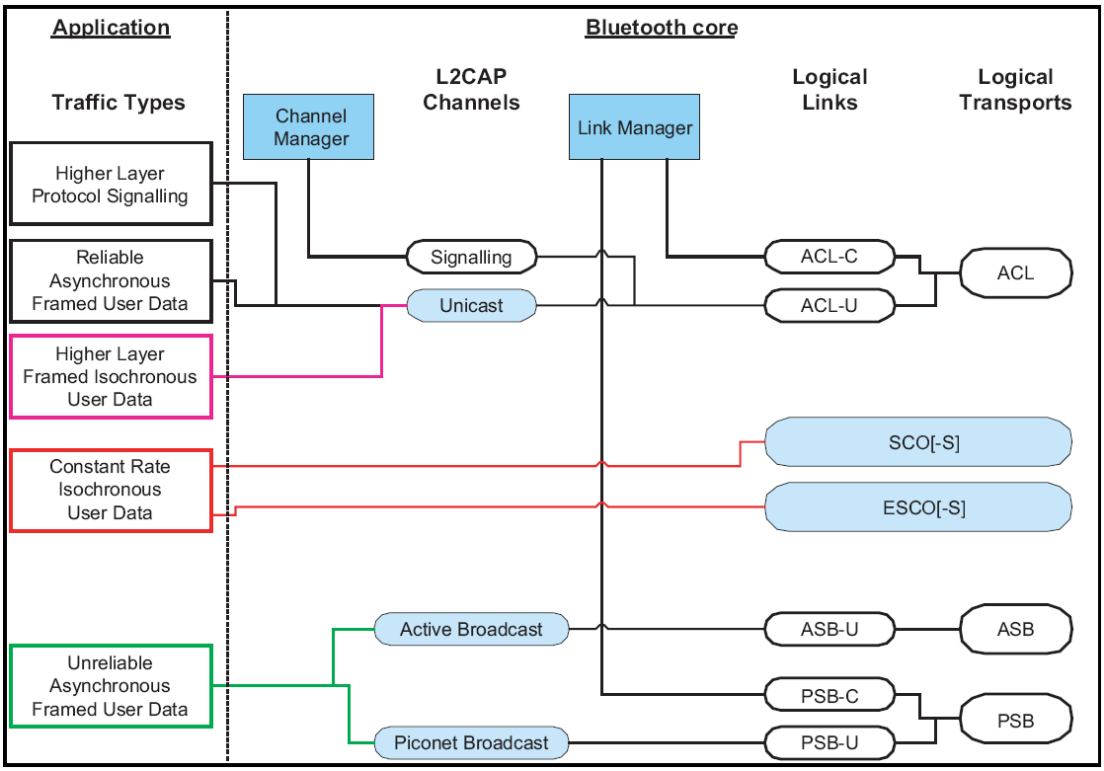


Figura 3.2 Portadores de tráfico Bluetooth [3].

El núcleo (*core*) de la especificación Bluetooth provee portadores de tráfico estándar para transportar datos de aplicaciones y protocolos de servicios, los cuales son mostrados en la Figura 3.2. Los portadores de tráfico Bluetooth que están disponibles para aplicaciones son mostrados en la Figura 3.2 representados por los rectángulos sombreados redondeados. Los tipos de tráfico de datos se enseñan en el lado izquierdo del diagrama, enlazados a portadores de tráfico que son típicamente los más convenientes para el transporte de tráfico de datos. Los enlaces lógicos son denominados usando los nombres de transporte lógico asociados y un sufijo que indica el tipo de datos que es transportado. (C para enlaces de control que portan mensajes LMP, U para enlaces L2CAP que portan datos de usuario (PDUs (*Protocol Data Unit a message*) L2CAP) y S para flujos (*streams*) que soportan datos síncronos y asíncronos sin formato).

Es común para el sufijo ser movido desde el enlace lógico sin introducir ambigüedad, así una referencia al transporte lógico ACL por defecto puede ser resuelta como enlace lógico ACL-C (*ACL Control*) en casos donde entre en conflicto el protocolo LMP, o enlace lógico ACL-U (*ACL User*) donde la capa L2CAP sufra el mismo conflicto.

El mapeado de tipos de tráfico de aplicaciones a portadores de tráfico Bluetooth se basa en emparejar las características del tráfico con las características de los portadores. Se recomienda utilizar dichos mapeados como el método más eficiente y natural de transporte de datos manteniendo sus características dadas.

Sin embargo, es posible que una aplicación (o implementación del sistema Bluetooth) pueda elegir un portador de tráfico diferente, o un mapeado diferente para lograr un resultado similar. Por ejemplo, en una *piconet* con solo un esclavo, el maestro puede elegir transportar *broadcast* L2CAP sobre enlaces lógicos ACL-U en vez de hacerlo sobre enlaces lógicos ASB-U (*ASB User*) o PSB-U (*PSB User*).

Esto será probablemente más eficiente en términos de ancho de banda (si la calidad del canal físico no se degrada). El uso de vías de transporte alternativo solo es aceptado si las características del tipo de trafico de la aplicación se conservan. Los tipos de tráfico de aplicaciones señalados en la Figura 3.2 son usados para clasificar los tipos de datos que pueden ser soportados por el sistema Bluetooth. El tipo de tráfico de datos original puede no ser el mismo tipo soportado por el sistema Bluetooth si un proceso intermedio lo modifica. Por ejemplo, los datos de video son generados a una tasa constante pero un proceso de codificación intermedio puede alterarlo variando su tasa [3].

3.2.1 Tráfico de datos en tramas

Los servicios de la capa L2CAP proveen un transporte orientado a tramas para datos de usuario síncronos y asíncronos. La aplicación convierte los datos a este servicio generando así tramas de tamaño variable (hasta un máximo negociado por el canal) y dichas tramas son entregadas en la misma forma a la aplicación correspondiente en el dispositivo remoto. No hay requisitos para que la aplicación inserte información de trama adicional en los datos, aunque puede hacerlo si así lo requiere (el entramado es invisible ante el sistema Bluetooth).

Los canales L2CAP orientados a conexión pueden ser creados para transportar datos *unicast* (punto a punto) entre dos dispositivos Bluetooth. Un canal L2CAP no orientado a conexión existe para datos *broadcast*. En el caso de la topología *piconet* el dispositivo maestro siempre es quien genera los datos *broadcast* y los dispositivos esclavos son los receptores. El tráfico en el canal L2CAP *broadcast* es unidireccional. Los canales L2CAP *unicast* pueden ser unidireccionales o bidireccionales.

Los canales L2CAP tienen un ajuste asociado a QoS que define ciertos parámetros en la entrega de tramas de datos. Dicho ajuste de QoS puede ser usado para indicar (por ejemplo) que los datos son asíncronos, por lo tanto tienen un tiempo de vida después de llegar a ser inválidos y que deben ser entregados en un periodo determinado de tiempo, o que los datos sean confiables y deban ser entregados sin errores. El gestor de canal L2CAP es responsable de arreglar el transporte de tramas de datos de un canal L2CAP en un apropiado enlace lógico banda base, posiblemente multiplexando esto en un enlace lógico banda base con otro canal L2CAP que posea características similares [3].

3.2.2 Tráfico de datos sin usar tramas

Si la aplicación no requiere entrega de datos en tramas, posiblemente es porque ésta incluye entramado en flujos (*streams*) o porque los datos son un flujo (*stream*) puro,

entonces puede evitar el uso de canales L2CAP y hacer uso directo del enlace lógico banda base.

El sistema Bluetooth soporta el transporte directo de aplicaciones de datos que son asíncronas y de tasa constante (de velocidad de transmisión en tasa de bits, o datos pre-tramados en tasa de tramas), usando a enlace lógico SCO-S (*Stream SCO (unframed)*) o eSCO-S (*Stream eSCO (unframed)*). Este enlace lógico reserva el ancho de banda del canal físico y provee una tasa constante de transporte sincronizada con el reloj de la *piconet*. Los datos son transportados en paquetes de tamaño fijo a intervalos fijos, donde estos parámetros son negociados en el establecimiento del canal. Los enlaces eSCO brindan la mejor elección en tasas de bits y confiabilidad, al momento de usar transmisiones limitadas en caso de error.

Las tasas de datos mejoradas son soportadas para eSCO, pero no para transporte lógico SCO. Los transportes lógicos SCO y eSCO no soportan enlaces lógicos multiplexados. Una aplicación puede elegir el número de flujos (*streams*) asociados a un *stream* SCO/eSCO, garantizando que los *streams* asociados son o aparentan ser, un *stream* de tasa constante.

Las aplicaciones pueden elegir el tipo de enlace lógico más apropiado disponible en banda base, crearlo y configurarlo para el transporte de flujo (*stream*) de datos y desocuparlo una vez completada la transmisión. (La aplicación normalmente usara un canal *unicast* L2CAP entramado para transportar su información de plano C a la aplicación ubicada en el dispositivo remoto). Si los datos de la aplicación son asíncronos y de tasa variable, entonces estos serán llevados a través de un canal *unicast* L2CAP, y por lo tanto serán tratados como datos entramados [3].

3.2.3 Confiablidad en los portadores de tráfico

Bluetooth es un sistema de comunicaciones inalámbricas. En ambientes RF de baja calidad, este sistema es considerado intrínsecamente no confiable. Para contrarrestar esto el sistema cuenta con niveles de protección en cada capa. Los encabezados de los paquetes banda base codifican con FEC (*Forward Error Correcting*) permitiendo la corrección de errores en recepción, además usan HEC[7] para detectar posibles errores después de la corrección. Ciertos tipos de paquetes banda base incluyen FEC en la carga útil. Adicional a esto, algunos tipos de paquetes banda base cuentan con CRC.

En transportes lógicos de ACL el resultado de algoritmos de detección de errores son usados para el manejo del protocolo simple ARQ (*Automatic Repeat Request*). Esto proporciona una confiabilidad mejorada al retransmitir paquetes que no superan el algoritmo de chequeo de errores en recepción. Es posible modificar este esquema para soportar paquetes sensibles al retardo descartando aquellos que fueron transmitidos sin éxito en la fuente de transmisión una vez haya expirado la vida útil del paquete. Los enlaces eSCO usan una versión modificada de este esquema para brindar confiabilidad permitiendo un número de retransmisiones limitado.

[7] Método de corrección de errores de cabecera, se especifica con mayor claridad en el volumen 2, parte B, sección 7.1.1 de la especificación Bluetooth 2.0 + EDR.

La confiabilidad resultante ofrecida por este esquema ARQ es tan efectiva como la capacidad de los códigos de detección de errores HEC y CRC. En la mayoría de los casos esto es suficiente, no obstante se ha demostrado que para los tipos de paquetes más grandes la probabilidad de no detectar un error es muy alta al soportar aplicaciones típicas, especialmente aquellas que transmiten gran cantidad de datos. La capa L2CAP brinda un nivel adicional de control de errores, el cual es diseñado para mostrar errores no detectados ocasionalmente en la capa banda base y solicitud de retransmisión de datos afectados. Esto proporciona un nivel de confiabilidad requerido por las aplicaciones más típicas de Bluetooth [3].

Los enlaces *broadcast* no tienen ruta alguna de realimentación, y son habilitados para usar el esquema ARQ (aunque el receptor aún pueda detectar errores de paquetes recibidos). En cambio cada paquete es transmitido varias veces con la esperanza que en recepción pueda llegar al menos una de las copias satisfactoriamente. A pesar de lo planteado, no existen garantías de recepción satisfactoria, considerando así estos enlaces como no confiables.

En resumen, si un enlace o canal se caracteriza por su confiabilidad esto implica que el receptor es capaz de detectar errores de paquetes recibidos y solicitar retransmisión hasta que los errores sean descartados. Debido al sistema de detección de errores usado, algunos errores no detectados pueden quedar en los datos recibidos.

Para canales L2CAP el nivel de éstos es comparable con otros sistemas de comunicaciones, aunque para enlaces lógicos el nivel de errores no detectados es un poco más alto. El transmisor puede descartar paquetes desde la cola de transmisión de tal forma que el receptor no reciba todos los paquetes en secuencia. Si esto sucede, la tarea de detección de paquetes que hacen falta se delega a la capa L2CAP.

En un enlace no confiable el receptor es capaz de detectar errores en paquetes recibidos pero no puede solicitar retransmisión. Los paquetes aprobados por el receptor pueden estar sin errores, pero no hay garantía que todos los paquetes en secuencia sean recibidos. Por lo tanto el enlace es considerado fundamentalmente no confiable.
Hay usos limitados para dichos enlaces, y estos usos normalmente dependen de la repetición continua de datos desde las capas superiores hasta que sea válido. Los enlaces de *streams* tienen características de confiabilidad tanto de enlaces confiables y no confiables, dependiendo de las actuales condiciones de operación [3].

3.2.4 Canales No Orientados A Conexión

Bluetooth no incluye canales no orientados a conexión actualmente como lo son soportados por otras tecnologías basadas en acceso a la conexión, tales como *Ethernet* u 802.11. Bluetooth permite un mecanismo no confiable de *broadcast* desde el maestro al número de esclavos conectados en ese momento a la *piconet*. Los paquetes *broadcast* no son reconocidos por los esclavos; por lo tanto el maestro no puede saber si el paquete *broadcast* fue recibido por cualquier esclavo.

El maestro puede repetir el envío de paquetes *broadcast* para mejorar la confiabilidad, pero el *stack* no filtra la retransmisión del primer fragmento de un paquete L2CAP. Esto conlleva a que múltiples fragmentos de paquetes L2CAP sean pasados a niveles superiores del *stack*.

Debido a la baja confiabilidad y creación de fragmentos duplicados de datos el uso de *broadcast* no es recomendado para el tráfico L2CAP, *broadcast* generalmente no es utilizado en la mayoría de aplicaciones Bluetooth. En términos de QoS, *broadcast* no orientado a conexión ofrece solamente flujo de datos en una dirección, sin garantizar QoS de cualquier tipo [20].

3.2.5 Tipos De Paquetes

Los paquetes usados en una *piconet* están relacionados con el transporte lógico que se esté usando para la comunicación. Tres transportes lógicos con distintos tipos de paquetes están definidos en la especificación Bluetooth: el transporte lógico SCO, eSCO, y el ACL. Para cada uno de éstos, existe una variedad de 15 tipos de paquetes que pueden ser definidos.

Para indicar los diferentes tipos de paquetes en el transporte lógico se usa un campo llamado Tipo con un código de cuatro bits. Los tipos de paquetes están divididos en cuatro segmentos. El primer segmento está reservado para paquetes de control, los cuales ocupan un solo slot o ranura de tiempo. El segundo segmento está reservado para paquetes que ocupan un slot único, mientras que el tercero y cuarto segmento están reservados para tipos de paquetes que ocupan 3 y 5 slots respectivamente.

Todos los paquetes con *payload* deben usar modulación GFSK (*Gaussian Frequency Shift Keying*) a menos que se especifique lo contrario. Los transportes lógicos ACL con Tasa de Datos Mejorada (EDR, Enhanced Data Rate) son específicamente seleccionados con el uso del protocolo LMP [3].

3.2.6 Canales Orientados A Conexión Síncronos (Paquetes SCO Y eSCO)

En los paquetes síncronos Bluetooth un *slot* se reserva en un intervalo fijo. En los paquetes síncronos SCO no hay retransmisión, y si un paquete no es recibido correctamente éste es descartado. En paquetes eSCO existe una ventana opcional de retransmisión, pero solamente hasta el siguiente *slot* reservado. En enlaces SCO el intervalo de encuesta es fijado por el tipo de paquete, mientras que en enlaces eSCO el intervalo de encuesta es negociable.

Los enlaces síncronos (SCO) sin retransmisión generan retardos fijos, con un valor cero en cuanto a la variación del retardo, pero una probabilidad finita de que los paquetes sean perdidos. Estas características son principalmente favorables para los flujos de datos que transporten audio donde hay solamente datos significativos por corta duración. Por lo tanto, si se pierden los paquetes, la aplicación de recepción tiene que aceptar la perdida y procurar reconstruir o sustituir los datos perdidos.

Los enlaces síncronos (eSCO) con retransmisión, permiten retransmisión limitada durante un tiempo hasta que el siguiente paquete deba ser retransmitido. En un enlace esto es equivalente a que un *flush timeout* sea igual al intervalo de encuesta.

Los paquetes SCO y eSCO no utilizan la capa L2CAP y se tratan por separado en la especificación Bluetooth; por lo tanto los parámetros de QoS L2CAP no se aplican a los datos síncronos (pero los mismos parámetros podrían ser fácilmente llevados a canales síncronos) [20].

3.2.7 Canales Orientados A Conexión Asíncronos (Paquetes ACL)

Los paquetes ACL satisfacen el tipo de conexión soportado por el perfil PAN y encapsulación de la información a través de BNEP que a su vez, es recibido por la capa L2CAP, definiendo así los tipos de paquetes ofrecidos en la Tabla 3.1.

Tipo	Encabezado Carga Útil (bytes)	Carga Útil (bytes)	FEC	CRC	Máx. Tasa Simétrica (Kbps)	Máx. Tasa Simétrica (Kbps)	
						Adelante	Atrás
DM1	1	0 - 17	2/3	si	108,8	108,8	108,8
DH1	1	0 - 27	no	si	172,8	172,8	172,8
DM3	2	0 -121	2/3	si	258,1	387,2	54,4
DH3	2	0 - 183	no	si	390,4	585,6	86,4
DM5	2	0 - 224	2/3	si	286,7	477,8	36,3
DH5	2	0 - 339	no	si	433,9	723,2	57,6
AUX1	1	0 - 29	no	no	185,6	185,6	185,6

Tabla 3.1 Tipos de paquete ACL [3].

Los paquetes ACL (Tabla 3.1) Bluetooth pueden tener diferentes tamaños. El paquete más pequeño, DM1 (*Data - Medium Rate Data packet type for medium rate*), transporta hasta 17 bytes de datos de usuario y usa una suma de verificación (FEC) de 2/3 para confiabilidad del mismo; usando una ranura de tiempo. El paquete DH1 (*Data-High Rate Data packet type for high rate data*) transporta hasta 27 bytes en una ranura de tiempo, omitiendo los bits correspondientes al FEC de 2/3. Los paquetes también pueden abarcar múltiples ranuras de tiempo: Cada paquete DM3 y DH3 usa 3 ranuras de tiempo, mientras que los paquetes DM5 y DH5 usan 5. Éstos existen en modos simétricos y asimétricos: el modo simétrico permite tanto al maestro como al esclavo enviar paquetes igual de grandes.

En el modo asimétrico, solo el maestro envía grandes paquetes, mientras que el esclavo está limitado a enviar paquetes de una ranura de tiempo. Esto aumenta el flujo del ancho de banda del maestro y disminuye el del esclavo. Los valores en la Tabla 1 son, por supuesto, valores teóricos sin el encabezado del paquete. Por ejemplo, en un enlace ACL usando DH5, se pueden enviar alrededor de 300 o 320 Kbps de datos de usuario UDP [3].

Bluetooth está diseñado fundamentalmente para el uso de dispositivos en estados conectados. Debido al soporte de QoS brindado por L2CAP existen factores que afectan directamente un canal ACL que deben ser considerados:

- Intervalo de encuesta - éste es el intervalo máximo en el cual el maestro de la *piconet* debe transmitir a un esclavo, permitiendo que el esclavo responda. Este intervalo es por lo tanto un factor determinante en el retardo máximo de transmisión de una aplicación. El retardo máximo también es afectada por otros factores por ejemplo la perdida de datos y ruido del radioenlace.

- Tipo de paquete - la mayoría de los paquetes contienen un *cheksum* CRC para comprobar errores, y los paquetes DM incluyen FEC para la corrección de errores. Los dispositivos que soportan Tasa de Datos que Maneja Calidad del Canal (CQDDR, *Channel Quality Driven Data Rate*) intentaran elegir el tipo de paquete para la tasa de datos máxima, bajo condiciones de radiocomunicación actuales.

- Modo de retransmisión - los paquetes ACL con CRC pueden ser retransmitidos en un período definido por el *flush timeout*.

- *Flush Timeout* - éste define por cuánto tiempo se retransmite un paquete y es declarado en banda base. Un *flush timeout* infinito significa que un paquete está siendo retransmitido continuamente hasta que se reciba correctamente. Con *timeout* finito, se desechan los paquetes si no transmitieron con éxito dentro del intervalo *timeout*.

- Modos de ahorro de energía (*Park, Hold* y *Sniff*). Estos modos permiten que los dispositivos estén ausentes de conexión, por lo tanto el retardo aumenta y reduce el ancho de banda para conservar energía. El cambio a estos estados es iniciado desde un estado de conexión normal. El uso de modos con baja potencia pueden ser iniciados por el maestro o el esclavo en cualquier momento durante una conexión. En general, cada dispositivo es responsable de sus propios modos de potencia, esperando que los dispositivos con baterías soliciten dichos modos cuando sea posible [20].

Al hablar de QoS en Bluetooth, deben tenerse en cuenta ciertos parámetros que definen el préstamo de éste:

- Ancho de Banda
- Retardo
- Tasa de Error (corregida)

Sin embargo, para cada uno de estos parámetros, es posible especificar límites en la variación de éstos, por ejemplo variación del retardo. Es también útil para sistemas que ofrecen como servicio el tener cierta información sobre variación en los parámetros, por ejemplo la pérdida del flujo de datos afecta el *buffering.*

Aunque hay solamente algunos factores que definen el uso de QoS, claramente existen diferentes maneras de usar una radiocomunicación para optimizar uno o más parámetros de QoS. Estos factores están ligados; por lo tanto hay un grado de libertad limitado al mejorar un factor sin afectar contrariamente a otro. Por ejemplo, si se requiere una tasa de error cero entonces puede dar como resultado un retardo infinito porque un paquete puede ser transmitido indefinidamente bajo condiciones de radio adversas.

Diferentes tecnologías de radiocomunicaciones tienen capacidades diferentes para brindar niveles de QoS. El mayor desafío es detallar los rangos de QoS que se puedan especificar de manera universal, tales que cualquier tecnología de radiocomunicación pueda implementar los servicios requeridos apropiadamente [19].

Cuando se establece un enlace ACL la configuración se distribuye en dos niveles:

- Banda Base: los tipos de paquetes, se negocia el intervalo de encuesta y *flush timeout*.

- L2CAP: L2CAP define dos niveles de servicio que son Mejor Desempeño y Garantizado. Mejor desempeño significa que no hay QoS de ningún tipo, mientras que Garantizado se define usando cinco parámetros adicionales de QoS:

    - *Token Rate*.
    - Tamaño del *Token Bucket*.
    - Ancho de banda máximo.
    - Retardo.
    - Variación del retardo.

Figura 3.3 Formato opcional de QoS que contiene la especificación del flujo [3].

Los campos de datos opcionales son los siguientes:

- *Flags* (8 bits): reservado para uso futuro y debe ser establecido con valor 0 e ignorado por el receptor.

- *Service type* (8 bits): este campo indica el nivel de servicio requerido. El valor por defecto es el de mejor desempeño, como también puede ser garantizado, sin tráfico o reservado. Si es dejado en el estado por defecto los parámetros restantes se deben tratar como opcionales en el dispositivo remoto, pero si es configurado sin tráfico los campos restantes deben ser ignorados por qué no hay datos enviados a través del canal.

- *Token Rate* (4 bytes): el valor de este campo representa la rata de datos promedio establecida por la aplicación. La aplicación debe enviar datos a una rata continua. En escala de tiempos cortos la aplicación puede enviar datos excediendo la rata de datos promedio, dependiendo del tamaño del *token bucket* y el ancho de banda máximo.

- Tamaño del *token bucket* (4 bytes): este campo especifica el límite de sobrecarga con la cual la aplicación puede transmitir datos. La aplicación puede ofrecer una carga de datos igual al tamaño instantáneo del *token bucket*, limitado por el ancho de banda máximo.

- Ancho de banda máximo: el valor de este campo, limita cuán rápido los paquetes son enviados desde la aplicación en una conexión extremo a extremo. Algunos sistemas pueden tomar ventaja de esta información, dando como resultado una asignación de recursos más eficiente.

- Latencia: el valor de este campo es el retardo máximo aceptable de un paquete L2CAP una vez es enviado desde banda base.

- Variación del retardo: el valor de este campo es la diferencia, en microsegundos, entre el máximo y el mínimo retardo posible de una comunicación entre 2 capas L2CAP [3].

3.3 MÉTRICAS DE FUNCIONAMIENTO

Debido a la prioridad de flujo y a los picos de tráfico, los equipos de red pueden perder paquetes de datos y producir retardos en la transmisión. Estos paquetes perdidos son retransmitidos y de este modo no se pierde información. Mientras las aplicaciones de datos no suelen verse afectadas, VoIP, es muy sensible a dichas pérdidas. Concretamente, se ve muy afectada a partir de un 5% de paquetes perdidos.

Una amplia lista de métricas para la evaluación del desempeño en TCP se describe en la primera parte del RFC Métricas para la Evaluación de Mecanismos de Control de Congestión (MRG, *Metrics for the Evaluation of Congestion Control Mechanisms*), este borrador (*draft*) plantea algunas métricas usadas comúnmente las cuales son descritas en

dicho documento. Basándose en el RFC y clasificando las métricas en métricas de red y métricas de uso, fueron seleccionadas las siguientes [21].

3.3.1 Throughput

Definido como la capacidad de un enlace de transportar información útil, representa a la cantidad de información útil que puede transmitirse por unidad de tiempo. El *Throughput se* puede medir como métrica basada en *routing* (enrutamiento) de uso del enlace, como métrica basada en flujo de tiempos de transferencia por conexión y como métrica basada en funciones de usuario para uso general o tiempos de espera del usuario. En la mayoría de los mecanismos de control de congestión el maximizar dicho *Throughput* es lo más importante, conforme a la demanda de aplicaciones y variaciones de otras métricas [21].

3.3.2 Retardo

El factor retardo, definido como el tiempo de tránsito de los paquetes desde el origen al destino y vuelta, influye directamente la QoS. A partir de cierto umbral puede empezar a ser incómodo mantener una conversación. Para una calidad alta el retardo debería mantenerse por debajo de 150ms. Para una calidad media, por debajo de 400ms.

Como el *Throughput*, el retardo puede ser medido como métrica basada en *routing* del retardo en la cola sobre tiempo, o como métrica basada en flujo en términos de tiempo de transferencia por paquete. Para transferencia confiable, el tiempo de transferencia por paquete considerado por la aplicación incluye el posible retardo de retransmisión de un paquete perdido [21].

3.3.3 Jitter

El *jitter* se define como la variación del tiempo de tránsito de los paquetes. No todos los paquetes sufren un retardo constante. Este retardo variable o *jitter* disminuye la calidad de la voz al superar el umbral de los 50ms.

El *jitter* es absolutamente importante para el tráfico sensible al retardo, tal como voz y vídeo. Un *jitter* grande requiere un tamaño de almacenamiento mayor en recepción y puede causar altas tasas de pérdida en requerimientos que exijan un retardo único [21].

3.3.4 Tasa de Pérdida

La tasa de pérdida de paquetes puede ser medida como una métrica basada en red o basada en flujo de datos, para obtener estadísticas de red, un método es medir la tasa de pérdida en la cola de embotellamiento. Para el tráfico fluyente e interactivo, la alta pérdida de paquetes resulta, al fallar en recepción el descifrar dicho paquete. El paquete recibido se considera perdido si su retardo es mayor que un umbral predefinido.

Basado en parámetros de QoS al prestar servicios de voz sobre IP, se llega a niveles de tolerancia dependiendo de la calidad que se requiera (Tabla 3.2) [21].

	Calidad Alta	Calidad Media	Calidad Baja
Pérdida de paquetes	1%	3%	5%
Retardo	150 mseg	400 mseg	600 mseg
Jitter	20 mseg	50 mseg	75 mseg

Tabla 3.2 Niveles de Tolerancia en VoIP [22].

En relación a estas métricas se debe tener en cuenta un factor muy importante, como es el ancho de banda (Tabla 3.3), donde dicho requisito dependerá del número de comunicaciones simultáneas que se quieran soportar. En un entorno LAN, donde las velocidades de transmisión van desde 10Mbps a 100Mbps, se permite el uso de G.711 con un ancho de banda de 84,7Kbps. En escenarios donde el ancho de banda es más escaso se puede elegir G.723 con un ancho de banda de 27,2Kbps o el G.729 con un ancho de banda de 28,8Kbps [22].

Códec de Audio	Ancho de banda comprimido	Ancho de banda paquetizado	Ancho de banda en Ethernet
G.723	6,3 Kbps	17 Kbps	27,2 Kbps
G.729	8 Kbps	24 Kbps	28,8 Kbps
G.711	64 Kbps	74,6 Kbps	84,7 Kbps
FAX	4,8 Kbps	12,8 Kbps	20,4 Kbps

Tabla 3.3 Ancho de Banda requerido por CODECs [22].

El ancho de banda puede reducirse en un 35% cuando se utiliza detección de silencios (VAD) y en tecnologías inalámbricas que demandan mayor ancho de banda, los *routers* (enrutadores) pueden utilizar la compresión de cabeceras IP (cRTP) para reducir las cabeceras de 40 a 24 bytes, llegando dicha reducción hasta 16,41kbps en el caso del G.723 [22].

Finalmente, la tecnología Bluetooth permite su interconexión con capacidades IP haciendo uso del perfil PAN, el cual se soporta en el protocolo BNEP y enfatizando el tipo de servicio que se quiere prestar como es el de voz, los canales más apropiados que brinda Bluetooth para este tipo de transporte que debe ser paquetizado son los ACL, que al ser tratados en la capa L2CAP ofrece un factor de vital importancia como es la QoS.

Los canales ACL brindan confiabilidad en presencia de interferencia, incluso cuando hay generación de grandes ráfagas de información. El retardo generado por enlaces ACL al retransmitir es pequeño, por ejemplo un ACK (*Acknowledge*) puede ser recibido dentro de los 1.25mseg siguientes. Esto genera la posibilidad de mejorar la retransmisión en aplicaciones sensibles al retardo, tales como aplicaciones de tiempo real y flujos (*streams*) de audio o video. El periodo de retransmisión puede ser configurado con la opción *Flush*

Timeout. Además, los enlaces ACL brindan calidad en presencia de interferencia respecto de los enlaces SCO.

Igualmente, los enlaces ACL pueden soportar anchos de banda variables y asimétricos requeridos por ciertas aplicaciones. Estas son las principales ventajas de usar enlaces ACL en vez de los SCO para aplicaciones de tiempo real. Los tipos de servicios garantizados por enlaces ACL brindan retardos y anchos de banda que se requieran, definiendo así ciertos parámetros que permitan la evaluación del comportamiento del enlace, tales como el retardo, ancho de banda, perdida de paquetes y *jitter*, a fin de prestar servicios de voz en vías de la validación y evaluación de la arquitectura propuesta en el capítulo 4.

CAPÍTULO 4 ARQUITECTURA PARA LA PRESTACIÓN DE SERVICIOS DE VOZ BAJO CRITERIOS Y PARÁMETROS DE INTEROPERABILIDAD DE LA TECNOLOGÍA BLUETOOTH CON REDES IP CABLEADAS ETHERNET.

Este capítulo presentará el diseño global de la arquitectura, formulado en base a un análisis de los requerimientos del proyecto, las metas deseadas y los criterios de interoperabilidad de tecnologías. La arquitectura propuesta mantiene todos los actores clásicos de un sistema VoIP, y agrega los componentes necesarios para soportar la infraestructura IP sobre los dispositivos móviles para la tecnología inalámbrica Bluetooth.

4.1 Diseño

Las redes de área personal, centrado sobre la tecnología inalámbrica Bluetooth, es vista como la dirección a tomar por muchos de los fabricantes de dispositivos móviles debido a los bajos costos y su alta penetración en el mercado. Una característica específica de la especificación Bluetooth v2.0 es la adición de una interfaz para *networking*, la cual permite la implementación de IP sobre una encapsulación de red Bluetooth. BNEP es soportado en la mayoría de los sistemas operativos actuales de dispositivos móviles.

La siguiente sección contiene una descripción representativa de una sesión telefónica VoIP, sus actores y componentes, con el fin de poner en perspectiva los requerimientos funcionales para un cliente VoIP en sistemas operativos específicos de los dispositivos móviles.

Una de las prioridades de la construcción de la arquitectura es crear una conexión IP transparente desde la fuente hasta el destino, cuyos datos viajen a través de un enlace inalámbrico Bluetooth y a través del enlace provisto por la red cableada IP tradicional *Ethernet*. Esta conexión tiene que ser independiente de la carga de paquete IP real y simétrica sobre el enlace Bluetooth con el objetivo de realizar el *tunneling* de los datos bidireccionalmente a los respectivos puntos de comunicación; todo esto partiendo de la premisa de interconectividad con redes cableadas IP tales como *Ethernet*.

Como en un sistema VoIP tradicional, los clientes deben estar registrados en un servidor de VoIP tal como un servidor H.323 o SIP (*Session Initiation Protocol*) para que éstos puedan establecer y realizar llamadas telefónicas entre ellos. En el caso de una red SIP, se debe establecer una sesión SIP típica entre un Cliente Usuario-Agente inicializador (UAC, *User Agent Client*) y un Servidor Usuario-Agente receptor (UAS, *User Agent Server*). El protocolo SIP realiza la señalización para establecer la conexión e intercambiar parámetros de conexión de datos, mientras que los datos de voz reales son intercambiados con el protocolo RTP.

SIP no fue el primero, ni es el único, protocolo de VoIP usado hoy en día (H.323, MGCP (*Media Gateway Control Protocol*), IAX (*Inter-Asterisk eXchange*), entre otros), pero actualmente éste parece ser el más popular entre los fabricantes de hardware. Las

ventajas del protocolo SIP radica en la amplia aceptación y la flexibilidad arquitectural (conocido como simplicidad), razón por la cual este protocolo es seleccionado como un componente de la arquitectura.

Estos elementos de llamadas VoIP son mantenidos en la arquitectura propuesta, por lo tanto, para mantener compatibilidad con los clientes de telefonía VoIP genéricos tales como los encontrados en los escenarios de comunicaciones actuales.

4.1.1 SIP y RTP

Se debe entender que SIP es un protocolo de señalización en la capa de aplicación que usa el "bien conocido" puerto 5060 para las comunicaciones mientras que RTP es usado para transmitir datos multimedia (por ejemplo voz) entre los puntos finales. Una topología común para ilustrar la relación entre SIP y RTP, es comúnmente nombrada como el "trapezoide SIP" mostrado en la Figura 4.1. Cuando el usuario A quiere llamar al usuario B, el teléfono de A contacta su servidor Proxy, y el Proxy trata de encontrar al usuario B (a menudo conectándolo a través de su proxy). Una vez los teléfonos han establecido la llamada, ellos se comunican directamente entre ellos (si es posible), así los datos no tienen que ocupar los recursos del Proxy [23].

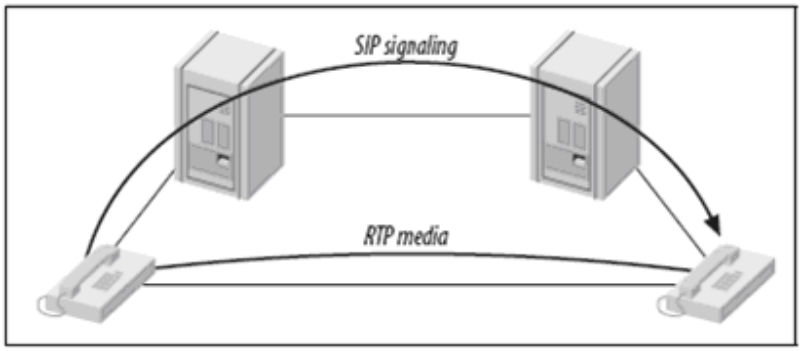


Figura 4.1 El trapezoide SIP [23].

La Figura 4.2 ilustra la arquitectura global proyectada y los componentes individuales necesarios para brindar soporte a comunicaciones de clientes VoIP funcionales dentro de un marco de interoperabilidad de las tecnologías. Específicamente, la aplicación necesaria en cada uno de los extremos para el establecimiento de la comunicación (*softphones*) no hace parte de la arquitectura planteada, sin embargo se incluye en el diagrama representativo debido a la evidente interacción presentada por parte de éste con los demás elementos de la arquitectura. En las siguientes secciones se describen con mayor profundidad cada uno de los componentes de la arquitectura así como las consideraciones hechas respecto a las aplicaciones necesarias.

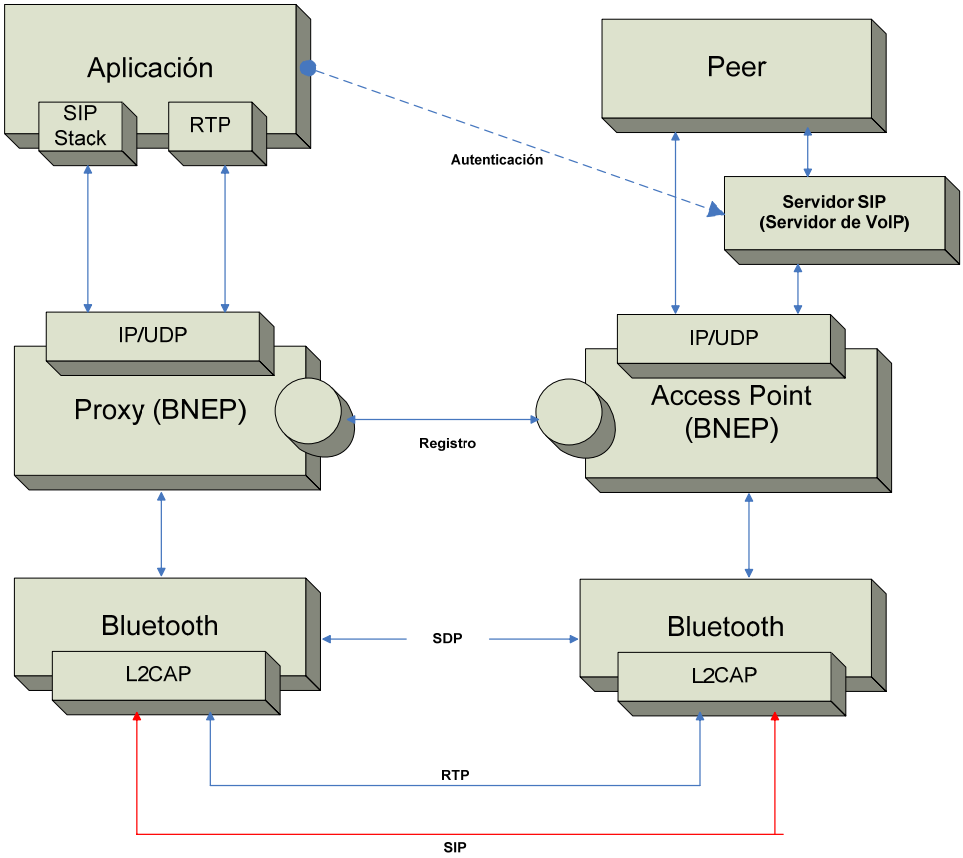


Figura 4.2 Arquitectura propuesta.

4.2 Registro

Para la futura implementación de servicios de voz que se soporten sobre la arquitectura propuesta, ésta debe mantener a los protocolos de capa superiores tales como SIP y RTP inadvertidos del "puente" que hace posible la interoperabilidad entre Bluetooth y la red cableada IP.

Así, la entidad o protocolo seleccionado para que sea posible la transición de los paquetes del mundo inalámbrico Bluetooth hacia el cableado *Ethernet* es BNEP, el cual permite la creación de flujos de datos independientes entre los puntos finales de la conexión. El protocolo BNEP se encarga entonces de los procesos de encapsulado, desencapsulado y manejo de cabeceras de tecnología de cada uno de los paquetes como se explicó en detalle en el capítulo 2.

Paquetes con fuente o destino desconocidos o no registrados deberían ser desechados para minimizar la congestión del puente.

4.3 Punto de Acceso

Para comunicarse con otros terminales pertenecientes a la red cableada *Ethernet*, el usuario móvil requiere un punto de acceso conectado a la red IP cuyo propósito sea el de reenviar datos desde y hacia el dispositivo de usuario. Este punto de acceso corresponde a uno de los dos puntos claves para el enlace inalámbrico Bluetooth, localizado en una terminal con conectividad hacia la red IP o en su defecto hacia Internet.

El punto de acceso es un nivel de abstracción por encima del protocolo IP, y no necesita interpretar el contenido real del paquete para funcionar apropiadamente sino realizar el puente en la brecha entre las interfaces Bluetooth e IP tan transparentemente como sea posible.

Existen dos formas para realizar esta función, representadas por dos perfiles diferentes en la especificación Bluetooth, cada uno con su respectivo soporte protocolar. Sin embargo la opción seleccionada establece el uso del perfil PAN con su protocolo de soporte BNEP. La Figura 4.3 detalla la diferencia entre las dos formas mencionadas [24] [25].

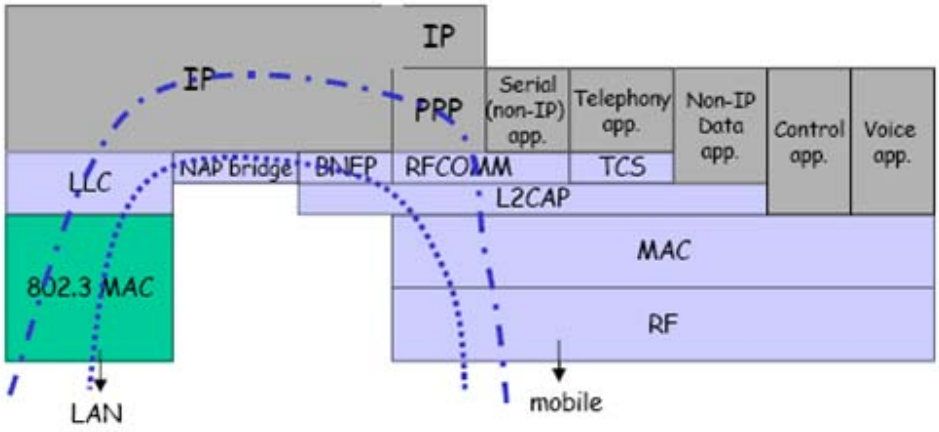


Figura 4.3 Estructura protocolar para soluciones de IP sobre Bluetooth [25].

El perfil de acceso a LAN incrementa el número de bytes por cabecera de paquetes (ver Figura 4.3) lo que tiene un impacto considerable en el ancho de banda consumido por comunicación (disminución del Throughput por sobrecarga de cabeceras), por consiguiente esto afecta la comunicación de voz especialmente las que utilizan CODECs con requerimientos significativos de ancho de banda tales como G.711.

Mientras que la interfaz Bluetooth ofrece una conexión inalámbrica regular para el dispositivo móvil, la interfaz del punto de acceso conectada a la red IP será el componente que comunica directamente a éste con los usuarios finales requeridos, determinados por el servidor SIP en el caso de señalización, o el cliente VoIP en caso de datos de audio RTP. El dispositivo móvil se auto-registrará en el servidor SIP, tal como se

realiza en sistemas de VoIP estándar, para posterior trámite de llamadas por parte de éste.

Para que el punto de acceso sea identificable por los dispositivos móviles, éste tendrá que publicar un servicio SDP o PAN conforme a la especificación Bluetooth. El punto de acceso no proveerá ningún otro servicio diferente al *tunneling* de datos altamente específico [24] [25].

4.3.1 Resolución de direcciones

Una de las funciones más importantes que deben ser incorporadas dentro de las capacidades del punto de acceso es la de resolución de direcciones.

Para lograr la interconexión entre los usuarios pertenecientes a la red Bluetooth con los usuarios pertenecientes a la red cableada *Ethernet*, el punto de acceso debe ser capaz de referenciarlos indistintamente. Esto es posible gracias a la incorporación del protocolo BNEP dentro del *stack* manejado por el punto de acceso. Así el punto de acceso debe estar provisto con una tabla de enrutamiento donde estarán registradas las interfaces de salida por la cual los paquetes deben ser reenviados dependiendo del usuario destino.

El punto de acceso conectado a la red Bluetooth sirve como un enrutador de interfaz dual para todos los paquetes entre el móvil y el destino objetivo. Esta característica puede ser implementada utilizando NAT (*Network Address Translation*), el cual permite intercambiar paquetes entre dos redes que se asignan mutuamente direcciones incompatibles y que consiste en convertir en tiempo real las direcciones utilizadas en los paquetes transportados [25].

4.4 Proxy (BNEP)

El segundo punto clave final para el enlace inalámbrico está localizado en el dispositivo móvil en sí. En lugar de incluir la funcionalidad Bluetooth directamente en el cliente VoIP, la segregación de funcionalidades interactuando entre ellas es una opción más favorable pensando en los servicios que debería soportar la arquitectura. En general debido a que una *piconet* puede consistir de 8 dispositivos (1 maestro y 7 esclavos), los dispositivos móviles pueden conectarse a más de un punto de Acceso a la vez. Sin embargo, las posibilidades de ejecutar múltiples conexiones y escaneos se deben evaluar particularmente en cada una de las implementaciones del *stack* Bluetooth posibles. Por ejemplo en el *stack Bluez*, el "demonio[8]" utilizado es *pand*, el cual permite la creación de solo una conexión por "demonio"; esto significa que si se quiere establecer conexiones con dos puntos de acceso se necesitarían correr dos *pand*.

Consideraciones especiales de funcionamiento simultáneo de estos "demonios" o procesos dependiendo del sistema operativo y la implementación del stack Bluetooth particular del dispositivo móvil deben ser tenidas en cuenta al momento de

[8] Demonio: proceso en terminología de sistemas operativos Unix.

implementación de servicios basados en la arquitectura. Por ejemplo cuando uno de los demonios establece una conexión con un punto de acceso, el otro constantemente trata de descubrir y conectarse a un punto de acceso interfiriendo de algún modo con la conexión ya establecida [26].

4.5 Conexiones Bluetooth

Como se describió en el capítulo 2, el perfil PAN brinda capacidades de red a los dispositivos Bluetooth para lo cual utiliza el Protocolo de Encapsulamiento de Red BNEP transportando directamente los datos sobre la capa de protocolo L2CAP de Bluetooth, haciendo posible que la red *Bluetooth* se comporte y forme parte de una red TCP/IP.

La conectividad ofrecida por Bluetooth, especialmente por los enlaces de las capas L2CAP del estándar, son explicadas en detalle en el capítulo 3, lo que complementa la definición del componente de la arquitectura tratado en éste ítem [26].

4.6 SDP

Una de las partes fundamentales de la arquitectura es la definición de la forma como los dispositivos móviles iniciarán las conexiones con los puntos de acceso disponibles; es decir la forma como dichos dispositivos buscarán y definirán el punto de acceso ideal, u óptimo para establecer una conexión confiable que permita la adecuada prestación de los servicios de voz.

El dispositivo móvil debe regularmente escanear dispositivos Bluetooth los cuales ofrezcan sus servicios como *gateways* hacia la red IP. Como se mencionó anteriormente el *stack* Bluetooth de Linux, *Bluez* contiene un “demonio” PAN completamente funcional, llamado *pand*. Windows también tiene soporte para PAN.

Para crear una conexión PAN, un extremo de la comunicación necesita escuchar las conexiones entrantes, y el otro extremo necesita inicializar la conexión. La elección obvia del dispositivo que debe escuchar es Punto de Acceso, mientras que el dispositivo móvil inicia la conexión. Inicialmente el dispositivo móvil puede simplemente escanear en busca de dispositivos que se encuentren dentro de su rango y una vez encontrados, indagar sobre el ofrecimiento de conexión PAN requerido para el establecimiento de la respectiva conexión [27].

4.7 Aplicación

Como se mencionó anteriormente, el componente de aplicación no hace parte de la arquitectura definida pero dada su inminente interacción con ésta, se evalúan las posibilidades y requerimientos necesarios para la correcta interacción con los demás componentes de la arquitectura.

Dicha aplicación tiene dependencia directa del *stack* Bluetooth implementado (propietario o abierto) y sistema operativo del dispositivo móvil sobre el cual se va a ejecutar, es decir, el principal requerimiento que se debe satisfacer es la incorporación del perfil PAN dentro de los perfiles soportados tanto por el sistema operativo como por el dispositivo móvil escogido.

La selección del protocolo BNEP como base para la interoperabilidad de tecnologías integradas en la arquitectura se hace basada en varias razones:

- Establecimiento como el estándar de facto dentro de la industria de dispositivo con tecnología Bluetooth incorporada.
- Estandarización completa y aprobada por organismos de peso tales como IEEE.
- Rápida adopción en el mercado de las dispositivos móviles, o dispositivos inteligentes, *smartphones*.
- Completa interoperabilidad con versiones subsiguientes del estándar Bluetooth.

4.7.1 Sistemas operativos

Dentro de los sistemas operativos con los que están provistos los dispositivos móviles actuales que incorporan el perfil PAN tenemos:

4.7.2 Windows Mobile

Actualmente existe un creciente número de dispositivos con Windows Mobile incorporado [28]. Estos dispositivos típicamente ofrecen suficiente capacidad de computación para VoIP, una interfaz de usuario “amigable”, librería para conexiones hacia Internet. Por ejemplo el dispositivo Hewlett-Packard iPAQ rx3715 [29] con la nueva versión *Windows Mobile 6*, incluye el soporte necesario del perfil PAN de Bluetooth.

Sin embargo una de las desventajas es el escaso número de aplicaciones de VoIP libres diseñadas para este sistema operativo, el software disponible es en su mayoría de código cerrado y de uso comercial.

4.7.3 Symbian OS (*Operating System*)

Symbian [30] es probablemente el sistema operativo más popular para los teléfonos inteligentes o *smartphones* [31]. Con la premisa de mantener su propósito principal de ofrecer telefonía móvil bajo la específica necesidad de bajo consumo de baterías, se han incorporado capacidades computacionales cada vez más potentes con el fin de proveer una plataforma para el desarrollo de aplicaciones y servicios. Actualmente se están emprendiendo grandes esfuerzos para crear un *framework*[9] de VoIP de fuente abierta. Por ejemplo, usando *GnuBox* [32] es posible usar una conexión de IP sobre Bluetooth.

[9] Framework: Una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

En su versión más reciente, *Symbian* OS [33] presenta características que favorecen la funcionalidad del sistema operativo dentro de la arquitectura. Sus características se presentan a continuación:

Symbian OS Version 9.3

- Personal Area Networking
- Bluetooth stereo headset support
- Bluetooth v2.0 (L2CAP, RFCOMM, SDP, GAP and SPP (*Serial Port Profile*))
- Bluetooth PAN-U and PAN GN
- IrDA
- USB (*Universal Serial Bus*) v2.0 High Speed (Mass storage, ACM (*Abstract Control Model*), WHCM (*Wireless Handset Control Model*))
- Serial
- Obex over Bluetooth, IrDA and USB
- PC Connectivity
    - o agenda and contacts sync framework
    - o file transfer
- OMA Data synchronization v1.2

4.7.4 Linux

Actualmente empresas del sector de los dispositivos móviles inteligentes han empezado a ofrecer sus productos en todo el mundo. Por ejemplo Motorola ofrece teléfonos móviles basados en Linux. No obstante el ambiente Linux en el teléfono puede variar significativamente con respecto al ambiente usual de escritorio Linux. Por ejemplo, la mayoría de las aplicaciones gráficas para Linux están escritas para el sistema *X-Windows*, mientras que la mayoría de los teléfonos generan los gráficos directamente sobre el *buffer* de marco del dispositivo. Sin embargo, esto no es un problema si se piensa que muchas aplicaciones pueden ejecutarse como aplicaciones de línea de comandos. Una gran ventaja de dichos dispositivos basados en Linux es la gran flexibilidad y las ventajas bien conocidas de las aplicaciones de código abierto. De este modo muchas aplicaciones VoIP pueden ajustarse para funcionar y satisfacer los requerimientos del bloque de aplicación de la arquitectura [34].

4.8 PROCEDIMIENTO DE CONEXIÓN

El procedimiento necesario para establecer una conexión entre los usuarios móviles y el punto de acceso (Figura 4.4) es:

1. Realizar una búsqueda de dispositivos Bluetooth disponibles (Inquiry).
2. De los dispositivos encontrados, escoger uno de ellos para conexión, que no haya sido seleccionado. Si no se encuentran dispositivos, regresar al paso 1.
3. Usar SDP para indagar sobre el ofrecimiento de conectividad para perfil PAN con rol NAP. Si el dispositivo no ofrece tal conectividad regresar al punto 2.
4. Conectarse al dispositivo. Si la conexión falla, regresar al punto 2.
5. Si la conexión se rompe, regresar al punto 1.

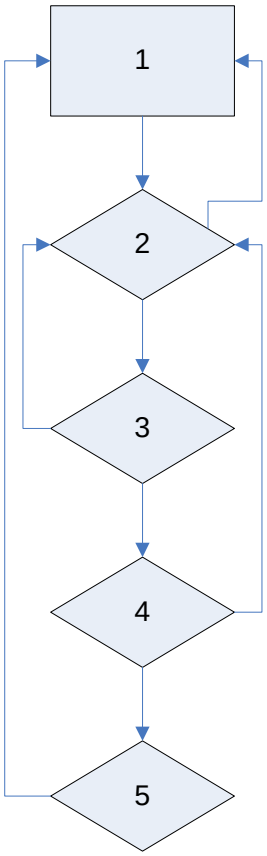


Figura 4.4 Proceso de establecimiento de conexión de los dispositivos móviles.

Una vez establecida la conexión Bluetooth con capacidades IP, se procede a realizar los procedimientos regulares para el establecimiento de una llamada telefónica VoIP, que pueden ser encontrados en [14].

5. RESULTADOS DE LA VALIDACIÓN DE LA ARQUITECTURA PLANTEADA PARA OFRECER SERVICIOS DE VOZ QUE REQUIERAN ALTA MOVILIDAD Y RÁPIDA LOCALIZACIÓN DE PERSONAL.

5.1 NECESIDAD DE LA SIMULACIÓN

Como una relativa nueva tecnología, muchos de los potenciales de Bluetooth están siendo explorados y todavía tiene un tópico activo de investigación. Una buena herramienta de simulación es esencial para conducir una investigación fructífera en cualquier área y en particular para ganar un más profundo entendimiento acerca de los elementos que se involucran en la prestación de servicios de voz sobre redes Bluetooth que se interconectan con redes IP tradicionales.

Para desarrollar tales proyectos de investigación existe una gran variedad de herramientas de simulación que pueden contribuir en menor o mayor escala al desarrollo específico en proceso. De ahí que se deba hacer un análisis profundo sobre las funcionalidades prestadas por estas herramientas para definir las que más se ajusten a los intereses y objetivos del proyecto, con el fin de cumplir a cabalidad las expectativas generadas por éste.

5.1.1 NECESIDAD DE LA HERRAMIENTA ADECUADA

Debido a que la simulación es una alternativa efectiva para llevar a cabo investigación de redes, varios simuladores Bluetooth han sido construidos, en general para la simulación de redes Bluetooth se han elaborado paquetes que se soportan sobre *frameworks* de simuladores de redes, generalmente sobre NS-2 (*Network Simulator-2*) u OPNET.

5.1.1.1 NS-2 vs. OPNET

Como dos de los simuladores de red más ampliamente usados, ambos NS-2 y OPNET son excelentes herramientas para simulación y modelado de redes. OPNET, por ser una herramienta comercial, ha incorporado una interfaz de usuario más amigable lo que ayuda a reducir el tiempo de la curva de aprendizaje. OPNET también tiene un conjunto de módulos que imitan los dispositivos del mundo real, además es una excelente herramienta para planeación y diagnóstico de tecnologías de Internet. Por otra parte, NS-2 es un proyecto de investigación principalmente basado en *software* libre. Como es conocido comúnmente, el software soportado por la comunidad de código abierto tiene grandes ventajas a la vez que presenta dificultades. En la tabla 5.1 se resume algunas diferencias de estas dos herramientas.

	NS-2	OPNET
LENGUAJE	C++,TCL (*Tool Command Language*)	C/C++
KIT DE DESARROLLO	Basado en texto	editor gráfico
EJECUTABLE	binario simple	modularizado
INTERFAZ DE USUARIO	script TCL	grafico, línea de comando
CURVA DE APRENDIZAJE	gradual	media
DISPONIBILIDAD	fuente abierta	licencia libre para uso académico
CODIGO FUENTE	disponible	*stack* de protocolos
PLATAFORMA	cualquier sistema basado en *unix*	*Windows*, *Solaris*, *Linux*

Tabla 5.1 NS-2 vs OPNET [35].

Las habilidades propias del desarrollador tienen un gran peso en la selección de uno u otro. NS-2 está basado en el lenguaje C++, por consiguiente la Programación Orientada a Objetos (OOP, *Object Oriented Programming*), ligada a C++ facilita la reutilización de código. OPNET está basado en el lenguaje C, sin embargo está empaquetado de tal forma que cierto diseño OOP también puede ser explotado. OPNET viene con un editor de diagrama de transición de estado, el cual es solamente útil si la MAC no es compleja. Vale la pena mencionar que el modelamiento del canal en OPNET es más preciso que el modelamiento en NS-2. Sin embargo, debido a que ambos son simuladores de red, y no de capa física, este hecho no tiene tanta relevancia como en un simulador hardware tal como *Matlab* que puede ser una mejor elección si el canal inalámbrico necesita ser modelado con precisión.

Debido a su mejor disponibilidad y soporte de la comunidad desarrolladora, se considera que NS-2 es una mejor elección como herramienta de simulación. A continuación se hace una descripción más profunda del simulador NS-2 para la definición del paquete de simulación Bluetooth que satisfaga las necesidades del proyecto [35].

5.1.1.2 Simulador NS-2.

El Simulador de Red 2 (*Network Simulator*, NS-2) es un simulador de eventos discretos de libre distribución diseñado y creado por la Universidad de *Berkeley*, que permite realizar simulaciones de diferentes tipos de redes como lo son las cableadas, inalámbricas y satelitales. La simulación tiene en cuenta tanto la topología de la red como el tráfico generado en ella con el fin de crear un diagnóstico que muestre el comportamiento de la red.

Este simulador implementa protocolos como TCP y UDP que pueden generar tráfico FTP, Telnet, Web, Velocidad Constante de Bit (CBR, *Constant Bit Rate*) y Velocidad Variable de Bit (VBR, *Variable Bit Rate*). Maneja diversos mecanismos de sistemas de colas que se generan en los *routers*, tales como *DropTail*, Detección Temprana Aleatoria (RED, *Random Earty Detection*), Colas Basadas en Clases (CQB, *Class Based Queuing*), algoritmo de enrutamiento como *Dijkstra*, *routing multicast* sobre redes cableadas e inalámbricas permitiendo a su vez simular QoS como por ejemplo *InmtServ* y *DiffServ*.

Para definir una simulación en NS-2 se utiliza un lenguaje de *script* llamado TCL que permite definir los distintos elementos de la red y su comportamiento, como también dispone de una interfaz gráfica para visualizar las simulaciones llamada Nam (*Network Animator*), la cual contiene un editor gráfico que permite crear animaciones sin la necesidad de usar código TCL. A través de Nam se puede visualizar los flujos de paquetes, colas y posibles descartes, así como comportamientos del protocolo: inicio lento de TCP, control de congestión, retransmisión rápida, recuperación, movimiento de nodos de redes inalámbricas, notas de los sucesos más importantes y estados del protocolo [36].

5.1.1.3 Paquetes de simulación Bluetooth

Al ser el primer simulador Bluetooth de fuente abierta disponible, el proyecto *Bluehoc* desarrollado por IBM ha sido usado bastante por la comunidad investigadora. Ofrece un modelo de banda base detallado, un modelo de propagación y desvanecimiento *indoor* derivado de las características radio del canal. Sin embargo, *BlueHoc* tiene un tamaño de código muy pequeño, lo cual implica que muchas funcionalidades están simplificadas y codificadas hacia el hardware. Por ejemplo, un nodo maestro y un nodo esclavo tienen diferentes estructuras internas, lo cual implica que un intercambio de roles es casi imposible sin una remodificación importante al simulador. Considerando su tamaño de código compacto, éste desempeña un buen trabajo si el escenario de simulación es estático. *BlueHoc* está basado en el popular *framework* NS-2, sin embargo, un nodo Bluetooth es bastante diferente de otros tipos de nodos en NS-2 y el código fuente de NS-2 es cambiado de tal manera que algunas funcionalidades de otros tipos de red están comprometidas. Este hecho hace incierta la incorporación de *BlueHoc* a la principal distribución de NS-2. El código está basado en ns-2.1b7a o ns-2.1b8a, versiones que están bastante desactualizadas ahora, y las cuales no pueden ser compiladas en una máquina Linux moderna sin errores significativos, tales como por ejemplo la degradación del compilador [37].

Blueware de MIT está basado en *BlueHoc*. En *Blueware*, muchas de las nuevas funcionalidades han sido añadidas y gran parte del código original ha sido reescrito. *Blueware* heredó muchas limitaciones de *BlueHoc*, por ejemplo usa una arquitectura de nodo similar a *BlueHoc* lo que representa diferencias considerables con los nodos normales de NS-2. También está basado en una versión desactualizada de NS-2 (ns-2.1b7a) y no ha sido actualizado desde su primera salida al entorno, generando las mismas dificultades de compilación en máquinas Linux modernas [38].

Existen otras extensiones para *BlueHoc* tales como el trabajo presentado en [39] el cual combina un nodo maestro y un esclavo en un nodo compuesto así el rol del nodo puede ser dinámicamente especificado y cambiado, también se ha agregado movilidad. Sin embargo, la extensión es bastante limitada por las limitaciones heredadas de *BlueHoc*. Un simulador del Perfil de Acceso a LAN se presentó en [40] y puede ser usado para abarcar la brecha existente entre la capa de enlace y la capa de red en una red IP del simulador *BlueHoc*; sin embargo, este perfil es prácticamente obsoleto y debería ser reemplazado por la especificación BNEP y el perfil PAN para ofrecer servicio IP, lo que por consiguiente también vuelve al simulador obsoleto.

Varios investigadores han usado el paquete *Simjava* y han agregado el stack Bluetooth a éste, pero los resultados divergen en gran medida de los resultados reales. Además existen simuladores enfocados específicamente hacia el modelado de la interferencia producida entre dispositivos Bluetooth y dispositivos IEEE 802.11. Estos simuladores son bastante útiles para estimar la interferencia debida al conocido uso de este espectro de frecuencia, sin embargo éste no es el enfoque principal del proyecto.

UCBT (*University of Cincinnati - BlueTooth*) ha sido de fuente abierta desde Junio de 2003, ha sido usado y probado por grupos de investigación en universidades y laboratorios de investigación industrial. Después de muchos años de desarrollo, UCBT se ha convertido en un robusto simulador con más de 34.000 líneas de código, lo cual es casi el doble de *Blueware* y 7 u 8 veces más grande que *BlueHoc*.

La principal razón para la selección del paquete de simulación UCBT es la posibilidad de simulación del perfil PAN con su correspondiente protocolo de soporte BNEP, lo que hace posible la validación de la arquitectura construida.

5.2 EXPLICACIÓN DEL PAQUETE DE SIMULACIÓN SELECCIONADO, UCBT – University of Cincinnati - BlueTooth

UCBT es un modulo de red Bluetooth basado en NS-2 el cual simula operaciones de red Bluetooth con gran detalle. La mayoría de las especificaciones en banda base y niveles superiores del *stack* de protocolos como LMP, L2CAP y capas BNEP son simuladas por UCBT (Figura 5.1), incluyendo esquemas de salto de frecuencia, descubrimiento de dispositivos, establecimiento de conexión, gestión de los modos *Hold*, *Sniff* y *Park*, conmutación, negociación de paquetes *multi-slot* y conexión de voz SCO, etc.

Brinda la opción de construir un *Cluster* con dispositivos Bluetooth donde el máximo permitido son 8 dispositivos, conocido en el estándar de Bluetooth como una *piconet*. También permite interconectar *piconets* usando nodos puente (*bridge nodes*), configuración denominada en el estándar de Bluetooth como *scatternet*.

UCBT adapta el perfil PAN con un protocolo de encapsulación para redes Bluetooth BNEP. UCBT toma una derivación de reloj, la cual es muy importante para la sincronización en la simulación o programación exacta de protocolos, a diferencia de los dispositivos que tomaran la derivación de un periodo [41].

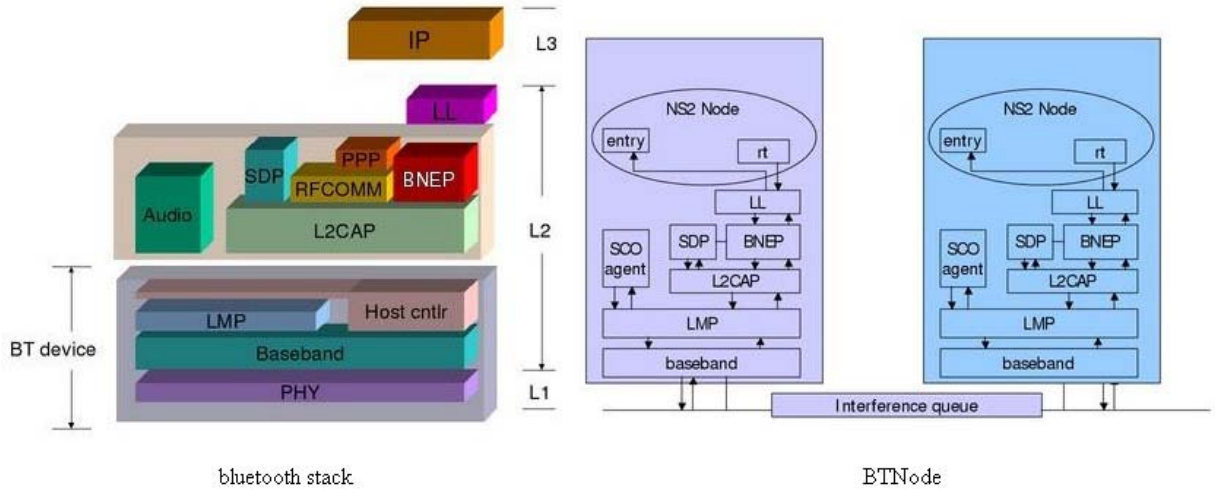


Figura 5.1 Paralelo entre el *Stack* de Protocolos Bluetooth y un Nodo BT [41].

UCBT también ha incluido la nueva Tasa de Bits Mejorada, especificación que es usada para simular dispositivos con ratas de *bit* de 2 y 3 Mbps. Una de las contribuciones principales es que UCBT proporciona un marco flexible en la investigación de *scatternets* referente a las configuraciones en Bluetooth. Una *scatternet* requiere compartir los tiempos de algunos dispositivos comunes (*bridges*) entre *piconets*, siendo todo un desafío la coordinación de un nodo puente en una *scatternet* en malla de gran cobertura. UCBT provee múltiples algoritmos para programar nodos puente habilitados en una *scatternet* haciendo que operen conjuntamente. Prototipos de *scatternets* auto-organizadas han sido diseñados y simulados [41].

5.2.1 CONSIDERACIONES ESPECÍFICAS DE UCBT

De acuerdo al enfoque del proyecto se deduce que las características del paquete de simulación que van de acuerdo a las necesidades del proyecto y su enfoque concreto son las siguientes:

- Modelado preciso de las capas bajas (Banda base y LMP).
- Soporte para el stack IP, perfil PAN y protocolo BNEP.
- Compatibilidad total con NS-2.
- Extensibilidad y flexibilidad.

Se cree que un modelado de los niveles más bajos es esencial para observar los comportamientos de los dispositivos correctamente. El paquete de simulación tiene potentes características con respecto al modelado de Banda Base y LMP de acuerdo a la especificación 2.0. La Banda Base básicamente maneja la transmisión y recepción de paquetes mientras que LMP maneja la organización, configuración y gestión del enlace. Bluetooth funciona de manera diferente a los dispositivos inalámbricos comunes IEEE 802.11 en donde éstos pueden transmitir entre ellos una vez se encuentren dentro de

cierto rango de proximidad. Un enlace tiene que ser explícitamente configurado antes de que cualquier información significativa pueda ser intercambiada. Además, un dispositivo tiene que frecuentemente sincronizarse con el canal. El modelado de estas características es crucial para la validación de los protocolos de capas más altas.

NS-2 es una de las herramientas de simulación más ampliamente usadas para conducir investigación de redes. Una compatibilidad plena con NS-2 y una interfaz consistente es crucial para que un simulador sea aceptado por la comunidad investigadora, y fácilmente actualizada hacia o integrada en futuras versiones de NS-2. En la implementación, todo lo concerniente a Bluetooth aparece como nuevas capas MAC/PHY (*Physic*) en NS-2. Cualquier cosa por encima de la capa LLC (*Logical Link Control*) permanece sin cambios.

5.2.1.1 Stack por capas

5.2.1.1.1 Capa BNEP

En el simulador NS-2 original solo se puede simular redes cableadas. Un equipo de investigación en CMU (*Central Michigan University*) ha extendido el soporte de NS-2 hacia las redes inalámbricas, y una nueva clase de nodo (llamado *MobileNode*) fue creado para representar nodos inalámbricos corriendo básicamente sobre protocolos 802.11. La familia IEEE 802 define una serie de estándares PHY/MAC los cuales están definidos bajo la misma subcapa LLC, la cual constituye la subcapa superior de la capa 2 (DLL, *Data Link Layer*) en el modelo de referencia OSI de 7 capas.

En la actual arquitectura de NS-2, cuando un paquete es bajado a la capa LLC (Objeto LL), este es colocado en una cola de interfaz (IFQ, *Interface Queue*), donde la MAC puede recoger un paquete para transmitirlo cuando éste asuma que el paquete puede ser transmitido de manera segura.

Como el estándar Bluetooth está construido alrededor del concepto del reemplazo de cables, éste no soporta el *networking* IP nativamente. Una manera vieja de construir IP sobre Bluetooth es crear una conexión PPP el cual trata el enlace Bluetooth como un “cable” inalámbrico. La gran sobrecarga de cabecera del protocolo y la inflexibilidad de las redes PPP han vuelto al protocolo casi obsoleto. Ahora, la forma recomendada y estándar es usar el Protocolo BNEP, el cual emula el *Ethernet* de tal manera que la capa LL común puede ser usada. El perfil PAN, construido y soportado sobre BNEP, ha definido una interfaz mini-puente la cual usualmente está situada entre las subcapas LL y MAC en la familia IEEE 802. Siempre que una conexión es configurada entre un par de dispositivos, un puerto es agregado a la interfaz puente y solo una conexión PAN/BNEP es posible entre cualquier par de dispositivos.

Cuando un paquete viaja a través de la capa BNEP/PAN, la dirección de próximo salto es usada para encontrar el puerto correcto o la conexión BNEP. Cada conexión BNEP es mapeada a un único canal L2CAP, y el paquete es bajado al canal L2CAP apropiado y encolado para una eventual transmisión. La interfaz de cola se implementa en la capa L2CAP. En lugar de usar una cola global para todas las conexiones, cada canal L2CAP tiene su propia IFQ así que puede ser implementada QoS fácilmente. Debe ser recalcado que ARP no es necesario debido a que un dispositivo siempre conoce la dirección MAC

de sus vecinos inmediatos. BNEP agrega una cabecera de 3 a 16 bytes a cada paquete de capa superior dependiendo de los diferentes formatos de direcciones usados.

En el lado del receptor, un paquete correctamente identificado por las capas más bajas es pasado a la capa BNEP desde el canal L2CAP. Después de que la cabecera es retirada, el paquete viaja hacia la capa superior la cual es especificada por el nodo de entrada en NS-2. Si este nodo es el destino, el paquete es pasado hacia la aplicación, de lo contrario es pasado al nodo de enrutamiento para conseguir la dirección del próximo salto, así el paquete podrá ser entregado posteriormente a su destino [42].

5.2.1.1.2 Capa L2CAP

L2CAP es utilizado para proporcionar Multiplexación, Segmentación y Re-ensamble (SAR). También ofrece *broadcast* para comunicaciones del grupo (dentro de la misma *piconet*). L2CAP también ofrece servicios de datos para aplicaciones diferentes a IP. Actualmente, trabajan 2 protocolos sobre la capa L2CAP (BNEP y SDP) la cual mantiene los canales entre un par de dispositivos. Diversos protocolos de capas superiores son mapeados a diferentes canales. La capa L2CAP incluye un encabezado de 4 bytes a cada paquete proveniente de capas superiores. En L2CAP, los requerimientos de QoS para diferentes canales pueden ser especificados y la MTU puede ser negociada entre un par de dispositivos. Puesto que como máximo un enlace ACL puede ser establecido entre un par de dispositivos, múltiples canales L2CAP deben ser mapeados a un solo enlace ACL y compartir el enlace físico. En este caso, se utiliza el enlace alternadamente y lo libera una vez se ha transmitido el paquete L2CAP completamente, procediendo la rutina de re-ensamble en recepción a re-ensamblar el paquete correctamente. Actualmente UCBT, re-ensambla los paquetes en la capa L2CAP. Sin embargo la fragmentación es dirigida a la capa LMP, como consecuencia, la decisión de fragmentación puede ser postergada al momento de transmitir. De esta manera, puede ser empleado un algoritmo de fragmentación el cual considera las condiciones dinámicas de enlace [42].

5.2.1.1.3 Capa LMP

LMP es una interfaz de gestión de enlace que maneja el establecimiento del enlace y sus derivaciones, negocia la QoS, gestiona el cambio de roles y modos de bajo consumo de potencia (*Sniff*, *Hold* y *Park*). Es muy importante porque una *scatternet* tiene que utilizar uno de estos modos de bajo consumo de potencia. Muchas operaciones LMP requieren un periodo de tiempo (segundos o más) bastante largo y la rata de éxito no es garantizada (por ejemplo, la paginación es determinada fallida si el otro extremo está fuera del rango) [42].

5.3 ESCENARIOS DE SIMULACIÓN

En esta sección se definen cada uno de los escenarios requeridos para el proceso de validación de la arquitectura propuesta durante el proyecto.

En cada uno de los escenarios se debe evaluar el desempeño del sistema y la viabilidad de la comunicación tanto como con flujos constantes de datos, como con flujos propios de sistemas con detección de actividad del protocolo RTP generados dependiendo del tipo de códec, pertenecientes al *stack* TCP/IP dentro del *stack* de simulación del simulador NS-2.

Factores a tener en cuenta durante la simulación:

- Definición de los tiempos de duración de las simulaciones.
- Evaluar el tipo de paquetes vs. desempeño del sistema
    - Retardo, pérdida de paquetes y ancho de banda consumido.

Parámetros a definir en cada simulación efectuada:

- Tipo de códec utilizado.
- No. De paquetes generados.
- No. De paquetes perdidos.
- Tipo de paquete utilizado.

Al hablar de un Sistema, se refiere al conjunto de elementos cuyo propósito es cumplir un fin lógico, mientras que el Estado del Sistema apunta al conjunto de variables necesarias para describir su comportamiento en determinado instante de tiempo. Dichas variables se clasifican en entradas, que serán los valores numéricos que permitan iniciar la simulación; y salidas, obtenidas de las entradas, y que serán objeto de estudio.

Entre las entradas se encuentran:

- Condiciones iníciales: Valores que expresan el estado del sistema al principio de la simulación.
- Datos determinísticos: Valores conocidos necesarios para realizar los cálculos que producen las salidas.
- Datos probabilísticos: Cantidades cuyos valores son inciertos pero necesarios para obtener las salidas del sistema. Los valores específicos de estos datos deben conocerse a través de una distribución de probabilidad.

5.4 Primer Escenario

En primera instancia, se plantea una comunicación donde el tipo de transmisión sea *simplex*, como su definición lo indica el flujo de información viajará en un solo sentido y se considera un flujo constante de datos (CBR), en el caso de Bluetooth bien puede ser del Maestro hacia el (los) Esclavo(s), ó del (los) Esclavo(s) hacia el Maestro. Además, teniendo en cuenta que una *piconet* puede soportar hasta 7 esclavos activos al mismo tiempo, obviamos el caso donde la transmisión la realiza un solo esclavo ya que sería un caso de 100% de eficiencia en paquetes recibidos, un retardo mínimo y ancho de banda dedicado a una sola comunicación.

Teniendo todo esto como precedente, se plantea las simulaciones donde la información es enviada desde 2 hasta 7 esclavos incrementando de uno en uno hacia el mismo maestro, y el caso opuesto donde el maestro transmite hacia 2 esclavos, incrementando estos últimos de uno en uno hasta 7, de tal forma que se sature la piconet y así poder analizar el caso más crítico desde el punto de vista de congestión en el canal de transmisión.

TRAFICO CBR - CODEC G.711						
		RETARDO (seg)	**ENVIADOS**	**PAQUETES PERDIDOS**	**% ENTREGA**	**BW (*Bandwidth*) CONSUMIDO (Kbps)**
PUNTO A MULTIPUNTO	1 a 2	0,1277835	85	0	100,00%	59,84
	1 a 3	0,1292063	83	0	100,00%	87,648
	1 a 4	0,12735725	82	0	100,00%	115,456
	1 a 5	0,1282146	84	0	100,00%	147,84
	1 a 6	0,35115533	98	0	100,00%	206,976
	1 a 7	0,64300142	103	0	100,00%	253,792
MULTIPUNTO A PUNTO	2 a 1	0,011426	163	0	100,00%	57,376
	3 a 1	0,011544	234	0	100,00%	82,368
	4 a 1	0,307585	458	0	100,00%	161,216
	5 a 1	0,707233	647	27	95,83%	227,744
	6 a 1	1,006452	832	69	91,71%	292,864
	7 a 1	1,003573	1273	135	89,40%	448,096

Tabla 5.2 Resultados de Simulación primer escenario.

Tomando como referencia las características de cada uno de los CODEC*s* ya mencionados en capítulos anteriores, estos serán tenidos en cuenta al evaluar este escenario, puesto que de acuerdo a los datos, el códec que requiere mayor tamaño de paquete y una alta tasa de transmisión es el G.711, con 200bytes y 64Kbps respectivamente. En cuanto a los tipos de paquetes manejados a nivel Bluetooth, encontramos el paquete DH5 el cual cuenta con la carga útil más grande que va desde 0 hasta 339 bytes. Finalmente los datos obtenidos al simular este escenario, se resumen en la Tabla 5.2.

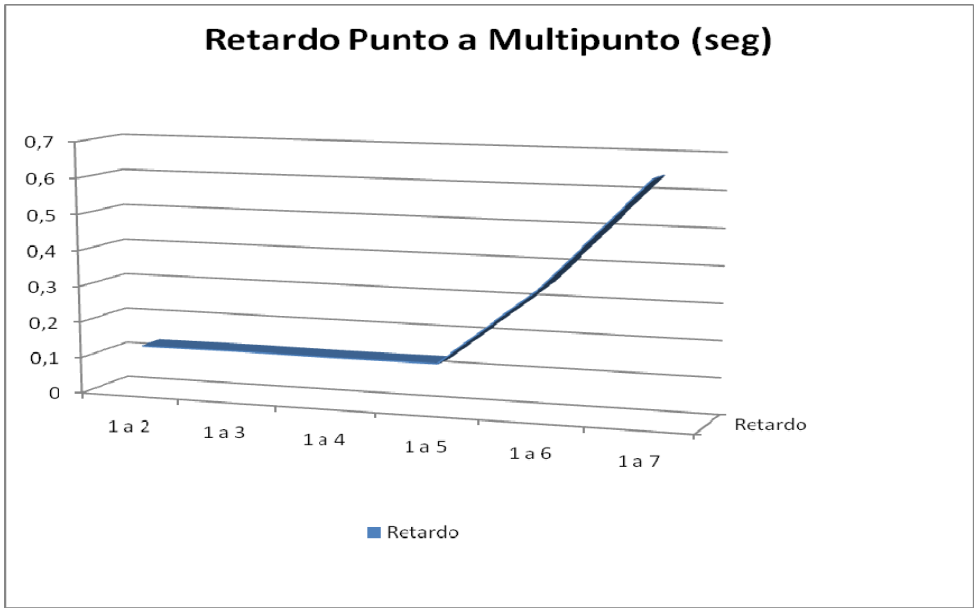


Figura 5.2 Retardo promedio conexiones punto a multipunto.

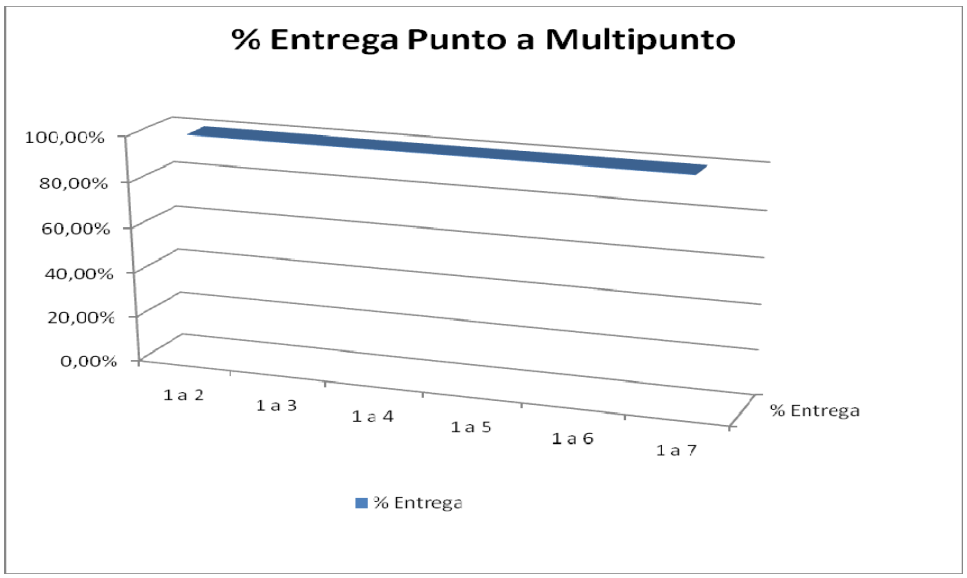


Figura 5.3 Porcentaje de entrega promedio conexiones punto a multipunto.

A partil de la Tabla 5.2 se generan las Figuras 5.2 hasta la Figura 5.7 las cuales muestran el retardo, porcentaje de entrega y ancho de banda consumido en relación al número de conexiones y flujos de datos generados para las respectivas conexiones Punto a Multipunto y Multipunto a Punto.

En la Figura 5.2 se aprecia que el desempeño del sistema con este tipo de conexiones para tráfico de voz cumple con las condiciones de Calidad de Servicio hasta cinco

conexiones. A partir de seis conexiones simultáneas el desempeño sufre una degradación considerable (el retardo alcanza un máximo de 0,643 seg).

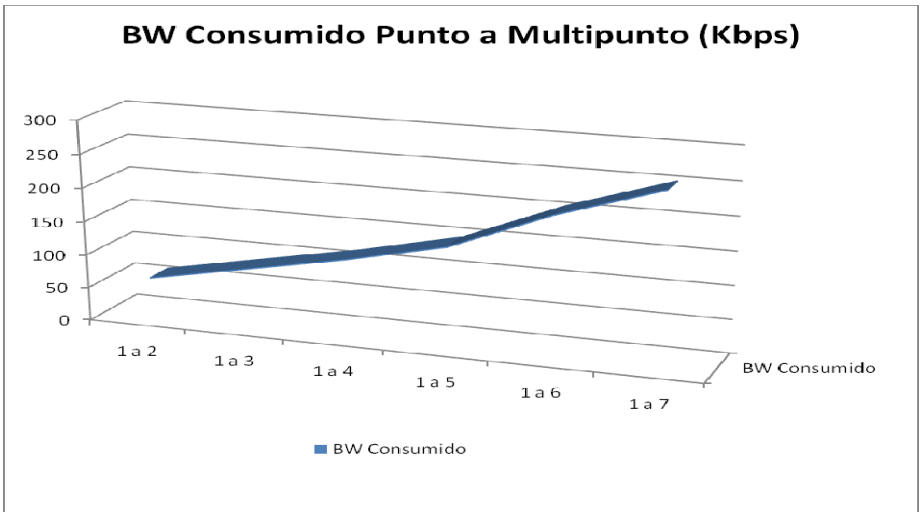


Figura 5.4 Ancho de banda consumido conexiones punto a multipunto.

La pérdida de paquetes reflejada por el porcentaje de entrega (Figura 5.3), refleja que es nula; además el ancho de banda consumido por los flujos de datos (Figura 5.4 y Figura 5.7) se ve en incremento conforme aumentan el número de conexiones soportadas por el punto de acceso de la *piconet*.

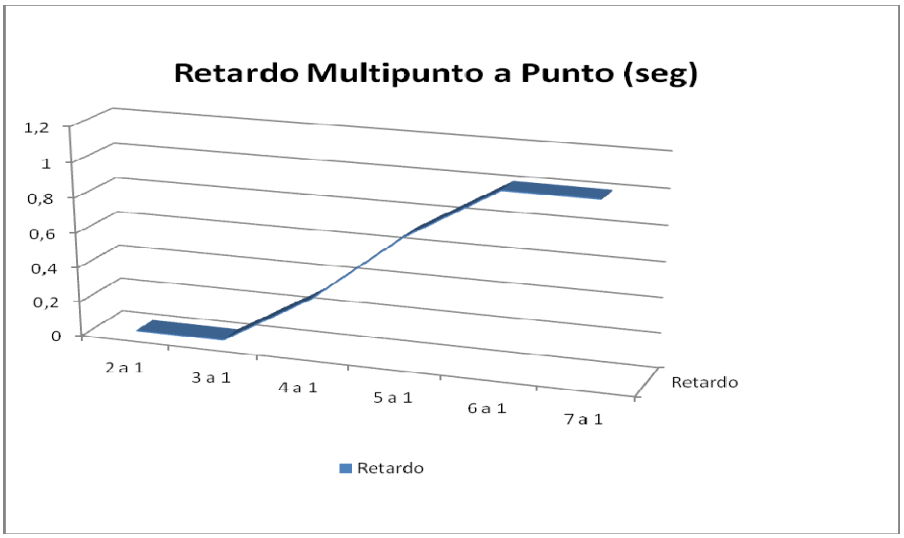


Figura 5.5 Retardo promedio conexiones multipunto a punto.

En la Figura 5.5 se aprecia que el retardo de las conexiones permanece por debajo de los 150mseg sólo hasta tres conexiones con el punto de acceso, es decir, a nivel de recepción se sufre una degradación de desempeño más considerable comparado con transmisión.

Este fenómeno ocurre, debido principalmente al sistema de manejo de colas en recepción. Esto se ve reflejado directamente en el número de paquetes perdidos promedio medidos durante la simulación (Figura 5.6); por consiguiente, es recomendable configurar una cola de recepción por cada conexión establecida que permita optimizar esta situación.

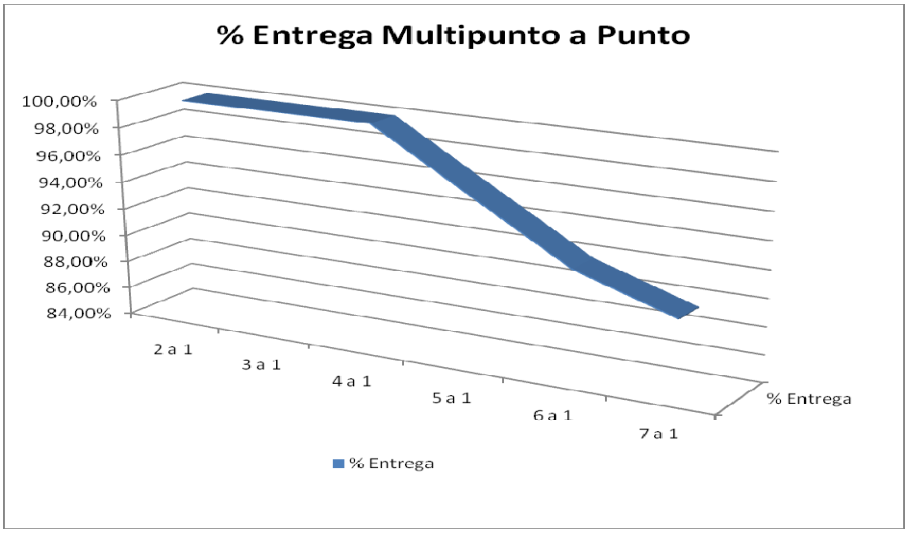


Figura 5.6 Porcentaje de entrega promedio conexiones multipunto a punto.

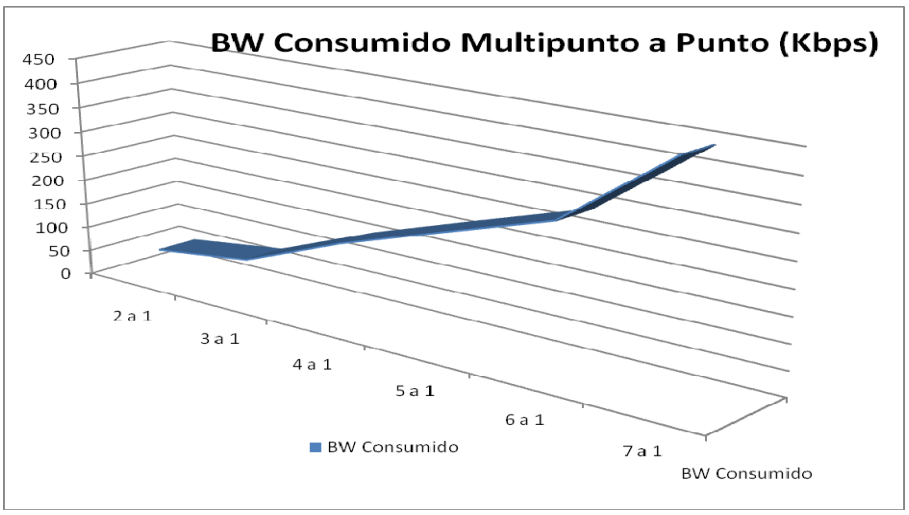


Figura 5.7 Ancho de banda consumido conexiones multipunto a punto.

5.5 Segundo Escenario

Luego de evaluar la conexión *simplex* con tráfico constante de datos (CBR), consideramos la conexión *full-duplex* como el tipo de conexión que representa una transmisión de voz, siendo el punto de acceso (representado por el nodo maestro) el encargado de enlazar dichas conexiones que representan conversaciones.

Teniendo en cuenta las características de cada uno de los CODECs, puesto que su tamaño de paquete y tasa de transferencia de datos varían para cada uno de ellos, evaluamos los 3 más representativos con los diferentes tipos de paquetes planteados en la especificación Bluetooth 1.1 y 1.2. Se evaluó el desempeño para los tipos de paquetes que no implementan FEC, ya que estos ofrecen una mayor capacidad de carga útil, por consiguiente mayor ancho de banda por enlace. Sumado a esto, a partir de la especificación Bluetooth 2.0 + EDR implementan 6 nuevos tipos de paquetes que operan con ratas de datos mejoradas (2-DH1, 3-DH1, 2-DH3, 3-DH3, 2-DH5 y 3-DH5) que a diferencia de las especificaciones anteriores, muchos de los dispositivos aún no los soportan, considerando así como paquetes apropiados los DH1, DH3 y DH5.

Los retardos medidos por el simulador representan cada uno de los retardos promedios por salto. Debido a que la comunicación se realiza entre dos terminales móviles a través de un punto de acceso representado por el nodo maestro, para obtener el retardo extremo a extremo de los paquetes durante la comunicación se deben sumar los retardos de cada uno de los dos saltos. En cuanto al ancho de banda, éste es calculado en base al número de paquetes enviados durante la comunicación, la tasa de transferencia y el tamaño de paquete propios de cada CODEC, además del numero de flujos de comunicaciones y el tiempo de simulación.

Por ejemplo, el ancho de banda consumido por el flujo de datos de tres comunicaciones simultaneas, CODEC G.711 y tipo de paquete DH1 se calcula de la siguiente manera:

$$BW\_consumido = \frac{N^o\_paq\_env * tamaño\_paq(bytes) * num\_flujos * 8bits}{tiempo\_de\_simulación\_(mseg)}(Kbps)$$

$$BW\_consumido = \frac{151 * 220 * 3 * 8}{5000} = 159.456(Kbps)$$

El tamaño de paquete en este ejemplo, es de 220 bytes conformados por 160 bytes de la carga útil del protocolo RTP codificado con G.711, 12 bytes de cabecera RTP, 8 bytes de cabecera UDP, 20 bytes de cabecera IP, 16 bytes de cabecera BNEP y 4 bytes de cabecera L2CAP. Una descripción más detallada del encapsulamiento de VoIP puede ser encontrada en la sección 2.3.7 del presente documento.

Por tanto el tamaño de los paquetes transmitidos a nivel L2CAP para cada uno de los CODECs es:

- G.711 (tamaño de paquete transmitido de 220 bytes).
- G.723.1 (tamaño de paquete transmitido de 84 bytes).
- G.729 (tamaño de paquete transmitido de 80 bytes).

En las tablas 5.3, 5.4 y 5.5 se resume la información correspondiente al escenario ya planteado.

FULL DUPLEX							
			RETARDO (seg)	**ENVIADOS**	**PAQUETES PERDIDOS**	**% ENTREGA**	**BW CONSUMIDO (Kbps)**
G.711	**DH1**	1 A 1	0,136926	127	0	100,00%	44,7040
		2 A 2	1,898128	167	35	79,04%	117,5680
		3 A 3	2,099620	151	44	70,86%	159,4560
	DH3	1 A 1	0,020024	123	0	100,00%	43,2960
		2 A 2	0,886681	142	0	100,00%	99,9680
		3 A 3	1,462808	161	43	73,29%	170,0160
	DH5	1 A 1	0,014737	122	0	100,00%	42,9440
		2 A 2	0,246639	130	0	100,00%	91,5200
		3 A 3	1,254720	161	35	78,26%	170,0160

Tabla 5.3 Resultados de Simulación *Full-duplex*, CBR y G.711

FULL DUPLEX							
			RETARDO (seg)	**ENVIADOS**	**PAQUETES PERDIDOS**	**% ENTREGA**	**BW CONSUMIDO (Kbps)**
G.723	**DH1**	1 A 1	0,045683	39	0	100,00%	5,2416
		2 A 2	0,045023	39	0	100,00%	10,4832
		3 A 3	0,352200	64	0	100,00%	25,8048
	DH3	1 A 1	0,040328	38	0	100,00%	5,1072
		2 A 2	0,041652	39	0	100,00%	10,4832
		3 A 3	0,156478	64	0	100,00%	25,8048
	DH5	1 A 1	0,040328	38	0	100,00%	5,1072
		2 A 2	0,041652	39	0	100,00%	10,4832
		3 A 3	0,156478	64	0	100,00%	25,8048

Tabla 5.4 Resultados de Simulación *Full-duplex*, CBR y G.723

FULL DUPLEX							
			RETARDO (seg)	ENVIADOS	PAQUETES PERDIDOS	% ENTREGA	BW CONSUMIDO (Kbps)
G.729	DH1	1 A 1	0,035698	51	0	100,00%	6,5280
		2 A 2	0,073628	53	0	100,00%	13,5680
		3 A 3	0,541566	84	0	100,00%	32,2560
	DH3	1 A 1	0,036766	51	0	100,00%	6,5280
		2 A 2	0,035545	52	0	100,00%	13,3120
		3 A 3	0,229973	85	0	100,00%	32,6400
	DH5	1 A 1	0,036766	51	0	100,00%	6,5280
		2 A 2	0,035545	52	0	100,00%	13,3120
		3 A 3	0,229973	85	0	100,00%	32,6400

Tabla 5.5 Resultados de Simulación *Full-duplex*, CBR y G.729

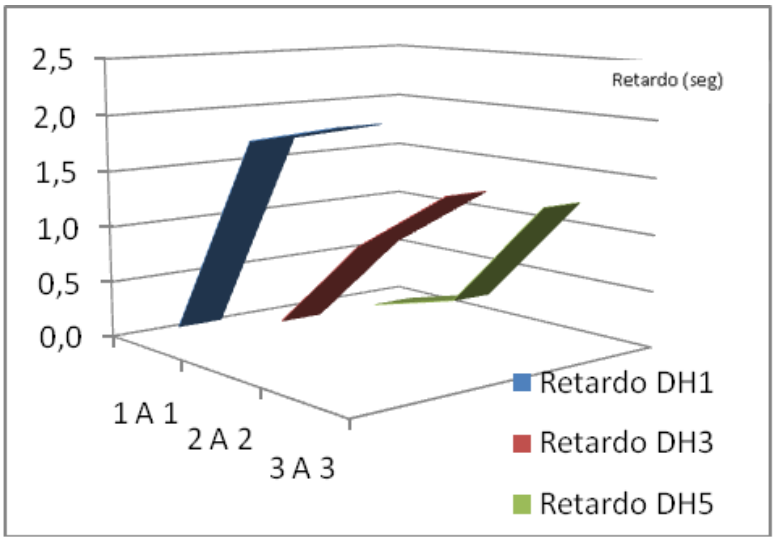


Figura 5.8 Retardo promedio para flujo de datos Full-Duplex CBR con CODEC G.711

Desde la Tabla 3.1 ubicada en el tercer capítulo del presente documento, tomamos los valores de carga útil máximos para los tipos de paquete analizados y seleccionados. Posteriormente, en base al tamaño de paquete transmitido por cada CODEC se puede hacer un análisis comparativo evaluando dicho tamaño con la carga útil de cada tipo de paquete a nivel de conexiones ACL.

- DH1 (carga útil máxima de 27 bytes)
- DH3 (carga útil máxima de 183 bytes)

- DH5 (carga útil máxima de 229 bytes)

En la Figura 5.8 se puede observar que el tipo de paquete DH5 presenta el menor retardo en la transmisión en comparación con los paquetes DH3 y DH1, cuando se utiliza el CODEC G.711. Esto se debe a que a nivel L2CAP se presenta mayor fragmentación para manejar la carga de paquetes generada por éste CODEC.

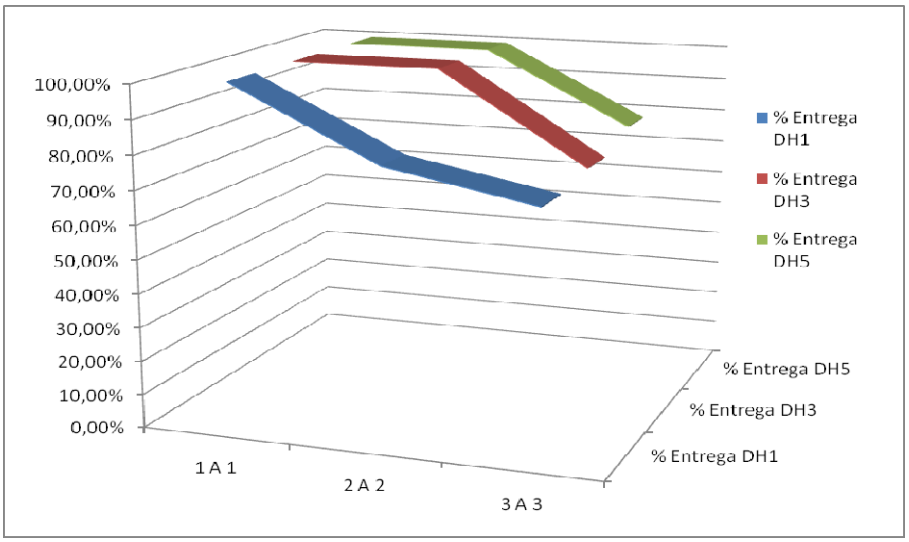


Figura 5.9 Porcentaje de entrega promedio para flujo de datos Full-Duplex CBR con CODEC G.711

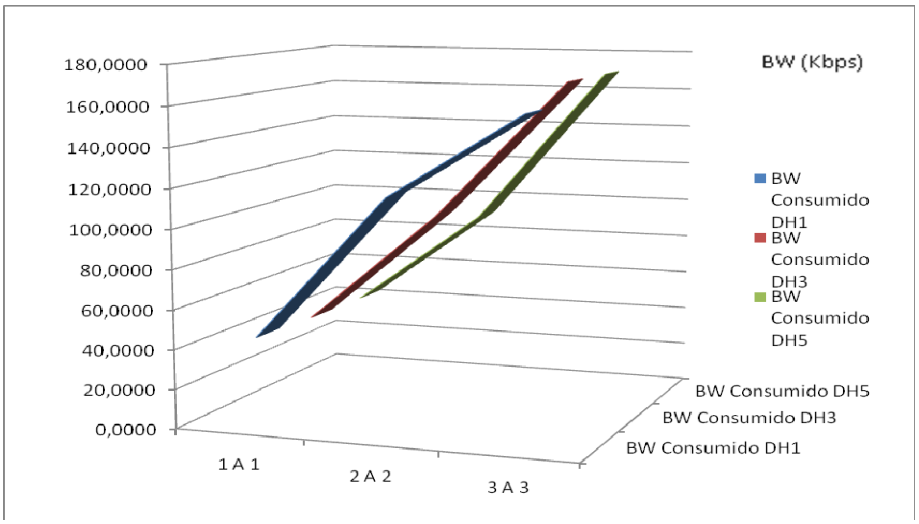


Figura 5.10 Ancho de banda consumido para flujo de datos Full-Duplex CBR con CODEC G.711

El tipo de paquete seleccionado también tiene una influencia directa sobre el porcentaje de paquetes perdidos (Figura 5.9), dado que hasta las conexiones 2 a 2 con DH5 la pérdida es de 0 porciento, y en conexiones 3 a 3 presenta el menor porcentaje de perdida en consideración con los CODECs G.723 y G.729. Respecto al ancho de banda (Figura 5.10), como ha de esperarse, incrementa conforme aumenta el número de conexiones soportadas por el punto de acceso.

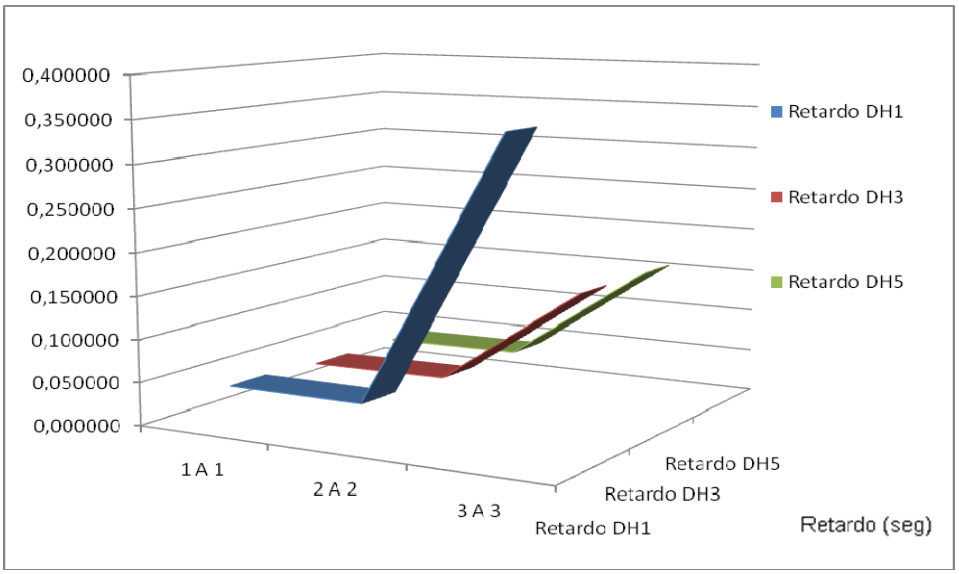


Figura 5.11 Retardo promedio para flujo de datos Full-Duplex CBR con CODEC G.723

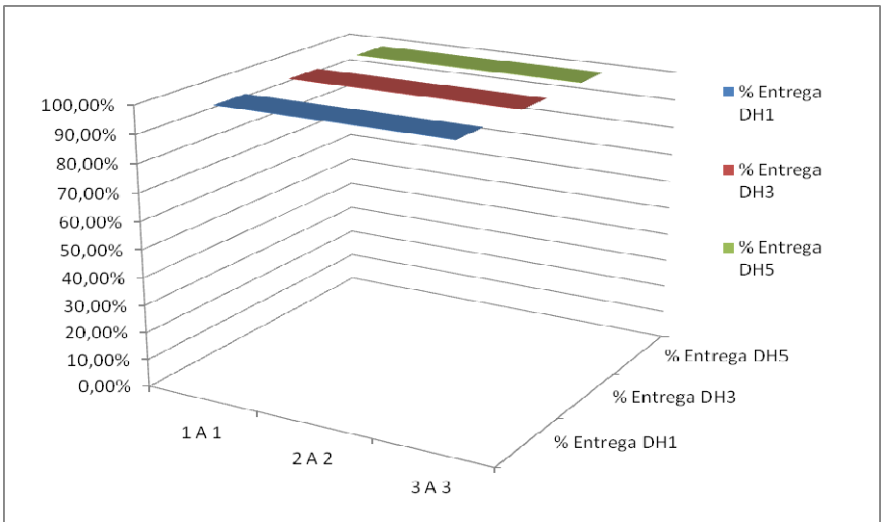


Figura 5.12 Porcentaje de entrega promedio para flujo de datos Full-Duplex CBR con CODEC G.723

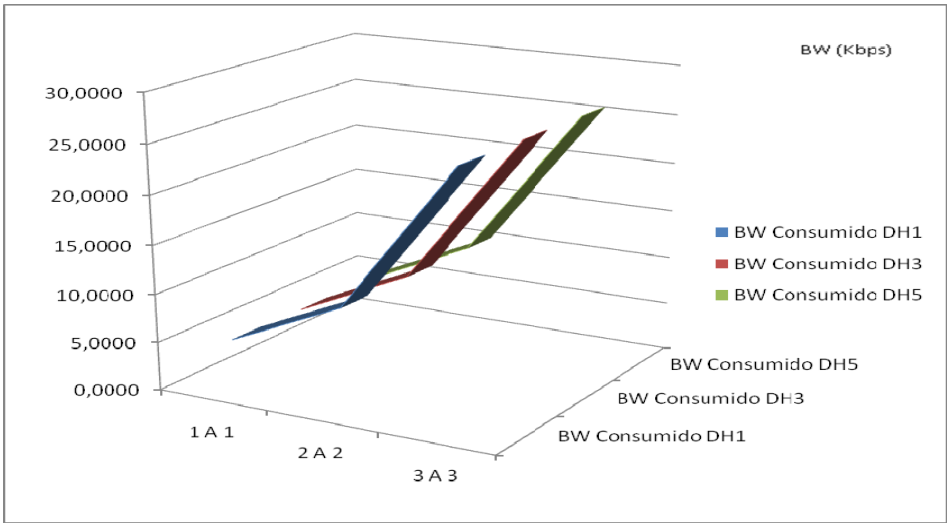


Figura 5.13 Ancho de banda consumido para flujo de datos Full-Duplex CBR con CODEC G.723

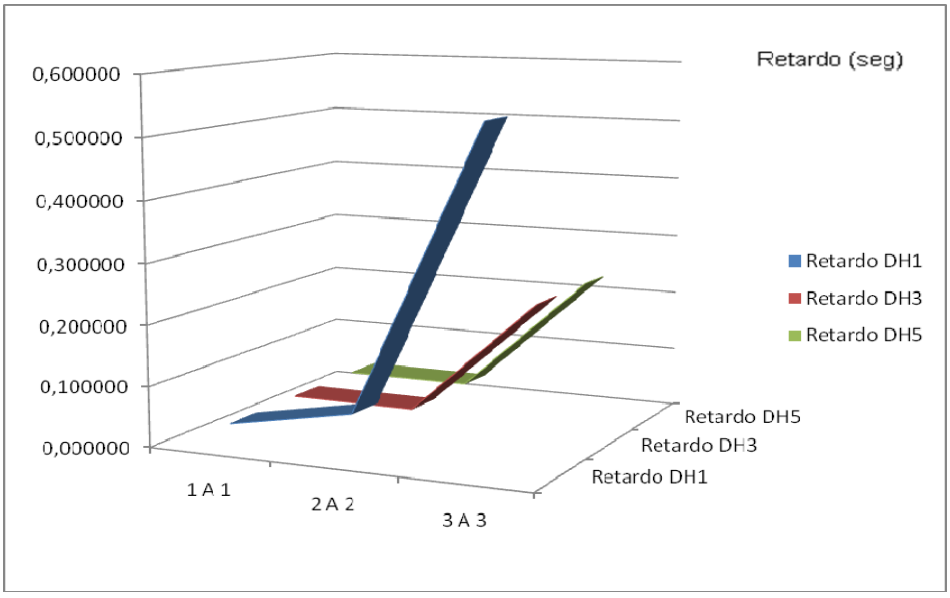


Figura 5.14 Retardo promedio para flujo de datos Full-Duplex CBR con CODEC G.729

En las Figuras 5.11 y 5.14 se puede observar que el tipo de paquete DH1 genera niveles de retardo más altos respecto a los paquetes DH3 y DH5, esto debido a que los CODECs G.723 y G.729 tienen un tamaño de paquete de 84 y 80 bytes respectivamente, posteriormente es pasada dicha información al nivel L2CAP la cual requiere mayor fragmentación si trabaja con paquetes DH1, de lo contrario su comportamiento es similar o casi igual al trabajar con paquetes DH3 y DH5 que con su carga útil máxima encapsulan perfectamente la información generada por los CODECs ya mencionados.

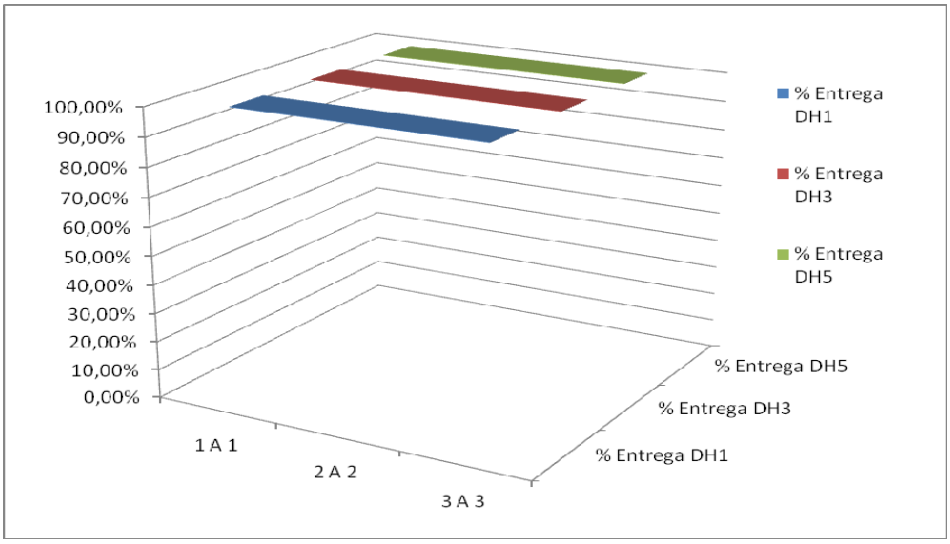


Figura 5.15 Porcentaje de entrega promedio para flujo de datos Full-Duplex CBR con CODEC G.729

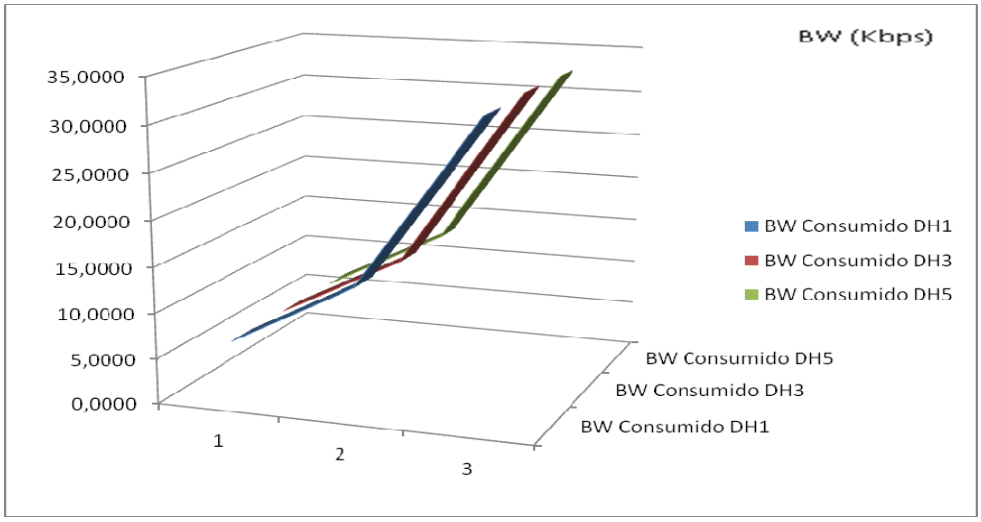


Figura 5.16 Ancho de banda consumido para flujo de datos Full-Duplex CBR con CODEC G.729

Respecto a los porcentajes de entrega manejados con los CODECs G.723 y G.729, tipo de conexión Full-duplex y tráfico CBR, tienen una efectividad del 100% (Figuras 5.12 y 5.15), además del incremento gradual en los niveles consumidos por el ancho de banda conforme aumentan la cantidad de conexiones (Figuras 5.13 y 5.16).

5.6 Tercer Escenario

Una vez evaluado el tráfico constante de datos (CBR) considerado en conexiones *full-duplex*, la mayor aproximación a una transmisión de voz con sistema de detección VAD ya que una conversación normal tiene momentos de silencio en el que cesa dicho flujo, es este escenario el punto de acceso (maestro de la *piconet)* sigue siendo el encargado de enlazar dichas conexiones que representan un dialogo.

FULL DUPLEX							
			RETARDO (seg)	**ENVIADOS**	**PAQUETES PERDIDOS**	**% ENTREGA**	**BW CONSUMIDO (Kbps)**
G.711	**DH1**	1 A 1	0,21478400	19	0	100,00%	6,68800
		2 A 2	0,54165300	51	0	100,00%	35,90400
		3 A 3	0,92311360	36	5	86,11%	38,01600
	DH3	1 A 1	0,18685600	19	0	100,00%	6,68800
		2 A 2	0,15959920	53	0	100,00%	37,31200
		3 A 3	0,40904100	55	1	98,18%	58,08000
	DH5	1 A 1	0,17449200	19	0	100,00%	6,68800
		2 A 2	0,14959040	36	0	100,00%	25,34400
		3 A 3	0,16495660	40	0	100,00%	42,24000

Tabla 5.6 Resultados de Simulación *Full-duplex*, VAD y G.711

FULL DUPLEX							
			RETARDO (seg)	**ENVIADOS**	**PAQUETES PERDIDOS**	**% ENTREGA**	**BW CONSUMIDO (Kbps)**
G.723	**DH1**	1 A 1	0,14930000	6	0	100,00%	0,80640
		2 A 2	0,12325400	10	0	100,00%	2,68800
		3 A 3	0,14481400	14	0	100,00%	5,64480
	DH3	1 A 1	0,14062000	6	0	100,00%	0,80640
		2 A 2	0,12982600	11	0	100,00%	2,95680
		3 A 3	0,13209800	14	0	100,00%	5,64480
	DH5	1 A 1	0,14062000	6	0	100,00%	0,80640
		2 A 2	0,12982600	11	0	100,00%	2,95680
		3 A 3	0,13209800	14	0	100,00%	5,64480

Tabla 5.7 Resultados de Simulación *Full-duplex*, VAD y G.723

Tomando de nuevo las características de cada uno de los CODECs, como son su tamaño de paquete y tasa de muestreo, evaluamos cada uno de ellos con los diferentes tipos de paquetes planteados en la especificación Bluetooth y escogidos previamente (DH1, DH3 y DH5). Los retardos medidos en simulación representan cada uno de los retardos promedios por salto, debido a que la comunicación se realiza entre dos nodos móviles a través de un punto de acceso representado por el nodo maestro, es decir se deben sumar los retardos por cada salto para obtener el retardo extremo a extremo de los paquetes.

FULL DUPLEX							
			RETARDO (seg)	**ENVIADOS**	**PAQUETES PERDIDOS**	**% ENTREGA**	**BW CONSUMIDO (Kbps)**
G.729	**DH1**	1 A 1	0,13836400	8	0	100,00%	1,02400
		2 A 2	0,10964400	20	0	100,00%	5,12000
		3 A 3	0,14252400	18	0	100,00%	6,91200
	DH3	1 A 1	0,15802600	8	0	100,00%	1,02400
		2 A 2	0,10336000	16	0	100,00%	4,09600
		3 A 3	0,12741200	21	0	100,00%	8,06400
	DH5	1 A 1	0,15802600	8	0	100,00%	1,02400
		2 A 2	0,10336000	16	0	100,00%	4,09600
		3 A 3	0,12741200	21	0	100,00%	8,06400

Tabla 5.8 Resultados de Simulación *Full-duplex*, VAD y G.729

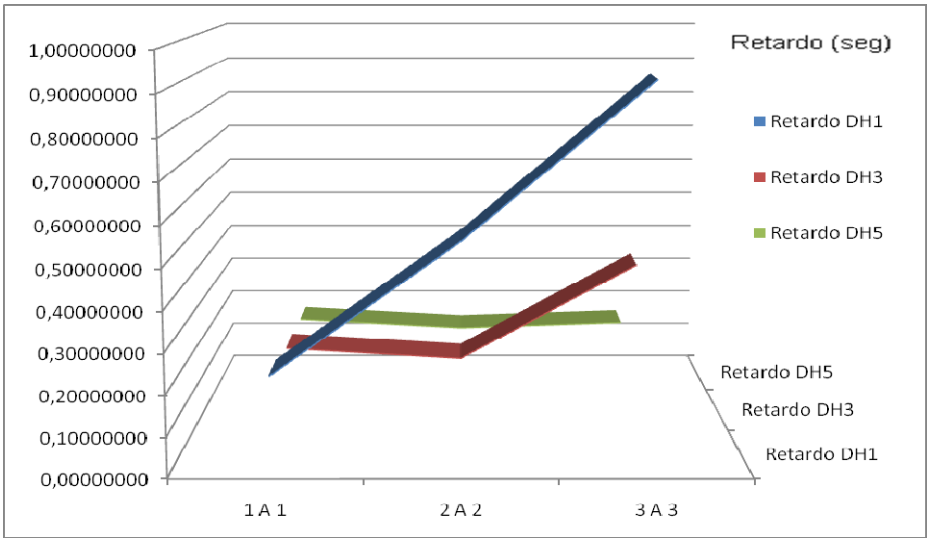


Figura 5.17 Retardo promedio para flujo de datos Full-duplex, VAD y CODEC G.711

Los resultados obtenidos con base a este escenario presentan variaciones en las tendencias de los retardos presentados en el escenario dos. Es decir, mientras que en el escenario dos los retardos aumentaban proporcionalmente con el número de conexiones llevadas a cabo, en el escenario tres (Figura 5.20 y 5.23) el retardo disminuía en las conexiones 2 a 2. Esto se debe a que el flujo de datos generado con la utilización de los CODECs G.723 y G.729 con sistema VAD se ajustan mejor cuando se generan 2 comunicaciones de voz simultáneas, mientras el CODEC G.711 genera un flujo de datos lo suficientemente grande para ajustarse a los tipos de paquetes DH3 y DH5 (Figura 5.17).

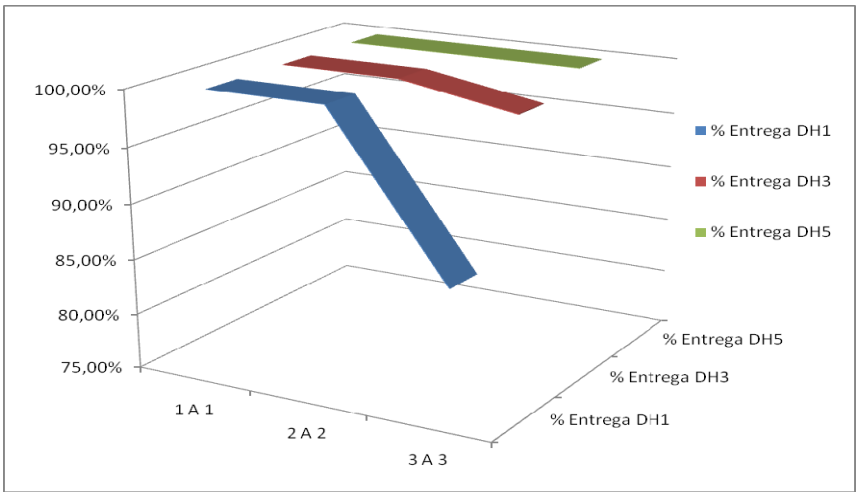


Figura 5.18 Porcentaje de entrega promedio para flujo de datos Full-duplex, VAD y CODEC G.711

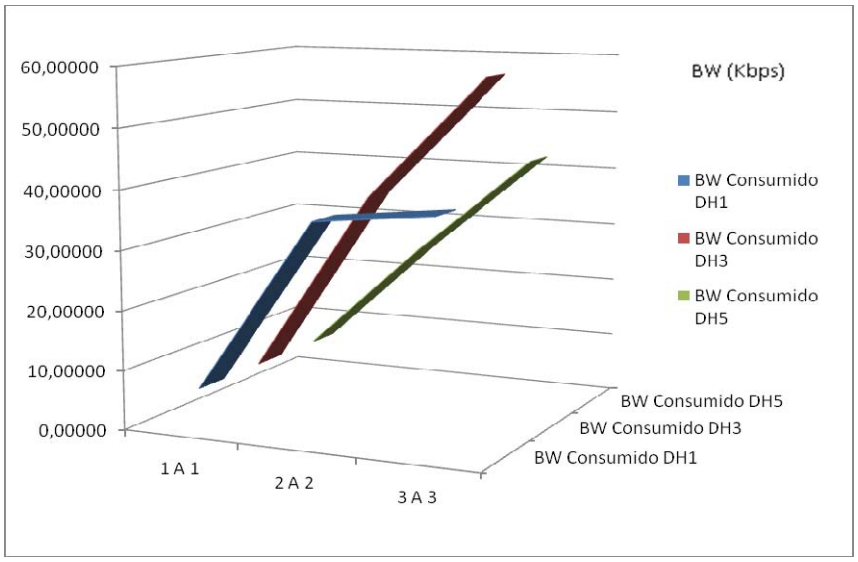


Figura 5.19 Ancho de banda consumido para flujo de datos Full-duplex, VAD y CODEC G.711

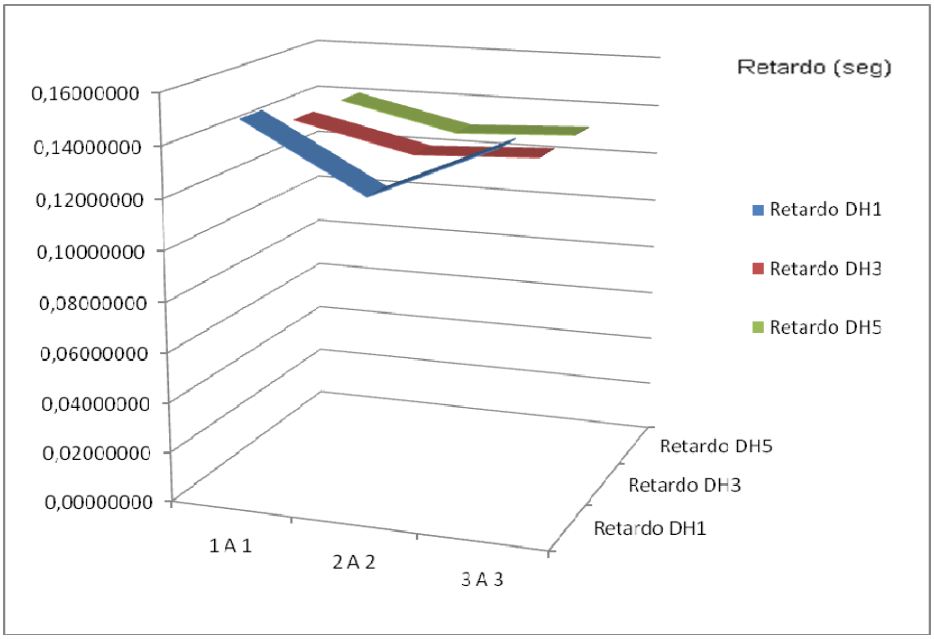


Figura 5.20 Retardo promedio para flujo de datos Full-duplex, VAD y CODEC G.723

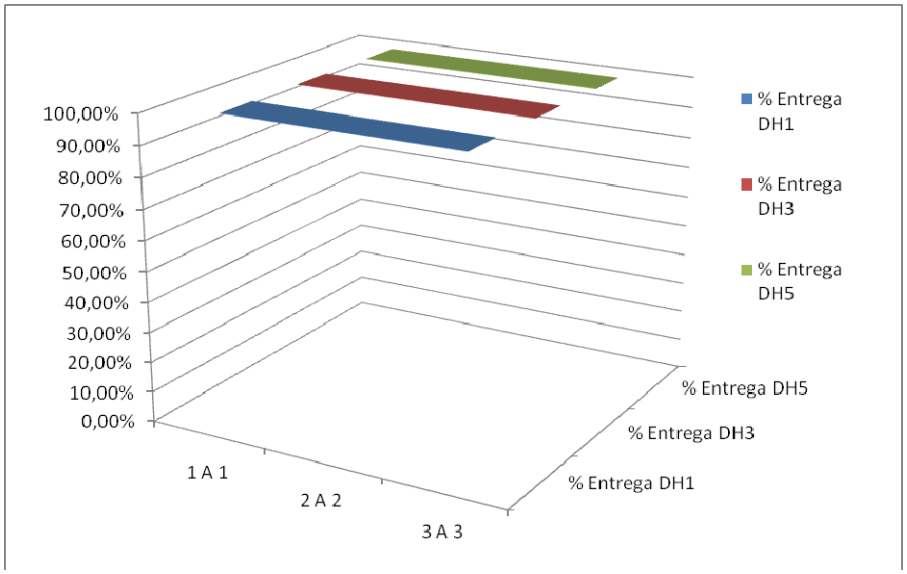


Figura 5.21 Porcentaje de entrega promedio para flujo de datos Full-duplex, VAD y CODEC G.723

En cuanto al porcentaje de entrega de los paquetes, los flujos generados con los CODECs G.723 y G.729 (Figuras 5.21 y 5.24) revelan una pérdida de paquetes de cero por ciento, siendo su tamaño la razón por la cual no requieren fragmentación a nivel de conexiones ACL, que al ser comparados con el gran tamaño de paquetes generados con

el CODEC G.711 se ve claramente como incrementa el porcentaje de pérdida de paquetes en este último (Figura 5.18), sumado a esto influye el número de conexiones de comunicación simultáneas.

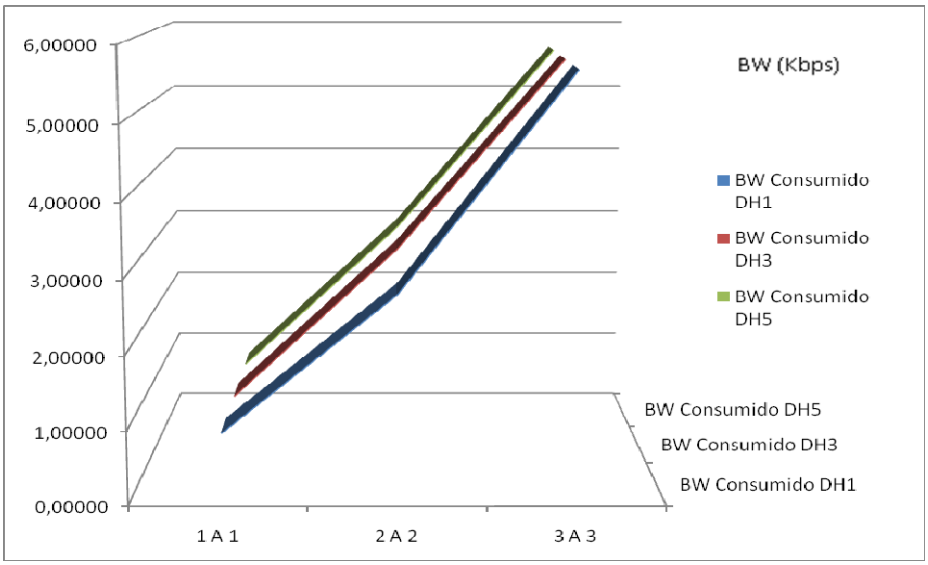


Figura 5.22 Ancho de banda consumido para flujo de datos Full-duplex, VAD y CODEC G.723

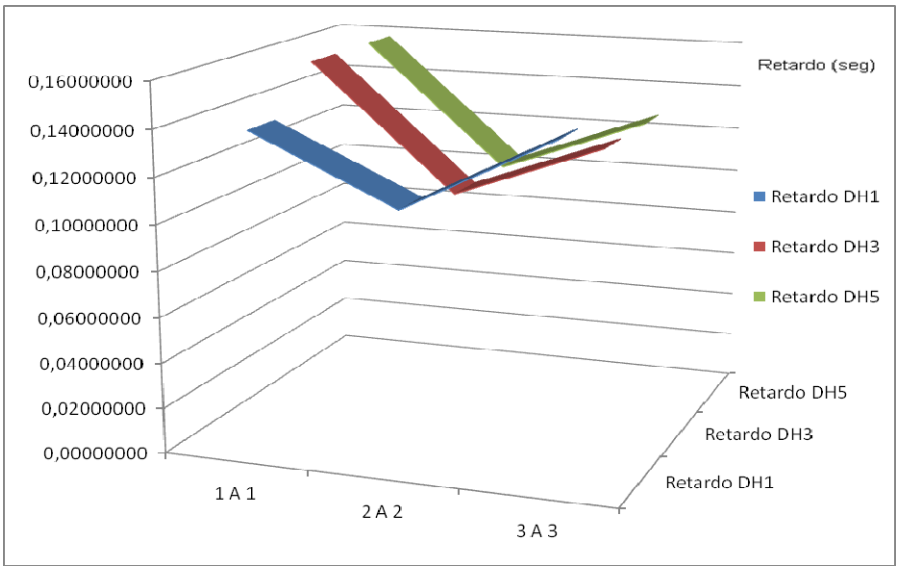


Figura 5.23 Retardo promedio para flujo de datos Full-duplex, VAD y CODEC G.729

Como era de esperarse, el aumento de conexiones asociadas al punto de acceso, revelan un mayor consumo de ancho de banda, reflejado así en las respectivas Figuras 5.19, 5.22 y 5.25.

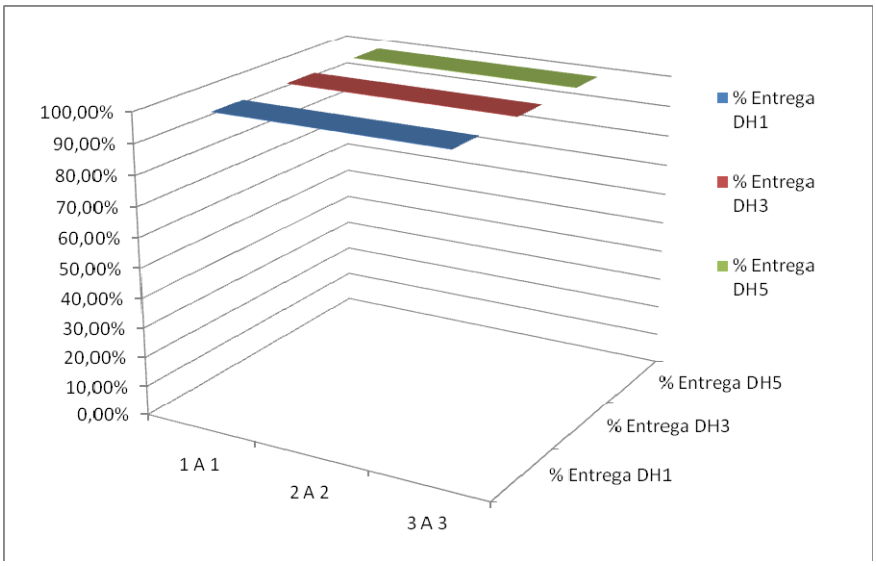


Figura 5.24 Porcentaje de entrega promedio para flujo de datos Full-duplex, VAD y CODEC G.729

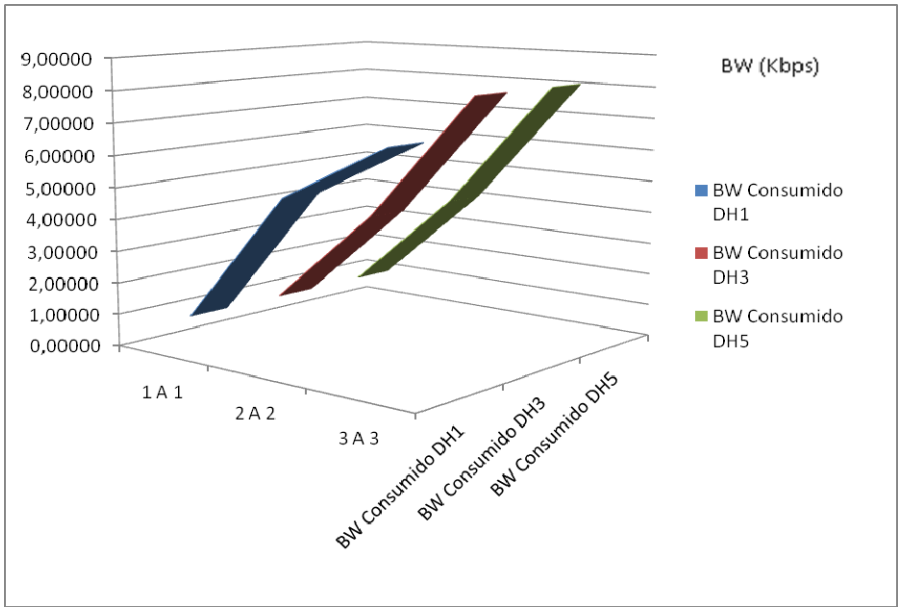


Figura 5.25 Ancho de banda consumido para flujo de datos Full-duplex, VAD y CODEC G.729

5.7 Cuarto Escenario

Con el fin de realizar una contra-validación simulando escenarios que no sean acordes con la arquitectura se procede a considerar el hecho de que 3 *piconets* activas en el mismo espectro de frecuencia con un flujo constante de datos y VAD, cuyos tipos de conexiones son *simplex* pueden afectarse entre sí, denominándolas conexiones paralelas.

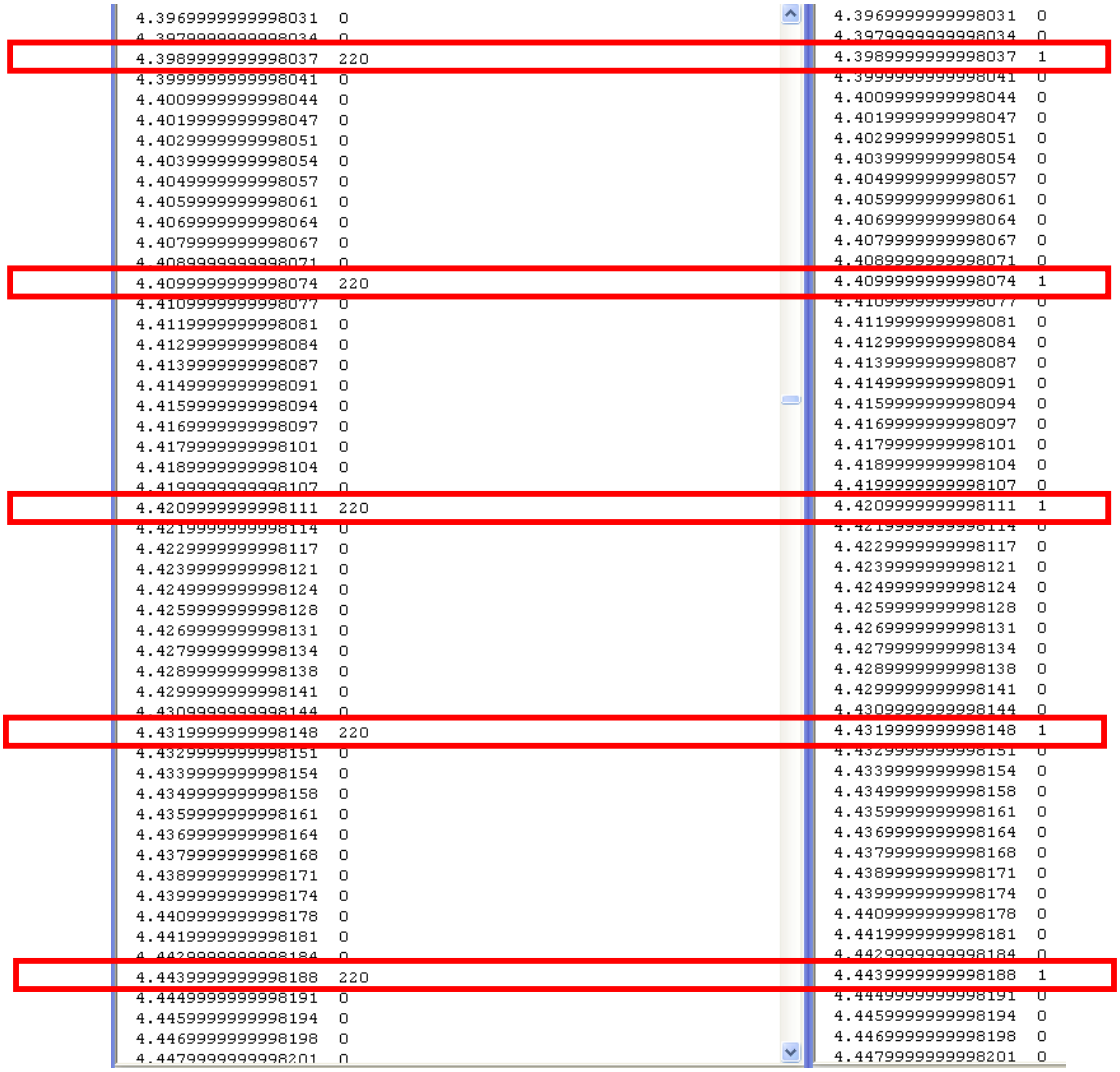
PARALELAS							
			RETARDO (seg)	ENVIADOS	PAQUETES PERDIDOS	% ENTREGA	BW CONSUMIDO (Kbps)
CBR	G.711	DH1	0,13578	84	0	100,00%	88,70400
		DH3	0,1293756	84	0	100,00%	88,70400
		DH5	0,1270853	83	0	100,00%	87,64800
	G.723	DH1	0,1317533	28	0	100,00%	11,28960
		DH3	0,126165	27	0	100,00%	10,88640
		DH5	0,126165	27	0	100,00%	10,88640
	G.729	DH1	0,1325233	37	0	100,00%	14,20800
		DH3	0,1239946	37	0	100,00%	14,20800
		DH5	0,1239946	37	0	100,00%	14,20800

Tabla 5.9 Resultados de Simulación 3 Conexiones paralelas CBR

PARALELAS							
			RETARDO (seg)	ENVIADOS	PAQUETES PERDIDOS	% ENTREGA	BW CONSUMIDO (Kbps)
VAD	G.711	DH1	0,16045130	35	0	100,00%	36,96000
		DH3	0,09471143	57	0	100,00%	60,19200
		DH5	0,16775600	35	0	100,00%	36,96000
	G.723	DH1	0,13109160	10	0	100,00%	4,03200
		DH3	0,11970630	10	0	100,00%	4,03200
		DH5	0,11970630	10	0	100,00%	4,03200
	G.729	DH1	0,15967660	14	0	100,00%	5,37600
		DH3	0,15217230	14	0	100,00%	5,37600
		DH5	0,15217230	14	0	100,00%	5,37600

Tabla 5.10 Resultados de Simulación 3 Conexiones paralelas VAD

Esta configuración difiere de la planteada en la arquitectura donde se especificó que la forma más conveniente de realizar la conexión de los usuarios móviles con los usuarios pertenecientes a la red cableada es a través de un punto de acceso único en cierto rango de cobertura.



4.3969999999998031	0	4.3969999999998031 0
4.3979999999998034	0	4.3979999999998034 0
4.3989999999998037	220	4.3989999999998037 1
4.3999999999998041	0	4.3999999999998041 0
4.4009999999998044	0	4.4009999999998044 0
4.4019999999998047	0	4.4019999999998047 0
4.4029999999998051	0	4.4029999999998051 0
4.4039999999998054	0	4.4039999999998054 0
4.4049999999998057	0	4.4049999999998057 0
4.4059999999998061	0	4.4059999999998061 0
4.4069999999998064	0	4.4069999999998064 0
4.4079999999998067	0	4.4079999999998067 0
4.4089999999998071	0	4.4089999999998071 0
4.4099999999998074	220	4.4099999999998074 1
4.4109999999998077	0	4.4109999999998077 0
4.4119999999998081	0	4.4119999999998081 0
4.4129999999998084	0	4.4129999999998084 0
4.4139999999998087	0	4.4139999999998087 0
4.4149999999998091	0	4.4149999999998091 0
4.4159999999998094	0	4.4159999999998094 0
4.4169999999998097	0	4.4169999999998097 0
4.4179999999998101	0	4.4179999999998101 0
4.4189999999998104	0	4.4189999999998104 0
4.4199999999998107	0	4.4199999999998107 0
4.4209999999998111	220	4.4209999999998111 1
4.4219999999998114	0	4.4219999999998114 0
4.4229999999998117	0	4.4229999999998117 0
4.4239999999998121	0	4.4239999999998121 0
4.4249999999998124	0	4.4249999999998124 0
4.4259999999998128	0	4.4259999999998128 0
4.4269999999998131	0	4.4269999999998131 0
4.4279999999998134	0	4.4279999999998134 0
4.4289999999998138	0	4.4289999999998138 0
4.4299999999998141	0	4.4299999999998141 0
4.4309999999998144	0	4.4309999999998144 0
4.4319999999998148	220	4.4319999999998148 1
4.4329999999998151	0	4.4329999999998151 0
4.4339999999998154	0	4.4339999999998154 0
4.4349999999998158	0	4.4349999999998158 0
4.4359999999998161	0	4.4359999999998161 0
4.4369999999998164	0	4.4369999999998164 0
4.4379999999998168	0	4.4379999999998168 0
4.4389999999998171	0	4.4389999999998171 0
4.4399999999998174	0	4.4399999999998174 0
4.4409999999998178	0	4.4409999999998178 0
4.4419999999998181	0	4.4419999999998181 0
4.4429999999998184	0	4.4429999999998184 0
4.4439999999998188	220	4.4439999999998188 1
4.4449999999998191	0	4.4449999999998191 0
4.4459999999998194	0	4.4459999999998194 0
4.4469999999998198	0	4.4469999999998198 0
4.4479999999998201	0	4.4479999999998201 0

Figura 5.26 Archivo de traza de paquetes

La herramienta de simulación UCBT sobre NS-2 permite la verificación de los paquetes recibidos a través de la generación de archivos de salida obtenidos a partir de un algoritmo de traza de paquetes. En la Figura 5.26 se puede verificar la correcta transmisión de éstos y el tamaño en bytes del paquete en recepción.

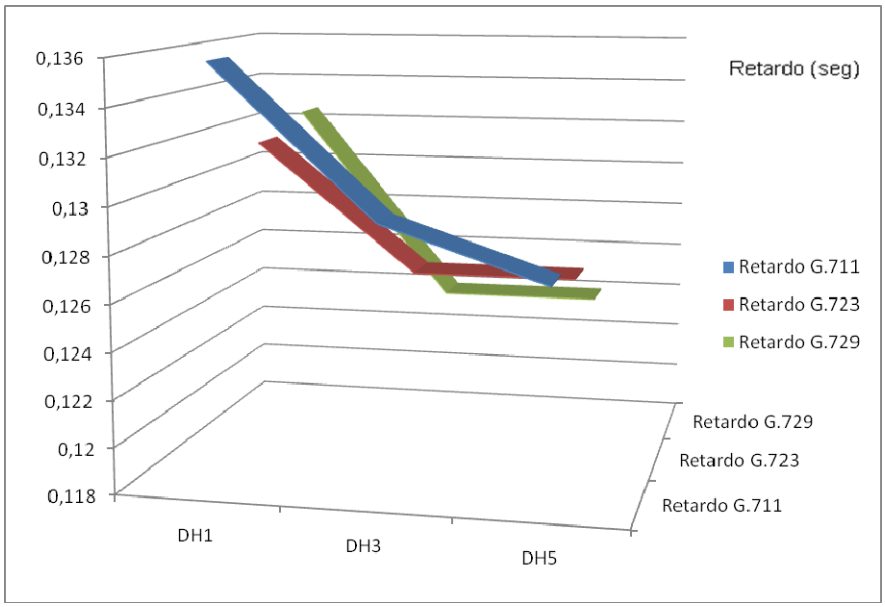


Figura 5.27 Retardo promedio para flujo de datos Simplex y CBR

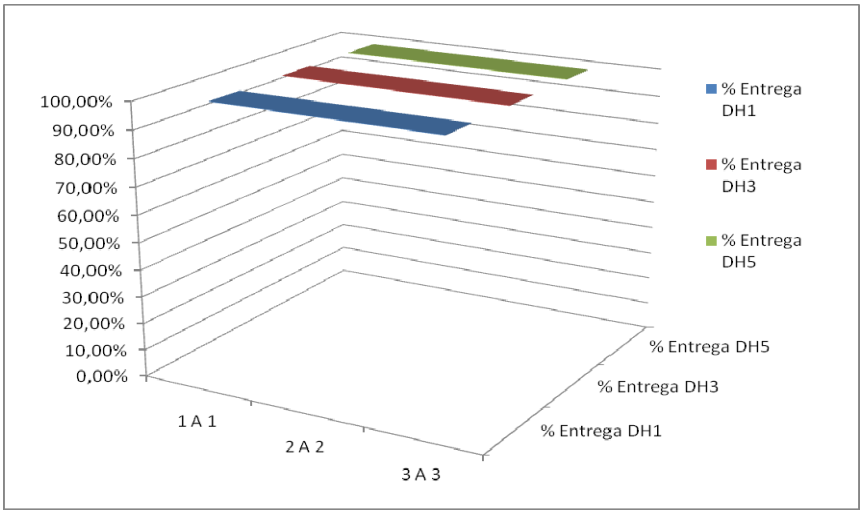


Figura 5.28 Porcentaje de entrega promedio para flujo de datos Simplex CBR

En la Figura 5.27 y 5.30 se aprecia que los retardos presentados durante todas la comunicaciones, alcanzan valores significativos (por encima de los 100ms), contrastando así con los valores obtenidos en el escenario uno.

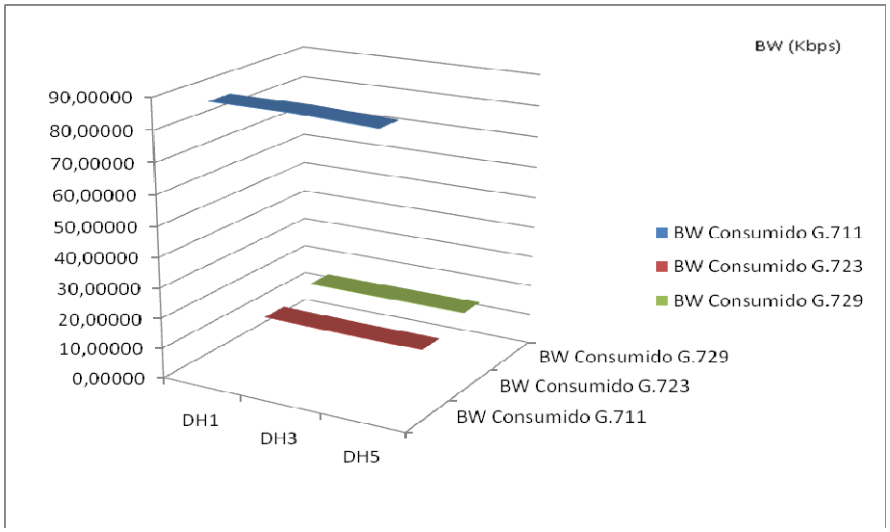


Figura 5.29 Ancho de banda consumido para flujo de datos Simplex y CBR

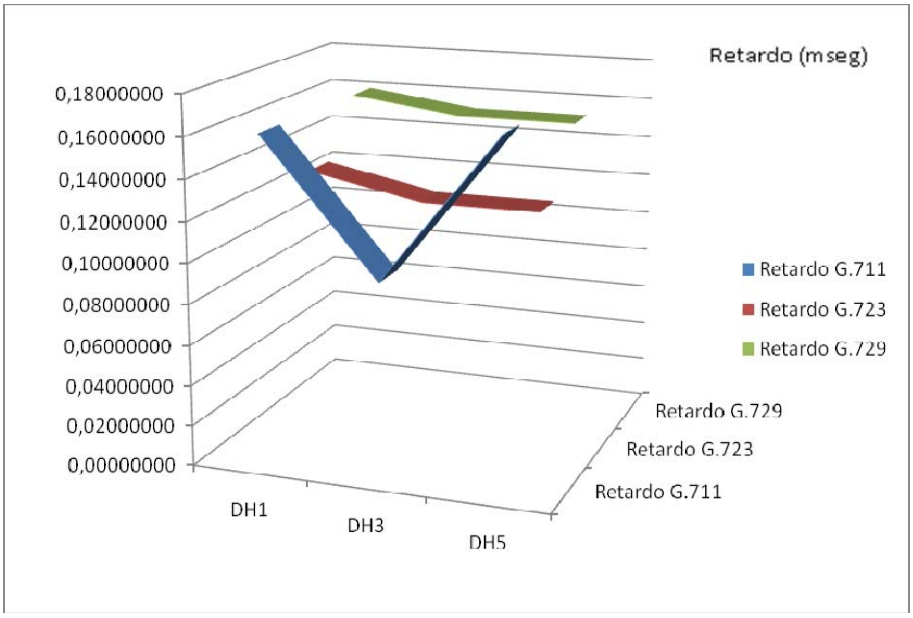


Figura 5.30 Retardo promedio para flujo de datos Simplex y VAD

Los parámetros de porcentaje de entrega (Figura 5.28 y 5.31) y ancho de banda consumido no presentan variaciones significativas (Figura 5.29), a excepción del ancho de banda consumido cuando se trabaja con tipo de tráfico VAD, CODEC G.711 y paquetes de datos DH3 (Figura 5.32), ya que la cantidad de paquetes generados es casi el doble de los enviados con DH1 y DH5.

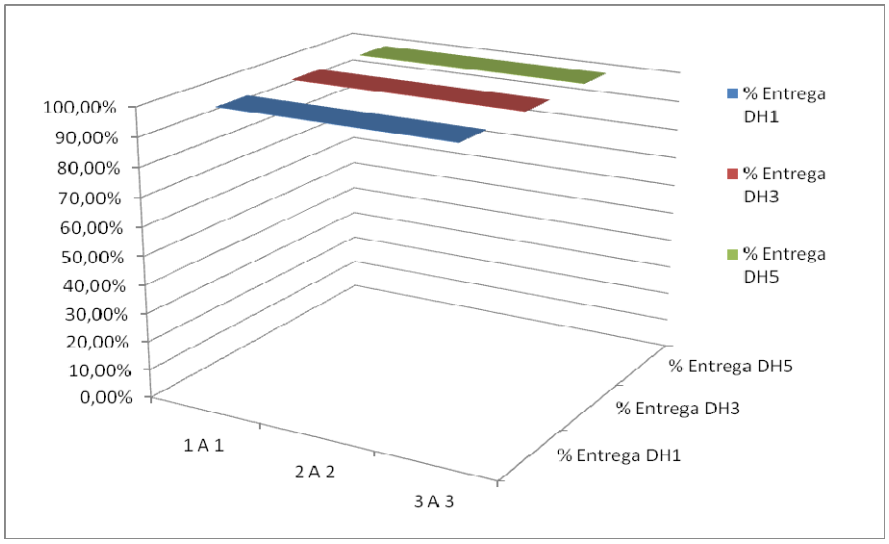


Figura 5.31 Porcentaje de entrega promedio para flujo de datos Simplex y VAD

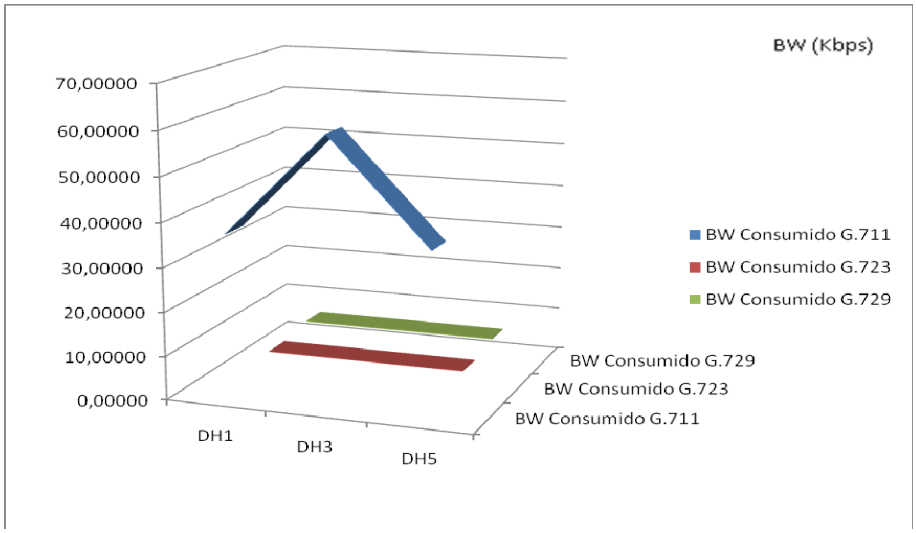


Figura 5.32 Ancho de banda consumido para flujo de datos Simplex y VAD

5.8 PROTOTIPO FUNCIONAL

Con el objetivo de contrastar los datos obtenidos en los escenarios de simulación, se implementó un prototipo funcional a fin de realizar el mismo tipo de mediciones de desempeño y conectividad. Un nodo Bluetooth se implementó como el punto de acceso definido en la arquitectura.

Como se menciono en el capítulo 4, el punto de acceso se encarga de realizar el puente para interconectar la tecnología Bluetooth y la red cableada IEEE 802.3, haciendo uso del perfil PAN e implementando la encapsulación de datos con BNEP; a la vez que implementa funcionalidades de enrutador de interfaz dual haciendo uso de NAT.

El nodo cliente a nivel Bluetooth fue implementado en una *laptop* donde su interfaz de red, fue un adaptador Bluetooth asociándose a la red PAN establecida por el punto de acceso ya mencionado, a fin de interactuar con equipos pertenecientes a la red IP cableada (Figura 5.33).

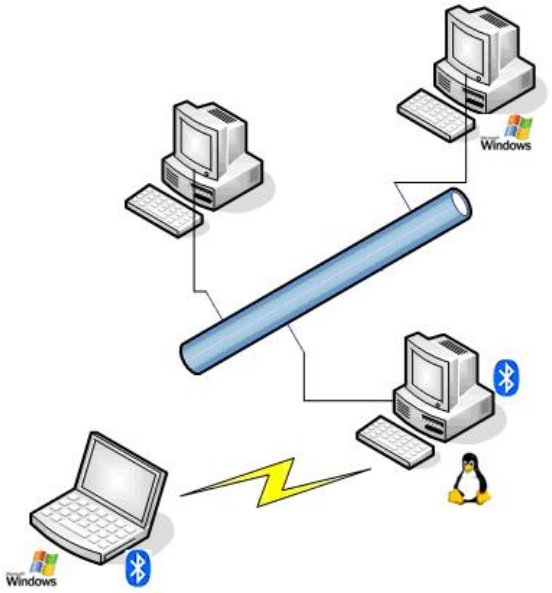


Figura 5.33 Diagrama de Conexión del Prototipo Funcional

A fin de obtener dichas medidas, se utilizaron generadores de tráfico como son Iperf [43] y D-ITG (*Distributed Internet Traffic Generator*) [44]. Con el generador de tráfico Iperf se midió el ancho de banda provisto por el enlace siendo el nodo Bluetooth esclavo, el cliente Iperf que generaba tráfico hacia un equipo perteneciente a la red IEEE 802.3 siendo éste último el servidor Iperf.

Estos datos se resumen en la tabla 5.11.

ANCHO DE BANDA CONEXIÓN EXTREMO A EXTREMO (Kbps)				
	TOMA 1		TOMA 2	
	CLIENTE	SERVIDOR	CLIENTE	SERVIDOR
Promedio	204	206	206	208
	291	293	136	136
	200	202	242	245
	221	222	206	208
	261	262	304	306
	279	281	310	312
	242,667	244,333	234,000	235,833
Parcial	243,500		234,917	
Total	239,208			

Tabla 5.11 Ancho de banda medido en Iperf

Además se consideró que la llamada sería en doble vía por lo que también era conveniente generar un tráfico ida y vuelta utilizando la herramienta D-ITG, por este motivo la generación de flujos de datos se realizó considerando la cantidad de conexiones, que a su vez generó un archivo (log) y al pasarlo al ITGDec se decodifica y presenta en formato "*human-readable*", en que se encuentran los datos de cantidad de flujos, retardo, jitter, pérdida de paquetes y ancho de banda consumido, generando tipos de flujos CBR (Tabla 5.12) y VAD (Tabla 5.13).

FULL-DUPLEX							
			RETARDO (seg)	% PERDIDA	PAQUETES PERDIDOS	% ENTREGA	BW CONSUMIDO (Kbps)
CBR	G.711	1 A 1	0,010485	0,00%	0	100,00%	70,35431
		2 A 2	0,504074	0,15%	6	99,85%	127,60231
		3 A 3	2,400781	0,62%	37	99,38%	139,42139
	G.723	1 A 1	0,022207	0,19%	1	99,81%	7,90696
		2 A 2	0,012846	0,00%	0	100,00%	15,81287
		3 A 3	0,015544	0,00%	0	100,00%	23,70149
	G.729	1 A 1	0,013178	0,00%	0	100,00%	11,20000
		2 A 2	0,015232	0,50%	10	99,50%	22,25671
		3 A 3	0,016630	2,87%	86	97,13%	32,56438

Tabla 5.12 Medidas de Desempeño del Prototipo Funcional para tráfico Full-duplex CBR

FULL-DUPLEX							
			RETARDO (seg)	% PERDIDA	PAQUETES PERDIDOS	% ENTREGA	BW CONSUMIDO (Kbps)
VAD	G.711	1 A 1	0,014736	0,80%	16	99,20%	47,50674
		2 A 2	0,021587	3,80%	152	96,20%	92,23326
		3 A 3	1,744981	0,22%	13	99,78%	111,17889
	G.723	1 A 1	0,017974	0,00%	0	100,00%	5,62642
		2 A 2	0,012828	0,00%	0	100,00%	11,24436
		3 A 3	0,010213	0,00%	0	100,00%	16,83946
	G.729	1 A 1	0,013016	0,00%	0	100,00%	8,40504
		2 A 2	0,009820	0,00%	0	100,00%	16,79192
		3 A 3	0,009719	0,00%	0	100,00%	25,14478

Tabla 5.13 Medidas de Desempeño del Prototipo Funcional para tráfico Full-duplex VAD

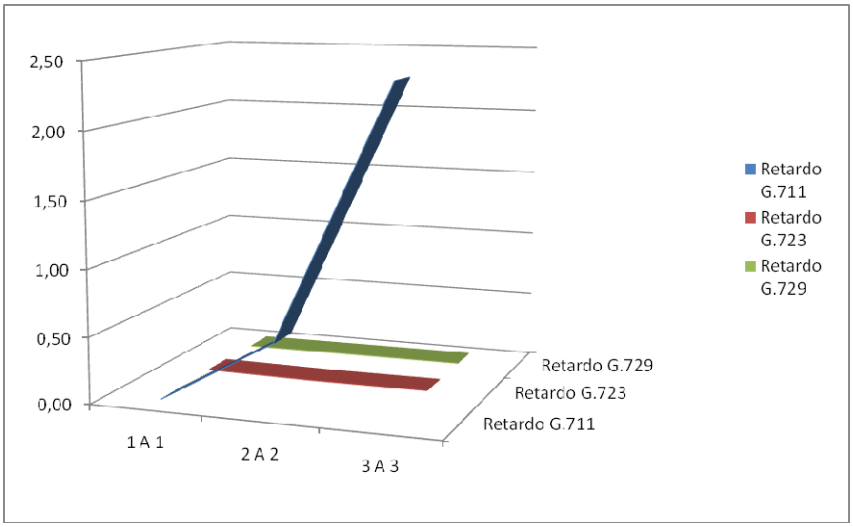


Figura 5.34 Retardo promedio del prototipo funcional Full-duplex y CBR

El ancho de banda consumido durante la evaluación del prototipo funcional con sistemas VAD (Figura 5.39) y CBR (Figura 5.36) presenta variaciones con respecto al mismo parámetro evaluado durante la simulación. Una de las razones probables es el tipo de generación de tráfico artificial utilizado durante el proceso de simulación y el tipo de generación de tráfico implementado por el generador de tráfico D-ITG. Otra de las razones, constituye la diferencia en el número de dispositivos móviles involucrados en los dos escenarios, es decir durante la evaluación del prototipo funcional se generaron los

flujos desde un solo cliente móvil (debido a la limitación de equipos) mientras que en la simulación se generaron los flujos desde diferentes clientes móviles.

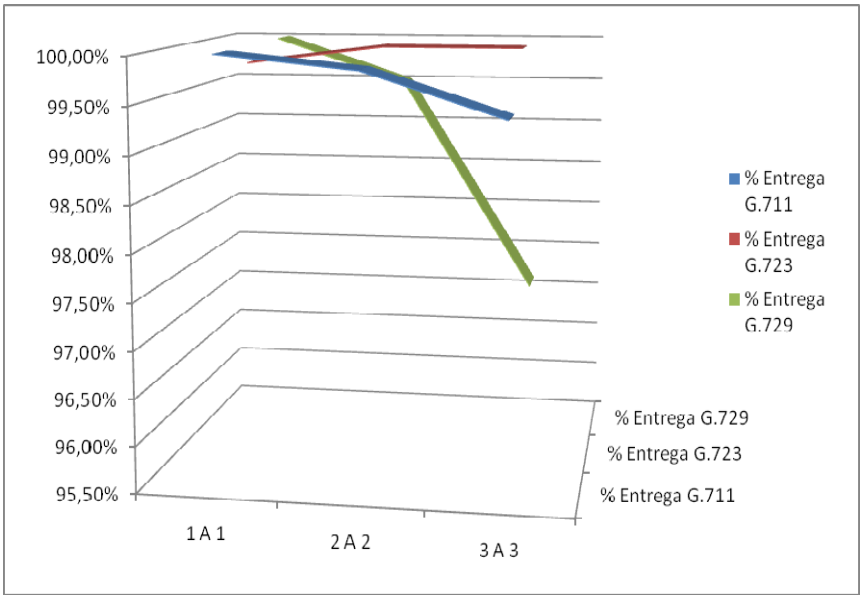


Figura 5.35 Porcentaje de entrega promedio del prototipo funcional Full-duplex y CBR

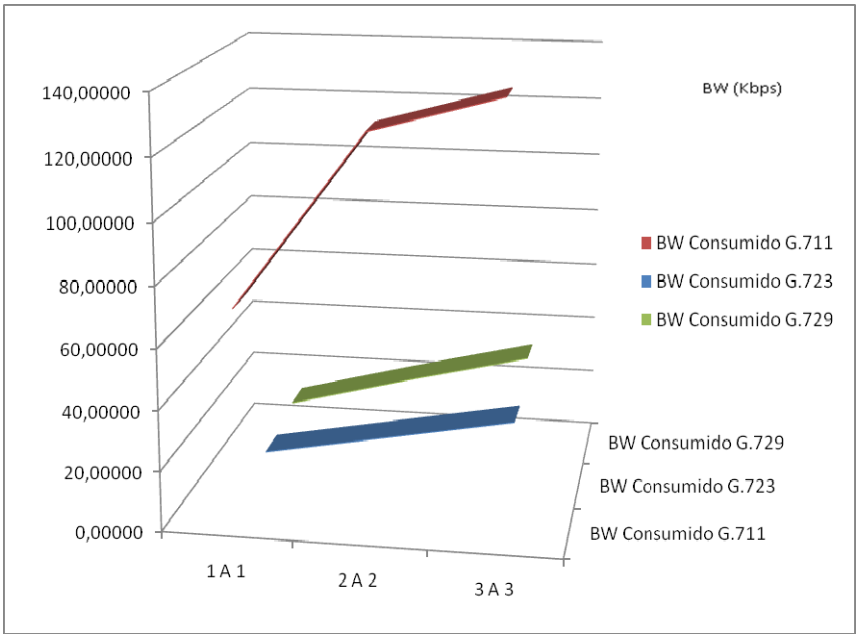


Figura 5.36 Ancho de banda consumido del prototipo funcional Full-duplex y CBR

Se observa, que tanto el retardo (Figura 5.34 y 5.37) como la perdida de paquetes se incrementan en el prototipo funcional conforme aumenta el número de conexiones que trabajan con flujos contantes de datos (Figura 5.35), a diferencia del tipo de flujo VAD, el cual no presenta perdidas cuando se utilizan los CODECs G.723 y G.729, ocurriendo una particularidad con el CODEC G.711, el cual presenta perdidas de paquetes debido al gran tamaño de éstos (Figura 5.38).

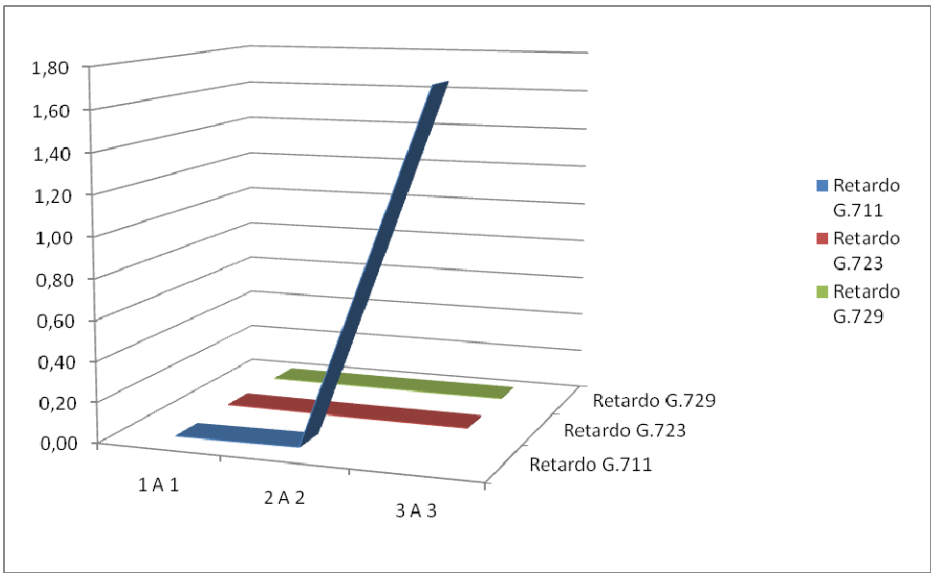


Figura 5.37 Retardo promedio del prototipo funcional Full-duplex y VAD

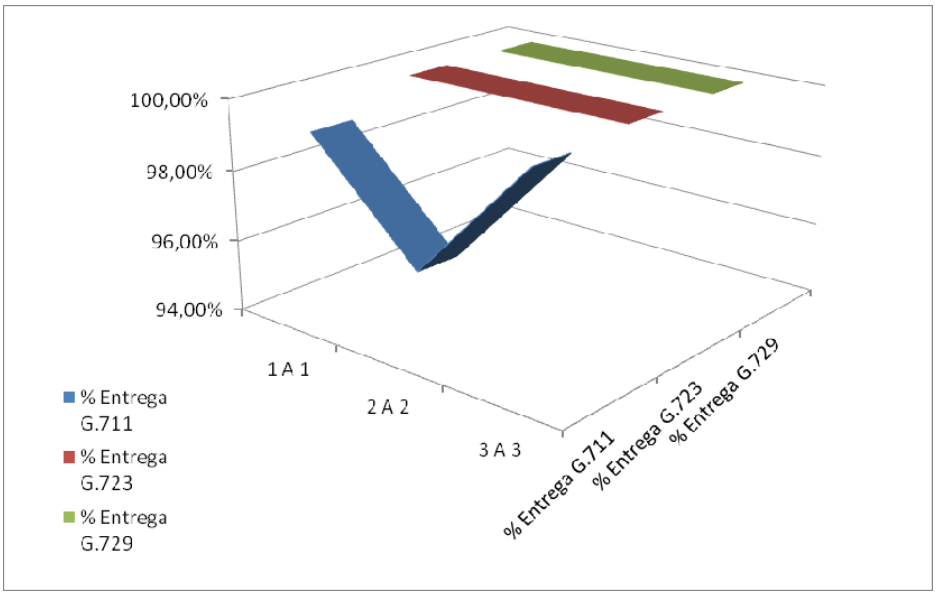


Figura 5.38 Porcentaje de entrega promedio del prototipo funcional Full-duplex y VAD

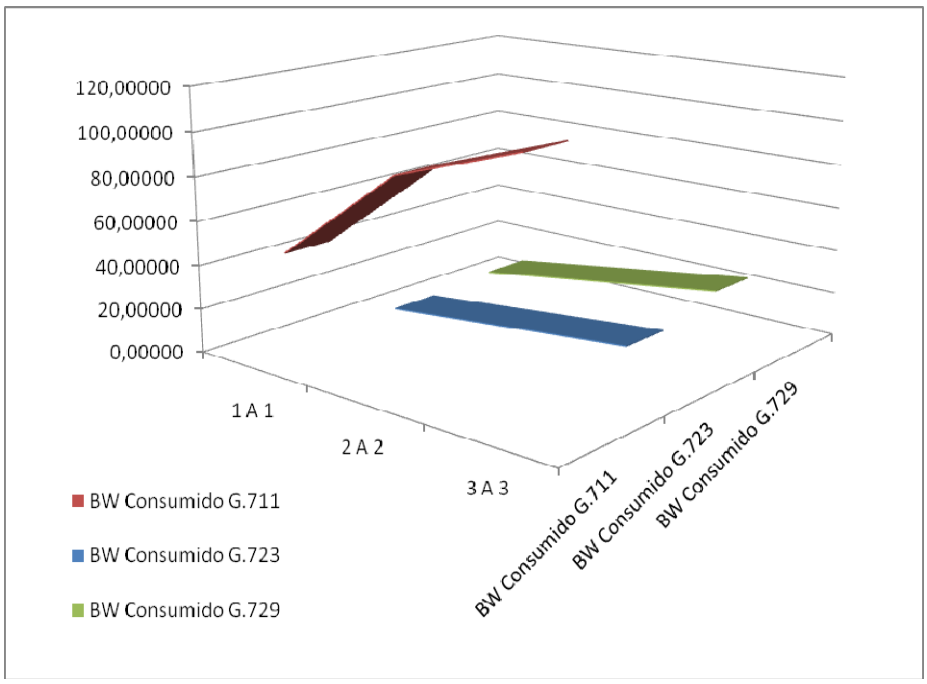


Figura 5.39 Ancho de banda consumido del prototipo funcional Full-duplex y VAD

En lo que respecta al ancho de banda consumido, el tamaño de paquete transmitido en base al CODEC utilizado, refleja la demanda de ancho de banda conforme incrementan la cantidad de conexiones (Figuras 5.36 y 5.39). Además, se realiza un análisis de los porcentajes de reducción del ancho de banda en conexiones Full-duplex tanto en simulación como en el prototipo. Dichas conexiones representan el tráfico de voz que puede ser CBR o VAD y se obtienen las Tablas 5.14 y 5.15.

Como se plantea en el literal 2.3.8 del presente documento, la detección de actividad de voz genera una reducción del ancho de banda en un 35% teóricamente, por tanto las Tablas 5.14 y 5.15 nos demuestran que el prototipo funcional al generar tráfico VAD se acerca a dicho porcentaje de reducción establecido para tráfico de VoIP en comparación con los datos obtenidos de la simulación, mostrando así que la interoperabilidad de la tecnología Bluetooth con redes cableadas IP puede alcanzar los estándares establecidos para un préstamo eficiente de servicios de voz sobre IP.

En general los resultados obtenidos a partir de la evaluación de la arquitectura por medio del prototipo funcional, fueron acordes con los obtenidos a partir de la simulación. Las variables medidas presentan concordancia, dependiendo del número de comunicaciones llevadas a cabo y el tipo de CODEC utilizado. Una de las observaciones más relevantes corresponde al retardo presentado cuando se realizan tres comunicaciones simultáneas. Dicho ítem presenta una diferencia significativa en ambas medidas lo que confirma la importancia de construcción de prototipos reales que contrasten los resultados de las simulaciones de redes.

FULL DUPLEX - SIMULACIÓN			
			% REDUCCIÓN BW
G.711	DH1	1 A 1	85,04%
		2 A 2	69,46%
		3 A 3	76,16%
	DH3	1 A 1	84,55%
		2 A 2	62,68%
		3 A 3	65,84%
	DH5	1 A 1	84,43%
		2 A 2	72,31%
		3 A 3	75,16%
G.723	DH1	1 A 1	84,62%
		2 A 2	74,36%
		3 A 3	78,13%
	DH3	1 A 1	84,21%
		2 A 2	71,79%
		3 A 3	78,13%
	DH5	1 A 1	84,21%
		2 A 2	71,79%
		3 A 3	78,13%
G.729	DH1	1 A 1	84,31%
		2 A 2	62,26%
		3 A 3	78,57%
	DH3	1 A 1	84,31%
		2 A 2	69,23%
		3 A 3	75,29%
	DH5	1 A 1	84,31%
		2 A 2	69,23%
		3 A 3	75,29%

Tabla 5.14 Porcentaje de reducción de ancho de banda – simulación

FULL DUPLEX - PROTOTIPO		
		% REDUCCIÓN BW
G.711	1 A 1	32,48%
	2 A 2	27,72%
	3 A 3	20,26%
G.723	1 A 1	28,84%
	2 A 2	28,89%
	3 A 3	28,95%
G.729	1 A 1	24,96%
	2 A 2	24,55%
	3 A 3	22,78%

Tabla 5.15 Porcentaje de reducción de ancho de banda – prototipo

6. CONCLUSIONES

- Durante el presente trabajo de grado se recopilaron conceptos asociados a las tecnologías Bluetooth y *Ethernet* con el fin de establecer un marco teórico para la formulación de criterios para la interoperabilidad de estas tecnologías, con un enfoque hacia la prestación de servicios de VoIP.

- Se estudiaron los principales factores que afectan la calidad de la VoIP en redes de paquetes, con el fin de estructurar un marco para la construcción de una arquitectura que incorporara elementos de interoperabilidad para la prestación de servicios de voz en tiempo real.

- Se definió y se dió una descripción detallada del perfil PAN como el encargado de brindar capacidades *Ethernet* a los dispositivos Bluetooth creando conexiones con capacidades IP. Se explicó a profundidad el protocolo BNEP que soporta dicho perfil y se plantearon los escenarios de aplicabilidad de éste donde se plantea un diagrama de cómo estarían ubicados los puntos de acceso en sistemas basados en la arquitectura propuesta.

- Se realizó un estudio a profundidad de los tipos de canales, enlaces, conexiones y tipos de paquetes brindados por la tecnología Bluetooth, especialmente los concernientes a comunicaciones basadas en envío y recepción de paquetes, incorporando un estudio sobre las capacidades de QoS brindadas por ésta.

- A partir de la definición de los criterios y parámetros de interoperabilidad, y las capacidades tecnológicas IP brindadas por Bluetooth, se planteó una arquitectura como soporte para la prestación de servicios de voz en tiempo real, en ambientes de rápida localización de personal.

- En base a un estudio concienzudo, se definió el simulador de redes y su paquete para Bluetooth que más se adaptaran a las necesidades del proyecto, con el fin de simular una serie de escenarios que pudieran conducir a la validación de la arquitectura propuesta.

- Adicionalmente, se implementó un prototipo funcional que ayudó a contrastar los resultados obtenidos a partir de la validación de la arquitectura vía simulación, con el objetivo de robustecer los resultados.

- En el proceso de validación se midieron variables críticas para la prestación de servicios de VoIP, tales como el retardo, porcentaje de entrega de paquetes y ancho de banda consumido, en base al tipo de CODEC utilizado, tipo de paquetes utilizados en la comunicación, con el fin de determinar el grado de desempeño de un sistema implementado en base a la arquitectura propuesta

- Se analizaron los resultados de cada uno de los escenarios propuestos estableciendo directivas de comportamiento del sistema que ayudan a formular recomendaciones

para implementaciones reales de sistemas de prestación de servicios de voz basados en la arquitectura construida.

- Se contrastaron y validaron los conceptos teóricos establecidos para el préstamo de servicios de voz, con los datos obtenidos, tabulados y analizados a partir de un prototipo funcional que permitió la validación de la arquitectura planteada en el presente proyecto de investigación.

- El diseño flexible y modular de la arquitectura permite la futura extensión del set de posibles aplicaciones que requieran una plataforma IP para su funcionamiento, soportado fundamentalmente sobre la completa interconectividad de tecnologías.

7. RECOMENDACIONES

- A pesar de las capacidades de QoS que pueden ser brindadas por la tecnología Bluetooth, este tema no fue abordado a profundidad, por lo que se recomiendan trabajos encaminados hacia una evaluación detallada de ésta capacidad, en lo que respecta al manejo de retardo en la transmisión, gestión de los *slots* y tamaños de paquetes en función de los servicios que los requieran.

- En el presente trabajo de grado se abordó la prestación de servicios de voz bajo la perspectiva de construir una arquitectura que soporte sistemas de dicha naturaleza, por consiguiente sería conveniente trabajos futuros de diseño de sistemas de servicios de voz basados en la arquitectura con la posible utilización de criterios de diseño para WLAN, teniendo en cuenta las diferencias existentes entre ambas tecnologías tales como método de acceso al medio, radios de cobertura, potencias de transmisión, diseños de antenas; siempre manteniendo presente la construcción de estos sistemas como un complemento de la extensibilidad de las redes ya existentes dentro de las instituciones.

- Las herramientas de simulación analizadas ofrecen características para evaluación de interferencia producida por tecnologías inalámbricas operando en la misma banda de frecuencia, campo que presenta un amplio panorama de investigación. Por lo tanto se recomienda hacer análisis utilizando las herramientas de simulación que permitan establecer la viabilidad de prestar servicios de VoIP en ambientes de interferencia.

- El *handover* o *roaming* en redes de área personal no fue abordado en detalle en el presente trabajo de grado. Definir algoritmos y métodos eficientes para cumplir esta tarea es un tópico de investigación que se recomienda para mejorar la funcionalidad de la arquitectura propuesta.

- Como se detalló durante el presente trabajo, las aplicaciones VoIP para usuarios móviles no se incorporaron dentro de la arquitectura propuesta, por consiguiente se recomienda enfocar investigaciones futuras hacia la implementación de estas aplicaciones que incluyan el *handover* en lenguajes de programación tales como J2ME para los diferentes sistemas operativos existentes, especialmente sistemas operativos para dispositivos móviles de fuente abierta.

- Aunque en el presente trabajo de grado se abordó la implementación de un prototipo funcional, éste fue de pequeña escala, con solo los elementos necesarios para la validación de la arquitectura. Se propone darle continuidad a este tipo de implementaciones evaluando prototipos o sistemas reales de mayor escala y complejidad.

- No obstante resulte un trabajo extenso y complejo, la implementación de un paquete de simulación Bluetooth que basado en las falencias de los simuladores implementados actualmente, incorporen nuevas capacidades, tales como simulación de nuevos servicios, protocolos, mejor manejo de resultados de simulación,

incorporación de medición de más variables de interés, etc. podrían ser un foco de investigación interesante y constructivo.

- Adicionalmente se recomiendan trabajos encaminados a lograr la compatibilidad del paquete de simulación UCBT con el entorno gráfico de NS-2, Nam, lo que beneficiaría en gran medida las posibilidades ofrecidas.

8. REFERENCIAS

[1] Bluetooth *Special Interest Group*. Disponible en Internet: http://www.bluetooth.com

[2] Ministerio De Comunicaciones De Colombia. CuadroAtribucion.pdf (Archivo PDF), Disponible en Internet: http://www.mincomunicaciones.gov.co.

[3] Bluetooth *Special Interest Group*. "*Bluetooth Core*", *Specification of the* Bluetooth *System,* Versión 2.0 + EDR, 4 de Noviembre de 2004. Disponible en Internet: https://www.bluetooth.org.

[4] Bluetooth *Special Interest Group.* Bluetooth *Profiles, Specification of the* Bluetooth *System*, Versión 1.1, 22 de Febrero de 2003. Disponible en Internet:`
http://www.bluetooth.org.

[5] IEEE. *IEEE standard 802.3 2000 edition*
URL: http://www.ieeexplore.org.

[6] Soporte Bluetooth en Linux 2.6, Álvaro del Castillo San Félix. Documento disponible en: http://acs.barrapunto.org/~acs/LinuxBluetooth.pdf

[7] Bluetooth *Special Interest Group. Personal Area Networking Profile. Specification of the* Bluetooth *System*, Versión 1.0, 14 de Febrero de 2003. Disponible en Internet:
http://www.bluetooth.org.

[8] Bluetooth *Special Interest Group.* Bluetooth *Network Encapsulation Protocol (BNEP) Specification. Specification of the* Bluetooth *System*, Versión 1.0, 14 de Febrero de 2003. Disponible en Internet: http://www.bluetooth.org.

[9] *Ethernet Vendor Address Components*. Disponible en Internet:
http://www.iana.org/assignments/ethernet-numbers

[10] *Digital Equipment Corporation, Intel Corporation, Xerox Corporation. The Ethernet - A Local Area Network*, Versión 1.0. Septiembre de 1980.

[11] Recomendación G.711. ITU-T. Disponible en: http://www.itu.int/rec/T-REC-G.711/es

[12] Recomendación G.723.1. ITU-T. Disponible en: http://www.itu.int/rec/T-REC-G.723.1/es

[13] Recomendación G.729. ITU-T. Disponible en: http://www.itu.int/rec/T-REC-G.729/es

[14] Christian Benavides – Javier Andrés Gómez: Ing. Guefry Agredo. Diseño y simulación de una red WLAN Multicelda "*Outdoor*" con soporte para 802.11e. Proyecto de Grado, Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca. Popayán, 2007.

[15] Cisco, *Understanding Delay in Packet Voice Networks*. Documento PDF disponible en: http://www.cisco.com/warp/public/788/voip/delay-details.pdf

[16] *Vocal Technologies Ltd*. CODECs VoIP. Página web disponible en: http://www.vocal.com/index.html

[17] *Newport Networks. VoIP Bandwidth Calculation*. Documento PDF disponible en: http://www.newport-networks.com/cust-docs/52-VoIP-Bandwidth.pdf

[18] Mónica Lombana y Juan Gonzáles. Director: Ing. Guefry Agredo. Prototipo Experimental de VoIP sobre WLAN para Entornos Empresariales. Proyecto de Grado, Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca. Popayán, 2005.

[19] Chan, Wah-Chun. *Quality-of-service in IP services over* Bluetooth *ad-hoc networks*. Documento PDF disponible en: http://portal.acm.org/citation.cfm?id=959217

[20] *The* Bluetooth *Special Interest Group*. *Quality of service in* Bluetooth *networking*. Documento PDF disponible en: http://ing.ctit.utwente.nl/WU4/Documents/Bluetooth_QoS_ING_A_part_I.pdf

[21] *Metrics for the Evaluation of Congestion Control Mechanisms*, S. Floyd, 30 de Julio de 2007. Página web disponible en: http://tools.ietf.org/html/draft-irtf-tmrg-metrics-10

[22] QoS en telefonía IP. Página web disponible en: http://www.um.es/atica/qos-en-telefonia-ip

[23] Smith, Jared. Meggelen, Jim Van. Madsen, Leif. *Asterisk, The future of Telephony*. Septiembre de 2005. Disponible en: http://books.google.com.co/books?id=9_wRFy5OGw4C&dq=&pg=PP1&ots=7gVfqMsG8i&sig=q3ZnRAspC9Tg5IeWbc9LE2BqM1Y&prev=http://www.google.com.co/search%3Fhl%3Des%26q%3DAsterisk%252C%2BThe%2Bfuture%2Bof%2BTelephony%255D%26btnG%3DBuscar%2Bcon%2BGoogle%26meta%3D&sa=X&oi=print&ct=title

[24] Vrizlynn L. L. Thing, Henry C. J. Lee, Xiao Ni, Daqing Zhang. *Design and Implementation of IP over Bluetooth for IP Mobility Support in a Heterogeneous Environment*. Documento PDF disponible en: http://whitepapers.techrepublic.com.com/whitepaper.aspx?docid=152166

[25] Dr. Zeev Weissman Chief Scientist. Bluetooth *vs. 802.11b: The access network perspective*. White Paper disponible en: http://www.tadlys.com/media/downloads/tadlys%20bluetooth%20vs%20802_11_b.pdf

[26] Al-Hawamdeh, Al-Muthanna. Khan, Adil. Hayat Khiyal, M.Sikander. *IPSec Based* Bluetooth *Security System*. European Journal of Scientific Research. Volume 16, No 1 Enero, 2007. Disponible en: www.eurojournals.com/ejsr%2016%201.pdf

[27] Chen, Ming-Chiao. Chen, Jiann-Liang. Lo, Fan-Yi. Yao, Pei-Chun. *Seamless Handoff Mechanisms for* Bluetooth-*IP Network Services*. Artículo PDF disponible en: www.ndhu.edu.tw/~rdoffice/exchange/CMC-paper.pdf

[28] *Windows for Devices. Windows Mobile Phones.* Página disponible en: http://www.windowsfordevices.com/articles/AT2468909181.html

[29] HP iPAQ rx3715 *Mobile Media Companion*. Página disponible en: http://h18000.www1.hp.com/products/quickspecs/11960_na/11960_na.HTML

[30] Symbian OS. *Open Source Comunity Center*. Página disponible en: http://www.symbianos.org/

[31] *Symbian OS*. *Smartphones*. Página disponible en: http://www.symbian.com/news/pr/2006/pr20063419.html

[32] GnuBox en teléfonos celulares SonyEricsson P80x/P90x/P910 y Nokia Serie 60. Página disponible en: http://gnubox.dnsalias.org/gnubox/

[33] *Symbian OS Version 9.3*. Página disponible en: http://www.symbian.com/files/rx/file7999.pdf

[34] *Linux Devices. Linux Mobile Phones.* Página disponible en: http://www.linuxdevices.com/articles/AT9423084269.html

[35] Herramienta avanzada de simulación OPNET. *Modeler Wireless Suite for Defense*. Acceso Agosto de 2007. Página disponible en: http://www.opnet.com/products/modeler/home.html

[36] Descripción del Simulador de Red NS-2. *The Network Simulator – ns-2*. Acceso Agosto de 2007. Página disponible en: http://www.isi.edu/nsnam/ns/

[37] Paquete de simulación basado en NS-2. *BlueHoc:* Bluetooth *Performance Evaluation Tool.* Acceso Agosto de 2007. Página disponible en: http://bluehoc.sourceforge.net/

[38] Modulo de simulación Bluetooth basado en NS-2. *Blueware: Bluetooth Simulator for ns*. Acceso Agosto de 2007. Página disponible en: http://nms.csail.mit.edu/projects/blueware/software/

[39] C. Hsu, Y. Joung. *An ns-based Bluetooth topology construction simulation environmen*". 36th Simposio de Simulación, Orlando – Florida (USA). Marzo 30 – Abril 02 de 2003. Documento disponible en: http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/8486/26746/01192808.pdf?arnumber=1192808

[40] C. Lee and A. Helal, *An ns-based* Bluetooth *LAP Simulator*. 26th conferencia anual de la IEEE en Redes de Computadores Locales (Local Computer Networks - LCN), Tampa, Florida, Nov 2001. Documento disponible en: http://ieeexplore.ieee.org/search/wrapper.jsp?arnumber=990832

[41] Módulo de red Bluetooth UCBT basado en NS-2. UCBT - Bluetooth *extension for NS2 at the University of Cincinnati*. Acceso Agosto de 2007. Página disponible en: http://www.ececs.uc.edu/~cdmc/ucbt/

[42] Wang, Qihe. *Scheduling and Simulation of Large Scale Wireless Personal Area Networks*. Tésis de PhD. Documento PDF de 136 páginas. Disponible en: http://www.ohiolink.edu/etd/send-pdf.cgi/Wang%20Qihe.pdf?acc_num=ucin1148050113

[43] Iperf. *University of Illinois* (USA). Disponible en: http://dast.nlanr.net/Projects/Iperf/

[44] *Distributed Internet Traffic Generator. Universita' degli Studi di Napoli "Federico II"* (Italia). Disponible en: http://www.grid.unina.it/software/ITG/