Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Laboratorio ICARO

Estudio de desempeño en un ambiente de comunicación *Bluetooth*. Caso de estudio Piconet y Scatternet.

Trabajo Especial de Grado presentado ante la Ilustre Universidad Central de Venezuela

Por los Bachilleres:
Héctor Rafael Navarro Belandria hectornucv@gmail.com
Eduardo José Pérez Quiñonez eduardoucv@yahoo.com

para optar al título de **Licenciado en Computación**

Tutor: David Pérez Abreu

Caracas, Junio de 2009

Universidad Central de Venezuela

Facultad de Ciencias
Escuela de Computación
Laboratorio ICARO

ACTA DEL VEREDICTO

Quienes suscriben, Miembros del Jurado designado por el Consejo de la Escuela de Computación para examinar el Trabajo Especial de Grado, presentado por los Bachilleres Héctor Navarro C.I.:15507400 y Eduardo Pérez C.I.:14049555, con el título **"Estudio de desempeño en un ambiente de comunicación *Bluetooth*. Caso de estudio *Piconet* y *Scatternet*."**, a los fines de cumplir con el requisito legal para optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído el trabajo por cada uno de los Miembros del Jurado, se fijó el día 16 de junio de 2009, a las 12:00 pm, para que sus autores lo defendieran en forma pública, en <lugarPresentación>, lo cual estos realizaron mediante una exposición oral de su contenido, y luego respondieron satisfactoriamente a las preguntas que les fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fe de lo cual se levanta la presente acta, en Caracas el <día> de <mes> de <año>, dejándose también constancia de que actuó como Coordinador del Jurado el Profesor Tutor David Pérez.

Prof. David Pérez
(Tutor)

Prof. Carlos Acosta
(Jurado Principal)

Prof. Juan Carlos Fernández
(Jurado Principal)

RESUMEN

Título:

Estudio de desempeño en un ambiente de comunicación *Bluetooth*. Caso de estudio Piconet y Scatternet.

Autor(es):

Héctor R. Navarro B.

Eduardo J. Pérez Q.

Tutor:

Prof. David Pérez

La siguiente investigación trata de presentar un método que permita la interconexión de varias *Piconets (*una red de dispositivos que se conectan utilizando *Bluetooth), estas agrupaciones son conocidas con el nombre de Scatternet* y estudiaremos el impacto en el desempeño de la comunicación *extremo a extremo* entre diversos dispositivos *Bluetooth*.

A pesar de la popularidad de la tecnología *Bluetooth* para la interconexión inalámbrica de dispositivos electrónicos tales como: teclados, mouse, impresoras, teléfonos móviles, auriculares, PDA (Asistente Digital Personal) y computadores; existen pocos trabajos de investigación para la implementación y medición de desempeño de la topología de comunicación *Scatternet Bluetooth*. Adicionalmente, tanto la especificación del grupo especial de investigación SIG (special interest group) *Bluetooth* creadores de la tecnología y el estándar de la IEEE 802.15.1 no definen un mecanismo para la creación de esta topología, lo que motivó el desarrollo de esta investigación.

En esta investigación se presenta una metodología para la implementación de la estructura de comunicación *Scatternet*, desarrollado sobre una plataforma Linux en particular la distribución Ubuntu 8.04 – Hardy Heron, adicionalmente se usaron los siguientes recursos: el perfil PAN (Personal Area Network), las bibliotecas y herramientas brindadas por Bluez (pila oficial de protocolos *Bluetooth* para Linux).

Se diseñaron un conjunto de escenarios con diversas topologías con el objeto de evaluar el desempeño en las comunicaciones con la tecnología *Bluetooth* e identificar aquellos factores que lo degradan

El conjunto de medidas obtenidas a partir de las pruebas realizadas, mostraron un bajo desempeño en topologías con saltos o nodos intermedios y en topologías pobladas con tráfico.

Palabras Claves: estándar 802.15.1, WPAN, SIG *Bluetooth*, *Piconet*, *Scatternet*, perfil PAN, Bluez.

ÍNDICE DE CONTENIDO

Índice de contenido

ÍNDICE DE FIGURAS

Índice de figuras

Índice de figuras

ÍNDICE DE TABLAS

1. INTRODUCCIÓN

La presente investigación tiene como objetivo la búsqueda de un método que permita la creación de la estructura de comunicación básica *Piconet*, dando lugar a redes ad hoc más extensas denominadas *Scatternet*, esta última muy poco estudiada e implementada, razón que motivó el estudio del desempeño de las comunicaciones en redes con tecnología *Bluetooth*.

La evolución de los dispositivos electrónicos, orientados a la movilidad, ha llevado a que los nuevos dispositivos tengan implementada tecnología de interconexión inalámbrica. Actualmente una de la más popular es la tecnología *Bluetooth* por su fiabilidad, bajo consumo de energía, mínimo costo en un pequeño microchip capaz de ser adaptado a dispositivos de tamaño reducido. Esto despertó el interés de la mayoría de las empresas de informática y de telecomunicaciones las cuales unieron estos dos mundos para desarrollar una interfaz abierta y de bajo costo, facilitando la interconexión y comunicación inalámbrica entre dispositivos.

Interconectar dispositivos electrónicos con tecnología inalámbrica, tales como: teclados, mouse, impresoras, teléfonos móviles, electrodomésticos, asistente personal digital (PDA) o un computador, ya sea en nuestro hogar o trabajo, es el uso más común de esta tecnología. La tecnología *Bluetooth* demuestra su versatilidad al permitir la creación de estructuras de redes de área personal conocidas como las WPAN (Wireless Personal Area Network).

Para realizar esta investigación, se estudiaron los aspectos más importantes de la tecnología *Bluetooth*, principalmente todo el proceso de conexión y comunicación, así como las funciones de cada capa de la pila de protocolo de *Bluetooth*. Esta investigación permitió definir el método para la implementacion de una *Scatternet Bluetooth* para la interconexión de varias *Piconets* que soporten la comunicación *extremo a extremo.*

Se aplicaron técnicas de *benchmarking [10]*, que revelaron resultados sobre el desempeño, de las redes con tecnología *Bluetooth*, útiles para identificar factores que disminuyen el desempeño de las comunicaciones. Asimismo ofrecer una metodología como aporte en este tema, para futuros trabajos académicos relacionados con la tecnología *Bluetooth*.

1.1 Planteamiento del problema

En la mayoría de los casos, las redes *Bluetooth* se establecen formando una *Piconet* con unos pocos dispositivos. Sin embargo, la especificación establece un mecanismo para extender el alcance de una red *Bluetooth* a través de una *Scatternet*. Hay pocos trabajos acerca de la topología *Scatternet* y menos aun de su desempeño. Tampoco es clara la forma cómo se puede formar una *Scatternet*

para soportar la comunicación *extremo a extremo* entre dispositivos que atraviesan dos o más *Piconets* conectadas en una *Scatternet*.

Por lo tanto, en este trabajo se pretende responder la siguiente incógnita:

¿Cómo se puede formar una Scatternet Bluetooth para la interconexión de varias Piconets que soporten la comunicación extremo a extremo, y cuál es el impacto en el desempeño de su comunicación?

1.2 Objetivo general

Establecer una metodología de diseño, para la implementación de una *Piconet* y *Scatternet Bluetooth* y estudiar el desempeño *extremo a extremo* de la red.

1.2.1 Objetivos específicos

- Definir los aspectos a considerar en el diseño o creación de una *Piconet y Scatternet Bluetooth*.
- Definir las métricas y los escenarios, que permitirán evaluar el comportamiento del desempeño de las comunicaciones en una *Piconet y Scatternet*.
- Determinar con base a los resultados obtenidos, las características y comportamiento que afecta el desempeño de las comunicaciones en las *Piconets y Scatternets Bluetooth*.

1.3 Justificación

- Pocos estudios sobre la implementación de la topología *Scatternet Bluetooth*.
- Aun cuando en la especificación se define el concepto de *Scatternet*, los mecanismos que permitan su establecimiento o creación no se encuentran bien definidos.
- Determinar las ventajas y alcances de la tecnología *Bluetooth* en ambientes de redes ad-hoc.
- Conocer que aspectos determinan el buen desempeño de ambientes *Bluetooth* que interconectan varias Piconets en una Scatternet.

1.4 Alcance

Los escenarios y pruebas se limitarán por la cantidad de *dongles* que se poseen. Para esta investigación se contó con un total de 12 *dongles Bluetooth*, fijandose un tope de 11 *Piconets* para ser creadas de forma simultánea en la realización de pruebas de comunicación *extremo a extremo*, además se pudieron realizar distintas variantes de topologías de *Piconets* en base a estas cantidades de *dongles* que se explicarán a lo largo de este documento. Para la interconexión

de los nodos, se usó el perfil PAN, el cual fue considerado como el único medio que permita la creación de la *Scatternet*.

1.5 Distribución del documento

Este documento consta de los siguientes siete capítulos:

- **Capítulo I:** contiene un breve resumen del documento, se explica el problema planteado, se verán los objetivos, justificación y alcance de esta investigación y finalmente la distribución del documento.

- **Capítulo II:** describe el marco teórico, en el cual se hará referencia específicamente de la tecnología *Bluetooth*, se explicarán los conceptos más importantes, sus características, funcionamiento, los aspectos más importantes de la pila de protocolos de la tecnología *Bluetooth*, la función de cada capa; la cual sirvió para tener una base que permita analizar los resultados obtenidos y sacar conclusiones respectivas que se puedan ofrecer en la investigación.

- **Capítulo III:** contiene una descripción detallada de los escenarios definidos con los cuales se pretende dar respuestas al problema planteado, a través de una serie de pruebas, siguiendo técnicas formales que permitan medir el desempeño en dichos escenarios. También contiene la descripción de los elementos tanto de hardware como de software usados y se explica la manera como se utilizaron y configuraron para realizar exitosamente las pruebas.

- **Capítulo IV:** muestra a través de gráficos y tablas los valores promedios de todas las pruebas realizadas sobre los escenarios planteados en el Capítulo III y se da una breve explicación de las posibles causas o factores que producen las variaciones entre los escenarios.

- **Capítulo V:** presenta las conclusiones, contribuciones, limitaciones de esta investigación y las recomendaciones para futuros trabajos.

- **Capítulo VI:** muestra las referencias bibliográficas usadas en esta investigación.

- **Capítulo VII:** muestra la sección de anexos que contienen los gráficos que complementan los resultados obtenidos en el Capítulo IV.

2. MARCO TEÓRICO

En este capítulo se explican las definiciones y conceptos básicos necesarios para entender la tecnología *Bluetooth*, así como sus características y estructura, para obtener una base teórica que permita comprender la investigación.

2.1 Historia

En 1994, Ericsson Mobile Communications, la compañía global de telecomunicaciones con base en Suecia, comenzó un estudio de viabilidad de una interfaz de radio de baja potencia y bajo costo entre teléfonos móviles y otros accesorios, con el objetivo de eliminar los cables. El estudio era parte de un proyecto más amplio que investigaba cómo conectar diferentes dispositivos de comunicaciones a la red celular a través de un teléfono móvil.

El trabajo de Ericsson en esta área atrajo la atención de IBM, Intel, Nokia y Toshiba. Estas compañías junto a Ericsson decidieron formar en febrero de 1998 un grupo especial de investigación denominado SIG (Special Interest Group) *Bluetooth*[1], con el objetivo de desarrollar, promover, definir y publicar las especificaciones de esta tecnología inalámbrica de corta distancia [1]. La idea era lograr un conjunto adecuado de áreas de negocio; dos líderes del mercado de las telecomunicaciones, dos líderes del mercado de computadoras portátiles y un líder de la fabricación de chips**.** El propósito de esta elección fue el de incidir fuertemente en el mercado de las comunicaciones inalámbricas, estableciendo un estándar de comunicación por radio que llamaron *Bluetooth*.

El primer objetivo para los productos *Bluetooth* de primera generación se basó en entornos de gente de negocios que viaja frecuentemente, por lo que se debería pensar en integrar el chip de radio *Bluetooth* en equipos usados por este grupo de personas, como: PCs portátiles, teléfonos móviles, PDAs y auriculares. Esto originaba una serie de cuestiones previas que deberían solucionarse, tales como:

- El sistema debería operar en todo el mundo.
- El emisor de radio deberá consumir poca energía, ya que debe integrarse en equipos portátiles alimentados por baterías.
- La conexión deberá soportar voz y datos, y por lo tanto aplicaciones multimedia.

[1] http://www.*Bluetooth*.com/*Bluetooth*/SIG

En mayo del mismo año, se invitó a otras compañías a participar en el grupo: Microsoft, Lucent Technologies, 3COM y Motorola. Se han publicado sucesivas versiones en las cuales se han logrado solucionar detalles, agregar mejoras y se han unido una gran cantidad de miembros del SIG [2].

2.1.1 Etimología

El nombre *Bluetooth* procede del rey danés del siglo X llamado Harald Blatand (traducido como Harold *Bluetooth*), conocido por sus habilidades comunicativas y por unificar las tribus en guerra de Noruega, Suecia y Dinamarca e iniciar el proceso de cristianización de la sociedad vikinga [1].

El logo de *Bluetooth* como se observa en la Figura 2.1, combina la representación de las runas nórdicas Hagalaz (transcrito por 'H') y Berkana (transcrito por 'B') en un mismo símbolo [1].

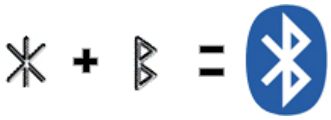


Figura 2.1 Combinaciones para la creación del logo de *Bluetooth* [1].

2.2 Redes inalámbricas de área personal - WPAN

Las redes inalámbricas de área personal WPAN (Wireless Personal Area Network) son redes con sistemas de comunicaciones de corto alcance, que por lo general suele ser de 10 metros a la redonda, usadas generalmente para conectar dispositivos periféricos (impresoras, teléfonos móviles, etc) o un asistente personal digital (PDA) a un computador sin conexión por cables. Esta comunicación de dispositivos *peer-to-peer* normalmente no requiere de altos índices de transmisión de datos, por esta razón y por su corto alcance tiene como resultado un bajo consumo de energía haciendo a la tecnología WPAN ideal para el uso con dispositivos móviles pequeños, que funcionan con baterías [2].

2.2.1 Grupos de trabajo

Para satisfacer las diferentes necesidades de comunicación dentro de un área personal, la IEEE ha dividido sus esfuerzos en cuatro grupos de trabajo para la tecnología WPAN.

- El grupo de trabajo **802.15.1** ha realizado un estándar basado en las especificaciones de la fundación *Bluetooth*. Este grupo de trabajo publicó el estándar IEEE 802.15.1 el 14 junio de 2002 [3].
- El grupo de trabajo **802.15.2** estudia los posibles problemas derivados de la coexistencia de WPAN's con otros dispositivos inalámbricos que utilicen las

bandas de frecuencia no reguladas, tales como redes inalámbricas de área local (WLAN)..

- El grupo de trabajo **802.15.3** trabaja para establecer el estatus de alta velocidad (20 Mbps) para WPAN, consumir poca energía y ofrecer soluciones a bajos costos así como aplicaciones multimedia.
- El grupo de trabajo T4 para el desarrollo IEEE **802.15.4** investiga y desarrolla soluciones que requieren una baja transmisión de datos y con ello una duración en las baterías de meses e incluso de años.

2.3 Definición de Bluetooth

El término *Bluetooth* se refiere a una especificación abierta para la tecnología de interconexión inalámbrica de corto alcance para transmitir voz y datos en cualquier parte del mundo. El SIG ha creado una especificación para la comunicación inalámbrica *Bluetooth* que esté disponible al público y sin el pago de derechos. Que sea una especificación abierta, ha sido el objetivo fundamental del SIG desde su formación para ayudar a fomentar la aceptación generalizada de la tecnología.

Los ambientes de cómputo y comunicaciones están cada vez más interrelacionados. La voz es ahora comúnmente transmitida y almacenada en formatos digitales. Dispositivos como los teléfonos móviles también se utilizan para aplicaciones de datos, como acceso a información o la navegación web. Las computadoras pueden ser controladas por la voz a través del reconocimiento de voz. Algunas tecnologías de comunicación inalámbrica están diseñadas para transportar sólo voz, mientras que otras sólo tráfico de datos. La comunicación inalámbrica *Bluetooth* permite ambas comunicaciones, tanto de voz como datos, por lo tanto esta es una tecnología ideal para unir estos dos mundos.

Otro punto importante es que la industria de telecomunicaciones es altamente regulada en muchas partes del mundo. La utilización del espectro de radio frecuencia requiere de una licencia estricta para poder transmitir por la misma. Sin embargo, existen algunos rangos de frecuencia disponibles que pueden ser usados sin licencia, y *Bluetooth* eligió operar dentro de un espectro de frecuencias libre de licencia en todo el mundo, la llamada ISM “Industrial, Scientific and Medical” (con ciertas limitaciones y restricciones en pocos países como Japón, Francia y España), haciendo de esta forma que cualquier dispositivo que emplee la comunicación inalámbrica *Bluetooth* pueda funcionar sin necesidad de modificaciones sin importar donde esté.

La tecnología inalámbrica de corto alcance *Bluetooth* es ideal para sustituir los numerosos cables que están asociados a los dispositivos de hoy en día. La especificación de *Bluetooth* define un medio de transporte inalámbrico para remplazar cables seriales, como los usados con un módem, cámaras digitales, asistentes personales y periféricos de un computador [4].

2.4 Topología de la comunicación

Una gran ventaja, en la que se demuestra la versatilidad del diseño de la tecnología *Bluetooth*, está en la fácil creación de redes entre distintos dispositivos de ésta misma tecnología.

Bluetooth ha sido diseñada para operar en un ambiente multi-usuario. Este presenta dos tipos de configuraciones posibles, las cuales se pueden expandir a un número considerable de elementos para conformar así las redes y sub-redes. La estructura que maneja esta tecnología está compuesta, en su forma más básica denominada *Piconet* y en una estructura un poco más compleja a la que se denomina *Scatternet*.

2.4.1 Piconet

Una *Piconet* se compone de dos a ocho dispositivos dentro del radio de cobertura común; éstos pueden establecer conexión entre ellos, ocupando el mismo canal físico y sincronizado con un mismo reloj y secuencia de salto. El reloj que se utiliza en una *Piconet* es idéntico al reloj *Bluetooth* de uno de los dispositivos que la conforman, al que se le llamará dispositivo *maestro*. En cuanto a la secuencia de salto, esta también se deriva del reloj y de la dirección del dispositivo *maestro*. Se admite un solo *maestro* y hasta 7 dispositivos sincronizados simultáneamente llamados *esclavos* como se puede observar en la Figura 2.2, aunque pueden permanecer hasta 256 *esclavos* vinculados al *maestro* en un estado especial de relativa inactividad denominado *parked*. Los términos *maestro* y *esclavo* se utilizan para describir las funciones dentro de una *Piconet*. Los participantes pueden intercambiar los papeles entre ellos, si un dispositivo esclavo quiere asumir el papel de maestro; sin embargo, sólo puede haber un *maestro* en la *Piconet* en un instante determinado.

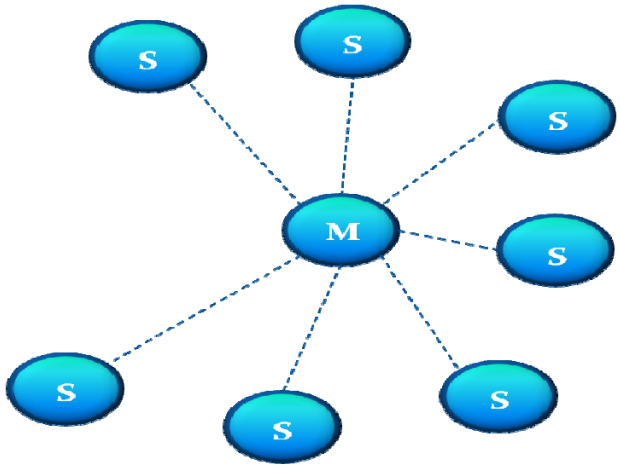


Figura 2.2 Ejemplo de Piconet

Durante el proceso de establecimiento de la conexión en una *Piconet*, el dispositivo *maestro* genera una tabla pseudoaleatoria con la secuencia de la frecuencia que deben utilizar los dispositivos pertenecientes a la *Piconet* durante las comunicaciones. El intercambio de la tabla de saltos desde el *maestro* hacia el (los) *esclavo* (*esclavos*) se realiza en un canal determinado del espectro de frecuencias, de forma que todos los dispositivos pueden acceder a éste.

Cada dispositivo de la *Piconet* utiliza la identidad y reloj de *maestro*, para seguir el canal de salto. Cuando se establece la conexión, el dispositivo *esclavo* recibe un paquete FHS (Frecuency Hop Synchronization) que le permite sincronizar su reloj interno con el reloj del *maestro* agregando un desplazamiento a su reloj interno. El reloj nativo mantiene siempre constante su frecuencia; sin embargo, los ajustes sólo son válidos mientras dura la conexión.

Los dispositivos maestros controlan en tráfico del canal. Éstas tienen la capacidad para reservar *slots* en los enlaces síncronos SCO (Synchronous Connection-Oriented). Para los enlaces asíncronos ACL (Asynchronous Connection-less), utilizan un esquema de sondeo.

En la misma ubicación puede haber varias *Piconets* distintas. Cada una tendrá un canal físico diferente, es decir, un dispositivo *maestro*, un reloj y una secuencia de salto independientes.

Un dispositivo *Bluetooth* puede utilizarse simultáneamente en dos o más *Piconets* mediante el uso de TDM (Time Division Multiplexing). Ahora bien, este dispositivo *Bluetooth* no actuará nunca como *maestro* en más de una *Piconet*. Esto se debe a que la *Piconet* está determinada por la sincronización del reloj *Bluetooth* del dispositivo *maestro*. En cambio, este dispositivo *Bluetooth* sí podrá hacer las veces de *esclavo* en diversas *Piconets* [2][5].

2.4.2 Scatternet

Cuando un dispositivo *Bluetooth* participa en dos o más *Piconets* forma parte de lo que se conoce como una *Scatternet*. Esto no implica, necesariamente, que el dispositivo tenga funciones de direccionamiento de redes. Los protocolos básicos de la tecnología *Bluetooth* no se han desarrollado para ofrecer tales funciones, ya que éstas dependen de protocolos de capas superiores los cuales no se incluyen en la especificación principal de *Bluetooth* [5].

Los equipos que comparten un mismo canal sólo pueden utilizar una parte de la capacidad de este. Aunque los canales tienen un ancho de banda de un 1MHz, cuantos más usuarios se incorporan a la *Piconet* disminuye la capacidad.

Aunque la especificación *Bluetooth* contempla el concepto de *Scatternet*, este no especifica un determinado protocolo de creación de dicha estructura. Conseguir una estructura de *Scatternet* óptima es objeto de investigaciones.

Diferentes propuestas planteadas intentan obtener una topología de *Scatternet* similar a la que se muestra en la Figura 2.3, en la cual para conectar dos *Piconets* se deben compartir uno o varios dispositivos puente. Los dispositivos puente pueden actuar como *maestro* en una *Piconet* y *esclavo* en otra o como *esclavo* en ambas, pero nunca como *maestro* en las dos *Piconets*, ya que cada *Piconet* está identificada unívocamente con el reloj y dirección física del dispositivo *maestro*, como ya se ha mencionado [4].

No se ha definido ninguna restricción en cuanto al número de roles que puede asumir un dispositivo, pero sí especifica que un dispositivo, en un determinado instante de tiempo, sólo puede estar activo, y por tanto transmitir y recibir información, en una de las *Piconets* a las que pertenece [6].

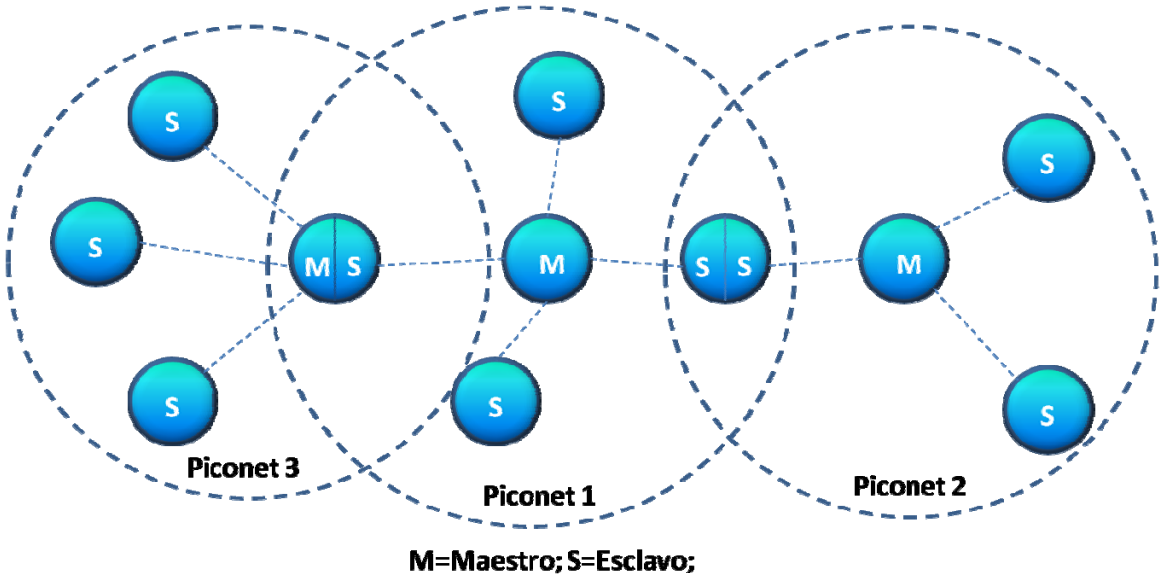


Figura 2.3 Ejemplo de una *Scatternet*

La mayoría de estos estudios no se han centrado en problemas de implementación. Así, para poder conectar las *Piconets* 1 y 3 de la Figura 2.3, el dispositivo puente *maestro*/*esclavo* (M/S) debe pasar a un estado de conexión no activo conocido como modo *Hold* en la *Piconet* 3, y pasar a modo activo con respecto a la *Piconet* 1. Esto implica que las comunicaciones en la *Piconet* 3 serán suspendidas hasta que expire el tiempo de la estación en modo *Hold*.

Por otra parte, para conectar las *Piconet* 1 y 2, el dispositivo puente *esclavo*/*esclavo* (S/S) pasará a modo *Hold* en la *Piconet* 2 y a modo activo en la *Piconet* 1. Durante el tiempo en modo *Hold* el dispositivo *maestro* de la *Piconet* 2 no enviará paquetes *POLL,* los cuales son destinados a asignar ranuras de acceso al canal, al dispositivo puente. Un dispositivo puente activo en una *Piconet* almacena paquetes de datos dirigidos a dispositivos de la *Piconet* adyacente, entregándolos posteriormente a las estaciones destino cuando el tiempo en modo

Hold termina. Así, todos los mensajes entre *Piconets* son enviados a través de los dispositivos puente [4].

2.4.3 Comunicación entre Piconets

En un conjunto de varias *Piconets*, éstas seleccionan diferentes saltos de frecuencia y están controladas por diferentes dispositivos principales, por lo que si un mismo canal de salto es compartido temporalmente por *Piconets* independientes, los paquetes de datos podrán ser distinguidos por el código de acceso que les precede, que es único en cada *Piconet*.

La sincronización de varias *Piconets* no está permitida en la banda ISM. Sin embargo, las dispositivos pueden participar en diferentes *Piconets* en base a un sistema TDM. Un dispositivo al incorporarse a una nueva *Piconet* debe realizar un ajuste interno de su reloj, para minimizar la diferencia con el reloj del *maestro*, por lo que gracias a este sistema se puede participar en varias *Piconets* realizando cada vez los ajustes correspondientes una vez conocidos los diferentes parámetros de la *Piconet*. Cuando un dispositivo abandona una *Piconet*, informa al *maestro* que ésta no estará disponible por un determinado período, que será en el que estará activa en otra *Piconet*. Durante su ausencia, el tráfico en la *Piconet* entre el *maestro* y otros *esclavos* continúa.

De la misma manera que un *esclavo* puede cambiar de una *Piconet* a otra, un *maestro* también lo puede hacer, con la diferencia de que el tráfico de la *Piconet* se suspende hasta la vuelta del dispositivo *maestro*. El *maestro* que entra en una nueva *Piconet* lo hace como *esclavo* [4].

2.5 Protocolo de conexión

En la Figura 2.4 se muestra el diagrama que ilustra los diferentes estados usados en un controlador de enlace *Bluetooth*. Los modos *STANDBY* y *CONNECTION* son los dos estados principales; además, existen los siguientes siete subestados, *Page*, *Page Scan*, *Inquiry*, *Inquiry Scan*, *Master Response*, *Slave Response* e *Inquiry* Response. Los subestados son los estados intermedios que se usan para añadir nuevos *esclavos* a una *Piconet*. Para moverse de un estado a otro, se utilizan ya sea comandos del administrador de enlace *Bluetooth*, o señales internas en el control de enlace [6].

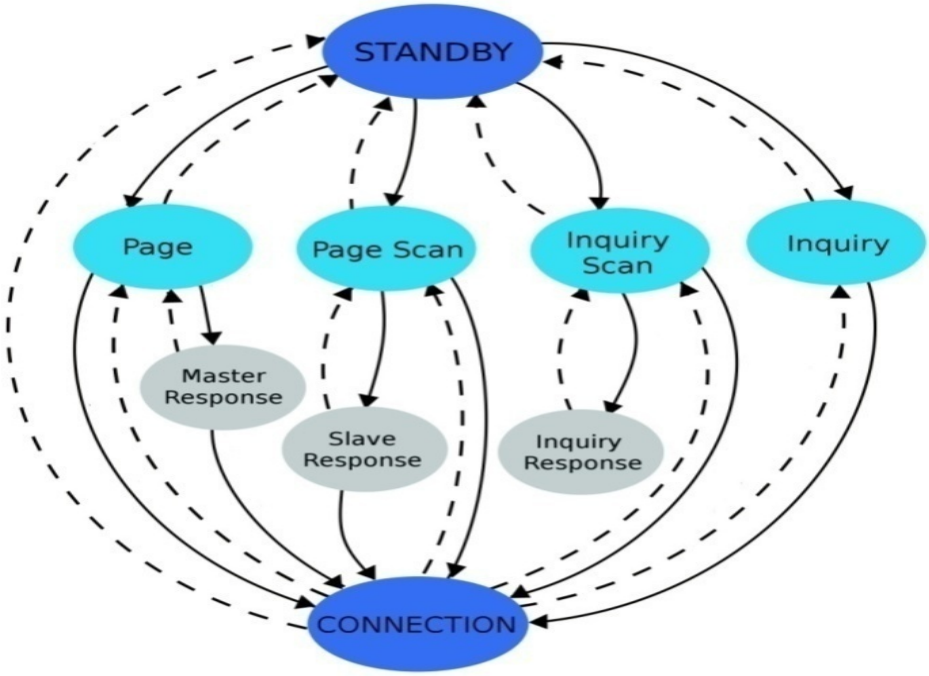


Figura 2.4 Diagrama de estados de conexión *Bluetooth* [6].

2.5.1 Standby

El estado de *STANDBY* es el estado que tienen por defecto las dispositivos *Bluetooth*. En dicho estado, el dispositivo *Bluetooth* está en un estado de bajo consumo de energía. Sólo su propio reloj se encuentra corriendo hasta que ocurra una señal de cambio.

El controlador puede dejar el estado de *STANDBY* para buscar mensajes de *Page* o *Inquiry*, o para el mismo hacer *Page* o *Inquiry*. Cuando responde a un mensaje de *Page*, el dispositivo no regresa al estado de *STANDBY* sino que entra al estado de *CONNECTION* como *esclavo*. Cuando transmite exitosamente un *Page*, el dispositivo entrará al estado de *CONNECTION* como dispositivo maestro [6].

2.5.2 Procedimientos de acceso

Para establecer nuevas conexiones se usan los procedimientos de *Inquiry* y paging. El procedimiento de *Inquiry* permite al dispositivo *maestro* descubrir otros dispositivos que se encuentran en su rango de cobertura, obtener conocimiento de cuáles son sus direcciones y su reloj. Con el procedimiento de paging es como realmente puede ser establecida la conexión. Sólo se requiere la dirección del dispositivo *Bluetooth* para establecer una conexión. El conocimiento del reloj

acelerará el procedimiento. El dispositivo que establezca una conexión realizará un procedimiento de *Page* y automáticamente será el *maestro* de la conexión [6].

- ***Page Scan*:** en el subestado *Page Scan*, un dispositivo escucha en búsqueda de su propio código de acceso de dispositivo o DAC (device access code) el tiempo que dure un periodo de búsqueda. Durante ese periodo, escucha en un solo salto de frecuencia. El periodo de búsqueda debe ser lo suficientemente largo para completar 16 frecuencias de *Page*.

  Cuando un dispositivo entra en el subestado de *Page Scan*, esta selecciona la frecuencia de *scan* de acuerdo a la secuencia de saltos de búsqueda correspondiente a este dispositivo. La secuencia de saltos de búsqueda es determinada por la dirección del dispositivo *Bluetooth* (BD_ADDR) y por su reloj nativo.

  - ***Page*:** el subestado *Page* es usado por el *maestro* (fuente) para activar y conectar a un *esclavo* (destino), el cual periódicamente despierta en el subestado *Page Scan*. El *maestro* trata de capturar al *esclavo* por medio de repetidas transmisiones de DAC del *esclavo* en diferentes canales. Ya que los relojes del *maestro* y el *esclavo* no están sincronizados, el *maestro* no conoce exactamente cuando el *esclavo* despertará y en qué frecuencia.

  - ***Page* response:** Cuando un mensaje de *Page* es recibido exitosamente por un *esclavo*, se dice que existe una sincronización FH (Frequency Hops), entre el *maestro* y el *esclavo*. Tanto el *maestro* como el *esclavo* entran en una rutina de respuestas para intercambiar información vital para continuar la configuración de la conexión.

    Los mensajes iníciales entre el *maestro* y el *esclavo* se muestran en la Tabla 2.1. En el paso 1, el dispositivo *maestro* está en el subestado *Page* y el dispositivo *esclavo* está en el subestado *Page Scan*. En el paso 2, una vez reconocido su código de acceso, el *esclavo* pasa al subestado *Slave Response*. El *maestro* espera una respuesta del *esclavo* y cuando esta llega, este entrará en el estado *Master Response* en el paso 3. Se debe notar que durante el mensaje inicial de intercambio, todos los parámetros son derivados del BD_ADDR (*Bluetooth* device address) del *esclavo*, y que la secuencia de saltos usados en el *Page* y *Page* response provienen también de un derivado de la BD_ADDR del *esclavo*.

Paso	Mensaje	Dirección	Secuencia de salto	Código de acceso y reloj
1	ID *esclavo*	*maestro* a *esclavo*	*Page*	*esclavo*
2	ID *esclavo*	*esclavo* a *maestro*	*Page* response	*esclavo*
3	FHS	*maestro* a *esclavo*	*Page*	*esclavo*
4	ID *esclavo*	*esclavo* a *maestro*	*Page* response	*esclavo*
5	1er paquete *maestro*	*maestro* a *esclavo*	canal	*maestro*
6	1er paquete *esclavo*	*esclavo* a *maestro*	canal	*maestro*

Tabla 2.1 Mensajes iniciales durante la activación

- ***Master Response*:** Cuando el *maestro* ha recibido un mensaje de respuesta del *esclavo* en el paso 2, este entrará en la rutina de *Master Response*. Asimismo hace una captura de su reloj para el esquema de selección de salto de búsqueda. Entonces el *maestro* transmitirá un paquete FHS (Frequency Hop Synchronization) en el paso 3, que contiene el tiempo real del reloj *Bluetooth* del *maestro*, los 48 bits de la dirección BD_ADDR del *maestro*, los bits de paridad, y una clase de dispositivo.

  Después de que el *maestro* ha enviado su paquete FHS, éste espera por una segunda respuesta del *esclavo* en el paso 4, si la respuesta no es recibida, el *maestro* retransmitirá el paquete FHS, hasta que una segunda respuesta de *esclavo* sea recibida, o se consuma el tiempo de 8 ranuras. En el caso anterior el *maestro* vuelve a subestado de *Page* y envía un mensaje de error al manejador de enlace.

  Si la respuesta del *esclavo* es en efecto recibida, el *maestro* entra en el estado de *CONNECTION* en el paso 5.

- ***Slave Response*:** después de haber recibido su propio DAC en el paso 1, el dispositivo *esclavo* transmite un mensaje de respuesta en el paso 2. Este mensaje de respuesta sólo consiste en el DAC del *esclavo*. Después de haber enviado el mensaje de respuesta, espera de la llegada de un paquete FHS.

  Si la configuración falla antes que el estado de *CONNECTION* sea alcanzado, el *esclavo* continuará escuchando en espera del paquete FHS hasta que sea excedido el tiempo de 8 ranuras. Si un paquete FHS es recibido por el *esclavo* en el subestado slave reponse, el *esclavo* regresará una respuesta (sólo el código de acceso del dispositivo *esclavo*) en el paso 4, para reconocer la recepción del paquete FHS. Entonces el *esclavo* obtiene el código de acceso al canal (*maestro*) y el reloj, los cuales provienen del paquete FHS. Finalmente, el *esclavo* entra en el estado *CONNECTION* en el paso 5.

2.5.3 Procedimiento de Inquiry

En el sistema *Bluetooth*, los procedimientos *Inquiry* son usados cuando la dirección del dispositivo destino no es conocida por la fuente. Durante un subestado de *Inquiry*, el dispositivo de descubrimiento recoge las direcciones y relojes de todos los dispositivos *Bluetooth* que respondan al mensaje de *Inquiry*. Esta puede entonces, si se desea, realizar una conexión con cualquiera de ellos a través de de los procedimientos de *Page* descritos anteriormente.

- ***Inquiry*:** el subestado *Inquiry* es usado por el dispositivo que quiere descubrir nuevos dispositivos. Este subestado es muy similar al subestado *Page*. El subestado *Inquiry* es continuo y dura hasta que el manejador de enlace del dispositivo *Bluetooth* lo decida, o hasta que se supere el tiempo límite del *Inquiry*.

- ***Inquiry Scan*:** el subestado *Inquiry Scan* es muy similar al subestado *Page Scan*. Sin embargo, en vez de escanear un código de acceso de dispositivo, el receptor espera de un código de acceso *Inquiry* adecuado para completar 16 frecuencias de *Inquiry*. El *scan* se realiza en un solo salto de frecuencia.

    - ***Inquiry* Response:** para la operación *Inquiry* sólo el *esclavo* responde. El *maestro* escucha entre mensajes *Inquiry* en espera de una respuesta, pero después de leer una respuesta, este continúa transmitiendo mensajes *Inquiry*.

2.5.4 Estados de conexión

Para cada estado, el estado de conexión se inicia con un paquete de *POLL*, enviado por el *maestro* para verificar que el *esclavo* se ha cambiado al canal del *maestro*.

Una vez que un *esclavo* está en el estado de conexión, puede estar en cualquiera de los siguientes modos de operación.

- **Modo activo:** el dispositivo *Bluetooth* participa activamente en el canal de la *Piconet*. El *maestro* y el *esclavo* transmiten en *slots* alternativos. El *maestro* planifica la transmisión basada en las demandas de tráfico hacia y desde los diferentes *esclavos*. Además, soporta transmisiones regulares para mantener a los *esclavos* sincronizados al canal y se realizan varias optimizaciones para ahorrar potencia. Los *esclavos* activos, que tienen el AM_ADDR (Active Member ADDRess), escuchan en los *slots* de tiempo *maestro* a *esclavo*. Si un *esclavo* activo no es direccionado, podría *dormir* hasta la próxima transmisión del *maestro*. En la Figura 2.5 se muestra el intercambio de paquetes, tanto SCO como ACL entre el *maestro* y los *esclavos* activos en la *Piconet*.

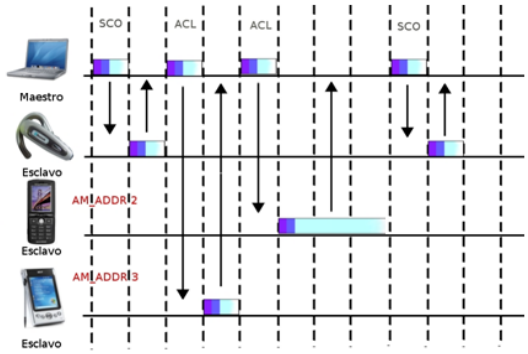


Figura 2.5 Ejemplo de modo activo

- **Modos de ahorro de energía:** los dispositivos miembros de una *Piconet* pueden entrar en uno de los tres modos de ahorro de energía (*Hold, Sniff y Park*), en los cuales su nivel de actividad es menor, para lograr efectivamente el uso óptimo de la energía o para cumplir con otros fines, como se verá a continuación.

  - **Modo Hold:** en este modo sólo está funcionando un contador interno. El dispositivo *esclavo* pueden solicitar ser puestas en modo *Hold*, para tener la posibilidad de entrar a otros subestados como: *Scan*, *Page*, *Inquiry* o atender a otra *Piconet*. La transferencia de datos vuelve a comenzar de forma instantánea cuando los dispositivos abandonan el modo *Hold*, instante en el que se ponen de acuerdo *maestro* y *esclavo*. Mientras dura este modo, el dispositivo *esclavo* guarda su AM_ADDR. Tiene un ciclo de trabajo intermedio entre los tres modos de ahorro de energía. En este modo, los paquetes ACL no son soportados, pero sí lo son los paquetes SCO. Por ejemplo, un teléfono *Bluetooth* en modo *Hold*, podría soportar enlaces de voz, pero no de mensajes de texto. En la Figura 2.6 se puede observar que el dispositivo *esclavo* teléfono, se coloca en modo *Hold*, y como resultado los paquetes ACL no serán recibidos por el *esclavo* en este modo.

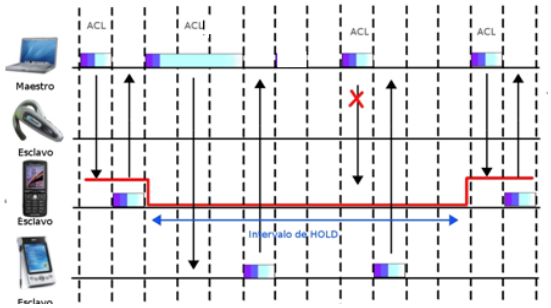


Figura 2.6 Ejemplo de modo Hold.

- **Modo Sniff:** sólo es aplicable para los dispositivos *esclavos*, estos escuchan en canal de la *Piconet* a una tasa de tiempo reducida, lo que reduce su ciclo de trabajo. El intervalo *Sniff* es programable y depende de la aplicación. Tiene el mayor ciclo de vida de los tres modos de ahorro de energía. Los *esclavos* entran en este modo cuando se lo ordena el *maestro* o bien si el *esclavo* así lo requiere, por ejemplo para participar en otra *Piconet*. En la Figura 2.7 se muestra como el *esclavo* PDA entra a modo Sniff y se coloca en estado activo de forma intermitente en periodos breves.

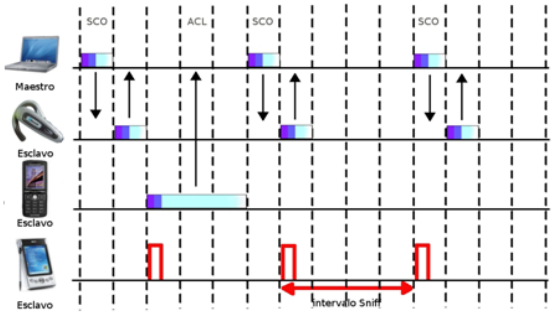


Figura 2.7 Ejemplo de modo Sniff.

- **Modo Park:** el dispositivo se encuentra aún sincronizado a la *Piconet* pero no participa en el tráfico. Este modo se usa para conectar más de siete *esclavos* a un único *maestro*. Tiene el ciclo de trabajo más corto de los tres modos de ahorro de energía, por lo que es un modo de muy baja potencia, ya que el *esclavo* tiene muy poca actividad. Como se puede observar en la Figura 2.8, el *esclavo* 3 ya no posee la AM_ADDR, y en vez de esto recibe el PM_ADDR (dirección de 8-bit que usa el *maestro* para hacer *unpark* a un *esclavo*) y el AR_ADDR (dirección de 8-bit que utiliza el *esclavo* para consultar al *maestro* cómo hacer *unpark* y salir así de este modo).

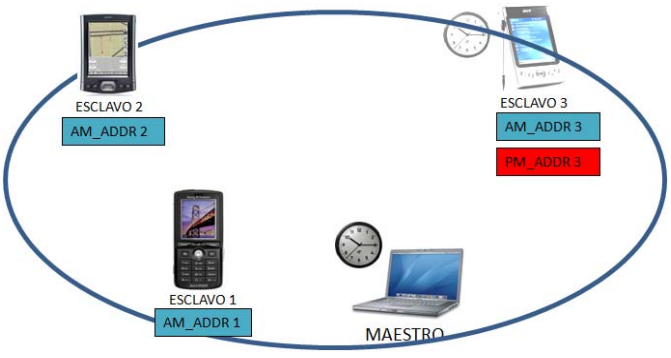


Figura 2.8 Ejemplo de modo Park.

2.5.5 Direcciones Bluetooth

Antes o durante las conexiones, los dispositivos *Bluetooth* manejan diferentes tipos de direcciones tales como:

- **BD_ADDR**: Llamado "*Bluetooth* Device Adddress", es una dirección única de *Bluetooth* para cada dispositivo, de 48 bits por dispositivo *Bluetooth*, derivada del estándar IEEE 802.
- **AM_ADDR:** Llamado "Active Member Address", es una dirección de 3 bits para cada miembro *esclavo* que forma parte de la *Piconet*. Un *esclavo* sólo acepta los paquetes que recibe si contienen su AM_ADDR o si son paquetes de broadcast (AM_ADDR=0).
- **PM_ADDR:** Llamado "Parked Member Address", un *esclavo* que se encuentre en estado *park* puede ser identificado por su dirección BD_ADDR o por la dirección PM_ADDR de 8 bits. La dirección PM_ADDR=0 se reserva para dispositivos en modo *park* que se identifican por su BD_ADDR.
- **AR_ADDR:** Llamado "Access Request Address", cuando un *esclavo* pasa a modo *park* se le asigna una dirección AR_ADDR. Esta dirección permite al *esclavo* en modo *park* determinar qué ranura puede usar para pedir acceso al *maestro*. La dirección AR_ADDR no es necesariamente única, diferentes *esclavos* en modo *park* pueden tener la misma AR_ADDR [3].

2.6 Pila de protocolos Bluetooth

En la mayoría de los casos, los diferentes protocolos utilizados en una determinada tecnología presentan una jerarquía predefinida. La capa base de la jerarquía normalmente contiene protocolos que se utilizan en todas las aplicaciones de la tecnología. Otras capas se apilan en la parte superior de esta capa base, conteniendo los protocolos que definen las funciones verticalmente, a esto se le llama pila de protocolos [5].

En la especificación de *Bluetooth*, hay una pila de protocolos global que incluye todos los posibles protocolos que pueden ser usados por las aplicaciones; cada aplicación individual también posee su propia pila de protocolos, conteniendo sólo aquellos protocolos que son usados por esta aplicación. En la terminología *Bluetooth*, se requiere una pila de protocolos por cada perfil [5].

La pila de protocolos de *Bluetooth* se muestra completa en la Figura 2.9. Naturalmente no todas las aplicaciones hacen uso de todos los protocolos de la pila principal; en lugar de ello, las aplicaciones individuales ejecutan una o más capas verticales de la pila, como se discutirá más adelante.

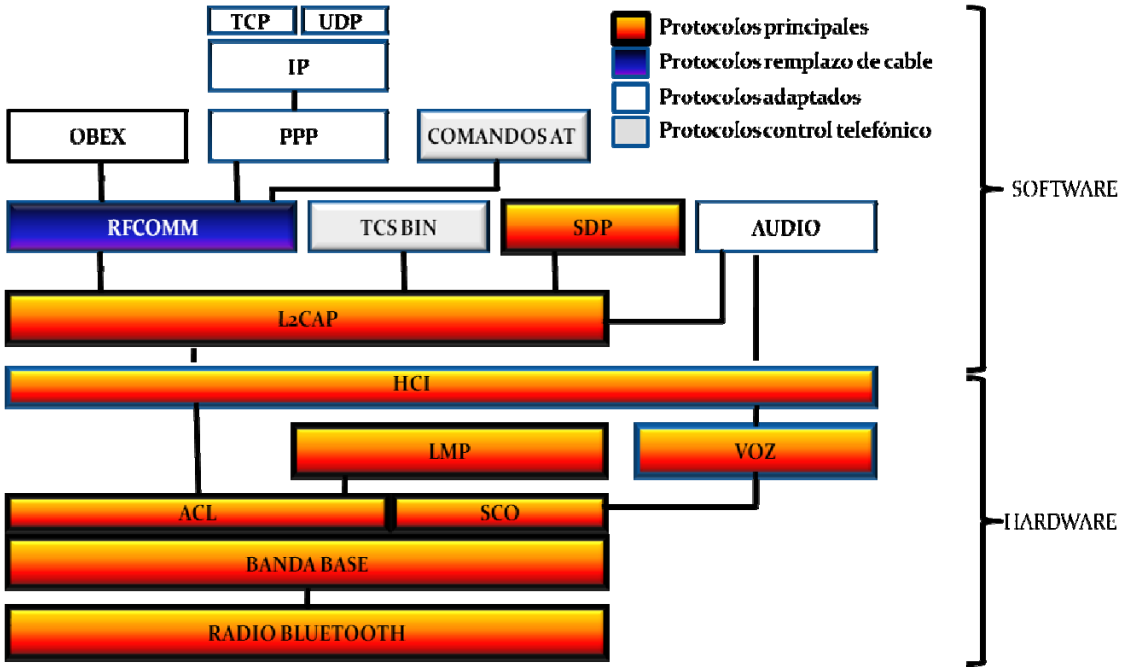


Figura 2.9 La pila completa de protocolos *Bluetooth* [5].

La pila de protocolos *Bluetooth* se divide en dos zonas:

- El módulo *Bluetooth* (hardware), encargado de las tareas relacionadas con el envío de información a través de la interfaz de radiofrecuencia.
- El modulo *Bluetooth* (software), encargado de la parte relacionada con las capas superiores de enlace y aplicación.

Ambas zonas están comunicadas por el HCI (Host Controller Interface).

Sobre la capa de protocolos específicos de *Bluetooth*, cada fabricante puede implementar su capa de protocolos de aplicación propietarios. De esta forma, la especificación abierta de *Bluetooth* expande enormemente el número de aplicaciones que pueden beneficiarse de las capacidades que ofrece esta tecnología inalámbrica. Sin embargo, la especificación *Bluetooth* exige que, a pesar de la existencia de diferentes pilas de protocolo de aplicación propietarios, se mantenga la interoperabilidad entre dispositivos que implementan diferentes pilas.

La pila de protocolo *Bluetooth* además es dividida en cuatro secciones principales, de acuerdo a su función. La Tabla 2.2 detalla las diferentes secciones de protocolos, y los protocolos específicos incluidos en cada sección [5][7].

Sección de protocolos	Pila de protocolo
Protocolo central	Banda Base / (BaseBand)
	Protocolo de Manejo de enlace / Link Manager Protocol (LMP)
	Protocolo de Control de Enlace Lógico y Adaptación / Logical Link Control and Adaptation Protocol (L2CAP)

	Protocolo de descubrimiento de Servicios / Service Discovery Protocol (SDP)
Protocolo de reemplazo de cable	RFCOMM
Protocolo de control de telefonía	Especificación de Telefonía de Control Binaria/ Telephony Control Specification-Binary (TCS-BIN)
	Comandos-AT / AT-Commands
Protocolos Adoptados	Protocolo Punto a Punto / Point-to-Point Protocol (PPP)
	Protocolo de Control de Transmisión / Transmission-Control-Protocol (TCP) Protocolo de Internet / Internet Protocol (IP) Protocolo de Datagramas de Usuario / User Datagram Protocol (UDP)
	Protocolo de Intercambio de Objetos / Object Exchange Protocol (OBEX)
	Comunicación de Móviles Infrarrojos / Infrared Mobile Communication (IrMC)
	Protocolo de Aplicaciones Inalámbricas / Wireless Application Protocol (WAP)
	Ambiente de Aplicaciones Inalámbricas / Wireless Application Environment (WAE)
	Formatos de Contenido / vCard, vCalendar, vMessage, and vNote

Tabla 2.2 Secciones de la pila de protocolos *Bluetooth* [5].

2.6.1 Radio Bluetooth

En la Figura 2.9 se observa que al igual que el modelo OSI, la primera capa de la pila *Bluetooth* corresponde a la capa de Radio *Bluetooth* o capa física, la cual es responsable de transportar bits entre sistemas o dispositivos adyacentes a través del aire. La capa de Radio *Bluetooth* se limita a lo siguiente:

- Recibir el flujo de bits de las capa superiores, y convertirlos para ser transmitidos como ondas de radio a estaciones asociadas.
- Recibir las ondas de radio de las estaciones asociadas, y convertirlas en flujos de bits para ser entregadas a su capa superior [3].

- **Bandas de frecuencia:** el estándar *Bluetooth* opera en la banda de 2.4 GHz, en todos los países esta banda está disponible ya que es definida para ISM, la cual esta reservada internacionalmente para uso no comercial de radiofrecuencia electromagnética en áreas industrial, científica y médica; el ancho de la banda puede diferir según el país. El rango exacto de la mayoría de las regiones o países se observa en la Tabla 2.3.

  Muchos dispositivos y tecnologías hacen uso de esa banda, tales como: monitores para bebé, controles para puertas de estacionamientos, teléfonos inalámbricos y hornos microondas (la fuente más fuerte de interferencia), por lo

que se hace necesario buscar la forma de evitar las interferencias, esto puede evitarse usando un esquema del espectro disperso.

GEOGRAFIA	RANGO REGULADO	Canales RF
EEUU, Europa y la mayoría de los países del mundo	2,400-2,483 Ghz	F=2402+k Mhz,k=0,.78
España	2,445-2,475 Ghz	F=2449+k Mhz,k=0,.22
Francia	2,446-2,483 Ghz	F=2454+k Mhz,k=0,.22
Japón	2,471-2,4835 Ghz	F=2473+k Mhz,k=0,.22

Tabla 2.3 *Bandas de frecuencia* [2].

- **Potencia:** los equipos de transmisión se agrupan en 3 clases según el nivel de potencia de emisión, tal y como se muestra a continuación[2]:
  - Clase 1: Dispositivos de largo alcance (aprox. 100m), con una potencia máxima de salida de 20dBm.
  - Clase 2: Dispositivos de medio alcance (aprox. 10m), con una potencia máxima de salida de 4dBm.
  - Clase 3: Dispositivos de corto alcance (aprox. de 1m a 3m), con una potencia máxima de salida de 0dBm.

  Debido a que el chip *Bluetooth* viene incorporado en dispositivos portátiles y alimentados con baterías, debe presentar un consumo de potencia muy reducido (hasta un 97% menos que un teléfono móvil).

- **Conmutación y velocidades:** el protocolo *Bluetooth* utiliza una conmutacion de circuitos y paquetes. Para asegurar que los paquetes no sean recibidos fuera de orden, se pueden reservar hasta 5 ranuras de tiempo.

  Los saltos de frecuencia son aplicados para evitar interferencia y desvanecimiento. Un salto de señal diferente es usado para cada paquete. La comunicación puede ser síncrona o asíncrona.

  Con modulación GFSK (Gaussian Frequency Shift Keying) se alcanzan tasas de transmisión nominales de 723 Kbps. Para la especificación *core* V2.0 + EDR (Enhanced Data Rate), con modulación π/4 DQPSK (π/4 Phase Differential Phase Shift Keying) la tasa de transmisión nominal es de 2 Mbps y con 8DQPSK (8 Phase Differential Phase Shift Keying) se tiene 3 Mbps [3].

2.6.2 Base Band Protocol

El protocolo de Banda base permite la conexión física RF (radio frecuencia) entre dos o más dispositivo *Bluetooth* que forman una *Piconet*. Este protocolo también sincroniza la frecuencia de transmisión por saltos y el reloj de los dispositivos *Bluetooth* en la *Piconet*.

Hay dos tipos distintos de conexión física provista por este protocolo banda base, un enlace orientada a conexión SCO, el otro tipo con un enlace no orientado a conexión ACL.

El dispositivo *maestro* controla el ancho de banda usado para cada enlace. Esta también decide cuánto debe ser el ancho de banda que se le dará a cada dispositivo *esclavo*.

Los datos binarios son típicamente transferidos de la capa de banda base a la capa LMP o L2CAP, el audio es enviado directamente de la aplicación a la capa de banda base, sin pasar por otras capas. Esto es esencialmente un enlace directo de audio entre dos dispositivos *Bluetooth* [5].

- **Enlace SCO (Synchronous Connection-Oriented):** son conexiones punto a punto entre el *maestro* y un *esclavo*, donde se reservan dos *slots* consecutivos en instantes fijos (un intervalo de tiempo para transmisión y el otro para recepción). SCO soporta conexiones por conmutación de circuitos, a una velocidad de transmisión de 64 KBps, y son típicamente usados para la transmisión de voz. Como muestra la Figura 2.10, un ejemplo de enlace SCO puede ser la voz entre un teléfono celular y un *headset*.

- **Enlace ACL (Asynchronous Connectionless Link):** son conexiones punto a multipunto entre el *maestro* y todos los *esclavos*, típicamente usadas en la transmisión de datos, donde se usan los *slots* restantes de SCO. Aquí, el *maestro* se encarga de configurar el tráfico. ACL soporta conexiones por conmutación de paquetes, y son típicamente usados para transmisión de datos, por ejemplo, entre un computador portátil y un teléfono móvil. Para los enlaces ACL se han definido los *slot*-1, *slot*-3 y *slot*-5.

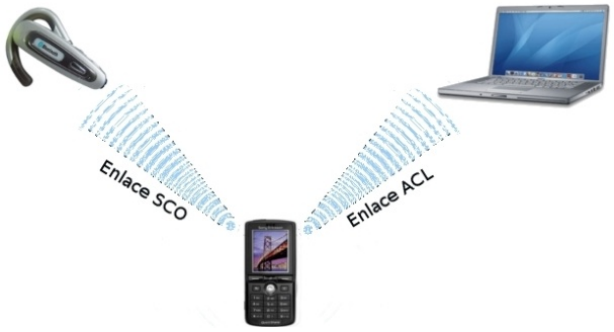


Figura 2.10 Enlace SCO y ACL

Descripción del canal *Bluetooth*: utiliza FH/TDD (Frequency Hopping/Time Division Duplex) (Figura 2.11). El canal contiene 79 frecuencias de radio diferentes, es dividido en intervalos de 625µs, llamados *slots* o ranuras, las cuales son accedidas de acuerdo a una secuencia de saltos aleatoria. Esto da

una tasa de saltos de 1600 saltos por segundo. Los *slots* son usados alternadamente para transmisión y recepción resultando un esquema TDD.

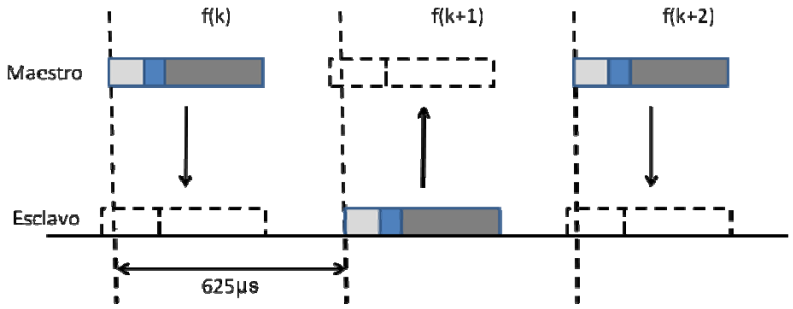


Figura 2.11 Ejemplo del canal FH/TDD utilizado en *Bluetooth*

El uso del esquema TDD hace que en el caso de que un canal este bloqueado las comunicaciones se vean muy poco afectadas.

En una transmisión, cada paquete debe estar alineado con el inicio de un *slot* y puede tener una duración de una, tres o cinco ranuras de tiempo**.** Durante la transmisión de un paquete la frecuencia es fija. Para evitar fallas en la transmisión, el *maestro* inicia enviando en las ranuras pares y los *esclavos* en las ranuras impares [2]. En la Figura 2.12 se puede ver este esquema de multiranuras.

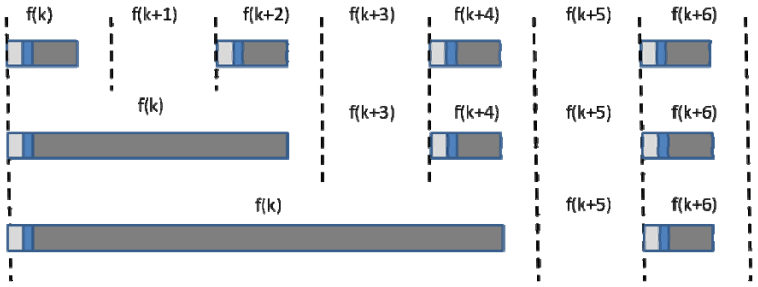


Figura 2.12 Paquetes multiranuras.

- **Salto de frecuencia:** debido a que la banda ISM está abierta a cualquiera, el sistema de radio *Bluetooth* deberá estar preparado para evitar las múltiples interferencias que se pudieran producir. Éstas pueden ser evitadas utilizando un sistema que busque una sección no utilizada del espectro o un sistema de salto de frecuencia. En los sistemas de radio *Bluetooth* se utiliza la modalidad de FHSS (Frequency Hopping Spread Spectrum) para minimizar interferencias, mejorar el nivel de seguridad y lograr una gran inmunidad a las interferencias. Esta tecnología puede ser integrada en equipos de baja potencia y bajo costo [6].

- **Reloj *Bluetooth*:** el tiempo y la secuencia de salto del transmisor-receptor es determinada por el reloj que posee cada dispositivo *Bluetooth*. Este es un reloj que transcurre libremente, nunca se ajusta y nunca se apaga. Ahora si desea sincronizarse con otros dispositivos, se utilizan valores de desplazamiento que al agregarse al reloj nativo proporcionan relojes temporales *Bluetooth* que son mutuamente sincronizados.

  Es importante tener en cuenta que los relojes *Bluetooth* no tienen relación con la hora y el día; por lo que pueden ser inicializados en cualquier valor. El reloj tiene un ciclo de alrededor de un día. El tiempo y la frecuencia de salto en el canal de una *Piconet* se determinan por el reloj del dispositivo *Bluetooth maestro*. Cuando se establece la *Piconet*, el reloj *maestro* se comunica a los *esclavos*. Cada *esclavo* agrega un offset a su propio reloj para sincronizarse con el reloj *maestro*.

❖ **Formato general de los paquetes:** los datos que se envían por los canales de una *Piconet* son transportados en paquetes. El formato del paquete general se observa en la Figura 2.13, el cual consiste básicamente de tres entidades: código de acceso, una cabecera y una carga útil o *payload*. En la figura se puede observar el número de bits correspondiente a cada entidad.

Figura 2.13 Formato del paquete estándar de banda base [6].

El Código de acceso y la Cabecera son de tamaño fijo, 72bits y 54bits respectivamente, en cambio la Carga útil puede variar su tamaño con un rango de 0 a 2745 bits [6].

- **Access code (Código de acceso):** Cada paquete comienza con un código de acceso. Si el campo cabecera (header) le sigue, entonces el campo código de acceso tendrá 72 bits en caso contrario usará solo 68 bits de longitud. El código de acceso es usado con un propósito de señalización. Este campo consta de un preámbulo (preamble), una palabra de sincronización (sync word), y una cola (trailer) como se ve en la Figura 2.14. El preámbulo le indica al receptor la llegada de un paquete. La palabra de sincronización es usada para sincronizar el tiempo con el receptor. El receptor correlaciona con la palabra de sincronización en el código de acceso, lo cual resulta en un mecanismo de señalización robusta. La cola es añadida a la palabra de sincronización como aviso de que el campo cabecera (header) sigue después del campo código de acceso.

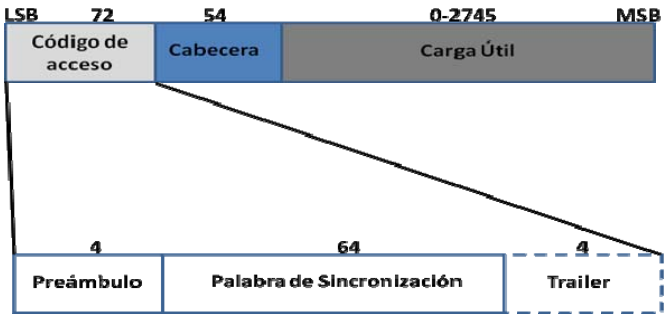


Figura 2.14 Los campos del código de acceso [6].

Las funciones provistas por el código de acceso pueden ser diferenciadas, dependiendo de el modo de operación en que se encuentre el dispositivo *Bluetooth*. De acuerdo a esto, existen tres tipos de código de acceso:

- **CAC (Channel access code):** es el código de acceso al canal que identifica a una *Piconet*. Este código está incluido en todos los paquetes intercambiados en un canal de *Piconet*.
- **DAC (Device access code):** es el código de acceso al dispositivo que es usado por procesos de señalización especiales, como lo son el *Page* y *Page* response.
- **IAC (*Inquiry* access code):** existen dos tipos de código de acceso de *Inquiry*: general y dedicado. Un código de acceso de investigación general es común para todos los dispositivos. Este es usado para descubrir otros dispositivos *Bluetooth* que se encuentren en el rango. El código de acceso dedicado es común para un grupo dedicado de dispositivos *Bluetooth* que comparten características comunes.

- **Header (Cabecera):** la cabecera contiene información de control de enlace y consta de seis campos totalizando 18 bits, se llega a 54 bits en la cabecera ya que es codificado con una tasa de 1/3 FEC, lo que hace básicamente es repetir 3 veces cada bit enviado dentro de la cabecera (Ver Figura 2.15) [6].

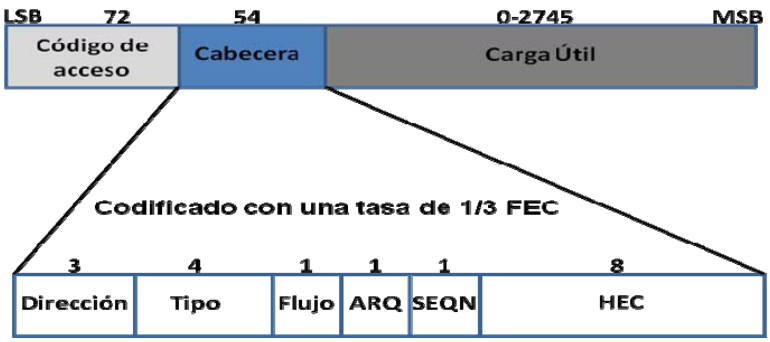


Figura 2.15 Formato de la Cabecera [6].

Tipos de paquetes comunes: Se han definido cuatro tipos de paquetes, los cuales son independientes de los tipos de enlace físicos subyacentes.

- **Paquete ID:** los paquetes ID están definidos para ser usados en los procedimientos de *Page* e *Inquiry*.
- **Paquete NULL:** no posee carga útil y además consiste de sólo un código de acceso al canal (CAC) y una cabecera, es usado para devolver la información del enlace a la fuente en cuanto sea exitosa la transmisión.
- **Paquete *POLL*:** Al contrario del paquete NULL, este requiere una confirmación. Este paquete puede ser usado por un *maestro* en una *Piconet* para encuestar *esclavos*, los cuales deben responder incluso si ellos no poseen información que enviar.
- **Paquete FHS:** El paquete FHS es un paquete de control especial para revelar entre otras cosas, la dirección del dispositivo *Bluetooth* y el reloj del remitente.

- **Payload (Carga útil):** Los datos que transporta el mensaje varían en tamaño de 0 a 2744 bits (Ver Figura 2.16). 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.*

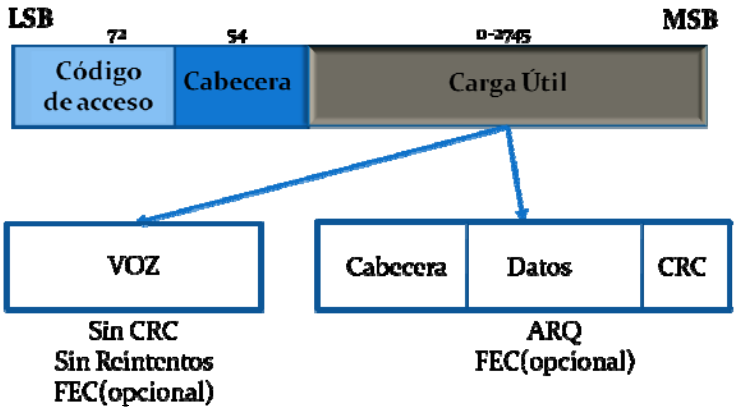


Figura 2.16 Formato de Carga Útil [6].

2.6.3 Link Manager Protocol (LMP)

Se encuentra justo arriba del protocolo de banda base, es responsable de la configuración de enlace y el control entre dos o más dispositivos *Bluetooth*. Esto incluye un número de aspectos de seguridad, como la autenticación, encriptación,

el control y negociación del tamaño de los paquetes de banda base. Este protocolo también controla los modos de ahorro de energía y los ciclos del radio *Bluetooth*, así como el estado de la conexión de los dispositivos *Bluetooth* cuando son añadidos a una *Piconet*.

El LMP soporta mensajes para: autenticación, paridad, encriptación, temporización y sincronización, versión y características, cambio 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, modos *Hold/sniff/park*, enlaces SCO*,* control de paquetes multiranuras, supervisión de enlace.

2.6.4 Host Controller Interface (HCI)

La *HCI* proporciona una interfaz de comando para el *controlador banda base* y a la *gestión de enlace,* además de acceso al hardware y a los registros de control. Esta interfaz brinda un método estándar para acceder a los recursos de banda base *Bluetooth*.

2.6.5 Logical Link Control and Adaptation Protocol (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 de aplicaciones la transmisión y recepción de paquetes de datos L2CAP de hasta 64 KB, con capacidad de multiplexación de protocolo, operación de segmentación y reensamblaje, y abstracción de grupos. Para cumplir sus funciones, L2CAP espera que la banda base suministre paquetes de datos en *full duplex*, 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 SDP (Service Discovery Protocol), RFCOMM y TCS (Telephony Control Specification) [2].

- **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 LMP*.* En el receptor los paquetes pequeños 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, funcionalmente es absolutamente necesaria para soportar protocolos usando paquetes más grandes que los soportados por la banda base. La Figura 2.17 muestra la segmentación L2CAP*.*

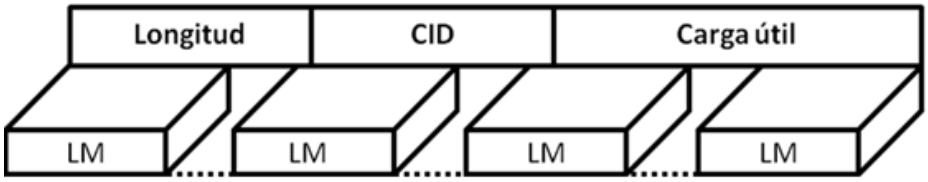


Figura 2.17 Segmentación L2CAP

- **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 2.18, los paquetes de canal orientado a la conexión están divididos en tres campos: longitud de la información, identificador de canal, e información.

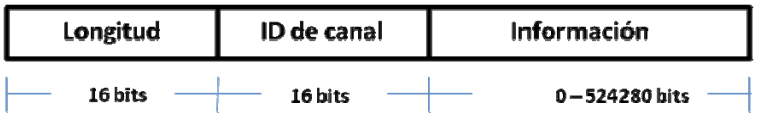


Figura 2.18 Paquete L2CAP [6].

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 [6][7].

2.6.6 Service Discovery Protocol (SDP)

Este protocolo provee los medios necesarios para que aplicaciones descubran qué servicios se encuentran disponibles y determinar las características de dichos servicios.

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 dispositivoes *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 el servidor responde con un mensaje. El SDP solamente soporta el descubrimiento del servicio, no la llamada del servicio [3][8].

2.6.7 Protocolo de reemplazo de cable

La especificación *Bluetooth* incluye sólo un protocolo que se refiere a la emulación inalámbrica de datos enviados a través de cable basadas en vínculos de RFCOMM.

- **RFCOMM** : la capa RFCOMM brinda emulación de puertos seriales sobre el protocolo L2CAP, es una capa simple de transporte provista adicionalmente de emulación de circuitos de puerto serial RS-232. El protocolo RFCOMM soporta hasta 60 puertos emulados simultáneamente. Dos dispositivos *Bluetooth* que usen RFCOMM en su comunicación pueden abrir varios puertos seriales emulados, los cuales son multiplexados entre sí. La Figura 2.19 muestra el esquema de emulación para varios puertos seriales.

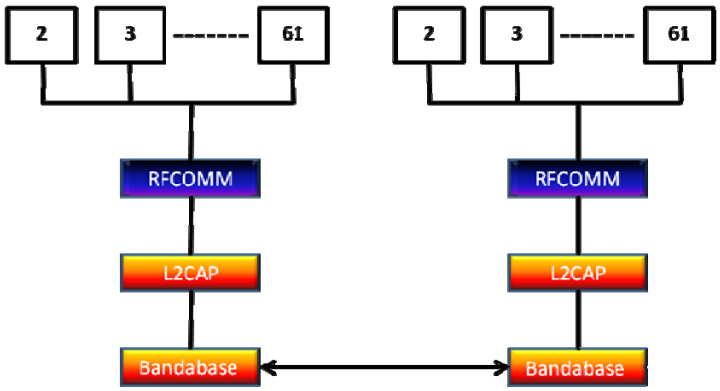


Figura 2.19 Varios puertos seriales emulados mediante RFCOMM [6].

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 modem nulo y usa a L2CAP para cumplir con el control de flujo requerido por alguna aplicación [6][7].

2.6.8 Protocolos adoptados

Además de los protocolos anteriores, una serie de protocolos de otras industrias fueron adoptados para ser usadas en la pila de protocolos *Bluetooth*. Esto hace que aplicaciones antiguas funcionen con la nueva tecnología *Bluetooth*, y a los dispositivos *Bluetooth* conectarse con la red de comunicación global [5].

De los protocolos adoptados que se muestran en la Tabla 2.2, los siguientes fueron usados en esta investigación.

- **Protocolo Punto a Punto (PPP):** el Protocolo de Punto a Punto PPP, desarrollado por el equipo de IETF (Internet Engineering Task Force), es un

protocolo de nivel de enlace asociado a la pila TCP/IP, en el cual los datos son transmitidos sobre un enlace serial punto a punto. Este protocolo es generalmente usado en las conexiones a Internet vía *dial-up*.

En la tecnología *Bluetooth*, PPP se ejecuta sobre el protocolo RFCOMM para establecer una conexión punto a punto entre dispositivos *Bluetooth*. El protocolo PPP puede ser usado en los accesos de red local, redes de acceso telefónico y perfiles de fax.

- **TCP (Transmission Control Protocol), IP (Internet Protocol) y UDP (User Datagram Protocol):** estos tres protocolos TCP, IP y UDP, son protocolos establecidos que definen en su mayoría, las comunicaciones basadas en Internet y las comunicaciones relacionadas con la red, así como las comunicaciones entre otros tipos de dispositivos informáticos y periféricos,. *Bluetooth* se ha apropiado de estos protocolos para facilitar la comunicación con cualquier otro dispositivo conectado a Internet.

  Adicionalmente existen otros tipos de protocolos adaptados, los cuales no fueron utilizados en esta investigación, tales como:

  - OBEX (OBject Exchange/Protocolo de Intercambio de Objetos)
  - IRMC (Infrared Mobile Communications/Comunicación de móviles infrarrojos)
  - WAP (Wireless Application Protocol/Protocolo de aplicaciones inalámbricas)
  - WAE (Wireless Application Enviromet/Ambiente de aplicaciones inalámbricas)

2.6.9 Perfiles Bluetooth

Los perfiles *Bluetooth* son guías que indican los procedimientos por los que los dispositivos equipados con tecnología *Bluetooth* se comunican entre sí. Existe un amplio número de perfiles que detallan los diferentes tipos de uso y aplicaciones de la tecnología inalámbrica *Bluetooth*. Al seguir las directrices proporcionadas en las especificaciones *Bluetooth*, los desarrolladores pueden crear aplicaciones compatibles con otros dispositivos que se ajusten a este estándar.

Cada perfil incluye, como mínimo, información sobre los siguientes aspectos:

- Dependencia de otros perfiles.
- Propuestas de formato de interfaz de usuario.
- Características concretas de la pila de protocolos *Bluetooth* utilizada por el perfil. Para realizar su función, cada perfil se sirve de ciertas opciones y

parámetros en cada capa de la pila. También se puede incluir un breve resumen de los servicios requeridos si resulta necesario.

En esta investigación el perfil que se utilizó fue:

Perfil PAN (Personal Area Network): describe cómo dos o más dispositivos con tecnología *Bluetooth* pueden formar una red ad hoc[2] y cómo ese mismo mecanismo permite acceder a la red de forma remota a través de un punto de acceso.

El PAN define tres funciones: punto de acceso a la red (NAP), grupo de red ad hoc (GN) y usuario de la PAN (PANU).

- **NAP (Network Access Point) y servicio NAP:** es un dispositivo con tecnología *Bluetooth* que proporciona algunas de las funciones de un puente Ethernet para hacer compatibles servicios de red.
- **GN (*Group Network*) y servicio GN:** es un dispositivo capaz de enviar paquetes Ethernet a cada uno de los dispositivos *Bluetooth* conectados.
- **PANU (Personal Area Network Users) y servicio PANU:** se trata del dispositivo *Bluetooth* que usa la NAP o el servicio GN.

El perfil PAN brinda capacidades de red a los dispositivos *Bluetooth* para lo cual utiliza el Protocolo de Encapsulamiento de Red BNEP (*Bluetooth* Network Encapsulation Protocol), de gran importancia ya que encapsula los paquetes provenientes de varios protocolos de red 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.

Este protocolo está implementado usando canales L2CAP orientados a conexión. Se considera a L2CAP como la capa de Control de Acceso al Medio MAC (Medium Access Control) de *Bluetooth*, BNEP especifica una mínima MTU (Maximum Transmission Unit) para L2CAP de 1691 bytes [9].

La Figura 2.19 muestra la ubicación del BNEP dentro de la pila de protocolos de *Bluetooth*.

[2] http://www.faqs.org/rfcs/rfc2501.html

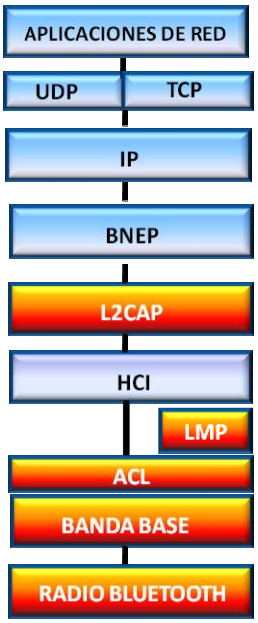


Figura 2.20 Ubicación del BNEP dentro de la pila de protocolos de *Bluetooth*.

La Figura 2.20 muestra la manera como BNEP remueve el encabezado de un trama 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*.

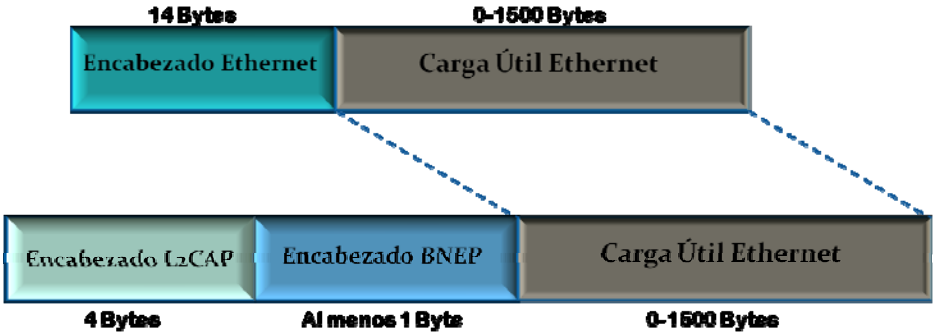


Figura 2.21 Encapsulamiento de un Paquete Ethernet en un paquete L2CAP

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.

Adicionalmente, como ya se mencionó, existe un amplio número de perfiles con diferentes funcionalidades, tales como:

- Perfil de Distribución de Audio Avanzado (A2DP)

- Protocolo de Control de Audio y Vídeo (AVRCP)
- Perfil Básico de Imagen (BIP)
- Perfil Básico de Impresión (BPP)
- Perfil de Telefonía Inalámbrica (CTP)
- Perfil de Red de Marcado (DUN)
- Perfil de Fax (FAX).
- Perfil de Transferencia de Archivos (FTP)
- Perfil de Distribución Genérica de Audio y Video (GAVDP)
- Perfil Genérico de Intercambio de Objetos (GOEP)
- Perfil Manos Libres (HFP)
- Perfil de Sustitución de Cable de Copia Impresa (HCRP)
- Perfil de Auricular (HSP)
- Perfil de Dispositivo de Interfaz Humana (HID)
- Perfil de Intercomunicador (ICP)
- Perfil de Introducción de Objetos (OPP)
- Perfil de Aplicación de Descubrimiento de Servicio (SDAP)
- Perfil de Servicio de Puerto (SPP)

3. METODOLOGÍA

En este Capítulo, se realiza una descripción detallada de los escenarios en donde se efectuaron las pruebas de desempeño para la investigación. Se dará una descripción de los elementos de hardware y software utilizados durante las pruebas, adicionalmente se indican los pasos necesarios para la configuración de los escenarios.

3.1 Descripción de los escenarios

Para este estudio se definieron una serie de escenarios basados en distintas topologías de redes con tecnología *Bluetooth*, los cuales fueron sometidos a una serie de pruebas para comprobar su desempeño. A continuación se describen cada uno de los escenarios; acompañados de una imagen que lo describe gráficamente. La Figura 3.1 muestra una leyenda de los elementos involucrados en cada uno de los escenarios.

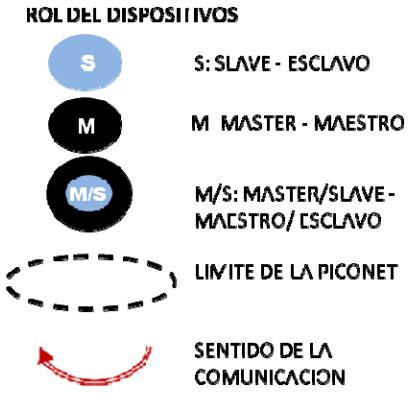


Figura 3.1 Descripción de partes de una *Piconet*

Escenario 1- E1

Se basa en una *Piconet* de dos nodos; un nodo M y un nodo S, donde la comunicación se realiza del nodo S al nodo M. Teóricamente, este es el escenario con el mejor desempeño posible; debido a su simplicidad. La Figura 3.2 muestra el escenario descrito.

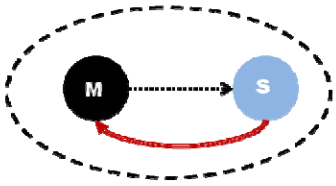


Figura 3.2 Escenario 1

Escenario 2- E2

Se basa en una *Piconet* con 3 nodos, un nodo M y 2 nodos S. La comunicación se realiza entre los nodos S pasando por un nodo M intermedio. Con este escenario se pretende observar los efectos causados por un nodo intermedio o nodo puente. La Figura 3.3 muestra el escenario descrito.

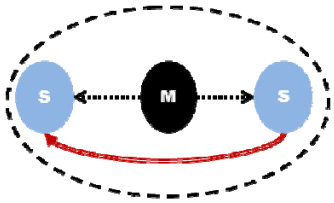


Figura 3.3 Escenario 2

Escenario 3 - E3

Se basa en una *Piconet* con el máximo de nodos permitidos para su conformación, el cual es 8, distribuidos en un *maestro* y 7 *esclavos*. La comunicación se realiza desde uno de los nodos S al nodo M (similar al escenario E1); pero en este caso, se comprobó el efecto en la comunicación al poblar de nodos la *Piconet*. La Figura 3.4 muestra el escenario descrito.

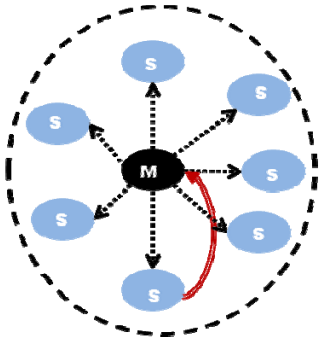


Figura 3.4 Escenario 3

Escenario 4 - E4

Esta topología es similar a la de E3; sin embargo, la comunicación se realiza entre nodos *esclavos* (similar a E2). En particular se desea observar el efecto de tener un salto o nodo intermedio en la comunicación, y el efecto de poblar al límite la *Piconet*. La Figura 3.5 muestra el escenario descrito.

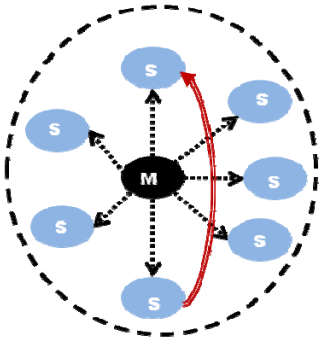


Figura 3.5 Escenario 4

Escenario 5 - E5

Este escenario introduce el concepto de *Scatternet*, es decir, la topología se encuentra conformada por 2 *Piconets* (*Piconet* 1 y *Piconet* 2). Este escenario se observa en la Figura 3.6; la *Piconet* 1 formada por 2 nodos, un *esclavo* y un *maestro* con doble función, el cual a su vez es *esclavo* de la *Piconet* 2. La *Piconet* 2 consta de un nodo *esclavo* con doble función y su *maestro*. La comunicación para este escenario se da en la *Piconet* 1, específicamente entre el nodo S y el nodo M/S. Se buscó comprobar el efecto que causa que un nodo tenga doble papel, al participar en dos *Piconets*.

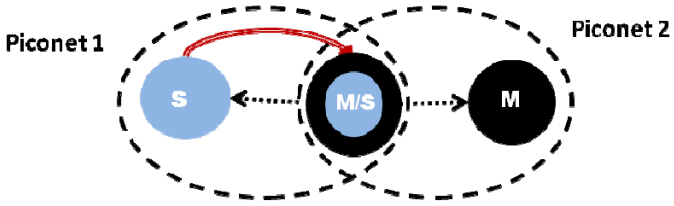


Figura 3.6 Escenario 5

Escenario 6 - E6

Esta topología es similar a E5; sin embargo, la comunicación se realiza del nodo S de la *Piconet* 1 al nodo M de la *Piconet* 2. Con este escenario se pretende observar los efectos causados por un salto o nodo intermedio, agregando el hecho de estar en una *Scatternet* y que el nodo intermedio desempeñe doble función. La Figura 3.7 muestra el escenario descrito.

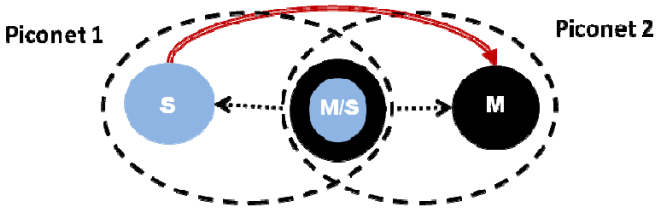


Figura 3.7 Escenario 6

Escenario 7 - E7

En este escenario conformado por la *Piconet* 2 de 2 nodos, un nodo M y un nodo M/S, el cual cumple funciones de *maestro* en la *Piconet* 1. La *Piconet* 1 consta de 5 nodos S. La comunicación para este escenario se da en la *Piconet* 2, específicamente entre M y M/S. Acá se busca comprobar el efecto que causa que un nodo tenga doble papel, al participar en dos *Piconets*, y el efecto de poblar considerablemente una de las *Piconet*s La Figura 3.8 muestra el escenario descrito.

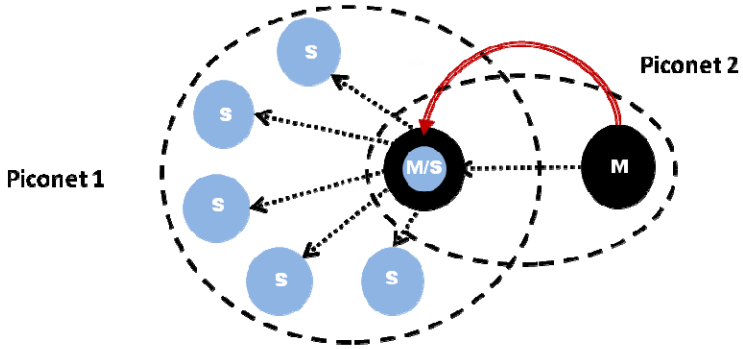


Figura 3.8 Escenario 7

Escenario 8 - E8

En este escenario similar a E7; sin embargo, la comunicación se realiza del nodo M de la *Piconet* 1 a un nodo S de la *Piconet* 2. Se desea observar el efecto que causa el salto entre las *Piconets* (semejante a E6) cuando el nodo intermedio tiene doble función, y la *Piconet* destino (*Piconet* 2) se encuentra poblada. La Figura 3.9, muestra el escenario descrito.

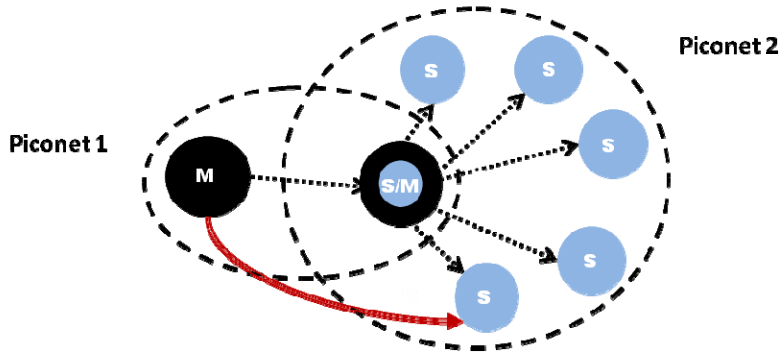


Figura 3.9 Escenario 8

Escenario 9 - E9

Este escenario es similar a E7, está conformado por la *Piconet* 1 de un nodo M, 5 nodos S y un nodo M/S de doble función, este cumple funciones de

maestro en la *Piconet* 2, la cual posee 5 nodos S, la comunicación para éste escenario ocurre en la *Piconet* 1, del nodo M al nodo M/S. De manera similar a E7, en este escenario se pretende comprobar el efecto que causa que un nodo tenga doble función (M/S), y el efecto de poblar considerablemente ambas *Piconets*. La Figura 3.10 muestra el escenario descrito.

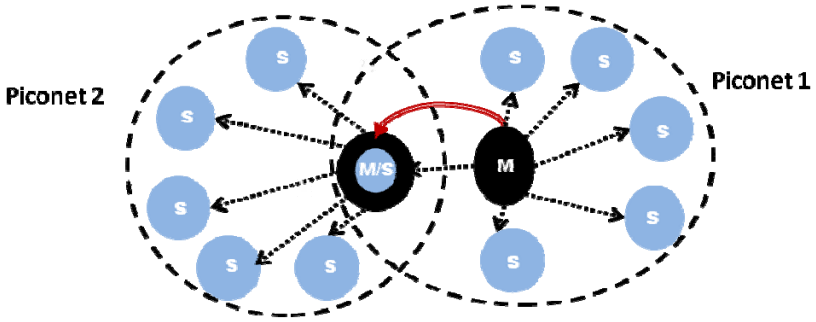


Figura 3.10 Escenario 9

Escenario 10 - E10

Este escenario es similar a E9; sin embargo, la comunicación se realiza del nodo M de la *Piconet* 1 a un nodo S de la *Piconet* 2. Se desea observar el efecto que causa el salto entre las *Piconets* (semejante a E8), cuando el nodo intermedio tiene doble función, y ambas *Piconets* se encuentran pobladas. La Figura 3.11 muestra el escenario descrito.

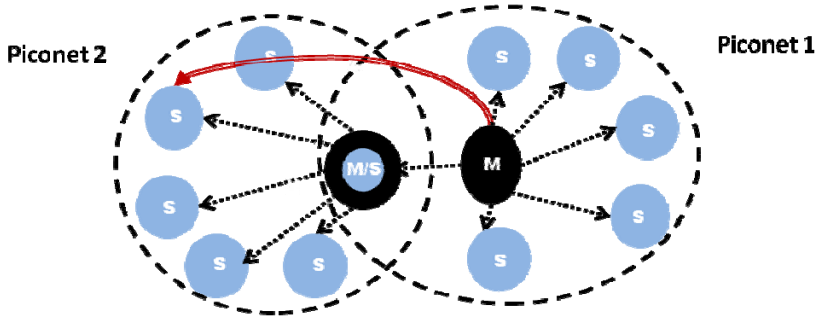


Figura 3.11 Escenario 10

Escenario 11 - E11

Este escenario es similar a E9; sin embargo, la comunicación se realiza de un nodo S en la *Piconet* 1 a un nodo S en la *Piconet* 2. Se desea observar el efecto que causan dos saltos en la comunicación, con ambas *Piconets* pobladas. La Figura 3.12 muestra el escenario descrito.

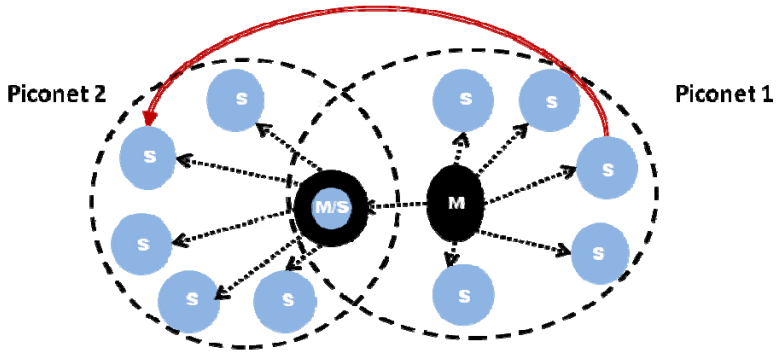


Figura 3.12 Escenario 11

Escenario 12 - E12

Este escenario consta de 6 *Piconets*, de dos nodos cada una. La *Piconet* 1 está formada por un nodo M y un nodo M/S, las *Piconets* 2, 3, 4, y 5 son *Piconets* donde los dos nodos son nodos M/S, y por último la *Piconet* 6 tiene un nodo M/S y un nodo S. La comunicación se realiza del *maestro* de la *Piconet* 1 al *esclavo* de la *Piconet* 6. Se desea observar el comportamiento de la comunicación con varios nodos puentes o intermedios (5 en este caso). La Figura 3.13 muestra el escenario descrito.

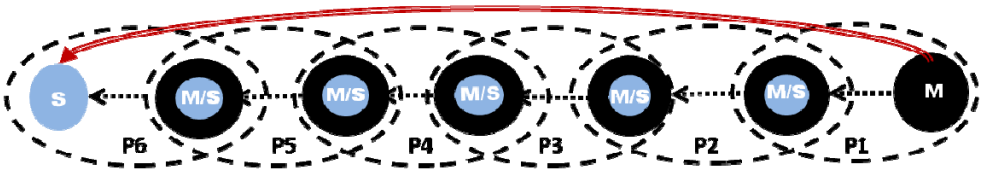


Figura 3.13 Escenario 12

Escenario 13 - E13

Este escenario es similar a E12; sin embargo, consta de 11 *Piconets*, de 2 nodos cada una, la *Piconet* 1 está formada por un nodo M y un nodo M/S, de la *Piconet* 2, a la *Piconet* 10; los dos nodos son M/S, y por último la *Piconet* 11 tiene un nodo M/S y un nodo S. La comunicación se realiza del *maestro* de la *Piconet* 1 al *esclavo* de la *Piconet* 11; para observar el comportamiento de la comunicación con una mayor cantidad de nodos puentes o intermedios (10 en este caso, limitados por el total de adaptadores *Bluetooth* para esta investigación). La Figura 3.14 muestra el escenario descrito.

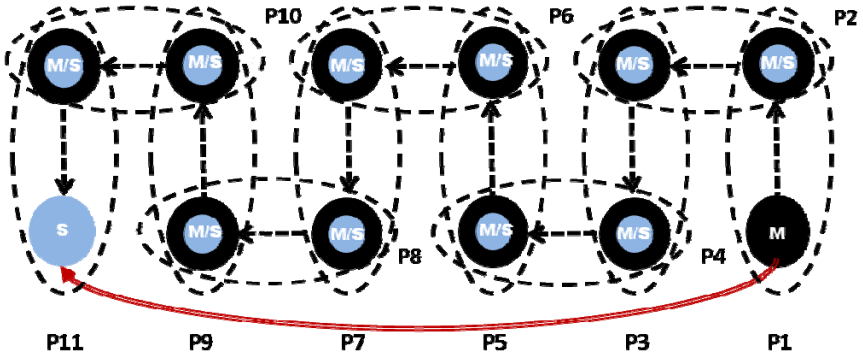


Figura 3.14 Escenario 13

Escenario 14 - E14.

En este caso no se intentará crear una topología para la comunicación, como la que se ve en la Figura 3.15, simplemente se desea verificar si es posible agregar más de 7 *esclavos* a una *Piconet*, como se define en la especificación de la tecnología *Bluetooth*.

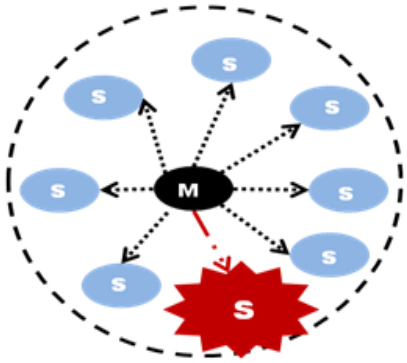


Figura 3.15 Escenario 14

Para finalizar la descripción de los escenarios, es importante resaltar que cada uno de los escenarios descritos anteriormente se implementó en un ambiente real, con hardware y software que serán descritos en las próximas secciones.

3.2 Métricas utilizadas.

Para realizar la medición del desempeño en la comunicación sobre los escenarios planteados, se tomo en consideración el principal grupo de métricas de desempeño en redes [11]:

- El *Bandwidth* o la capacidad del canal nominal y efectiva, busca medir la máxima cantidad de bits que se pueden transmitir por unidad de tiempo, depende del ancho de banda del medio físico, ondas electromagnéticas,

capacidad de procesamiento de los elementos transmisores, eficiencia de los algoritmos de acceso al medio, codificación de canal y compresión, la carga adicional (*overhead*) de los protocolos en las varias capas y limitaciones en los dispositivos extremos; los valores de esta métrica de desempeño se obtendrá del *Packet rate*, que muestra la tasa de paquetes recibidos por unidad de tiempo obtenidas a través del generador de tráfico D-ITG[3]. Con *Packet rate* se busca medir el desempeño de la red y tener conocimiento de la capacidad efectiva del canal según las variantes de los escenarios.

- El *Delay* es la métrica que busca medir el tiempo transcurrido en transmitir un paquete durante su trayecto desde el nodo origen al nodo destino, se usaran dos tipos de métricas de Delay: el IPDV (*IP Packet Delay Variation)* conocido como *Jitter* o dispersión del retardo, que muestra la diferencia entre el tiempo en que llega un paquete y el tiempo que se cree que llegara el paquete, este valor se obtendrá a través del generador de tráfico D-ITG y la medición de ida y vuelta RTT (round-trip-time) obtenidas por la herramienta *ping,* usada para comprobar de la conectividad en cada escenario. Con el Jitter se busca comprobar que tipo de tráfico se podría manejar, de este modo comprobar la calidad, confiabilidad y fallas presentadas en el enlace.

- *Loss and errors* es la métrica que describe cuanta información es transmitida desde el origen y no es entregada o es recibida con errores en el destino, esta métrica generalmente es medida en porcentaje de errores de bits o *Packets dropped* (paquetes descartados), siendo este ultimo el tomado para la investigación y obtenido a través del generador de tráfico D-ITG. Con *Packets dropped* se busca comprobar la confiabilidad de la red.

En cuanto a los criterios usados para determinar el desempeño de cada escenario, se consideró que el ambiente deseado debería poseer las siguientes características: un valor de *Packet rate* elevado, un valor de *Packets dropped* cercano a cero, y un *Jitter* cercano a cero. Un ambiente con estas características presentaran un óptimo desempeño en la comunicación, debido a que contaremos con el uso efectivo de la capacidad máxima del canal, una alta confiabilidad en la entrega de los paquetes libre de errores, y un alto nivel de calidad en el enlace.

Parámetros adicionales como: *total package* (total de paquetes), y el *bitrate* (tasa de bits) no fueron considerados para ser presentados en este Capítulo; debido a que estos se derivan de *Packet rate*. Los valores para estos parámetros se observan en las tablas de resultados de cada escenario, y sus gráficos se pueden observar en el Capítulo 7.

[3] http://www.grid.unina.it/software/ITG

3.3 Generación de tráfico.

Para la generación de tráfico usado en la medición de desempeño en las comunicaciónes sobre los escenarios se utilizo la herramienta D-ITG. Adicionalmente la generación de tráfico se realizó siguiendo las recomendaciones del RFC 2544[10].

Para cada uno de los escenarios se realizaron pruebas generando paquetes UDP con tamaños de 64, 256, 1024 y 1518 bytes respectivamente, por prueba sólo se generaban paquetes con uno de los tamaños anteriormente definidos, la duración de cada prueba fue de 300 segundos, y se fijo una tasa de 500 paquetes por segundo, ya que en pruebas preliminares se logro comprobar que no sobrepasa la tasa de 400 paquetes por segundos, con la idea de forzar al maximo a los nodos.

Cada una de las pruebas se realizaron 20 veces en un ambiente libre de tráfico, de igual modo también se realizaron 20 pruebas en un ambiente con tráfico; para contar con una muestra significativa de datos. En las pruebas realizadas bajo un ambiente con tráfico, aproximadamente un tercio de los nodos o dispositivos participantes de los escenarios generan tráfico en la red (*Piconet* o *Scatternet*), siendo dicho tráfico de naturaleza VoIP con paquetes de tamaño de 120 bytes a una tasa de transferencia de 100 paquetes por segundo. Esto último para estrezar la red y estudiar su comportamiento en condiciones de congestión.

Durante la realización de las pruebas los equipos se encontraban en línea de vista, a distancias equidistantes de aproximadamente 1 a 2 metros entre ellos; esto para recrear un ambiente controlado.

Los valores de las métricas obtenidas por las 20 repeticiones de pruebas realizadas sobre los escenarios con o sin tráfico, se promediaron para estudiar el comportamiento de la comunicación.

Para verificar la conectividad en cada uno de los escenarios se uso la herramienta de administración de redes Ping (*Packet Internet Groper*), estas pruebas se realizaron solo en un ambiente libre de tráfico y se realizaron 1000 ping para cada escenario, tomando en consideración sólo como métrica el promedio del rtt (*round-trip-time*), para el análisis del desempeño.

3.4 Descripción del hardware usado.

Para poder implementar los escenarios anteriormente mostrados y realizar las pruebas respectivas, se empleó un conjunto de equipos con componentes de hardware similares, tratando de contar con un ambiente homogéneo pero debido a las limitaciones y diferencia, las pruebas se realizaron finalmente en un ambiente heterogéneo el cual se adapta mejor a un ambiente real de comunicación.

En el caso de los *dongles* o adaptadores *Bluetooth*, tampoco se pudo contar con dispositivos de las mismas características. Por lo anterior se separaron los adaptadores en grupos, de manera tal que los equipos involucrados en las comunicaciones (origen, destino y nodos intermedios) tuvieran las mismas características. Los grupos restantes de adaptadores se utilizaron básicamente para poblar las *Piconets* en los escenarios pertinentes. Sin embargo, en E12 y E13 todos los *dongle Bluetooth* fueron utilizados para la transmisión de datos, debido a la naturaleza de sus topologías.

La descripción detallada de los equipos utilizados se observa en la Tabla 3.1.

CANT	MARCA	MODELO	CARACTERISTICAS
8	HP	Worstation xw 4600	CPU Intel Core 2 Duo E6750 2,66Ghz con Caché L2 de 4 Mb, bus 1333Mhz, Memoria Ram 2Gb DDR2 667, Disco Duro 250GB Sata 7200rpm y Puertos USB 2.0
2	HP	Pavillion dv6000	CPU Intel Core 2 Duo T5600 1,83 GHz con Caché L2 de 2 Mb, bus 667Mhz, Memoria Ram 2Gb DDR2 667, Disco Duro 160GB Sata 7200rpm y Puertos USB 2.0
1	HP	Compaq nx6320	CPU Intel Core 2 Duo T5600 1,83 GHz con Caché L2 de 2 Mb, bus 667Mhz, Memoria Ram 1Gb DDR2 533, Disco Duro 100GB Ide 5400rpm y Puertos USB 2.0
1	Toshiba	Satellite 2400-S201	CPU Intel Pentium 4 1.6 GHz, Memoria Ram 256 Mb DDR 266, Disco Duro 80GB IDE y Puertos USB 2.0

Tabla 3.1 Equipos utilizados en las pruebas.

La descripción detallada de los adaptadores o *dongles Bluetooth* se observa en la Tabla 3.2

CANT	MARCA	MODELO	CARACTERISTICAS
5	MSI	BToes 2.0	*Bluetooth* Class II V2.0, rango 10 m, tasa de transferencia 3 Mbps, USB 2.0
3	Genérico	Genérico	*Bluetooth* Class I V2.0, rango 100 m, tasa de transferencia 3 Mbps, USB 2.0
1	MSI	BToes MS6970	*Bluetooth* Class II V1.1, rango 10 m, tasa de transferencia 723 Kbps, USB 1.1
1	Targus	MBT-1203	*Bluetooth* Class II V1.0, rango 10 m, tasa de transferencia 723

			Kbps, USB 1.1
1	BAFU	BF-7221	*Bluetooth* Class II V1.1, rango 10 m, tasa de transferencia 723 Kbps, USB 1.1
1	INSTEN	INSTEN 1	*Bluetooth* Class II V1.2, rango 10 m, tasa de transferencia 723 Kbps, USB 1.1

Tabla 3.2 *Dongles Bluetooth* usados en las pruebas

3.5 Descripción del software usado

A diferencia del hardware, para el software se logro homogeneidad, dado que en todos los equipos se utilizaron los siguientes paquetes de software:

- Sistema Operativo Linux, en particular la distribución Ubuntu 8.04 – Hardy Heron.

- Pila oficial de protocolos *Bluetooth* compatible con dicha distribución, conocida como BlueZ[4]. El módulo de BlueZ provee bibliotecas y herramientas para trabajar y llevar a cabo las conexiones con la tecnología *Bluetooth*. La pila Bluez incluye el perfil PAN, que brinda capacidades de red a los dispositivos *Bluetooth* para lo cual utiliza BNEP, de gran importancia ya que encapsula los paquetes provenientes de varios protocolos de red 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 [9]. La descripción detallada de los paquetes de Bluez instalados se muestra en la Tabla 3.3.

- Para la medición del desempeño en cada escenario se utilizó la aplicación D-ITG (Distributed Internet Traffic Generator) disponible para los sistemas operativos Linux y Windows. Esta aplicación es compatible con IPv4 y IPv6 y es capaz de generar tráfico en las capas de red, transporte y aplicación asociados a la Pila TCP/IP; específicamente soporta protocolos tales como TCP, UDP, ICMP, DNS, Telnet, VoIP (G.711, G.723, G.729). D-ITG también brinda la oportunidad de configurar el tiempo de envío, tamaños de los paquetes y almacenar en un archivo los valores de salida (total de paquetes, jitter, latencia, bitrate, paquetes dañados, etc.).

Paquete	Versión	Descripción
Bluetooth	3.26-0ubuntu6	Provee utilidades para la gestión de la pila de protocolos de *Bluetooth*
bluez-audio	3.26-0ubuntu6	Brinda soporte para conexiones de audio vía *Bluetooth*

[4] www.bluez.org

bluez-btsco	1:0.50-0ubuntu2	Provee la forma de utlizar un manos libres con Linux
bluez-gnome	0.25-0ubuntu1	Provee applets GNOME para detectar y configurar adaptadores *Bluetooth*
bluez-hcidump	1.40-0ubuntu1	Permite el monitoreo de la actividad *Bluetooth* en una conexión
lib*Bluetooth*2	3.29-0ubuntu1	Biblioteca para el uso de la pila *Bluetooth* BlueZ de Linux
bluez-utils	3.26-0ubuntu6	Contiene herramientas y procesos de sistema para el uso de los dispositivos *Bluetooth*

Tabla 3.3 Paquetes pertenecientes a la pila *Bluetooth* BlueZ

Además se utilizaron otros paquetes .que brindan utilidades y herramientas para la conexión y establecimiento de una *Scatternet*. Ver Tabla 3.4.

Paquete	Versión	Descripción
bridge-utils	1.2-2	Provee una utilidad para configurar puentes Ethernet en Linux
Gnome-*Bluetooth*	0.11.0-0ubuntu1	Contiene herramientas para la gestión y manipulación de los dispositivos *Bluetooth* en el escritorio GNOME

Tabla 3.4 Paquetes adicionales

3.5.1 Herramientas y utilidades de software

Las siguientes herramientas y utilidades fueron provistas por la implementación del protocolo *Bluetooth* bluez.

- **hcitool**. Permite, según el parámetro que se le agregue, indagar los dispositivos que se encuentren cercanos; así como también ver y configurar las conexiones que se encuentran activas, entre otras funcionalidades.
- **pand**. Permite realizar la conexión directa con cada dispositivo, y también levanta un proceso en el sistema que escucha y atiende las conexiones entrantes. Además muestra una lista de las interfaces que se crean cada vez que se logra una conexión.

Las siguiente herramienta fue provista por el paquete bridge-utils:

- **brctl.** Permite configurar un puente Ethernet para lograr comunicar las interfaces creadas.

Las siguientes herramientas y utilidades fueron provistas por la aplicación D-ITG.

- **ITGSend.** Permite el envío de tráfico.
- **ITGRecv.** Permite la recepción del tráfico.

- **ITGDec.** Permite decodificar el archivo donde se almacena la captura de las mediciones de desempeño en la comunicación.

3.5.2 Configuración de los nodos

A continuación se muestra como configurar, paso a paso los distintos tipos de nodos (*esclavo*, *maestro*, *maestro/esclavo* y puentes), para lograr las conexiones. Se deben ejecutar todos los comandos en modo administrador del sistema.

Inicialización:

Se describen los pasos necesarios para la configuración inicial de un nodo, haciendo uso de la Figura 3.16:

1: Primero se debe abrir un intérprete de comandos, ejecutando el comando **hcitool** *dev*, para conocer la dirección del dispositivo *Bluetooth* que se está configurando.

```
1:  # hcitool dev
    Devices:
         hci0       00:0C:76:9A:31:03
2:  #
```

Figura 3.16 Ejemplo del comando hcitool dev

1: Luego se escanea con el comando **hcitool** *scan* en búsqueda de dispositivos que se encuentren cercanos, como se muestra en la Figura 3.17, con el fin de visualizar las direcciones de los dispositivo *Bluetooth* y el nombre del equipo, serán usados mas adelante.

```
1:  # hcitool scan
    scanning ...
         00:0C:76:9A:31:03      hp-laptop-0
         00:02:72:D1:65:4B      icaro01-0
         00:02:72:D1:68:BE      icaro02-0
         00:02:72:D1:85:42      icaro07-0
         00:02:72:D1:56:A2      Toshiba-0
         00:02:72:D1:02:13      icaro04-0

2:  #
```

Figura 3.17 Ejemplo del comando hcitool scan

Nodo *esclavo*:

A continuación se describen los pasos necesarios para la configuración de un nodo *esclavo*, haciendo uso de la Figura 3.18:

1: Se carga el módulo que activa las interfaces bnep (interfaces virtuales basadas en el protocolo BNEP).

2: Se crea un proceso para atender las conexiones entrantes.

3: Por último se comprueba que el servicio se encuentre activo.

```
1: # modprobe bnep
2: # pand --listen
3: # ps ax | grep pand
    6427 ?           Ss    0:00  pand --listen
    6429 pts/0       S+    0:00  grep pand
4: #
```

Figura 3.18 Configuración de nodo *esclavo*

En este punto el nodo sólo está esperando a recibir una conexión de algún *maestro* que desee conectarse con él, para la creación de una *Piconet*.

Nodo *maestro*:

A continuación se describen los pasos necesarios para la configuración de un nodo *maestro*, haciendo uso de la Figura 3.19.

1: Este nodo *maestro* controlará la conexión, utilizando el comando **pand** --*connect* y la dirección del dispositivo al que se desea conectar. En este ejemplo, se conectará el dispositivo que posee la dirección (00:0C:76:9A:31:03).

```
1: # pand --connect 00:0C:76:9A:31:03 -n
   pand[6362]: Bluetooth PAN daemon version 3.26
   pand[6362]: Connecting to 00:0C:76:9A:31:03
   pand[6362]: bnep0 connected
2: #
```

Figura 3.19 Conexión del nodo *maestro*

En este punto el nodo *maestro* ya posee una conexión con el nodo *esclavo* al que se conectó. Este nodo puede repetir hasta 7 veces esta operación con otros dispositivos (un nodo *maestro* sólo puede poseer 7 *esclavos* o enlaces conectados). En este nodo se habrán creado tantas interfaces *bnep* como conexiones hayan logrado.

1: En la Figura 3.20 se pueden observar las interfaces que se crearon y su número asociado, se puede verificar a que interfaz bnep está asociado cada dispositivo y su rol dentro de la conexión, como se puede observar en la Figura 3.20 se muestra la palabra PANU, lo que indica que posee dos *esclavos* conectados.

```
1:  # pand -show
    bnep0: 00:0C:76:9A:31:03 PANU
    bnep1: 00:0C:DF:95:31:08 PANU
2:  #
```

Figura 3.20 Vista de las conexiones del *maestro*

Nodo *maestro/esclavo*:

La configuracion de este nodo se basa en sólo realizar los procedimientos de conexión del *esclavo* y para luego realizar el procedimiento de conexión del *maestro*, siguiendo estrictamente ese orden. De este modo el nodo *maestro/esclavo* quedará preparado para recibir alguna conexión (como *esclavo*) y a la vez preparado para conectar a cualquier otro nodo (como *maestro*).

1: Para visualizar las interfaces que se han creado producto de la conexiones, como se observa en la Figura 3.21.

```
1:  # pand -show
    bnep0: 00:0C:76:9A:31:03 GN
    bnep1: 00:0C:DF:34:36:08 PANU
    bnep2: 00:0C:DF:56:E7:5A PANU
    bnep3: 00:0C:DF:56:31:21 PANU
2:  #
```

Figura 3.21 Vista de las conexiones *maestro/esclavo*

Se puede observar que existen 3 conexiones, identificadas con la palabra PANU como ya se habia visto estos serían 3 *esclavos*; y existe otra conexión en la cual él es *esclavo*, indicando con la palabra GN.

Puente:

Para lograr que exista conexión entre todos los nodos de la misma *Piconet* y entre *Piconets* distintas (*Scatternet*), se debe configurar un puente entre las interfaces para que los dispositivos logren enrutar los paquetes en la red. La Figura 3.22 describe como se configura el puente con la herramiente **brctl**.

1: Con el comando **brctl** *addbr* se crea una interfaz llamada puente.

2: Con el comando **brctl** *addif* se añaden al puente las interfaces bnep que se deseen comunicar, en este caso todas las interfaces bnep.

3: Con el comando **brctl** *show* se visualizan las interfaces que estan asociadas al puente.

```
1: #brctl addbr puente
2: #brctl addif puente bnep0 bnep1 bnep2
3: #brctl show
   bridge name bridge id         STP enable        interfaces
   puente            8000.00272b87     no                bnep0
                                                       bneo1
                                                       bnep2
4: #
```

Figura 3.22 Configuración del puente

Cabe destacar que esta configuración se hace en los nodos que posean más de una interfaz bnep o mas de una conexión, es decir nodos *maestros* o nodos *maestros*/*esclavos*.

Asignaciones de direcciones IPs:

Una vez completados todos los pasos de configuración, se debe asignar direcciones IP que pertenezca al mismo segmento de red; para lograr la comunicación entre los dispositivos.

1: Los nodos *esclavos* sólo poseen una interfaz (bnep0), se debe configurar como se indica en la Figura 3.23. En este ejemplo se coloca una direccion IP perteneciente a la red 10.0.0.0.

```
1: # ifconfig bnep0 10.0.0.2
2: #
```

Figura 3.23 Configuración de IP para nodos *esclavos*

Para los nodos *maestros* y *maestro*/*esclavo* se deben hacer configuraciones extras, como se observa en la Figura 3.24.

1: La interfaz de comunicación para estos nodos es la interfaz puente descrita anteriormente, y se le asigna una dirección IP.

2-4: Para cada interfaz bnep asociada al puente se le asigna el valor 0, para la activación de la misma.

```
1:  # ifconfig puente 10.0.0.2
2:  # ifconfig bnep0 0
3:  # ifconfig bnep1 0
4:  # ifconfig bnep2 0
5:  #
```

Figura 3.24 Configuración de IP para los nodos puentes

Una vez realizados estas configuraciones los dispositivos estarán conectados entre sí.

A continuación se describen los pasos necesarios para la generación de tráfico utilizando la herramienta D-ITG:

Generación de tráfico

1: Desde el directorio */bin* de la carpeta D-ITG se ejecuta el comando *ITGSend* para iniciar el envío de paquetes UDP con cada uno de los tamaños seleccionados. Un ejemplo del comando se observa en la Figura 3.25.

```
1:  ~/bin/# ./ITGSend -a 10.0.0.1 -rp 8000 -sp 8100 -t 300000 -C 500 -c 64 -x
    e3st64-1
2:  ~/bin/#
```

Figura 3.25 Ejemplo de generación de tráfico UDP con D-ITG

Significado de los parametros del comando:

- -a: dirección IP del nodo destino que recibirá el flujo de paquetes.
- -rp: puerto de recepción en el nodo destino.
- -sp: puerto de envío en el nodo fuente.
- -t: tiempo de duración en milisegundos.
- -C: número de paquetes por segundo que intentará enviar.
- -c: tamaño del paquete a enviar
- -x: nombre del archivo log donde se guardarán los resultados

Enviando tráfico de congestión

1: Desde el directorio */bin* de la carpeta D-ITG se ejecuta el comando *ITGSend* para realizar los envios de paquetes VoIP, para congestionar la red. Un ejemplo del comando se observa en la Figura 3.26.

```
1:  ~/bin/# ./ITGSend -a 10.0.0.1 -rp 8000 -t 999999999 VoIP -x G.711.1 -h CRTP
    -VAD
2:  ~/bin/#
```

Figura 3.26 Ejemplo de generación de tráfico VoIP con D-ITG

Del lado del receptor

1: Desde el directorio */bin* de la carpeta D-ITG se ejecuta el comando *ITGRecv* para recibir tanto el tráfico de medición como el tráfico de congestión. Un ejemplo del comando se observa en la Figura 3.27.

```
1:  ~/bin/# ./ITGRecv
    Press Ctrl-C to terminate
2:  ~/bin/#
```

Figura 3.27 Ejemplo de recepción de paquetes

Para visualizar los resultados

1: Desde el directorio */bin* de la carpeta D-ITG se ejecuta el comando *ITGDec* para decodificar los archivos en donde se guardaron los resultados de la transmisión de paquetes. El archivo a decodificar se encuentra en el directorio */bin*. Un ejemplo del comando se observa en la Figura 3.28.

```
1:  ~/bin/# ./ITGDec e3st64-1
    Flow number: 1
    From 10.0.0.4:34771
    To 10.0.0.3:9501
    ***************** TOTAL RESULTS ******************
    Number of flows = 1
    Total time = 300.001837 s
    Total packets = 10000
    Minimum delay = 0.445701 s
    Maximum delay = 33.464808 s
    Average delay = 13.449749 s
    Average jitter = 0.000706 s
    Delay standard deviation = 0.036939 s
    Bytes received = 7498028
    Average bitrate = 900.320692 Kbps
    Average packet rate = 399.816334 pkt/s
    Packets dropped = 0 ( 0 %)
    Error lines = 0
2:  ~/bin/#
```

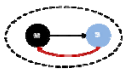
Figura 3.28 Ejemplo del decodificador de resultados de desempeño

4. RESULTADOS Y ANÁLISIS DE PRUEBAS

Este Capítulo muestra los resultados obtenidos luego de realizar las pruebas descritas en el Capítulo 3, sobre los escenarios descritos en ese capítulo. Estos resultados se presentan en forma de gráficas y tablas; haciendo énfasis en los parámetros o métricas ya mencionados en el Capitulo 3. Adicionalmente se presenta un análisis de cada uno de los resultados.

Los valores graficados poseen una leyenda por eje de coordenadas; siendo el común denominador para el eje X, tamaño de paquetes en bytes. Se presentaran las graficas en parejas sin tráfico y otra inyectando tráfico.

4.1 Escenario 1 - Resultados



A continuación se presentan los resultados de las pruebas realizadas sobre el E1 (Figura 3.2). La Tabla 4.1 muestra los valores promedios de las pruebas realizadas. Como se mencionó en el Capítulo 3; este escenario es considerado como base, ya que teóricamente debe presentar las mejores prestaciones.

	SIN TRÁFICO				CON TRÁFICO			
	64	256	1024	1518	64	256	1024	1518
Total de Paquetes recibidos	107103,7143	99199,8333	36490,1667	22654,6667	98089,3333	71584,1667	30359,1667	20528,5000
Avg. Jitter (ms)	0,0055	0,0060	0,0161	0,0256	0,0060	0,0083	0,0195	0,0285
Bitrate (Kb/s)	181,7347	674,8138	992,9756	913,7141	166,3026	486,0168	825,8226	828,1568
Packet rate (Paq/s)	354,9507	329,4989	121,2128	75,2400	324,8097	237,3129	100,8084	68,1947
Packets dropped (%)	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

Tabla 4.1 Resultados de pruebas de desempeño en el escenario 1

La Figura 4.1 (A) muestra el *Packet rate* por segundo en E1 con un ambiente sin tráfico; mientras que la Figura 4.1 (B) muestra el *Packet rate* por segundo en E1 con un ambiente con tráfico. Se observó un decremento en *Packet rate* en el ambiente con tráfico; con un promedio aproximado de 16%.

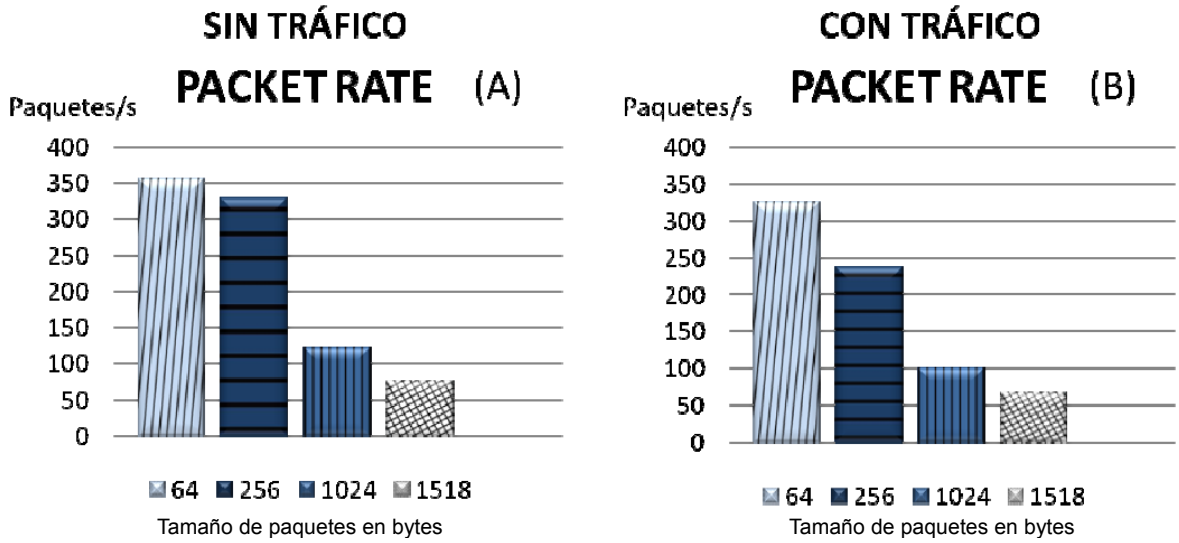


Figura 4.1 Gráficas comparativas de la tasa de paquetes por segundo escenario 1

La Figura 4.2 (A) muestra el *Avg Jitter* en E1 con un ambiente sin tráfico; mientras que la Figura 4.2 (B) muestra el *Avg Jitter* en E1 con un ambiente con tráfico. Se observó un incremento del *Avg Jitter* en el ambiente con tráfico; con un promedio aproximado de 20%.

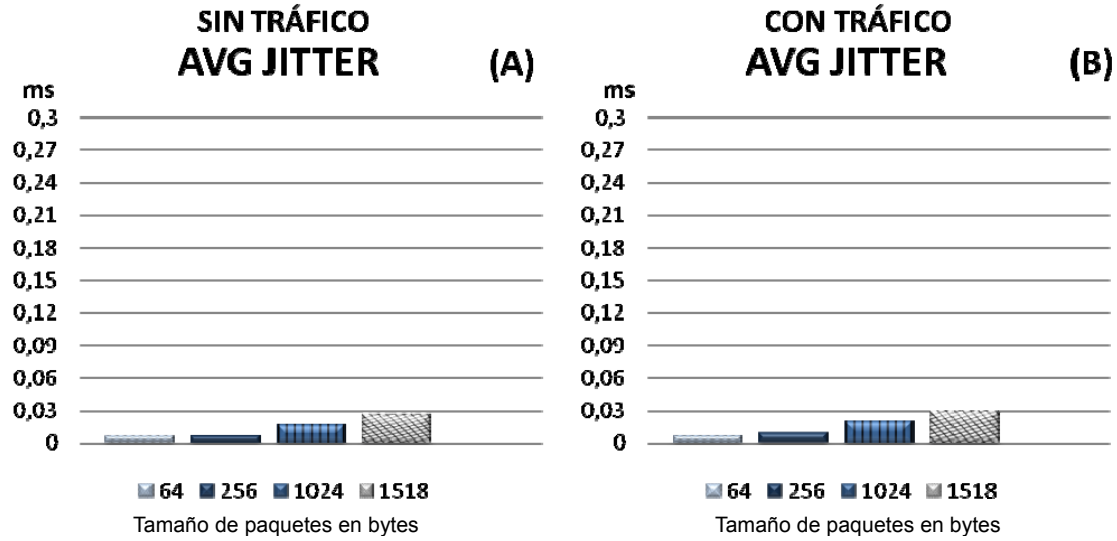


Figura 4.2 Gráficas comparativas del promedio del Jitter escenario 1

La Figura 4.3 (A) muestra el *Packets dropped* en E1 con un ambiente sin tráfico; mientras que la Figura 4.3 (B) muestra el *Packets dropped* en E1 con un ambiente con tráfico. No se observaron paquetes descartados.

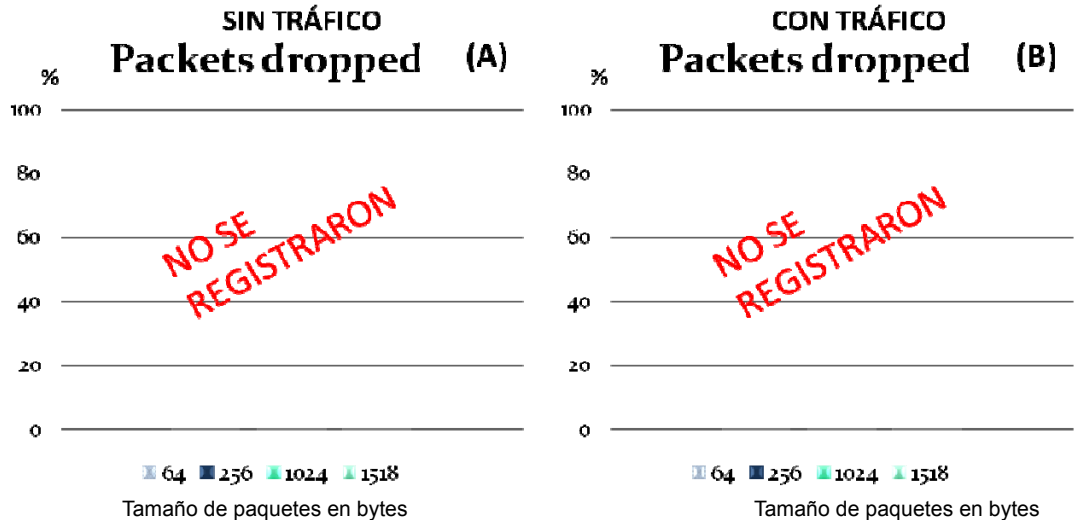
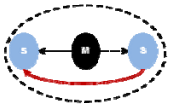


Figura 4.3 Gráficas comparativas de los paquetes descartados escenario 1

Esa disminución observada entre los escenarios con un ambiente sin tráfico y una ambiente con tráfico, se debe básicamente a la congestión generada por el tráfico inyectado por el dispositivo M.

4.2 Escenario 2 - Resultados



A continuación se presentan los resultados de las pruebas realizadas sobre E2 (Figura 3.3). La Tabla 4.2 muestra los valores promedios de las pruebas realizadas. En este escenario (E2) se observó un decremento en el desempeño global con respecto a E1. Esto puede atribuirse a la presencia del nodo intermedio o puente, lo que causa que la comunicación no sea directa.

	SIN TRÁFICO				CON TRÁFICO			
	64	256	1024	1518	64	256	1024	1518
Total de Paquetes recibidos	57057,3333	54398,5000	25629,0000	17004,3333	49647,8333	46054,5000	21670,8333	14589,5000
Avg. Jitter (ms)	0,0103	0,0109	0,0230	0,0343	0,0119	0,0129	0,0272	0,0401
Bitrate (Kb/s)	96,3142	369,0344	696,7083	684,8795	83,7223	311,8060	588,4548	587,2681
Packet rate (Paq/s)	188,1137	180,1926	85,0474	56,3965	163,5202	152,2490	71,8329	48,3587
Packets dropped (%)	46,8750	0,0000	0,0000	0,0000	63,8583	18,6983	0,0000	0,0000

Tabla 4.2 Resultados de pruebas de desempeño en el escenario 2

La Figura 4.4 (A) muestra el *Packet rate* por segundo en E2 con un ambiente sin tráfico; mientras que la Figura 4.4 (B) muestra el *Packet rate* por segundo en E2 con un ambiente con tráfico. Se observó un decremento en *Packet rate* en el ambiente con tráfico; con un promedio aproximado de 16%. Luego de comparar los resultados obtenidos en E1 con E2, se observó un decremento general en *Packet rate* con un promedio aproximado de 36%.

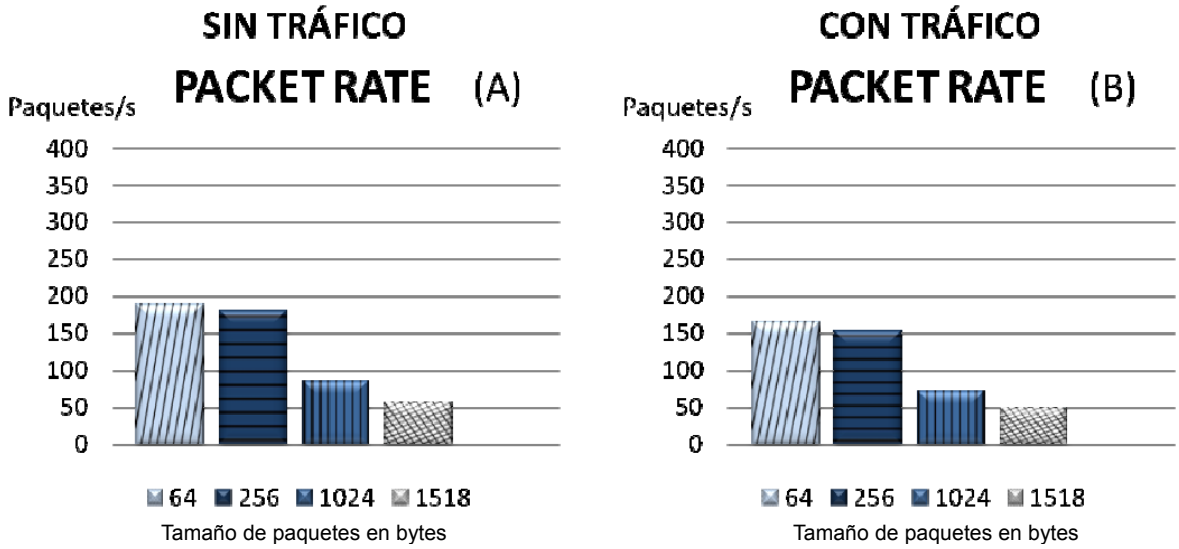


Figura 4.4 Gráficas comparativas de la tasa de paquetes por segundo escenario 2

La Figura 4.5 (A) muestra el *Avg Jitter* en E2 con un ambiente sin tráfico, mientras que la Figura 4.5 (B) muestra el *Avg Jitter* en E2 con un ambiente con tráfico. Se observó un incremento del *Avg Jitter* en el ambiente con tráfico; con un promedio aproximado de 17%. Luego de comparar los resultados obtenidos en E1 con E2, se observó un incremento general en *Avg Jitter* con un promedio aproximado de 60% manteniendo la tendencia de E1; lo cual indica un retraso considerable en la entrega de paquetes.

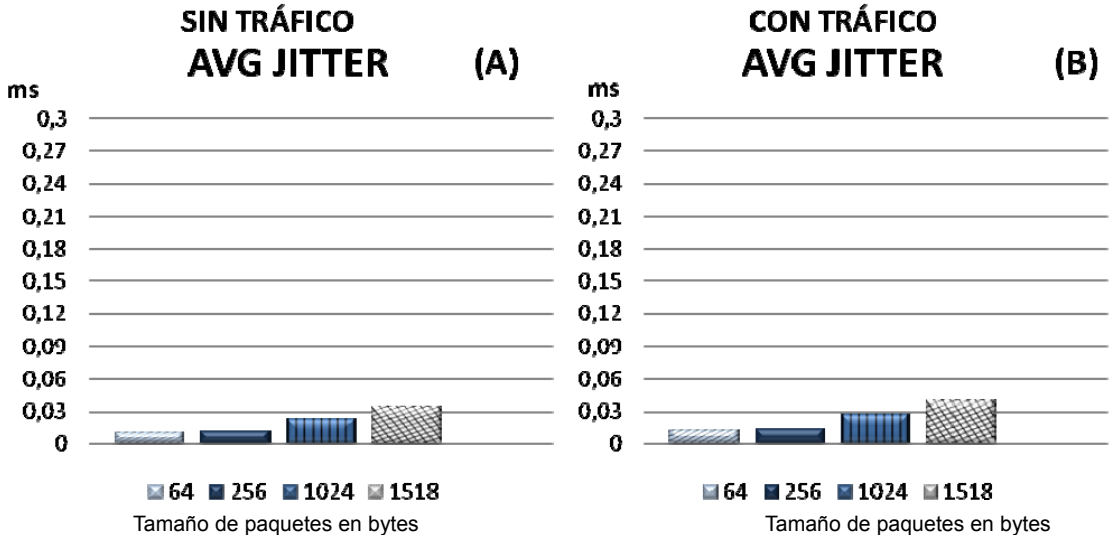


Figura 4.5 Gráficas comparativas del promedio del Jitter escenario 2

La Figura 4.6 (A) muestra el *Packets dropped* en E2 con un ambiente sin tráfico; mientras que la Figura 4.6 (B) muestra el *Packets dropped* en E2 con un ambiente con tráfico. Se observó en el ambiente sin tráfico descarte de paquetes

con tamaño de paquetes de 64 bytes; mientras que en el ambiente con tráfico se observó este mismo fenómeno con tamaños de paquetes de 64 y 256 bytes respectivamente, este descarte en los casos de paquetes de 64 y 256 bytes se podría deber a que debido al tamaño reducido del mismo en comparación con los de 1024 y 1518 bytes, se podrían generar un mayor flujo de paquetes el cual congestionaría en un mayor grado al nodo intermedio, el cual no tendría la capacidad de manejarlo.

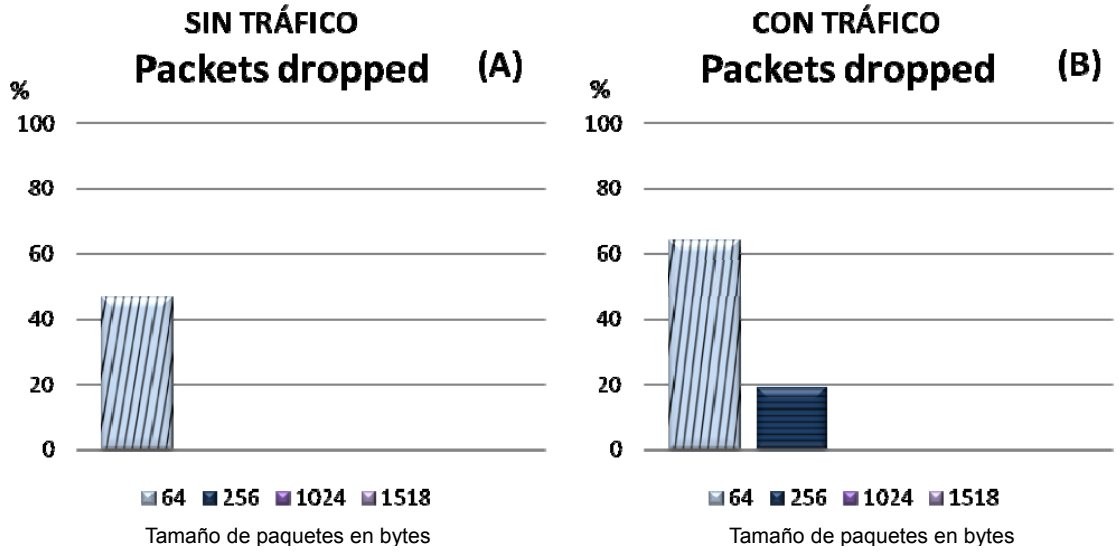
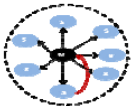


Figura 4.6 Gráficas comparativas de los paquetes descartados escenario 2

Este comportamiento podría apuntar a un saturación en las estructuras de datos de recepción de paquetes en el nodo intermedio, lo cual señala un primer indicio del efecto negativo que causa en la comunicación los nodos intermedios o puentes.

4.3 Escenario 3 - Resultados



A continuación se presentan los resultados de las pruebas realizadas sobre E3 (Figura 3.4). La Tabla 4.3 muestra los valores promedios de las pruebas realizadas. En este escenario (E3) se observó un ligero decremento en el desempeño en el ambiente sin tráfico con respecto a E1; sin embargo, en el ambiente con tráfico el decremento fue mayor. Esto se atribuye a la presencia de un mayor número de nodos en la *Piconet*, lo cual causa un alto grado de stress en el nodo M al ser el único punto de convergencia del tráfico.

	SIN TRÁFICO				CON TRÁFICO			
	64	256	1024	1518	64	256	1024	1518
Total de Paquetes recibidos	90876,3333	85201,5000	27078,5000	16190,8333	37214,1667	36190,3333	15101,0000	8707,0000
Avg. Jitter (ms)	0,0065	0,0070	0,0218	0,0361	0,0157	0,0164	0,0391	0,0673
Bitrate (Kb/s)	154,0753	579,2659	736,2272	652,5714	62,5389	244,6937	409,2621	349,3487
Packet rate (Paq/s)	300,9284	282,8447	89,8715	53,7361	122,1462	119,4794	49,9587	28,7672
Packets dropped (%)	0,0000	0,0000	0,0000	0,0000	73,6767	48,7067	0,0000	0,0000

Tabla 4.3 Resultados de pruebas de desempeño en el escenario 3

La Figura 4.7 (A) muestra el *Packet rate* por segundo en E3 con un ambiente sin tráfico; mientras que la Figura 4.7 (B) muestra el *Packet rate* por segundo en E3 con un ambiente con tráfico. Se observó un decremento en *Packet rate* en el ambiente con tráfico; con un promedio aproximado de 54%. Luego de comparar los resultados obtenidos en E1 con E3, se observó un decremento en el ambiente sin tráfico de 21%, mientras que en el ambiente con tráfico el decremento observado fue de 55%.

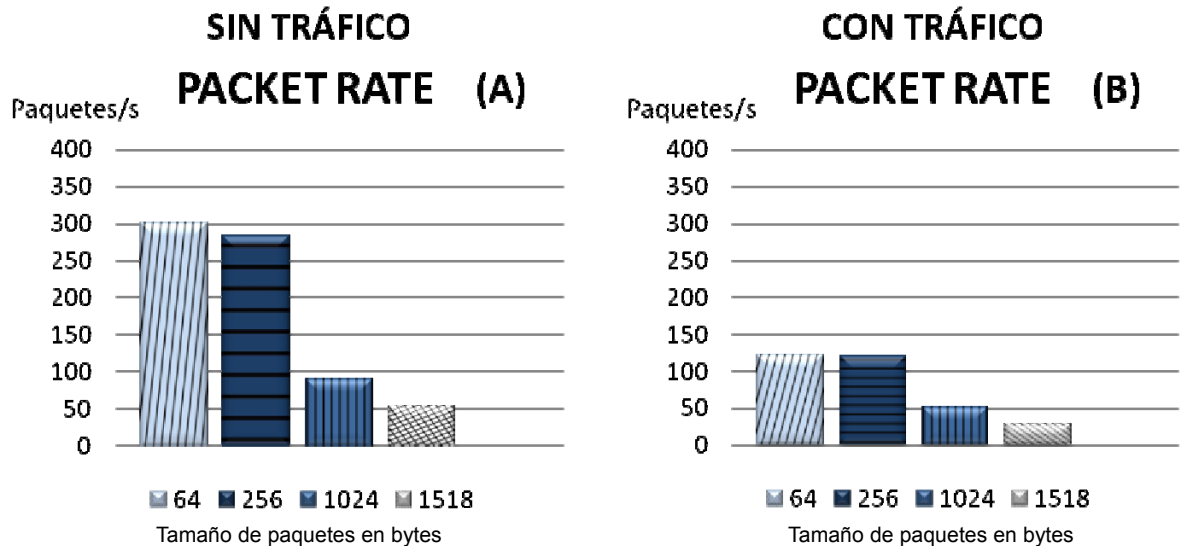


Figura 4.7 Gráficas comparativas de la tasa de paquetes por segundo escenario 3

La Figura 4.8 (A) muestra el *Avg Jitter* en E3 con un ambiente sin tráfico, mientras que la Figura 4.8 (B) muestra el *Avg Jitter* en E3 con un ambiente con tráfico. Se observó un incremento del *Avg Jitter* en el ambiente con tráfico; con un promedio aproximado de 111%. Luego de comparar los resultados obtenidos en E1 con E3, se observó un incremento en *Avg Jitter* en el ambiente sin tráfico, con un promedio aproximado de 28%, mientras que en el ambiente con tráfico el incremento fue de 124%; lo cual indica un retraso considerable en la entrega de paquetes en un ambiente con tráfico.

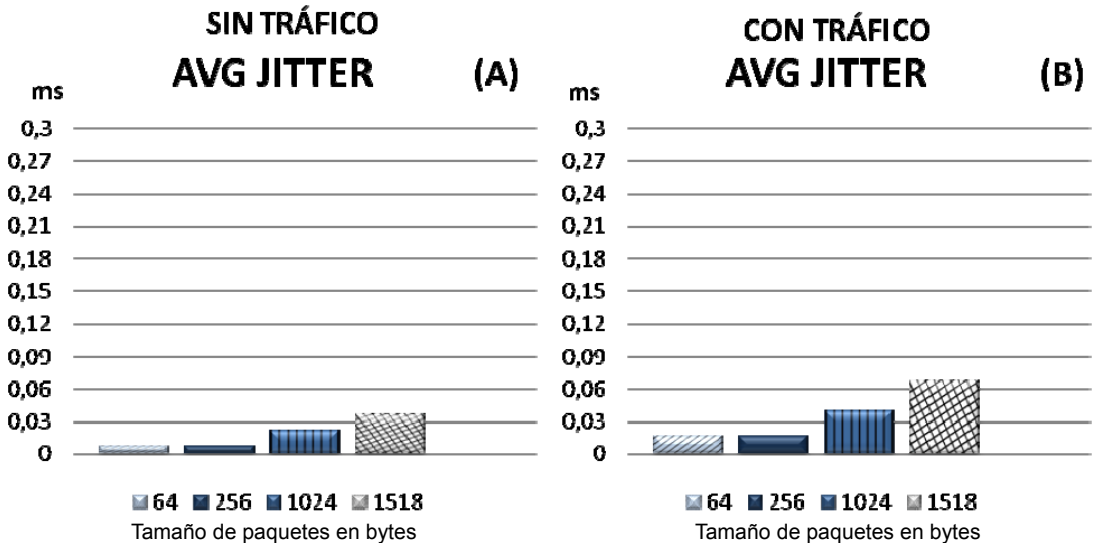


Figura 4.8 Gráficas comparativas promedio del Jitter escenario 3

La Figura 4.9 (A) muestra el *Packets dropped* en E3 con un ambiente sin tráfico; mientras que la Figura 4.9 (B) muestra el *Packets dropped* en E3 con un ambiente con tráfico. No se observaron paquetes descartados en el ambiente sin tráfico, mientras en el ambiente con tráfico, se observó descarte de paquetes con tamaño de paquetes de 64 y 256 bytes.

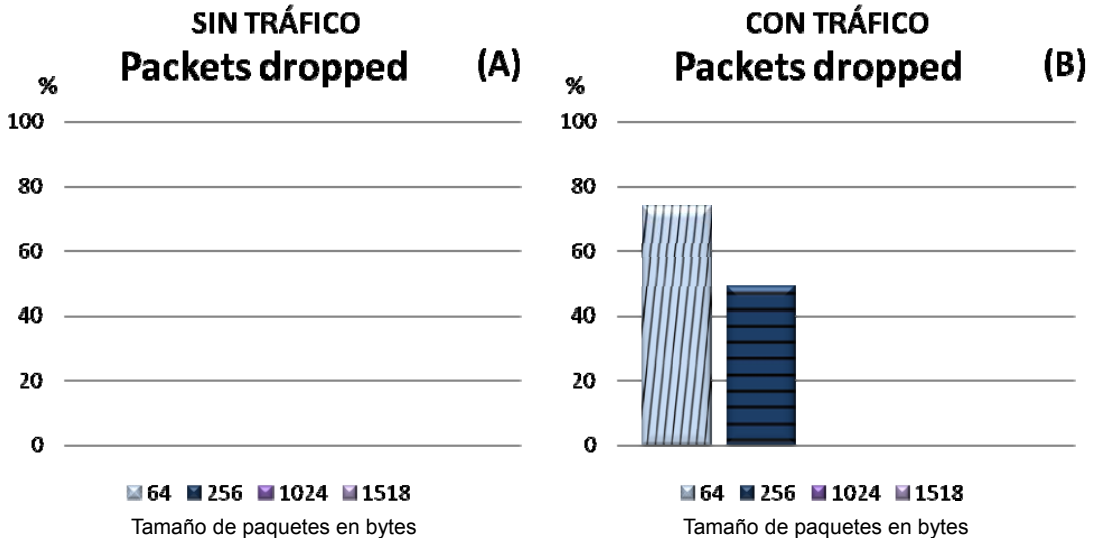
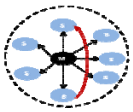


Figura 4.9 Gráficas comparativas de los paquetes descartados escenario 3

Las diferencias observadas en los ambientes con y sin tráfico, permite deducir que el hecho de que una *Piconet* se encuentre poblada; no es el factor determinante en la merma del desempeño de la comunicación; más si lo es el

hecho de que los miembros de la *Piconet* transmitan información, la cual converge en el *maestro* causando un cuello de botella.

4.4 Escenario 4 - Resultados



A continuación se presentan los resultados de las pruebas realizadas sobre E4 (Figura 3.5). La Tabla 4.4 muestra los valores promedios de las pruebas realizadas. En este escenario (E2) se observó un decremento considerable en el desempeño global con respecto a E1. Esto puede atribuirse a los factores identificados en E2 y E3; la presencia del nodo intermedio o puente, y una *Piconet* totalmente poblada.

	SIN TRÁFICO				CON TRÁFICO			
	64	256	1024	1518	64	256	1024	1518
Total de Paquetes recibidos	49283,8333	45192,6667	16193,1667	9910,8333	29477,5000	28719,6667	11852,3333	7538,1667
Avg. Jitter (ms)	0,0120	0,0131	0,0365	0,0591	0,0201	0,0205	0,0498	0,0793
Bitrate (Kb/s)	83,1044	306,1425	438,8891	397,8111	48,1255	191,4041	317,3302	290,3079
Packet rate (Paq/s)	162,3133	149,4836	53,5753	32,7578	93,9951	93,4591	38,7366	23,9055
Packets dropped (%)	52,0267	20,9933	0,0000	0,0000	74,7417	78,6817	91,3983	93,4983

Tabla 4.4 Resultados de pruebas de desempeño en el escenario 4

La Figura 4.10 (A) muestra el *Packet rate* por segundo en E4 con un ambiente sin tráfico; mientras que la Figura 4.10 (B) muestra el *Packet rate* por segundo en E4 con un ambiente con tráfico. Se observó un decremento en *Packet rate* en el ambiente con tráfico; con un promedio aproximado de 34%. Luego de comparar los resultados obtenidos en E1 con E4, se observó un decremento en el ambiente sin tráfico de 55%, mientras que en el ambiente con tráfico el decremento fue de 65%.

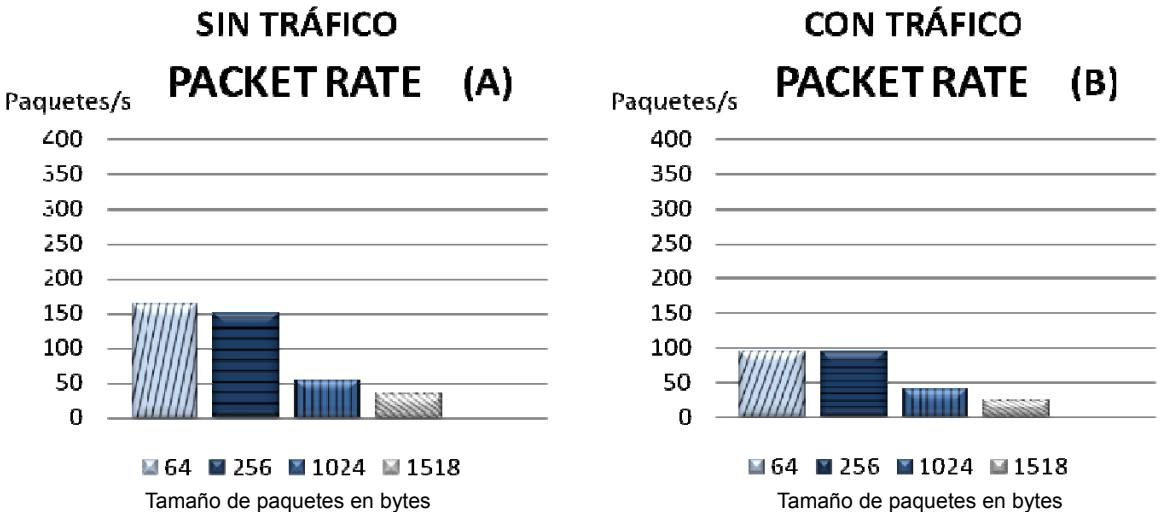


Figura 4.10 Gráficas comparativas de la tasa de paquetes por segundo escenario 4

La Figura 4.11 (A) muestra el *Avg Jitter* en E4 con un ambiente sin tráfico, mientras que la Figura 4.11 (B) muestra el *Avg Jitter* en E4 con un ambiente con tráfico. Se observó un incremento del *Avg Jitter* en el ambiente con tráfico; con un promedio aproximado de 49%. Luego de comparar los resultados obtenidos en E1 con E4, se observó un incremento en *Avg Jitter* en el ambiente sin tráfico, con un promedio aproximado de 124%, mientras que en el ambiente con tráfico el incremento fue de 178%; lo cual indica un retraso más significativo en la entrega de paquetes que en E2 y E3.

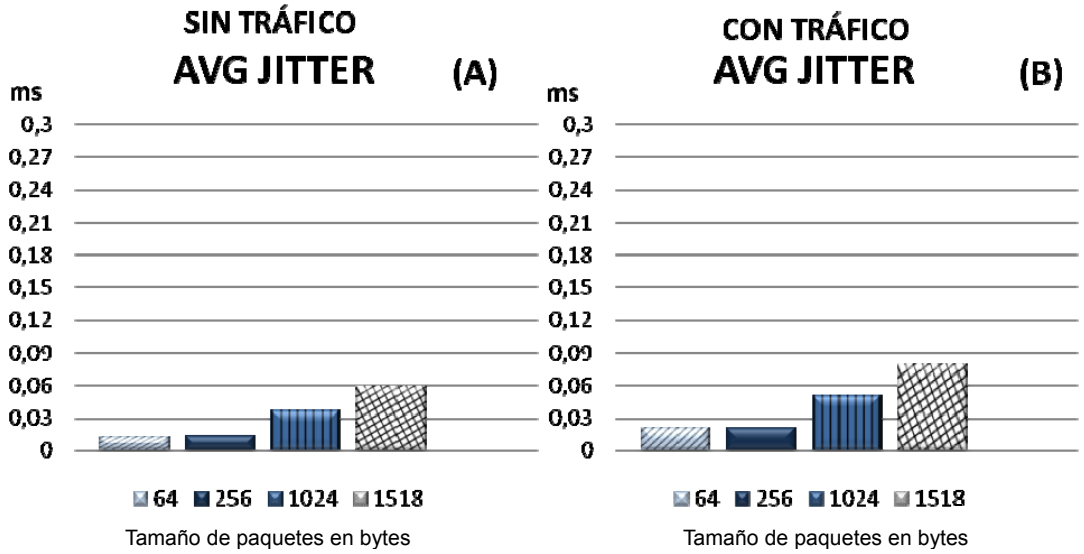


Figura 4.11 Gráficas comparativas del average del Jitter escenario 4

La Figura 4.12 (A) muestra el *Packets dropped* en E4 con un ambiente sin tráfico; mientras que la Figura 4.12 (B) muestra el *Packets dropped* en E4 con un ambiente con tráfico. Se observó en el ambiente sin tráfico descarte de paquetes con tamaño de paquetes de 64 y 256 bytes; mientras que en el ambiente con tráfico se observó altos porcentajes (aproximadamente 85%) de este fenómeno para todos los tamaños de paquetes. Esto apunta a una saturación en las estructuras de datos de recepción de paquetes en el nodo intermedio, lo cual corrobora el indicio del efecto negativo que causa en la comunicación los nodos intermedios o puentes, junto al efecto de la *Piconet* poblada generando tráfico.

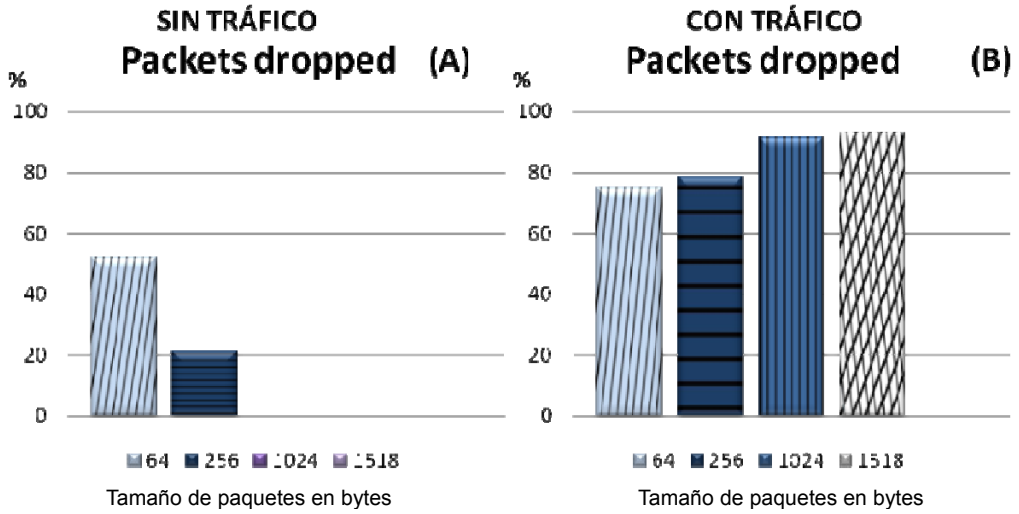
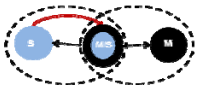


Figura 4.12 Gráficas comparativas de los paquetes descartados escenario 4

Este escenario se vio afectado por los factores identificados en E2 y E3. Se observó en el ambiente sin tráfico, la presencia de un decremento considerable en el desempeño, lo que permite concluir que la existencia de un nodo intermedio o puente es un factor determinante en el desempeño de las comunicaciones.

4.5 Escenario 5 - Resultados



A continuación se presentan los resultados de las pruebas realizadas sobre E5 (Figura 3.6). La Tabla 4.5 muestra los valores promedios de las pruebas realizadas. En este escenario se inician las pruebas en topología tipo *Scatternet*. En E5 se observó un decremento en el desempeño global de las comunicaciones con respecto a E1; posiblemente por la presencia de un nodo con doble función (M/S), lo que conlleva que dicho nodo realice una serie de procesos para mantener la interconexión de la *Scatternet*.

	SIN TRÁFICO				CON TRÁFICO			
	64	256	1024	1518	64	256	1024	1518
Total de Paquetes recibidos	84788,0000	80395,8333	28022,6667	17092,5000	38740,5000	35020,8333	13993,6667	9116,3333
Avg. Jitter (ms)	0,0070	0,0074	0,0210	0,0342	0,0152	0,0170	0,0423	0,0642
Bitrate (Kb/s)	143,5488	546,4969	762,1450	688,9163	65,0845	236,6395	378,9035	366,0643
Packet rate (Paq/s)	280,3688	266,8442	93,0353	56,7289	127,1182	115,5466	46,2529	30,1436
Packets dropped (%)	0,0000	0,0000	0,0000	0,0000	72,4383	59,2350	0,0000	0,0000

Tabla 4.5 Resultados de pruebas de desempeño en el escenario 5

La Figura 4.13 (A) muestra el *Packet rate* por segundo en E5 con un ambiente sin tráfico; mientras que la Figura 4.13 (B) muestra el *Packet rate* por segundo en E5 con un ambiente con tráfico. Se observó un decremento en el ambiente con tráfico; con un promedio aproximado de 52%. Luego de comparar

los resultados obtenidos en E1 con E5, se observó un decremento en el ambiente sin tráfico de 22%, mientras que en el ambiente con tráfico el decremento fue de 56%.

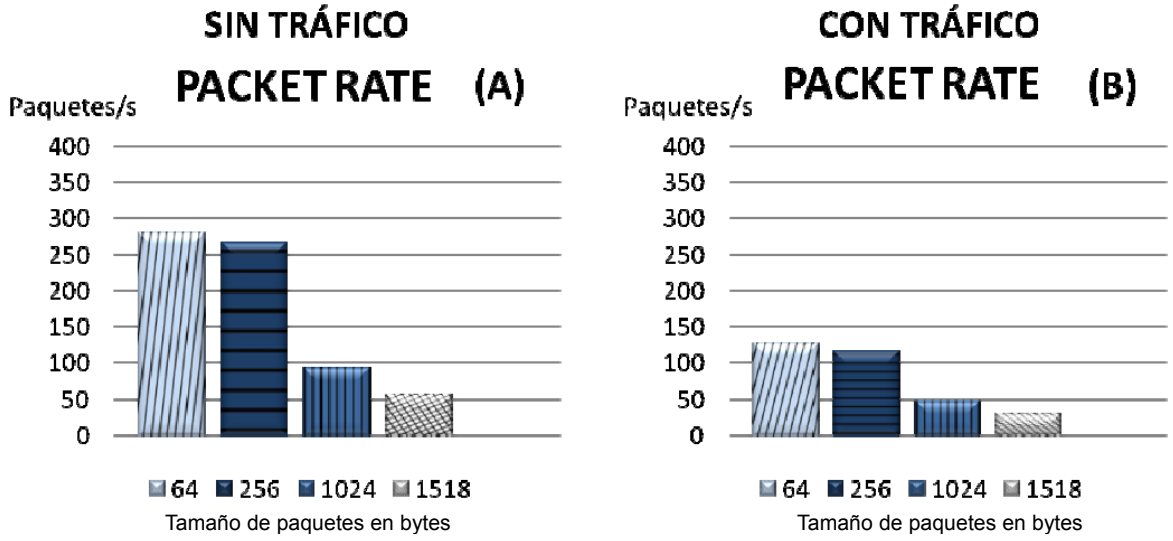


Figura 4.13 Gráficas comparativas de la tasa de paquetes por segundo escenario 5

La Figura 4.14 (A) muestra el *Avg Jitter* en E5 con un ambiente sin tráfico, mientras que la Figura 4.14 (B) muestra el *Avg Jitter* en E5 con un ambiente con tráfico. Se observó un incremento del *Avg Jitter* en el ambiente con tráfico; con un promedio aproximado de 109%. Luego de comparar los resultados obtenidos en E1 con E5, se observó un incremento en *Avg Jitter* en el ambiente sin tráfico, con un promedio aproximado de 29%, mientras que en el ambiente con tráfico el incremento fue de 125%.

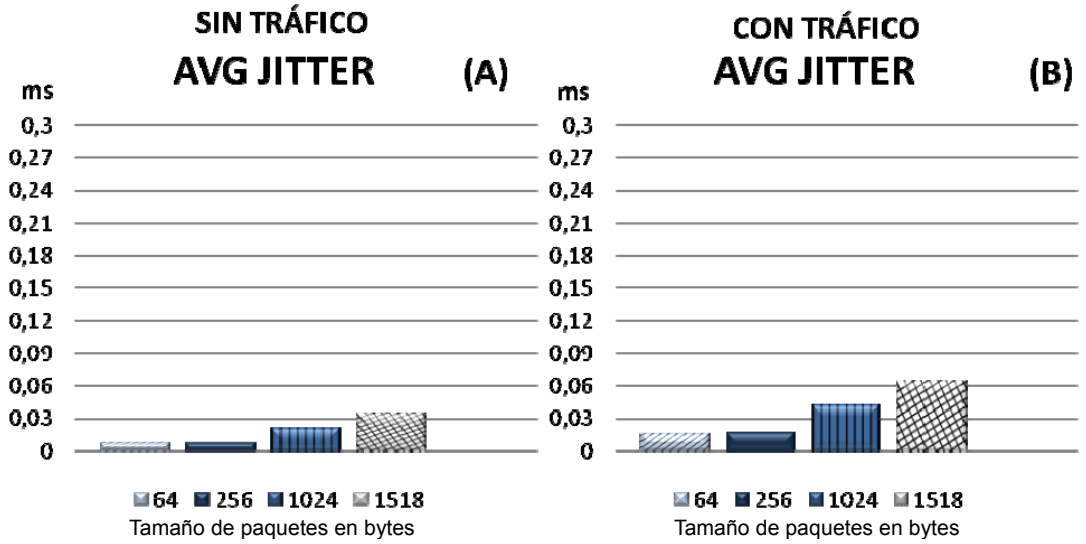


Figura 4.14 Gráficas comparativas del promedio del Jitter escenario 5

La Figura 4.15 (A) muestra el *Packets dropped* en E5 con un ambiente sin tráfico; mientras que la Figura 4.15 (B) muestra el *Packets dropped* en E5 con un ambiente con tráfico. No se observaron paquetes descartados en el ambiente sin tráfico, mientras que en el ambiente con tráfico, se observó descarte de los paquetes con tamaño de paquetes de 64 y 256 Bytes

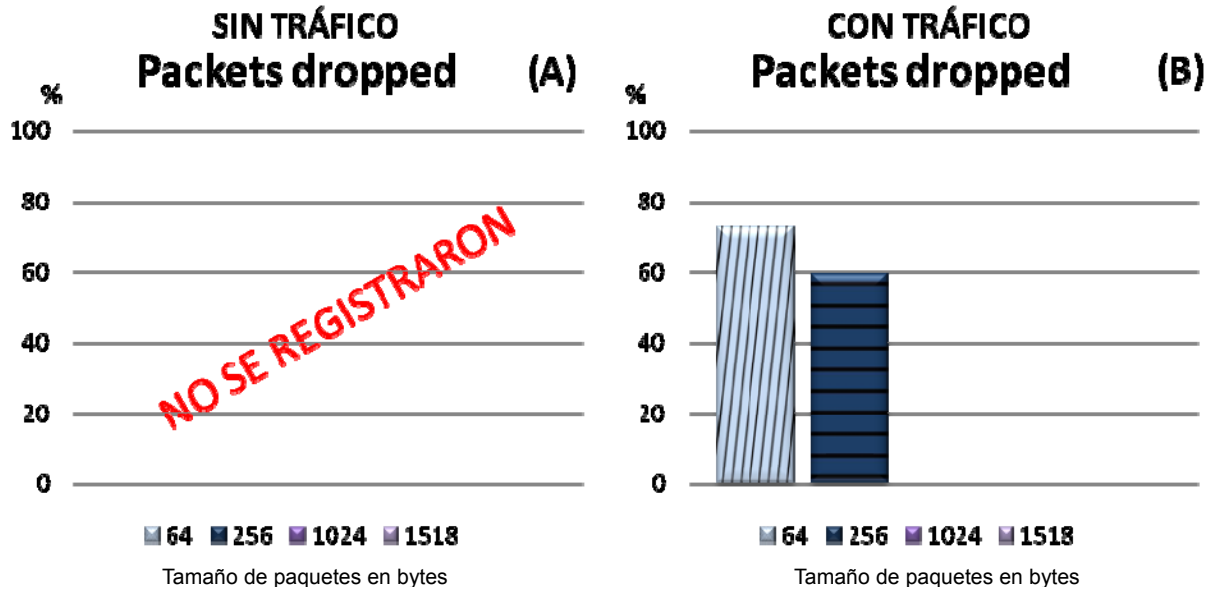
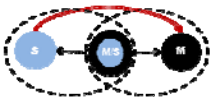


Figura 4.15 Gráficas comparativas de los paquetes descartados escenario 5

En este escenario se observó una disminución en el desempeño de la comunicación, esto debido a que el nodo M/S debe pasar por los procesos de ajustes necesarios para lograr la comunicación en la *Piconet* respectiva. Dadas las características de la tecnología *Bluetooth*, el M/S sólo estará activo en una *Piconet* a la vez; durante ese tiempo no podrá recibir o enviar paquetes en otra *Piconet*, afectando directamente el retardo en la comunicación y finalmente el desempeño.

4.6 Escenario 6 - Resultados



A continuación se presentan los resultados de las pruebas realizadas sobre E6 (Figura 3.7). La Tabla 4.6 muestra los valores promedios de las pruebas realizadas. En E6 se observó un decremento en el desempeño global con respecto a E1. Esto puede atribuirse a la presencia del nodo intermedio o puente con doble función.

	SIN TRÁFICO				CON TRÁFICO			
	64	256	1024	1518	64	256	1024	1518
Total de Paquetes recibidos	46717,6667	29385,1667	9430,8333	6142,0000	22819,8333	21714,5000	9481,6667	5680,6667
Avg. Jitter (ms)	0,0126	0,0201	0,0627	0,0954	0,0259	0,0273	0,0623	0,1032
Bitrate (Kb/s)	78,7265	198,1583	254,7236	245,8023	38,0144	145,9159	256,0011	226,8689
Packet rate (Paq/s)	153,7628	96,7570	31,0942	20,2406	74,2468	71,2480	31,2501	18,6816
Packets dropped (%)	66,0150	79,0583	87,2087	90,5163	83,2233	84,5050	87,7507	95,8433

Tabla 4.6 Resultados de pruebas de desempeño en el escenario 6

La Figura 4.16 (A) muestra el *Packet rate* por segundo en E6 con un ambiente sin tráfico; mientras que la Figura 4.16 (B) muestra el *Packet rate* por segundo en E6 con un ambiente con tráfico. Se observó un decremento en *Packet rate* en el ambiente con tráfico; con un promedio aproximado de 21%. Luego de comparar los resultados obtenidos en E1 con E6, se observó un decremento general en *Packet rate* con un promedio aproximado de 70%.

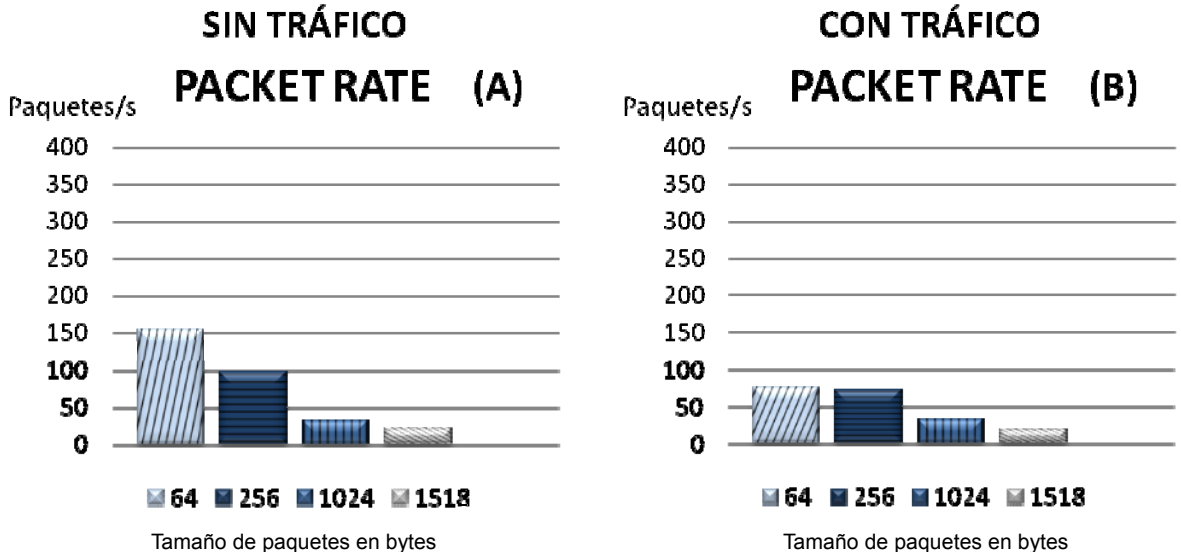


Figura 4.16 Gráficas comparativas de la tasa de paquetes por segundo escenario 6

La Figura 4.17 (A) muestra el *Avg Jitter* en E6 con un ambiente sin tráfico, mientras que la Figura 4.17 (B) muestra el *Avg Jitter* en E6 con un ambiente con tráfico. Se observó un incremento del *Avg Jitter* en el ambiente con tráfico; con un promedio aproximado de 37%. Luego de comparar los resultados obtenidos en E1 con E6, se observó un incremento general en *Avg Jitter* con un promedio aproximado de 246%; lo cual indica una variación del retardo elevada entre paquetes.

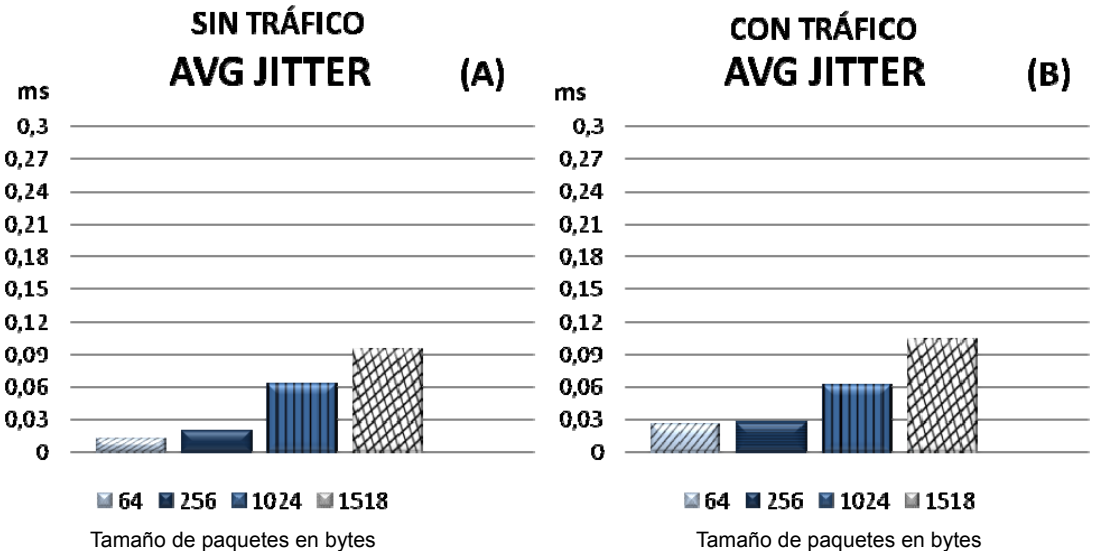


Figura 4.17 Gráficas comparativas del promedio del Jitter escenario 6

La Figura 4.18 (A) muestra el *Packets dropped* en E6 con un ambiente sin tráfico; mientras que la Figura 4.18 (B) muestra el *Packets dropped* en E6 con un ambiente con tráfico. Se observó que por primera vez en ambos ambientes sin tráfico y con tráfico, se descartaron paquetes con todos los tamaños; en mayor medida en el ambiente con tráfico. Este comportamiento apunta a una saturación en las estructuras de datos de recepción de paquetes en el nodo intermedio, aunado al hecho de los procesos necesarios realizados por M/S con el fin de realizar la interconexión entre *Piconets*.

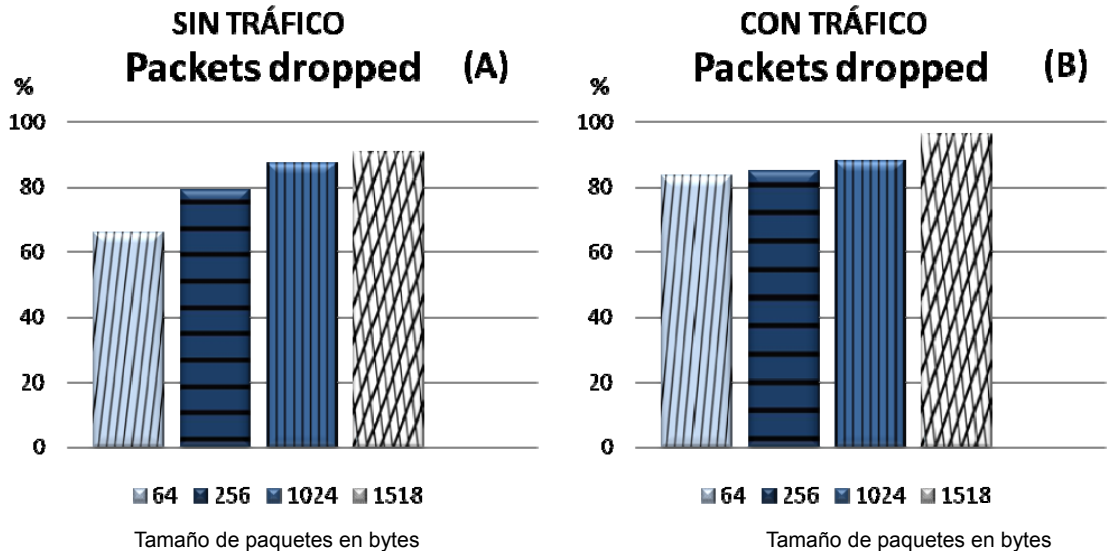
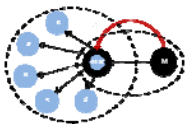


Figura 4.18 Gráficas comparativas de los descartados escenario 6

En este escenario se observaron dos factores que influyen en el decremento del desempeño de la comunicación, la existencia de un nodo intermedio (M/S) el cual además debe cumple un doble rol en la *Scatternet*.

4.7 Escenario 7 - Resultados



A continuación se presentan los resultados de las pruebas realizadas sobre E7 (Figura 3.8). La Tabla 4.7 muestra los valores promedios de las pruebas realizadas. En este escenario (E7) se observó un decremento en el desempeño global con respecto a E1. Esto puede atribuirse a dos factores: la presencia un nodo con doble función, y la población de una de las *Piconets*.

	SIN TRÁFICO				CON TRÁFICO			
	64	256	1024	1518	64	256	1024	1518
Total de Paquetes recibidos	76815,8333	72981,1667	20378,5000	13791,6667	69369,0000	62319,5000	17087,8333	13556,8333
Avg. Jitter (ms)	0,0077	0,0081	0,0289	0,0424	0,0085	0,0095	0,0347	0,0431
Bitrate (Kb/s)	130,1091	495,7193	553,8525	555,2043	117,3707	423,0028	463,4074	545,8775
Packet rate (Paq/s)	254,1194	242,0505	67,6089	45,7184	229,2397	206,5443	56,5683	44,9504
Packets dropped (%)	7,6733	0,0000	0,0017	0,0000	32,5533	0,0000	0,0000	0,0000

Tabla 4.7 Resultados de pruebas de desempeño en el escenario 7

La Figura 4.19 (A) muestra el *Packet rate* por segundo en E7 con un ambiente sin tráfico; mientras que la Figura 4.19 (B) muestra el *Packet rate* por segundo en E7 con un ambiente con tráfico. Se observó un decremento en *Packet rate* en el ambiente con tráfico; con un promedio aproximado de 11%. Luego de comparar los resultados obtenidos en E1 con E7, se observó un decremento general en *Packet rate* con un promedio aproximado de 32%.

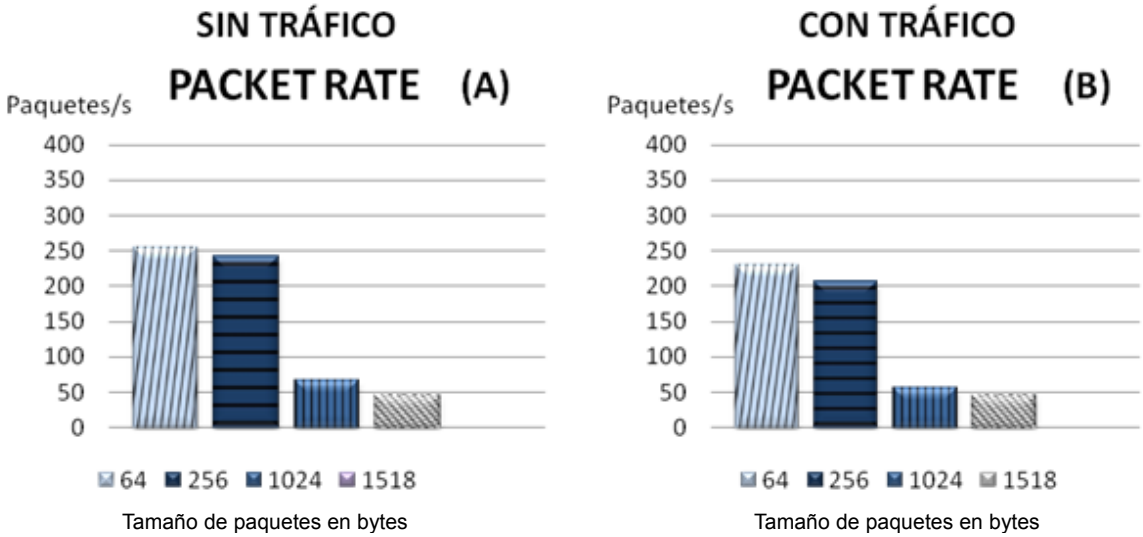


Figura 4.19 Gráficas comparativas de la tasa de paquetes por segundo escenario 7

La Figura 4.20 (A) muestra el *Avg Jitter* en E7 con un ambiente sin tráfico, mientras que la Figura 4.20 (B) muestra el *Avg Jitter* en E7 con un ambiente con tráfico. Se observó un incremento del *Avg Jitter* en el ambiente con tráfico; con un promedio aproximado de 12%. Luego de comparar los resultados obtenidos en E1 con E7, se observó un incremento general en *Avg Jitter* con un promedio aproximado de 51% manteniendo la tendencia de E1; lo cual indica un retraso considerable en la entrega de paquetes.

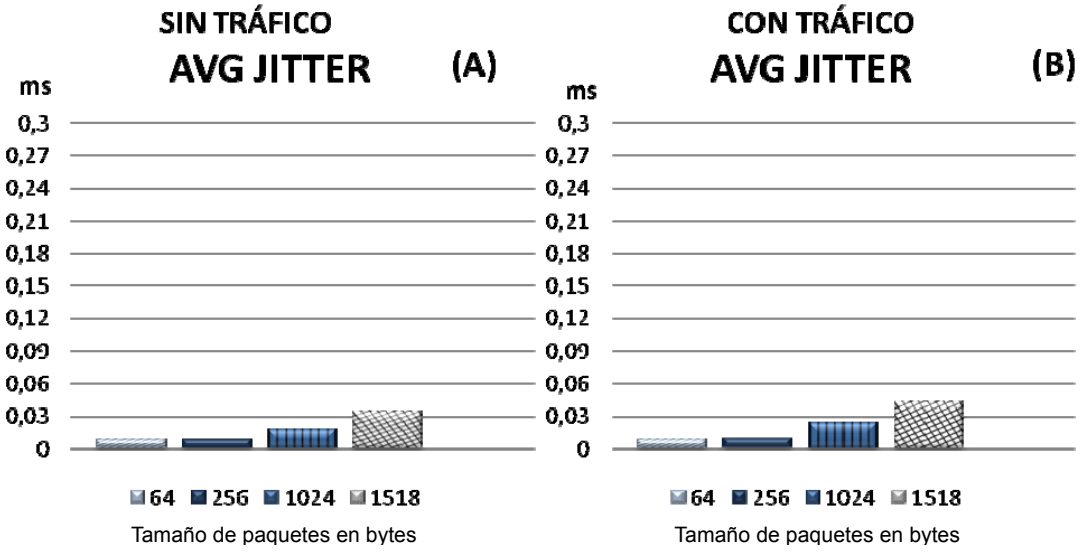


Figura 4.20 Gráficas comparativas del promedio del Jitter escenario 7

La Figura 4.21 (A) muestra el *Packets dropped* en E7 con un ambiente sin tráfico; mientras que la Figura 4.21 (B) muestra el *Packets dropped* en E7 con un ambiente con tráfico. Se observó descarte de paquetes con tamaño de paquetes de 64 bytes; para ambientes con tráfico y sin tráfico.

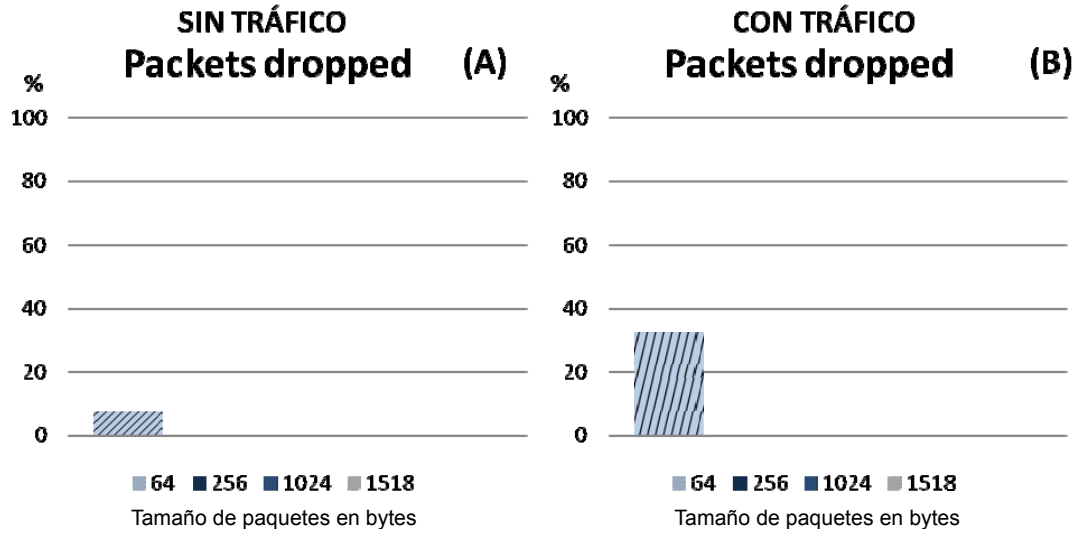
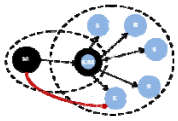


Figura 4.21 Gráficas comparativas de los paquetes descartados escenario 7

En este escenario se observó un mejor desempeño en presencia de tráfico con respecto a E5, a pesar que se encontraba poblada la *Piconet* 2 en E7 (Figura 3.8). De acuerdo a las especificaciones de la tecnología *Bluetooth* un nodo S no posee el control sobre el canal; este debe someterse a las decisiones y prioridades de su nodo M respectivo [2]. El comportamiento descrito anteriormente produce un decremento en el desempeño, cuando la comunicación es realizada como en E5 (desde un nodo S a un nodo M/S), del mismo modo se concluye que al realizar una comunicación como en E7 (desde un M a un M/S) es posible obtener mejores prestaciones.

Con base a los resultados anteriores, se puede concluir que el desempeño de la comunicación es incrementado, cuando un *maestro* inicia dicho proceso. Este aumento en el desempeño se debe a que el nodo M es el responsable del control y las prioridades del canal.

4.8 Escenario 8 - Resultados



A continuación se presentan los resultados de las pruebas realizadas sobre E8 (Figura 3.9). La Tabla 4.8 muestra los valores promedios de las pruebas realizadas. En este escenario (E8) se observó un decremento en el desempeño global con respecto a E1. Esto puede atribuirse a la presencia de tres factores que podrían afectar el desempeño, la presencia de un nodo con doble función, el salto sobre dicho nodo y la población de una de las *Piconets*.

	SIN TRÁFICO				CON TRÁFICO			
	64	256	1024	1518	64	256	1024	1518
Total de Paquetes recibidos	17590,1667	13680,5000	7571,5000	4110,0000	10063,1667	11035,1667	4980,8333	3433,0000
Avg. Jitter (ms)	0,0338	0,0410	0,0795	0,1472	0,0585	0,0548	0,1209	0,1764
Bitrate (Kb/s)	28,6768	92,4691	185,0839	147,7938	15,8916	70,9299	128,6172	127,6901
Packet rate (Paq/s)	56,0094	45,1509	22,5932	12,1701	31,0382	34,6338	15,7003	10,5147
Packets dropped (%)	87,1217	89,4733	93,7817	96,4183	91,4983	91,8600	96,3950	97,3683

Tabla 4.8 Resultados de pruebas de desempeño en el escenario 8

La Figura 4.22 (A) muestra el *Packet rate* por segundo en E8 con un ambiente sin tráfico; mientras que la Figura 4.22 (B) muestra el *Packet rate* por segundo en E8 con un ambiente con tráfico. Se observó un decremento en *Packet rate* en el ambiente con tráfico; con un promedio aproximado de 28%. Luego de comparar los resultados obtenidos en E1 con E8, se observó un decremento en general *Packet rate* con un promedio aproximado de 85%.

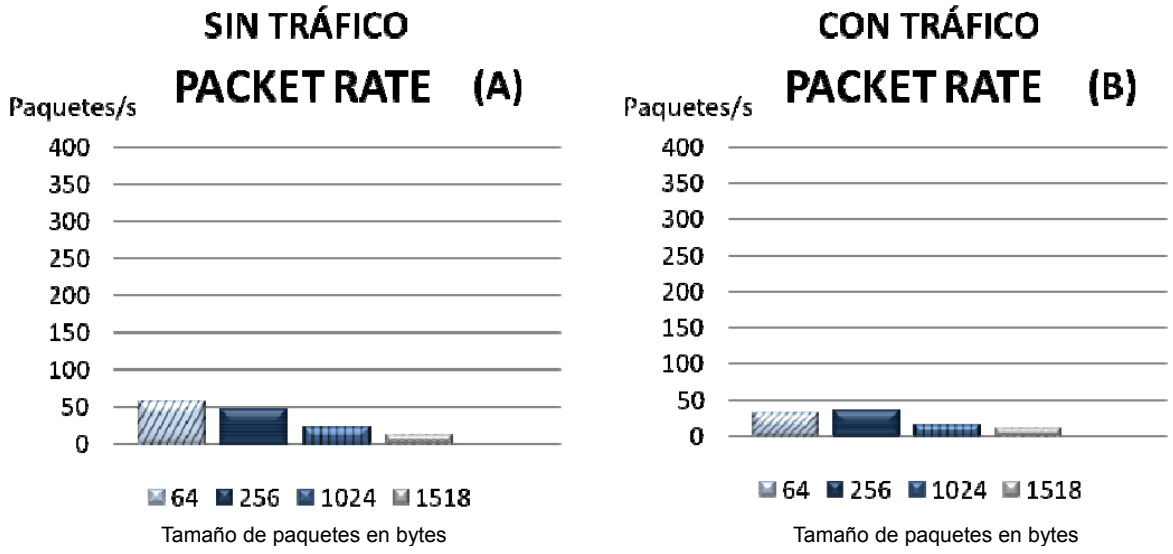


Figura 4.22 Gráficas comparativas de la tasa de paquetes por segundo escenario 8

La Figura 4.23 (A) muestra el *Avg Jitter* en E8 con un ambiente sin tráfico, mientras que la Figura 4.23 (B) muestra el *Avg Jitter* en E8 con un ambiente con tráfico. Se observó un incremento del *Avg Jitter* en el ambiente con tráfico; con un promedio aproximado de 45%. Luego de comparar los resultados obtenidos en E1 con E8, se observó un incremento en *Avg Jitter* en el ambiente sin tráfico, con un promedio aproximado de 492%, mientras que en el ambiente con tráfico el incremento fue de 617%. También se observó con respecto al E2, en donde también existió un salto o nodo intermedio que el incremento también fue alto, sin tráfico el incremento fue de 269% y con tráfico de 350%.

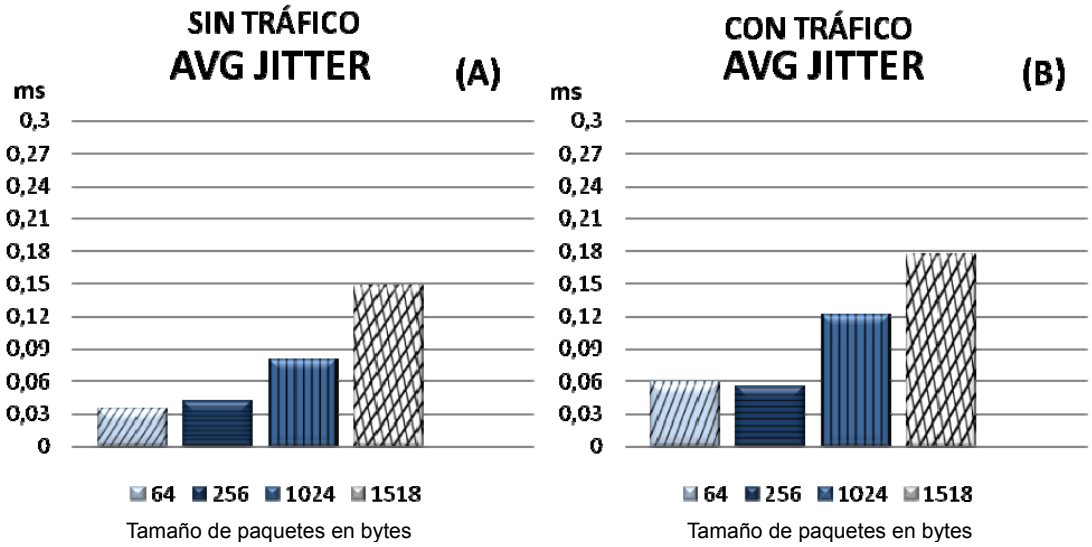


Figura 4.23 Gráficas comparativas del promedio del Jitter escenario 8

La Figura 4.24 (A) muestra el *Packets dropped* en E8 con un ambiente sin tráfico; mientras que la Figura 4.24 (B) muestra el *Packets dropped* en E8 con un ambiente con tráfico. Se observó que en ambos ambientes sin tráfico y con tráfico, se descartaron paquetes con todos los tamaños; alcanzando índices porcentuales muy elevados, todos superiores a 90%.

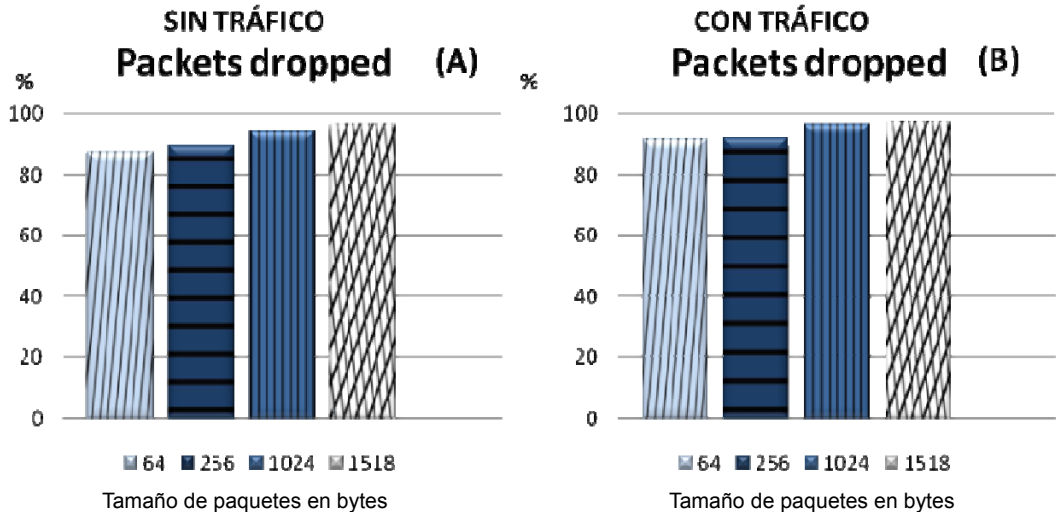
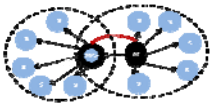


Figura 4.24 Gráficas comparativas de los paquetes descartados escenario 8

En este escenario se observó una disminución en el desempeño de la comunicación, esto debido a que el nodo M/S debe pasar por los procesos de ajustes necesarios para lograr la comunicación en la *Piconet* respectiva. Dadas las características de la tecnología *Bluetooth*, el M/S sólo estará activo en una *Piconet* a la vez; durante ese tiempo no podrá recibir o enviar paquetes en otra *Piconet*. Adicionalmente, una de las *Piconets* se encuentra poblada, afectando directamente el retardo en la comunicación y finalmente el desempeño.

4.9 Escenario 9 - Resultados



A continuación se presentan los resultados de las pruebas realizadas sobre E9 (Figura 3.10). La Tabla 4.9 muestra los valores promedios de las pruebas realizadas. En este escenario (E9) se observó un decremento en el desempeño global con respecto a E1. Esto puede atribuirse a dos factores que podrían afectar el desempeño: la presencia del nodo con doble función y la población ambas *Piconets*.

	SIN TRÁFICO				CON TRÁFICO			
	64	256	1024	1518	64	256	1024	1518
Total de Paquetes recibidos	65205,1667	62215,3333	18960,3333	11678,0000	22397,5000	21889,8333	8024,5000	4414,6667
Avg. Jitter (ms)	0,0091	0,0095	0,0312	0,0502	0,0264	0,0271	0,0736	0,1331
Bitrate (Kb/s)	110,2775	422,3967	514,6227	469,6569	37,1442	146,8980	215,8722	175,6535
Packet rate (Paq/s)	215,3857	206,2484	62,8202	38,6740	72,5472	71,7275	26,3516	14,4642
Packets dropped (%)	34,1133	0,0000	0,0000	0,0000	83,2067	84,2900	93,4467	96,8200

Tabla 4.9 Resultados de pruebas de desempeño en el escenario 9

La Figura 4.25 (A) muestra el *Packet rate* por segundo en E9 con un ambiente sin tráfico; mientras que la Figura 4.25 (B) muestra el *Packet rate* por segundo en E9 con un ambiente con tráfico. Se observó un decremento en *Packet rate* en el ambiente con tráfico, con un promedio aproximado de 63%. Luego de comparar los resultados obtenidos en E1 con E9, se observó un decremento general en *Packet rate* en el ambiente sin tráfico, con un promedio aproximado de 43%, mientras que en el ambiente con tráfico el decremento fue de 75%.

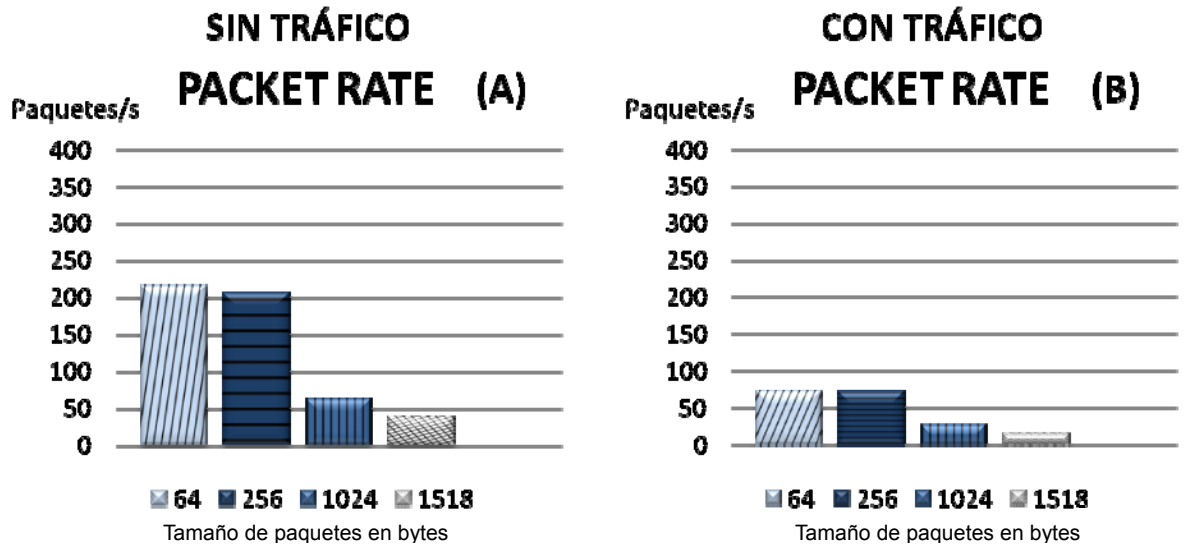


Figura 4.25 Gráficas comparativas de la tasa de paquetes por segundo escenario 9

La Figura 4.26 (A) muestra el *Avg Jitter* en E9 con un ambiente sin tráfico, mientras que la Figura 4.26 (B) muestra el *Avg Jitter* en E9 con un ambiente con tráfico. Se observó un incremento del *Avg Jitter* en el ambiente con tráfico; con un promedio aproximado de 63%. Luego de comparar los resultados obtenidos en E1 con E9, se observó un incremento en *Avg Jitter* en el ambiente sin tráfico, con un promedio aproximado de 79%, mientras que en el ambiente con tráfico el incremento notablemente superior de 302%.

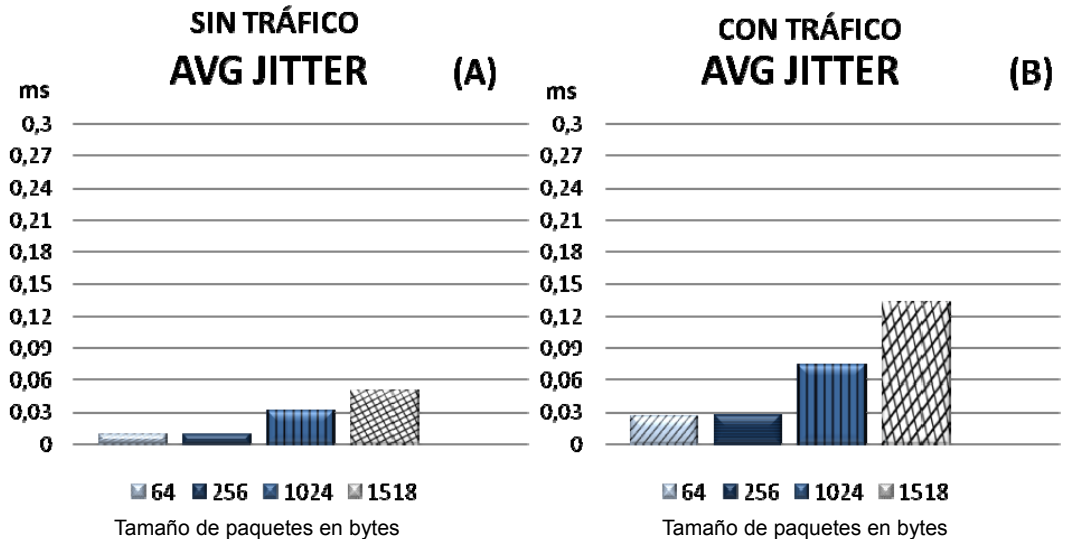


Figura 4.26 Gráficas comparativas del promedio del Jitter escenario 9

La Figura 4.27 (A) muestra el *Packets dropped* en E9 con un ambiente sin tráfico; mientras que la Figura 4.27 (B) muestra el *Packets dropped* en E9 con un ambiente con tráfico. Se observó en el ambiente sin tráfico descarte de paquetes con tamaño de paquetes de 64 bytes; mientras que en el ambiente con tráfico se observó este mismo fenómeno con todos los tamaños de paquetes, con índices muy elevados, todos por encima del 83% hasta un 97%. Este comportamiento podría apuntar a una saturación en las estructuras de datos de recepción de paquetes en los nodos origen y destino, ya que ambos cumplen funciones de *maestros*, los cuales están sometidos a una alta congestión.

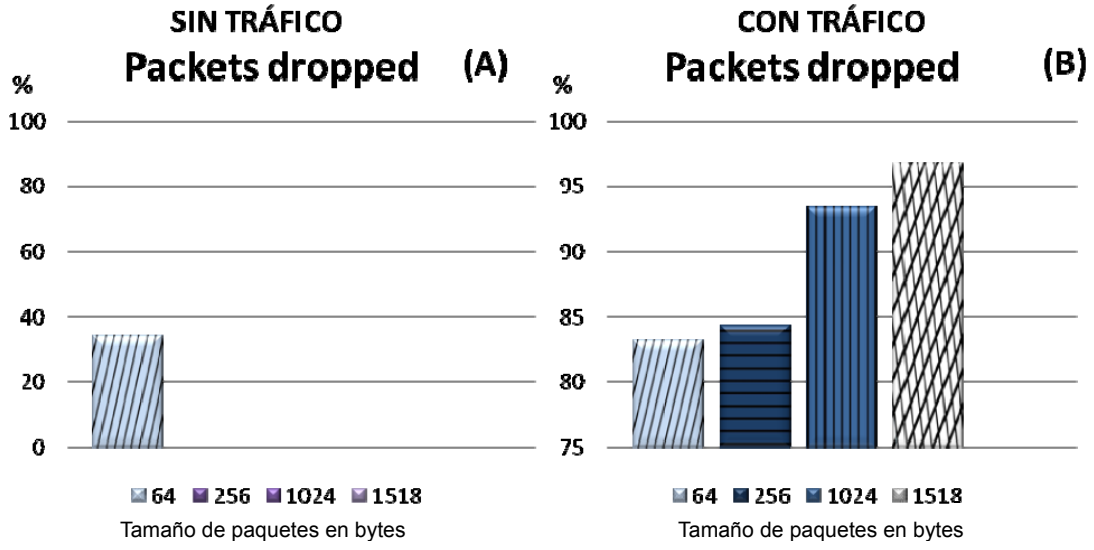
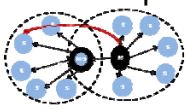


Figura 4.27 Gráficas comparativas de los paquetes descartados escenario 9

En este escenario se observó una disminución en el desempeño de la comunicación de forma significativa en el ambiente con tráfico, a pesar que la

comunicación era directa, los nodos M y M/S se llegan a congestionar por la generación de tráfico en cada *Piconet* donde cumplen funciones de *maestro*.

4.10 Escenario 10 - Resultados



A continuación se presentan los resultados de las pruebas realizadas sobre E10 (Figura 3.11). La Tabla 4.10 muestra los valores promedios de las pruebas realizadas. En este escenario (E10) se observó un decremento en el desempeño global con respecto a E1. Esto puede atribuirse a la presencia de tres factores que podrían afectar el desempeño, la presencia de un nodo con doble función, el salto sobre dicho nodo y la población de ambas *Piconets*.

	SIN TRÁFICO				CON TRÁFICO			
	64	256	1024	1518	64	256	1024	1518
Total de Paquetes recibidos	12016,6667	12206,5000	7371,0000	3705,3333	10160,8333	9987,3333	5429,5000	2810,1667
Avg. Jitter (ms)	0,0489	0,0483	0,0821	0,1585	0,0595	0,0603	0,1162	0,2231
Bitrate (Kb/s)	19,5554	80,4127	184,5296	138,0425	15,5761	61,5563	122,9752	93,4629
Packet rate (Paq/s)	38,1941	39,2640	22,5256	11,3671	30,4221	30,0568	15,0116	7,6962
Packets dropped (%)	91,5050	90,9450	94,7200	97,1833	92,0367	92,3133	95,3467	97,8817

Tabla 4.10 Resultados de pruebas de desempeño en el escenario 10

La Figura 4.28 (A) muestra el *Packet rate* por segundo en E10 con un ambiente sin tráfico; mientras que la Figura 4.28 (B) muestra el *Packet rate* por segundo en E10 con un ambiente con tráfico. Se observó un decremento en *Packet rate* en el ambiente con tráfico, con un promedio aproximado de 27%. Luego de comparar los resultados obtenidos en E1 con E10, se observó un decremento general en *Packet rate* con un promedio aproximado de 86%.

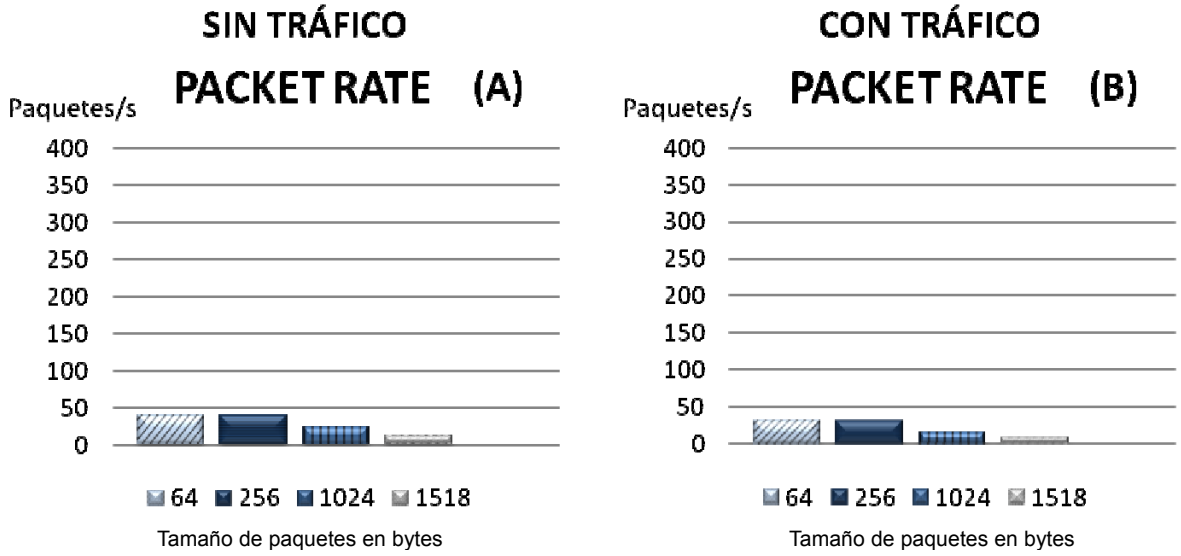


Figura 4.28 Gráficas comparativas de la tasa de paquetes por segundo escenario 10

La Figura 4.29 (A) muestra el *Avg Jitter* en E10 con un ambiente sin tráfico, mientras que la Figura 4.29 (B) muestra el *Avg Jitter* en E10 con un ambiente con tráfico. Se observó un incremento del *Avg Jitter* en el ambiente con tráfico, con un promedio aproximado de 32%. Luego de comparar los resultados obtenidos en E1 con E10, se observó un incremento en *Avg Jitter* en el ambiente sin tráfico, con un promedio aproximado de 606%, mientras que en el ambiente con tráfico el incremento fue de 673%.

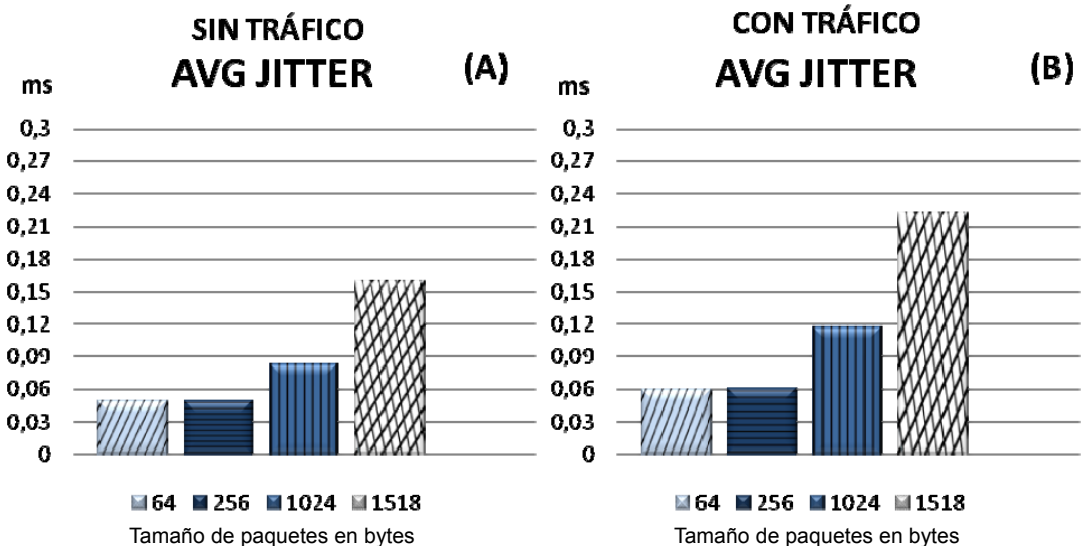


Figura 4.29 Gráficas comparativas del promedio del Jitter escenario 10

La Figura 4.30 (A) muestra el *Packets dropped* en E10 con un ambiente sin tráfico; mientras que la Figura 4.30 (B) muestra el *Packets dropped* en E10 con un ambiente con tráfico. Se observó que en ambos ambientes, sin tráfico y con tráfico, se descartaron paquetes con todos los tamaños; alcanzando índices porcentuales muy elevados, todos superiores a 94%.

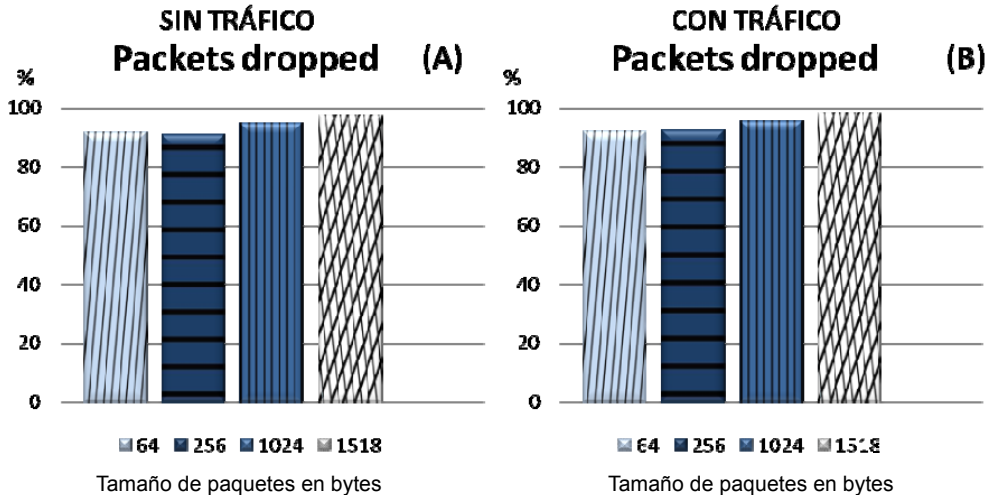
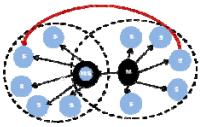


Figura 4.30 Gráficas comparativas de los paquetes descartados escenario 10

En este escenario se observó una disminución en el desempeño de la comunicación, esto debido a que el nodo M/S debe pasar por los procesos de ajustes necesarios para lograr la comunicación en la *Piconet* respectiva. Dadas las características de la tecnología *Bluetooth*, el M/S sólo estará activo en una *Piconet* a la vez; durante ese tiempo no podrá recibir o enviar paquetes en otra *Piconet*. Adicionalmente, a diferencia de E8, ambas *Piconets* se encuentran pobladas, afectando aun más el retardo en la comunicación y finalmente el desempeño.

4.11 Escenario 11 - Resultados



A continuación se presentan los resultados de las pruebas realizadas sobre E11 (Figura 3.12). La Tabla 4.11 muestra los valores promedios de las pruebas realizadas. En este escenario (E2) se observó un decremento en el desempeño global con respecto a E1. Esto puede atribuirse a la presencia de 2 nodos intermedio o saltos, lo que causaría mayor congestión.

	SIN TRÁFICO				CON TRÁFICO			
	64	256	1024	1518	64	256	1024	1518
Total de Paquetes recibidos	11520,5000	10516,6667	5885,8333	3138,5000	8983,5000	8417,5000	4940,5000	2349,3333
Avg. Jitter (ms)	0,0505	0,0557	0,1022	0,1845	0,0659	0,0703	0,1239	0,2511
Bitrate (Kb/s)	18,9233	69,1490	146,5631	118,4383	13,8978	52,7398	115,2797	80,6449
Packet rate (Paq/s)	36,9597	33,7642	17,8910	9,7528	27,1441	25,7519	14,0722	6,6407
Packets dropped (%)	91,6417	92,4233	95,5583	97,6400	93,5267	94,0150	96,1133	97,8717

Tabla 4.11 Resultados de pruebas de desempeño en el escenario 11

La Figura 4.31 (A) muestra el *Packet rate* por segundo en E11 con un ambiente sin tráfico; mientras que la Figura 4.31 (B) muestra el *Packet rate* por segundo en E11 con un ambiente con tráfico. Se observó un decremento en *Packet rate* en el ambiente con tráfico; con un promedio aproximado de 26%. Luego de comparar los resultados obtenidos en E1 con E11, se observó un decremento general en *Packet rate* con un promedio aproximado de 89%.

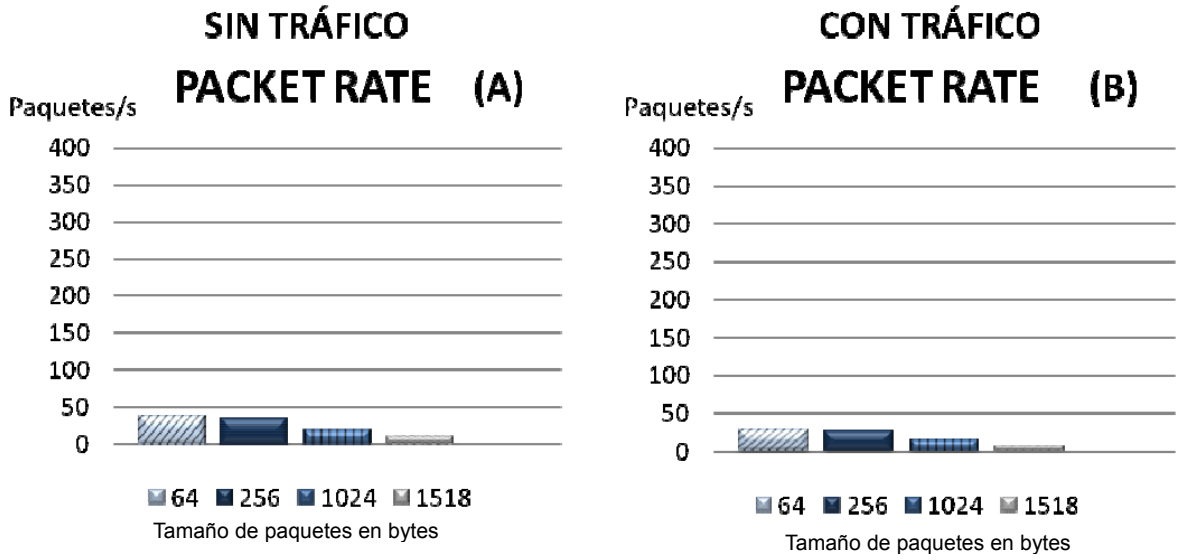


Figura 4.31 Gráficas comparativas de la tasa de paquetes por segundo escenario 11

La Figura 4.32 (A) muestra el *Avg Jitter* en E11 con un ambiente sin tráfico, mientras que la Figura 4.32 (B) muestra el *Avg Jitter* en E11 con un ambiente con tráfico. Se observó un incremento del *Avg Jitter* en el ambiente con tráfico, con un promedio aproximado de 26%. Luego de comparar los resultados obtenidos en E1 con E11, se observó un incremento en *Avg Jitter* en el ambiente sin tráfico, con un promedio aproximado de 701%, mientras que en el ambiente con tráfico el incremento fue de 764%.

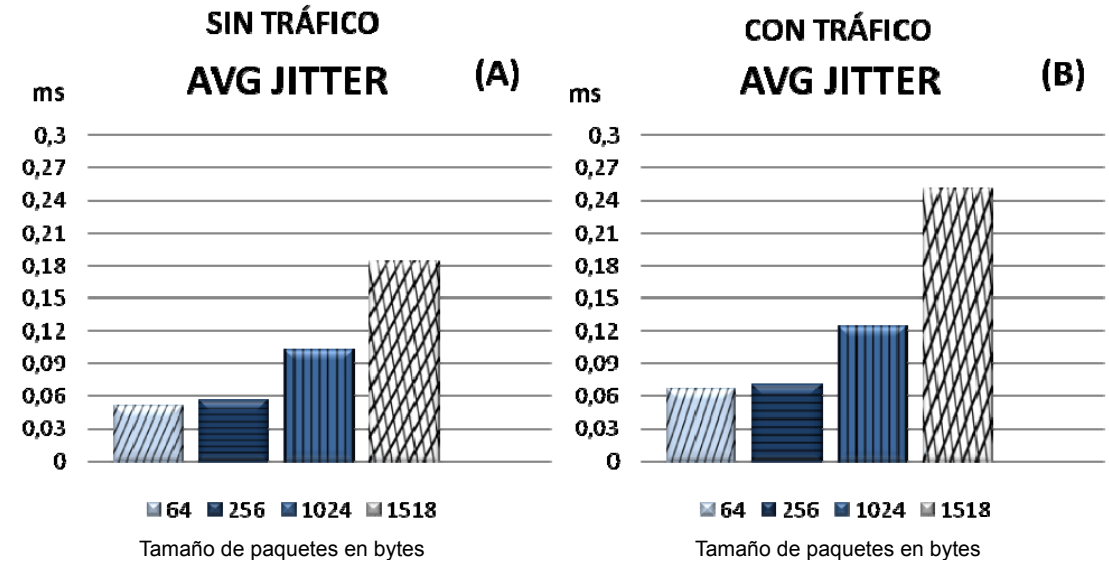


Figura 4.32 Gráficas comparativas del promedio del Jitter escenario 11

La Figura 4.33 (A) muestra el *Packets dropped* en E11 con un ambiente sin tráfico; mientras que la Figura 4.33 (B) muestra el *Packets dropped* en E11 con un ambiente con tráfico. Se observó que en ambos ambientes, sin tráfico y con

tráfico, se descartaron paquetes con todos los tamaños; alcanzando índices porcentuales muy elevados, todos superiores a 95%.

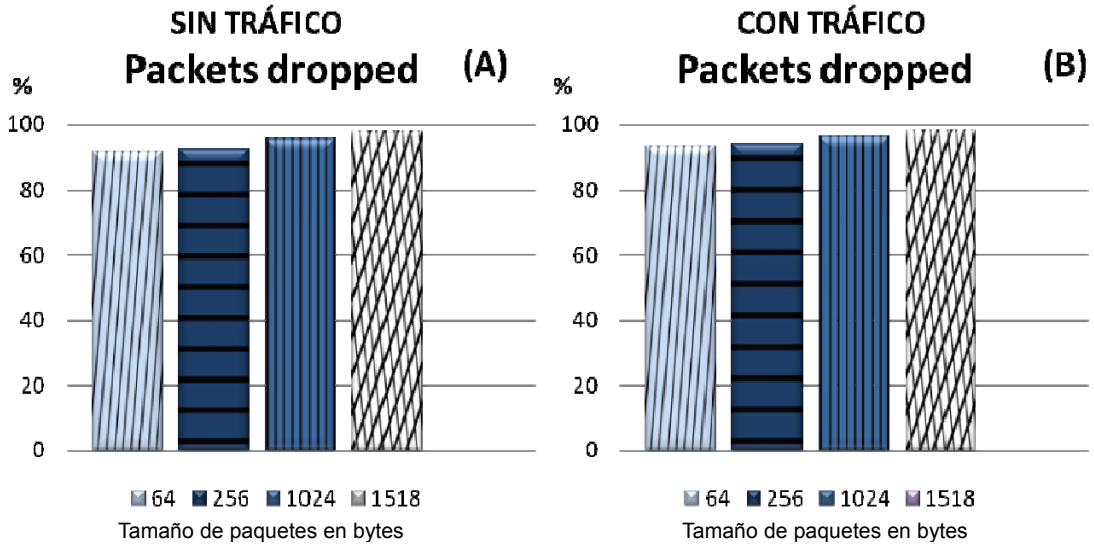
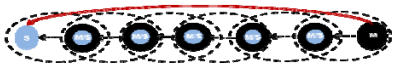


Figura 4.33 Gráficas comparativas de los paquetes descartados escenario 11

En este escenario se observó una disminución en el desempeño de la comunicación, muy similar a E10, al igual que en otros escenarios el salto de un nodo afecta considerablemente la comunicación. Se podría deducir que el primer salto de la comunicación crea un cuello de botella donde el nodo intermedio o salto se congestiona, y sólo logra procesar una pequeña cantidad de tramas para su reenvío, el cual es recibido por el próximo salto a una tasa reducida, teniendo una mayor posibilidad de procesar y retransmitir las tramas.

4.12 Escenario 12 - Resultados



A continuación se presentan los resultados de las pruebas realizadas sobre E12 (Figura 3.13). La Tabla 4.12 muestra los valores promedios de las pruebas realizadas. En este escenario (E12) se observó un decremento en el desempeño global con respecto a E1. Esto puede atribuirse a la presencia de múltiples saltos en la comunicación, en este caso cinco (5).

	SIN TRÁFICO				CON TRÁFICO			
	64	256	1024	1518	64	256	1024	1518
Total de Paquetes recibidos	13597,8333	11747,1667	2591,6667	2256,3333	11528,6667	8175,0000	2491,6667	1658,3333
Avg. Jitter (ms)	0,0426	0,0510	0,1628	0,2385	0,0510	0,0741	0,1826	0,3160
Bitrate (Kb/s)	21,8559	73,1291	66,9587	79,5343	17,1821	48,7076	60,3503	55,7574
Packet rate (Paq/s)	42,6874	35,7076	8,1737	6,5493	33,5588	23,7830	7,3670	4,5914
Packets dropped (%)	90,2967	91,4083	95,5150	97,6783	91,7317	93,9600	96,1583	98,3033

Tabla 4.12 Resultados de pruebas de desempeño en el escenario 12

La Figura 4.34 (A) muestra el *Packet rate* por segundo en E12 con un ambiente sin tráfico; mientras que la Figura 4.34 (B) muestra el *Packet rate* por segundo en E12 con un ambiente con tráfico. Se observó un decremento en *Packet rate* en el ambiente con tráfico, con un promedio aproximado de 24%. Luego de comparar los resultados obtenidos en E1 con E12, se observó un decremento general en *Packet rate* con un promedio aproximado de 91%.

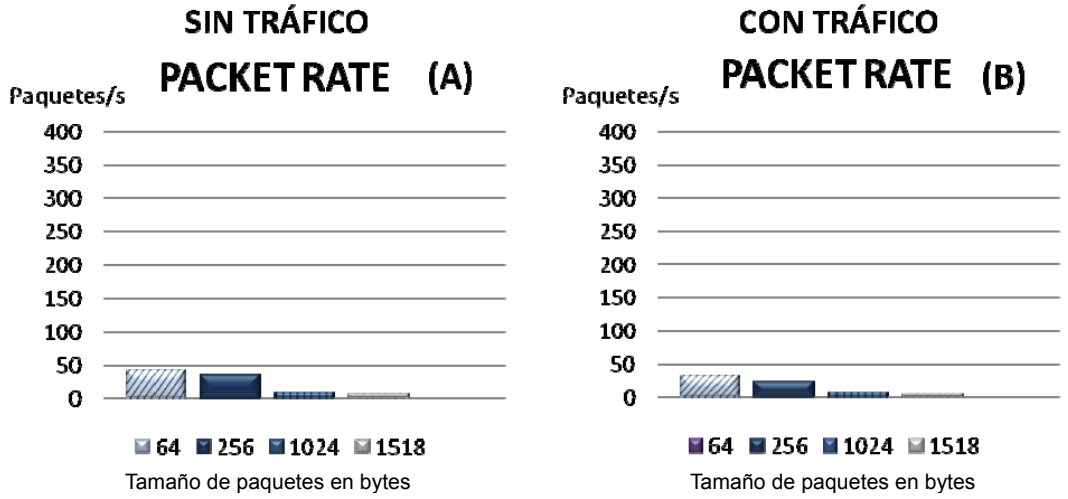


Figura 4.34 Gráficas comparativas de la tasa de paquetes por segundo escenario 12

La Figura 4.35 (A) muestra el *Avg Jitter* en E12 con un ambiente sin tráfico, mientras que la Figura 4.35 (B) muestra el *Avg Jitter* en E12 con un ambiente con tráfico. Se observó un incremento del *Avg Jitter* en el ambiente con tráfico, con un promedio aproximado de 24%. Luego de comparar los resultados obtenidos en E1 con E12, se observó un incremento en *Avg Jitter* en el ambiente sin tráfico, con un promedio aproximado de 792%, mientras que en el ambiente con tráfico el incremento fue de 845%.

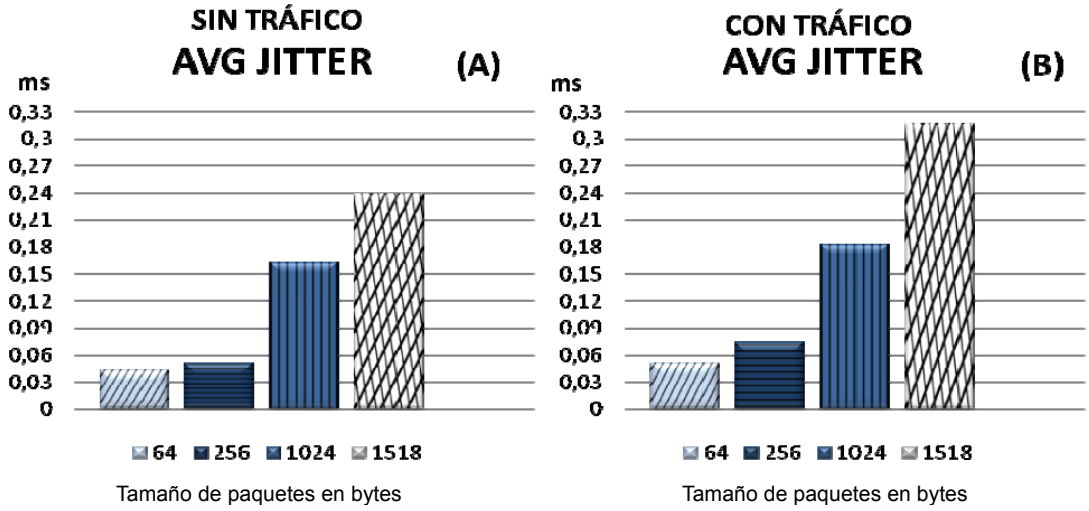


Figura 4.35 Gráficas comparativas del promedio del Jitter escenario 12

La Figura 4.36 (A) muestra el *Packets dropped* en E12 con un ambiente sin tráfico; mientras que la Figura 4.36 (B) muestra el *Packets dropped* en E12 con un ambiente con tráfico. Se observó que en ambos ambientes sin tráfico y con tráfico, se descartaron paquetes con todos los tamaños; alcanzando índices porcentuales muy elevados, todos superiores a 94%.

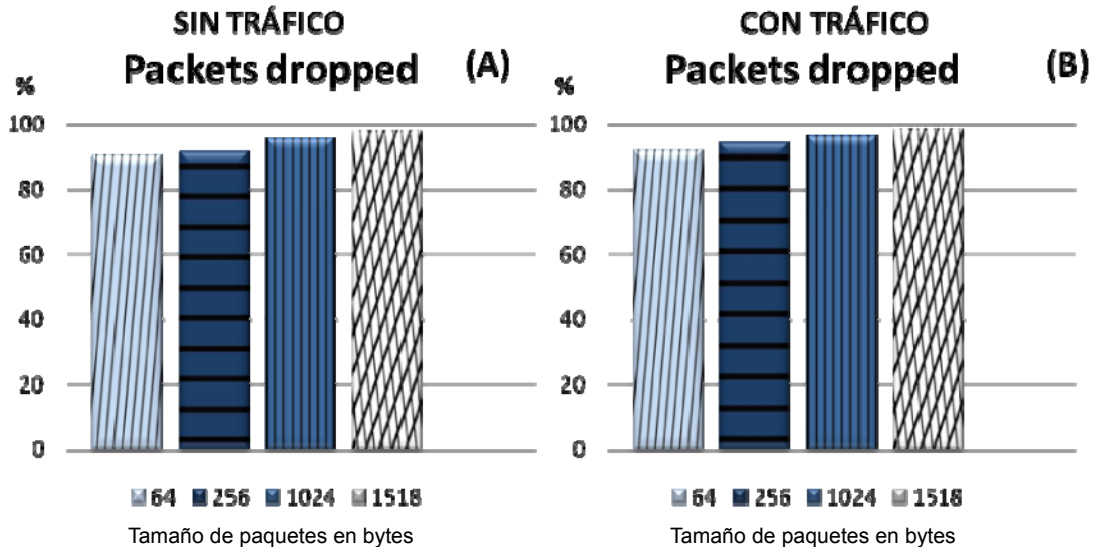
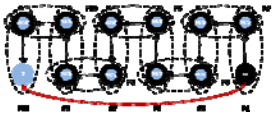


Figura 4.36 Gráficas comparativas de los paquetes descartados escenario 12

Con estos resultados se refuerza la teoría planteada en E11, ya que los resultados fueron muy similares; sin embargo, el numero de saltos aumentó de 2 nodos a 5 nodos, lo que demuestra que en el primer salto o nodo intermedio, es donde se afecta el desempeño en gran medida.

4.13 Escenario 13 - Resultados



A continuación se presentan los resultados de las pruebas realizadas sobre E13 (Figura 3.14). La Tabla 4.13 muestra los valores promedios de las pruebas realizadas. En este escenario (E13) se observó un decremento en el desempeño global con respecto a E1. Esto puede atribuirse a la presencia de múltiples saltos en la comunicación, en este caso diez (10).

	SIN TRÁFICO			
	64	256	1024	1518
Total de Paquetes recibidos	5457,5000	3825,0000	1550,0000	850,0000
Avg. Jitter (ms)	0,1181	0,1574	0,4258	0,6656
Bitrate (Kb/s)	7,5940	20,9066	26,2157	24,9112
Packet rate (Paq/s)	14,8319	10,2083	3,2002	2,0513
Packets dropped (%)	95,4717	96,7050	98,4350	98,9750

Tabla 4.13 Resultados de pruebas de desempeño en el escenario 13

La Figura 4.37 (A) muestra el *Packet rate* por segundo en E13 con un ambiente sin tráfico. Luego de comparar los resultados obtenidos en E1 con E13, se observó un decremento general en *Packet rate* con un promedio aproximado de 91%.

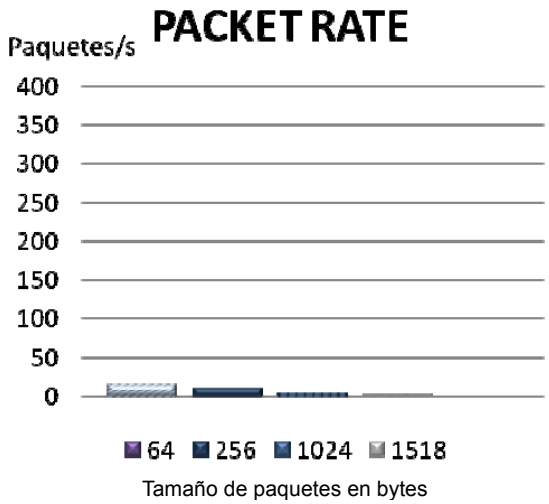


Figura 4.37 Grafica de la tasa de paquetes por segundo escenario 13

La Figura 4.38 muestra el *Avg Jitter* en E13 con un ambiente sin tráfico, se puede observar un incremento exponencial con respecto al E1 de un 1900%.

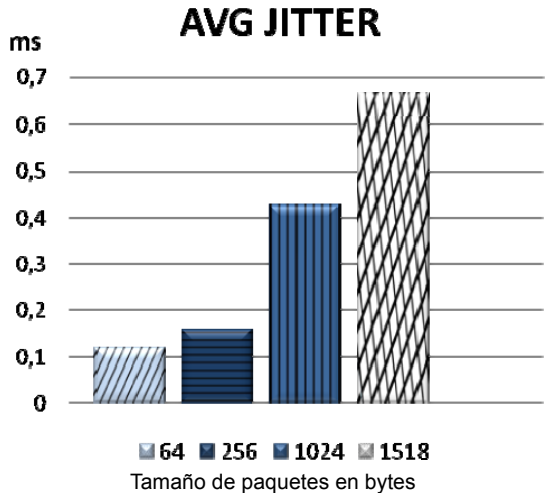


Figura 4.38 Grafica del promedio del Jitter escenario 13

La Figura 4.39 muestra el *Packets dropped* en E13 con un ambiente sin tráfico, se observaron los índices más altos de paquetes descartados de todos los escenarios, todos en promedios sobre el 97% como se pueden observar en la gráfica.

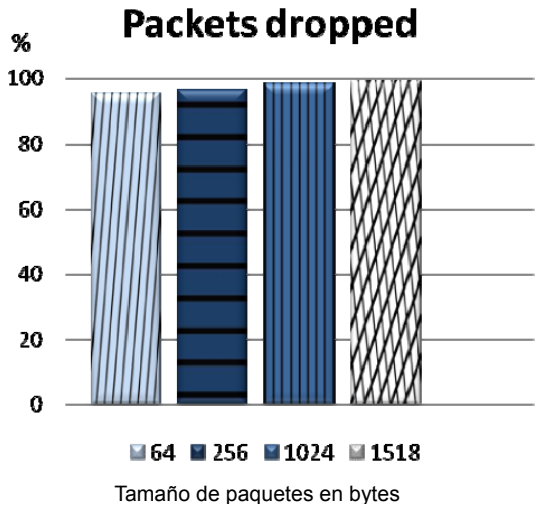
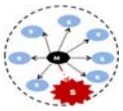


Figura 4.39 Grafica de los paquetes descartados escenario 13

Con estos resultados se puede comprobar que a medida que se agreguen saltos en una topología, tiende a perderse la conexión entre los nodos extremos, dando como resultado un desempeño sumamente pobre, en el cual no es viable realizar ningún tipo de comunicación.

4.14 Escenario 14 - Resultados



Este escenario E14 (Figura 3.15), no se implementó con el fin de realizar pruebas de análisis en el desempeño, sino simplemente para demostrar la limitación de las *Piconets* y de la tecnología, que define que sólo soporta 7 enlaces o nodos *esclavos*.

En la Figura 4.40 se puede observar el proceso donde se agregaron a 7 nodos *esclavos*, en la línea 1 se ejecuta el comando que nos muestra los nodos que se encuentran próximos con los que se podría establecer una conexión, de la línea 2 a la 8 se muestran los comandos necesarios para agregar a cada nodo a la *Piconet*, en la línea 9 muestra los 7 miembros activos de la *Piconet* y finalmente en la línea 10 se intentó agregar un octavo nodo el cual fue rechazado a través del mensaje **"Connect to 00:1A:6B:F3:EA:21 failed. Too many links(31)",** como se puede observar en la figura.

```
1:  # hcitool scan
    scanning ...
            00:0C:76:48:06:DF           icaro04-0
            00:0C:76:9A:31:03           icaro03-0
            00:02:72:D1:65:4B           icaro01-0
            00:1A:6B:F3:EA:21           hector-laptop
            00:02:72:D1:65:43           icaro07-0
            00:02:72:D1:62:7E           icaro05-0
```

```
        00:02:72:D1:65:48           icaro08-0
        00:0C:55:F5:A3:98           icaro02-0
2:  root@icaro06:/home/icaro# pand -c 00:0C:76:48:06:DF –n
    pand[6379]: Bluetooth PAN daemon version 3.26
    pand[6379]: Connecting to 00:0C:76:48:06:DF
    pand[6379]: bnep0 connected
3:  root@icaro06:/home/icaro# pand -c 00:0C:76:9A:31:03 -n
    pand[6394]: Bluetooth PAN daemon version 3.26
    pand[6394]: Connecting to 00:0C:76:9A:31:03
    pand[6394]: bnep1 connected

4:  root@icaro06:/home/icaro# pand -c 00:02:72:D1:65:4B -n
    pand[6402]: Bluetooth PAN daemon version 3.26
    pand[6402]: Connecting to 00:02:72:D1:65:4B
    pand[6402]: bnep2 connected

5:  root@icaro06:/home/icaro# pand -c 00:02:72:D1:65:43 -n
    pand[6413]: Bluetooth PAN daemon version 3.26
    pand[6413]: Connecting to 00:02:72:D1:65:43
    pand[6413]: bnep3 connected

6:  root@icaro06:/home/icaro# pand -c 00:02:72:D1:62:7E -n
    pand[6421]: Bluetooth PAN daemon version 3.26
    pand[6421]: Connecting to 00:02:72:D1:62:7E
    pand[6421]: bnep4 connected

7:  root@icaro06:/home/icaro# pand -c 00:02:72:D1:65:48 -n
    pand[6429]: Bluetooth PAN daemon version 3.26
    pand[6429]: Connecting to 00:02:72:D1:65:48
    pand[6429]: bnep5 connected

8:  root@icaro06:/home/icaro# pand -c 00:0C:55:F5:A3:98 -n
    pand[6437]: Bluetooth PAN daemon version 3.26
    pand[6437]: Connecting to 00:0C:55:F5:A3:98
    pand[6437]: bnep6 connected

9:  root@icaro06:/home/icaro# pand --show
    bnep6 00:0C:55:F5:A3:98 PANU
    bnep5 00:02:72:D1:65:48 PANU
    bnep4 00:02:72:D1:62:7E PANU
    bnep3 00:02:72:D1:65:43 PANU
    bnep2 00:02:72:D1:65:4B PANU
    bnep1 00:0C:76:9A:31:03 PANU
    bnep0 00:0C:76:48:06:DF PANU

10: root@icaro06:/home/icaro# pand -c 00:1A:6B:F3:EA:21 -n
    pand[6445]: Bluetooth PAN daemon version 3.26
    pand[6445]: Connecting to 00:1A:6B:F3:EA:21
    pand[6445]: Connect to 00:1A:6B:F3:EA:21 failed. Too
    many links(31)
```

Figura 4.40 Ejemplo de intento de agregar 8 nodos *esclavos* a una *Piconet* escenario14

4.15 Prueba de conectividad

Con el uso de Ping se puede observar en la Figura 4.41, los resultados del promedio del rtt en milisegundos por escenario, se pudo observar que se logró la conectividad en cada uno de los escenarios, presentando un pico en el promedio

del rtt en los escenarios con nodos intermedios o puentes, reforzando la idea del efecto negativo en la merma del desempeño de los mismos.

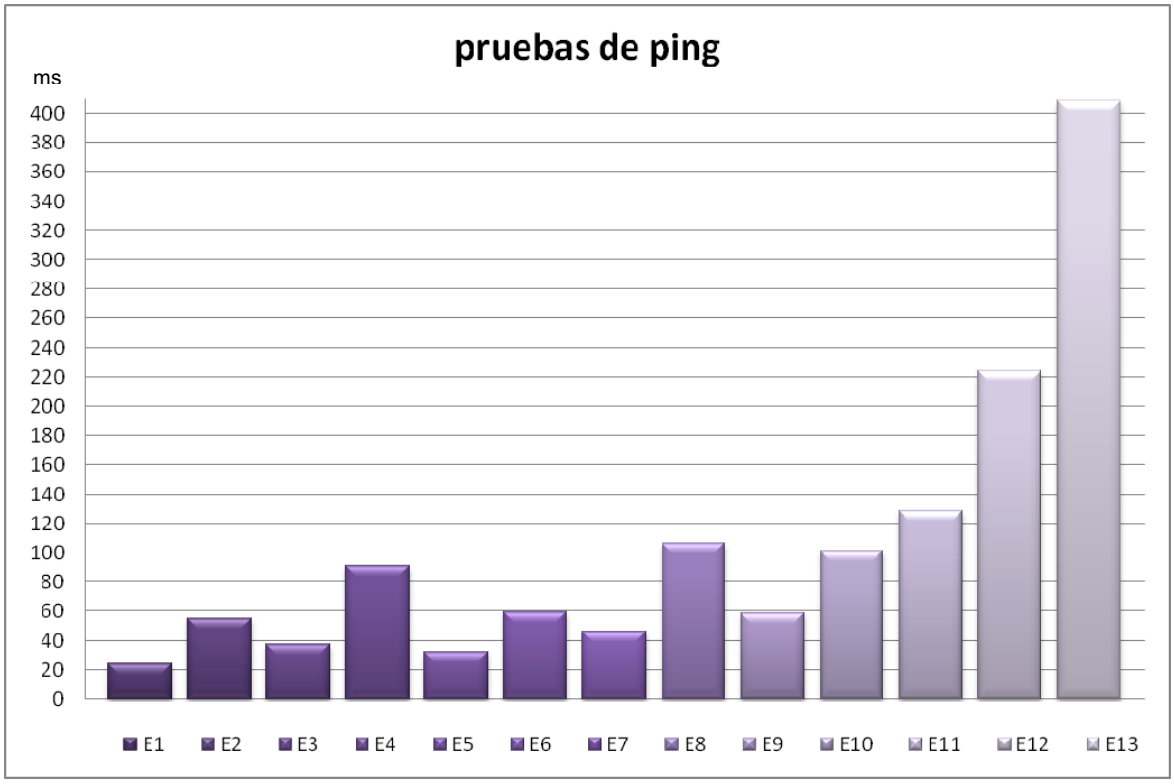


Figura 4.41 Prueba de conectividad con la herramienta ping

La Tabla 4.14 muestra los valores promedios del RTT para cada escenario definidos en el Capítulo 3.

ESCENARIO	1	2	3	4	5	6	7	8	9	10	11	12	13
AVG RTT	24,119	55,424	37,574	90,986	32,55	59,489	45,922	106,6	58,787	101,37	128,75	224,05	408,82

Tabla 4.14 Resultados de los promedios de RTT en las pruebas de ping

- **Análisis general de los resultados obtenidos**

Una vez finalizadas las pruebas realizadas sobre los escenarios planteados en el Capítulo 3, y después de analizar y comparar los resultados obtenidos se lograron identificar factores que afectan el desempeño en las comunicaciones de una red *Bluetooth*.

En el caso de la *Piconet*, un factor determinante en la disminución del desempeño fue la presencia del nodo intermedio o puente, el cual podría apuntar a una saturación en las estructuras de datos de recepción de paquetes en el nodo intermedio; adicionalmente, el otro factor determinante en la merma del desempeño de la comunicación, es que los miembros de una *Piconet* poblada transmitan información concurrentemente, la cual converge en el *maestro* causando un cuello de botella.

En el caso de las *Scatternets*, al igual como sucede en las *Piconets* se ven afectadas por los nodos intermedios o puente y *Piconets* pobladas que transmitan información; sin embargo, la proporción en la merma del desempeño es mayor, debido a que estos nodos cumplen una doble función (*maestro*/*esclavo*), los cuales ejecutan procesos necesarios con el fin de realizar la interconexión entre las *Piconets*.

Una posible causa de la congestión del nodo intermedio se puede originar en el diseño de la tecnología, debido a que las tramas de datos definidos por el protocolo banda base están limitados en tamaño. Las tramas L2CAP grandes deben ser segmentadas en varias tramas banda base más pequeñas antes de transmitirse; adicionalmente las tramas deben pasar por el proceso de ascenso y descenso por la estructura de capas de la pila de protocolo de *Bluetooth* y también lo deben hacer por la estructura de capas de la pila de protocolo TCP/IP, ocasionando un retardo, debido al tiempo de cómputo adicional usado en el procesamiento de las tramas.

Otra causa de la congestión de los nodos intermedios, pero sólo en el caso de los nodos con doble función (*Scatternet*), se atribuye a que los nodos deben cambiar constantemente su modo de conexión activo y pasar a un estado de conexión no activo conocido como modo *Hold*, modificar el ajuste interno de su reloj, realizar los ajustes correspondientes de los parámetros para incorporarse a otra *Piconet*. Esto implica que la comunicación dirigida a estos nodos se verá interrumpida mientras se encuentre activa en la otra *Piconet*. Un nodo intermedio activo en una *Piconets* almacena tramas dirigidas a nodos de la *Piconet* adyacente, entregándolos posteriormente a las estaciones destino cuando el tiempo en modo *Hold* termine. Estos procesos de ajustes y cambios de modos de conexión implican otro tiempo de cómputo adicional al ya mencionado.

Estos factores influyen directamente en el retardo de las comunicaciones y la disminución en el desempeño. Adicionalmente, es posible que la tasa de llegada

de paquetes sobrepase la capacidad de procesamiento de las tramas en los nodos intermedios, causando una saturación en los *buffers* de recepción.

Es importante mencionar que se observó que las comunicaciones presentan un mejor desempeño cuando son iniciadas desde un *maestro* a un *esclavo*, esto es debido a que el *maestro* es el dispositivo encargada de controlar el canal la comunicación en su *Piconet*.

Otro evento significativo se detectó durante las transmisiones, donde se observó por medio de capturas, utilizando un *sniffer* Wireshark[5], el envío de forma constante de paquetes ARP. Las constantes actualizaciones de las tablas ARP causan una congestión y retardo en los canales de comunicación, siendo un posible factor en la disminución del desempeño en las redes de comunicación con tecnología *Bluetooth*.

[5] http://www.wireshark.org

5. CONCLUSIONES Y RECOMENDACIONES

Se estudió detalladamente la especificación *Bluetooth*, las utilidades de la pila de protocolos Bluez, así como la especificación BNEP. Los conceptos de *Bluetooth* adquiridos fueron esenciales para el desarrollo de esta investigación y se aplicaron específicamente en la implementación de la red inalámbrica *Bluetooth*.

La comunicación se logro empleando en todos los equipos la pila de protocolo Bluez (basado en *Bluetooth*) para Linux, y herramientas que sirvieron para crear un entorno de red a través de los enlaces *Bluetooth*, tales como el protocolo de encapsulamiento de red BNEP de *Bluetooth* y el perfil PAN.

Se definió un grupo de escenarios, al cual se le aplicó una serie de pruebas con técnicas de benchmarking, que sirvieron para identificar los factores que afectan el desempeño.

A lo largo de esta investigación se han podido corroborar los principales factores, en el caso de las *Piconets*, la figura del nodo intermedio o puente y el otro factor determinante es que los miembros de una *Piconet* poblada transmitan información concurrentemente. En el caso de las *Scatternets*, se ve afectado por los mismos factores; sin embargo, la proporción en la merma del desempeño es mayor, debido a la doble función que desempeñan los nodos *maestro*/*esclavos* y los procesos de ajustes y sincronización correspondientes.

Adicionalmente se le atribuye un factor en el retardo de las comunicaciones a los procesos de ascenso y descenso de las tramas, tanto en las capas de la pila de protocolos de *Bluetooth* como la pila de protocolo de TCP/IP. Otra posible causa del bajo desempeño, es la segmentación y reensamblaje entre las capas L2CAP y Banda Base, por la diferencia tan alta de sus MTU. Todas estas causas hacen pensar en la posibilidad de que la tasa de arribo de paquetes sobrepase la capacidad de procesamiento de las tramas en los nodos intermedios, causando una saturación en los buffers de recepción.

Finalmente se puede concluir que las comunicaciones con tecnología *Bluetooth* no son factibles para ser usadas en un ambiente de red de área local, en donde se requiere de tasas de transferencias y ancho de banda con niveles superiores a los presentados por la tecnología *Bluetooth* en esta investigación, ya que en redes Ethernet con tecnología 10BaseT, 100BaseTX y la Wi-Fi con tasas de 10Mbps, 100Mbps y hasta 600Mbps respectivamente, superan con creces las tasas de transferencia con tecnologia *Bluetooth* que en el mejor caso alcanzó sólo 1Mbps y en el peor caso solo alcanzó 8Kbps.

Esta tecnología no fue pensada para su uso en ambiente de redes, sino para la comunicación punto a punto entre dos dispositivos, sustituyendo el cable

serial, y no para desempeñar de forma eficiente las funciones realizada por un router o switch.

Finalmente se puede afirmar que se logro cumplir satisfactoriamente con todos los objetivos planteados. Se establecieron los mecanismos para la formación e interconexión de varias *Piconets*, lo que se conoce como *Scatternet*, se empleó un testbed que permitió sacar conclusiones acerca del comportamiento de la tecnología *Bluetooth* en las comunicaciones de las topologías *Piconet* y *Scatternet*.

Contribuciones

Los aportes que se generaron en este Trabajo Especial de Grado fueron:

- Implementación de una metodología para la creación de la topología de comunicación *Bluetooth* llamada *Scatternet*, la cual no presenta muchos estudios ni implementaciones.
- Estudio de conectividad y desempeño de las distintas topologías, *Piconet* y *Scatternet*.
- Generación de resultados que puedan ser útiles para futuros estudios sobre esta tecnología, que aún se encuentra en desarrollo.

Limitaciones

- Sólo se contó con 12 *dongles Bluetooth*, no todos eran de la misma marca, modelo ni versión.
- No se contó con un equipo capturador y analizador de tramas en la capa Banda Base, que permitiera estudiar el comportamiento más a fondo de las comunicaciones.

Recomendaciones

- Realizar la implementación de la topología *Scatternet* sobre el sistema operativo Windows.
- Adquirir más *dongles* de versiones recientes y de marca reconocidas, debido a que se presentaron conflictos con *dongles* de marca genérica.
- Adquirir un equipo que permita capturar los paquetes que están debajo de la capa HCI en la pila la tecnología *Bluetooth*, para estudiar más a fondo todas sus capas y el comportamiento de las mismas, para futuros estudios.
- Como posible tema de investigación, analizar y calcular el tiempo que demora un dispositivo intermedio en cambiar el estado de conexión de una *Piconet* a otra.
- Estudiar el desempeño en las comunicaciones con nodos que cumplan funciones de *esclavo* en dos *Piconets*.
- Mejorar la estructura de la pila de protocolos *Bluetooth* para realizar el direccionamiento en redes sin hacer uso de los protocolos adoptados.

- Mejorar la estructura de la pila de protocolos *Bluetooth* con el fin de lograr que nodos con doble rol puedan estar activos en un instante de tiempo en dos o más Piconets, para obtener una comunicación fluida.

6. BIBLIOGRAFÍA

[1] **A. Moreno Tablado.** Seguridad en *Bluetooth*. *Tesis de grado.* [En línea] Universidad Pontificia Comillas, Escuela Técnica Superior de Ingeniería, junio de 2006. [Citado el: 1 de Abril de 2009.] www.telefonica.net/web2/telamarinera/docus/PFC.Seguridad.en.*Bluetooth*.pdf .

[2] ***Bluetooth* SIG, Inc.** [En línea] 2008. [Citado el: 24 de Marzo de 2009.] http://spanish.*Bluetooth*.com/*Bluetooth*/Technology/ .

[3] **IEEE Computer Society.** IEEE Std 802.15.1™. *Part 15.1: Wireless Medium Acces Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs).* s.l. : The Istitute of Electrical and Electronics Engineers, Inc, 2002.

[4] **A. Miller Brent , C. Bisdikian.** *Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communications.* s.l. : Prentice Hall, 2000.

[5] **M. Miller.** *Discovering Bluetooth.* s.l. : SYBEX Inc, 2001.

[6] ***Bluetooth* SIG, Inc.** *BLUETOOTH* SPECIFICATION Version 2.1 + EDR. [En línea] [Citado el: 1 de Abril de 2009.] http://spanish.*Bluetooth*.com/*Bluetooth*/Technology/Building/Specifications/ .

[7] **N. Muller.** *Bluetooth Demystified.* s.l. : McGraw-Hill Professional, 2000.

[8] Manual de FreeBSD. *Capítulo 29. Networking avanzado "Bluetooth".* [En línea] [Citado el: 1 de Abril de 2009.] http://www.freebsd.org/doc/es_ES.ISO8859-1/books/handbook/network-*Bluetooth*.html .

[9] ***Bluetooth* Special I**[10] **Grupo de Trabajo de la Red de S. Bradner.** RFC 2544. Benchmarking Methodology for Network Interconnect Devices. Marzo 1999. [En línea] [Citado el: 1 de Abril de 2009.] http://www.ietf.org/rfc/rfc2544.txt

[11] **M. Molina (DANTE) A. Van Maele (Belnet).** Deliverable DJ1.2.3Network Metric Report. Febrero 2006. [En línea] [Citado el: 16 de junio de 2009.] http://www.geant.net/upload/pdf/GN2-05-265v4-Deliverable_DJ1-2-3_Network_Metric_Report.pdf

nterest Group. *Bluetooth* Network Encapsulation Protocol (BNEP) Specification. Specification of the *Bluetooth* System, Versión 1.0. 14 de Febrero de 2003. [En línea] [Citado el: 1 de Abril de 2009.] http://*Bluetooth*.com/NR/rdonlyres/E4B8D286-9DA5-4465-82E0-B1883FB2AC59/912/BNEPSpecification2.pdf

7. ANEXOS

Escenario 1 – Gráficas adicionales

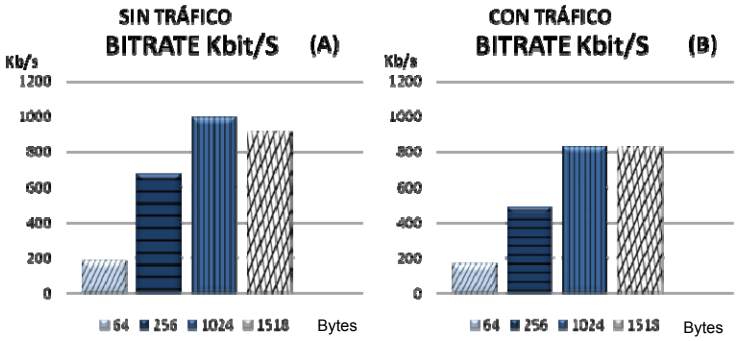


Figura 7.1 Gráficas comparativas del *bitrate* escenario 1

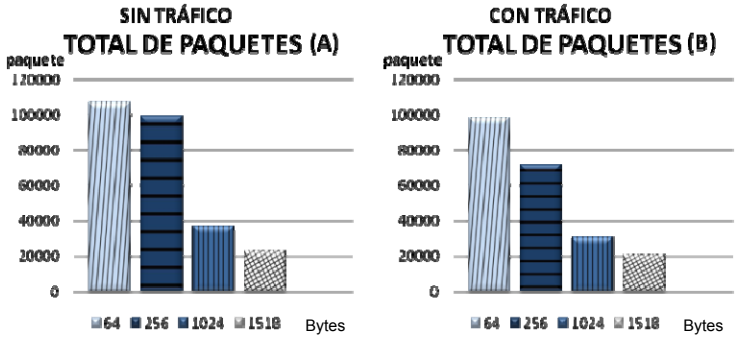


Figura 7.2 Gráficas comparativas del total de paquetes recibidos escenario 1

Escenario 2 – Gráficas adicionales

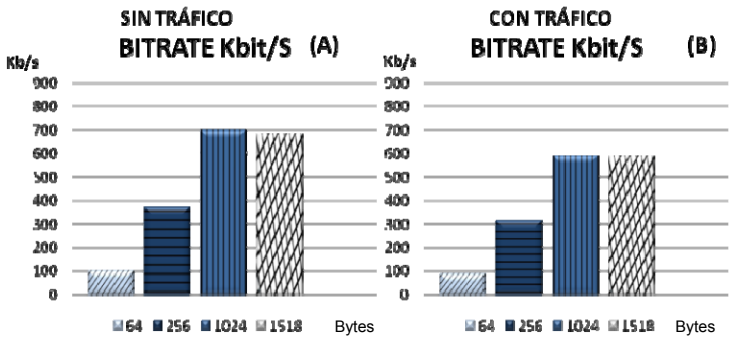


Figura 7.3 Gráficas comparativas del *bitrate* escenario 2

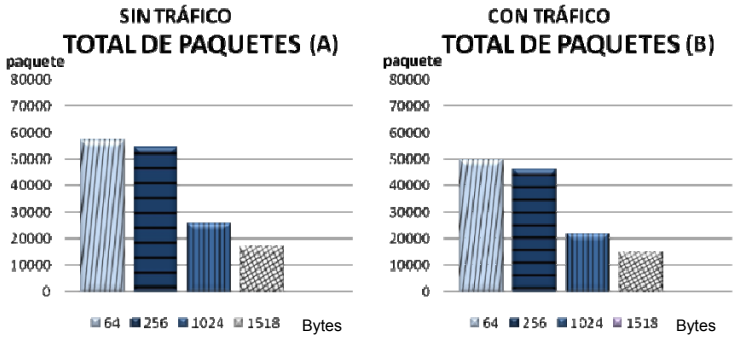


Figura 7.4 Gráficas comparativas del total de paquetes recibidos escenario 2

Escenario 3 – Gráficas adicionales

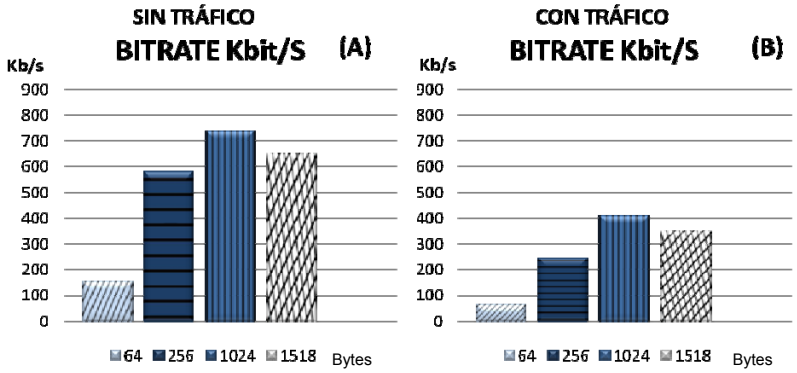


Figura 7.5 Gráficas comparativas del *bitrate* escenario 3

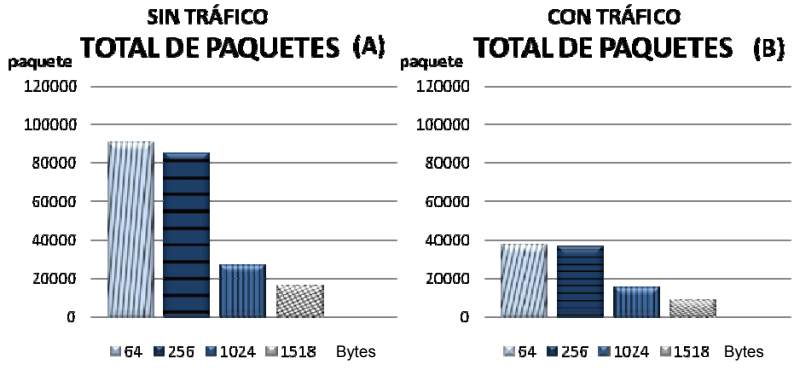


Figura 7.6 Gráficas comparativas del total de paquetes recibidos escenario 3

Escenario 4 – Gráficas adicionales

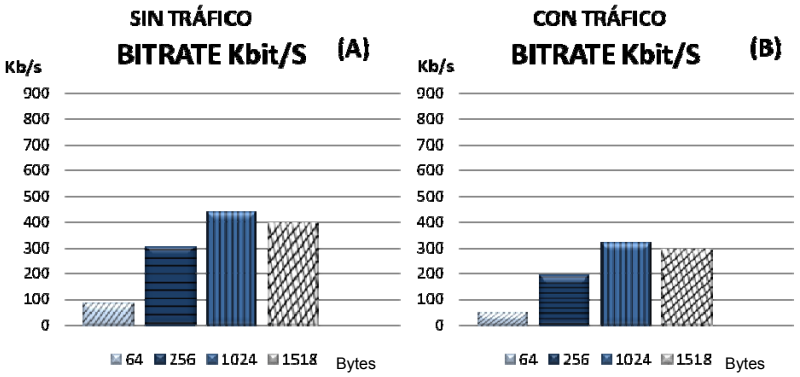


Figura 7.7 Gráficas comparativas del *bitrate* escenario 4

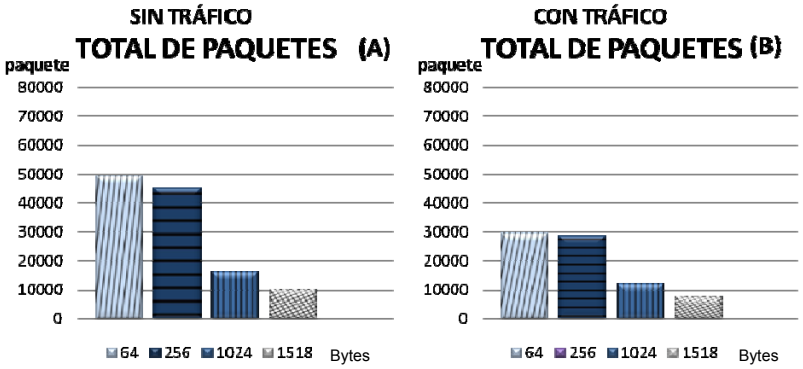


Figura 7.8 Gráficas comparativas del total de paquetes recibidos escenario 4

Escenario 5 – Gráficas adicionales

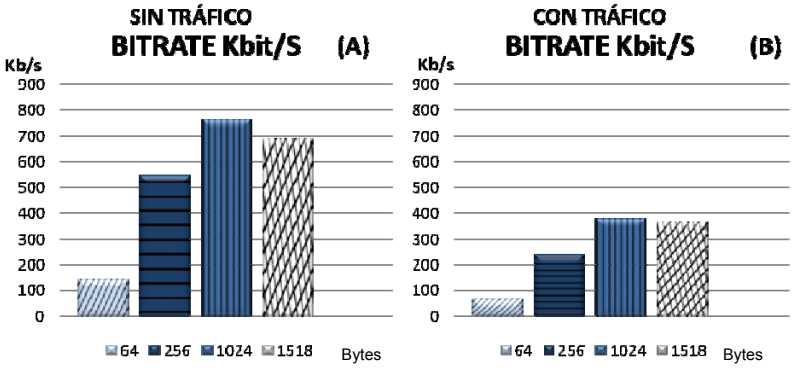


Figura 7.9 Gráficas comparativas del *bitrate* escenario 5

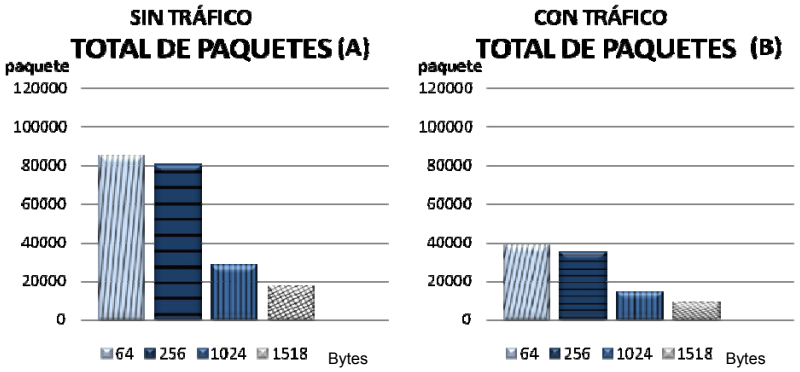


Figura 7.10 Gráficas comparativas del total de paquetes recibidos escenario 5

Escenario 6 – Gráficas adicionales

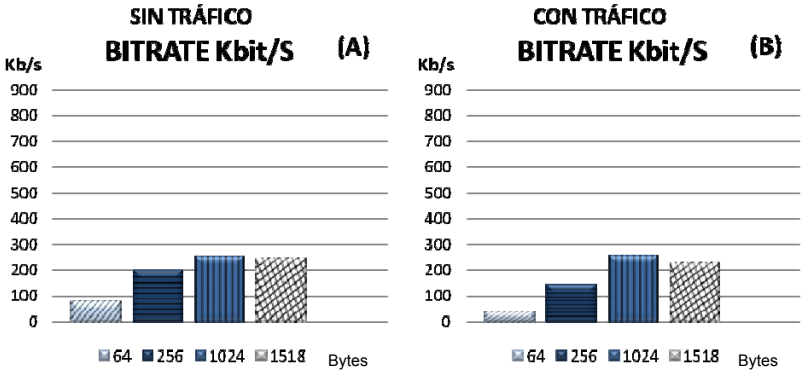


Figura 7.11 Gráficas comparativas del *bitrate* escenario 6

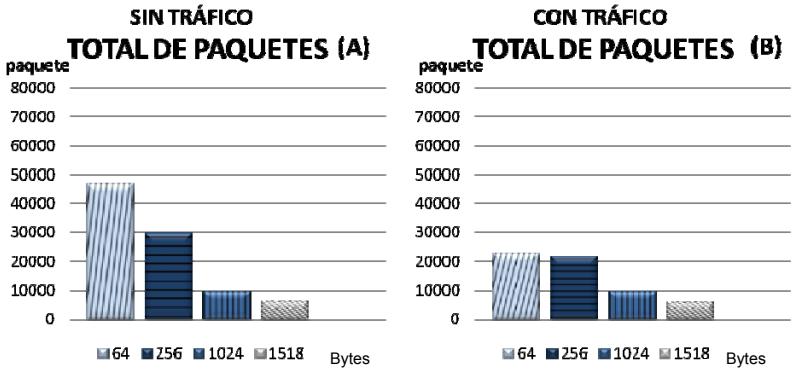


Figura 7.12 Gráficas comparativas del total de paquetes recibidos escenario 6

Escenario 7 – Gráficas adicionales

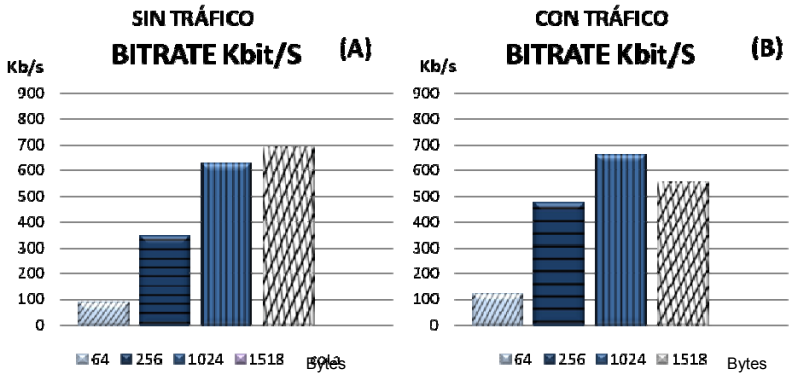


Figura 7.13 Gráficas comparativas del *bitrate* escenario 7

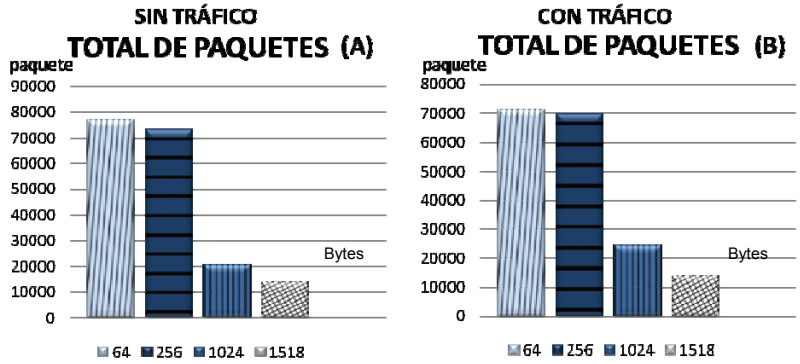


Figura 7.14 Gráficas comparativas del total de paquetes recibidos escenario 7

Escenario 8 – Gráficas adicionales

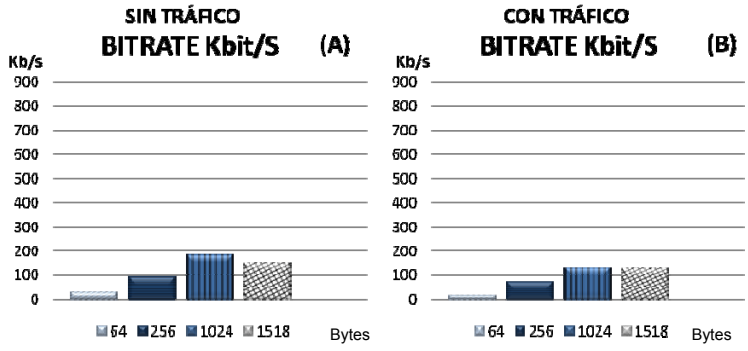


Figura 7.15 Gráficas comparativas del *bitrate* escenario 8

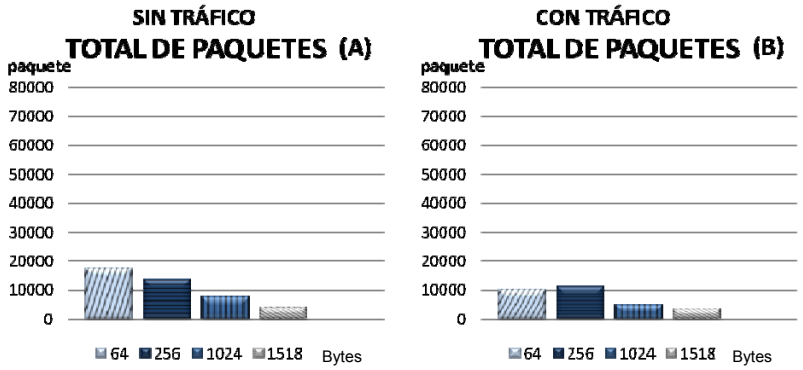


Figura 7.16 Gráficas comparativas del total de paquetes recibidos escenario 8

Escenario 9 – Gráficas adicionales

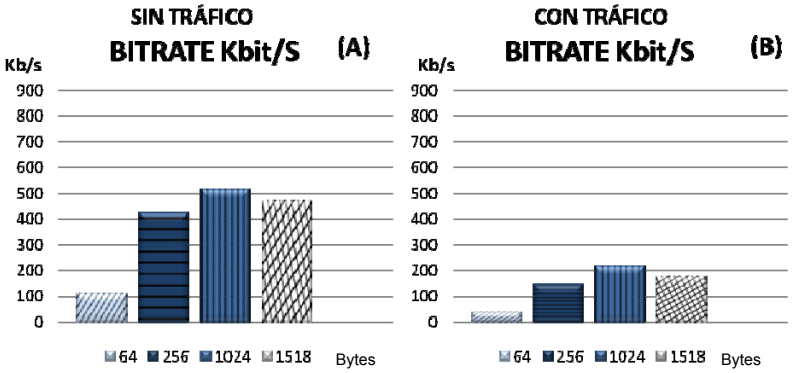


Figura 7.17 Gráficas comparativas del *bitrate* escenario 9

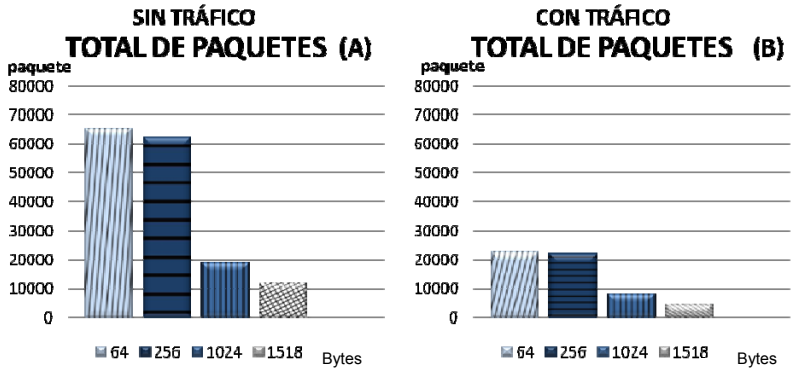


Figura 7.18 Gráficas comparativas del total de paquetes recibidos escenario 9

Escenario 10 – Gráficas adicionales

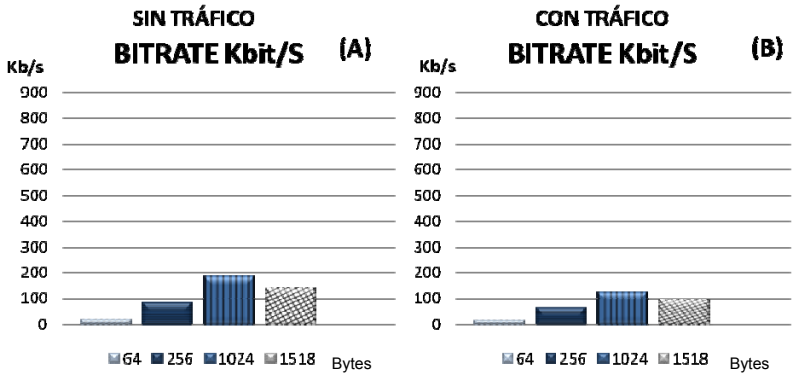


Figura 7.19 Gráficas comparativas del *bitrate* escenario 10

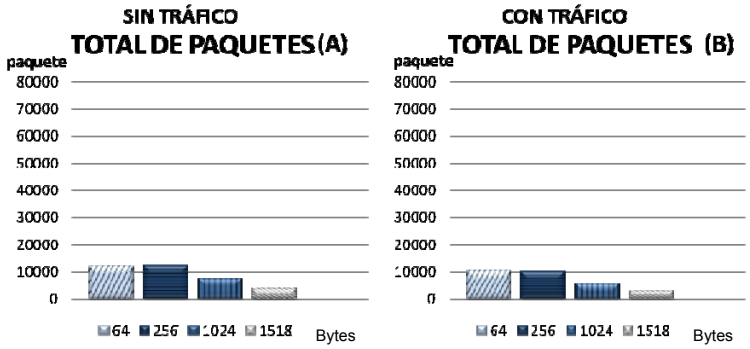


Figura 7.20 Gráficas comparativas del total de paquetes recibidos escenario 10

Escenario 11 – Gráficas adicionales

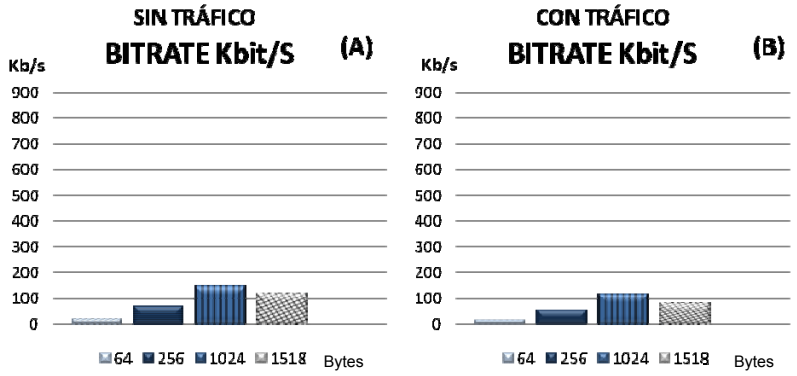


Figura 7.21 Gráficas comparativas del *bitrate* escenario 11

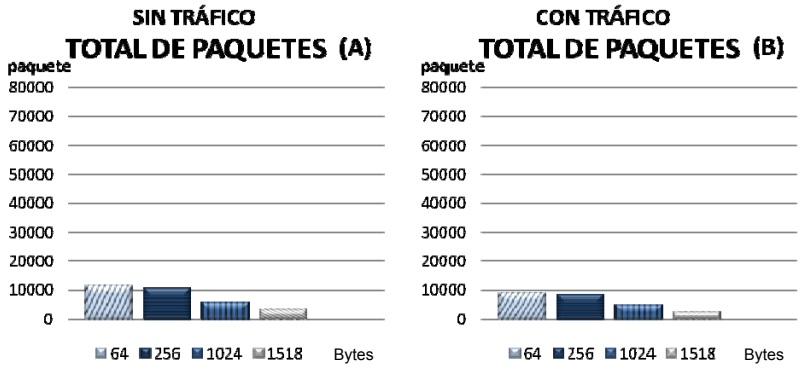


Figura 7.22 Gráficas comparativas del total de paquetes recibidos escenario 11

Escenario 12 – Gráficas adicionales

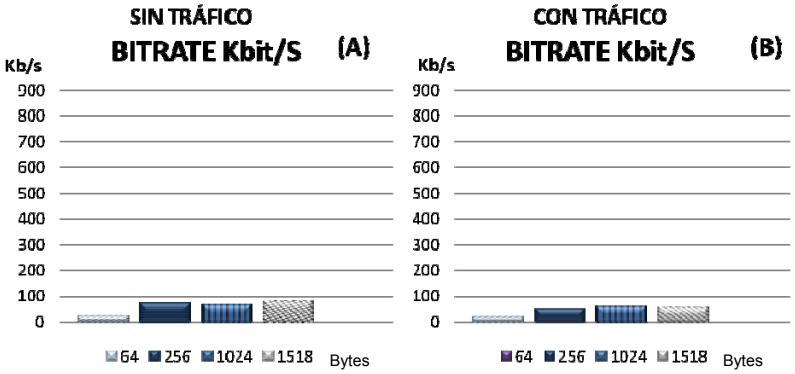


Figura 7.23 Gráficas comparativas del *bitrate* escenario 12

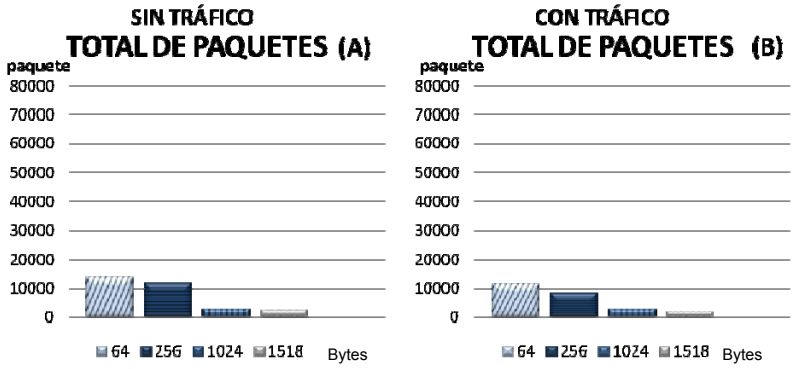


Figura 7.24 Gráficas comparativas del total de paquetes recibidos escenario 12

Escenario 13 – Gráficas adicionales.

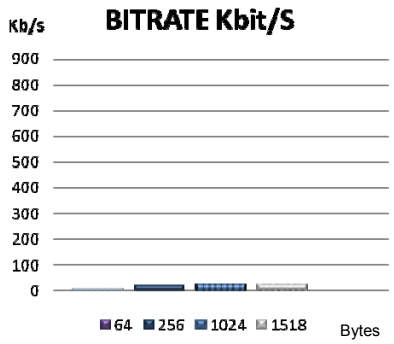


Figura 7.25 Gráficas comparativas del *bitrate* escenario 13

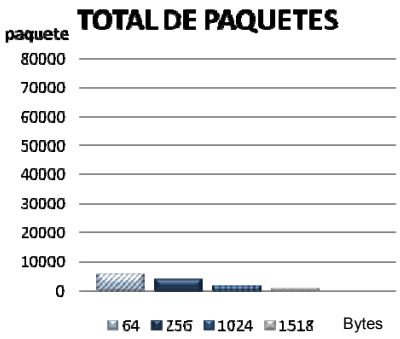


Figura 7.26 Gráficas comparativas del total de paquetes recibidos escenario 13