

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro para las Investigaciones en Comunicaciones y Redes
Laboratorio de Redes Móviles, Inalámbricas y Distribuidas (ICARO)

**Estudio de los Protocolos de
Enrutamiento para Redes Malladas
con Restricciones de Ancho de
Banda en el Borde**

Trabajo Especial de Grado
presentado ante la Ilustre
Universidad Central de Venezuela
por los Bachilleres:

Carolina Balderrama
C.I.: 17144582
E-mail: cmbalderrama@gmail.com

Mariana Colombo
C.I.: 17423414
E-mail: mariana.colombo.m@gmail.com

para optar al título de Licenciado en Computación

Tutora: María Elena Villapol

Caracas, Noviembre 2010

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro para las Investigaciones en Comunicaciones y Redes
Laboratorio de Redes Móviles, Inalámbricas y Distribuidas (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 Carolina Balderrama C.I.: 17144582 y Mariana Colombo C.I.: 17423414, con el título **“Estudio de los Protocolos de Enrutamiento para Redes Malladas con Restricciones de Ancho de Banda en el Borde”**, 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 01 de Noviembre de 2010, a las 4:00 pm, para que sus autores lo defendieran en forma pública, en el Laboratorio Internet 2, 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 01 de Noviembre de 2010, dejándose también constancia de que actuó como Coordinador del Jurado la Profesora Tutora María Elena Villapol.

Prof. María Elena Villapol
(Tutora)

Prof. Aparicio Peña
(Jurado Principal)

Prof. Ana Morales
(Jurado Principal)

AGRADECIMIENTOS

A mi mamá “Anama” por motivarme al ver su pasión por los estudios, y a pesar de la distancia siempre apoyarme y ayudarme en todo. Y más que nada por ser el mejor ejemplo de que con dedicación y trabajo duro siempre puedes lograr lo que quieres.

A mi papá Enrique por siempre estar pendiente, apoyarme y ayudarme en todo lo que he necesitado.

A mi abuela “mama lili” quien siempre me ha enseñado a trabajar duro para lograr las metas. Gracias por siempre apoyarme.

A mi hermano Alejandro a quien le debo un pedacito de mi título y le agradezco haberme ayudado siempre que lo he necesitado.

A Cristina quien no solo ha sido mi hermana sino mi profesora, por toda su ayuda y buenos consejos.

A Emilio por toda su paciencia, apoyo y amor, gracias por soportarme. Y a toda su familia por hacerme sentir como una más de ellos.

A “la profe” María Elena Villapol, quien definitivamente ha sido una excelente profesora y tutora en todo momento. Gracias por apoyarnos y guiarnos en todo.

A Karima Velásquez, por toda su ayuda, sus consejos y su amistad, no solo durante la realización de la tesis, sino desde que me recibió con los brazos abiertos en Comunicación de Datos y en adelante.

A David Pérez, por dejar que lo molestáramos tanto en las vacaciones, como en toda nuestra estadía en el laboratorio ICARO, y a pesar de eso siempre estar dispuesto a ayudarnos y aconsejarnos.

A todos mis compañeros de la universidad, David (dos), Mafer, Joel, Diana, Andres, Andreina y Auri, por todos los buenos momentos, no solo académicos sino también fuera de la universidad.

A todos mis amigos de la vida, Alessandra, Lorena, Pili, Beto, Sandra, Miguel, Sandra (dms) y a todos los demás, por siempre estar pendiente y apoyarme en los momentos estresantes. Especialmente a Harold Martínez quien nunca ha tenido más que buenos consejos y ayuda para mí.

Por último pero no menos importante, a la que se convirtió en una amiga muy especial durante toda la carrera, mi compañera de tesis, Mariana Colombo, por siempre estar conmigo en todas las actividades de la universidad, y por compartir conmigo momentos especiales en la vida. Definitivamente la mejor compañera de tesis que pude haber elegido. Gracias por todo.

**Gracias a todos!
Carolina Balderrama**

AGRADECIMIENTOS

Para comenzar quisiera agradecer a Dios por estar a mi lado en cada una de las etapas de mi vida, por darme la fuerza y el ánimo para seguir adelante, por poner siempre angelitos en mi camino y por nunca dejarme sola.

A mis padres Pasquale y Cielo, porque simplemente por ellos estoy aquí, por enseñarme cada día a ser una mejor persona, especialmente a mi mamá, por su apoyo incondicional, por darme todo lo que estaba en sus manos con una sonrisa en la cara, por creer en mí y tener siempre la disposición de ayudarme, gracias mami bella!!!

A mis hermanos, Rita y Héctor, por siempre estar a mi lado, por todas las experiencias compartidas y por apoyarme en cada una de mis decisiones. A mi hermana porque a pesar de la distancia sé que cuento con ella para lo que sea.

A nuestra tutora, Prof. María Elena Villapol, por todo el apoyo dado durante el desarrollo de este trabajo, por su comprensión, paciencia y toda la confianza ofrecida. Gracias profe por entendernos y ayudarnos a alcanzar la meta de graduarnos en Diciembre.

A los profesores David Pérez y Karima Velásquez por su apoyo durante la realización de este trabajo, por la paciencia y toda la ayuda durante estos meses de presión y estrés, mil gracias por hacernos reír y darnos ánimos cuando lo necesitábamos.

A mi compañera de Tesis y amiga, Boni, por ser uno de los angelitos que Dios puso en mi camino, por su amistad sincera durante estos años en la UCV, por ser mi siamesa en la Facultad, por las horas incontables de trabajos, proyectos, risas y buenos recuerdos, mil gracias!!. También quiero agradecer a su familia por toda la ayuda ofrecida durante la realización de este trabajo.

A mis amigos, Luis y la china, mis otros angelitos, porque sin ellos este camino hubiese sido mil veces más difícil, por abrirme las puertas de sus hogares y hacerme sentir como en mi casa, por las incontables noches de estudio, colas y favores, simplemente por ser mis amigos incondicionales.

A mis amigas Cori, Lisseth, Jarcki, Silvy y Sof por ser más que mis amigas, mis hermanas, por ayudarme y ofrecerme su apoyo incondicional, por todos los buenos momentos y por siempre estar ahí cuando las necesito.

A mis compañeros Andres, Emilio, David, Andre, Auri, Mafer, Joe y Diana por todos los momentos compartidos, por los buenos consejos, por las tardes de chismes y risas y por siempre ofrecer su ayuda de forma desinteresada.

Para finalizar quisiera agradecer a mi casa de estudios la UCV, "la casa que vence las sombras" por estos años de formación académica y humana.

**A todos MUCHAS GRACIAS!!!
Mariana Colombo**

RESUMEN

La siguiente investigación se basó en diseñar un *testbed* para estudiar distintos protocolos de enrutamiento para redes malladas, y así poder determinar cuál de ellos podría ofrecer las mejores prestaciones para una red con diferentes números de dispositivos y especificando diversos valores de ancho de banda, al brindarle a la misma acceso a Internet a través de un cliente de la red que alcance la señal de un enrutador inalámbrico que se encuentre en una zona cercana y que cuente con dicho servicio.

Dicha investigación surge por la idea de utilizar la señal de enrutadores inalámbricos que se encuentren en urbanizaciones de Caracas para brindarle acceso a Internet a barrios aledaños, ya que muchos hogares en estas urbanizaciones, cuentan con este servicio y muchas veces sucede que la señal de los enrutadores inalámbricos es propagada más allá de la frontera del hogar de la persona que cuenta con este servicio y por lo tanto puede ser utilizada para brindar acceso a Internet a los usuarios en los barrios aledaños via una red mallada. El desarrollo de una solución como la descrita anteriormente debe comenzar por conocer que protocolo de enrutamiento podría ser el más adecuado para este diseño de red, de otra forma podría conllevar a una red con un pobre rendimiento que no cumpliría con los objetivos deseados.

Para dicha investigación se diseñó un *testbed* implementando la estructura de la red descrita anteriormente, donde todos los dispositivos trabajaban sobre una plataforma Linux (distribución Ubuntu 10.04) y se hizo uso de diferentes protocolos de enrutamiento como OLSR, BATMAN y Babel (protocolos de enrutamiento proactivos para redes malladas) y herramientas como D-ITG (generador de tráfico).

Se plantearon diferentes escenarios donde variaban aspectos como números de clientes y el ancho de banda disponible para la red mallada, y sobre ellos se realizaron diversas pruebas con el fin de determinar cuál es el protocolo de enrutamiento que puede ofrecer las mejores prestaciones para el funcionamiento de una red mallada en las condiciones planteadas.

Al analizar los resultados de las diversas métricas de desempeño obtenidas se pudo determinar que el protocolo que mejor se adapta a las condiciones planteadas fue Babel.

Palabras Claves: Redes malladas, Protocolos de enrutamiento, OLSR, Babel, BATMAN.

Índice de Contenido

Índice de Figuras.....	13
Índice de Tablas	17
1. Introducción.....	19
1.1. Planteamiento del problema	20
1.2. Objetivo general	20
1.2.1. Objetivos específicos	20
1.3. Justificación	21
1.4. Distribución del documento	21
2. Marco teórico	23
2.1. Redes Inalámbricas	23
2.1.1. Definición	23
2.1.2. Características	23
2.1.3. Clasificación	24
2.1.4. IEEE 802.11	25
2.2. Redes Malladas	28
2.2.1. Definición	28
2.2.2. Características	28
2.2.3. Arquitectura	29
2.2.4. Ventajas/Desventajas.....	31
2.2.5. Protocolos de Enrutamiento	32
3. Metodología, Herramientas y Métricas.....	39
3.1. Fases para el desarrollo	39
3.1.1. Escogencia y descripción de los escenarios	39
3.1.2. Implementación de los escenarios	39
3.1.3. Selección y ejecución de pruebas.....	39
3.1.4. Análisis de resultados	39
3.2. Herramientas utilizadas en el desarrollo	40
3.2.1. Hardware	40
3.2.2. Software	41
3.3. Métricas utilizadas en el desarrollo.....	42
3.3.1. Jitter	42
3.3.2. Throughput.....	42
3.3.3. Retardo (Delay).....	42
3.3.4. Paquetes Perdidos (Packet Loss).....	42
4. Implementación de los Escenarios y Ejecución de las Pruebas.....	43
4.1. Escogencia y descripción de los escenarios	44
4.2. Implementación de los escenarios.....	45
4.2.1. Configuración de la red	45
4.2.2. Control de ancho de banda.....	47
4.3. Selección y ejecución de pruebas	47

4.3.1. Selección de pruebas.....	47
4.3.2. Ejecución de pruebas.....	48
5. Resultados y Análisis de Resultados.....	51
5.1. Resultados – Escenarios 1, 2 y 3.	51
5.1.1. Throughput.....	51
5.1.2. Jitter	54
5.1.3. Retardo	57
5.1.4. Paquetes Perdidos.....	60
5.2. Resultados - Escenarios 4, 5 y 6	61
5.2.1. Throughput.....	61
5.2.2. Jitter	64
5.2.3. Retardo	67
5.2.4. Paquetes Perdidos.....	70
5.3. Resultados – Escenarios 7, 8 y 9	71
5.3.1. Throughput.....	71
5.3.2. Jitter	74
5.3.3. Retardo	77
5.3.4. Paquetes Perdidos.....	80
5.4. Resultados – Escenarios 1, 4 y 7	81
5.4.1. Throughput.....	81
5.4.2. Jitter	84
5.4.3. Retardo	87
5.4.4. Paquetes Perdidos.....	90
5.5. Resultados – Escenarios 2, 5 y 8	91
5.5.1. Throughput.....	91
5.5.2. Jitter	94
5.5.3. Retardo	97
5.5.4. Paquetes Perdidos.....	100
5.6. Resultados – Escenarios 3, 6 y 9	101
5.6.1. Throughput.....	101
5.6.2. Jitter	104
5.6.3. Retardo	107
5.6.4. Paquetes Perdidos.....	110
5.7. Resumen de los Resultados	111
6. Conclusiones.....	113
6.1. Contribuciones.....	113
6.2. Limitaciones.....	113
6.3. Recomendaciones	114
Referencias	115
Anexo A.....	119
Anexo B	123

Índice de Figuras

Figura 1. - Clasificación de las Redes Inalámbricas basada en el área de cobertura.	24
Figura 2. - Red Mallada con Infraestructura/Troncal.	29
Figura 3. - Red Mallada de Clientes.	30
Figura 4. - Red Mallada Híbrida.	31
Figura 5. - Script para sincronización.	41
Figura 6. - Elementos de los escenarios.	43
Figura 7. - Creación de una red Ad-Hoc parte I.	46
Figura 8. - Creación de una red Ad-Hoc parte II.	46
Figura 9. - Gráfica de <i>throughput</i> con paquetes de 64 bytes y 5 clientes.	51
Figura 10. – Gráfica de <i>throughput</i> con paquetes de 256 bytes y 5 clientes.	52
Figura 11. – Gráfica de <i>throughput</i> con paquetes de 1024 bytes y 5 clientes.	52
Figura 12. – Gráfica de <i>throughput</i> con paquetes de 1518 bytes y 5 clientes.	53
Figura 13. – Gráfica de <i>jitter</i> con paquetes de 64 bytes y 5 clientes.	54
Figura 14. – Gráfica de <i>jitter</i> con paquetes de 256 bytes y 5 clientes.	55
Figura 15. – Gráfica de <i>jitter</i> con paquetes de 1024 bytes y 5 clientes.	55
Figura 16. – Gráfica de <i>jitter</i> con paquetes de 1518 bytes y 5 clientes.	56
Figura 17. – Gráfica de retardo con paquetes de 64 bytes y 5 clientes.	57
Figura 18. – Gráfica de retardo con paquetes de 256 bytes y 5 clientes.	58
Figura 19. – Gráfica de retardo con paquetes de 1024 bytes y 5 clientes.	58
Figura 20. – Gráfica de retardo con paquetes de 1518 bytes y 5 clientes.	59
Figura 21. – Gráfica de paquetes perdidos para 5 clientes.	60
Figura 22. – Gráfica de <i>throughput</i> con paquetes de 64 bytes y 15 clientes.	61
Figura 23. – Gráfica de <i>throughput</i> con paquetes de 256 bytes y 15 clientes.	62
Figura 24. – Gráfica de <i>throughput</i> con paquetes de 1024 bytes y 5 clientes.	62
Figura 25. – Gráfica de <i>throughput</i> con paquetes de 1518 bytes y 15 clientes.	63
Figura 26. – Gráfica de <i>jitter</i> con paquetes de 64 bytes y 15 clientes.	64
Figura 27. – Gráfica de <i>jitter</i> con paquetes de 256 bytes y 15 clientes.	65
Figura 28. – Gráfica de <i>jitter</i> con paquetes de 1024 bytes y 15 clientes.	65
Figura 29. – Gráfica de <i>jitter</i> con paquetes de 1518 bytes y 15 clientes.	66
Figura 30. – Gráfica de retardo con paquetes de 64 bytes y 15 clientes.	67
Figura 31. – Gráfica de retardo con paquetes de 256 bytes y 15 clientes.	68
Figura 32. – Gráfica de retardo con paquetes de 1024 bytes y 15 clientes.	68
Figura 33. – Gráfica de retardo con paquetes de 1518 bytes y 15 clientes.	69
Figura 34. – Gráfica de paquetes perdidos para 15 clientes.	70
Figura 35. – Gráfica de <i>throughput</i> con paquetes de 64 bytes y 25 clientes.	71
Figura 36. – Gráfica de <i>throughput</i> con paquetes de 256 bytes y 25 clientes.	72
Figura 37. – Gráfica de <i>throughput</i> con paquetes de 1024 bytes y 25 clientes.	72
Figura 38. – Gráfica de <i>throughput</i> con paquetes de 1518 bytes y 25 clientes.	73
Figura 39. – Gráfica de <i>jitter</i> con paquetes de 64 bytes y 25 clientes.	74
Figura 40. – Gráfica de <i>jitter</i> con paquetes de 256 bytes y 25 clientes.	75
Figura 41. – Gráfica de <i>jitter</i> con paquetes de 1024 bytes y 25 clientes.	75
Figura 42. – Gráfica de <i>jitter</i> con paquetes de 1518 bytes y 25 clientes.	76
Figura 43. – Gráfica de retardo con paquetes de 64 bytes y 25 clientes.	77
Figura 44. – Gráfica de retardo con paquetes de 256 bytes y 25 clientes.	78
Figura 45. – Gráfica de retardo con paquetes de 1024 bytes y 25 clientes.	78

Figura 46. – Gráfica de retardo con paquetes de 1518 bytes y 25 clientes.....	79
Figura 47. – Gráfica de paquetes perdidos para 25 clientes.....	80
Figura 48. – Gráfica de <i>throughput</i> con paquetes de 64 bytes y 25% de ancho de banda.	81
Figura 49. – Gráfica de <i>throughput</i> con paquetes de 256 bytes y 25% de ancho de banda.....	82
Figura 50. – Gráfica de <i>throughput</i> con paquetes de 1024 bytes y 25% de ancho de banda.....	82
Figura 51. – Gráfica de <i>throughput</i> con paquetes de 1518 bytes y 25% de ancho de banda.....	83
Figura 52. – Gráfica de <i>jitter</i> con paquetes de 64 bytes y 25% de ancho de banda.....	84
Figura 53. – Gráfica de <i>jitter</i> con paquetes de 256 bytes y 25% de ancho de banda.....	85
Figura 54. – Gráfica de <i>jitter</i> con paquetes de 1024 bytes y 25% de ancho de banda.....	85
Figura 55. – Gráfica de <i>jitter</i> con paquetes de 1518 bytes y 25% de ancho de banda.....	86
Figura 56. – Gráfica de retardo con paquetes de 64 bytes y 25% de ancho de banda.	87
Figura 57. – Gráfica de retardo con paquetes de 256 bytes y 25% de ancho de banda. ..	88
Figura 58. – Gráfica de retardo con paquetes de 1024 bytes y 25% de ancho de banda. 88	
Figura 59. – Gráfica de retardo con paquetes de 1518 bytes y 25% de ancho de banda. 89	
Figura 60. – Gráfica de paquetes perdidos con 25% de ancho de banda.....	90
Figura 61. – Gráfica de <i>throughput</i> con paquetes de 64 bytes y 50% de ancho de banda.	91
Figura 62. – Gráfica de <i>throughput</i> con paquetes de 256 bytes y 50% de ancho de banda.	92
Figura 63. – Gráfica de <i>throughput</i> con paquetes de 1024 bytes y 50% de ancho de banda.....	92
Figura 64. – Gráfica de <i>throughput</i> con paquetes de 1518 bytes y 50% de ancho de banda.....	93
Figura 65. – Gráfica de <i>jitter</i> con paquetes de 64 bytes y 50% de ancho de banda.....	94
Figura 66. – Gráfica de <i>jitter</i> con paquetes de 256 bytes y 50% de ancho de banda.....	95
Figura 67. – Gráfica de <i>jitter</i> con paquetes de 1024 bytes y 50% de ancho de banda.....	95
Figura 68. – Gráfica de <i>jitter</i> con paquetes de 1518 bytes y 50% de ancho de banda.....	96
Figura 69. – Gráfica de retardo con paquetes de 64 bytes y 50% de ancho de banda.	97
Figura 70. – Gráfica de retardo con paquetes de 256 bytes y 50% de ancho de banda. ..	98
Figura 71. – Gráfica de retardo con paquetes de 1024 bytes y 50% de ancho de banda. 98	
Figura 72. – Gráfica de retardo con paquetes de 1518 bytes y 50% de ancho de banda. 99	
Figura 73. – Gráfica de paquetes perdidos con 50% de ancho de banda.....	100
Figura 74. – Gráfica de <i>throughput</i> con paquetes de 64 bytes y 100% de ancho de banda.	101
Figura 75. – Gráfica de <i>throughput</i> con paquetes de 256 bytes y 100% de ancho de banda.....	102
Figura 76. – Gráfica de <i>throughput</i> con paquetes de 1024 bytes y 100% de ancho de banda.....	102
Figura 77. – Gráfica de <i>throughput</i> con paquetes de 1518 bytes y 100% de ancho de banda.....	103
Figura 78. – Gráfica de <i>jitter</i> con paquetes de 64 bytes y 100% de ancho de banda.....	104
Figura 79. – Gráfica de <i>jitter</i> con paquetes de 256 bytes y 100% de ancho de banda... 105	
Figura 80. – Gráfica de <i>jitter</i> con paquetes de 1024 bytes y 100% de ancho de banda. 105	
Figura 81. – Gráfica de <i>jitter</i> con paquetes de 1518 bytes y 100% de ancho de banda. 106	
Figura 82. – Gráfica de retardo con paquetes de 64 bytes y 100% de ancho de banda. 107	

Figura 83. – Gráfica de retardo con paquetes de 256 bytes y 100% de ancho de banda.	
.....	108
Figura 84. – Gráfica de retardo con paquetes de 1024 bytes y 100% de ancho de banda.	
.....	108
Figura 85. – Gráfica de retardo con paquetes de 1518 bytes y 100% de ancho de banda.	
.....	109
Figura 86. – Gráfica de paquetes perdidos con 100% de ancho de banda.	110

Índice de Tablas

Tabla 1. - Descripción de las computadoras utilizadas.	40
Tabla 2. - Descripción del enrutador utilizado.	40
Tabla 3. - Descripción de la tarjeta inalámbrica utilizada.....	40
Tabla 4. - Tabla de Resultados	111

1. Introducción

Las redes malladas son una extensión de las redes Ad-Hoc, las cuales están formadas por una serie de dispositivos que se comunican de forma inalámbrica sin usar infraestructura. Estas redes han sido extendidas a través de la incorporación de enrutadores mallados que permiten la comunicación más allá de los límites existentes en las redes Ad-Hoc.

Por otra parte, hoy en día el crecimiento de la Internet y de la cantidad de información almacenada en los diversos sitios que forman esta red, han hecho que el acceso a la misma se haya vuelto una necesidad para una gran mayoría de personas en el mundo. Dicho acceso es provisto por los llamados Proveedores de Servicios de Internet (ISP) quienes usualmente cobran por prestar dicho servicio, siendo así no accesible para personas que cuentan con pocos recursos económicos, sobre todo aquellos que viven en las zonas más pobres de las ciudades.

La ciudad de Caracas tiene una distribución tal que una gran parte de urbanizaciones limitan con barrios. Las personas en las urbanizaciones suelen pagar por el servicio de acceso a Internet, el cual es prestado 24 horas al día por 365 días al año. Sin embargo este servicio es subutilizado durante las horas en que las personas usuarias están fuera de sus casas (trabajando o estudiando). Para la gente que vive en los barrios es más difícil pagar por este servicio, dado a sus restringidos recursos económicos. En este trabajo se propone una forma de aliviar este problema a través del uso de las redes malladas. La idea es que los potenciales usuarios de los barrios puedan acceder a la Internet a través de la conexión existente en las urbanizaciones aledañas. Esto se podría lograr a través del uso de enrutadores inalámbricos localizados en viviendas ubicadas en urbanizaciones que limitan con los barrios y cuyo alcance se extiende mas allá de este límite. La información de los usuarios en los barrios viajaría cruzando la red mallada hasta llegar al dispositivo de borde, representando por el enrutador inalámbrico mas cercano al barrio cuyo alcance vaya más allá del límite entre el barrio y la urbanización.

Con la finalidad de no desmejorar el servicio de los usuarios que están pagando por el mismo, se propone establecer un límite en el uso del ancho de banda de la conexión a los clientes de la red mallada, a través de un dispositivo ubicado en el borde entre el barrio y una o más urbanizaciones.

Como punto inicial de esta investigación y haciendo uso de una metodología incremental, en este trabajo se estudia el rendimiento de la red mallada acorde al uso de diversos protocolos de enrutamiento para redes Ad-Hoc cuando se cambian los valores de ancho de banda provisto por la conexión a Internet en el borde de la red.

1.1. Planteamiento del problema

A diferencia de las redes con infraestructura, en una red Ad-Hoc cada dispositivo tiene facilidades para poder enrutar la información. Los algoritmos de enrutamiento tradicionales no funcionan con este tipo de redes debido a su cambio frecuente en la topología, y a lo restringido del ancho de banda de los enlaces que conectan a los dispositivos, entre otros factores.

Muchas son las investigaciones realizadas en el desarrollo de los protocolos que funcionen en este tipo de redes y muchos también han sido los estudios comparativos de los protocolos propuestos. Sin embargo no se ha conseguido ningún estudio donde se compare el rendimiento de estos protocolos en una arquitectura como la planteada en este trabajo ni donde exista alguna limitación en el ancho de banda. En este trabajo se ha planteado cómo una red mallada basada en dispositivos conectados de forma Ad-Hoc podría soportar la conexión a Internet de usuarios localizados en barrios a través de enrutadores inalámbricos ubicados en los bordes entre el barrio y una o más urbanizaciones.

El desarrollo de una solución como la propuesta debe comenzar por conocer que protocolo de enrutamiento podría ser el más adecuado para este diseño de red. Por lo tanto, con esta investigación se pretende resolver la siguiente interrogante:

¿Cuál es el protocolo de enrutamiento que podría ofrecer las mejores prestaciones para el funcionamiento de una red mallada que cuente con acceso a Internet a través de un dispositivo en el borde con restricciones de ancho de banda?

1.2. Objetivo general

Estudiar el rendimiento de una red mallada con diferentes números de dispositivos usando varios protocolos de enrutamiento y especificando diversos valores de ancho de banda en la red mallada.

1.2.1. Objetivos específicos

Los objetivos específicos del trabajo son:

- Seleccionar las implementaciones de protocolos de enrutamiento existentes para redes malladas.
- Diseñar un *testbed* (cama de prueba) para estudiar los protocolos seleccionados.
- Realizar pruebas de rendimiento con cada uno de los protocolos seleccionados.
- Analizar los resultados para escoger el protocolo que presente un mejor rendimiento.

1.3. Justificación

Una solución como la propuesta en este documento no se puede implementar sin antes realizar los estudios de rendimiento de la red que determinen cual podría ser el comportamiento de la misma bajo ciertas restricciones de ancho de banda y usando diversos protocolos de enrutamiento. Una implementación sin la ejecución de pruebas como las planteadas en este trabajo podrían conllevar a una red con un pobre rendimiento que no cumpliría con los objetivos deseados.

1.4. Distribución del documento

Este documento consta de los siguientes siete capítulos:

- **Capítulo 1:** Se describe el problema de investigación, los objetivos que se pretenden alcanzar, la justificación para realizar esta investigación y la distribución del documento.
- **Capítulo 2:** Contiene el marco teórico, donde se tocan temas como las redes inalámbricas, redes malladas, su funcionamiento y los protocolos de enrutamiento existentes para este tipo de redes.
- **Capítulo 3:** Se describe la metodología a seguir, así como también los elementos de software y hardware y las métricas necesarias para el desarrollo de esta investigación.
- **Capítulo 4:** Se describe en detalle el diseño del *testbed*, los escenarios planteados y las pruebas que permitieron medir el desempeño en cada escenario para poder lograr los objetivos y resolver el problema de investigación.
- **Capítulo 5:** Contiene tablas comparativas y gráficos de todos los resultados obtenidos de las pruebas realizadas para cada escenario y el análisis de los mismos.
- **Capítulo 6:** Presenta las conclusiones, contribuciones, limitaciones y recomendaciones para futuras investigaciones.

2. Marco teórico

2.1. Redes Inalámbricas

2.1.1. Definición

Por red inalámbrica se entiende una red que utiliza ondas electromagnéticas para propagar la información, en lugar de un cableado estándar, brindando así la funcionalidad y beneficios de las redes convencionales, sin la limitante de los medios guiados. Es por esto que tanto la transmisión como la recepción de la información se hacen a través de antenas.

Estas redes surgen de la necesidad de poder conectar diferentes dispositivos sin el uso de engorrosos cables, de tener que abrir agujeros en las paredes, o de modificar de manera significativa la infraestructura de los lugares, permitiendo así la conexión sin problemas de dispositivos remotos. Cabe destacar que esto ha sido una gran influencia en el rápido crecimiento y alta demanda de este tipo de redes.

2.1.2. Características

Las características de las redes inalámbricas se presentan a continuación [1]

- Este tipo de redes son muy flexibles pues dentro de la zona de cobertura de la red inalámbrica los nodos se pueden comunicar sin estar atados a un cable.
- Como las ondas que transportan la información, se transmiten a través de medios no guiados, cualquiera con un dispositivo adecuado para la recepción de las mismas podría escuchar lo que se está enviando por la red, lo que se convierte en una de las mayores desventajas de estas redes en cuanto a seguridad. Sin embargo, se han diseñado diferentes mecanismos para contrarrestar esta desventaja, como la encriptación de los datos, entre otras.
- El espectro electromagnético por donde se propagan las ondas se ven reguladas por entes especializados, ya que aunque estas ondas se transmiten a través de diferentes dispositivos, ya sean de uso personal, científico, militar, etc., todas son propensas a interferencias, por lo que los países cuentan con regulaciones que definen los rangos de frecuencia y potencias de transmisión para cada caso.
- Las redes inalámbricas permiten a los usuarios mantenerse conectados mientras se desplazan dentro de determinadas áreas geográficas, por lo cual muchas veces el término de “movilidad” viene asociado con estas redes.
- Estas redes requieren de una menor planificación a la hora de decidir la distribución física de los equipos. No es necesario ver donde poner las computadoras para que alcance el cable o este tipo de problema.

2.1.3. Clasificación

Las redes inalámbricas pueden clasificarse de acuerdo a diferentes criterios. Como se puede apreciar en la Figura 1 basado en el tamaño del área física de cobertura de la red como criterio, se pueden clasificar en:

- **Red Inalámbrica de Área Personal (Wireless Personal-Area Network/ PAN):** Este tipo de redes son usadas generalmente para conectar dispositivos periféricos, como impresoras, teléfonos celulares, electrodomésticos, etc. Tienen un alcance de aproximadamente 10 metros.
- **Red Inalámbrica de Área Local (Wireless Local-Area Network/LAN):** Este tipo de redes tienen mayor alcance que las redes PAN, como para cubrir un edificio completo. Son utilizadas principalmente para interconectar computadores personales y estaciones de trabajo. Estas redes son siempre privadas y se caracterizan por ser pequeñas y con altas velocidades.
- **Red Inalámbrica de Área Metropolitana (Wireless Metropolitan-Area Network/MAN):** Estas redes básicamente son como una versión más grande de una red LAN y normalmente utilizan tecnologías similares. Pueden ser públicas o privadas y soportan tanto voz como datos. Su alcance es suficiente para cubrir una ciudad, es decir, desde cientos de kilómetros a miles de kilómetros.
- **Red Inalámbrica de Área Amplia (Wireless Wide-Area Network /WAN):** Estas redes cubren una amplia región geográfica, a menudo países o hasta continentes. Tienen el alcance más amplio de todas las redes mencionadas anteriormente.

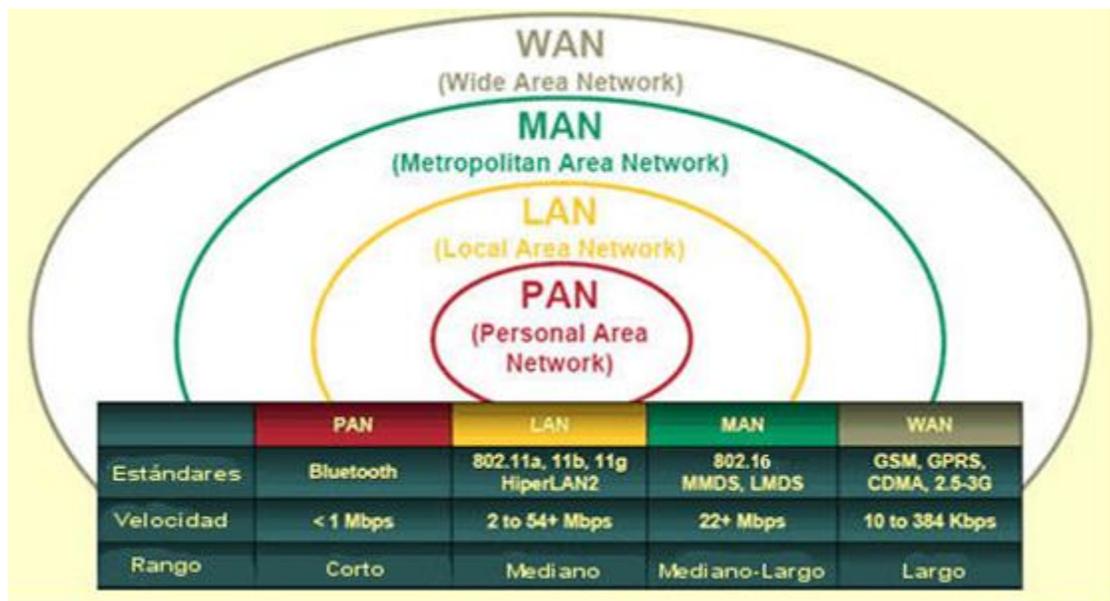


Figura 1. - Clasificación de las Redes Inalámbricas basada en el área de cobertura.

Usando como criterio la infraestructura de la red, las redes inalámbricas también pueden clasificarse en redes con infraestructura y sin infraestructura.

- **Redes con Infraestructura:** En las redes con infraestructura los nodos de la red precisan de un elemento de coordinación, como un punto de acceso (AP) o estación base para comunicarse con el resto de los nodos. Si el punto de acceso se conecta a una red cableada, los clientes inalámbricos pueden acceder a la red fija a través del punto de acceso.
- **Redes sin Infraestructura (Ad-Hoc):** Podemos definir una red inalámbrica Ad-Hoc como una red inalámbrica compuesta por dos o más computadoras, conectadas entre ellas sin necesidad de usar dispositivos adicionales como enrutadores o puntos de acceso, sino usando las tarjetas inalámbricas que tengan instaladas los propios equipos. En una red Ad-Hoc el rendimiento es menor a medida que el número de nodos crece. Para conectar una red Ad-Hoc a una red de área local (LAN) cableada o a Internet, se requiere instalar una Pasarela o Gateway especial.

2.1.4. IEEE 802.11

Este estándar originalmente concebido como parte del estándar IEEE 802.4 de Token Bus por la IEEE (Institute of Electrical and Electronics Engineers), posteriormente fue creado como un grupo de trabajo diferente que define el funcionamiento de la capa física y de enlace de datos en una WLAN [2]. Existen múltiples versiones de este estándar, con la ventaja de que todas son compatibles entre sí, y tienen en común ciertas características. Las redes descritas por este estándar cuentan con cuatro componentes físicos principales:

- **Sistema de Distribución:** Cuando varios puntos de acceso están conectados para formar una gran área de cobertura, deben comunicarse los unos con los otros para poder rastrear los movimientos de las estaciones móviles. El sistema de distribución es el componente lógico del estándar 802.11 usado para enviar las tramas a su destino. Este estándar no especifica ninguna tecnología particular para el sistema de distribución. En la mayoría de los productos comerciales el sistema de distribución es implementado a través de una red troncal, como por ejemplo Ethernet.
- **Puntos de Acceso:** Las tramas en una red 802.11 deben ser convertidas en otro tipo de tramas para que puedan ser enviadas al resto del mundo. Los dispositivos llamados puntos de acceso realizan la función de transformar tramas inalámbricas a cableadas (además de otras funciones).
- **Medio Inalámbrico:** Para mover tramas de estación en estación, el estándar utiliza un medio inalámbrico. Existen varias capas físicas definidas para utiliza este medio.

- **Estaciones:** Las redes están construidas para transferir datos entre estaciones. Las estaciones son dispositivos de cómputo con interfaces de red inalámbricas. Típicamente las estaciones son laptops que operan con baterías, u otros dispositivos móviles, aunque no hay razón por la que una estación tenga que ser un dispositivo portable; sin embargo la mayoría de las estaciones utilizan las redes inalámbricas para aprovechar la movilidad que estas brindan.

Este estándar define el uso de tres capas físicas diferentes para las comunicaciones inalámbricas [6]: Infrarrojo, FHSS y DSSS.

Infrarrojo (IR)

Existe una especificación de IEEE 802.11 que permite obtener velocidades de 1Mbps y 2Mbps, al hacer rebotar la luz en techos y paredes para proveer conectividad a un cuarto o una pequeña oficina. Esta especificación tiene mucho tiempo vigente, pero no es muy popular ya que no se conocen fabricantes que hagan productos compatibles con esta tecnología.

Frequency Hopping Spread Spectrum (FHSS)

Bajo este esquema una estación transmite a través de una frecuencia dada durante un corto periodo de tiempo y luego salta a otra frecuencia para continuar con la comunicación. El algoritmo de saltos de frecuencia es conocido por todas las estaciones. A través de un número pseudo aleatorio conocido solo por la estación emisora y la receptora, la cual utiliza como una variable para el algoritmo, calculan el conjunto de frecuencias que utilizará durante la transmisión.

Direct Sequence Spread Spectrum (DSSS)

Esta es otra técnica de transmisión generada por los militares para evitar la interferencia de los enemigos, la cual consiste en la generación de un patrón de bits redundante, para cada uno de los bits que componen la señal de información que será enviada y la posterior modulación de la señal resultante mediante una portadora. En recepción es necesario realizar el proceso inverso para obtener la señal de información original.

Una red mallada Wi-Fi permite que diferentes dispositivos se puedan conectar y comunicar entre ellos de forma inalámbrica y para operar sólo necesitan clientes ordinarios 802.11.

Las siguientes son algunas de las especificaciones del estándar 802.11:

- 802.11a

Este estándar define una serie de nuevos métodos de modulación que permiten tasas de transmisión de datos de hasta 54Mbps. Las tasas de transmisión más altas se obtienen usando OFDM (Orthogonal Frequency Division Multiplexing), una técnica en la cual la frecuencia es subdividida en canales que son modulados

individualmente. El estándar 802.11a trabaja en la frecuencia de 5GHz. Esto quiere decir que equipos que soporten este estándar no son totalmente compatibles con el estándar 802.11 básico, ya que este opera en la banda de frecuencia de 2.4 GHz. Además como las frecuencias más altas se atenúan más rápido que las frecuencias bajas, tiene como resultado que las estaciones inalámbricas 802.11a tengan un rango menor que las que trabajan en frecuencias de 2.4GHz. Esto requiere que se tenga una mayor organización de puntos de acceso de la que se tendría con el estándar 802.11 para cubrir un área geográfica similar.

- 802.11b

Bajo este estándar DSSS fue usado con dos métodos de modulación nuevos, para poder alcanzar tasas de transferencias de datos de 11Mbps y 5.5Mbps. El estándar 802.11b también provee compatibilidad con los equipos 802.11 DSSS que operan a 2Mbps o 1Mbps. Para proveer esta compatibilidad el estándar 802.11b especifica el uso de frecuencia de 2.4GHz.

- 802.11g

Comparando los estándares anteriores con el estándar 802.11 ambos tienen sus ventajas y desventajas. Aunque el 802.11a tenga tasas de transferencia de datos superiores su uso en la frecuencia de 5GHz hace que tenga una corta distancia de transmisión, mientras que con el estándar 802.11b provee una mayor distancia de transmisión pero una menor tasa de transmisión de datos. Combinando el método de modulación usado en el estándar 802.11a con la frecuencia utilizada en el estándar 802.11b, la IEEE proporcionó un mecanismo para extender tanto las tasas de datos como la distancia de transmisión de las redes inalámbricas LAN, resultando en el estándar 802.11g. Para mantener compatibilidad con versiones anteriores de la gran cantidad de equipos 802.11b el estándar 802.11g también soporta operaciones DSSS a 11, 5.5, 2 y 1Mbps.

- 802.11n

IEEE 802.11n fue diseñado para mejorar significativamente el desempeño de la red más allá de los estándares anteriores, tales como 802.11b y 802.11g, con un incremento significativo en la velocidad máxima de transmisión. Está construido basándose en estándares previos de la familia 802.11, agregando Multiple-Input Multiple-Output (MIMO) y unión de interfaces de red (Channel Bonding), además de agregar tramas a la capa MAC. MIMO usa múltiples antenas transmisoras y receptoras para mejorar el desempeño del sistema. MIMO es una tecnología que usa múltiples antenas para manejar mayor información (cuidando la coherencia) que la posible con una sola antena. Dos beneficios importantes que provee a 802.11n son la diversidad de antenas y el multiplexado espacial.

2.2. Redes Malladas

2.2.1. Definición

Las redes malladas son redes de tipo Ad-Hoc en las que cada nodo provee conectividad a nodos adyacentes, conectándose entre sí formando una especie de malla. Las redes malladas consisten de dos tipos de nodos, enrutador mallado y cliente mallado [3]. La principal diferencia entre un enrutador convencional y un enrutador mallado, es que este último puede lograr la misma cobertura con menor potencia de transmisión; además también cuentan con información de enrutamiento adicional para soportar redes malladas. Los clientes mallados, también tienen ciertas funciones malladas específicas, lo cual permite que en algún momento dado también puedan comportarse como enrutadores.

Este tipo de redes se auto-configuran estableciendo los enlaces entre los nodos y manteniendo la conectividad, lo cual se hace cada cierto tiempo ya que puede aparecer o desaparecer algún nodo en la red. Dado que brindan un servicio no centralizado las redes malladas son normalmente usadas para brindar servicio en zonas que por alguna razón no pueden ser cubiertas por compañías que brindan servicios de Internet.

Cualquier nodo inalámbrico, como computadores o teléfonos celulares puede convertirse en un nodo mallado simplemente mediante modificaciones de software.

Un detalle que es importante considerar es la ubicación de los nodos enrutadores, ya que si la red mallada solo posee un punto de acceso a Internet se debe tratar de ubicar lo más centrado posible para que así el número de saltos desde el hasta cada uno de los clientes sea similar y evitar generar un bajo rendimiento para aquellos nodos que estén muy alejados de él.

2.2.2. Características

A continuación se describen las características de las redes malladas [3]

- Cada nodo se puede comunicar con cualquier otro, independiente del punto de acceso, es decir, no es necesario que todas las estaciones de usuario tengan línea de vista con el punto de acceso.
- En una red mallada, tanto la formación como la organización son automáticas y sin intervención manual.
- La movilidad depende del tipo de nodo mallado. Los clientes mallados pueden ser nodos fijos o móviles.
- Estas redes tienen la posibilidad de conectarse a uno o más puntos de acceso por lo que se puede mejorar el rendimiento de la red y las hace mucho más estables, ya que si en algún momento un punto de acceso falla los nodos hallaran el camino hacia otro.

2.2.3. Arquitectura

Se pueden observar tres categorías diferentes de arquitecturas basadas en la funcionalidad de los nodos [5].

- Redes Malladas con Infraestructura/Troncal

En este tipo de arquitectura interactúan dos tipos de dispositivos, los enrutadores mallados y los clientes mallados. Se dice que se crea una red mallada con infraestructura cuando varios enrutadores mallados mediante la autoconfiguración crean enlaces entre ellos formando una red troncal a la que se pueden conectar los diferentes tipos de clientes, como se puede ver en la Figura 2, donde las líneas continuas representan enlaces cableados y las líneas punteadas representan los enlaces inalámbricos. Los enrutadores mallados pueden utilizar diferentes tecnologías de radio y así si los clientes usan estas mismas tecnologías pueden conectarse directamente a ellos, sino podrían hacerlo a través de conexión Ethernet con estos mismos o con las estaciones bases que se conectan a los enrutadores mallados. Este tipo de topología permite que se conecten clientes convencionales y además permite la integración de otras redes malladas a través de las funcionalidades de los enrutadores mallados.

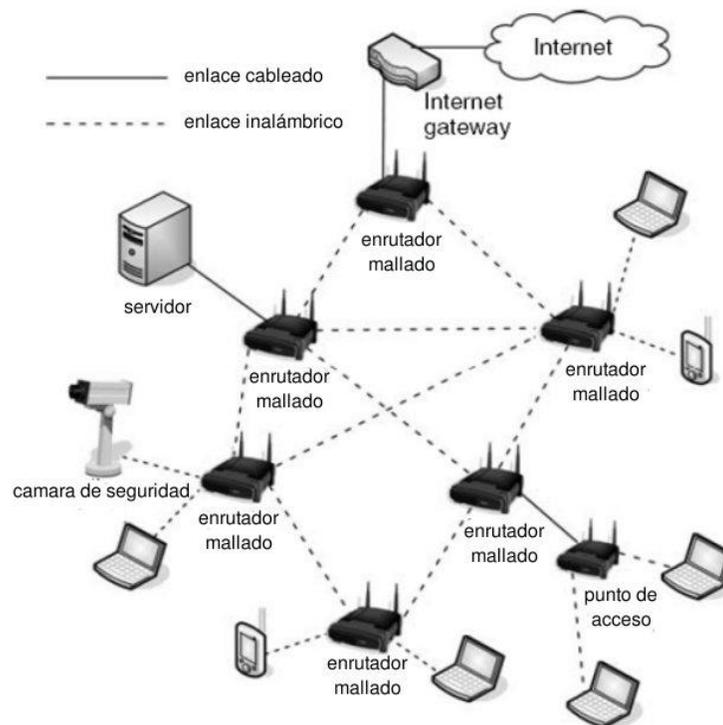


Figura 2. - Red Mallada con Infraestructura/Troncal.

- Redes Malladas de Clientes

En este tipo de arquitectura, la red mallada solo está conformada por clientes mallados y estos forman redes “peer to peer” realizando funciones de configuración y enrutamiento. Los paquetes son entregados a su destino saltando a través de otros dispositivos. Esto se puede observar en la Figura 3.

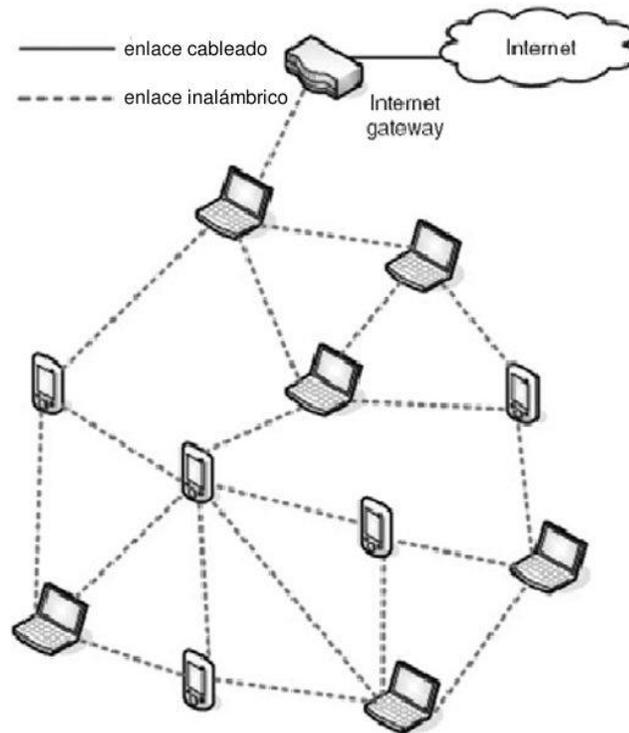


Figura 3. - Red Mallada de Clientes.

- Redes Malladas Híbridas

Como se puede apreciar en la Figura 4, este tipo de arquitectura es una combinación de las redes malladas con infraestructura y las redes malladas de clientes. Donde los clientes pueden conectarse con otros a través de los enrutadores mallados o a través de otros clientes mallados. Mientras que la infraestructura proporciona conectividad con otras redes como la Internet, la capacidad de enrutamiento de los clientes mejora la conectividad y la cobertura en el interior de las redes malladas.

2.2.5. Protocolos de Enrutamiento

El término enrutamiento se refiere al proceso necesario para determinar el camino o ruta de comunicación entre dos nodos de una red. Para ello se utilizan protocolos que se encargan de determinar dichas rutas y así poder establecer comunicación entre la fuente y el destino a través de los enrutadores. Para esto se deben realizar ciertas tareas:

- Descubrir nodos: estar al tanto de nuevos nodos que aparezcan en la red o de nodos que desaparezcan.
- Descubrir la frontera: estar al tanto de donde se encuentran los límites de una red que normalmente son los nodos donde se encuentra el acceso a Internet.
- Mediciones del enlace: calcular la calidad de los enlaces y de las rutas, los parámetros para determinar dicho valor puede variar dependiendo del protocolo.
- Asignación de rutas: usando los valores de calidad del enlace, asignar la mejor ruta.
- Manejo de conexiones a redes externas (Internet).

Una red mallada es una tecnología relativamente nueva, por lo que hoy en día, los protocolos de enrutamiento que deben ser usados en estas redes siguen siendo tema de investigación.

En la actualidad los protocolos de enrutamiento pueden ser clasificados de diferentes maneras, una de ellas es en protocolos proactivos y reactivos. Los *protocolos proactivos* están basados en tablas de información que posteriormente los nodos intercambian de forma periódica para aprender la topología de la red, por ejemplo están los protocolos OLSR, MMRP, OSPF, HWMP y BATMAN. En los *protocolos reactivos* los nodos sólo intercambian información de control cuando quieren iniciar una comunicación con otro nodo como por ejemplo los protocolos AODV y DSR.

2.2.5.1. Protocolos Proactivos

A continuación se describen los protocolos proactivos estudiados en este trabajo.

Better Approach to Mobile Ad-Hoc Networking (BATMAN)

El protocolo de enrutamiento BATMAN [8] es uno de los más populares cuando se trata de una red mallada y se basa en mantener información sobre todos los nodos de la red que son accesibles en uno o varios saltos.

La estrategia de este protocolo es determinar para cada nodo de la red, un vecino de un salto, el cual puede ser utilizado como mejor gateway para comunicarse con dicho nodo de destino. Es decir la tabla de rutas de un nodo debe contener un gateway para cada nodo de la red. El algoritmo BATMAN se enfoca exclusivamente en aprender sobre el mejor próximo salto para cada destino. No hay necesidad de encontrar o calcular la ruta completa, lo que hace posible una implementación muy rápida y eficiente.

Para el cálculo de la mejor ruta, BATMAN realiza un análisis estadístico de la pérdida de paquetes del protocolo y velocidad de propagación, y no depende del estado o información topológica de otros nodos. Los paquetes del protocolo BATMAN contienen sólo una limitada cantidad de información y por lo tanto son muy pequeños. El enfoque del algoritmo BATMAN es dividir el conocimiento sobre el mejor camino, entre los nodos de la red. Cada nodo percibe y mantiene sólo la información sobre el próximo mejor salto hacia todos los otros nodos. De ahí que la necesidad del conocimiento global acerca de los cambios en la topología local es innecesaria.

El algoritmo del protocolo BATMAN puede ser descrito de la siguiente manera: como primer paso cada nodo transmite mensajes de broadcast, llamados Mensajes Originadores (OGM), para informar a los nodos vecinos de su existencia. Estos vecinos que reciben los OGMs los retransmiten de acuerdo a reglas específicas del protocolo BATMAN. Quedando así toda la red inundada con mensajes originadores. Este proceso de inundación será después realizado por los vecinos de un salto como segundo paso, luego por los vecinos de dos saltos como tercer paso y así sucesivamente hasta que cada nodo haya recibido los mensajes OGMs al menos una vez, hasta que se pierdan debido a la pérdida de paquetes o hasta que los valores de TTL expiren.

Los OGMs, son pequeños, siguiendo su típico tamaño de 52 bytes incluyendo el sobre encabezado IP y UDP. Los OGMs, contienen al menos la dirección del originador, la dirección del nodo retransmitiendo el paquete, un tiempo de vida (TTL) y un número de secuencia.

Para poder encontrar la mejor ruta hacia un originador en particular BATMAN cuenta los OGMs recibidos y guarda de que vecino fue recibido. Usando esta información BATMAN mantiene una tabla con el mejor siguiente salto hacia cada originador de la red. Para asegurar que cada OGM es contado una sola vez, y no hay duplicados, se utiliza un número de secuencia embebido en los mensajes originadores.

La cantidad de OGMs recibidos de un originador en específico a través de cada vecino también sirven para estimar la calidad de una ruta (de uno o más saltos).

Optimized Link State Routing Protocol (OLSR)

OLSR [7] es un protocolo para redes móviles Ad-Hoc y es una optimización del clásico algoritmo de estado del enlace pero diseñado para redes inalámbricas. El punto clave es que se introduce el uso de “multipoint relays” (MPR) que no son mas que nodos seleccionados para reenviar paquetes en la red, lo que reduce substancialmente las transmisiones requeridas y por ende el tráfico en comparación con otros mecanismos en los que todos los nodos reenvían los paquetes cuando son recibidos por primera vez. OLSR se considera un protocolo eficiente por dos razones, primero porque la información sobre el estado de la red solo es reenviada por nodos seleccionados como MPRs y segundo porque en este protocolo se trata de minimizar el número de mensajes de control necesarios. Este protocolo se basa en intercambiar regularmente información sobre la topología de la red con otros nodos de la misma. Cada nodo selecciona un conjunto de

sus nodos vecinos como MPRs. Los nodos que han sido seleccionados como MPRs por algún vecino lo notifican periódicamente a través de sus mensajes de control.

Por otra parte OLSR solo requiere información parcial del estado del enlace y lo mínimo que se requiere es que todos los nodos seleccionados como MPRs deben declarar las rutas de sus selectores. Para optimizar la reacción que tiene este protocolo en cuanto a cambios en la topología de la red se puede reducir el intervalo de tiempo en el que se deben enviar mensajes de control. Para saber cual mensaje de control es el más reciente, cada mensaje contiene un número de secuencia, el cuál es incrementado con cada mensaje. Básicamente cada nodo de la red selecciona un conjunto de nodos que estén a un salto de distancia y que puedan retransmitir sus mensajes. Este conjunto de nodos vecinos es llamado conjunto MPR. Los vecinos de un nodo N que no estén en el conjunto MPR reciben y procesan mensajes pero no retransmiten mensajes recibidos desde el nodo N. Cada nodo mantiene información sobre el conjunto de vecinos que lo han seleccionado como MPR.

Este grupo es llamado conjunto de selectores MPR de un nodo. Un nodo obtiene esta información desde mensajes "hello" que son enviados periódicamente desde sus vecinos.

OLSR se compone de un núcleo y un conjunto de funciones auxiliares. El núcleo es lo mínimo necesario para que el protocolo opere y provea enrutamiento en una red sin infraestructura. Cada función auxiliar provee funcionalidades adicionales que pueden ser útiles en escenarios específicos. Por ejemplo, un nodo quiere proveer conectividad entre una red sin infraestructura y otra red.

El núcleo OLSR especifica el comportamiento de un nodo equipado con una interfaz OLSR y que es parte de una red sin infraestructura y que trabajan con este protocolo. Esto incluye una especificación universal de los mensajes del protocolo OLSR y su transmisión a través de la red, como también los enlaces que existan, la topología de la red y el cálculo de rutas.

Específicamente el núcleo está compuesto por:

Formato de paquetes y reenvío: Se especifica el formato de los paquetes usados en este protocolo y un mecanismo optimizado de inundación como mecanismo de transporte para el control del tráfico. OLSR funciona usando un formato único de paquete para toda la data relacionada con este protocolo. Estos paquetes son embebidos en datagramas UDP para transmitirlos en la red. Cada paquete encapsula uno o más mensajes y todos los mensajes comparten un formato de cabecera, lo que facilita a los nodos reconocer acertadamente los mensajes que correspondan a este protocolo. Cada paquete contiene un campo llamado "Paquet Sequence Number" y este debe ser incrementado cada vez que se transmite un nuevo paquete OLSR. También posee otro campo llamado "Originator Address" el cual indica la dirección principal del nodo que originó el mensaje, esta dirección no debe ser confundida con la dirección fuente la cuál es cambiada cada vez que el mensaje es retransmitido, en cambio el campo "Originator Address" nunca debe cambiar. Con el campo "Time to live" el nodo que origina el mensaje puede limitar el radio de alcance del mensaje. Para evitar el reprocesamiento de paquetes cada nodo mantiene información acerca de los mensajes que han sido recibidos recientemente en el "Duplicate Set".

Los tipos de mensajes requeridos por el protocolo son:

Mensajes "hello": se encargan de descubrir los enlaces entre los nodos, detectar los vecinos y señalar los nodos MPR.

Mensaje TC: se encargan de difundir información sobre la topología de la red.

Mensajes MID: se encargan de declarar la presencia de nodos con múltiples interfaces.

Mensajes HNA: se encargan de proveer un mecanismo para inyectar información sobre redes externas al dominio OLSR.

Estado del enlace: Esto se logra a través de la emisión de mensajes "hello" a través de cada una de las interfaces activas. Como resultado de esto se obtiene el conjunto local de enlaces, describiendo enlaces entre interfaces locales e interfaces remotas. Cada nodo debe detectar los enlaces entre sus nodos vecinos y él, y debe asegurarse que sea un enlace bidireccional. Toda esta información se guarda en el "Link Set". Cuando un nodo recibe un mensaje "hello", el nodo debe actualizar su "Link Set". Los mensajes "hello" no deben ser retransmitidos ni tampoco se debe guardar información de los mismos en el "Duplicate Set".

Detección de vecinos: Dada una red con nodos que posean una sola interfaz de red, estos podrían deducir su conjunto de vecinos luego de conocer el estado del enlace. La dirección principal de un nodo con una sola interfaz de red, es la dirección de la única interfaz de ese nodo. Por otra parte, para una red con nodos que posean más de un interfaz de red, se requiere información adicional para mapear direcciones de interfaces a una dirección principal y dicha información se obtiene a través del intercambio de mensajes MID. Dichos mensajes deben contener las direcciones de todas las interfaces de un nodo exceptuando la dirección principal que va colocada en el campo "Originator Address". Aquellos nodos que solo tengan una interfaz no deben generar mensajes MID. Cada nodo con múltiples interfaces debe enviar periódicamente a los otros nodos de la red información describiendo la configuración de sus interfaces. Esto se logra inundando la red con mensajes MID usando el mecanismo característico de este protocolo, es decir, usando nodos MPR. Cada nodo de la red mantiene información acerca de los otros nodos y es usada para el cálculo de rutas.

Selección y señalización MPR: Cada nodo realiza una selección MPR. Esto tiene como objetivo seleccionar un sub-conjunto de sus vecinos, de tal forma que si un mensaje es transmitido por difusión, sea solo retransmitido por ese sub-conjunto seleccionado y sea recibido por todos los nodos que estén a dos saltos de distancia. El conjunto MPR de un nodo es formado de tal forma que se cumpla esta condición para cada una de las interfaces de un nodo. La señalización MPR es una de las tareas que cumple los mensajes "hello". Cada interfaz de un nodo de la red selecciona su propio conjunto de nodos MPRs entre sus vecinos y esta selección es notificada a sus vecinos con un mensaje "hello". Los nodos que reciben un mensaje "hello" notificando que fue escogido como un MPR deben guardar esta información en su "MPR Selector Set".

Difusión de mensajes de control de la topología: Estos mensajes son difundidos con el propósito de proveer a cada nodo de la red con información suficiente sobre el estado de la red para generar el cálculo de rutas. Conocer el estado de los enlaces en la red ofrece a cada nodo de la red obtener una lista de nodos con los cuales puede comunicarse directamente y, en combinación con el formato del paquete y el reenvío de los mismos realizar una optimización del mecanismo de propagación de mensajes por la red a través de los MPRs. Basado en esto, información sobre la topología de la red es diseminada por toda la red a través de mensajes TC. Un mensaje TC es enviado por un nodo para declarar el conjunto de enlaces al menos a los nodos que están en su "MPR Selector Set", es decir a los nodos que lo han seleccionado como MPR y dichos mensajes contienen una lista de direcciones principales de sus nodos vecinos.

Cálculo de rutas: Dada la información sobre el estado del enlace que se obtuvo a partir del intercambio de mensajes entre nodos se puede proceder al cálculo de la tabla de rutas para cada nodo. Dicha tabla de rutas se actualiza cada vez que cambia algo sobre los enlaces, topología, conjunto de vecinos, etc. Para construir o actualizar la tabla de rutas de un nodo X se hace uso de un algoritmo para calcular el camino más corto entre el resto de nodos de la red.

Funciones auxiliares: A parte de las funcionalidades que se encuentran en el núcleo OLSR, hay veces en donde es necesario algunas funciones adicionales. Por ejemplo, en situaciones donde un nodo tiene más de una interfaz y algunas de ellas pertenece a otro dominio de rutas y para poder inyectar información de ruteo externa a la red mallada, un nodo con alguna interfaz hacia una red cableado por ejemplo, envía periódicamente mensajes HNA, que contienen suficiente información para que los receptores construyan una tabla de rutas apropiada. Un mensaje HNA contiene una lista de direcciones de red con su respectiva máscara de subred. Se debe ofrecer cualquiera de las funcionalidades auxiliares cuando sean requeridas.

Protocolo Babel

Babel [9] es un protocolo proactivo para redes móviles Ad-Hoc como también para redes cableadas, basado en el algoritmo vector-distancia. Este protocolo tiene dos características distintivas. La primera es que hace una selección de rutas tomando en cuenta el historial previo para minimizar el impacto en la red cuando se anuncia como llegar a un destino vía una ruta y luego otra. En la situación en la que un nodo cambia continuamente su ruta preferida entre el origen y el destino puede conllevar a un enrutamiento inestable. Por esto, cuando existe más de una ruta con la calidad del enlace similar, la selección de rutas favorece al camino previamente establecido en vez de alternar entre dos rutas. La segunda es que Babel ejecuta una actualización reactiva y fuerza una petición de información de rutas cuando detecta una falla en algún enlace de cualquiera de sus vecinos favoritos. Dado que las mediciones de calidad del enlace fueron previamente completadas en la etapa inicial, Babel asegura tener un tiempo de convergencia de rutas casi inmediato cuando se ejecuta una actualización específica.

En este protocolo, a cada nodo de la red se le asigna un identificador único, que es incluido en las actualizaciones de rutas que envíe cada nodo. Cuando un nodo reciba una solicitud de un nuevo número de secuencia, este aumentará el número de identificador.

Los paquetes de este protocolo son enviados en el cuerpo de un datagrama UDP destinados a una dirección multicast o unicast. Los paquetes que no pertenezcan a la red local o que el puerto de origen sea diferente al establecido son descartados.

En este protocolo cada nodo cuenta con distintas tablas para guardar información útil para el funcionamiento del mismo. Una de ellas es la tabla de interfaces, que contiene la lista de interfaces en donde el protocolo Babel está corriendo. Cada entrada en esta tabla contiene un número de secuencia que es enviado en cada mensaje "hello" por esa interfaz y es incrementado cada vez que es enviado. Posee dos temporizadores asociados con cada entrada de esta tabla, el temporizador "hello" que regula el envío periódico de mensajes "hello" y mensajes IHU, y el temporizador "update" que regula el envío periódico de actualización de rutas.

La tabla de vecinos contiene una lista de todas las interfaces vecinas desde donde han sido recibidos recientemente paquetes de este protocolo. La relación de esta tabla se basa en interfaces y no en nodos. Por lo tanto dos nodos con múltiples interfaces pueden participar en múltiples relaciones de vecinos.

La tabla de rutas contiene todas las rutas conocidas por el nodo. Cada ruta tiene asociado un prefijo. Además cuenta con temporizador de expiración de rutas.

En este protocolo cada nodo envía mensajes "hello" sobre cada una de sus interfaces, cada uno identificado con un número de secuencia, periódicamente. También cada nodo envía periódicamente mensajes IHU ("I Heard You" - "Te escuché") a cada uno de los vecinos, de los cuales ha recibido mensajes "hello" recientemente. A partir de la información recibida de los mensajes "hello" y los mensajes IHU de un nodo B, por ejemplo, un nodo A podría calcular el costo del enlace entre A y B.

Cada nodo anuncia a sus vecinos su selección de rutas enviando paquetes de actualización en todas sus interfaces. Estos anuncios también pueden forzarse a través de una solicitud de rutas, la cual lleva asociado el prefijo de la ruta seleccionada. El nodo que reciba esta solicitud de ruta, revisará su tabla de rutas y buscará una entrada con este prefijo y si la ruta existe entonces enviará un paquete de actualización de rutas.

3. Metodología, Herramientas y Métricas

Para el estudio de los protocolos de enrutamiento en una red mallada se utilizará una metodología basada en *testbeds* o camas de pruebas, es decir se dispondrá de un lugar equipado con elementos de software y hardware donde se realizarán una serie de pruebas sobre los protocolos que serán estudiados.

3.1. Fases para el desarrollo

El diseño del *testbed* se dividirá en pasos, donde cada uno será primordial para que se lleve a cabo un correcto desarrollo de todas las etapas en conjunto. Dichos pasos son: escogencia y descripción de los escenarios, implementación de los escenarios, selección y ejecución de pruebas, y análisis de los resultados obtenidos. Estos pasos serán definidos a continuación.

3.1.1. Escogencia y descripción de los escenarios

Un escenario en este trabajo se define como el ambiente de red sobre el cual se realizarán las pruebas. En cada escenario se varían elementos necesarios para comprobar el rendimiento de la red. Para la definición de cada uno de los escenarios que van a ser objeto de evaluación, se deberá tener claro que los mismos serán seleccionados de acuerdo a criterios de disponibilidad, por ejemplo la cantidad de equipos disponibles, para poder pasar a diseñar cada escenario que será objeto de evaluación.

3.1.2. Implementación de los escenarios

Para la implementación de los escenarios lo primero que se deberá realizar es la configuración de los dispositivos así como también la ejecución del software requerido, acorde a los escenarios definidos en la fase anterior.

3.1.3. Selección y ejecución de pruebas

En esta fase se escogerán el conjunto de pruebas que serán necesarias para la evaluación de cada uno de los escenarios implementados. Al momento de seleccionar cada una de las pruebas a realizar se deben tomar en cuenta ciertos criterios que permitan elegir pruebas donde los resultados obtenidos sean significativos. En esta fase también se ejecutarán las pruebas seleccionadas por lo que se deberá configurar y ejecutar el software necesario para la realización de las mismas.

3.1.4. Análisis de resultados

Una vez recopilados todos los resultados, estos serán interpretados dentro de esta fase. El proceso de análisis de resultado se llevará a cabo haciendo comparaciones entre todos los protocolos probados. Luego de ser realizada esta fase de análisis se obtendrá información que será crucial a la hora de llegar a una conclusión.

3.2. Herramientas utilizadas en el desarrollo

Para la elaboración del Trabajo Especial de Grado hicieron falta tanto herramientas de hardware como de software que se especifican a continuación.

3.2.1. Hardware

Para este trabajo se utilizaron un máximo de 29 computadoras, incluyendo a los clientes de la red mallada, los servidores, el *gateway* y el propietario del enrutador inalámbrico. Todas estas computadoras eran estaciones fijas que contaron con tarjetas de red inalámbrica, exceptuando 2 computadoras que eran laptops. Otro componente de hardware utilizado fue una tarjeta de red inalámbrica USB, para ser utilizada como una segunda interfaz inalámbrica en el computador que sirvió de *gateway*. Adicionalmente también se utilizó un enrutador inalámbrico para interconectar la red mallada y la red del propietario a la Internet.

La descripción detallada de las computadoras utilizadas se presenta en la Tabla 1.

CANT	MARCA	MODELO	CARACTERISTICAS
27	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	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

Tabla 1. - Descripción de las computadoras utilizadas.

En la Tabla 2 se presenta la descripción detallada del enrutador inalámbrico.

CANT	MARCA	MODELO	CARACTERISTICAS
1	Linksys	WRT54GL	Conmutador integrado de 4 puertos, Velocidad de transferencia de datos de 54 Mbps, Banda de frecuencia 2.4 Ghz, Estándares Ethernet, Fast Ethernet, IEEE 802.11b, IEEE 802.11g , 2 Antenas omnidireccionales, 13 Canales seleccionables

Tabla 2. - Descripción del enrutador utilizado.

En la Tabla 3 se presenta la descripción detallada de la tarjeta inalámbrica USB.

CANT	MARCA	MODELO	CARACTERISTICAS
1	D-Link	WUA-1340	Estándares IEEE 802.11g y IEEE 802.11b, Interfaz USB 2.0, Rangos de Frecuencia de 2.4GHz a 2.462GHz, Antena Omnidireccional.

Tabla 3. - Descripción de la tarjeta inalámbrica utilizada.

3.2.2. Software

Para la elaboración de este trabajo todas las computadoras utilizadas contaron con el Sistema Operativo Linux, distribución Ubuntu 10.04 – Lucid Lynx.

Adicionalmente en todos los equipos corría un script que ejecutaba una línea de sincronización con un servidor NTP cada 5 minutos, para asegurar que todas las computadoras tuvieran el mismo tiempo de reloj y no hubiese problemas con las mediciones, ya que según la documentación de la herramienta D-ITG, ésta no provee ningún tipo de sincronización entre el dispositivo que envía y el que recibe.

En la Figura 5 se presenta el script utilizado para la sincronización entre los clientes y el servidor.

```
1  #!/bin/bash
2  While /bin/true
3  Do
4  ntpdate ntp.ubuntu.com
5  Sleep 300
6  done
```

Figura 5. - Script para sincronización.

Las computadoras pertenecientes a la red mallada contaron con las implementaciones de tres protocolos distintos. Las implementaciones fueron:

- Protocolo Babel: se usó la implementación de babeld desarrollada por Laboratoire PPS, Université de Paris 7 [10]. Se usó la última versión disponible (versión 1.0.1) para ambientes Linux.
- Protocolo BATMAN: se usó la última versión estable (versión 0.3.2) disponible de la implementación batmand, del protocolo de enrutamiento BATMAN [11]. Esta implementación está desarrollada para ambientes Linux.
- Protocolo OLSR: se usó la versión 0.5.6-r7 de la implementación olsrd, del protocolo OLSR para ambientes Linux. También se utilizó el plugin olsrd-plugins-0.5.6-r7-1 [12]

Para la inyección de paquetes y medición de rendimiento de la red, se utilizó la aplicación D-ITG (Distributed Internet Traffic Generator) [13]. D-ITG es de código abierto y se utiliza para la generación de tráfico. Es capaz de calcular métricas como el *jitter*, el retardo y el *throughput*. Sigue el modelo Cliente -Servidor y posee un manual donde se explica todas las opciones y los parámetros que se pueden especificar al usar esta aplicación. Hay cuatro herramientas que conforman la aplicación: ITGSend, ITGRecv, ITGLog, e ITGDec. ITGSend actúa como el cliente, es decir como generador de flujo de datos. ITGRecv actúa como el servidor y puede recibir varios flujos de datos de diferentes clientes de forma simultánea. ITGLog funciona como un servidor de almacenamiento de logs y recibe información de ITGSend e ITGRecv. Por último ITGDec que es la herramienta que nos permite visualizar los resultados de los experimentos realizados.

En cuanto al enrutador inalámbrico, éste contaba con el firmware DD-WRT v21 y fue actualizado a la versión v24 [14].

Para el manejo de ancho de banda, se utilizó el script *wondershaper* [15] que nos permite limitar eficientemente el ancho de banda de una conexión.

3.3. Métricas utilizadas en el desarrollo

Para realizar las mediciones de desempeño de la red se utilizaron las siguientes métricas: *jitter*, *throughput*, retardo (*delay*) y tasa de paquetes perdidos, las cuales se describen a continuación.

3.3.1. Jitter

El *jitter* también es conocido como dispersión del retardo y es la variabilidad del tiempo que se tarda en llegar un paquete del nodo origen al nodo destino [16]. Este efecto es particularmente indeseado en las aplicaciones multimedia en Internet, como telefonía IP o video conferencias ya que hace que algunos paquetes lleguen demasiado pronto o tarde para poder entregarlos a tiempo. Midiendo el jitter se busca saber que tipo de tráfico se podría manejar en la red. Estas mediciones se realizaron con el generador de tráfico D-ITG.

3.3.2. Throughput

El *throughput* se refiere a la cantidad de data que puede ser enviada desde un origen al destino en un tiempo determinado [16]. Generalmente es medido en bits por segundos o paquetes por segundos, para este trabajo se utilizó la segunda opción. Utilizando el Packet Rate, que muestra la tasa de paquetes enviados por segundo, se evaluaron los valores de esta métrica de desempeño a través de la herramienta de generación de tráfico D-ITG. Con esto se busca medir el desempeño de la red y tener conocimiento de la capacidad efectiva del canal según las variantes de los escenarios.

3.3.3. Retardo (Delay)

El retardo de un paquete en una red es el tiempo que le toma llegar al destino desde el origen [16]. Éste depende de la velocidad de transmisión del medio y de los retardos que puede haber en dispositivos intermedios como enrutadores, etc. Que el retardo tenga un valor bajo es indicio de una red más eficiente. Las mediciones del retardo se realizaron con el generador de tráfico D-ITG.

3.3.4. Paquetes Perdidos (Packet Loss)

Ocurre cuando uno o más paquetes que viajan a través de la red, no pueden llegar a su destino [16]. La pérdida de paquetes puede ocurrir por diferentes motivos, como la congestión del canal, la degradación de la señal, paquetes corruptos, etc. En este trabajo, esta métrica fue medida mediante la herramienta *ping*.

4. Implementación de los Escenarios y Ejecución de las Pruebas

En este capítulo se describe el diseño del *testbed*, comenzando por escoger los escenarios en los que se realizaron las pruebas de desempeño, además de los pasos a seguir para la implementación de dichos escenarios y la selección y ejecución de las pruebas necesarias para la obtención de resultados y posterior análisis.

En la Figura 6 se pueden apreciar los elementos involucrados en todos los escenarios y la manera en la que interactúan.

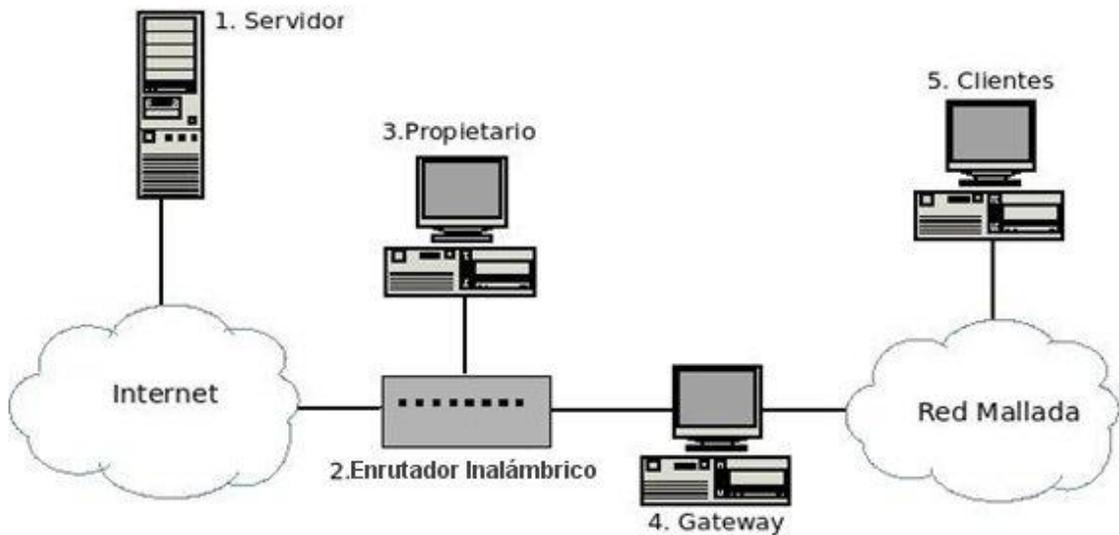


Figura 6. - Elementos de los escenarios.

1. Servidor: Está ubicado en cualquier parte de la Internet. Recibe peticiones de los clientes de la red mallada y del propietario.
2. Enrutador Inalámbrico: Interconecta tanto al *gateway* como al propietario a la Internet.
3. Propietario: Usuario de la red en la vivienda donde está el enrutador inalámbrico que permitirá la conexión de los clientes en la red mallada a la Internet.
4. Gateway: Cuenta con dos (2) interfaces de red inalámbrica, una para conectarse al enrutador inalámbrico y otra para conectarse a la red mallada.
5. Clientes: Dependiendo del escenario varía la cantidad de clientes que se interconectan entre sí formando la red mallada.

Los elementos 1, 3, 4, y 5 son computadoras con características similares descritas en la sección 3.2. (Tabla 1)

4.1. Escogencia y descripción de los escenarios

Para la escogencia de los escenarios de estudio fue necesario determinar que elementos pertenecientes al *testbed* podrían influenciar en el rendimiento de la red mallada y de acuerdo a esto, se plantearon los distintos escenarios que fueron objeto de evaluación.

- Número de dispositivos: El número de dispositivos que pertenezcan a la red mallada puede influir en el rendimiento que muestre la red. Se decidió escoger escenarios donde el número de dispositivos varíen entre 5, 15 y 25 clientes.
- Ancho de Banda: El ancho de banda que se le permita usar a la red mallada puede influir en el rendimiento de la red, por lo que se decidió escoger escenarios donde se varíe el porcentaje de ancho de banda que se le concede entre 25%, 50% y 100%. En éste último caso todo el ancho de banda es compartido entre los clientes mallados y el propietario.

A continuación se describen una serie de escenarios basados en la combinación de los elementos descritos anteriormente para medir el rendimiento de la red.

- **Escenario 1:** Se basa en una red mallada de 5 clientes. El ancho de banda que se le permite usar a la red es limitado al 25% de su totalidad. En este escenario se desea observar el desempeño de la red mallada cuando se limita el ancho de banda a menos de la mitad.
- **Escenario 2:** Se basa en una red mallada de 5 clientes. El ancho de banda que se le permite usar a la red mallada es limitado al 50% de su totalidad. Con este escenario se busca medir el desempeño de la red mallada cuando se limita el ancho de banda a la mitad.
- **Escenario 3:** Se basa en una red mallada de 5 clientes. El ancho de banda que se le permite usar a la red mallada es del 100%.
- **Escenario 4:** Se basa en una red mallada de 15 clientes. El ancho de banda que se le permite usar a la red es limitado al 25% de su totalidad. En este escenario se desea observar el desempeño de la red mallada cuando se limita el ancho de banda a menos de la mitad.
- **Escenario 5:** Se basa en una red mallada de 15 clientes. El ancho de banda que se le permite usar a la red mallada es limitado al 50% de su totalidad. Con este escenario se busca medir el desempeño de la red mallada cuando se limita el ancho de banda a la mitad.
- **Escenario 6:** Se basa en una red mallada de 15 clientes. El ancho de banda que se le permite usar a la red mallada es del 100%.
- **Escenario 7:** Se basa en una red mallada de 25 clientes. El ancho de banda que se le permite usar a la red es limitado al 25% de su totalidad. En este escenario se

desea observar el desempeño de la red mallada cuando se limita el ancho de banda a menos de la mitad.

- **Escenario 8:** Se basa en una red mallada de 25 clientes. El ancho de banda que se le permite usar a la red mallada es limitado al 50% de su totalidad. Con este escenario se busca medir el desempeño de la red mallada cuando se limita el ancho de banda a la mitad.
- **Escenario 9:** Se basa en una red mallada de 25 clientes .El ancho de banda que se le permite usar a la red mallada es del 100%.

4.2. Implementación de los escenarios

Esta fase consiste en la implementación de los escenarios descritos en la fase anterior, los cuales están compuestos por los mismos tipos de dispositivos inalámbricos, variando la cantidad de clientes pertenecientes a la red mallada y el ancho de banda disponible para la misma.

4.2.1. Configuración de la red

A continuación se describen todos los pasos para configurar la red mallada.

Clientes red mallada:

Para crear una red Ad-Hoc en Ubuntu 10.04 deben seguirse los siguientes pasos:

- Dirigirse al menú principal Sistema > Preferencias > Conexiones de Red.
- Elegir la pestaña “Inalámbrica” y hacer click en el botón “Añadir”.
- En la pestaña “Inalámbrica”, escribir el nombre del SSID y de la conexión. Elegir el modo Ad-Hoc.
- En la pestaña “de Ajustes de Ipv4” seleccionar método Manual y añadir la Dirección IP, Máscara de Red, Puerta de Enlace y Servidor DNS.
- Presionar el botón “Aplicar”.

La Figura 7 y Figura 8 se muestran los pasos 3 y 4 respectivamente.



Figura 7. - Creación de una red Ad-Hoc parte I.



Figura 8. - Creación de una red Ad-Hoc parte II.

Gateway:

Para configurar el equipo que sirvió como Gateway debieron seguirse los siguientes pasos:

- Realizar los pasos para crear una red Ad-Hoc mencionados anteriormente.
- Abrir un intérprete de comandos y ejecutar las siguientes líneas:

```
~$ sudo iptables -A POSTROUTING -t nat -j MASQUERADE  
~$ sudo -s  
~$ echo 1 > /proc/sys/net/ipv4/ip_forward
```

La primera línea es para habilitar NAT (Network Address Translation) y la última para activar el IP Forwarding. El NAT traduce las direcciones IP privadas en direcciones IP públicas.

IP Forwarding hace el reenvío de paquetes IP de una red a otra.

4.2.2. Control de ancho de banda

Para poder variar el ancho de banda del cuál dispondrá la red mallada, se ejecutó la siguiente línea en el gateway:

```
~$ sudo wondershaper valor wlan1
```

La variable valor es el límite de velocidad que tendrá la interfaz indicada, en este caso wlan1 es la interfaz por la que el equipo se conectó al enrutador inalámbrico.

4.3. Selección y ejecución de pruebas

Luego de implementar y verificar el buen funcionamiento de cada uno de los escenarios en la fase anterior, se procedió a la selección y ejecución de las pruebas para la evaluación de cada escenario. La instalación de los protocolos y el resto del software se detallan en el Anexo B. Las pruebas realizadas consisten en la utilización de diferentes implementaciones de protocolos de enrutamiento para redes malladas y la medición del rendimiento de la red con cada uno de ellos. También en esta fase se describirá la ejecución de las pruebas realizadas. El análisis de los resultados se presenta en el próximo capítulo.

4.3.1. Selección de pruebas

Los protocolos BATMAN, Babel y OLSR se probaron en base a los escenarios 1 al 9 definidos anteriormente. En este trabajo solo se estudió el tráfico HTTP, ya que esta investigación se enfocó en brindar acceso a Internet a los usuarios de la red mallada para

que realicen tareas simples como correo electrónico y navegación web, entre otros. Se usaron los tamaños de paquetes 64, 256, 1024 y 1518 bytes siguiendo las recomendaciones del RFC 2544 [17] para realizar pruebas de estos protocolos. Además en cada escenario se usó la herramienta de administración de redes *Ping* para obtener valores promedios de paquetes perdidos para el posterior análisis del desempeño de la red.

Para cada escenario se realizaron pruebas diferentes variando tanto el protocolo de enrutamiento como los tamaños de paquetes, generando solo un tamaño de paquete por prueba. El ancho de banda para todos los dispositivos conectados al enrutador inalámbrico fue limitado a 1024 Kbps de subida y 1024 Kbps de bajada, mediante una opción de QoS en el enrutador inalámbrico. Dichos valores equivalen el 100% del ancho de banda para nuestro trabajo. En cada uno de los escenarios se hicieron limitaciones de diferentes magnitudes para el uso del ancho de banda en la red mallada. Esto da un total de 135 pruebas (ver anexo A).

4.3.2. Ejecución de pruebas

Para ejecutar las pruebas primero se debió configurar y ejecutar los protocolos de enrutamiento a utilizar para luego comenzar la generación de tráfico y así poder realizar las mediciones de las diferentes métricas de desempeño.

Configuración y ejecución de los protocolos de enrutamiento

A continuación se describen los pasos necesarios para configurar y ejecutar los protocolos utilizados. Para todos los protocolos en los escenarios del 1 a 6, es decir los que contaban con 5 y 15 clientes, se utilizaron las opciones de configuración por defecto. Para los escenarios 7, 8 y 9 se cambiaron las métricas para lograr que la red soportara los 25 clientes bajo el protocolo OLSR y el protocolo BATMAN y poder hacer una comparación entre todos los protocolos en dichos escenarios. Con el protocolo Babel no hubo problemas con la configuración por defecto para soportar 25 clientes.

1. Babel:

Configuración por defecto:
HELLO Interval : 4 segundos

Para correr este protocolo, en todos los clientes de la red mallada y el gateway, se abrió un intérprete de comando y se ejecutó la siguiente línea:

```
~$ sudo babeld wlan0
```

Siendo wlan0 la interfaz por la que el equipo se conectó a la red mallada.

2. BATMAN:

Configuración por defecto:
HELLO Interval : 1 segundo

Configuración para 25 clientes:
HELLO Interval : 4 segundos

Para correr este protocolo, en todos los clientes de la red mallada y el gateway, se abrió un intérprete de comando y se ejecutó la siguiente línea:

```
~$ sudo batmand -d 1 wlan0
```

Siendo wlan0 la interfaz por la que el equipo se conectó a la red mallada. Y la opción -d 1 indica el nivel de depuración.

3. OLSR:

Configuración por defecto:
HELLO Interval : 0.6 segundos
TC Interval: 0.5 segundos

Configuración para 25 clientes:
HELLO Interval : 6 segundos
TC Interval: 3 segundos

Primero debió abrirse el archivo de configuración olsrd.conf ubicado en el directorio /etc y modificarse la interfaz de red, en la línea 32 del archivo, que en nuestro caso utilizó la interfaz wlan0. También debieron descomentarse las líneas del archivo que hacían referencia al *plugin* "olsrd_dyn_gw.so.0.4", en la línea 295.

Luego, para correr este protocolo, en todos los clientes de la red mallada y el gateway, se abrió un intérprete de comando y se ejecutó la siguiente línea:

```
~$ sudo olsrd -d 2
```

El parámetro -d 2 indica el nivel de depuración.

Generación de tráfico

Para la generación de tráfico se utilizó la herramienta D-ITG. Para iniciar el envío de paquetes TCP, se debe ejecutar el comando ITGSend desde el directorio /bin de la siguiente forma:

```
./ITGSend -a 0.0.0.0 -rp 8000 -sp 8100 -T TCP -C 999 -c 99 -t 9999
```

Donde:

- a indica la dirección ip del receptor.
- rp indica el puerto de recepción del equipo destino.
- sp indica el puerto de envío del nodo fuente.
- T indica el protocolo
- C indica el número de paquetes por segundo que se intentará enviar.
- c indica el tamaño del paquete
- t indica el tiempo que durará el envío de paquetes en milisegundos.

En nuestro caso, se ejecutó:

```
~$ ./ITGSend -a 150.185.74.108 -rp 8001 -sp 8100 -T TCP -C 2000 -c 64 -t 300000
```

En cada uno de los escenarios planteados se generaron paquetes TCP ya que las aplicaciones que generan tráfico web trabajan sobre TCP (HTTP/HTTPS, SMTP/POP3, FTP), la tasa de paquetes por segundo que se estableció fue de 2000, esto debido a que se realizaron pruebas preliminares donde se determinó que no se sobrepasaba la tasa de 1800 paquetes por segundo. En cada uno de los escenarios se generaron paquetes de 64, 256, 1024 y 1518 bytes, generando solo un tamaño de paquete por prueba durante 300 segundos. Todo esto siguiendo las recomendaciones del RFC 2544.

Para la recepción del flujo TCP, se debe ejecutar el comando `./ITGRecv` desde el directorio `/bin` de la siguiente forma:

```
~$ ./ITGRecv -l log
```

Donde:

- l indica el nombre del log donde se guardan los resultados.

Estos archivos son guardados en el directorio `/bin`.

Para poder visualizar el contenido de los archivos logs donde se guardaron los resultados se ejecuta el comando `ITGDec` en el directorio `/bin`, indicando el nombre del archivo que se quiere abrir.

```
~$ ./ITGDec log
```

Para la medición de *throughput*, *jitter* y retardo se realizaron 108 pruebas y cada una de ellas se repitió 10 veces. Para la medición de paquetes perdidos se utilizó la herramienta ping. Se realizaron 500 ping en cada escenario a la par que se generaba tráfico de congestión con paquetes que variaban su tamaño entre 64 y 1518 bytes, esto con la herramienta D-ITG.

5. Resultados y Análisis de Resultados

En este capítulo se describen los resultados obtenidos al realizar las pruebas de los protocolos de enrutamiento para los escenarios descritos anteriormente. En las secciones 5.1, 5.2 y 5.3 se muestran los parámetros de rendimiento comparados por el ancho de banda del cuál dispone la red mallada. En las secciones 5.4, 5.5 y 5.6 se muestran los resultados comparativos por número de clientes en la red mallada. Luego se presenta un análisis de los resultados.

5.1. Resultados – Escenarios 1, 2 y 3.

A continuación se presentan los resultados de las pruebas realizadas sobre los escenarios 1, 2 y 3, es decir, en una red mallada con 5 clientes, con los protocolos de enrutamiento BATMAN, Babel y OLSR y paquetes de 64, 256, 1024 y 1518 bytes, agrupados por la métrica de desempeño analizada.

5.1.1. Throughput

Desde la Figura 9 a la Figura 12 se muestra el *throughput* en función de la cantidad de ancho de banda disponible en el borde de la red para la conexión a Internet para tamaño de paquetes de 64, 256, 1024 y 1518 bytes en una red mallada con 5 clientes.

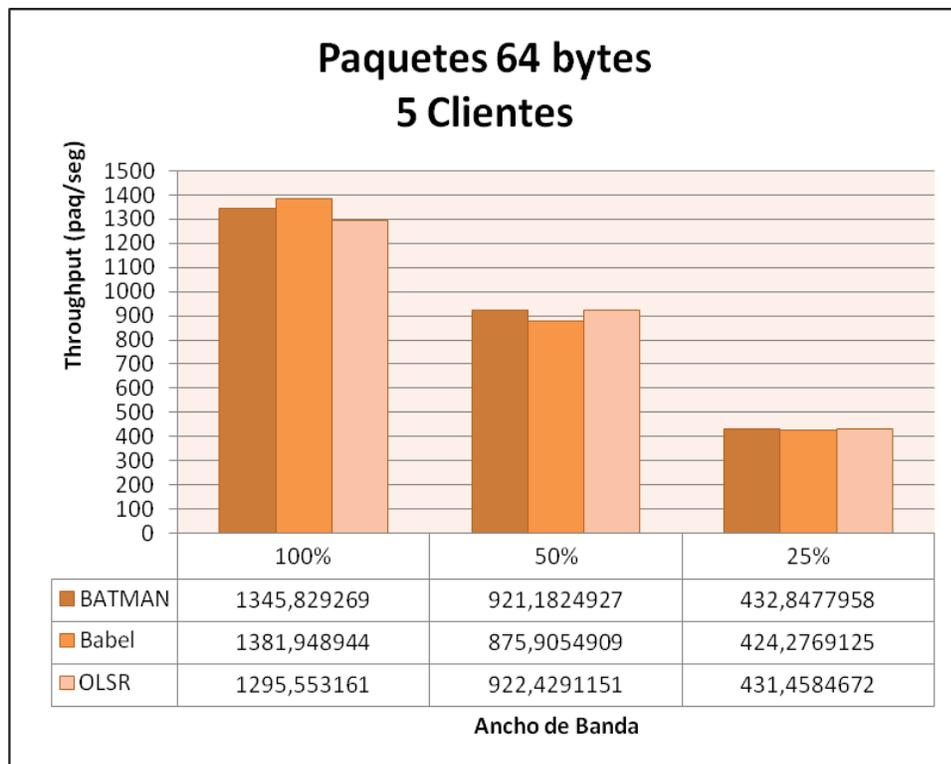


Figura 9. - Gráfica de *throughput* con paquetes de 64 bytes y 5 clientes.

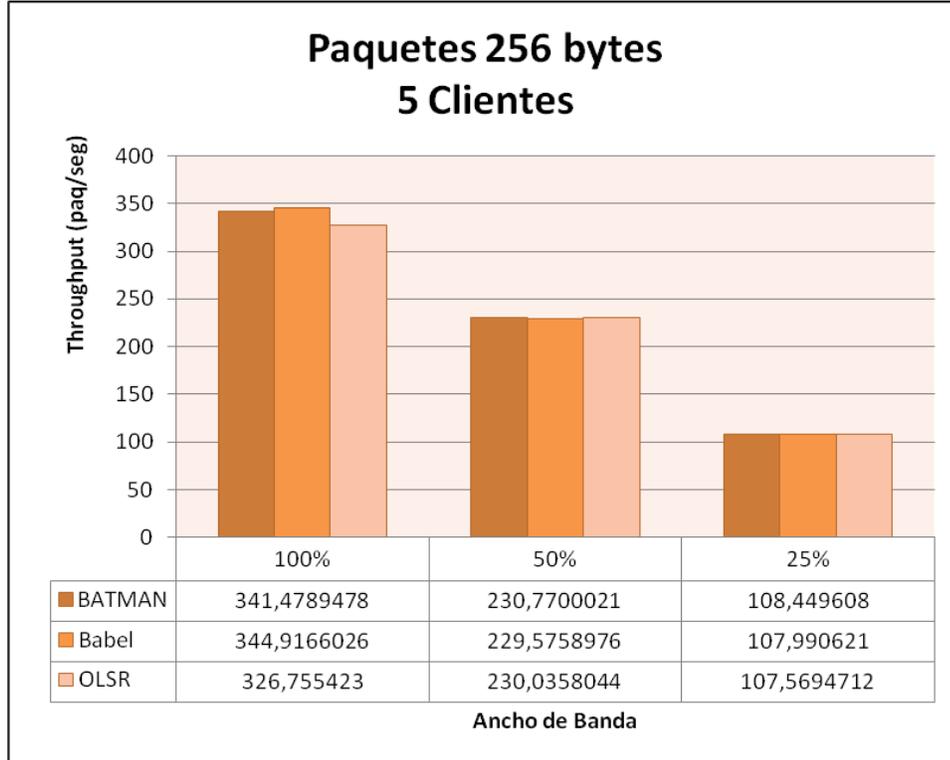


Figura 10. – Gráfica de *throughput* con paquetes de 256 bytes y 5 clientes.

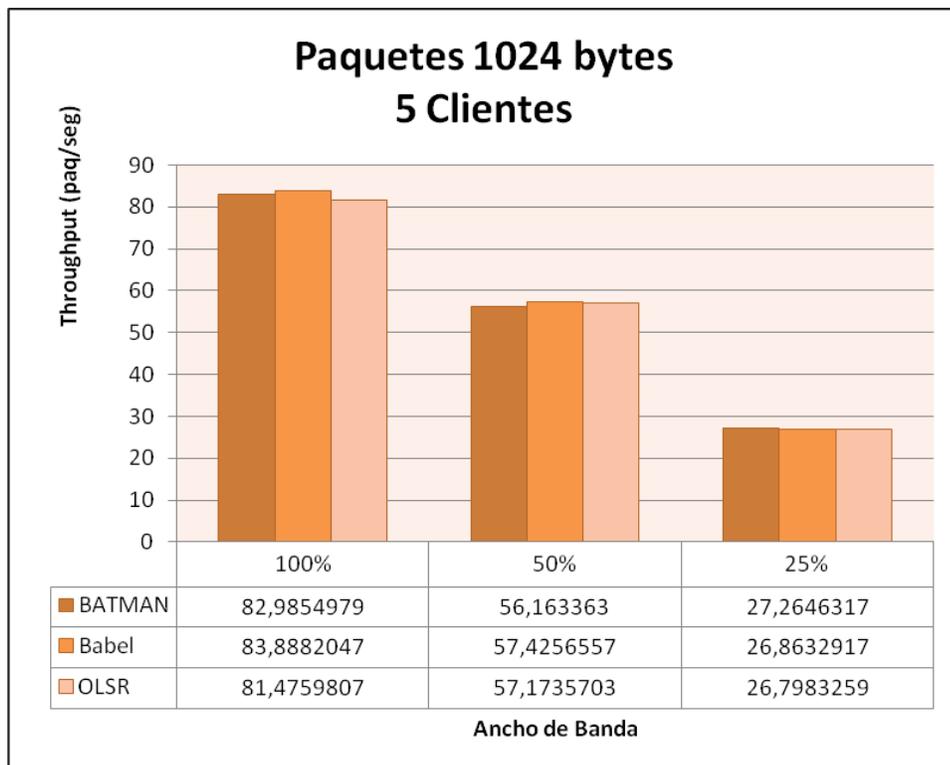


Figura 11. – Gráfica de *throughput* con paquetes de 1024 bytes y 5 clientes.

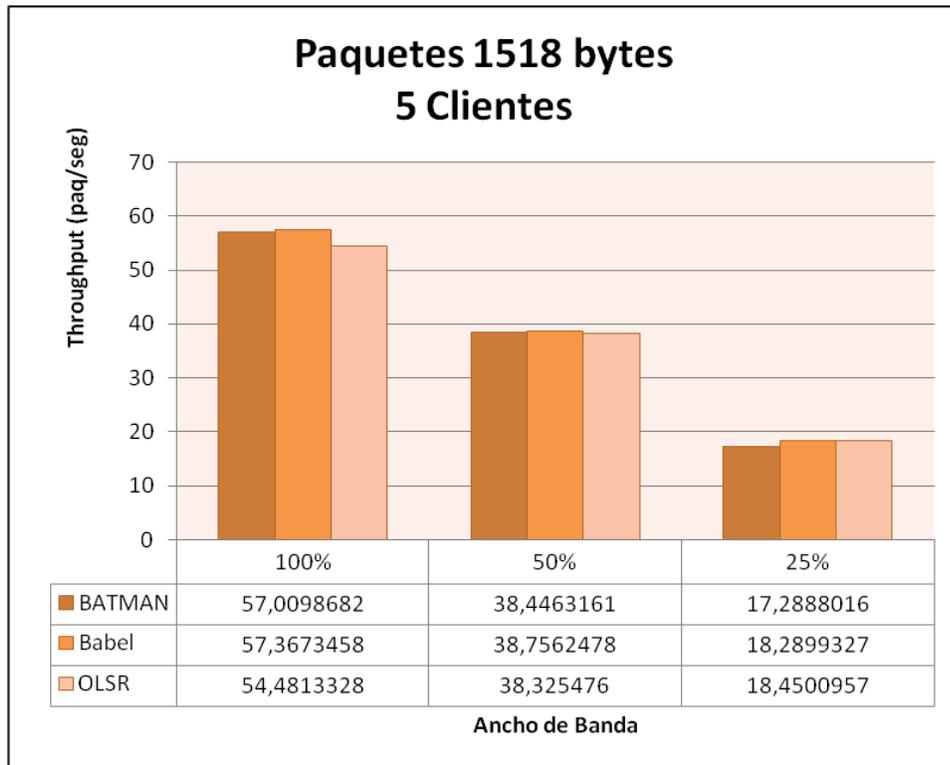


Figura 12. – Gráfica de *throughput* con paquetes de 1518 bytes y 5 clientes.

De los resultados mostrados anteriormente en las figuras, se puede observar que el *throughput* decrece a medida que el ancho de banda es restringido en el borde de la red, como se esperaba. También el *throughput* decrece a medida que aumenta el tamaño de los paquetes.

El rendimiento de la red en términos del *throughput* es similar independientemente del protocolo de enrutamiento utilizado y del ancho de banda disponible, observándose una ligera mejoría en el rendimiento de la red cuando se usa el protocolo Babel en una red con 100% de ancho de banda disponible en el borde.

5.1.2. Jitter

Desde la Figura 13 a la Figura 16 se muestra el *jitter* en función de la cantidad de ancho de banda disponible en el borde de la red para la conexión a Internet para tamaño de paquetes de 64, 256, 1024 y 1518 bytes en una red mallada con 5 clientes.

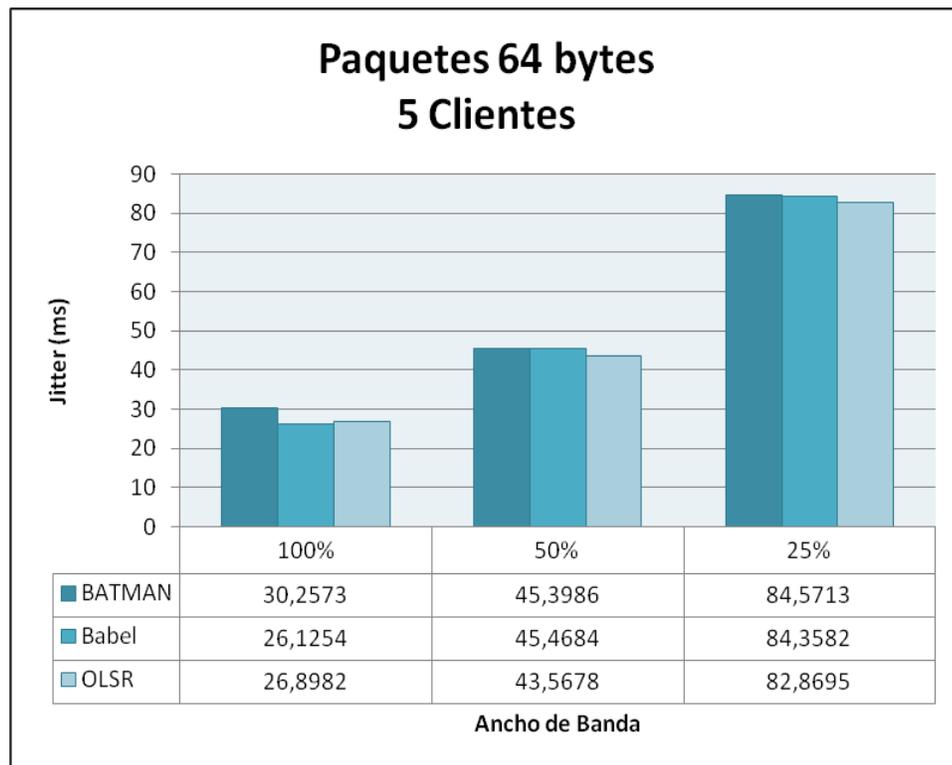


Figura 13. – Gráfica de *jitter* con paquetes de 64 bytes y 5 clientes.

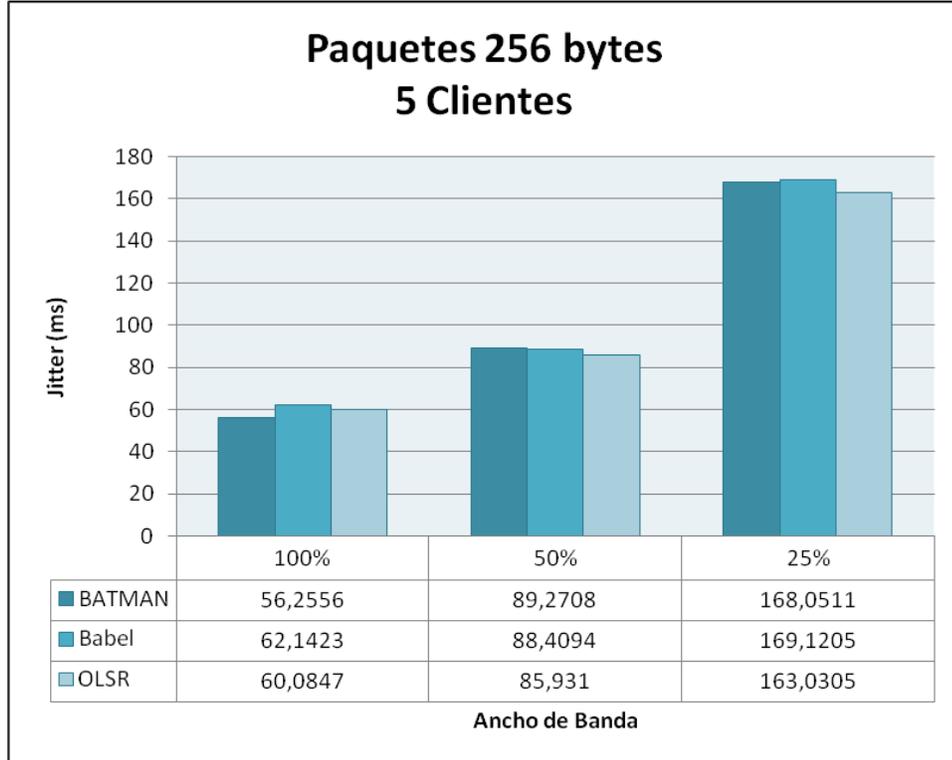


Figura 14. – Gráfica de *jitter* con paquetes de 256 bytes y 5 clientes.

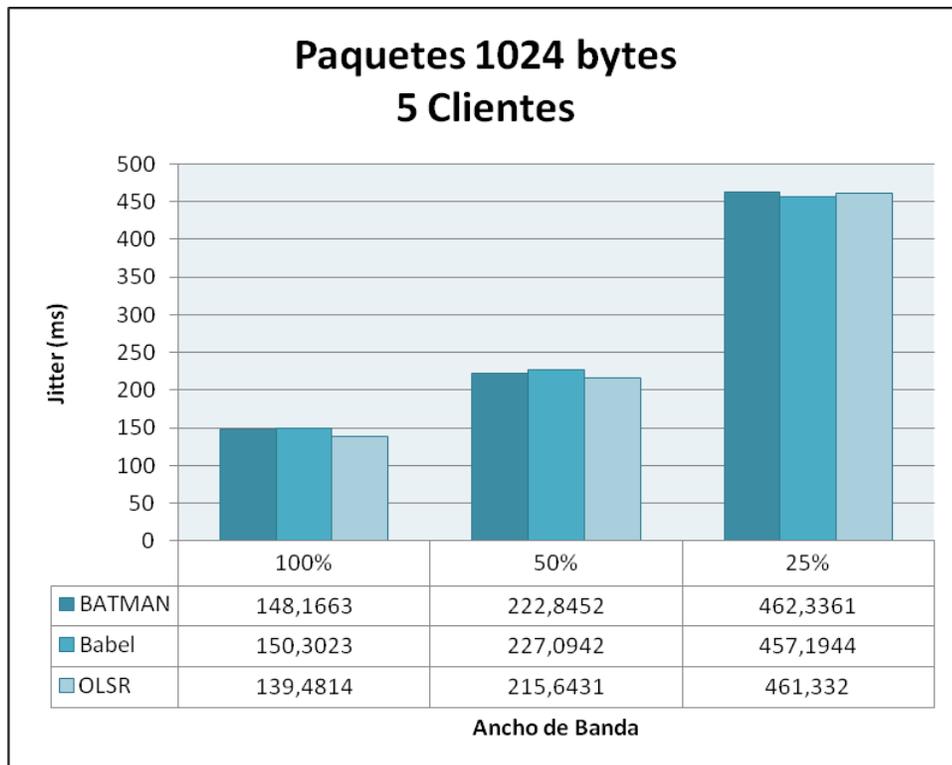


Figura 15. – Gráfica de *jitter* con paquetes de 1024 bytes y 5 clientes.

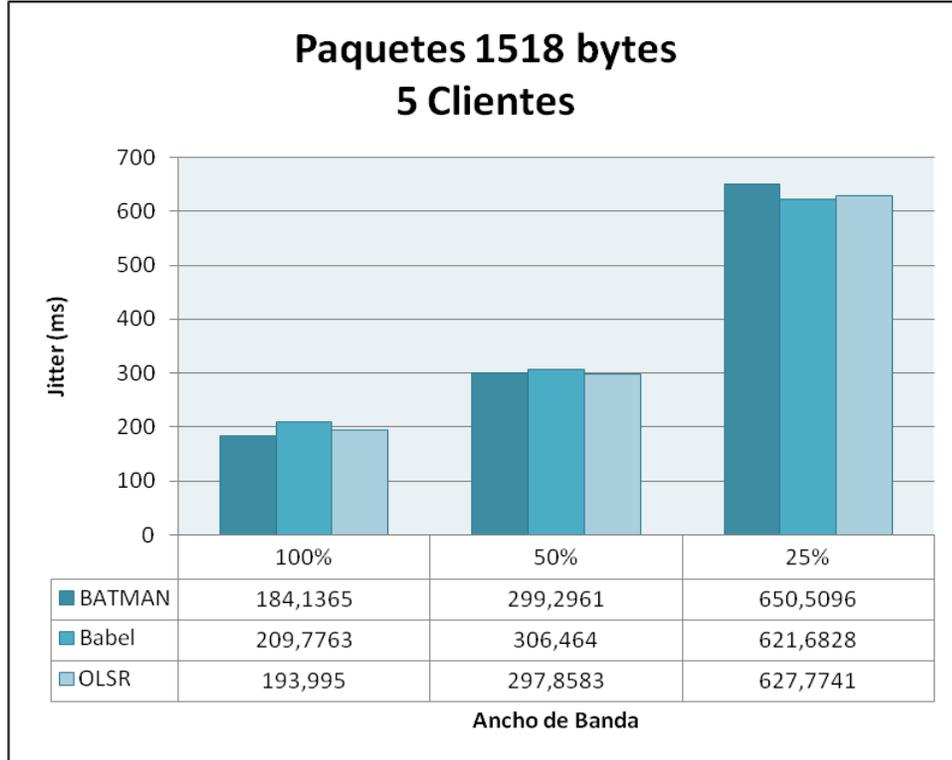


Figura 16. – Gráfica de *jitter* con paquetes de 1518 bytes y 5 clientes.

De los resultados mostrados anteriormente en las figuras, se puede observar que los valores obtenidos de *jitter* aumentan a medida que el ancho de banda es restringido en el borde de la red. También aumentan a medida que el tamaño de los paquetes es mayor.

Adicionalmente se puede observar que el rendimiento de la red en términos de *jitter* es similar independientemente del protocolo de enrutamiento utilizado, sin embargo se obtienen valores ligeramente mejores cuando se usa OLSR en la mayoría de los escenarios estudiados.

5.1.3. Retardo

Desde la Figura 17 a la Figura 20 se muestra el retardo en función de la cantidad de ancho de banda disponible en el borde de la red para la conexión a Internet para tamaño de paquetes de 64, 256, 1024 y 1518 bytes en una red mallada con 5 clientes.

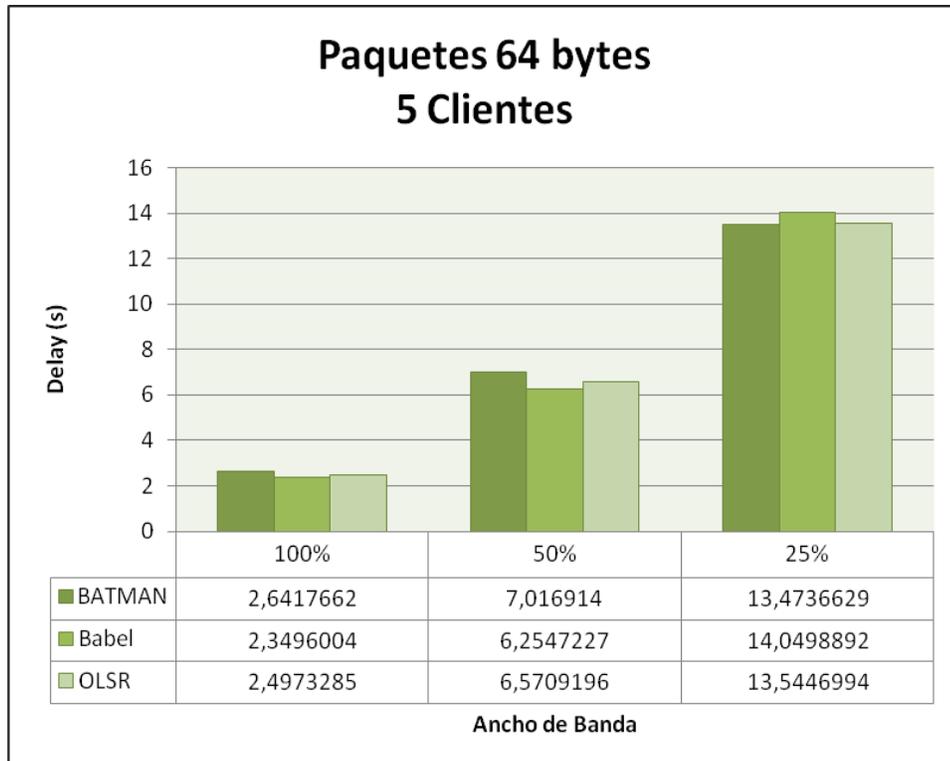


Figura 17. – Gráfica de retardo con paquetes de 64 bytes y 5 clientes.

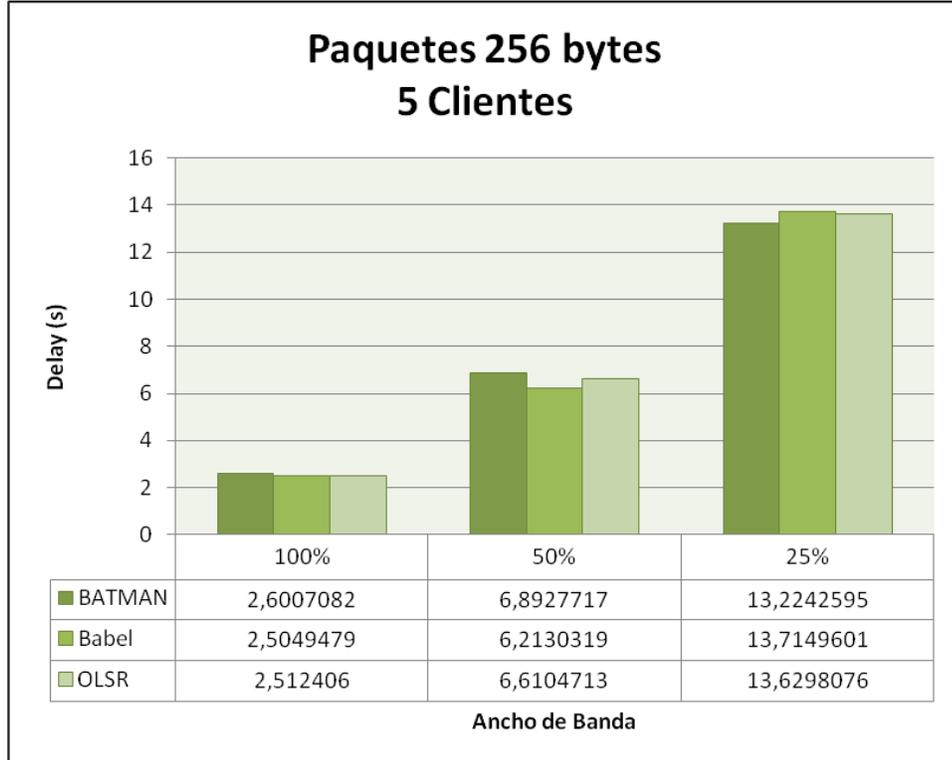


Figura 18. – Gráfica de retardo con paquetes de 256 bytes y 5 clientes.

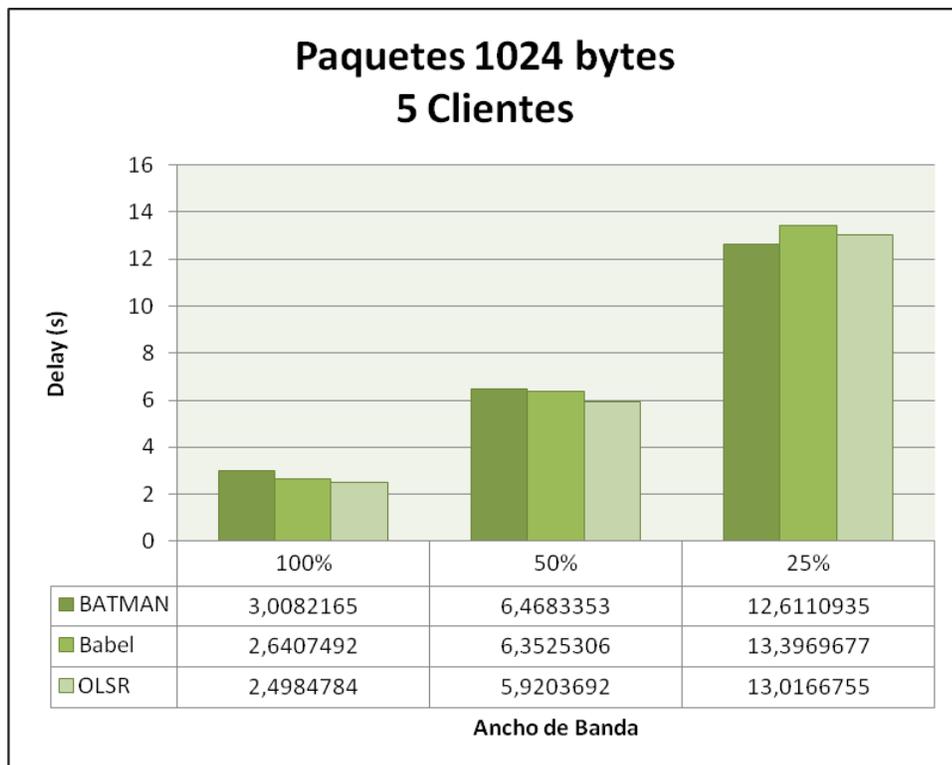


Figura 19. – Gráfica de retardo con paquetes de 1024 bytes y 5 clientes.

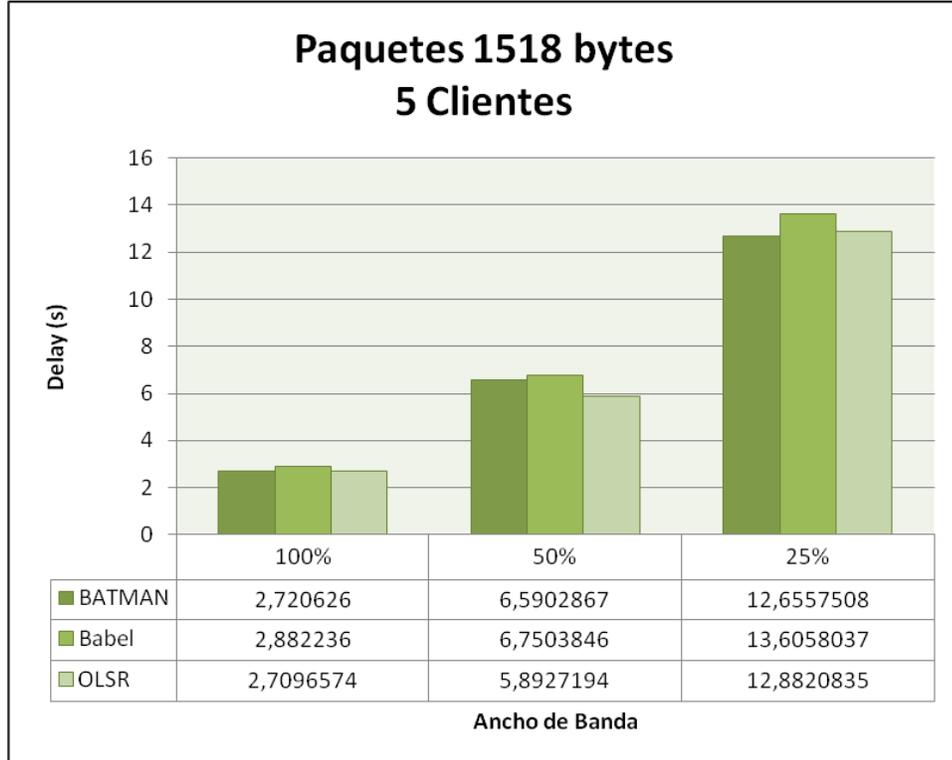


Figura 20. – Gráfica de retardo con paquetes de 1518 bytes y 5 clientes.

De los resultados mostrados anteriormente en las figuras, se puede observar que el retardo aumenta a medida que el ancho de banda es restringido en el borde de la red. Adicionalmente, se puede notar que cuando se restringe el ancho de banda al 50% o menos, la red tiene un mejor rendimiento en término del retardo cuando se usan los protocolos Babel u OLSR, mientras que cuando el ancho de banda es restringido un 75%, los resultados favorecen al protocolo BATMAN.

5.1.4. Paquetes Perdidos

En la Figura 21 se muestra el promedio de pérdida de paquetes en función de la cantidad de ancho de banda disponible en el borde de la red para la conexión a Internet en una red mallada con 5 clientes.

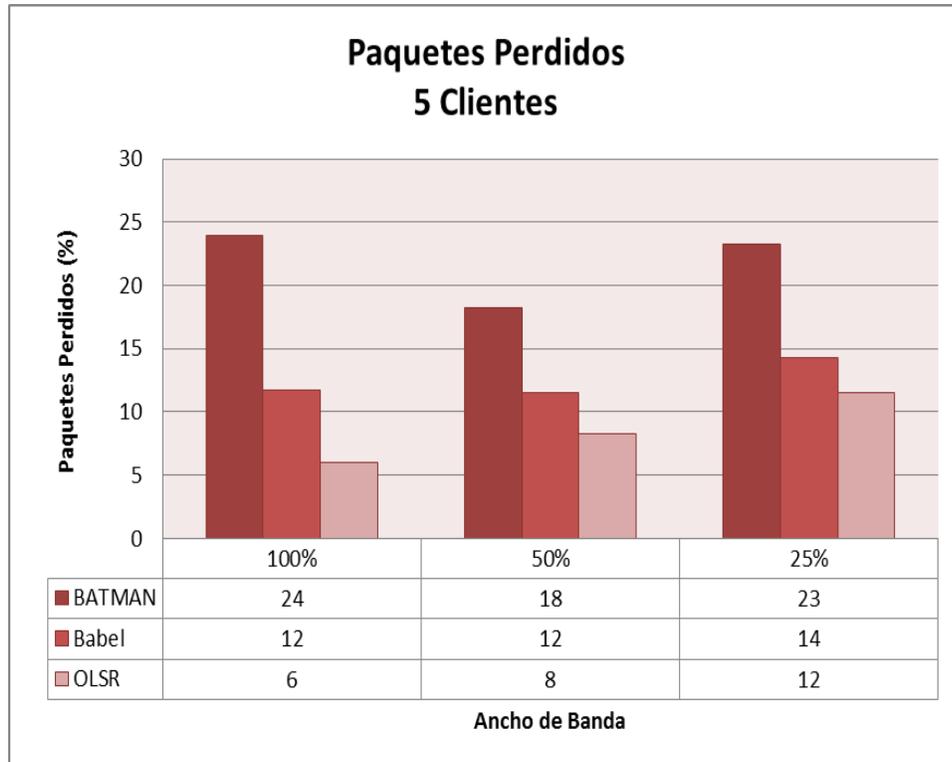


Figura 21. – Gráfica de paquetes perdidos para 5 clientes.

La figura anterior muestra que para cada protocolo de enrutamiento utilizado, el porcentaje de paquetes perdidos se mantiene muy similar a medida que el ancho de banda es restringido en el borde. Adicionalmente, se puede notar que el protocolo con el que se obtiene una menor tasa de paquetes perdidos independientemente del ancho de banda del cuál dispone la red mallada, es OLSR, siendo en promedio un 50% menor que Babel y 75% menor que BATMAN.

5.2. Resultados - Escenarios 4, 5 y 6

A continuación se presentan los resultados de las pruebas realizadas sobre los escenarios 4, 5 y 6, es decir, en una red mallada con 15 clientes, con los protocolos de enrutamiento BATMAN, Babel y OLSR y paquetes de 64, 256, 1024 y 1518 bytes, agrupados por la métrica de desempeño analizada.

5.2.1. Throughput

Desde la Figura 22 a la Figura 25 se muestra el *throughput* en función de la cantidad de ancho de banda disponible en el borde de la red para la conexión a Internet para tamaño de paquetes de 64, 256, 1024 y 1518 bytes en una red mallada con 15 clientes.

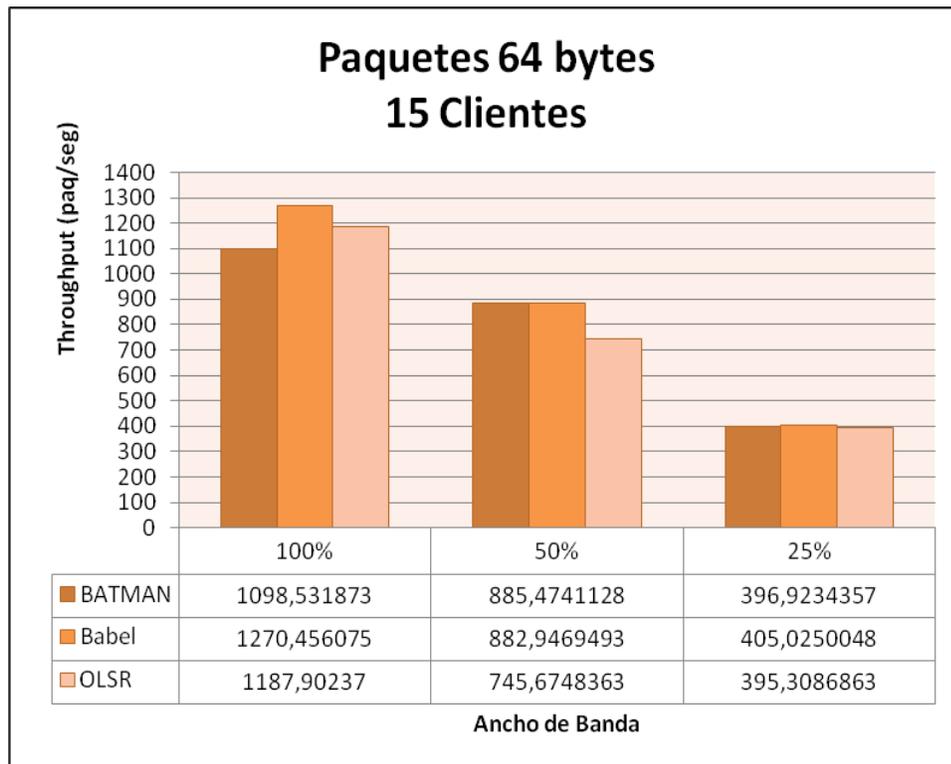


Figura 22. – Gráfica de *throughput* con paquetes de 64 bytes y 15 clientes.

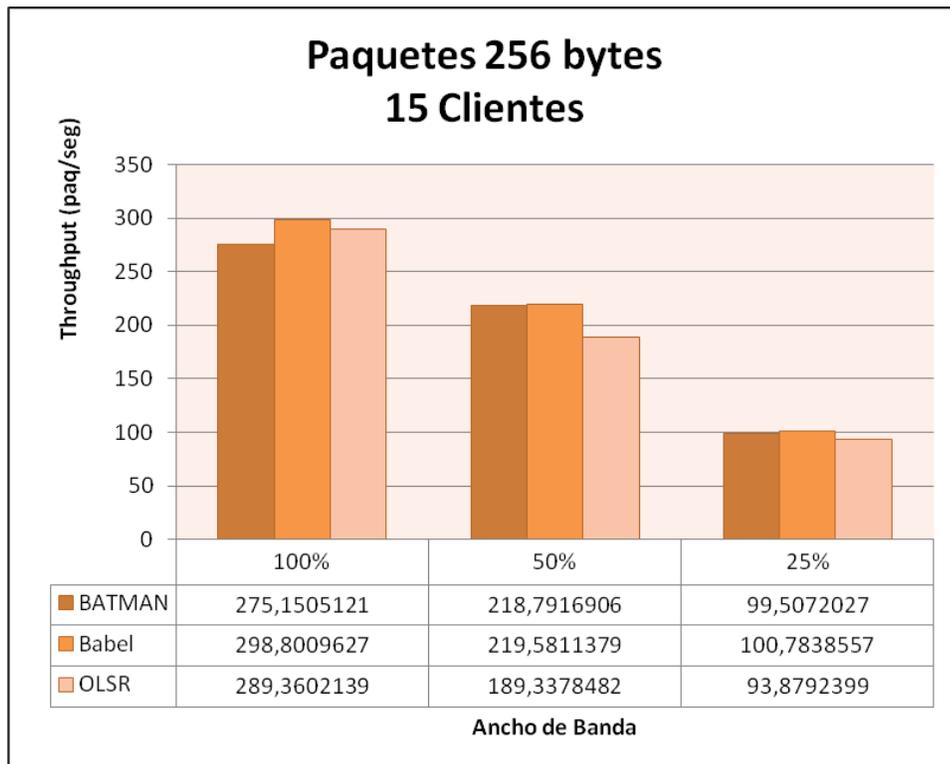


Figura 23. – Gráfica de *throughput* con paquetes de 256 bytes y 15 clientes.

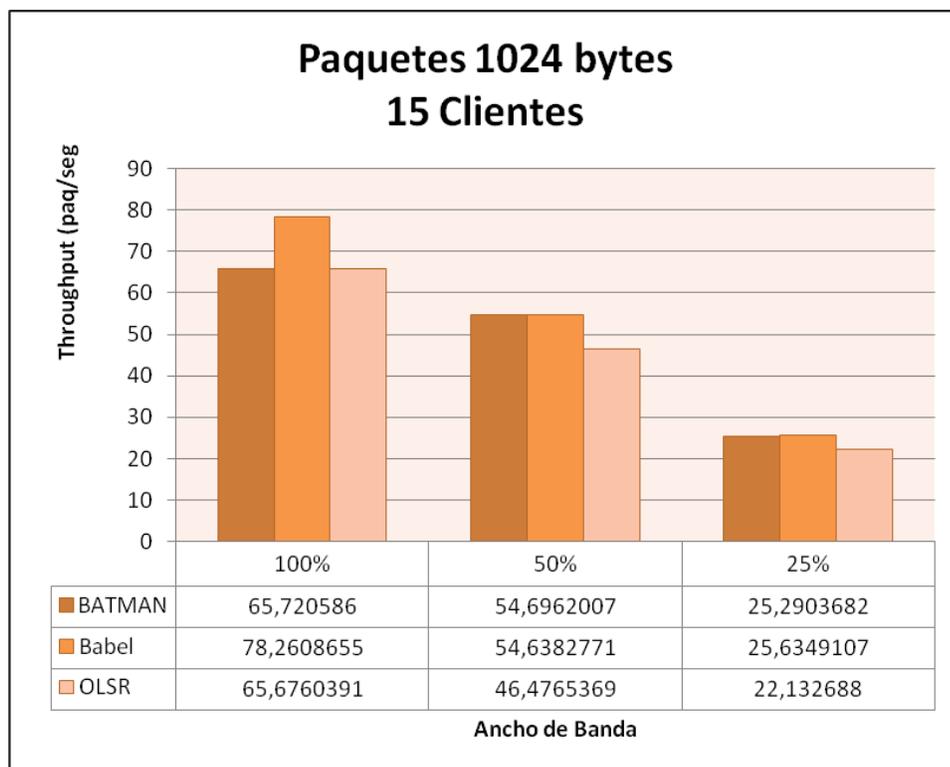


Figura 24. – Gráfica de *throughput* con paquetes de 1024 bytes y 5 clientes.

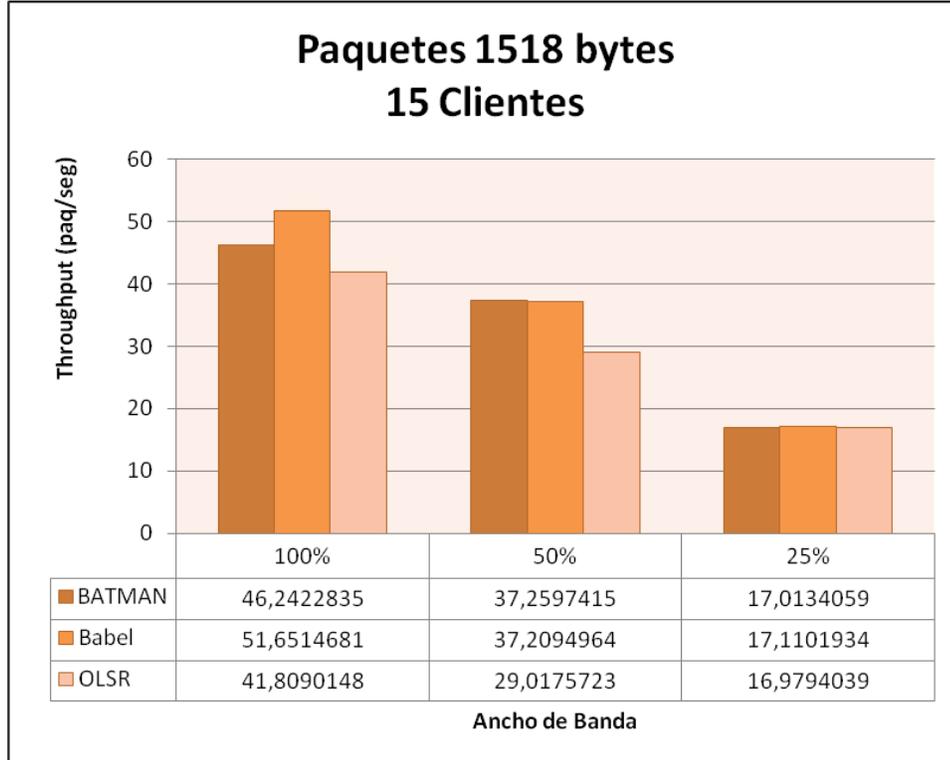


Figura 25. – Gráfica de *throughput* con paquetes de 1518 bytes y 15 clientes.

De los resultados mostrados anteriormente en las figuras, se puede observar que el *throughput* decrece a medida que el ancho de banda es restringido en el borde, como se esperaría. También el *throughput* decrece a medida que aumenta el tamaño de los paquetes.

Adicionalmente, se puede observar que cuando el ancho de banda disponible para los usuarios de la red mallada es del 100%, el uso del protocolo Babel supone un mejor rendimiento de la red en términos del *throughput* independientemente del tamaño de los paquetes que viajan por la red. En los otros casos, el uso tanto de Babel como BATMAN supone un mejor rendimiento de la red. Dicho rendimiento mejora notablemente cuando la restricción del ancho de banda es del 50%.

5.2.2. Jitter

Desde la Figura 26 a la Figura 29 se muestra el *jitter* en función de la cantidad de ancho de banda disponible en el borde de la red para la conexión a Internet para tamaño de paquetes de 64, 256, 1024 y 1518 bytes en una red mallada con 15 clientes.

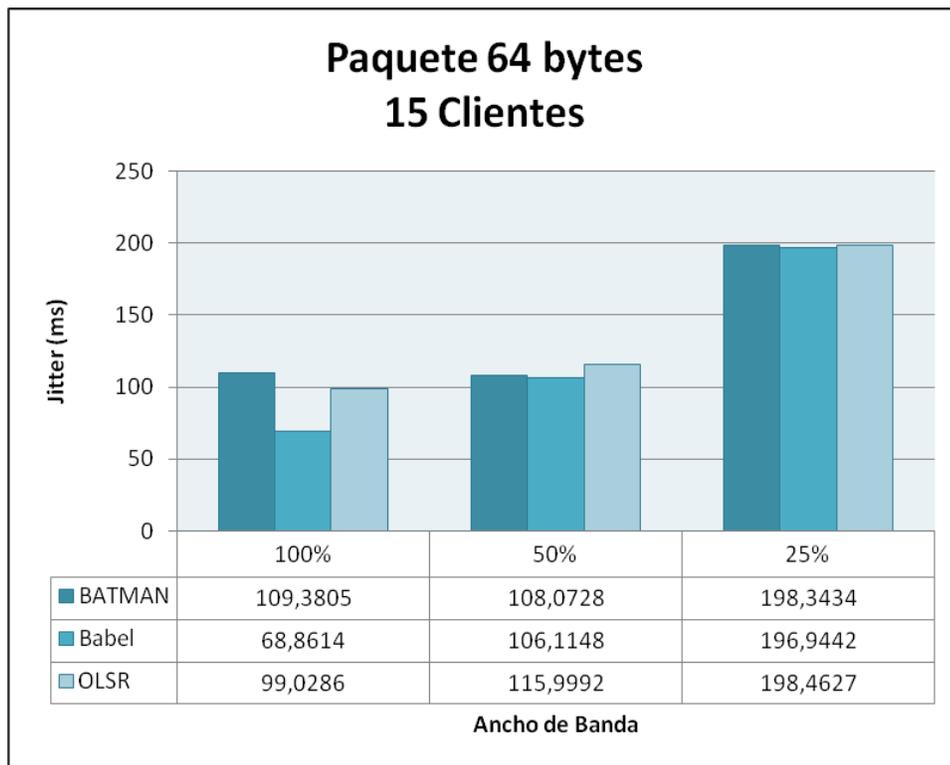


Figura 26. – Gráfica de *jitter* con paquetes de 64 bytes y 15 clientes.

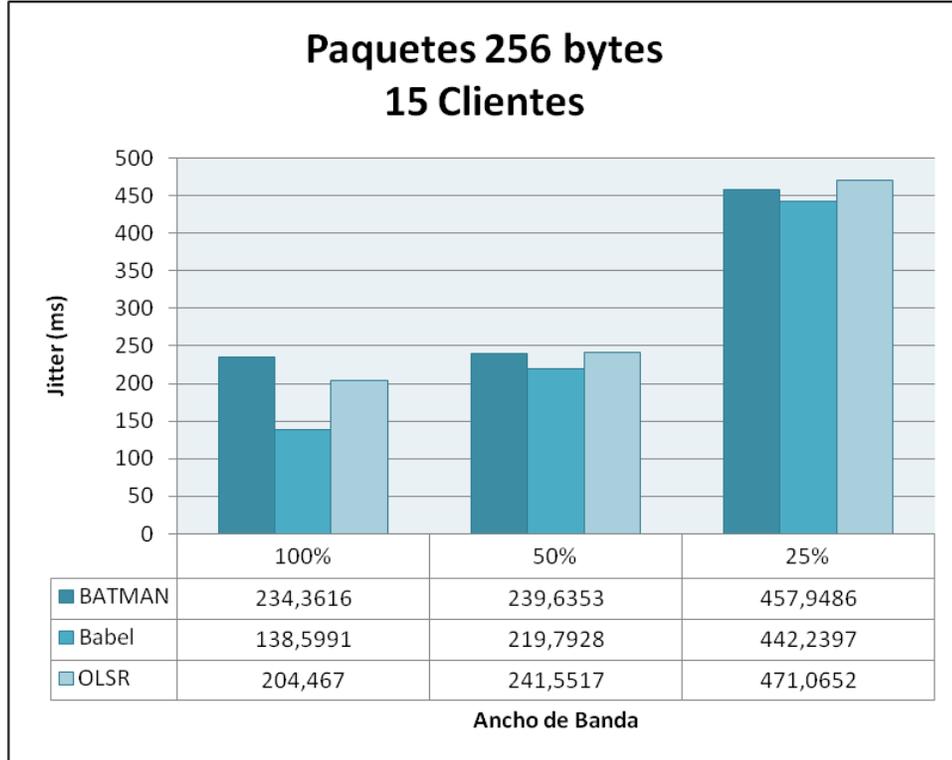


Figura 27. – Gráfica de *jitter* con paquetes de 256 bytes y 15 clientes.

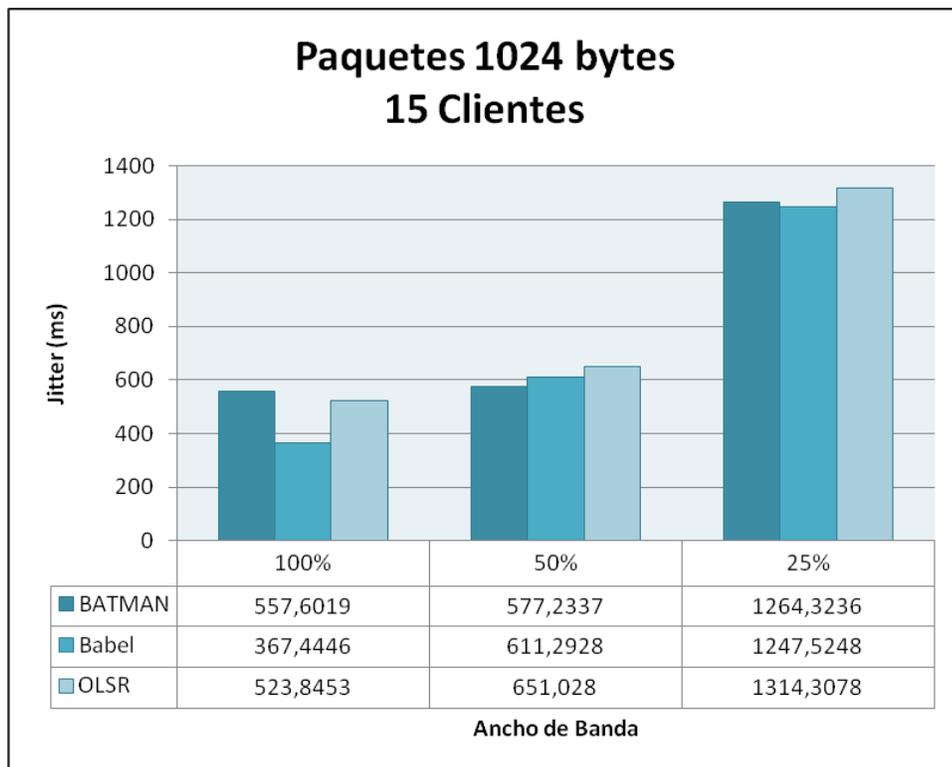


Figura 28. – Gráfica de *jitter* con paquetes de 1024 bytes y 15 clientes.

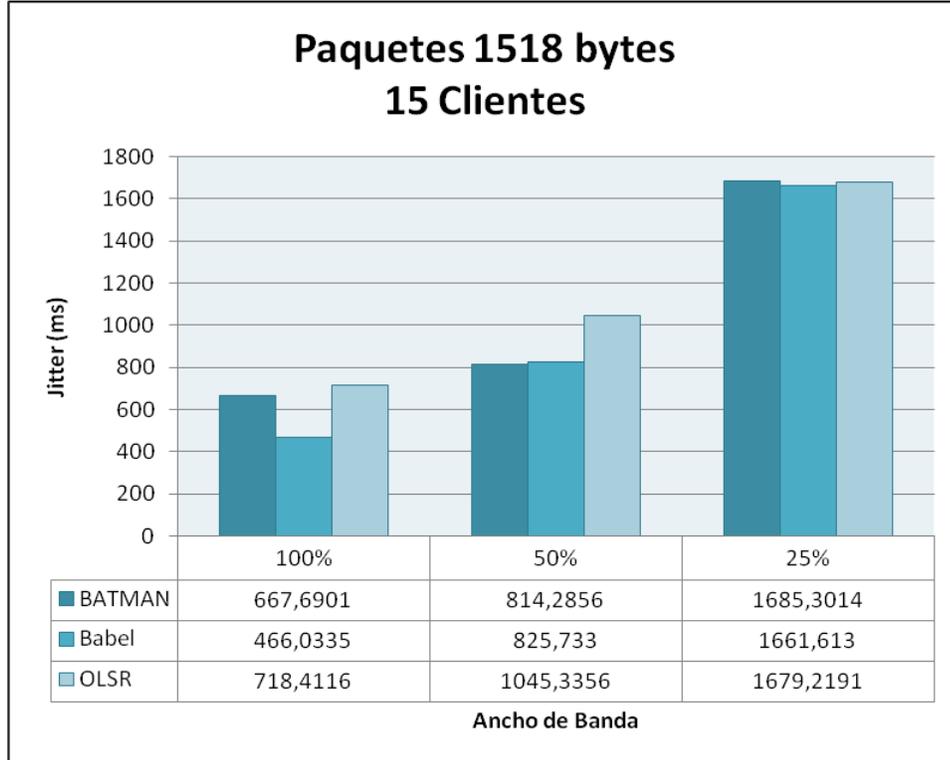


Figura 29. – Gráfica de *jitter* con paquetes de 1518 bytes y 15 clientes.

De los resultados mostrados anteriormente en las figuras, se puede observar que el *jitter* aumenta a medida que el ancho de banda es restringido en el borde. También aumenta a medida que aumenta el tamaño de los paquetes.

Cuando se aumenta la red mallada a 15 nodos y no se aplica restricción en el ancho de banda, el protocolo con el que se obtiene un menor *jitter* es con Babel. Al aplicar una restricción del 50% la variación del retardo obtenida con el protocolo Babel sube significativamente alcanzando al protocolo BATMAN, por lo que en este caso, con ambos protocolos se alcanzan valores similares de *jitter* y menores que con el protocolo OLSR. Al aplicar una restricción del 75% se obtienen valores muy similares con todos los protocolos con una leve mejoría en los valores obtenidos con el protocolo Babel.

5.2.3. Retardo

Desde la Figura 30 a la Figura 33 se muestra el retardo en función de la cantidad de ancho de banda disponible en el borde de la red para la conexión a Internet para tamaño de paquetes de 64, 256, 1024 y 1518 bytes en una red mallada con 15 clientes.

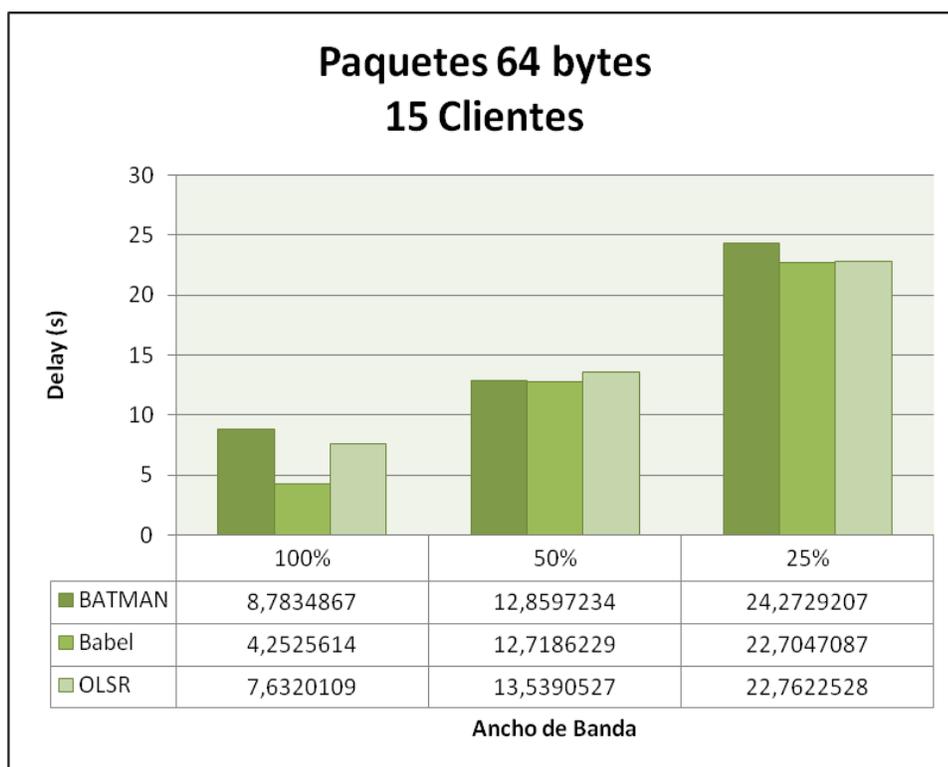


Figura 30. – Gráfica de retardo con paquetes de 64 bytes y 15 clientes.

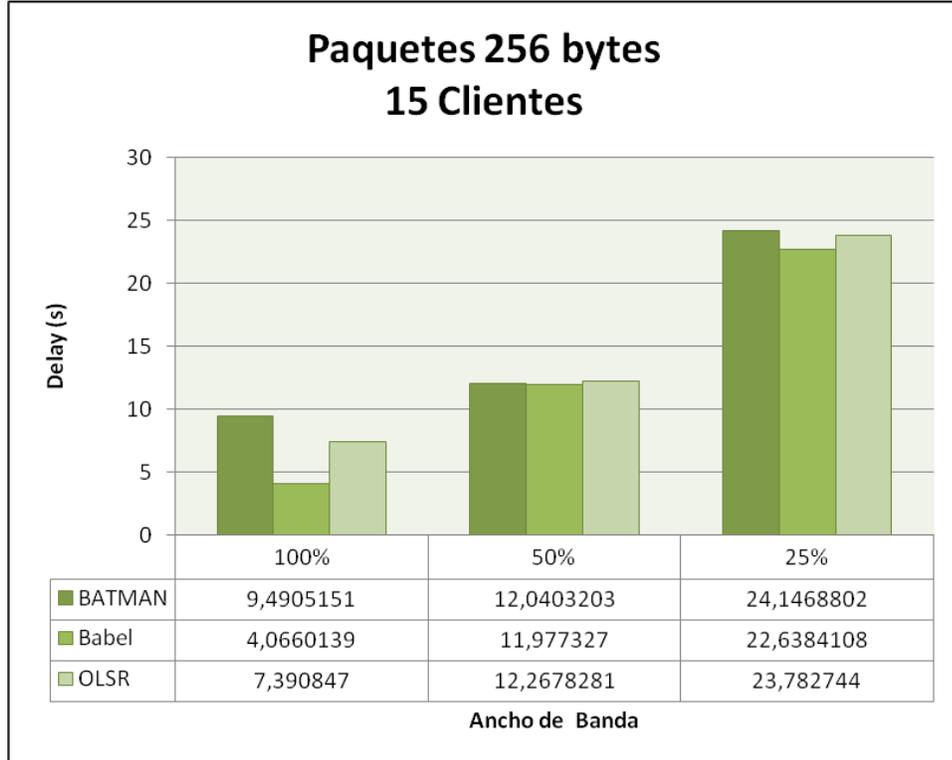


Figura 31. – Gráfica de retardo con paquetes de 256 bytes y 15 clientes.

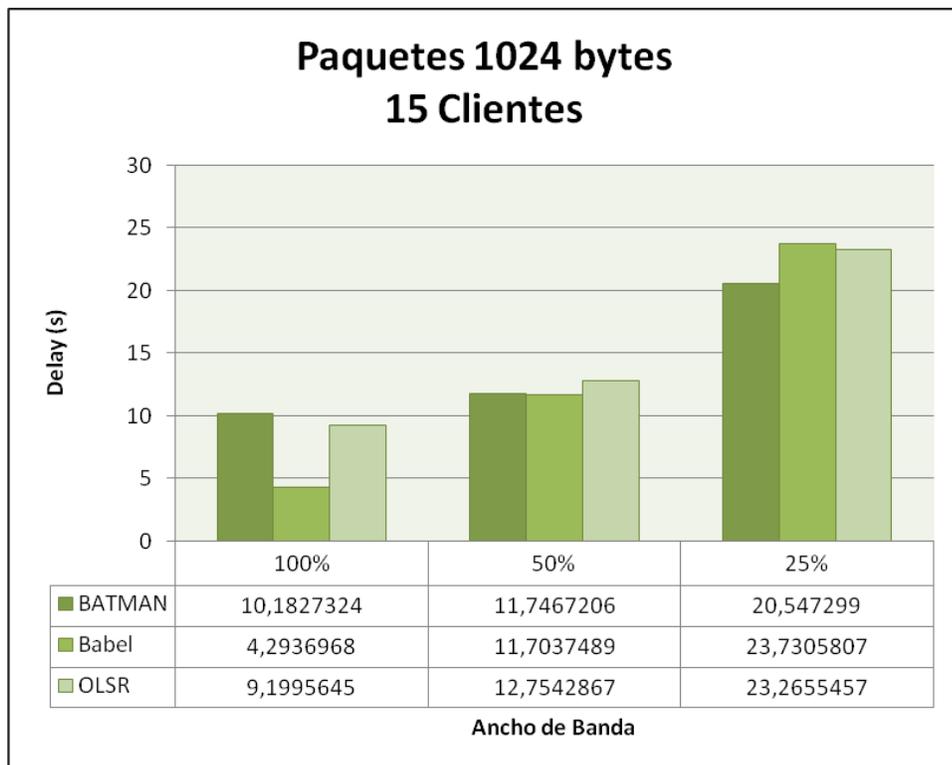


Figura 32. – Gráfica de retardo con paquetes de 1024 bytes y 15 clientes.

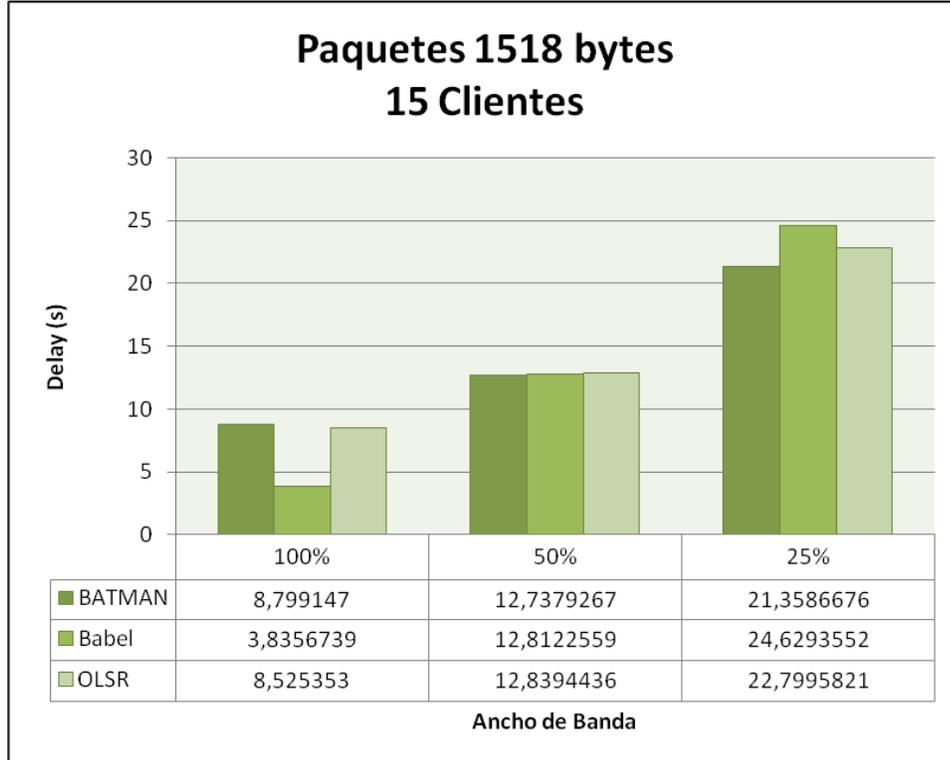


Figura 33. – Gráfica de retardo con paquetes de 1518 bytes y 15 clientes.

De los resultados mostrados anteriormente en las figuras, se puede observar que el retardo aumenta a medida que el ancho de banda es restringido en el borde.

Adicionalmente, se observa que al aumentar el número de nodos de la red mallada y no aplicar restricción en el ancho de banda, el protocolo con el que se obtiene un menor valor de retardo fue Babel con una diferencia significativa al resto de los protocolos. Al aplicar una restricción del 50% en el ancho de banda, el valor de retardo obtenido con Babel aumenta significativamente y en este caso, con todos los protocolos se obtienen valores similares de retardo. Al aplicar una restricción del 75% en el ancho de banda se obtiene un menor retardo con el protocolo BATMAN con paquetes de 1024 y 1518 bytes, mientras que con paquetes de menor tamaño se obtiene un menor retardo con el protocolo Babel.

5.2.4. Paquetes Perdidos

En la Figura 34 se muestra el promedio de pérdida de paquetes en función de la cantidad de ancho de banda disponible en el borde de la red para la conexión a Internet en una red mallada con 15 clientes.

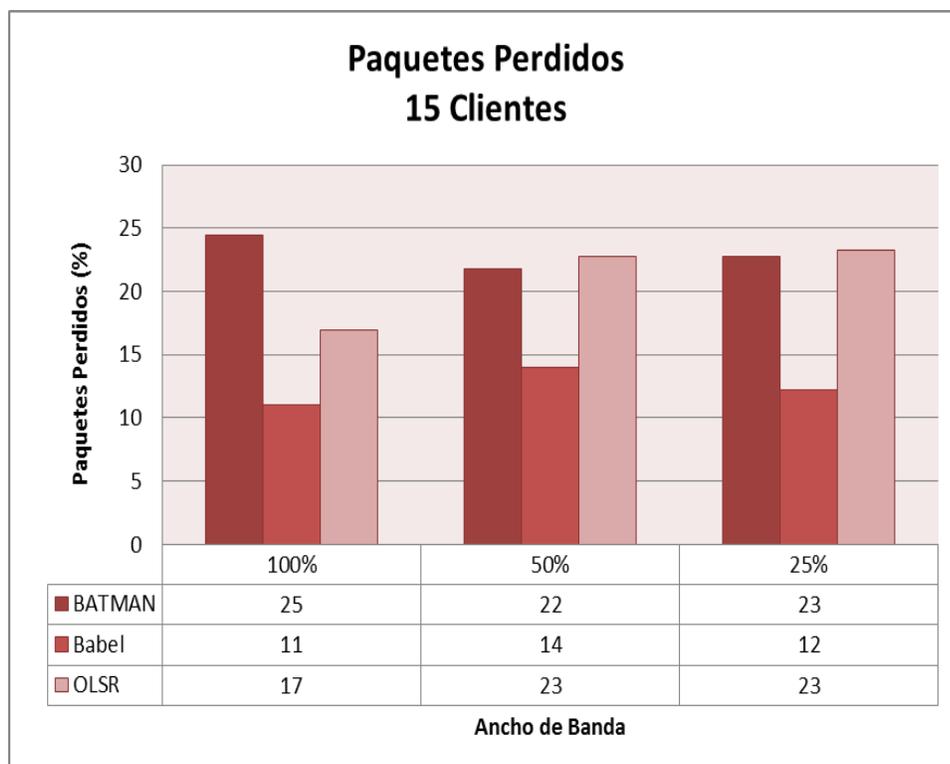


Figura 34. – Gráfica de paquetes perdidos para 15 clientes.

La figura anterior muestra que para cada protocolo de enrutamiento utilizado, el porcentaje de paquetes perdidos se mantiene muy similar a medida que el ancho de banda es restringido en el borde. El protocolo con el que se obtiene una menor tasa de paquetes perdidos independientemente del ancho de banda del cuál dispone la red mallada, es Babel, siendo en promedio un 35% menor que OLSR y 75% menor que BATMAN.

5.3. Resultados – Escenarios 7, 8 y 9

A continuación se presentan los parámetros de rendimiento de las pruebas realizadas sobre los escenarios 7, 8 y 9, es decir, en una red mallada con 25 clientes, con los protocolos de enrutamiento BATMAN, Babel y OLSR y paquetes de 64, 256, 1024 y 1518 bytes, agrupados por la métrica de desempeño analizada.

5.3.1. Throughput

Desde la Figura 35 a la Figura 38 se muestra el *throughput* en función de la cantidad de ancho de banda disponible en el borde de la red para la conexión a Internet para tamaño de paquetes de 64, 256, 1024 y 1518 bytes en una red mallada con 25 clientes.

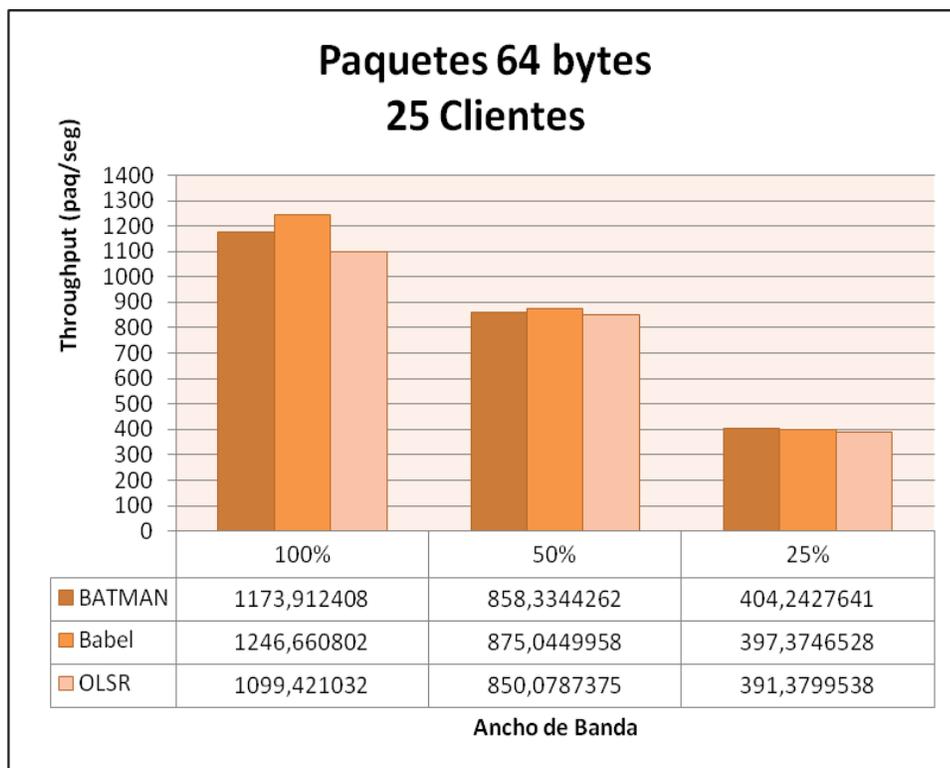


Figura 35. – Gráfica de *throughput* con paquetes de 64 bytes y 25 clientes.

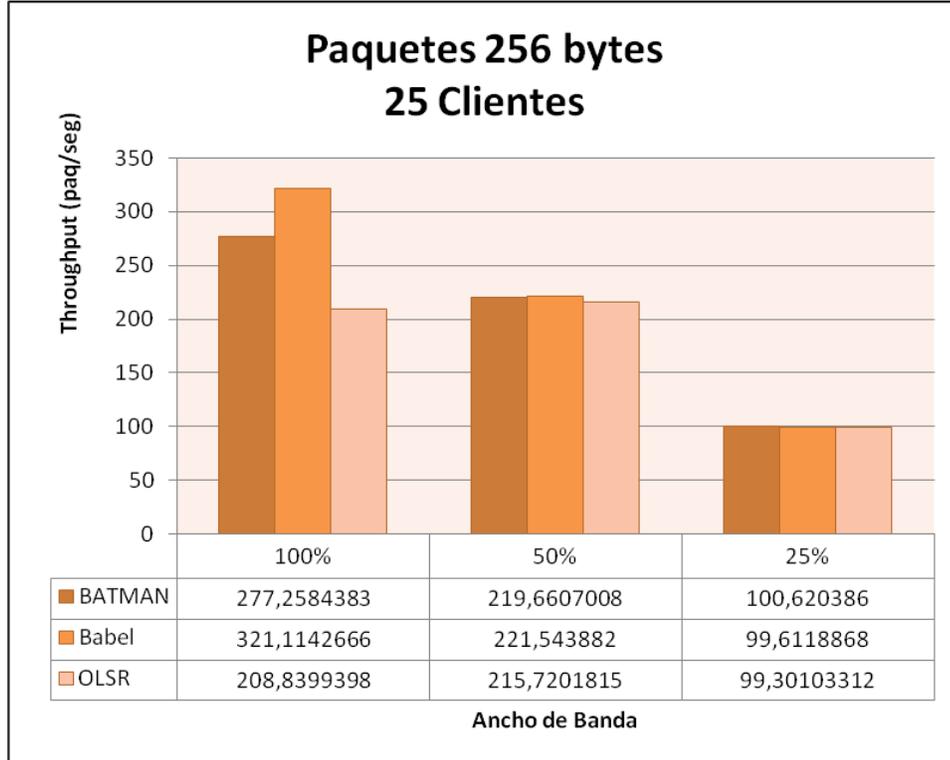


Figura 36. – Gráfica de *throughput* con paquetes de 256 bytes y 25 clientes.

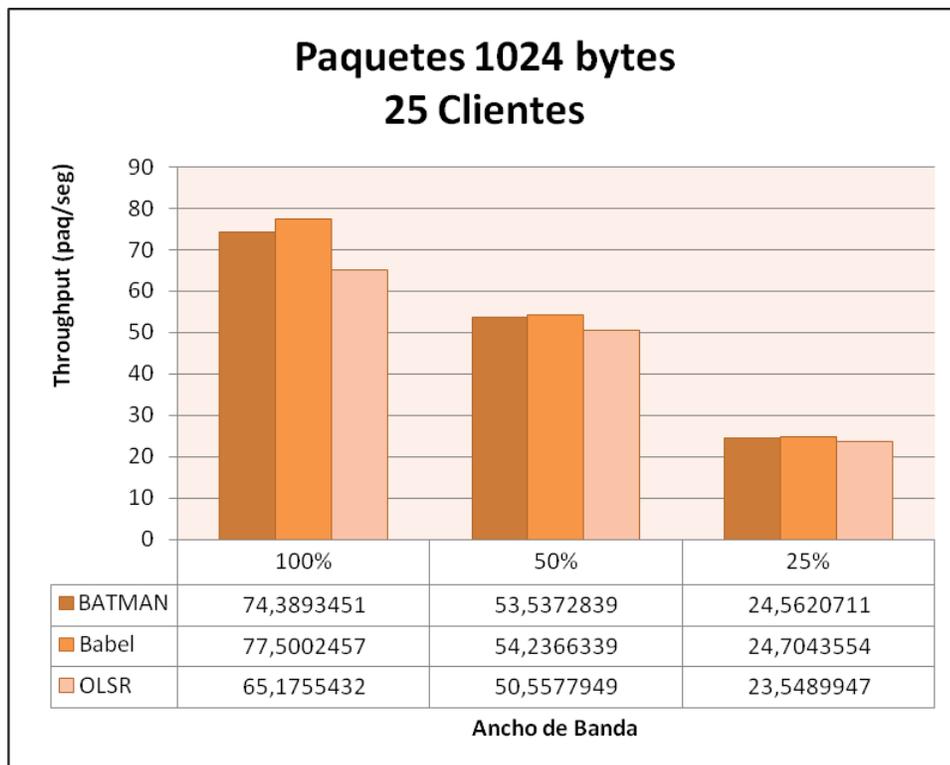


Figura 37. – Gráfica de *throughput* con paquetes de 1024 bytes y 25 clientes.

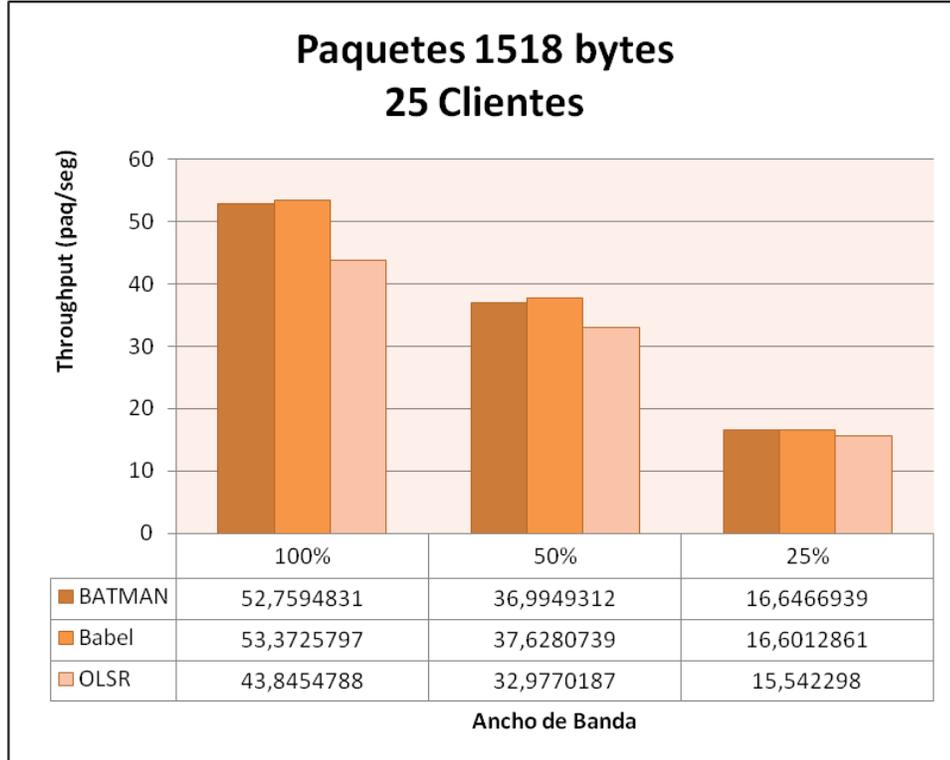


Figura 38. – Gráfica de *throughput* con paquetes de 1518 bytes y 25 clientes.

De los resultados mostrados anteriormente en las figuras, se puede observar que el *throughput* decrece a medida que el ancho de banda es restringido en el borde, como se esperaría. También el *throughput* decrece a medida que aumenta el tamaño de los paquetes. El rendimiento de la red en términos del *throughput* es similar independientemente del protocolo de enrutamiento utilizado. Sólo en el caso cuando la red dispone del 100% del ancho de banda, ésta muestra un mejor desempeño cuando se usa el protocolo Babel.

Es decir cuando se aumenta el número de nodos de la red mallada a 25 y no se aplica restricción en el ancho de banda el protocolo que alcanza una mayor tasa de paquetes por segundo es Babel, mientras que al aplicar algún tipo de restricción en el ancho de banda todos los protocolos alcanzan una tasa de paquetes por segundo similar aunque ligeramente mayor con Babel.

5.3.2. Jitter

Desde la Figura 39 a la Figura 42 se muestra el *jitter* en función de la cantidad de ancho de banda disponible en el borde de la red para la conexión a Internet para tamaño de paquetes de 64, 256, 1024 y 1518 bytes en una red mallada con 25 clientes.

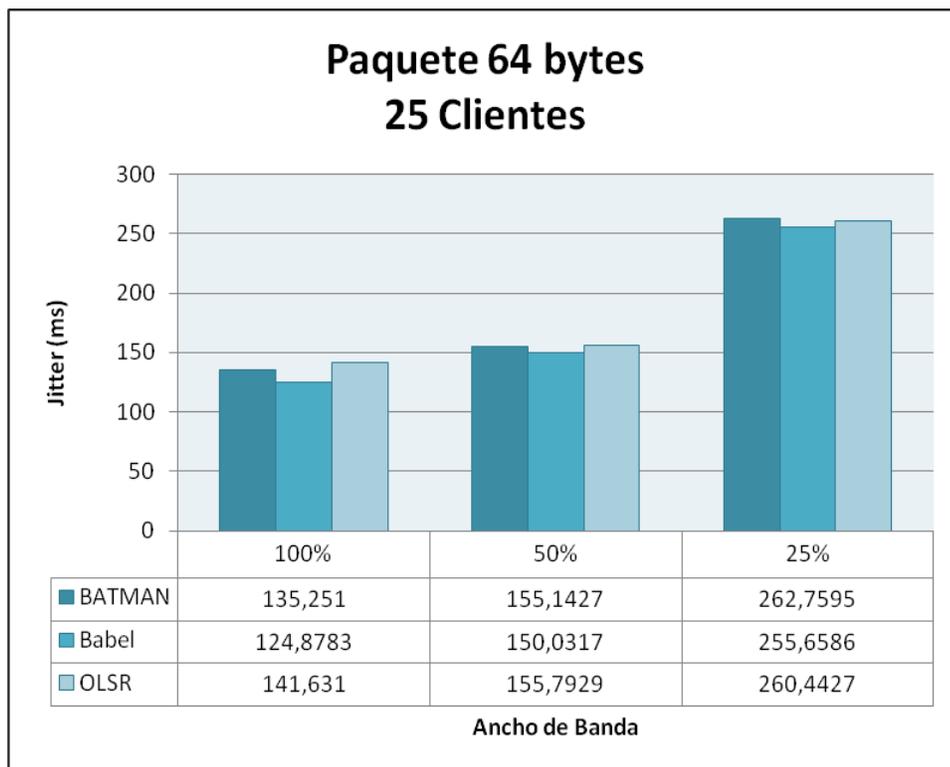


Figura 39. – Gráfica de *jitter* con paquetes de 64 bytes y 25 clientes.

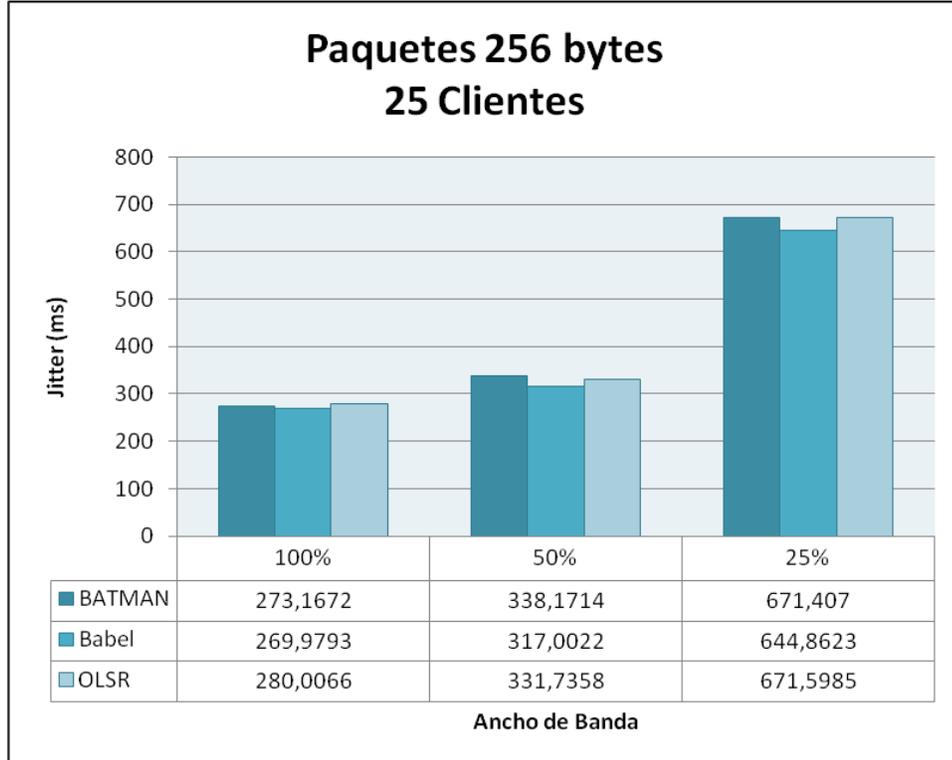


Figura 40. – Gráfica de *jitter* con paquetes de 256 bytes y 25 clientes.

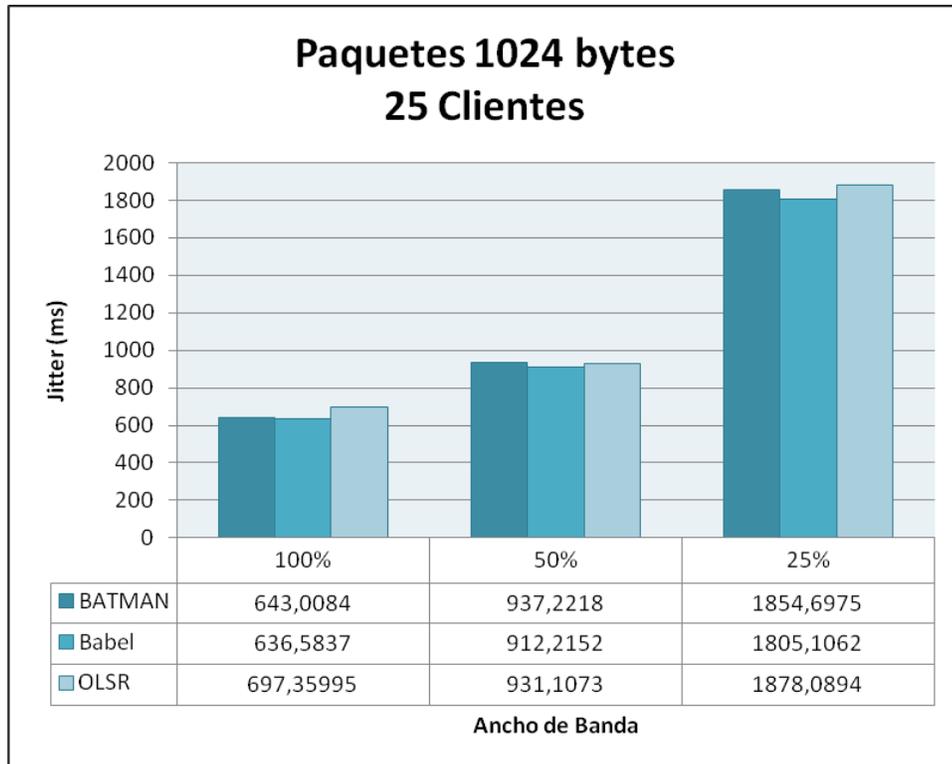


Figura 41. – Gráfica de *jitter* con paquetes de 1024 bytes y 25 clientes.

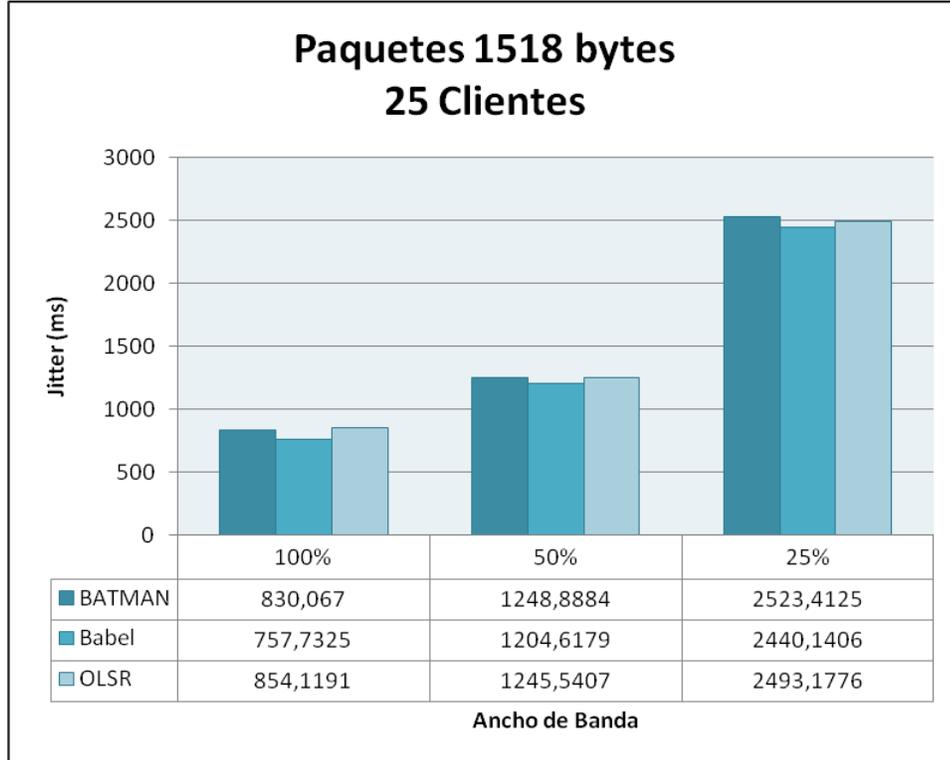


Figura 42. – Gráfica de *jitter* con paquetes de 1518 bytes y 25 clientes.

De los resultados mostrados anteriormente en las figuras, se puede observar que al restringir el ancho de banda, todos los protocolos aumentan los valores de *jitter* obtenidos, en mayor proporción cuando se trata de paquetes de mayor tamaño. El protocolo con el que se obtiene un menor valor de *jitter* es Babel, tanto para cuando se aplica restricción en el ancho de banda como cuando no se aplica.

5.3.3. Retardo

Desde la Figura 43 a la Figura 46 se muestra el retardo en función de la cantidad de ancho de banda disponible en el borde de la red para la conexión a Internet para tamaño de paquetes de 64, 256, 1024 y 1518 en una red mallada con 25 clientes.

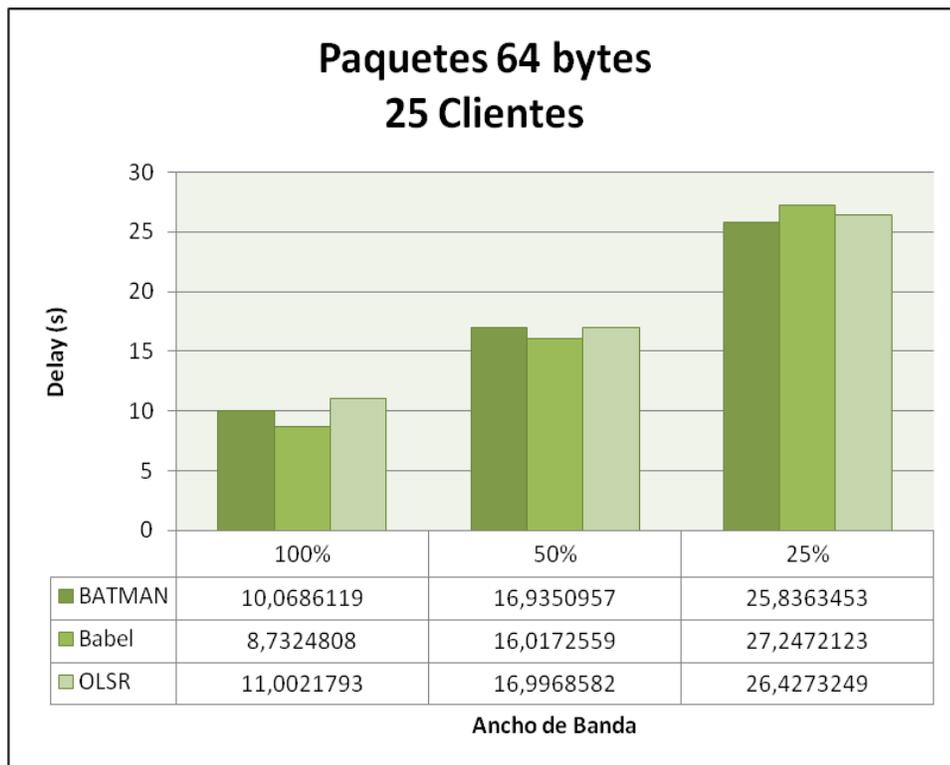


Figura 43. – Gráfica de retardo con paquetes de 64 bytes y 25 clientes.

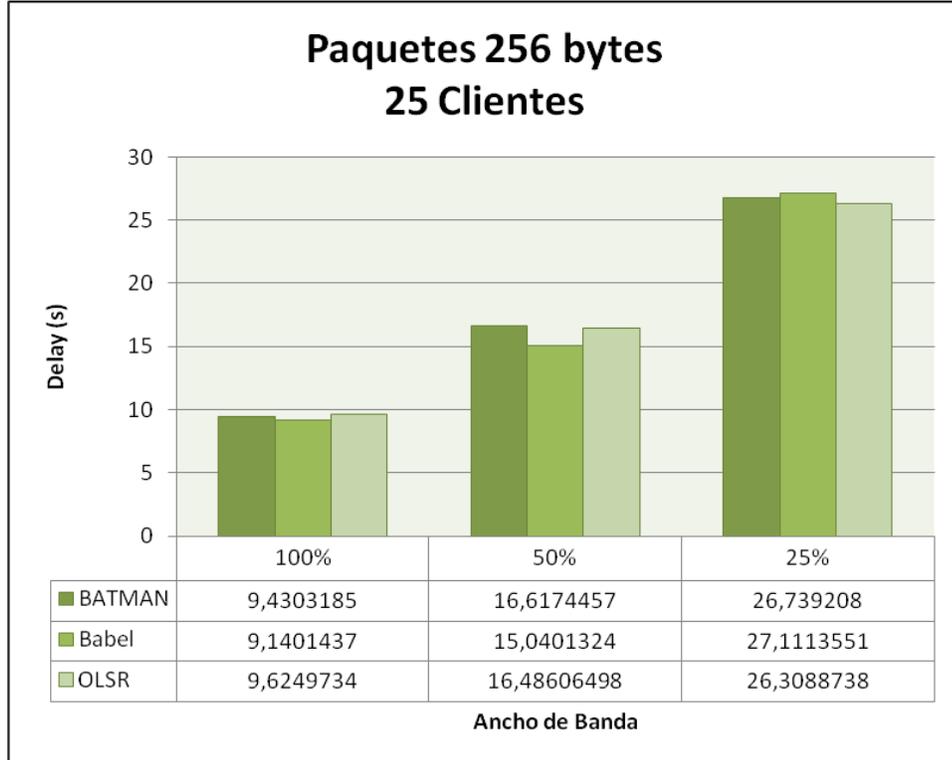


Figura 44. – Gráfica de retardo con paquetes de 256 bytes y 25 clientes.

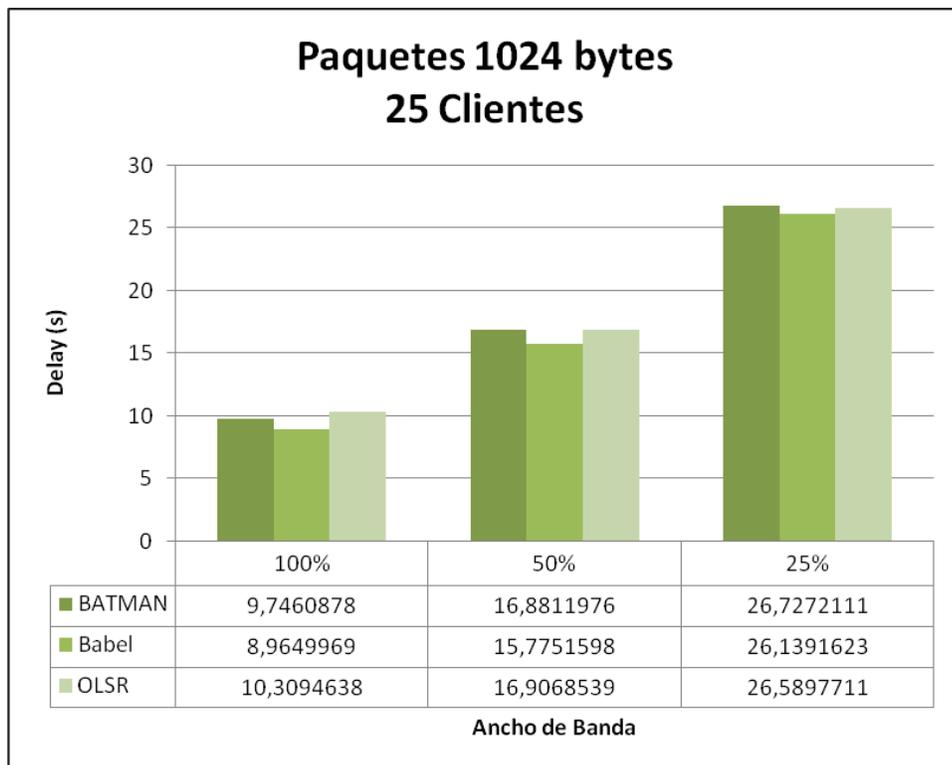


Figura 45. – Gráfica de retardo con paquetes de 1024 bytes y 25 clientes.

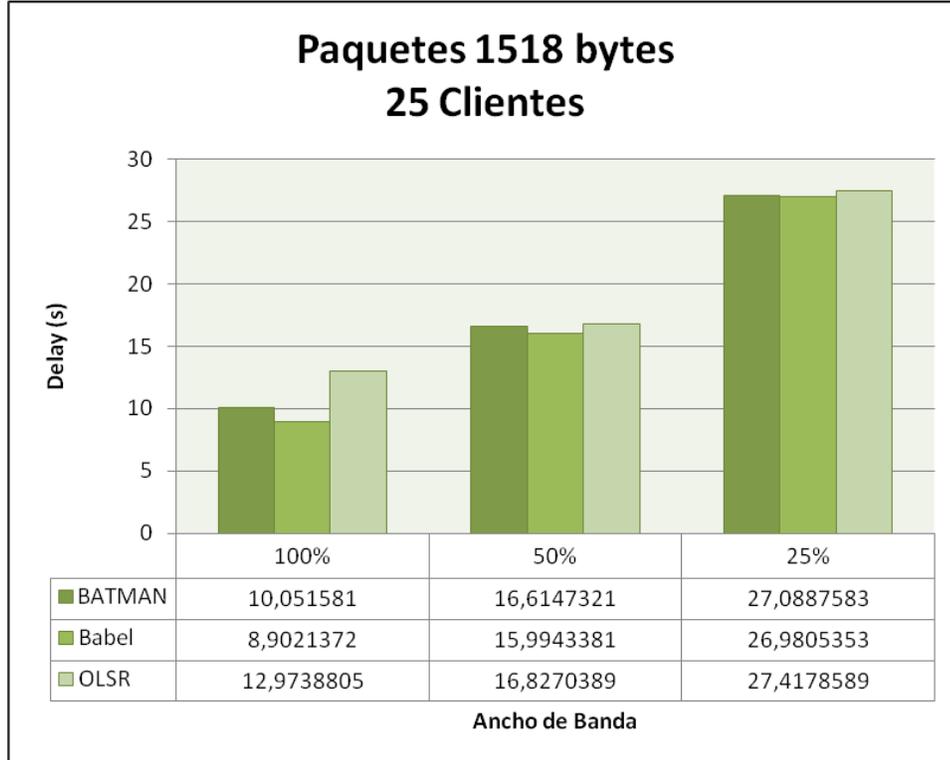


Figura 46. – Gráfica de retardo con paquetes de 1518 bytes y 25 clientes.

De los resultados mostrados anteriormente en las figuras, se puede observar que el retardo aumenta a medida que el ancho de banda es restringido en el borde. En este caso, el protocolo con el que se obtienen valores ligeramente menores en la mayoría de las pruebas es con Babel.

Es decir en una red mallada con 25 clientes cuando no se aplica restricción en el ancho de banda el protocolo con los que se obtiene un menor retardo es Babel. Al aplicar una restricción con los tres protocolos se obtienen valores de retardo muy similares observándose en la mayoría de las pruebas una leve mejoría con el protocolo Babel.

5.3.4. Paquetes Perdidos

En la Figura 47 se muestra el promedio de pérdida de paquetes en función de la cantidad de ancho de banda disponible en el borde de la red para la conexión a Internet en una red mallada con 25 clientes.

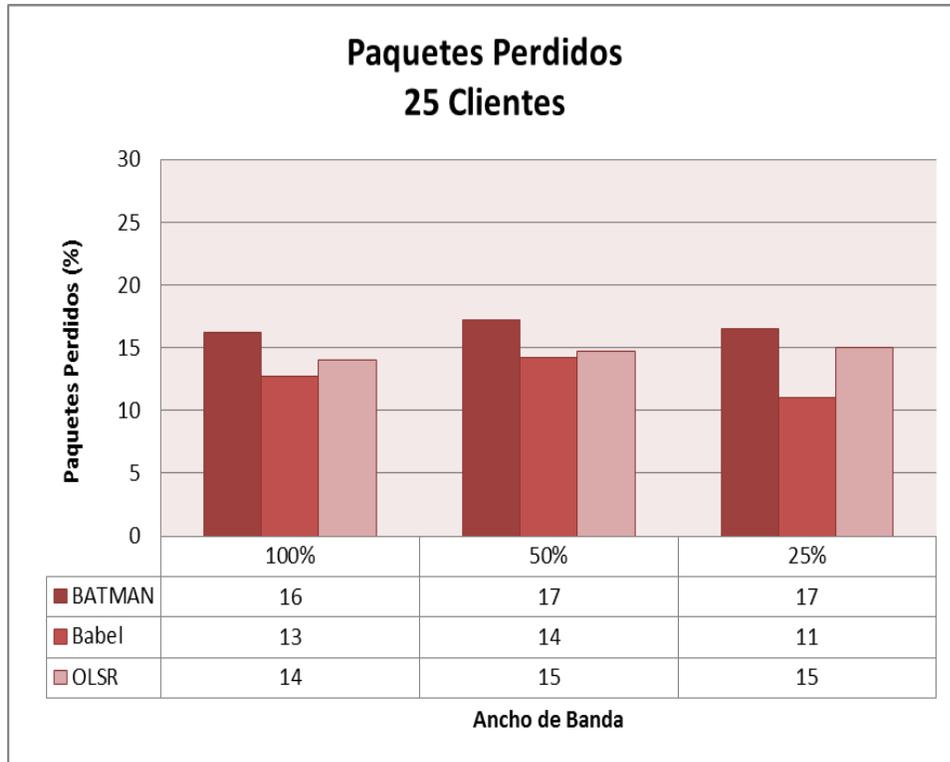


Figura 47. – Gráfica de paquetes perdidos para 25 clientes.

La figura anterior muestra que para cada protocolo de enrutamiento utilizado, el porcentaje de paquetes perdidos se mantiene muy similar a medida que el ancho de banda es restringido en el borde. Se observa que el protocolo con el que se obtiene una menor tasa de paquetes perdidos independientemente del ancho de banda del cual dispone la red mallada, es Babel, siendo en promedio un 9% menor que OLSR y 21% menor que BATMAN.

5.4. Resultados – Escenarios 1, 4 y 7

A continuación se presentan los parámetros de rendimiento de las pruebas realizadas sobre los escenarios 1, 4 y 7, es decir, en una red mallada que dispone del 25% del ancho de banda total, con los protocolos de enrutamiento BATMAN, Babel y OLSR y paquetes de 64, 256, 1024 y 1518 bytes, agrupados por la métrica de desempeño analizada.

5.4.1. Throughput

Desde la Figura 48 a la Figura 51 se muestra el *throughput* en función de la cantidad de clientes pertenecientes a la red mallada para tamaño de paquetes de 64, 256, 1024 y 1518 bytes en una red mallada que dispone del 25% del ancho de banda total.

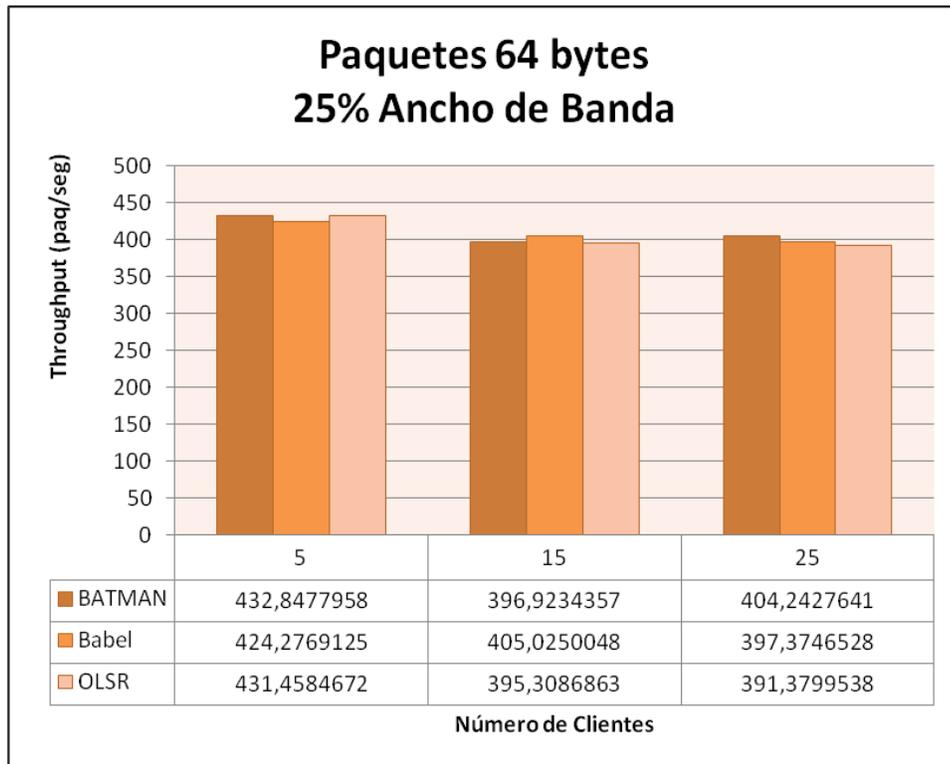


Figura 48. – Gráfica de *throughput* con paquetes de 64 bytes y 25% de ancho de banda.

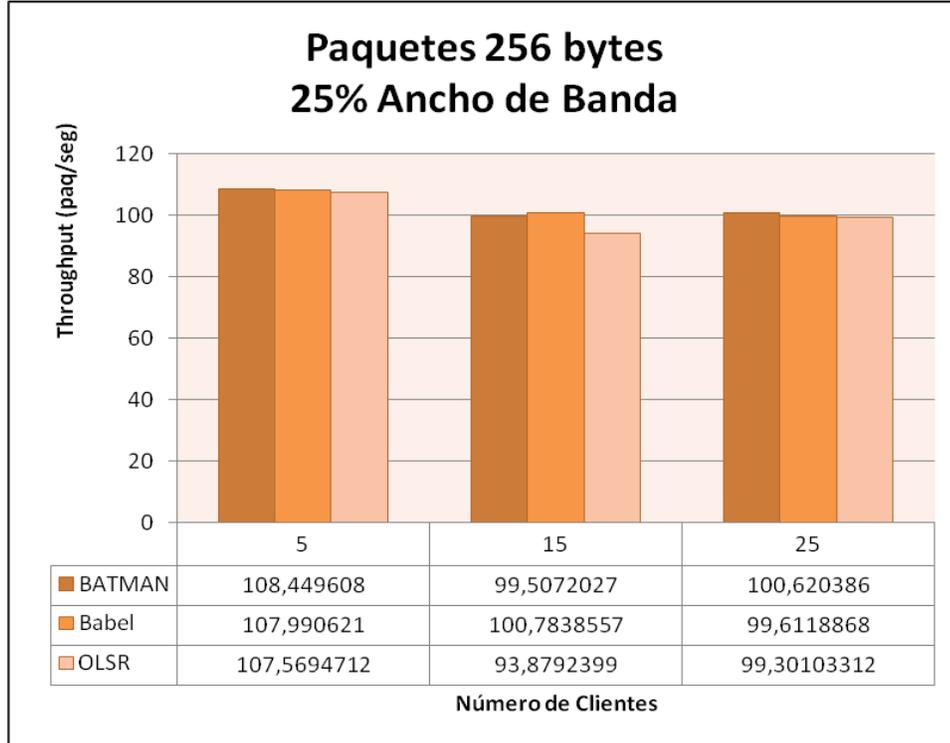


Figura 49. – Gráfica de *throughput* con paquetes de 256 bytes y 25% de ancho de banda.

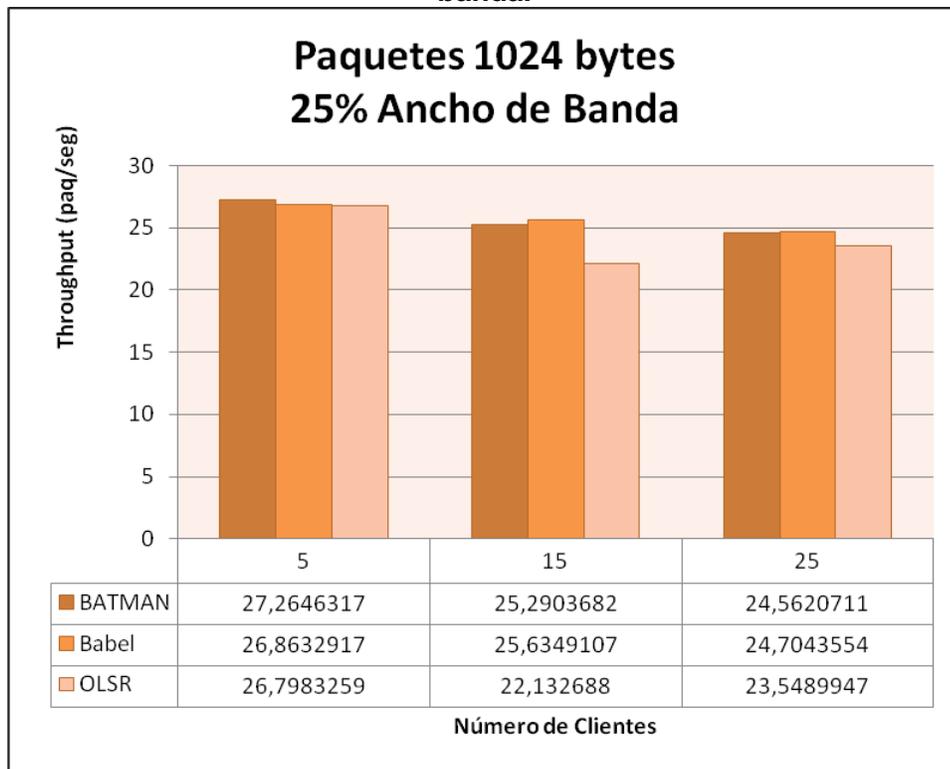


Figura 50. – Gráfica de *throughput* con paquetes de 1024 bytes y 25% de ancho de banda.

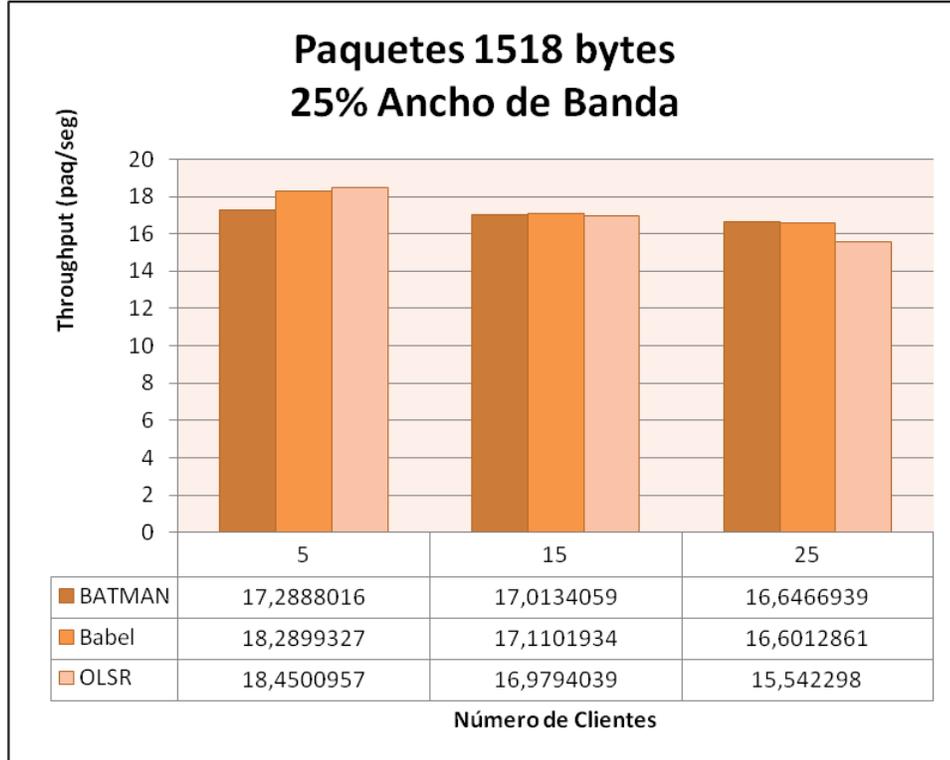


Figura 51. – Gráfica de *throughput* con paquetes de 1518 bytes y 25% de ancho de banda.

De los resultados mostrados anteriormente en las figuras, se puede observar que para cada protocolo de enrutamiento utilizado el *throughput* decrece a medida que aumenta el tamaño de los paquetes y no se ve realmente afectado por el número de clientes en la red mallada.

Al restringir el ancho de banda en un 75% en una red mallada, el rendimiento de la red en términos del *throughput* es similar independientemente del protocolo de enrutamiento utilizado y el número de clientes en la red mallada.

5.4.2. Jitter

Desde la Figura 52 a la Figura 55 se muestra el *jitter* en función de la cantidad de clientes pertenecientes a la red mallada para tamaño de paquetes de 64, 256, 1024 y 1518 bytes en una red mallada que dispone del 25% del ancho de banda total.

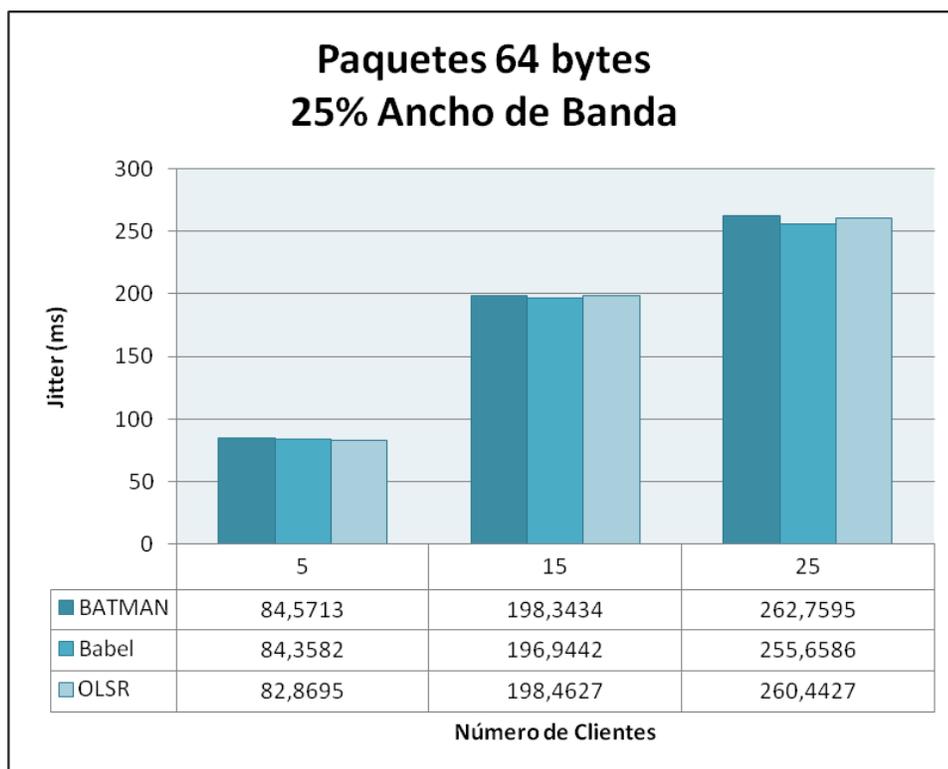


Figura 52. – Gráfica de *jitter* con paquetes de 64 bytes y 25% de ancho de banda.

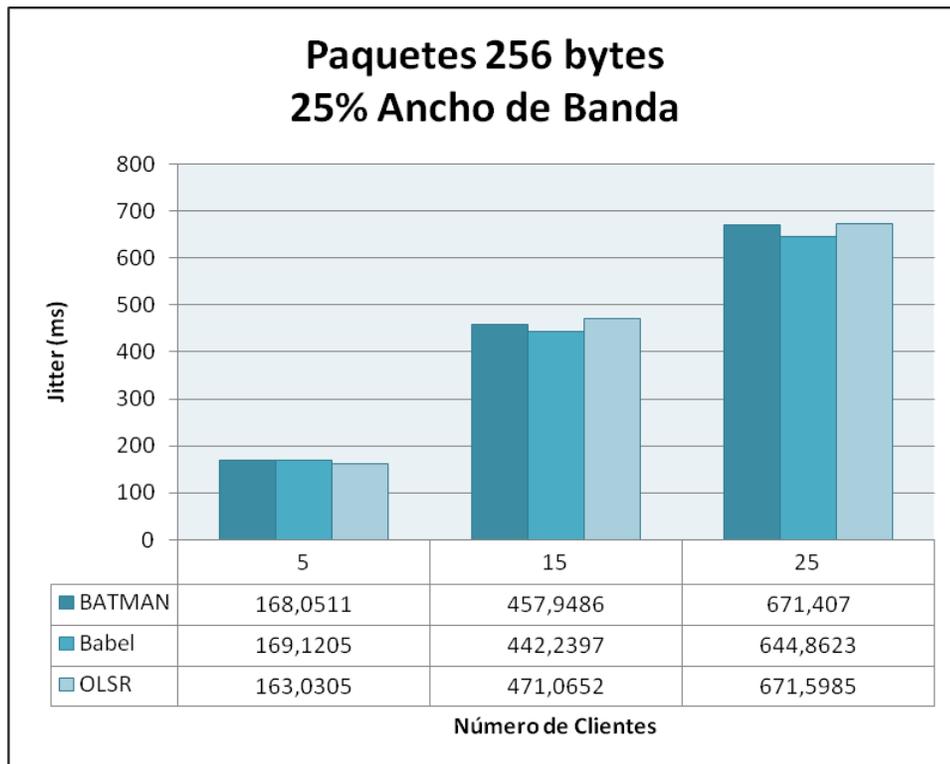


Figura 53. – Gráfica de *jitter* con paquetes de 256 bytes y 25% de ancho de banda.

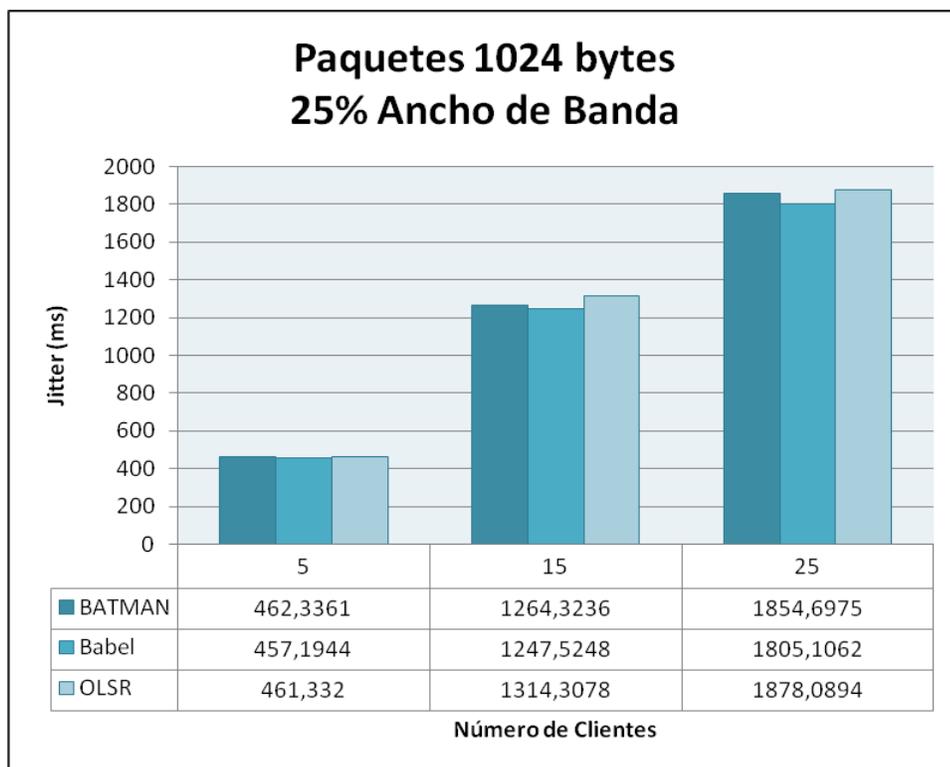


Figura 54. – Gráfica de *jitter* con paquetes de 1024 bytes y 25% de ancho de banda.

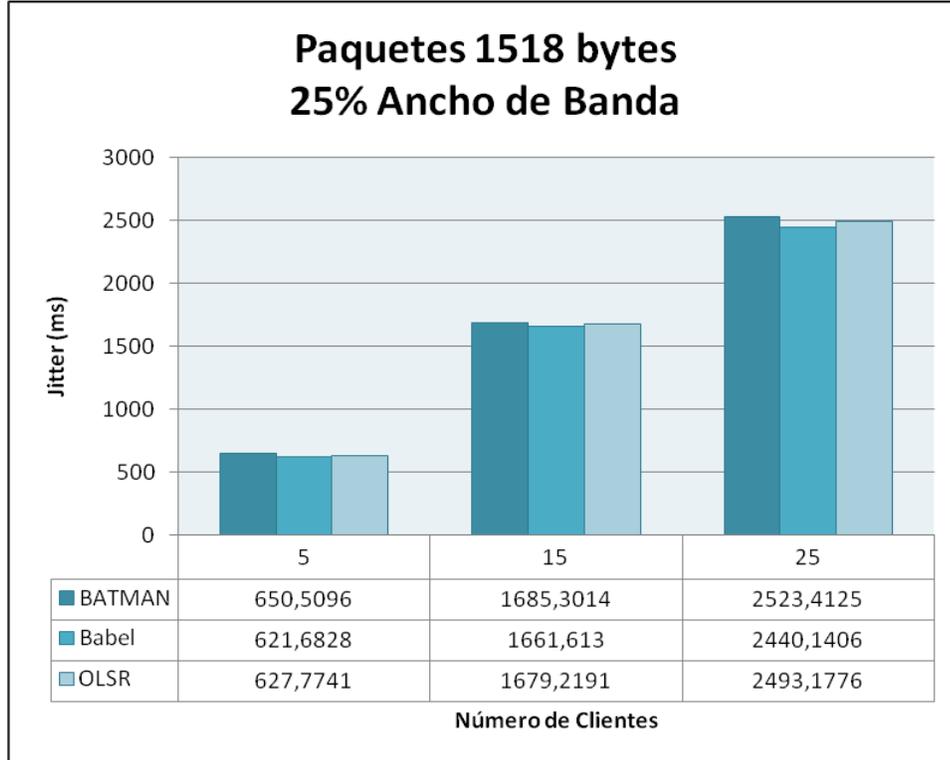


Figura 55. – Gráfica de *jitter* con paquetes de 1518 bytes y 25% de ancho de banda.

De los resultados mostrados anteriormente en las figuras se puede observar que al aumentar el número de clientes de la red mallada, todos los protocolos aumentan los valores de *jitter* obtenidos, en mayor proporción cuando se trata de paquetes de mayor tamaño.

Al restringir el ancho de banda en un 75% en una red mallada con pocos clientes, con todos los protocolos se obtienen valores de *jitter* muy similares. Al aumentar el número de clientes a 15 y a 25, el protocolo con el que se obtienen menores valores de *jitter* es Babel.

5.4.3. Retardo

Desde la Figura 56 a la Figura 59 se muestra el retardo en función de la cantidad de clientes pertenecientes a la red mallada para tamaño de paquetes de 64, 256, 1024 y 1518 bytes en una red mallada que dispone del 25% del ancho de banda total.

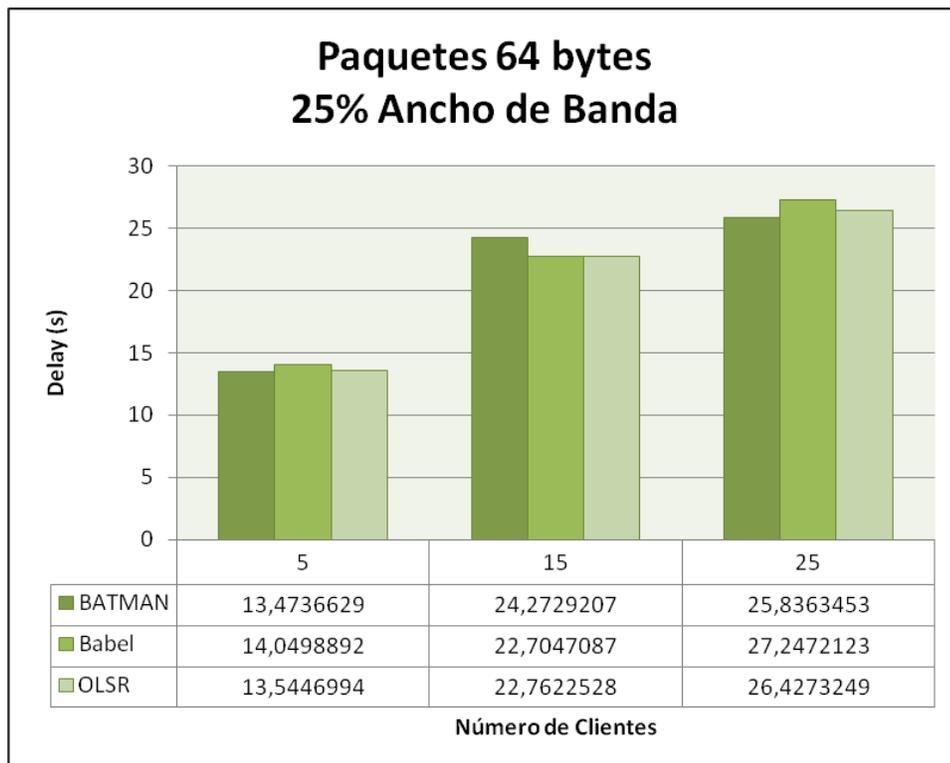


Figura 56. – Gráfica de retardo con paquetes de 64 bytes y 25% de ancho de banda.

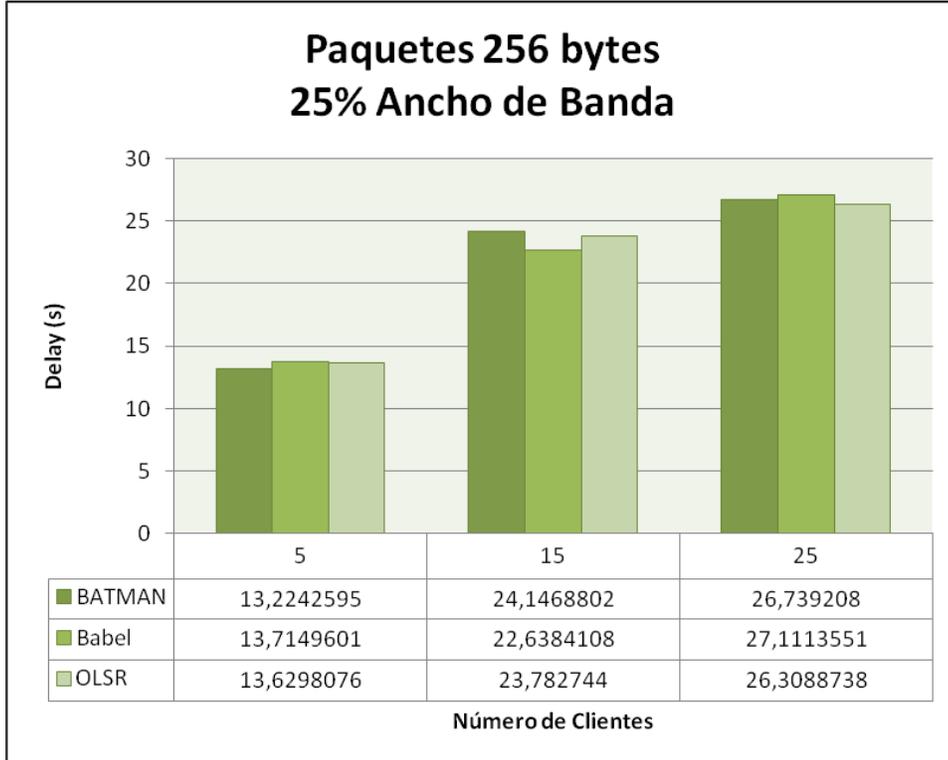


Figura 57. – Gráfica de retardo con paquetes de 256 bytes y 25% de ancho de banda.

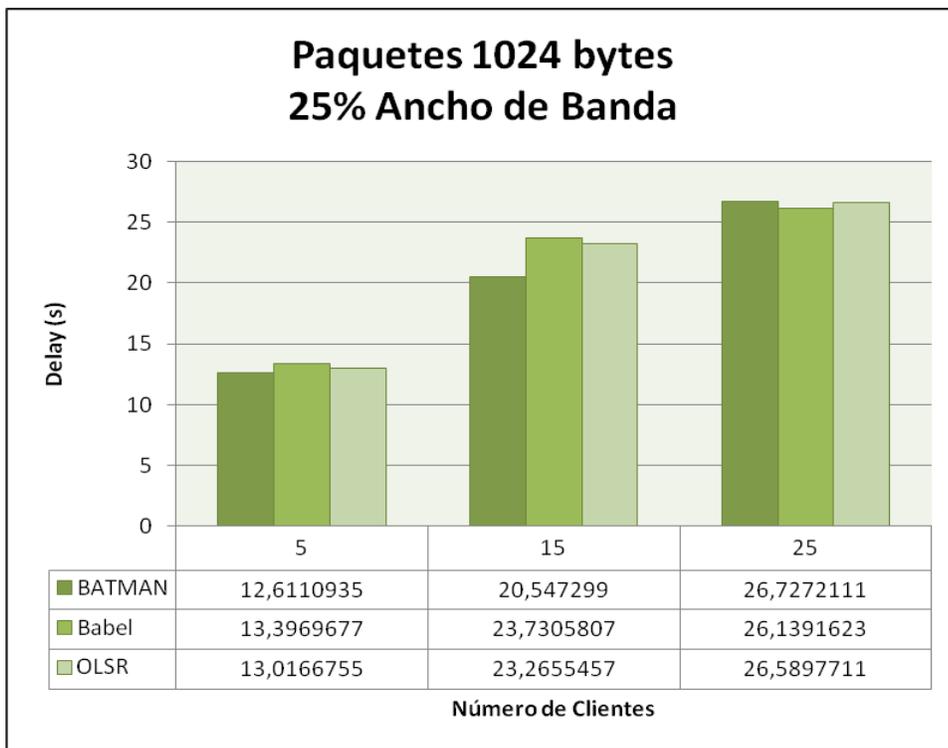


Figura 58. – Gráfica de retardo con paquetes de 1024 bytes y 25% de ancho de banda.

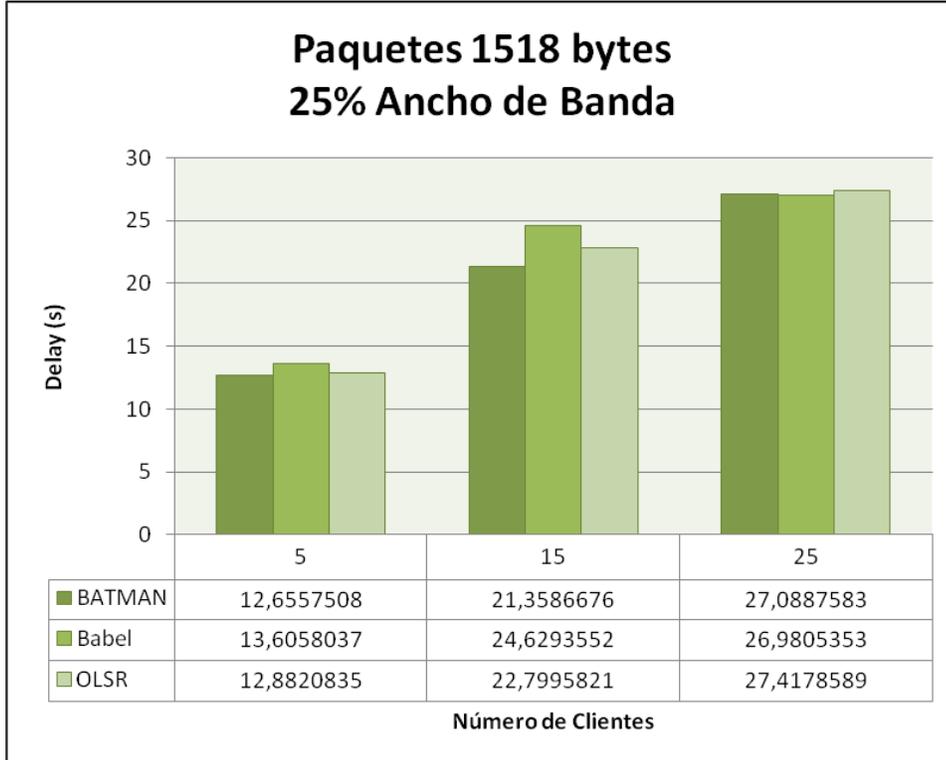


Figura 59. – Gráfica de retardo con paquetes de 1518 bytes y 25% de ancho de banda.

De los resultados mostrados anteriormente en las figuras, se puede observar que al aumentar el número de clientes de la red mallada, todos los protocolos aumentan los valores de retardo obtenidos.

Al restringir el ancho de banda en un 75% en una red mallada con 5 clientes, con todos los protocolos se obtienen valores de retardo muy similares, observándose una leve mejoría con el protocolo BATMAN. Al aumentar el número de clientes a 15, se obtiene un menor valor de retardo para paquetes de 1024 y 1518 bytes con el protocolo BATMAN, mientras que en el resto de las pruebas se obtiene con el protocolo Babel. Al aumentar el número de clientes a 25, se obtienen valores muy similares con todos los protocolos.

5.4.4. Paquetes Perdidos

En la Figura 60 se muestra el porcentaje de paquetes perdidos en función de la cantidad de clientes pertenecientes a una red mallada que dispone del 25% del ancho de banda total.

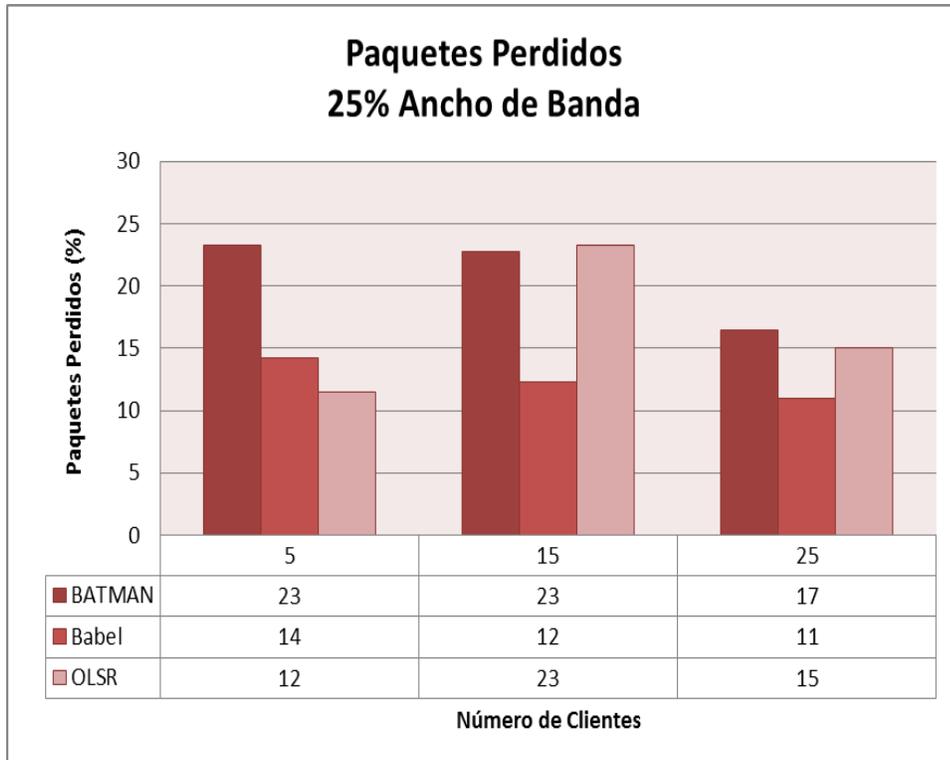


Figura 60. – Gráfica de paquetes perdidos con 25% de ancho de banda

La figura anterior muestra que al restringir el ancho de banda en 75% en una red mallada con 5 clientes, se obtiene una menor pérdida de paquetes con el protocolo OLSR. Al aumentar el número de clientes a 15, el protocolo con el que se obtiene una menor pérdida de paquetes es Babel, comportamiento que se mantiene al aumentar el número de clientes a 25.

5.5. Resultados – Escenarios 2, 5 y 8

A continuación se presentan los parámetros de rendimiento de las pruebas realizadas sobre los escenarios 2, 5 y 8, es decir, en una red mallada que dispone del 50% del ancho de banda total, con los protocolos de enrutamiento BATMAN, Babel y OLSR y paquetes de 64, 256, 1024 y 1518 bytes, agrupados por la métrica de desempeño analizada.

5.5.1. Throughput

Desde la Figura 61 a la Figura 64 se muestra el *throughput* en función de la cantidad de clientes pertenecientes a la red mallada para tamaño de paquetes de 64, 256, 1024 y 1518 bytes en una red mallada que dispone del 50% del ancho de banda total.

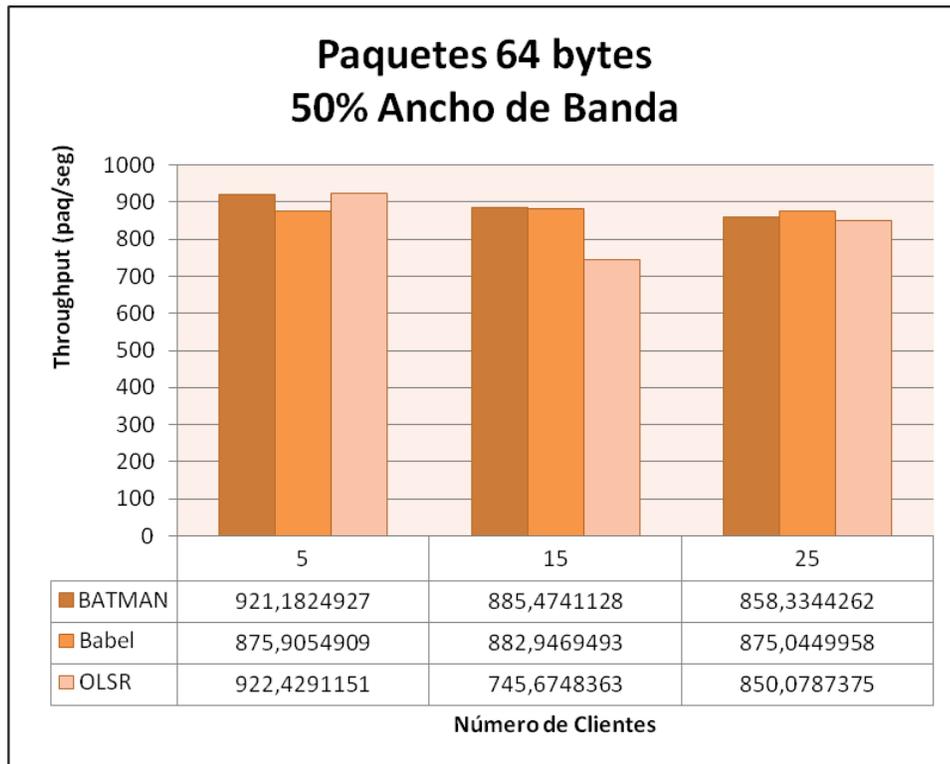


Figura 61. – Gráfica de *throughput* con paquetes de 64 bytes y 50% de ancho de banda.

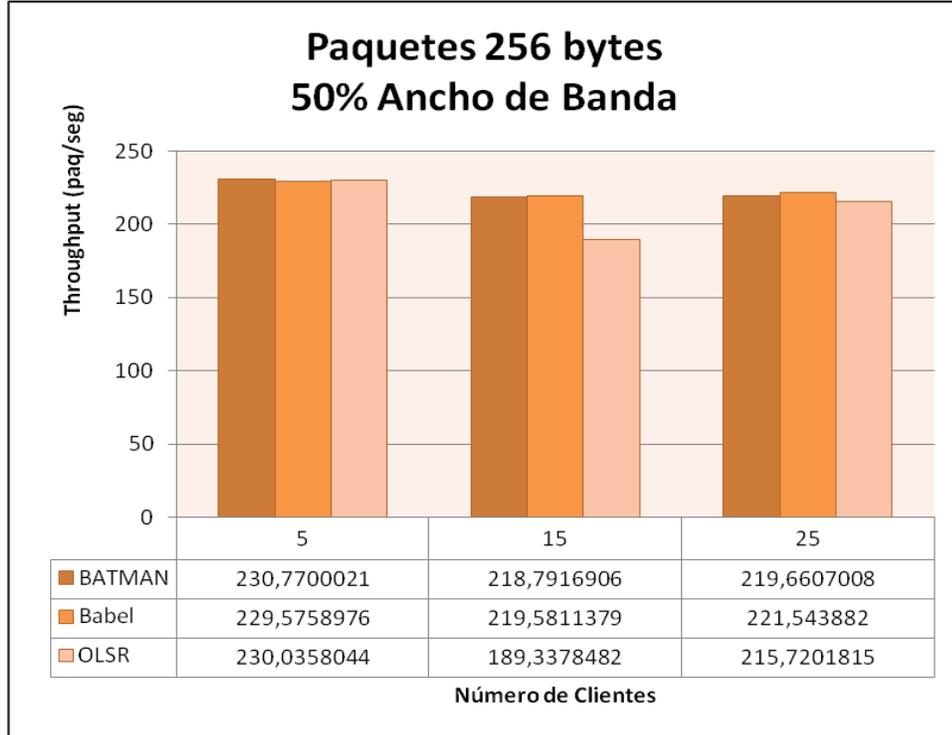


Figura 62. – Gráfica de *throughput* con paquetes de 256 bytes y 50% de ancho de banda.

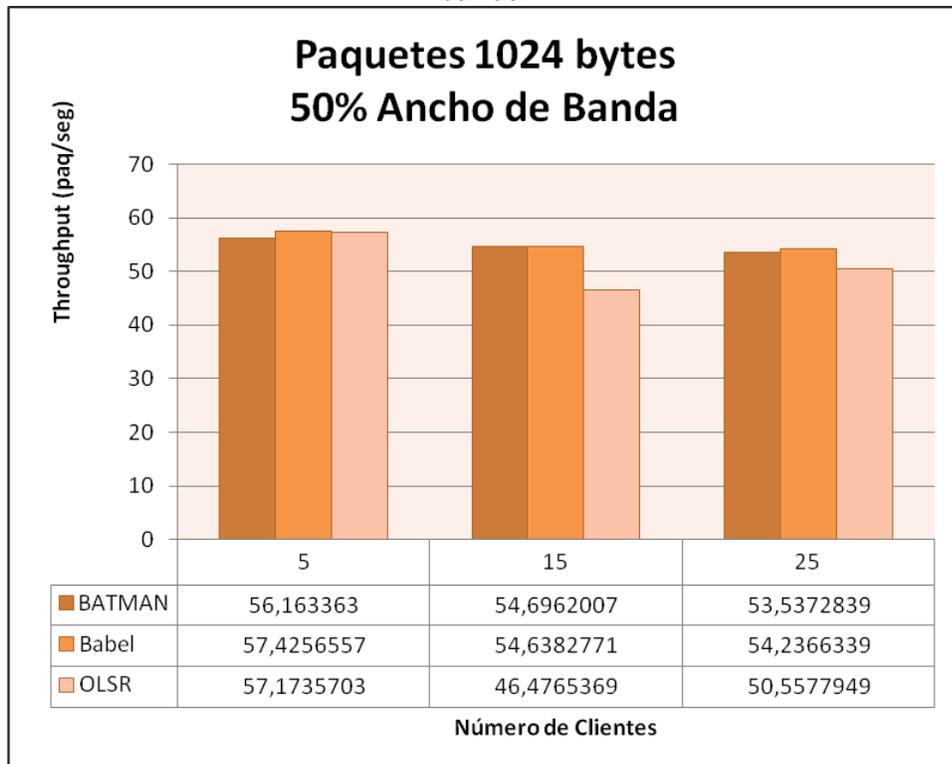


Figura 63. – Gráfica de *throughput* con paquetes de 1024 bytes y 50% de ancho de banda.

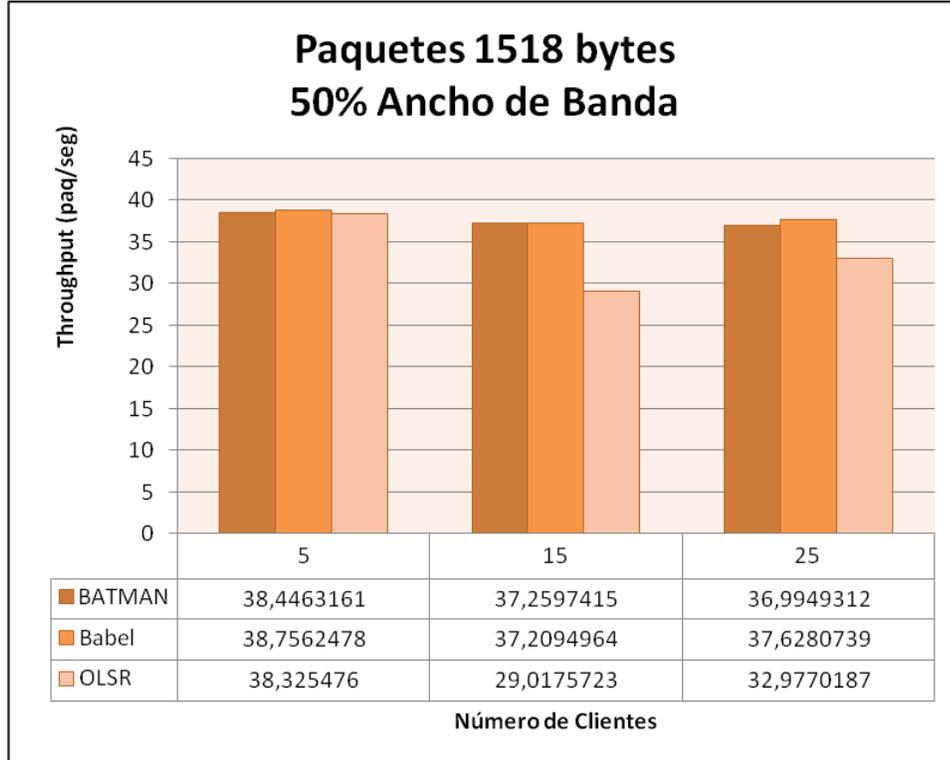


Figura 64. – Gráfica de *throughput* con paquetes de 1518 bytes y 50% de ancho de banda.

De los resultados mostrados anteriormente en las figuras, se puede observar que para cada protocolo de enrutamiento utilizado, el *throughput* decrece a medida que aumenta el tamaño de los paquetes y en general no se ve muy afectado por el número de clientes en la red mallada.

Al restringir el ancho de banda en un 50% en una red mallada de 5 nodos con todos los protocolos se obtienen una tasa de paquetes por segundo muy similar. Al aumentar el número de clientes a 15, los protocolos BATMAN y Babel obtienen una tasa de paquetes por segundo muy similar y mayor que la obtenida con el protocolo OLSR, comportamiento que se mantiene al aumentar el número de clientes a 25.

5.5.2. Jitter

Desde la Figura 65 a la Figura 68 se muestra el *jitter* en función de la cantidad de clientes pertenecientes a la red mallada para tamaño de paquetes de 64, 256, 1024 y 1518 bytes en una red mallada que dispone del 50% del ancho de banda total.

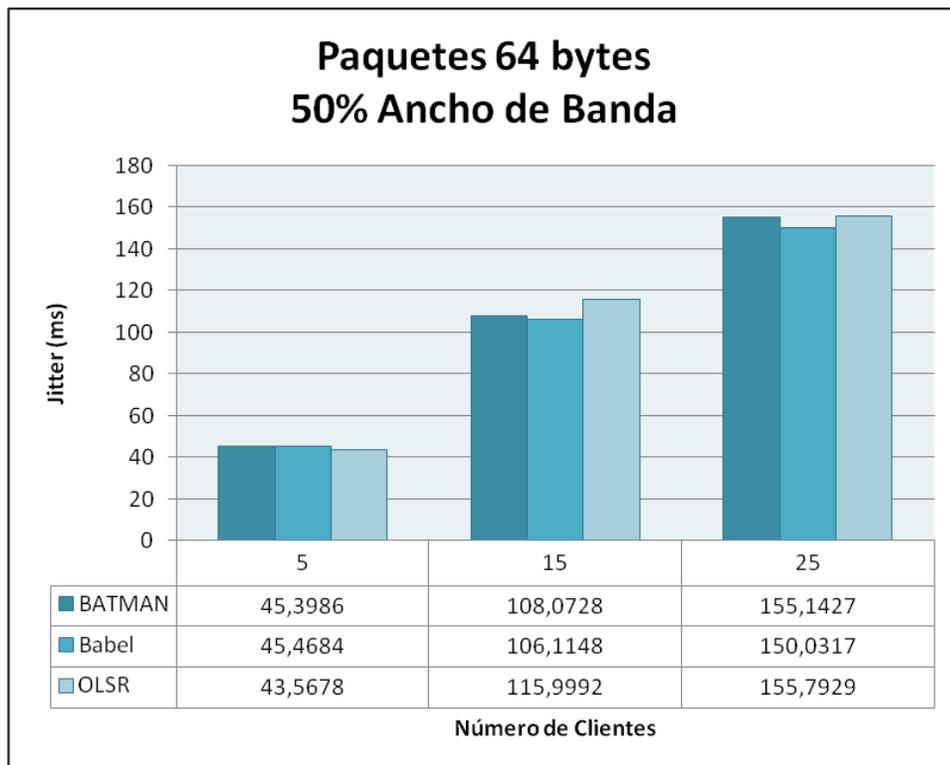


Figura 65. – Gráfica de *jitter* con paquetes de 64 bytes y 50% de ancho de banda.

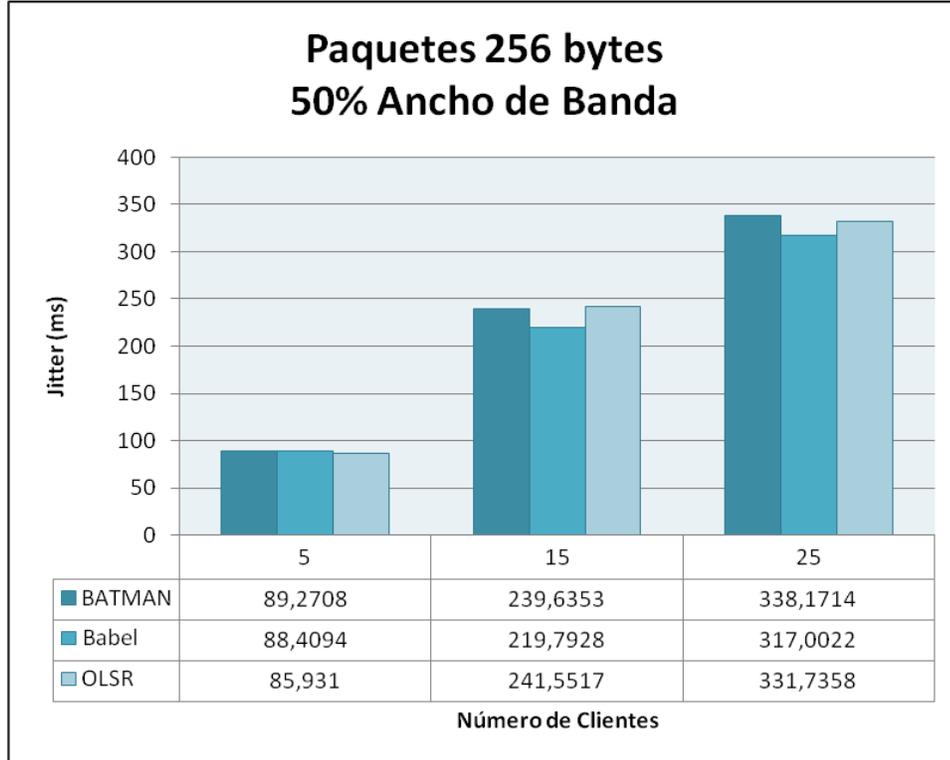


Figura 66. – Gráfica de *jitter* con paquetes de 256 bytes y 50% de ancho de banda.

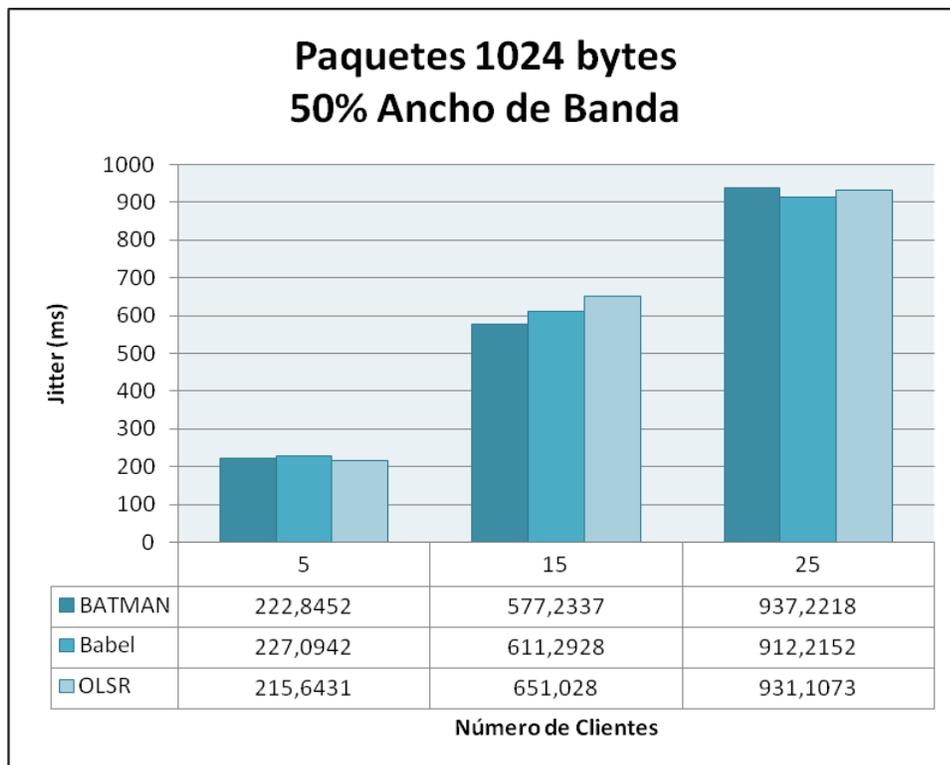


Figura 67. – Gráfica de *jitter* con paquetes de 1024 bytes y 50% de ancho de banda.

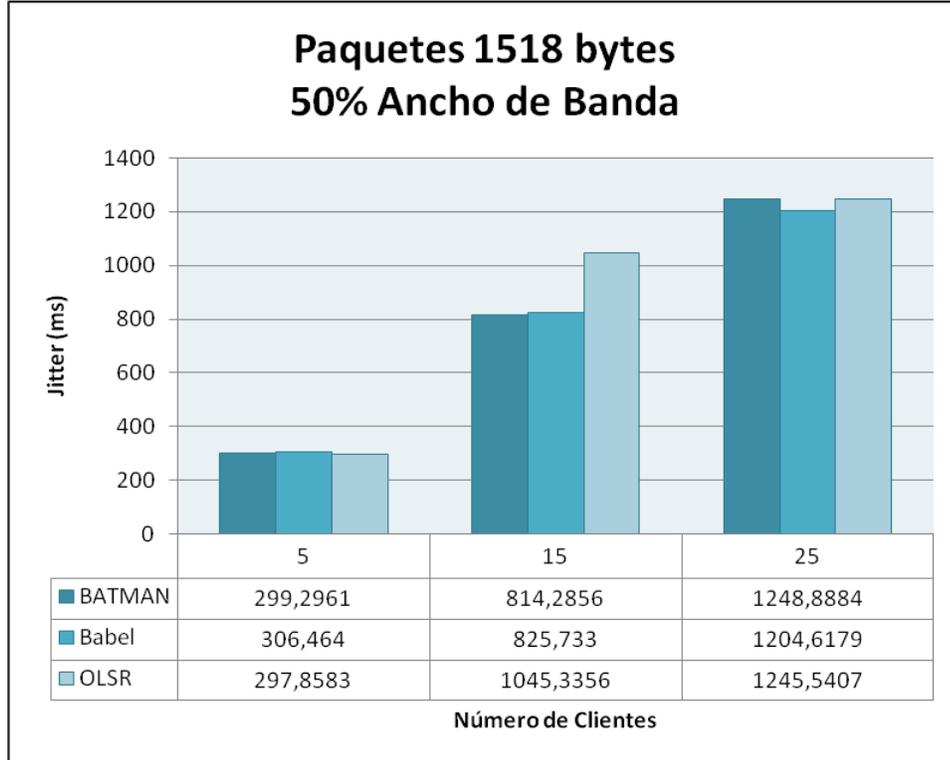


Figura 68. – Gráfica de *jitter* con paquetes de 1518 bytes y 50% de ancho de banda.

De los resultados mostrados anteriormente en las figuras, se puede observar que al aumentar el número de clientes de la red mallada, todos los protocolos aumentan los valores de *jitter* obtenidos, en mayor proporción cuando se trata de paquetes de mayor tamaño.

Al restringir el ancho de banda en un 50% en una red mallada con 5 clientes, el *jitter* obtenido es menor con el protocolo OLSR. Al aumentar el número de clientes a 15, se obtiene un menor valor de *jitter* para paquetes de 1024 y 1518 bytes con el protocolo BATMAN, mientras que en el resto de las pruebas se obtiene con el protocolo Babel. Al aumentar el número de clientes a 25 el protocolo con el que se obtuvo menor *jitter* fue Babel.

5.5.3. Retardo

Desde la Figura 69 a la Figura 72 se muestra el retardo en función de la cantidad de clientes pertenecientes a la red mallada para tamaño de paquetes de 64, 256, 1024 y 1518 bytes en una red mallada que dispone del 50% del ancho de banda total.

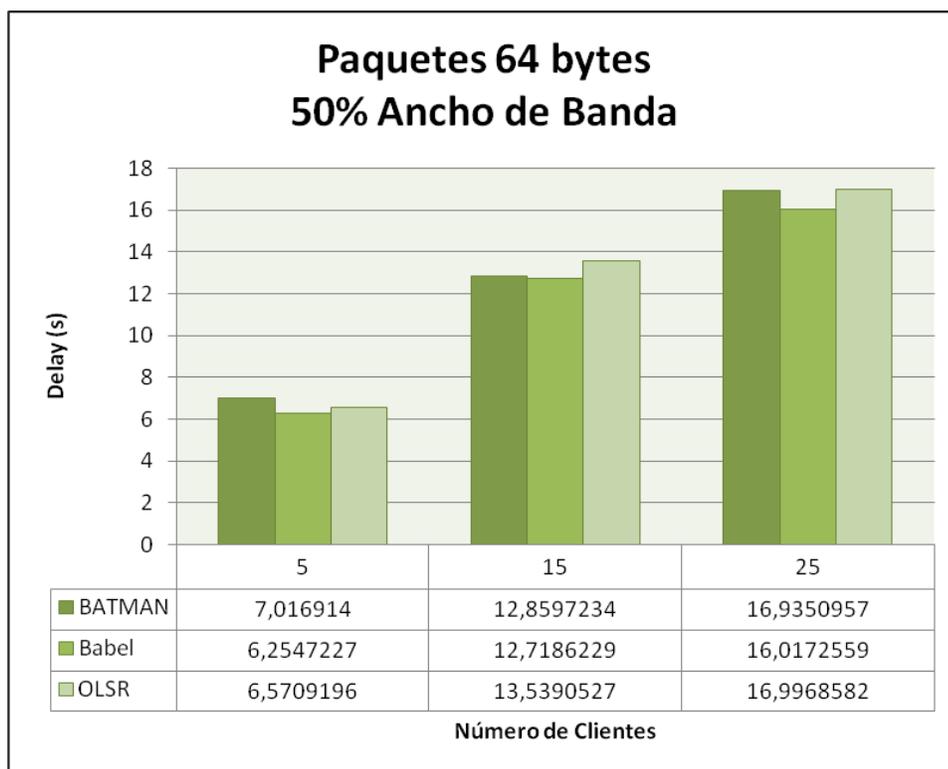


Figura 69. – Gráfica de retardo con paquetes de 64 bytes y 50% de ancho de banda.

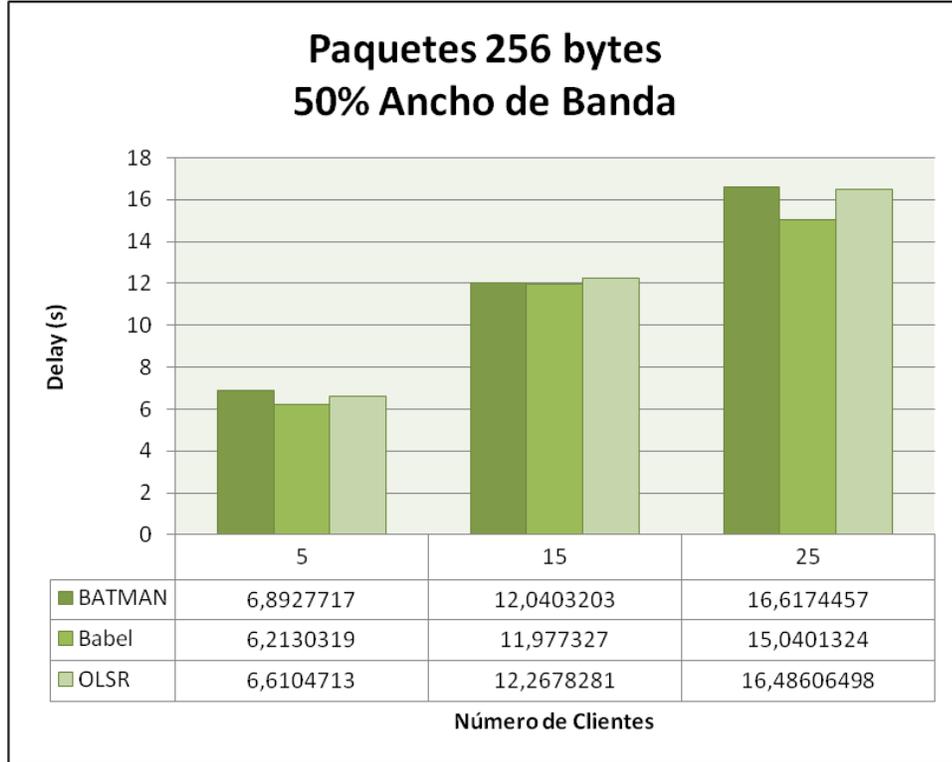


Figura 70. – Gráfica de retardo con paquetes de 256 bytes y 50% de ancho de banda.

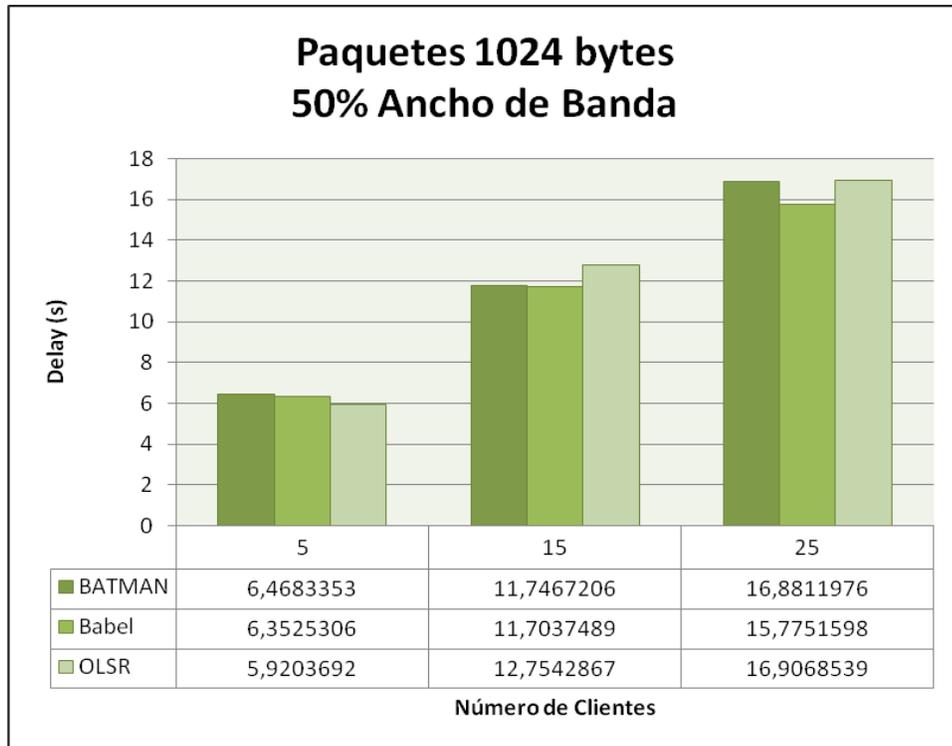


Figura 71. – Gráfica de retardo con paquetes de 1024 bytes y 50% de ancho de banda.

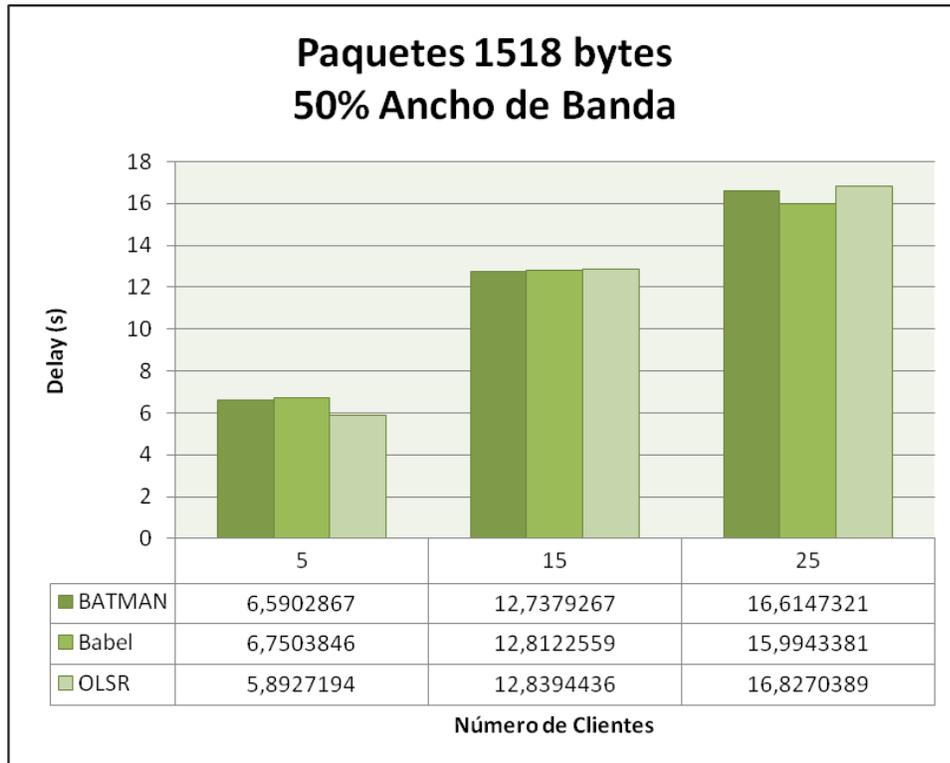


Figura 72. – Gráfica de retardo con paquetes de 1518 bytes y 50% de ancho de banda.

De los resultados mostrados anteriormente en las figuras se puede observar que al aumentar el número de clientes de la red mallada, todos los protocolos aumentan los valores de retardo obtenidos.

Al restringir el ancho de banda en un 50% en una red mallada con 5 clientes, el retardo obtenido es similar independientemente del protocolo de enrutamiento utilizado, obteniéndose valores ligeramente menores con Babel y OLSR. Al aumentar el número de clientes a 15 se obtienen valores muy similares con todos los protocolos. Al aumentar el número de clientes a 25, se obtiene en la mayoría de las pruebas valores ligeramente menores con el protocolo Babel.

5.5.4. Paquetes Perdidos

En la Figura 73 se muestra el porcentaje de paquetes perdidos en función de la cantidad de clientes pertenecientes a una red mallada que dispone del 50% del ancho de banda total.

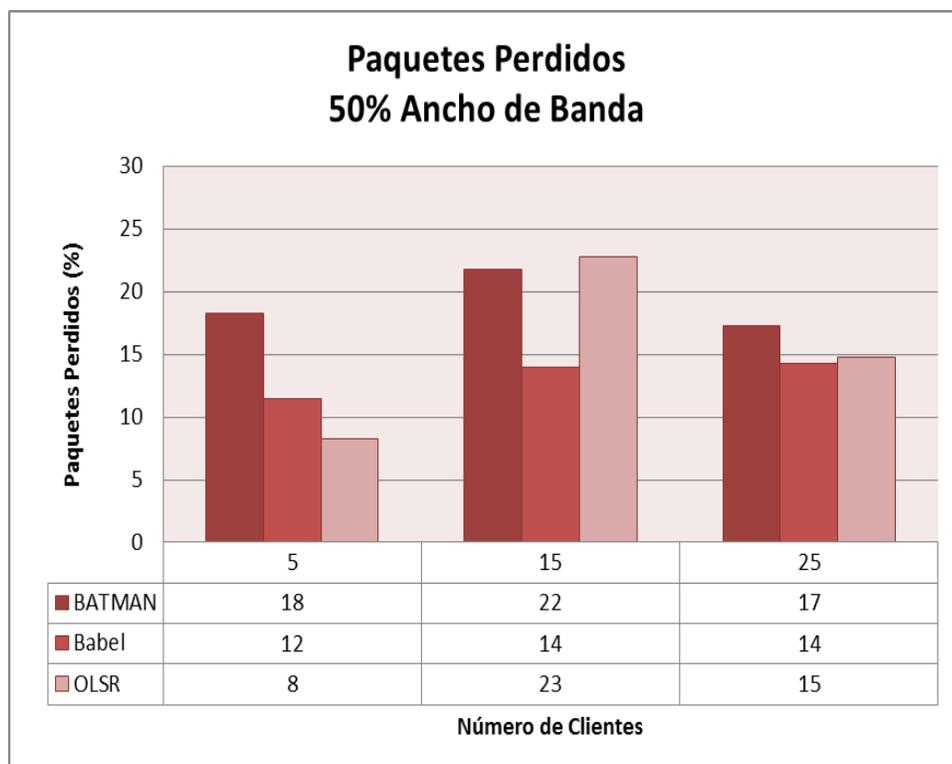


Figura 73. – Gráfica de paquetes perdidos con 50% de ancho de banda.

La figura anterior muestra que al restringir el ancho de banda en 50% en una red mallada con 5 clientes, se obtiene una menor pérdida de paquetes con el protocolo OLSR. Al aumentar el número de clientes a 15, el protocolo Babel es con el que se obtiene una menor pérdida de paquetes, comportamiento que se mantiene al aumentar el número de clientes a 25.

5.6. Resultados – Escenarios 3, 6 y 9

A continuación se presentan los parámetros de rendimiento de las pruebas realizadas sobre los escenarios 3, 6 y 9, es decir, en una red mallada que dispone del 100% del ancho de banda total, con los protocolos de enrutamiento BATMAN, Babel y OLSR y paquetes de 64, 256, 1024 y 1518 bytes, agrupados por la métrica de desempeño analizada.

5.6.1. Throughput

Desde la Figura 74 a la Figura 77 se muestra el *throughput* en función de la cantidad de clientes pertenecientes a la red mallada para tamaño de paquetes de 64, 256, 1024 y 1518 bytes en una red mallada que dispone del 100% del ancho de banda total.

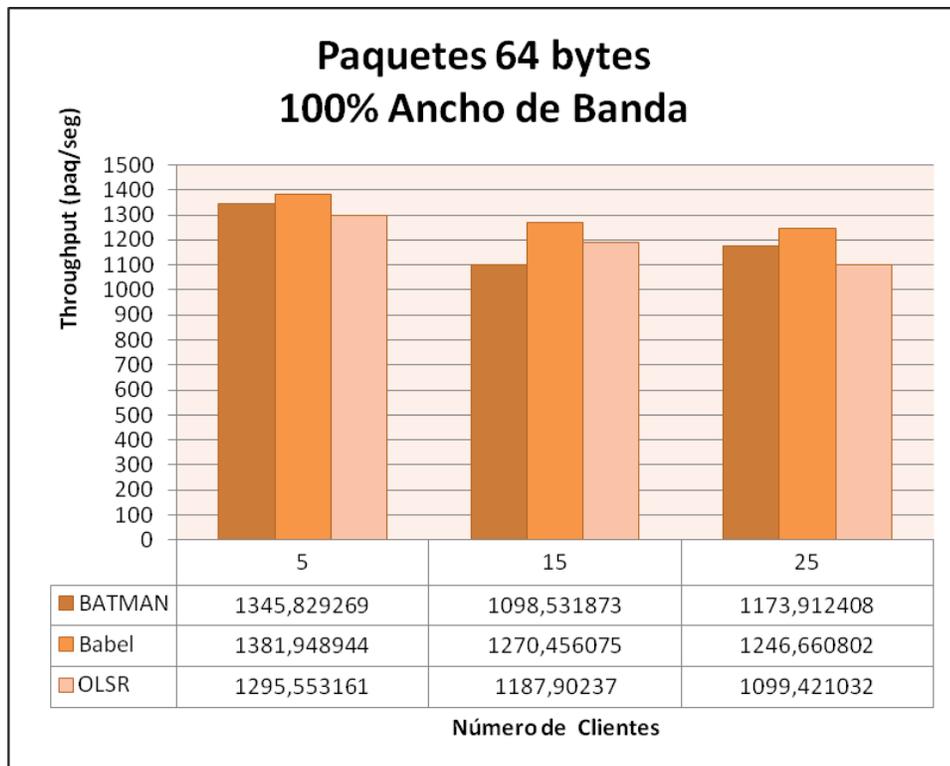


Figura 74. – Gráfica de *throughput* con paquetes de 64 bytes y 100% de ancho de banda.

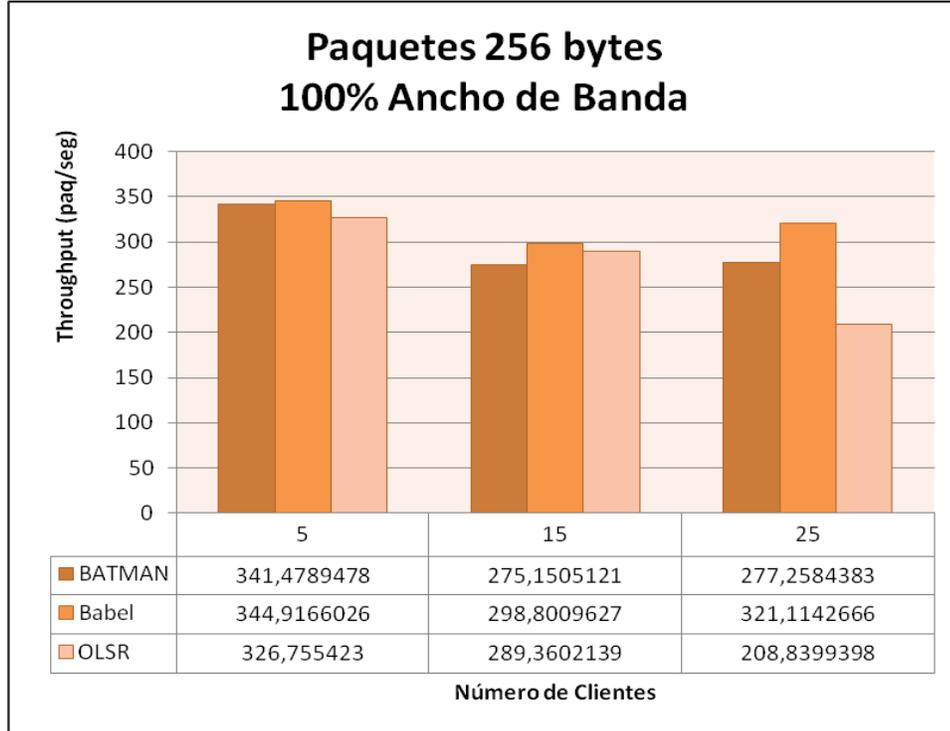


Figura 75. – Gráfica de *throughput* con paquetes de 256 bytes y 100% de ancho de banda.

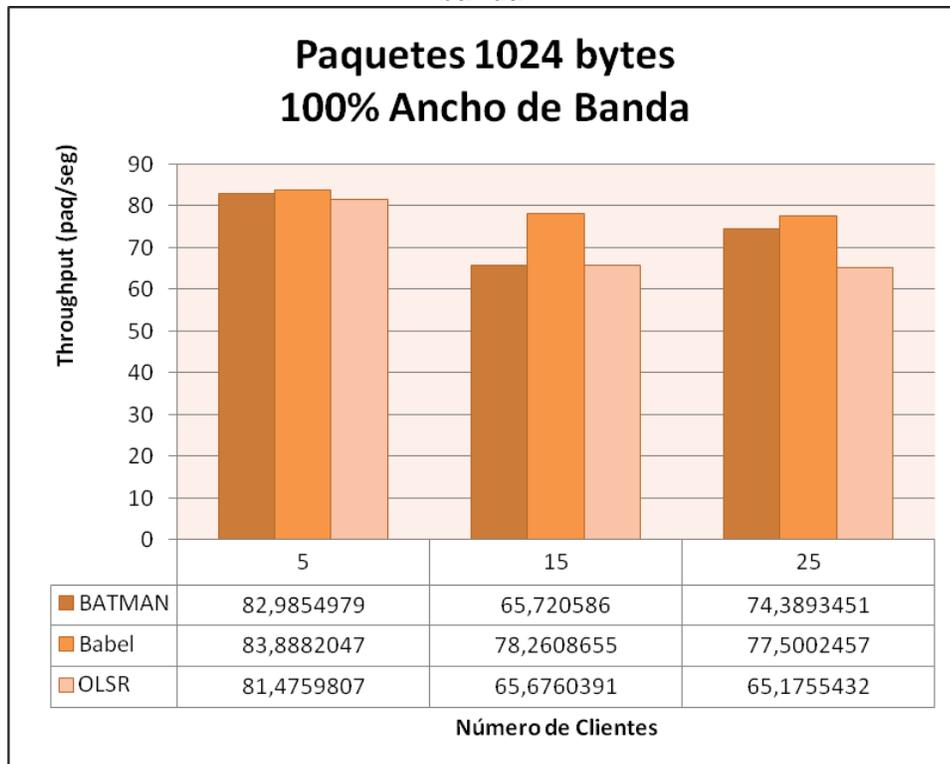


Figura 76. – Gráfica de *throughput* con paquetes de 1024 bytes y 100% de ancho de banda.

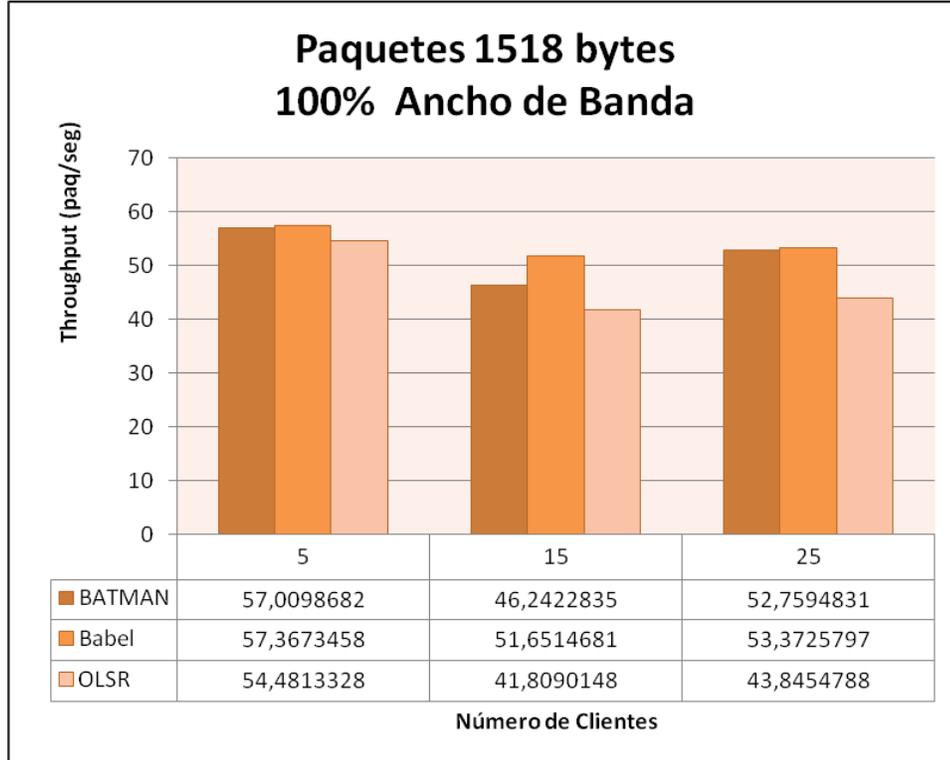


Figura 77. – Gráfica de *throughput* con paquetes de 1518 bytes y 100% de ancho de banda.

De los resultados mostrados anteriormente en las figuras se puede observar que para cada protocolo de enrutamiento utilizado, el *throughput* decrece a medida que aumenta el tamaño de los paquetes y en general no se ve muy afectado por el número de clientes en la red mallada.

Cuando no se aplica restricción en el ancho de banda, el protocolo Babel es con el que se obtiene una mayor tasa de paquetes por segundo, independientemente del número de clientes en la red mallada.

5.6.2. Jitter

Desde la Figura 78 a la Figura 81 se muestra el *jitter* en función de la cantidad de clientes pertenecientes a la red mallada para tamaño de paquetes de 64, 256, 1024 y 1518 en una red mallada que dispone del 100% del ancho de banda total.

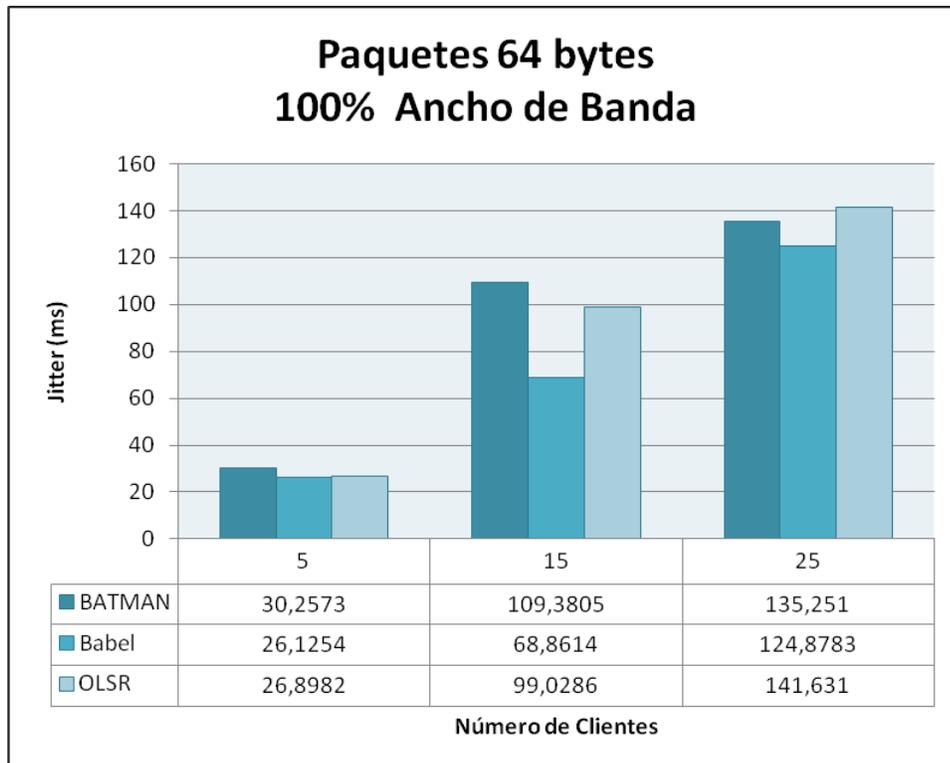


Figura 78. – Gráfica de *jitter* con paquetes de 64 bytes y 100% de ancho de banda.

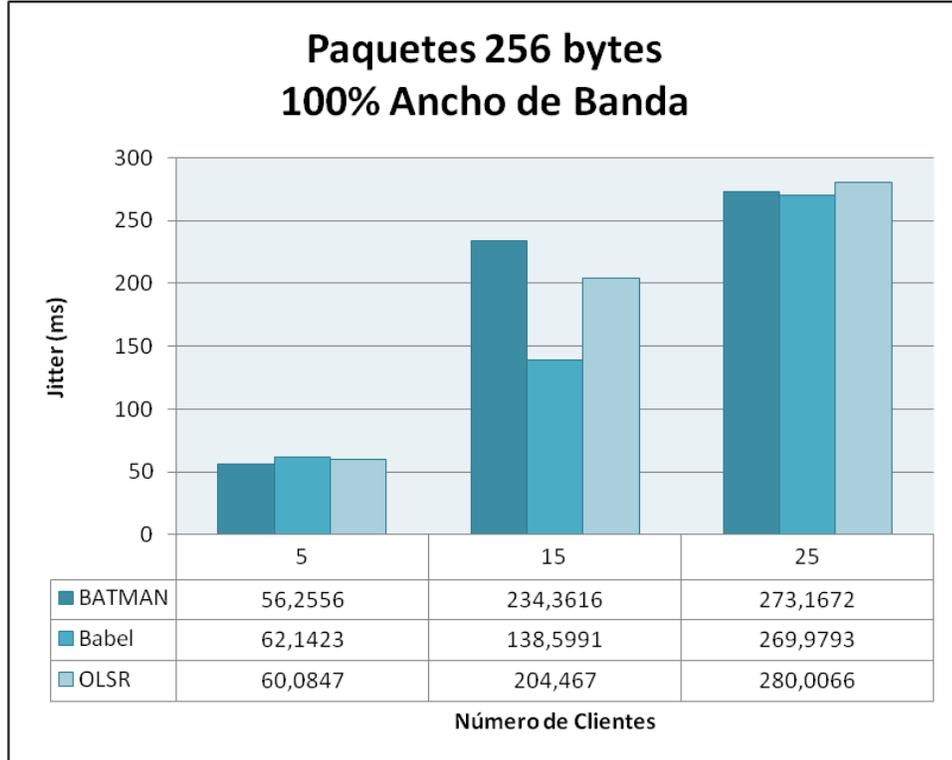


Figura 79. – Gráfica de *jitter* con paquetes de 256 bytes y 100% de ancho de banda.

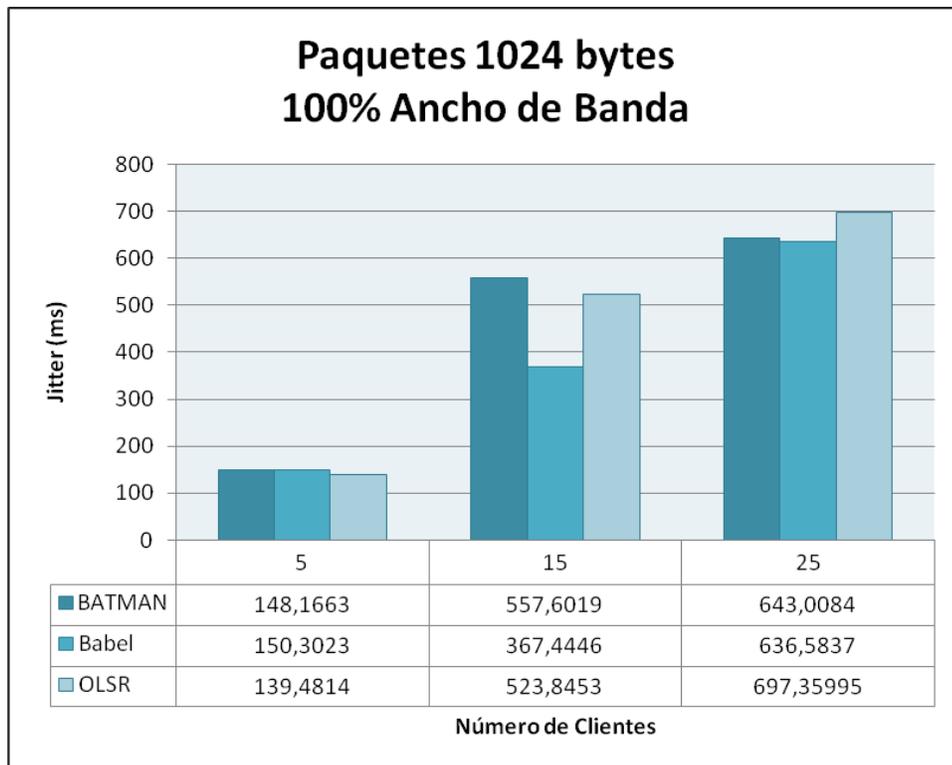


Figura 80. – Gráfica de *jitter* con paquetes de 1024 bytes y 100% de ancho de banda.

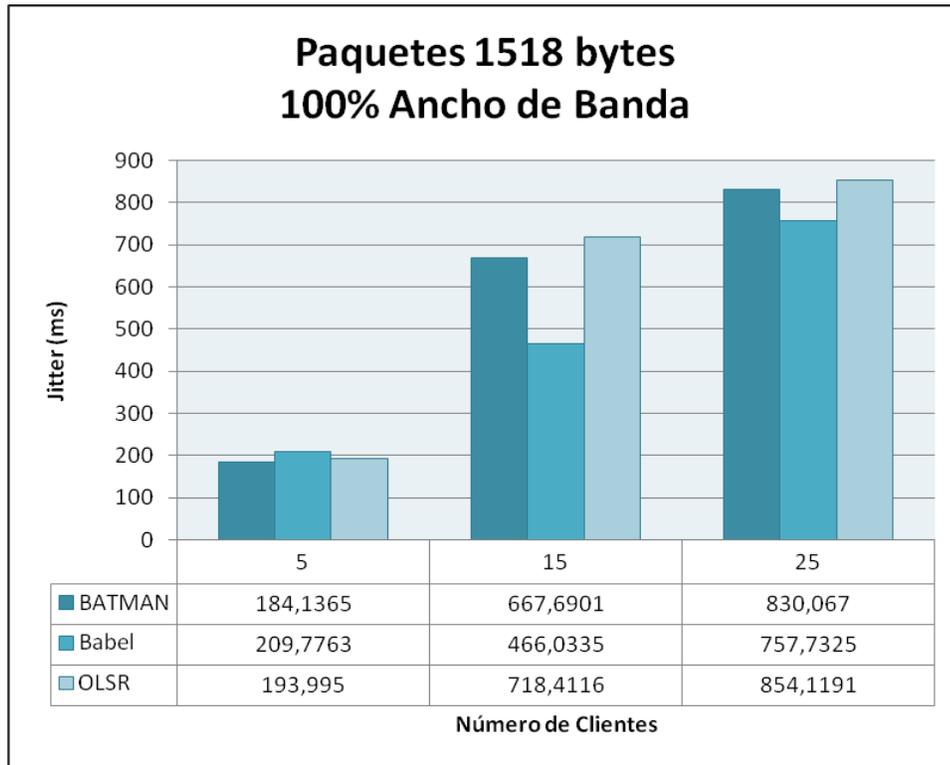


Figura 81. – Gráfica de *jitter* con paquetes de 1518 bytes y 100% de ancho de banda.

De los resultados mostrados anteriormente en las figuras, se puede observar que al aumentar el número de clientes de la red mallada, todos los protocolos aumentan los valores de *jitter* obtenidos, en mayor proporción cuando se trata de paquetes de mayor tamaño.

Se observa que cuando no se aplica restricción en el ancho de banda en una red mallada con 5 clientes se obtiene un *jitter* muy similar con todos los protocolos. Cuando se aumenta el número de clientes a 15 se obtiene un menor valor de *jitter* con el protocolo Babel, comportamiento que se mantiene al aumentar el número de clientes a 25.

5.6.3. Retardo

Desde la Figura 82 a la Figura 85 se muestra el retardo en función de la cantidad de clientes pertenecientes a la red mallada para tamaño de paquetes de 64, 256, 1024 y 1518 bytes en una red mallada que dispone del 100% del ancho de banda total.

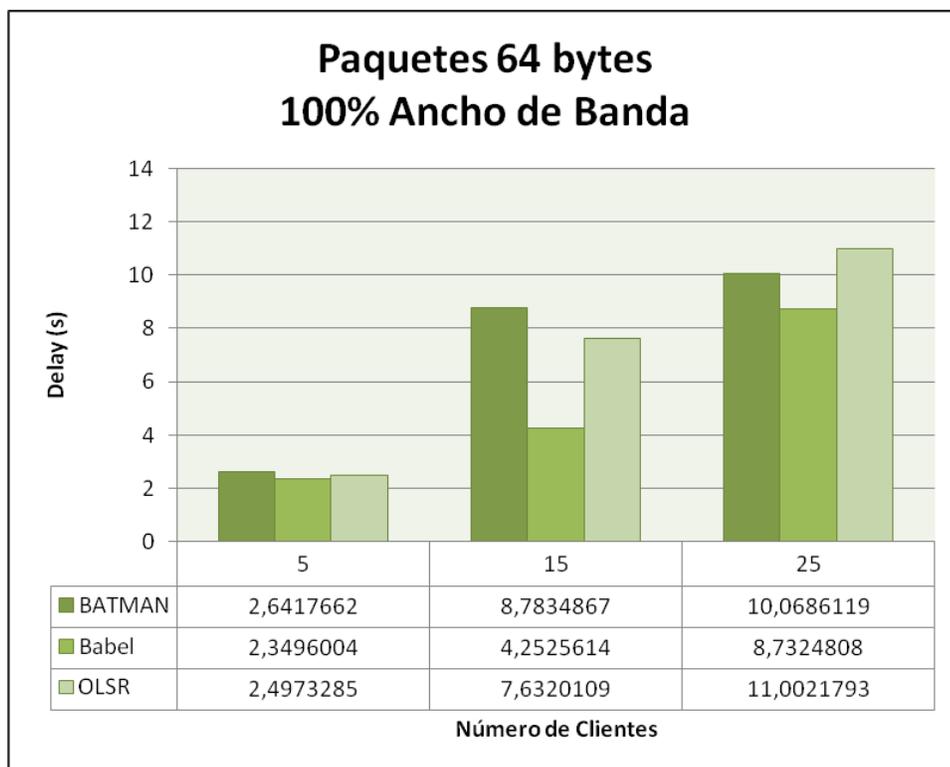


Figura 82. – Gráfica de retardo con paquetes de 64 bytes y 100% de ancho de banda.

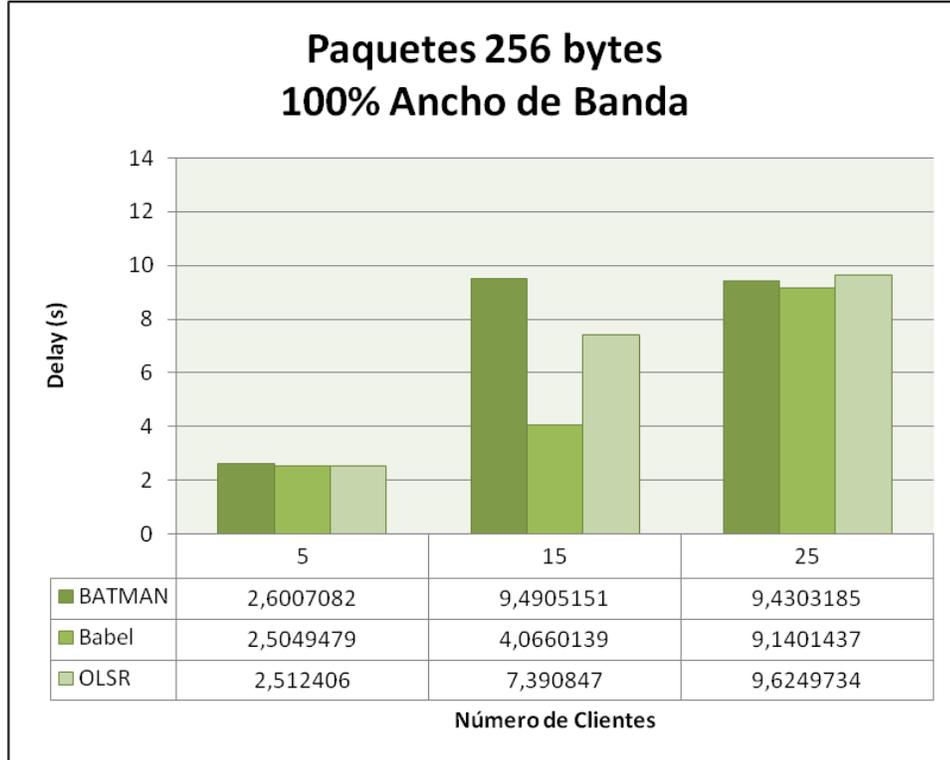


Figura 83. – Gráfica de retardo con paquetes de 256 bytes y 100% de ancho de banda.

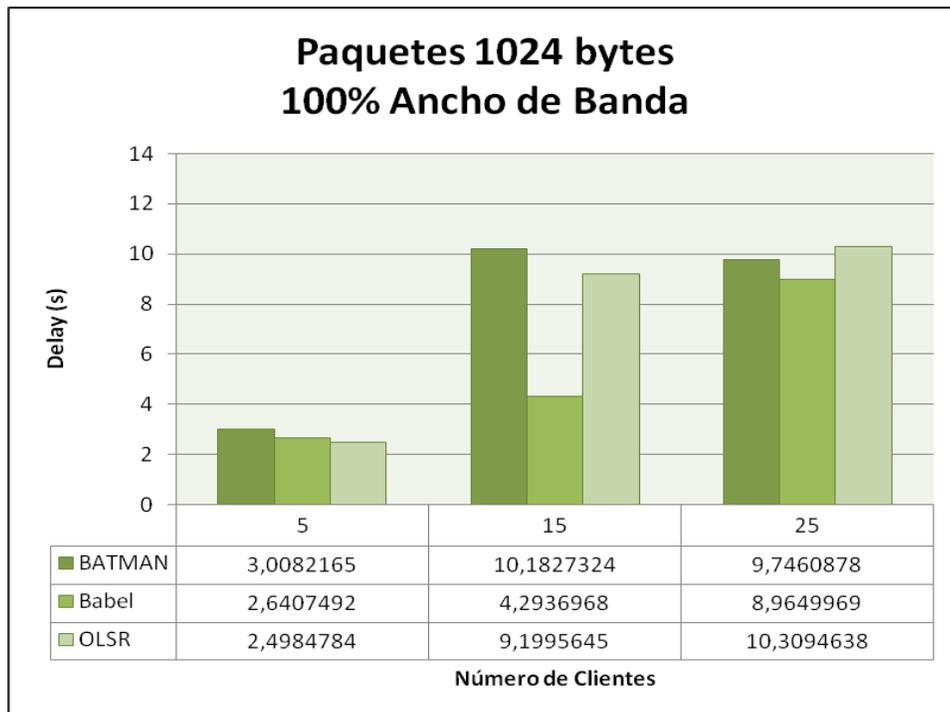


Figura 84. – Gráfica de retardo con paquetes de 1024 bytes y 100% de ancho de banda.

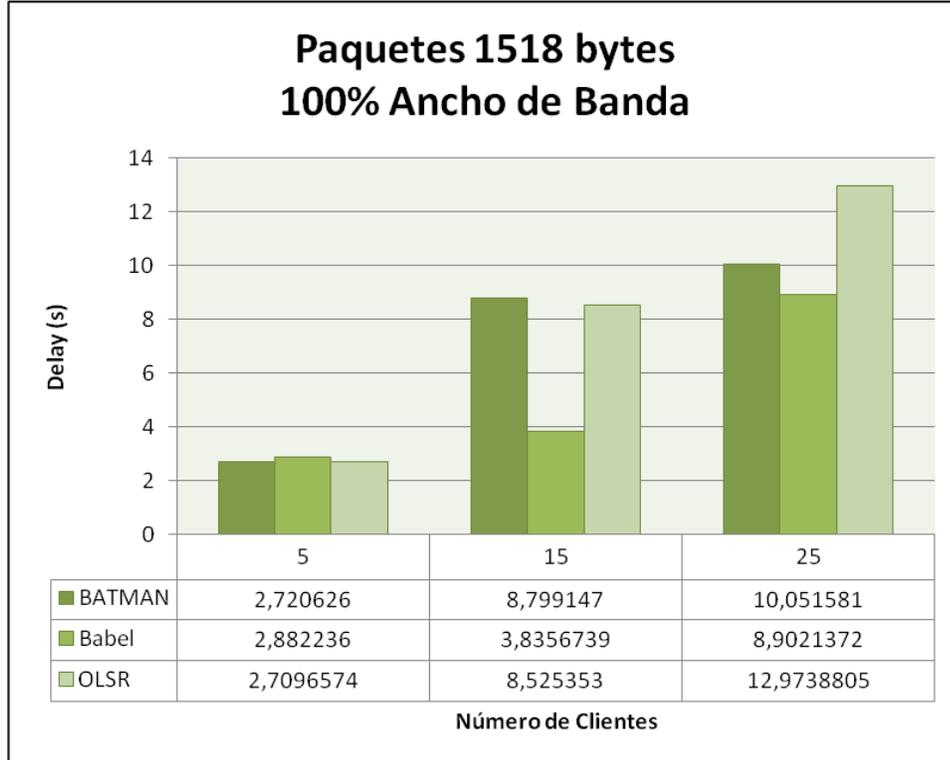


Figura 85. – Gráfica de retardo con paquetes de 1518 bytes y 100% de ancho de banda.

De los resultados mostrados anteriormente en las figuras, se puede observar que al aumentar el número de clientes de la red mallada, todos los protocolos aumentan los valores de retardo obtenidos.

Adicionalmente se puede observar que cuando no se aplica restricción en el ancho de banda en redes malladas de 5 nodos, con todos los protocolos se obtienen valores de retardo muy similares, levemente menores con los protocolos Babel y OLSR. Al aumentar el número de clientes a 15 se obtiene un menor valor de retardo con el protocolo Babel, comportamiento que se mantiene al aumentar el número de clientes a 25.

5.6.4. Paquetes Perdidos

En la Figura 86 se muestra el porcentaje de paquetes perdidos en función de la cantidad de clientes pertenecientes a una red mallada que dispone del 100% del ancho de banda total.

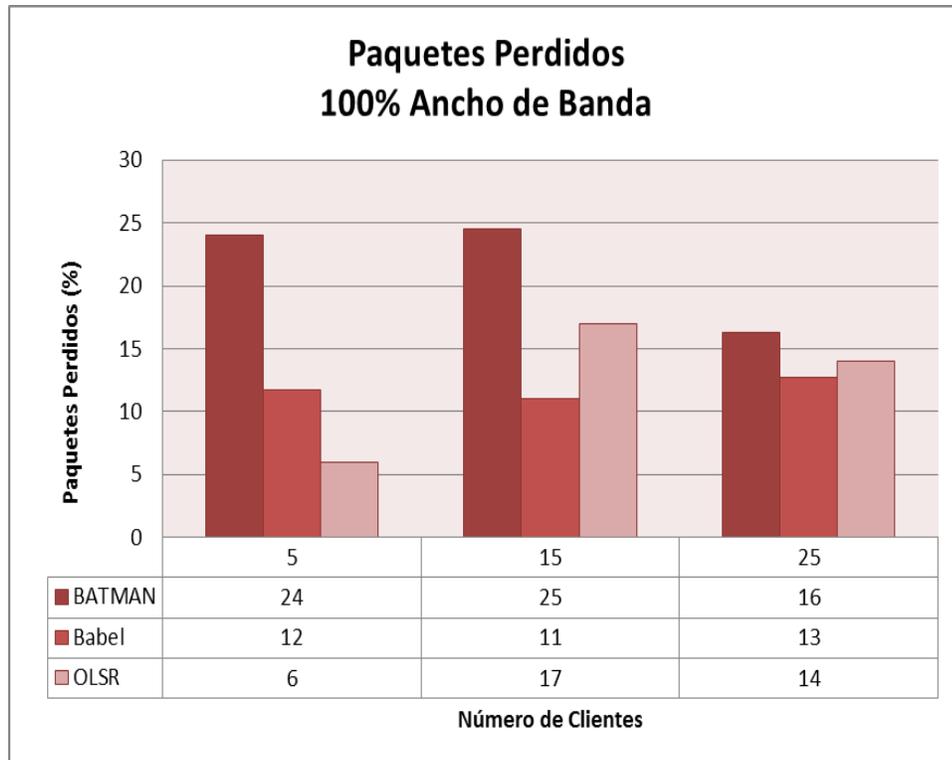


Figura 86. – Gráfica de paquetes perdidos con 100% de ancho de banda.

La figura anterior muestra que al no aplicar restricción en el ancho de banda de una red mallada con 5 clientes, se obtiene una menor pérdida de paquetes con el protocolo OLSR. Al aumentar el número de clientes a 15, el protocolo con el que se obtiene una menor pérdida de paquetes es Babel, comportamiento que se mantiene al aumentar el número de clientes a 25.

5.7. Resumen de los Resultados

La Tabla 4 muestra cual protocolo produce mejores resultados para cada una de las métricas y en cada uno de los escenarios planteados.

Número de Clientes	Ancho de banda	Jitter	Retardo	Throughput	Paquetes Perdidos
5	100%	Todos	Babel/OLSR	Babel	OLSR
	50%	OLSR	Babel/OLSR	Todos	OLSR
	25%	Todos	BATMAN	Todos	OLSR
15	100%	Babel	Babel	Babel	Babel
	50%	Babel/BATMAN	Todos	BATMAN/Babel	Babel
	25%	Babel	Babel/BATMAN	Todos	Babel
25	100%	Babel	Babel	Babel	Babel
	50%	Babel	Babel	Babel	Babel
	25%	Babel	Todos	Todos	Babel

Tabla 4. - Tabla de Resultados

La Tabla 4 muestra que cuando la red mallada posee pocos clientes se obtienen valores muy similares de *throughput*, *jitter* y retardo con todos los protocolos y se obtiene menor número de paquetes perdidos con el protocolo OLSR. También se observa claramente que el protocolo con el que se obtienen mejores resultados en la mayoría de las métricas de desempeño evaluadas al aumentar el número de clientes de la red mallada a 15 y a 25 clientes fue Babel.

En resumen se puede concluir que:

- La tendencia de los valores de retardo se ve claramente afectada al aplicar restricción en el ancho de banda y al aumentar el número de clientes de la red mallada.
- La tendencia de los valores de *jitter* se ve claramente afectada por el número de clientes, por el tamaño de los paquetes usados en la comunicación y al aplicar restricción en el ancho de banda.
- Cuando se aumenta el número de clientes de la red mallada de 5 a 15 los valores obtenidos de *jitter*, retardo y paquetes perdidos con los protocolos OLSR y BATMAN aumentan drásticamente. Se cree que esto sucede debido a la mayor cantidad de paquetes de control generados por dichos protocolos, en comparación con Babel. También se cree que por esta razón los protocolos no fueron capaces de mantener una conexión estable con el servidor al realizar las pruebas en una red mallada formada por 25 clientes. La solución encontrada fue modificar los valores que traen por defecto dichos protocolos y así poder obtener resultados con los 3 protocolos en redes malladas formadas por 25 clientes.

En general, se puede concluir que se obtiene un desempeño de la red muy similar con los 3 protocolos estudiados, siendo el protocolo babel el que ofrece un ligero mejor rendimiento, por lo que al implementar una solución como la planteada en este trabajo, se recomienda el uso de este protocolo.

6. Conclusiones

En este trabajo de investigación se estudió el rendimiento de la red al hacer uso de diversos protocolos de enrutamiento para redes Ad- Hoc cuando se restringe el ancho de banda provisto por la conexión a Internet en el borde de la red mallada.

Se investigaron implementaciones de protocolos de enrutamiento para redes malladas que funcionaran correctamente y contaran con la funcionalidad de interconectar la red con otras redes. Dichos protocolos fueron BATMAN, Babel y OLSR. Se identificó la situación que se quiso estudiar y se diseñó el *testbed* para estudiar los protocolos seleccionados.

Se plantearon una serie de escenarios sobre los cuales se realizaron un conjunto de pruebas que sirvieron para medir métricas de desempeño necesarias para comparar el comportamiento de los protocolos seleccionados.

Luego de finalizar las pruebas realizadas en todos los escenarios descritos en el capítulo 4, se compararon y analizaron los resultados y se pudo concluir que se obtienen resultados muy similares con todos los protocolos estudiados, siendo el protocolo de enrutamiento Babel el que muestra un ligero mejor funcionamiento en comparación a los demás, por lo que al implementar una solución como la planteada en este trabajo, es decir, al utilizar redes malladas como una manera de brindarle acceso a internet a personas que habitan en zonas desfavorecidas de Caracas, se recomienda el uso de este protocolo.

6.1. Contribuciones

Las siguientes fueron contribuciones hechas por este trabajo:

- Planteamiento de una solución para proporcionar acceso a Internet a personas que viven en zonas desfavorecidas de Caracas.
- Estudios de rendimiento de la red mallada usando diversos protocolos de enrutamiento como primer paso hacia la implementación de la solución propuesta.
- Se logró alcanzar todos los objetivos propuestos en esta investigación. Se utilizó un *testbed* donde se logró probar todos los protocolos seleccionados lo que permitió poder comparar los resultados obtenidos y con ello concluir acerca de cuál de ellos muestra un mejor comportamiento en los escenarios planteados.

6.2. Limitaciones

La única limitación que se presentó durante el desarrollo de esta investigación fue la siguiente:

- No se consiguieron implementaciones de protocolos de enrutamiento reactivos que funcionaran correctamente o que contaran con la funcionalidad de interconectar la red mallada con otras redes.

6.3. Recomendaciones

Entre las recomendaciones que se pueden hacer para futuras investigaciones se encuentran las siguientes:

- Desarrollo de una aplicación en el borde que maneje el ancho de banda de la conexión y lo adapte a ciertos parámetros, tales como hora, tráfico, etc.
- Estudio de los protocolos de enrutamiento incorporando modelos que consideren rompimiento de enlaces e interferencias en la red mallada.
- Entonación de los valores por defecto de los protocolos de enrutamiento.
- Revisión y modificación de las implementaciones de protocolos reactivos existentes para que soporten la funcionalidad de interconectar la red mallada con otras redes.

Referencias

- [1] C. Varela, L. Dominguez. Redes Inalámbricas. Universidad de Valladolid. 2002.
- [2] G. Aggelou. Mobile Ad-Hoc networks: from wireless LANs to 4G networks. Mc Grow Hill Mobile Ad-Hoc networks: from wireless LANs to 4G networks. 2005.
- [3] I. F. Akyildiz, X. Wang. A survey on Wireless Mesh Networks. IEEE Radio Communications. Septiembre 2005.
- [4] D. Acuña, R. Roncallo. Redes Inalámbricas Enmalladas Metropolitanas. Agosto 2007.
- [5] A. Daniilidis, K. Khalil. IEEE 802.11s Wireless Mesh Networks. Universidad Lund. 2007.
- [6] G. Held. Wireless mesh networks. CRC Press. 2005.
- [7] T. Clausen, P. Jacques. Optimized Link State Routing Protocol (OLSR). RFC 3626. Octubre 2003.
- [8] A. Neumann, C. Aichele, M. Lindner, S. Wunderlich. Better Approach To Mobile Ad-Hoc Networking (B.A.T.M.A.N.). Abril 2008.
- [9] J. Chroboczek. The Babel Routing Protocol draft-chroboczek-babel-routing-protocol-04. Internet-Draft. Julio 2010.
- [10] Implementación del protocolo Babel. <http://www.pps.jussieu.fr/~jch/software/babel/>.
- [11] Implementación del protocolo BATMAN. <http://www.open-mesh.org/>.
- [12] Implementación del protocolo OLSR. <http://www.olsr.org/>.
- [13] Distributed Internet Traffic Generator. <http://www.grid.unina.it/software/ITG/>.
- [14] Actualización Firmware DD-WRT. <http://www.dd-wrt.com/wiki/index.php/Installation>.
- [15] Manejador de Ancho de Banda Wondershaper. <http://lartc.org/wondershaper/>.
- [16] S. Bradner. Benchmarking Terminology for Network Interconnection Devices. RFC 1242. Julio 1991.
- [17] S. Bradner, J. McQuaid. Benchmarking Methodology for Network Interconnect Devices. RFC 2544. Marzo 1999.

Anexos

Anexo A

Pruebas realizadas con 5 clientes

#	Protocolo	Ancho Banda	# Clientes	Paquetes
1	BATMAN	100%	5	64
2	BATMAN	100%	5	256
3	BATMAN	100%	5	1024
4	BATMAN	100%	5	1518
5	Babel	100%	5	64
6	Babel	100%	5	256
7	Babel	100%	5	1024
8	Babel	100%	5	1518
9	OLSR	100%	5	64
10	OLSR	100%	5	257
11	OLSR	100%	5	1024
12	OLSR	100%	5	1518
13	BATMAN	50%	5	64
14	BATMAN	50%	5	256
15	BATMAN	50%	5	1024
16	BATMAN	50%	5	1518
17	Babel	50%	5	64
18	Babel	50%	5	256
19	Babel	50%	5	1024
20	Babel	50%	5	1518
21	OLSR	50%	5	64
22	OLSR	50%	5	256
23	OLSR	50%	5	1024
24	OLSR	50%	5	1518
25	BATMAN	25%	5	64
26	BATMAN	25%	5	256
27	BATMAN	25%	5	1024
28	BATMAN	25%	5	1518
29	Babel	25%	5	64
30	Babel	25%	5	256
31	Babel	25%	5	1024
32	Babel	25%	5	1518
33	OLSR	25%	5	64
34	OLSR	25%	5	257
35	OLSR	25%	5	1024
36	OLSR	25%	5	1518

Pruebas realizadas con 15 clientes

#	Protocolo	Ancho Banda	# Clientes	Paquetes
37	BATMAN	100%	15	64
38	BATMAN	100%	15	256
39	BATMAN	100%	15	1024
40	BATMAN	100%	15	1518
41	Babel	100%	15	64
42	Babel	100%	15	256
43	Babel	100%	15	1024
44	Babel	100%	15	1518
45	OLSR	100%	15	64
46	OLSR	100%	15	257
47	OLSR	100%	15	1024
48	OLSR	100%	15	1518
49	BATMAN	50%	15	64
50	BATMAN	50%	15	256
51	BATMAN	50%	15	1024
52	BATMAN	50%	15	1518
53	Babel	50%	15	64
54	Babel	50%	15	256
55	Babel	50%	15	1024
56	Babel	50%	15	1518
57	OLSR	50%	15	64
58	OLSR	50%	15	257
59	OLSR	50%	15	1024
60	OLSR	50%	15	1518
61	BATMAN	25%	15	64
62	BATMAN	25%	15	256
63	BATMAN	25%	15	1024
64	BATMAN	25%	15	1518
65	Babel	25%	15	64
66	Babel	25%	15	256
67	Babel	25%	15	1024
68	Babel	25%	15	1518
69	OLSR	25%	15	64
70	OLSR	25%	15	257
71	OLSR	25%	15	1024
72	OLSR	25%	15	1518

Pruebas realizadas con 25 clientes

#	Protocolo	Ancho Banda	# Clientes	Paquetes
73	BATMAN	100%	25	64
74	BATMAN	100%	25	256
75	BATMAN	100%	25	1024
76	BATMAN	100%	25	1518
77	Babel	100%	25	64
78	Babel	100%	25	256
79	Babel	100%	25	1024
80	Babel	100%	25	1518
81	OLSR	100%	25	64
82	OLSR	100%	25	257
83	OLSR	100%	25	1024
84	OLSR	100%	25	1518
85	BATMAN	50%	25	64
86	BATMAN	50%	25	256
87	BATMAN	50%	25	1024
88	BATMAN	50%	25	1518
89	Babel	50%	25	64
90	Babel	50%	25	256
91	Babel	50%	25	1024
92	Babel	50%	25	1518
93	OLSR	50%	25	64
94	OLSR	50%	25	257
95	OLSR	50%	25	1024
96	OLSR	50%	25	1518
97	BATMAN	25%	25	64
98	BATMAN	25%	25	256
99	BATMAN	25%	25	1024
100	BATMAN	25%	25	1518
101	Babel	25%	25	64
102	Babel	25%	25	256
103	Babel	25%	25	1024
104	Babel	25%	25	1518
105	OLSR	25%	25	64
106	OLSR	25%	25	257
107	OLSR	25%	25	1024
108	OLSR	25%	25	1518

Pruebas realizadas con la herramienta *ping*

#	Protocolo	Ancho Banda	# Clientes
109	BATMAN	100%	25
110	BATMAN	100%	15
111	BATMAN	100%	5
112	Babel	100%	25
113	Babel	100%	15
114	Babel	100%	5
115	OLSR	100%	25
116	OLSR	100%	15
117	OLSR	100%	5
118	BATMAN	50%	25
119	BATMAN	50%	15
120	BATMAN	50%	5
121	Babel	50%	25
122	Babel	50%	15
123	Babel	50%	5
124	OLSR	50%	25
125	OLSR	50%	15
126	OLSR	50%	5
127	BATMAN	25%	25
128	BATMAN	25%	15
129	BATMAN	25%	5
130	Babel	25%	25
131	Babel	25%	15
132	Babel	25%	5
133	OLSR	25%	25
134	OLSR	25%	15
135	OLSR	25%	5

Anexo B

Instalación del Software:

- **Babel:**

Se descargó el archivo **babeld-1.0.1.tar.gz** del sitio web. Una vez descargado el archivo, se descomprimió, se abrió un intérprete de comandos y se ejecutaron las siguientes líneas:

```
~$ cd babeld-1.0.1
~$ make
~$ sudo make install
```

- **BATMAN:**

Se descargó el archivo **batman-0.3.2.tar.gz** del sitio web. Una vez descargado el archivo, se descomprimió, se abrió un intérprete de comandos y se ejecutaron las siguientes líneas:

```
~$ cd batman-0.3.2
~$ make
~$ sudo make install
```

- **OLSR:**

Se descargó el archivo **olsrd-0.5.6-r7.tar.gz** del sitio web. Una vez descargado el archivo, se descomprimió, se abrió un intérprete de comandos y se ejecutaron las siguientes líneas:

```
~$ cd olsrd-0.5.6-r7
~$ make
~$ sudo make install
```

Adicionalmente para instalar el plugin que permite el soporte de gateway para este protocolo se ejecutaron las siguientes líneas:

```
~$ sudo aptitude install olsrd plugins
```

Luego debió abrirse el archivo de configuración `olsrd.conf` ubicado en el directorio `/etc` y modificar la interfaz de red, en la línea 32 del archivo, en nuestro caso se utilizó la interfaz `wlan0`. También debieron descomentarse las líneas del archivo que hacían referencia al plugin "`olsrd_dyn_gw.so.0.4`", en la línea 295.

- **D-ITG:**

Se descargó el archivo **D-ITG-2.7.0-Beta2** del sitio web. Una vez descargado el archivo, se descomprimió, se abrió un intérprete de comandos y se ejecutaron las siguientes líneas:

```
~$ cd D-ITG-2.7.0-Beta2/bin  
~$ make
```

- **Wondershaper**

Para su instalación se ejecutaron las siguientes líneas:

```
~$ sudo aptitude install wondershaper
```