

UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA  
*La Universidad Católica de Loja*



MODALIDAD ABIERTA Y A DISTANCIA  
ESCUELA DE CIENCIAS DE LA COMPUTACIÓN

"Simulación de QoS en una red Ad Hoc"

TESIS DE GRADO PREVIA A LA  
OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
INFORMÁTICA Y COMPUTACIÓN

**AUTORA:** Ligia Elena Maza Jiménez

**DIRECTOR:** Ing. Rommel Torres Tandazo

**CENTRO UNIVERSITARIO CARIAMANGA**

**2011**

Ing. Rommel Torres Tandazo

**DIRECTOR DEL PRESENTE TRABAJO DE TESIS**

**CERTIFICA:**

Que la tesis de grado "Simulación de QoS en una red Ad Hoc", desarrollada por la señorita Ligia Elena Maza Jiménez, ha sido orientada y revisada en sus aspectos de fondo y forma, por lo cual autorizo su presentación.

Loja, 15 de Octubre del 2011

-----  
Rommel Torres

# Declaración y Cesión de Derechos

“Yo, Ligia Elena Maza Jiménez, declaro ser autora del presente trabajo y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales.

Adicionalmente, declaro conocer y aceptar la disposición del Art. 67 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado que se realicen a través, o con el apoyo financiero, académico o institucional (operativo) de la Universidad”

-----  
Ligia Maza

# Auditoría

Las ideas, criterios, procedimientos y resultados vertidos en el presente trabajo, son de exclusiva responsabilidad de la autora.

-----  
Ligia Maza

# Dedicatoria

*Con mucho amor y cariño a mi hijo Gerson Jael, quien ha sido mi más grande inspiración y fortaleza, a mis padres, por todos sus sabios consejos, a mis hermanas, hermanos por su apoyo incondicional, en fin a toda mi familia quienes con su afán, sacrificio y comprensión contribuyeron para culminar una etapa más de mi formación; dedico para ellos cada pensamiento y esfuerzo de este trabajo.*

LIGIA MAZA

# Agradecimientos

*Antes que a todos quiero agradecer a Dios por darme las fuerzas necesarias en los momentos más difíciles cuando ya empezaba a perder las esperanzas y porque siempre está a mi lado.*

*A la Universidad Técnica Particular de Loja , en su Modalidad Abierta y a Distancia , Escuela de Ciencias de la Computación, y a todos quienes conforman la misma de manera especial al departamento de telecomunicaciones quienes de alguna manera me han orientado en todo el proceso de mi formación profesional. A la vez agradezco a las siguientes personas por la ayuda que me han brindado durante la realización de este Trabajo de Investigación de fin de carrera:*

Al Ing. Rommel Torres

*Director de tesis, le agradezco por compartir sus conocimientos y valores desinteresadamente, por su ayuda en cuanto a la dirección de tesis durante todo su desarrollo ya que así he podido llevar a cabo la realización de la misma.*

A mi familia

*Por haberme ayudado y apoyado tanto moral como económicamente durante el tiempo que he estado realizando este proyecto.*

A todos mi amigos

*Por su ayuda y apoyo moral, así como sus consejos y ánimos de seguir adelante.*

# Resumen

En los últimos años hemos sido testigos de la gran utilidad que se les da a las redes inalámbricas, las cuales han facilitado la interconexión de equipos proporcionando una movilidad y flexibilidad en el acceso a internet y otros fines, en la actualidad esta forma de conexión ya resulta bastante natural porque casi toda la sociedad dispone de un equipo inalámbrico para el acceso a internet desde cualquier lugar que se encuentren.

Todos los avances de telecomunicaciones han permitido el desarrollo de un gran número de servicios multimedia, los cuales se están desplegando debido a los recursos y tecnologías disponibles para su comunicación. De esta forma, cada día resulta más común el hacer uso de redes sociales, videoconferencia y estar inmersos en el mundo del internet, ya que si se le da un buen uso se logra buenos resultados como también beneficios. Por desventaja, estos avances tecnológicos no han venido implementados con algún mecanismo que nos asegure un adecuado acceso a estos servicios, es decir, un soporte de Calidad de Servicio (QoS).

En el caso de redes inalámbricas, específicamente las redes Manet (Mobile Ad Hoc Networks) más conocidas como redes Ad Hoc, caracterizadas por la comunicación directa de dispositivos móviles sin utilizar ningún medio físico, deben ofrecer soporte de calidad de servicio en todos sus procesos, es por eso que en la presente tesis, se propone hacer uso de un modelo de QoS, como es el de Servicios Diferenciados (Diff-Serv), mediante el uso de distintos escenarios, nodos y conexiones, a través de los cuales se comprobará si dicho modelo se ajusta específicamente a las características de estas redes, si ofrece un soporte de calidad de servicio eficiente y al mismo tiempo si permite una gestión óptima de recursos.

# Índice general

<b>1. Introducción y Objetivos</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Objetivo General . . . . .	3
1.2.1. Objetivos Específicos . . . . .	3
<b>2. Estado del Arte</b>	<b>4</b>
2.1. Redes Móviles Ad Hoc . . . . .	4
2.2. Calidad de Servicios (QoS) . . . . .	5
2.2.1. Servicios Integrados (Integrated Services - IntServ) . . . . .	5
2.2.2. Servicios Diferenciados (Differentiated Services - DiffServ) . . . . .	6
2.3. Calidad de Servicio en Redes Ad Hoc . . . . .	8
2.3.1. Servicios Integrados (IntServ) en una red Ad Hoc . . . . .	8
2.3.2. Servicios Diferenciados (DiffServ) en una red Ad Hoc . . . . .	9
2.4. Conclusiones . . . . .	9
<b>3. Problemática de QoS en Redes Ad Hoc</b>	<b>10</b>
3.1. La capacidad baja y altamente variable de los enlaces inalámbricos . . . . .	10
3.2. Topologías altamente dinámicas . . . . .	11
3.3. Capacidad limitada de recursos . . . . .	11
3.4. Seguridad reducida . . . . .	11

3.4.1. Autenticación . . . . .	12
3.4.2. Confidencialidad . . . . .	12
3.4.3. Disponibilidad . . . . .	12
3.4.4. Integridad . . . . .	12
3.5. Conclusiones . . . . .	13
<b>4. Metodología para simular QoS en una red Ad Hoc</b>	<b>14</b>
4.1. Fase de diseño y codificación . . . . .	14
4.1.1. Simuladores de red . . . . .	15
4.1.2. Arquitectura de Servicios Diferenciados (DiffServ) aplicada a re- des Ad Hoc . . . . .	19
4.2. Fase de análisis y comparación . . . . .	27
4.2.1. Indicadores de QoS para redes Ad Hoc . . . . .	27
4.3. Conclusiones . . . . .	28
<b>5. Interpretación y Discusión de Resultados</b>	<b>29</b>
5.1. Rendimiento . . . . .	30
5.2. Variación del retardo . . . . .	33
5.3. Retardo extremo a extremo . . . . .	34
5.4. Paquetes borrados . . . . .	35
5.5. Paquetes perdidos . . . . .	36
5.6. Comportamiento del router frontera y núcleo . . . . .	37
5.7. Conclusiones . . . . .	39
<b>6. Conclusiones</b>	<b>41</b>
6.1. Recomendaciones . . . . .	43
6.2. Trabajos futuros . . . . .	44

<b>Referencias</b>	<b>44</b>
<b>A. Script de simulación</b>	<b>48</b>
A.1. Resultados obtenidos con el TraceGrahp . . . . .	63

# Índice de figuras

4.1. Comparación de simuladores de red . . . . .	17
4.2. Estructura del escenario de una red Ad Hoc . . . . .	22
5.1. Rendimiento promedio de la simulación en el traceGraph . . .	31
5.2. Datos cuantitativos del rendimiento promedio de la simulación	31
5.3. Comparación del rendimiento promedio en la red . . . . .	32
5.4. Comparación de la variación del retardo promedio en la red .	33
5.5. Comparación del retardo promedio extremo a extremo en la red	34
5.6. Comparación de los paquetes borrados en la red . . . . .	35
5.7. Comparación de los paquetes perdidos en la red . . . . .	36
5.8. Paquetes borrados desde el nodo 15 al router frontera . . . . .	37
5.9. Datos del router núcleo, protocolo AODV y modelo diffServ .	38
5.10. Datos del router núcleo, protocolo DSDV y modelo diffServ .	38
5.11. Datos del router núcleo, protocolo DSR y modelo diffServ . . .	39
A.1. Información del comportamiento de la red con traceGrahp . .	63

# Capítulo 1

## Introducción y Objetivos

### 1.1. Introducción

En la actualidad, existe una gran necesidad de acceder rápidamente a la información que ofrece internet y comunicarse eficientemente con varios puntos de conexión haciendo uso de diferente tecnología que en nuestros medios existe, es por eso que el mundo de la comunicación ha ido evolucionando significativamente, y como muestra a ello se tiene las redes inalámbricas Ad Hoc, las mismas que actúan descentralizadamente sin la necesidad de utilizar ningún punto de acceso como intermediario.

El presente trabajo de investigación se enmarca principalmente en el contexto de Calidad de Servicio (QoS) de una red Ad Hoc, la misma que surgió como complemento a las redes cableadas y tienen la ventaja de accesibilidad y movilidad. El soporte de calidad de servicio desde el nivel de enlace no tiene ningún sentido si no existe una gestión adecuada desde el nivel de red, es por eso que se estudiará los modelos de QoS, servicios integrados (IntServ) y servicios diferenciados (DiffServ); pero nos centraremos precisamente en el modelo que nos ofrece mejores ventajas y beneficios como es el de Servicios Diferenciados, con el cual se realizará simulaciones con diferentes escenarios y conexiones para demostrar que tan eficiente es la comunicación que se provee a los usuarios.

Con la finalidad de facilitar la localización de información dentro de este documento, seguidamente se muestra como está estructurado:

En el Capítulo 1, se hace una introducción breve al tema propuesto, así como también se describen los objetivos general y específicos; en el Capítulo 2, se desarrollo del estado del arte relativo al tema que consta de: redes móviles Ad Hoc, calidad de servicio (QoS), calidad de servicio (QoS) en redes Ad Hoc, modelos de calidad de servicios IntServ y DiffServ; en el Capítulo 3, se presenta la problemática de QoS en las redes Ad Hoc; en el Capítulo 4, se detalla la metodología para simular QoS en una red Ad Hoc, así como también sus fases de diseño y codificación, y, análisis y comparación; en el Capítulo 5, se interpreta y compara los resultados basados en indicadores; y finalmente en el Capítulo 6 se concluye de acuerdo a lo investigado.

## 1.2. Objetivo General

Analizar la simulación de calidad de servicios (QoS) en una red Ad Hoc, para conocer el comportamiento la red al momento de realizar una comunicación y así poder establecer un criterio final.

### 1.2.1. Objetivos Específicos

- Comprobar si el modelo de QoS DiffServ ofrece calidad de servicio en la comunicación de la red
- Trabajar con la herramienta de simulación tecnológica más eficiente con la finalidad de que nos arroje los mejores resultados
- Representar gráficamente los resultados obtenidos para un mejor entendimiento del usuario
- Medir la Calidad de Servicio (QoS) de acuerdo a los indicadores necesarios para este tipo de redes

# Capítulo 2

## Estado del Arte

### 2.1. Redes Móviles Ad Hoc

Con la evolución de la tecnología y debido a la necesidad de comunicación entre diferentes dispositivos móviles, surgieron las redes Ad Hoc [1] [2], las cuales son un conjunto de dispositivos que ofrecen servicios de comunicación de extremo a extremo y están compuestas solo por dispositivos terminales, es por eso que ofrecen grandes beneficios y nos permiten intercambiar información sin necesidad de utilizar cualquier infraestructura de red fija ya existente.

Entre las características más importantes de una red Ad Hoc tenemos [3]: movilidad, ya que el usuario está en constante movimiento, y pese a esto no debe perder la comunicación; redes relativamente dispersas, en donde la implementación es justificada para nodos lejanos, además se caracterizan por ser una red multi-salto dinámica y flexible, con una topología que varía aleatoriamente, no requiere infraestructura fija, su red es fácilmente reconfigurable y de alta adaptabilidad.

## 2.2. Calidad de Servicios (QoS)

El tema de redes ha venido evolucionando a pasos agigantados, de aquí que se desprende la necesidad de Calidad de Servicio (QoS) [4], *"La QoS puede ser descrita como un conjunto de parámetros que describe la calidad de una corriente específica de los datos"* [5]. Diversas aplicaciones y medios de comunicación comparten esta necesidad ya que en la actualidad es indispensable disponer de calidad de servicio porque garantiza un adecuado rendimiento, seguridad, beneficios y resultados eficientes para el usuario.

A continuación se analizarán a nivel de capa de red los modelos para implementar QoS en una red Ad Hoc:

### 2.2.1. Servicios Integrados (Integrated Services - IntServ)

El objetivo principal de IntServ es la reserva de recursos en la red por flujo de tráfico [6], indicando las características que resultarán necesarias como (ancho de banda, retardo, jitter, etc.). Para la provisión del servicio cada router del núcleo de la red debe mantener una tabla con el estado de reserva por flujo, un "flujo" es una secuencia de paquetes IP desde un único transmisor destinado a un único receptor que pertenece de un único usuario y requiere la mismo QoS. Para realizar la configuración de la red IntServ se usa el protocolo de señalización RSVP (Protocolo de reserva de recursos) [7], dicho protocolo es un mecanismo de señalización para llevar a los parámetros de calidad de servicio desde el emisor hasta el receptor y poder realizar reservas de recursos a lo largo del camino.

IntServ define tres clases de servicios:

**Servicios garantizados.-** Son los que brindan condiciones seguras y garantizadas en una comunicación entre extremos. Cada router del trayecto debe ofrecer las garantías solicitadas, aunque a veces esto no es posible por las características del medio físico en que se encuentre.

**Servicio de carga controlada.-** Es el que provee la misma QoS que un flujo recibiría si la red estuviera aliviada, pero asegurando que el servicio se conservaría aún cuando la red estuviera sobrecargada, es decir un buen tiempo de respuesta, pero sin garantías estrictas ya que se pueden producir retardos grandes.

**Servicio de mejor esfuerzo.-** Este servicio no garantiza ningún servicio debido a la falta de recursos disponibles en la red.

**Ventajas de IntServ:** *"Simplicidad, que facilita que toda la red mantenga una política de administración integrada; La posibilidad de crear reglas de QoS para flujos discretos, lo que posibilita la generación de llamadas de voz, y la Capacidad de Control de Admisión de Llamadas (CAC), lo que permite conocer a los nodos extremos sobre la disponibilidad de ancho de banda"*

**Desventajas de IntServ:** *"Todos los elementos deben mantener el estado e intercambiar mensajes de señalización por cada flujo que manejen; Se necesitan mensajes periódicos de actualización para mantener las sesiones, lo que aumenta el tráfico en la red siendo susceptible a pérdidas de paquetes, por lo tanto todos los nodos intermedios deben tener RSVP en sus funciones" <sup>1</sup>.*

### 2.2.2. Servicios Diferenciados (Differentiated Services - Diff-Serv)

El modelo DiffServ [9], sigue la filosofía de la asignación de los flujos múltiples en los niveles de servicio, solucionando de esta forma los problemas de escalabilidad de Intserv, es orientado a la no reservación, no establece un canal virtual sobre la red y por lo tanto no realiza ninguna reserva de servicios, cada nodo adopta diferentes comportamientos de acuerdo a sus características disponibles y de acuerdo a su fabricante. Se usan técnicas de marcado de paquetes, como precedencia IP y su sucesor Differentiated Service Code Points (DSCP), que hace referencia al segundo byte en la cabecera de los paquetes IP que se utiliza para diferenciar la calidad en la comunicación que requieran

---

<sup>1</sup>Gustavo Mercado, Hector Raimondo y Javier Días. *Calidad de Servicio en Redes IP.pdf [8]*

los datos que se transportan y por ende el comportamiento que por defecto se coloca en el Per Hop Behavior (PHB) cuando el tráfico no cumple con los requisitos de cualquiera de las otras clases definidas. De esta manera se soluciona el problema del overhead en el mantenimiento de la información de un flujo particular, evitando la señalización fuera de banda.

DiffServ define tres clases de servicios[10]:

**Servicio de Reenvío Acelerado.-** Este servicio es el de mayor calidad ya que ofrece un servicio equivalente a una línea dedicada virtual y debe garantizar un caudal y tasa mínima de pérdida de paquetes y un retardo medio.

**Servicio de Reenvío Asegurado.-** Asegura un trato preferente, pero no garantiza caudales, retardos, etc. Se definen cuatro clases posibles pudiéndose asignar a cada clase una cantidad de recursos en los routers (ancho de banda, espacio en buffers, etc.).

**Servicio de Mejor Esfuerzo.-** Este servicio se caracteriza por tener a cero los tres primeros bits del DSCP, en este caso los dos bits restantes pueden utilizarse para marcar una prioridad, este servicio no ofrece ningún tipo de garantías.

**Ventajas de Diffserv:** *"Los flujos múltiples se asignan a un nivel de servicio único por lo que las garantías de QoS cualitativa se proporcionan a aplicaciones con diferentes niveles de servicio"*

**Desventajas de Diffserv:** *"La principal desventaja es que los servicios no están estrictamente garantizados, ya que al no haber reserva de ancho de banda entre extremos, cualquier nodo mal configurado que no soporte el PHB propuesto puede descartar flujos de clases establecidas" <sup>2</sup>.*

---

<sup>2</sup>Gustavo Mercado, Hector Raimondo y Javier Días. *Calidad de Servicio en Redes IP.pdf*

## **2.3. Calidad de Servicio en Redes Ad Hoc**

La QoS, desde el punto de vista inalámbrico específicamente en una red Ad Hoc [11], es muy importante ya que se debe tener en cuenta que los dispositivos móviles y factores como movilidad, tiempo, latencia, entre otros, deben funcionar correctamente, ofrecer un rendimiento eficiente extremo a extremo y garantizar la QoS máxima en la comunicación y en aquellas aplicaciones que lo demanden y para ello es necesario explorar e utilizar apropiadamente cada uno de los recursos disponibles tanto en el medio inalámbrico como en los nodos móviles en la red, así como también la estabilidad de tales recursos de red disponibles.

A continuación se analizará los modelos de QoS para redes Ad Hoc:

### **2.3.1. Servicios Integrados (IntServ) en una red Ad Hoc**

Los nodos en función de una red Ad Hoc tienen que implementar funcionalidades de los servicios integrados [9], esto se traduce en sobrecarga de procesamiento grande para los nodos móviles que no es deseable debido a sus limitados recursos relativos, a la capacidad de memoria, procesamiento y tiempo. Las limitaciones de ancho de banda son siempre bien escasas en los sistemas de comunicación inalámbrica, por lo tanto, la señal inalámbrica debe reducirse al mínimo siempre que sea posible ya que hay probabilidad de pérdida de paquetes porque las tasas de error de los sistemas inalámbricos son grandes y el número de los flujos transportados por cada enlace será limitada. Por estas razones la calidad de servicio en redes ad hoc, no es adecuada para redes móviles ad hoc extensas porque la cantidad de información de estado se incrementa proporcionalmente con el número de flujos y la topología cambia con frecuencia, resultando en más sobrecarga de señalización y almacenamiento fuera del alcance, dando así problemas de escalabilidad.

### **2.3.2. Servicios Diferenciados (DiffServ) en una red Ad Hoc**

DiffServ [12], está basado en un modelo ligero debido a la ventaja de priorización de clases de tráfico el mismo que facilita los requisitos de los modelos de QoS y evita el problema de la escalabilidad mediante un número de comportamientos por salto (PHB) en los routers borde de red y por asociar otro (DSCP) en la cabecera IP del paquete. Por lo tanto, DiffServ es escalable, pero no garantiza los servicios de extremo a extremo totalmente porque dificulta la implementación de DiffServ en internet.

Después de haber analizado los dos modelos de QoS, se concluye que el principal problema de IntServ es la necesidad de mantener información sobre cada flujo de tráfico en todos los routers de la red, lo cual lleva a problemas de escalabilidad, en cambio el modelo DiffServ, se basa en la división del tráfico en diferentes clases y en la asignación de prioridades, también utiliza diferente información de la cabecera de los paquetes para distinguir, clasificar y conocer el tratamiento que debe recibir el tráfico en los nodos. Por lo que se deduce que el modelo DiffServ se ajusta adecuadamente a las condiciones de provisión de calidad de servicio en las redes Ad Hoc y por ende éste será el modelo con el cual se trabajará.

## **2.4. Conclusiones**

En el presente capítulo se ha revisado una serie de conceptos necesarios para tener una visión clara de la relación y funciones que tienen dentro de las redes Ad Hoc, entre los cuales se analizó: calidad de servicio, servicios integrados (IntServ) y servicios diferenciados (DiffServ), específicamente el modelo de servicios diferenciados que es con el que se trabajará en ésta investigación.

# Capítulo 3

## Problemática de QoS en Redes Ad Hoc

Cabe destacar que la provisión de calidad de servicio en redes Ad Hoc resulta ser un problema complejo debido a las propiedades intrínsecas de este tipo de redes [13], por lo cual a continuación se detallará los problemas que entorpecen la comunicación en una red:

### 3.1. La capacidad baja y altamente variable de los enlaces inalámbricos

La capacidad baja en los enlaces inalámbricos [14], es uno de los problemas que mayormente se da en este tipo de redes, ya que al realizar una simulación de QoS en redes Ad Hoc se requiere gran capacidad de procesamiento y memoria, la variabilidad en los enlaces inalámbricos está limitada por la interferencia de las transmisiones simultáneas entre los nodos y porque los mismos se encuentran en constante movimiento, así como también por la gran magnitud de interferencias electromagnéticas que existe en el medio como los altos niveles de ruido, atenuación e interferencia que son superiores al de las infraestructuras fijas, es por eso que los nodos de una red Ad Hoc tienen un ancho de banda limitado y son propensos a errores de transmisión, dichos factores hacen que las rutas se modifiquen de forma repentina y con ello el retardo y validez de datos se presenten en los enlaces, lo que afecta directamente a la QoS. Problema que se medirá en la simulación mediante el indicador de la variación del retardo de los paquetes.

## 3.2. Topologías altamente dinámicas

Debido a la libertad de movimiento que tienen cada uno de los nodos en una red Ad Hoc, pudiendo asociarse y desasociarse de la red de forma espontánea es que la topología puede cambiar de forma rápida y arbitraria a lo largo del tiempo, provocando que los enlaces se rompan frecuentemente; por otra parte hay que tener en cuenta que el ancho de banda disponible de una red Ad Hoc es inferior al de un medio cableado. La naturaleza dinámica de éstas redes provoca que los protocolos de enrutamiento utilizados en redes cableadas no sean válidos ni eficientes y sea necesaria la utilización de nuevos protocolos más adecuados a las características particulares de estas redes. El retardo extremo a extremo es el indicador por medio del cual se puede comprobar mediante la simulación el grado en el que afectan o no las topologías dinámicas en las redes Ad Hoc.

## 3.3. Capacidad limitada de recursos

Con una capacidad limitada de recursos [15], la redes Ad Hoc tendrán un mal rendimiento porque algunos de los equipos que envían/reciben información carecen o son pobres en ciertas características como la fuente de energía (batería o pila), su memoria es limitada, la capacidad de procesamiento es bajo y todo esto debido a la exigencia de los nodos al enviar paquetes a los demás, es por eso que se dificulta la eficiencia entre el envío y la recepción de paquetes. En este caso no se ha medido el problema con ningún indicador, debido a que no se trabajará directamente con dispositivos físicos sino que se hará una simulación virtual para saber el comportamiento de la red ad hoc.

## 3.4. Seguridad reducida

La seguridad es quizá uno de los requisitos más importantes que debe proporcionar una red Ad Hoc para ofrecer QoS en todos sus procesos, pero es complicado ya que las comunicaciones inalámbricas son transmitidas utilizando ondas de radio compartidas, por cuanto estas redes son vulnerables a las amenazas y ataques debido a que sus nodos son independientes, inalámbricos y móviles impidiendo proveer una total seguridad, así como también el número de integrantes en una red Ad Hoc puede variar

dinámicamente, al punto de saturarla o de tener algún nodo intruso. El nivel de seguridad dependerá del escenario en las que se vayan a implementar las mismas es por eso que se debe garantizar que haya lo siguiente:

### **3.4.1. Autenticación**

Existe autenticación cuando se habilita a un nodo para asegurar que la identidad del otro nodo en el otro extremo es el correcto, caso contrario un adversario puede simular ser otro nodo y tener acceso a los recursos e información importante e interferir en las operaciones de los otros nodos.

### **3.4.2. Confidencialidad**

Se refiere a asegurar que cierta información nunca sea descubierta por algún usuario u entidad no autorizada.

### **3.4.3. Disponibilidad**

Asegurar siempre la disponibilidad de los servicios de la red aún cuando ocurran ataques u interferencia, porque se pueden dar en cualquier nivel de la red.

### **3.4.4. Integridad**

Con la integridad se garantiza que un mensaje que es transmitido por la red no sea corrompido, ya que este puede ser corrompido por ataques a la información.

En lo referente a seguridad en una red Ad Hoc, no se toma en cuenta ningún indicador ya que el tema de tesis no se trata específicamente de seguridad, pero se lo ha tomado en cuenta porque es un aspecto de gran importancia ya que este tipo de redes son más susceptibles a ataques de seguridad que cualquier otro tipo de redes.

### 3.5. Conclusiones

En éste capítulo se analizó los problemas que con mas frecuencia se presentar en las redes Ad Hoc, debido a su comunicación inalámbrica y nodos móviles que se utilizan para llevar a cabo el envío y recepción de datos como la capacidad baja y variable de los enlaces inalámbricos que produce interferencia en los enlaces, la topología dinámica que se dá por el movimiento constante de los nodos, la capacidad limitada de recursos y seguridad reducida, pues estas deficiencias dificultan controlar sus procesos.

# Capítulo 4

## Metología para simular QoS en una red Ad Hoc

En el presente capítulo se describirá la metodología que utilizaremos para llevar a cabo la simulación de QoS en una red Ad Hoc, valiéndonos del estudio de fases y procesos para alcanzar los objetivos propuestos. La metodología se da desde un estudio descriptivo hacia un estudio específico, lo cual ayudará a la toma de decisiones a la hora de implementar algún sistema y de esta manera no cometer errores en el sistema real.

A continuación se presenta las siguientes fases a seguir para el desarrollo del trabajo de investigación:

- **Fase de diseño y codificación**
- **Fase de análisis y comparación**

### 4.1. Fase de diseño y codificación

Esta fase trata sobre los elementos y técnicas que se utilizarán para llevar a cabo el diseño de los escenarios y codificación de scripts de la red Ad Hoc, tomando en cuenta la naturaleza de sus problemas para posteriormente ser simulada. Por estas razones se hará el proceso de selección de simuladores de red que sea más apropiado y

que garantice la eficacia de los resultados, ya que será necesaria para llevar a cabo esta fase.

#### **4.1.1. Simuladores de red**

##### **\* OPNET**

Opnet [16], fue introducido en el mercado como el primer simulador comercial, tiene una estructura basada en bibliotecas e incorpora librerías para facilitar el modelado y simulación de una red. El desarrollo de los modelos se realiza mediante la conexión de diferentes tipos de nodos, incorporando un lenguaje de simulación propietario, orientado a las telecomunicaciones, que permite el análisis de diversos aspectos. El paquete de software incluye una completa interfaz gráfica, un editor, y varias herramientas de análisis.

##### **\* OMNET++**

Omnet++[17], es una herramienta enfocada al área académica y desarrollada para modelar y simular eventos discretos en redes de comunicaciones; básicamente este simulador de redes recrea dichos eventos discretos por medio de módulos orientados a objetos, puede ser utilizado para modelar el tráfico de información sobre las redes, los protocolos de red, las redes de colas, multiprocesadores y otros sistemas de hardware distribuido; además para validar arquitecturas de hardware y evaluar el rendimiento de sistemas complejos.

##### **\* NS-2**

NS2 [18], es un simulador con un amplio rango de uso para redes de eventos discretos orientado principalmente para la investigación y uso educativo; Emplea dos lenguajes de programación, C++ y OTcl (Object Tool Command Language), ambos orientados a objetos y soporta protocolos de las capas de aplicación y transporte, los cuales pueden ser implementados tanto en redes cableadas como inalámbricas locales o vía satélite; son aplicables a grandes redes con topologías complejas y con un gran

número de generadores de tráfico.

En la Figura. 3.1, se muestra las características más relevantes de los simuladores antes mencionados [19], realizando una comparación para saber cuál es el más eficiente y que mejor se adapta al modelo de simulación de QoS en una red Ad Hoc.

COMPARACION DE HERRAMIENTAS DE SIMULACION DE REDES		
OPNET	OMNET ++	NS-2
<p>• El programa incluye librerías para acceder a un extenso grupo de aplicaciones y protocolos.</p> <p>• Los APIs de simulación permiten acceder libremente al código fuente, lo cual facilita la programación de nuevos protocolos de red.</p> <p>• Permite mostrar el tráfico por la red a partir de una animación, durante y después de la simulación. Los resultados se exhiben mediante gráficos estadísticos.</p>	<p>• Mediante la programación por módulos, es posible simular procesos paralelos y distribuidos los cuales pueden utilizar varios mecanismos para comunicarse entre sí.</p> <p>• Es multiplataforma, disponible para Linux y Windows.</p> <p>• Fácilmente adaptable (programando en C++)</p>	<p>• Está disponible gratuitamente para los ámbitos de investigación y educación.</p> <p>• Soporta una gran cantidad de protocolos de las capas de aplicación y de transporte.</p> <p>• Por ser una de las más antiguas herramientas de simulación, se ha convertido en un estándar de su área, esto ha llevado a sea ampliamente utilizado y que se encuentre una gran variedad de información.</p> <p>• Este programa contiene módulos que cubran un extenso grupo de aplicaciones, protocolos de ruteo, transporte y diferentes tipos de enlace.</p>
<p>• Es necesario obtener la licencia para poder utilizar el software, ya que no existen versiones académicas o de prueba.</p> <p>• El tiempo de aprendizaje es elevado.</p> <p>• Diseñado para simular casos prácticos/concretos.</p>	<p>• Por ser un software de aplicación en áreas comerciales y para efectos de investigación y desarrollo, tiene un alto grado de complejidad en su manejo.</p> <p>• Para fines de investigación y desarrollo, es necesario saber programar en lenguaje NED, ya que el trabajo con el editor gráfico, es un poco más rígido.</p> <p>• Pocos modelos de equipos y enlaces.</p>	<p>• La configuración de las simulaciones a través de código, hace que sea mayor el tiempo de desarrollo. Además se incrementa el tiempo necesario para el aprendizaje del software.</p> <p>• NS-2 requiere varios componentes instalados para su correcto funcionamiento.</p>
V E N T A J A S		
D E S V E N T A J A S		

Figura 4.1: Comparación de simuladores de red

De acuerdo a la comparación realizada anteriormente, el simulador con el cual se trabajará es el NETWORK SIMULATOR 2 (NS-2) en la versión 2.34, bajo la plataforma del sistema operativo Ubuntu 10.04. Las razones generales por las que se ha elegido este software son por ser de código libre, estar ampliamente difundido dentro del ámbito académico y telecomunicaciones permitiendo su utilización en muchas topologías de redes, incluyendo la de nuestro interés. A continuación se describirá de mejor manera esta herramienta para enfocarnos en su funcionamiento.

#### **4.1.1.1 Network Simulator NS-2**

El NS-2 [18] [20], es un simulador de redes dirigido por eventos. El uso de este simulador está ampliamente extendido en ambientes académicos debido a que está escrito en código abierto y se pueden simular gran variedad de protocolos tanto unicast como multicast y sobretodo muchos de los protocolos propios de las redes Ad Hoc.

NS-2 está encapsulado dentro del lenguaje Tcl (Tool Command Language), el motor del simulador está implementado en C++, ya que la simulación requiere de este lenguaje para manipular eficientemente bytes, paquetes, cabeceras e implementar algoritmos, está configurado y controlado mediante la interfaz Tcl. Utilizando las instrucciones de NS-2 se define la topología de la red, se configuran las fuentes de tráfico, se recogen los resultados en un fichero de salida y se invoca el simulador. Las instrucciones de NS-2 permiten invocar los procedimientos Tcl desde puntos arbitrarios de la simulación, ofreciendo un mecanismo que generan datos de salida y se almacenan en un fichero traza para luego analizar los resultados de la red simulada, ya sea con xgraph para ver gráficas o con otros lenguajes de programación como Network Animator (NAM), que permite observar visualmente el envío y recepción de paquetes entre nodos.

#### **La Visualización NAM**

Es una herramienta de soporte para poder interpretar y entender las simulaciones de un modo visual, es imprescindible el uso de este programa, sobre todo al principio, para entender que es lo que está sucediendo en la simulación y así saber como

tratar los resultados obtenidos.

### **Gráfico de seguimiento (TraceGraph)**

El TraceGraph [21], sirve para graficar y especialmente analizar las trazas generadas por el simulador de redes NS2. Entre las utilidades más importantes tenemos:

- La posibilidad de obtener información y estadísticas de la red entera y relaciones entre nodos (links).
- Los resultados pueden ser guardados como archivos de textos, gráficos, imágenes JPG y TIFF.
- Toda traza que posea 2 o 3 columnas de información puede ser graficada.
- Información en los ejes X,Y y Z, como mínimos, máximos, desviaciones, etc.
- Todo script al ser procesado es analizado automáticamente.

Además otra utilidad y la más importante en nuestro caso es que gracias a TraceGraph podemos observar y analizar las diferencias que existen en ámbitos de calidad de servicio (QoS), entre los distintos tipos de protocolos de redes. Esto debido a la gran variedad de análisis que ofrece TraceGraph, como por ejemplo Throughput, paquetes perdidos, paquetes enviados, paquetes recibidos, entre otros.

#### **4.1.2. Arquitectura de Servicios Diferenciados (DiffServ) aplicada a redes Ad Hoc**

La arquitectura de DiffServ se basa principalmente en la realización de un script en NS2 que contenga las características y configuración necesarias para llevar a cabo la simulación. Motivo por el cual a continuación se detallan los componentes y funciones que intervienen en una red con modelo DiffServ:

### 4.1.2.1 Definición de escenario

Los nodos son parte de los elementos importantes que conforman la red y en nuestro caso trabajaremos con un escenario de 100 nodos distribuidos en dos bloques laterales, de los cuales 3 son router, marcados con el número 1 y 2 como router frontera (Edge 1 y Edge 2), y el router núcleo (Core) marcado con el número 0, los mismos que actúan como puente entre las dos partes, el escenario tiene un área de 1400m x 600m. Se ha escogido este esquema para poder ver la diferencia de mejor manera, ya que cabe destacar que cualquier nodo de la red es capaz de actuar como router frontera y como router núcleo.

**Router Frontera (Edge router).**- El router frontera de una red Ad Hoc [22], sirve para conectar a los routers centrales o internos, éstos son responsables de asignar los puntos de código para los paquetes de acuerdo a la política especificada por el administrador de red y para ello se miden los parámetros del tráfico de entrada de cada flujo. En sí las características más importantes de este router es clasificación de paquetes, control de admisión y administración de la configuración. En la simulación, el nodo 1 toma el nombre de Edge1 y el nodo 2 el nombre de Edge2, y serán a quien se les asigne y configure las funciones que se mencionan seguidamente:

**Clasificación** La clasificación es la parte inicial de las funciones, en donde identifica el perfil de comportamiento por saltos PHB al que corresponde un flujo de tráfico. Los clasificadores de agregados son aquellos que utilizan únicamente el código DSCP, mientras que los clasificadores multicampo utilizan más información de diferentes campos disponibles en el paquete.

**Acondicionamiento** Esta función tiene por objetivo conseguir que el tráfico que ingrese a un dominio Diffserv se ajuste a un acuerdo de tráfico acondicionado TCA [23]. Para realizar la función de acondicionamiento se realizan una serie de subfunciones como:

- **Medidor** El medidor comprueba si el tráfico de entrada se ajusta a un patrón de tráfico determinado y transmite esta información al resto de módulos implicados

para que puedan tomar las medidas necesarias.

- **Marcador** El marcador se encarga de asignar un código DSCP a los paquetes de entrada, determinando de esta forma el agregado al que pertenecen. Los paquetes de entrada pueden venir previamente marcados o no, y esta información puede influir en la decisión de marcado.
- **Conformador y Descarte** Finalmente, estas dos funciones se encargan de que el tráfico de entrada se ajuste al TCA. Para ello el conformador implementa un buffer que puede retardar la salida de los paquetes y en caso de que el buffer se desborde, se produce el descarte de los mismos.

**Core Router.-** El core router [22], está diseñado para operar en el backbone de internet y debe ser capaz de soportar múltiples interfaces de telecomunicaciones, enviar paquetes IP a velocidad máxima y ser compatible con los diferentes protocolos de enrutamiento, ya que el enfoque básico de DiffServ es tratar y evitar la congestión, y, mantener la inteligencia en el borde de la red. De la misma manera quien tomará este atributo será el nodo 0, llamado Core, el mismo que asigna la debida prioridad a los paquetes de acuerdo a su código de marca y se traduce a los parámetros de la programación y del lanzamiento de decisiones en los routers núcleo.

Para tener una mejor percepción de lo antes mencionado en la figura 4.2 se describe el escenario y las partes que lo conforman.

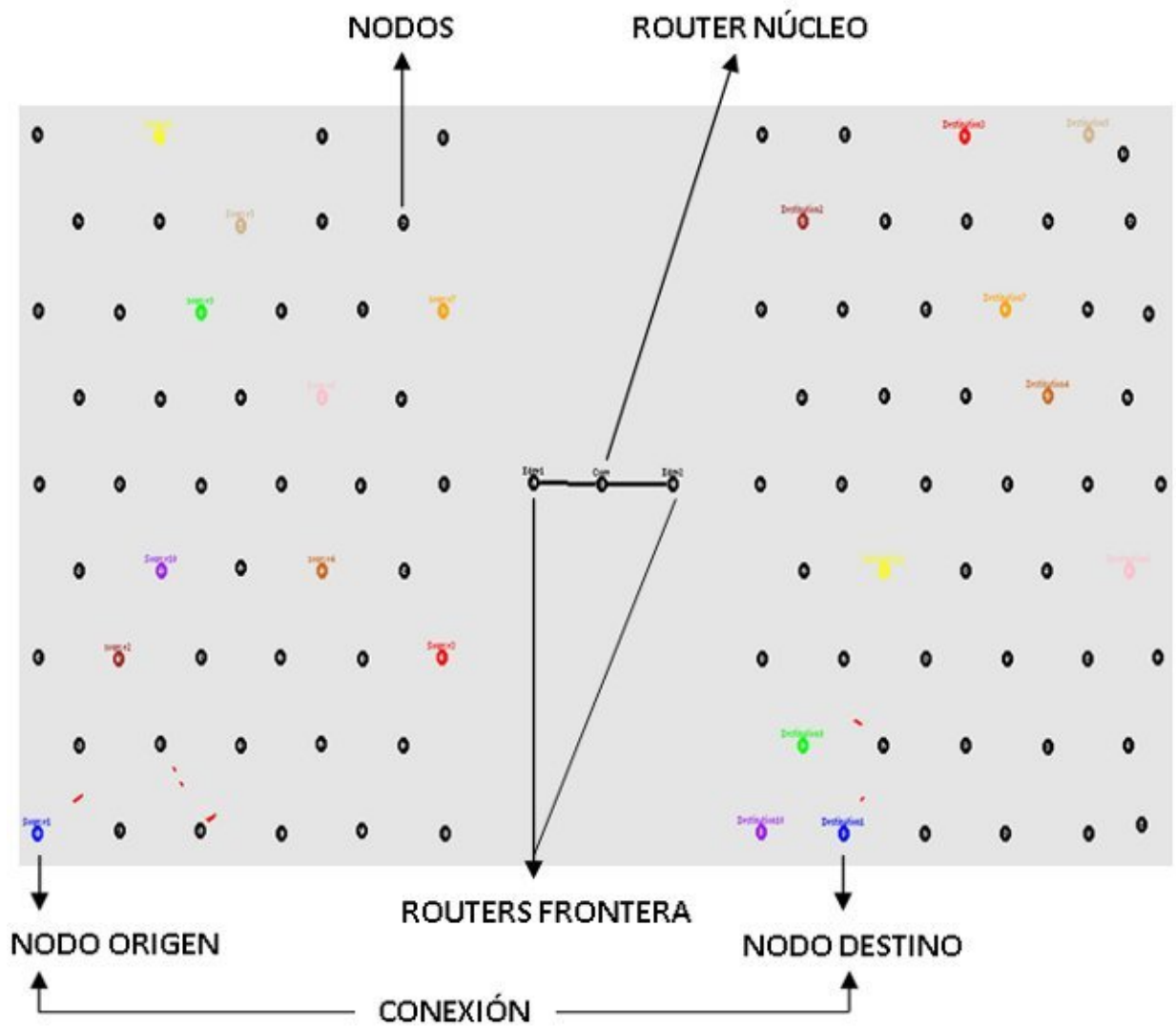


Figura 4.2: Estructura del escenario de una red Ad Hoc

### 4.1.2.2 Definición de scripts

Para realizar la simulación será necesario utilizar 2 scripts, en su mayoría será estándar para las simulaciones, destacando que se modificará el tipo de protocolo a utilizar como son (AODV, DSDV, DSR), y la configuración del modelo de Calidad de Servicio DiffServ, con la finalidad de demostrar sus diferencias. A continuación se presentan las características principales del script con la arquitectura de modelo de QoS DiffServ.

Área de simulación.....	1400m x 600m
N. de terminales.....	100
Modelo de tráfico.....	CBR
Ancho de banda nominal.....	11 Mbps
Protocolo de encaminamiento ad hoc.....	AODV
Capa de enlace.....	802.11
Tamaño de paquetes en cola.....	500 paquetes
Tiempo de simulación.....	500 ms
Número de conexiones.....	10
Modelo de QoS.....	DiffServ

Se ha establecido el área de simulación de acuerdo a nuestros requerimientos, ya que se trabajará con 100 nodos y con una estructura adaptable a los routers central y frontera que se encuentran como intermediarios. El modelo de tráfico que se utilizará es de tasa de velocidad constante (CBR) porque divide la capacidad total del ancho de banda y se encarga de esa parte con un flujo constante de tráfico, además es uno de los modelos que más se emplea en los enlaces extremo a extremo. Por su parte el ancho de banda nominal es de 11Mbps ya que toma en cuenta el estándar 802.11 que permite mayores niveles tanto de movilidad como de portabilidad en una red inalámbrica y su rango de frecuencia alcanza los 2.4 GHz, cabe destacar que el estándar más conveniente con el que deberíamos trabajar es el 802.11e ya que está destinado a mejorar la cali-

dad del servicio en el nivel de la capa de enlace de datos, definiendo los requisitos de diferentes paquetes en cuanto al ancho de banda y al retardo de transmisión, pero no se ha trabajado con dicho estándar por no estar implementado en la herramienta de simulación NS-2. Se ha utilizará los protocolos de enrutamiento más importantes y que están implementados en la herramienta de simulación como AODV, DSDV y DSR, en la siguiente sección de analizarán más detalladamente estos protocolos. Así mismo el tiempo de simulación se lo establecerá de acuerdo al número de nodos, conexiones y estructura del escenario a simular. Un ejemplo más claro de codificación de un script se encuentra en el apéndice A.

### 4.1.2.3 Protocolos de Enrutamiento

Los protocolos de enrutamiento [24], permiten a los routers de la red poder dirigir o enrutar los paquetes usando tablas de enrutamiento, por lo tanto es importante definir estos conceptos debido a que en el presente trabajo de investigación se utilizará 2 routers frontera y 1 router central, los mismos que necesitan de un protocolo para construir y actualizar su tabla de enrutamiento determinando la mejor ruta para poder llegar a cualquier nodo remoto, motivos por lo que es necesario conocer el papel que cumple cada uno de éstos a la hora de realizar el proceso de comunicación, ya que en los escenarios que se simulará se hará uso de los routers y protocolos de encaminamiento que están implementados en la herramienta NS2 como: DSDV, AODV y DSR.

**Destination-Sequenced Distance Vector (DSDV)** DSDV [25], es un protocolo proactivo que proporciona básicamente la obtención de rutas sin bucles mediante la introducción de números de secuencia, suministrando una sola ruta para cada destino, siendo ésta la ruta con la trayectoria más corta posible.

Entre sus características tenemos:

- Los paquetes de señalización son intercambiados entre nodos vecinos a intervalos regulares de tiempo o emitidos por eventos
- Cada nodo perteneciente a la red tiene una tabla de enrutamiento que indica para

cada destino cuantos saltos (hop) hacen falta atravesar y cual es el sucesivo

**Ad Hoc On Demand Distance Vector Routing (AODV)** AODV [26], es un protocolo reactivos, diseñado para redes móviles Ad Hoc que establece rutas bajo demanda, es decir, no mantiene actualizaciones periódicas entre nodos, sino que descubre y mantiene rutas solamente cuando son necesarias, a través de los procesos de descubrimiento y de mantenimiento de ruta, usa tablas de enrutamiento en cada nodo para de esta manera evitar transportar las rutas en los paquetes de datos y utiliza números de secuencia para evitar los lazos de enrutamiento que pudieran darse.

Una particularidad de AODV es la reparación a nivel local de un enlace caído que forma parte de una ruta activa y procede a intentar repararlo comenzando un proceso de descubrimiento de ruta hacia el destino y encolando los paquetes de datos recibidos para el destino, hasta localizar una nueva ruta.

Entre las características más importantes tenemos:

- Las rutas se establecen por demanda
- Se utilizan números de secuencia para tener rutas actualizadas hacia el nodo destino
- Reduce la cantidad de difusiones (broadcasts)
- Utiliza números de secuencia de destino para evitar “ciclos” y mantener información reciente
- Almacena la información de encaminamiento en forma de tablas o de rutas
- Emplea mensajes hello o beacons para que los nodos anuncien a sus vecinos su pertenencia a la red
- AODV es una combinación de ambos protocolos DSR y DSDV
- AODV busca establecer una ruta sólo cuando es necesario
- AODV es un protocolo muy simple que funciona bien bajo condiciones normales

**Dynamic Source Routing (DSR)** DSR [27], es un protocolo reactivo basado en la fuente, crea las rutas únicamente en el caso de que un nodo fuente necesite enviar datos hacia un nodo destino. Se diseñó para optimizar el uso del ancho de banda dentro de una red Ad Hoc. Este protocolo hace que los nodos no posean una tabla de enrutamiento, debido a que la ruta viaja en el paquete de datos y solo mantiene un registro de rutas, así mismo cada nodo asocia un temporizador a la ruta con el fin de poderla eliminar en el caso de que esté inactiva por un cierto periodo de tiempo.

Entre las características más importantes tenemos:

- Es otro protocolo de enrutamiento bajo demanda
- No requiere ningún tipo de mensajes periódicos
- Utiliza enrutamiento de origen en lugar de confiar en la tabla de enrutamiento en cada dispositivo
- Pocos mensajes de control
- No requiere broadcasts periódicos de información de encaminamiento
- Overhead reducido o sea que se gestionan sólo las rutas entre los nodos que desean comunicarse
- Pocos mensajes de control
- No requiere broadcasts periódicos de información de encaminamiento
- Las rutas aprendidas se utiliza para encaminar los paquetes
- La ruta de respuesta sólo se genera si el mensaje ha llegado al nodo destino
- Ofrece la posibilidad de obtener, con la solicitud de una ruta, múltiples caminos posibles hacia el destino

## 4.2. Fase de análisis y comparación

Esta fase trata sobre realizar un análisis con todos los resultados, como datos numéricos y gráficas que arroje el simulador y con todo esto llevar a cabo una comparativa parcial para llegar a una conclusión determinada. Por lo tanto cumpliremos esta fase valiendonos de indicadores que ayudarán a obtener mejores resultados y en el siguiente capítulo se detallará específicamente esta comparativa.

### 4.2.1. Indicadores de QoS para redes Ad Hoc

Los indicadores de QoS se los ha propuesto analizar en base a los problemas que se presentan en una red Ad Hoc según [5], son los siguientes:

#### **Rendimiento (throughput)**

Se define el throughput como la cantidad de información (paquetes) recibida en el nodo destino entre la cantidad de información (paquetes) enviados por el nodo fuente. Para su cálculo hay muchos aspectos a tener en cuenta como: cabeceras de paquetes, paquetes de señalización, tiempos de espera, colisiones, retransmisiones de paquetes, etc.

#### **Variación del retardo de los paquetes (Jitter)**

Se define como un tiempo de latencia variable sobre los datos de recepción, osea la variación de los retardos en la llegada de los paquetes entre su origen y el destino, producida por congestión de tráfico en algún punto de la red o diferencia en el tiempo de transito de paquetes cuando estos viajan por diferentes rutas. Mediante la simulación de este indicador se comprobará como actúa la capacidad variable de los enlaces inalámbricos en una red Ad Hoc.

## **Retardo extremo a extremo (Delay)**

Es el tiempo que emplea un paquete para viajar desde la fuente al destino, el retardo en redes MANET suele ir desde milisegundos hasta varios cientos de segundos. En general se encuentra compuesto por el retardo de proceso, retardo de cola, retardo de transmisión y retardo de propagación. Indicador que nos ayudará a medir el problema de las topologías dinámicas que se presentan en estas redes.

## **Paquetes borrados**

Cada vez que un router no está sobrecargado y no deja caer los paquetes, aunque el rendimiento es malo, se supone que se borran paquetes en la red. En este caso, la red se refiere a una gran variedad de cosas, incluyendo los medios de comunicación y dispositivos de red, así como toda la gama de características que se pueden poner en práctica en el transporte de paquetes entre el cliente y el servidor.

## **Paquetes perdidos**

Son los paquetes transmitidos que han sido descartados o perdidos en la red, esto se puede dar debido a muchos factores variables como la calidad del enlace, información de enrutamiento desactualizada y por errores en alguno de los equipos que permiten la conectividad de la red o por sobrepasar la capacidad de algún buffer de un equipo o aplicación en momentos de congestión.

## **4.3. Conclusiones**

En esta sección se analizó la metodología a utilizar para la simulación de la red Ad Hoc, tomando en cuenta la fase de diseño y codificación, en donde se presenta al NS-2 como simulador de trabajo, así como también los scripts y escenarios, en donde se puede observar el comportamiento de los protocolos que se simulará. La segunda fase es de análisis y comparación de resultados, en la que se utilizará cinco indicadores de calidad de servicio para las comparativas respecto al tema, como el rendimiento, variación del retardo de paquetes, retardo extremo a extremo, paquetes borrados y paquetes perdidos.

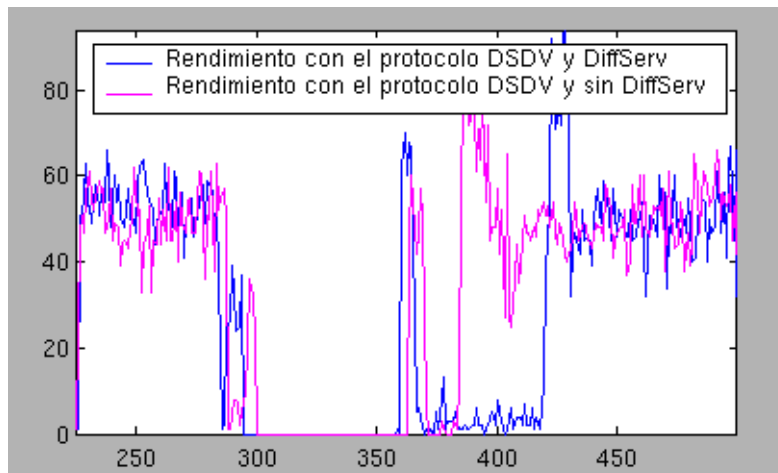
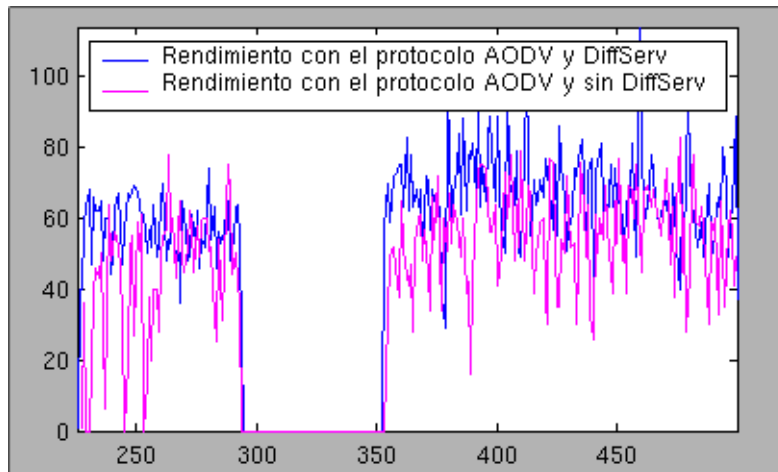
# Capítulo 5

## Interpretación y Discusión de Resultados

A continuación, se mostrarán los resultados de las pruebas realizadas tanto para el script con diffServ como sin diffServ específicamente con 10 conexiones, los datos fueron tomados a partir de las simulaciones que se hicieron mediante la herramienta de simulación NS2 incluyendo las gráficas y análisis de las mismas, todo esto con los indicadores mencionados en el Capítulo 4.

## 5.1. Rendimiento

Para evaluar en que grado y con que eficiencia funciona la transmisión de los paquetes entre el nodo fuente y el nodo destino, realizamos la simulación en un tiempo de 500 segundos, para ambos escenarios (con DiffServ y sin DiffServ), 10 conexiones y con los 3 protocolos (AODV,DSDV,DSR), dándonos como resultado lo que está expresado en la figura 5.1, datos que han sido expuestos numéricamente en la figura 5.2.



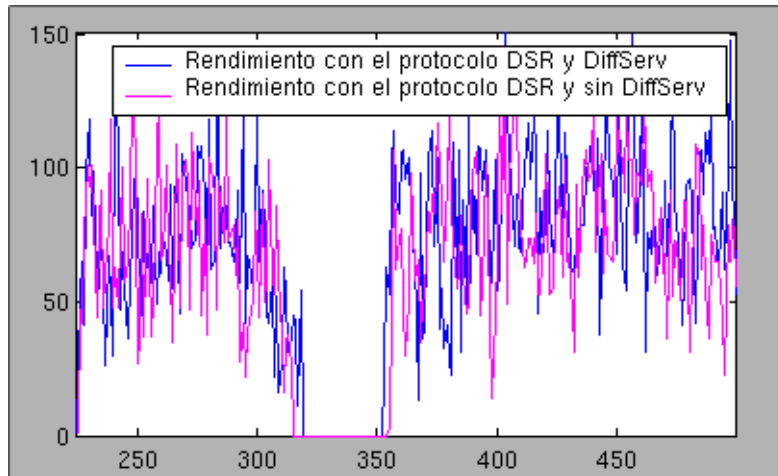


Figura 5.1: Rendimiento promedio de la simulación en el traceGraph

PROTOCOLOS	RENDIMIENTO PROMEDIO SIN SERVICIOS DIFERENCIADOS	RENDIMIENTO PROMEDIO CON SERVICIOS DIFERENCIADOS
AODV	80	100
DSDV	80	90
DSR	140	150

Figura 5.2: Datos cuantitativos del rendimiento promedio de la simulación

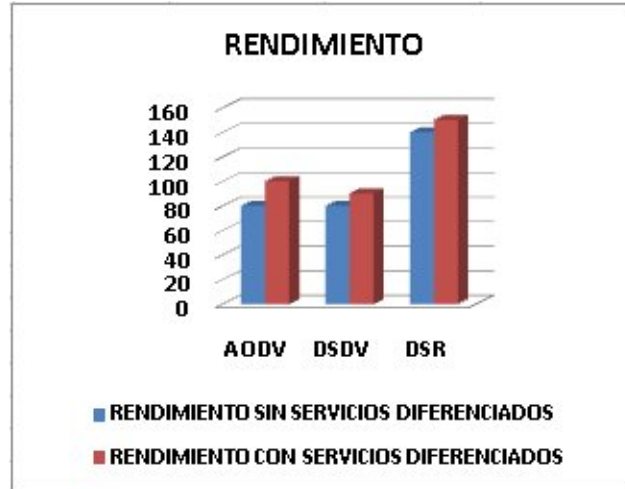


Figura 5.3: Comparación del rendimiento promedio en la red

En la gráfica anterior se puede observar como el rendimiento promedio se comporta de forma más eficiente con la configuración de servicios diferenciados en la red Ad Hoc y con todos los protocolos, esto se debe a que el modelo asigna flujos múltiples en los niveles de servicio y la cantidad de recursos asignados a cada paquetes es buena, proporcionando precisión, fiabilidad y disponibilidad de la transferencia de información desde su inicio hasta su final.

## 5.2. Variación del retardo

La variación del retardo debe ser baja para poder asegurar que los paquetes puedan ser entregados a tiempo y en este caso vamos a ver los resultados que se obtuvieron:

PROTOCOLOS	VARIACIÓN DE RETARDO PROMEDIO SIN SERVICIOS DIFERENCIADOS	VARIACIÓN DE RETARDO PROMEDIO CON SERVICIOS DIFERENCIADOS
AODV	0,5	0,45
DSDV	1	2
DSR	0,5	1,2

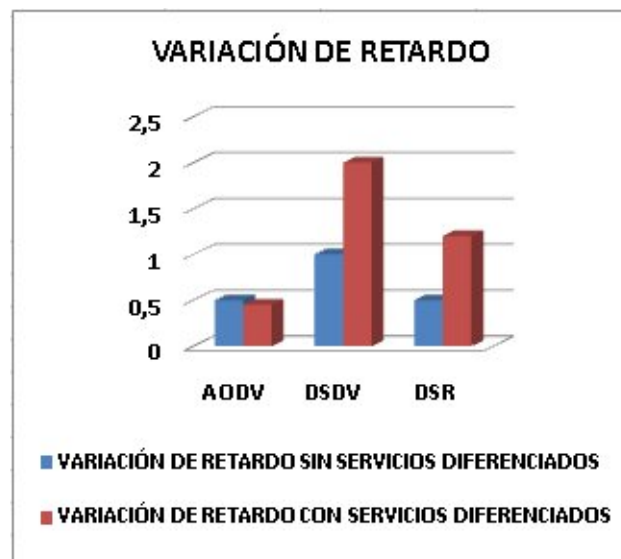


Figura 5.4: Comparación de la variación del retardo promedio en la red

En la grafica 5.4, se puede apreciar que la variación del retardo promedio es más baja con el protocolo AODV y con modelo de servicios diferenciados, ya que el flujo de datos sigue un nivel de prioridad de paquetes a enviar y porque cada uno de éstos utiliza un proceso de señalización de QoS, lo que ayuda a tener una conexión rápida y con un número menor de saltos garantizando su rendimiento y eficiencia de datos en cada paquete.

### 5.3. Retardo extremo a extremo

El objetivo con éste indicador es emplear el menor tiempo posible entre el viaje de un paquete desde la fuente al destino, al momento de realizar la simulación se obtuvieron los siguientes datos:

PROTOCOLOS	RETARDO PROMEDIO EXTREMO A EXTREMO SIN SERVICIOS DIFERENCIADOS	RETARDO PROMEDIO EXTREMO A EXTREMO CON SERVICIOS DIFERENCIADOS
AODV	0,021280876	0,008175761
DSDV	0,01083232	0,007955846
DSR	0,102426316	0,044320298



Figura 5.5: Comparación del retardo promedio extremo a extremo en la red

Como se puede observar en la figura 5.5, todos los protocolos con la configuración de servicios diferenciados mantienen un nivel de tiempo bajo, esto se da porque los routers, son los encargados de clasificar los paquetes y acondicionar el tráfico para asignar a cada conexión la ruta más corta, así mismo el comportamiento por salto (PHB) proporciona una clase con abstracción simple y un ancho de banda de enlace mínimo garantizado, que es utilizado para construir el servicio de extremo a extremo.

## 5.4. Paquetes borrados

En todo el proceso de envío de paquetes se pueden borrar varios de ellos por diferentes razones, ya sea por la demora en el envío, por exceder en el número de saltos y no llegar correctamente al destino. Pues el objetivo es obtener el menor número de paquetes borrados así que seguidamente se presenta los datos que la simulación arrojó.

PROTOCOLOS	PAQUETES BORRADOS SIN SERVICIOS DIFERENCIADOS	PAQUETES BORRADOS CON SERVICIOS DIFERENCIADOS
AODV	160977	158014
DSDV	162307	163775
DSR	166188	166849

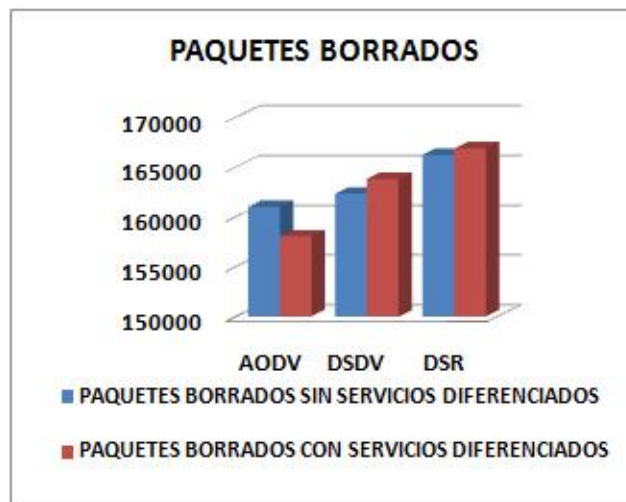


Figura 5.6: Comparación de los paquetes borrados en la red

De igual manera en la figura 5.6, con el protocolo AODV (reactivo) y modelo de servicios diferenciados se sigue obteniendo mejores resultados ya que tiene un número menor de paquetes borrados y porque en dicho modelo se utiliza todos los recursos necesarios tratando cada enlace de manera diferente de acuerdo a su prioridad para reducir la congestión en la red.

## 5.5. Paquetes perdidos

La mayor cantidad de paquetes perdidos en una red Ad Hoc es muy probable ya que la movilidad, interferencia y recursos que se usan lo hacen posible, por lo tanto lo que todo usuario desea es que la información llegue completa, correcta y en el menor tiempo posible, seguidamente se puede apreciar los resultados de la simulación.

PROTOCOLOS	PAQUETES PERDIDOS SIN SERVICIOS DIFERENCIADOS	PAQUETES PERDIDOS CON SERVICIOS DIFERENCIADOS
AODV	0	0
DSDV	20840	19988
DSR	24497	32689

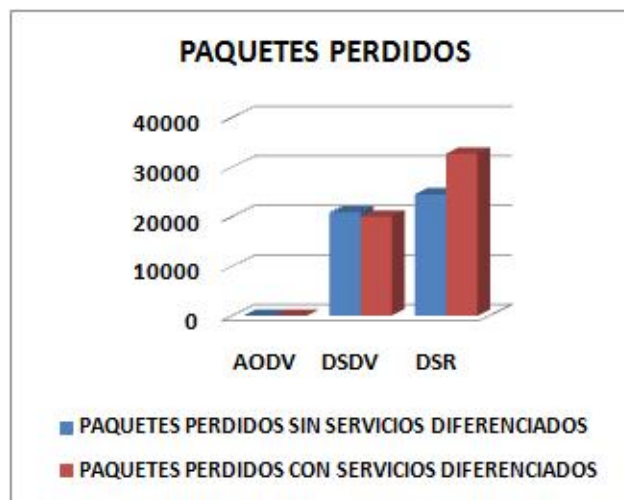


Figura 5.7: Comparación de los paquetes perdidos en la red

En cuanto a los paquetes perdidos en la figura 5.7, se puede visualizar que con el protocolo AODV y con el modelo de Servicios Diferenciados no hay paquetes perdidos, eso se debe a que no hubo desbordamiento de buffer y porque se puede reenviar paquetes tan pronto como se detecte alguna inconsistencia en la red.

## 5.6. Comportamiento del router frontera y núcleo

Para demostrar la función que realizan el router frontera y núcleo se ha escogido específicamente una de las 10 conexiones del ejemplo con el cual se está trabajando, debido a que las demás realizan los mismos procesos aunque con diferentes resultados, y con una es más que suficiente para demostrar sus funciones dentro de la comunicación entre nodos, en este caso la conexión es desde el nodo 15 al nodo 60 y como routers el edge 1 y core 0. Cabe destacar que en los escenarios se trabajará con 2 router frontera pero en este caso para demostrar el comportamiento se ha tomado en cuenta uno de ellos ya que cumplen funciones parecidas.

### Router Frontera

PROTOCOLOS	PAQUETES BORRADOS SIN SERVICIOS DIFERENCIADOS	PAQUETES BORRADOS CON SERVICIOS DIFERENCIADOS
AODV	1	0
DSDV	0	0
DSR	3	6

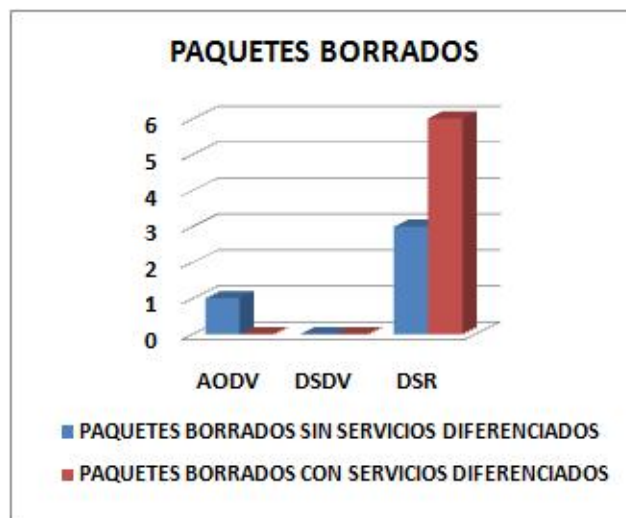


Figura 5.8: Paquetes borrados desde el nodo 15 al router frontera

En la figura 5.8, se puede ver la función que realiza el router frontera ya que borra el menor número de paquetes, los cuales una vez que entran en la red estarán marcados, es decir, el encabezado del paquete se ajusta a algún valor y la marca que el paquete recibe depende de las propiedades medidas temporalmente por flujo del paquete. Después de haber sido marcados, el paquete puede permitir la entrada en la red inmediatamente, el retraso por algún tiempo antes de su envío o descartados por completo.

Esto se realiza por la función de acondicionamiento de tráfico para garantizar el paso de la información sin datos erróneos, luego de pasar por el router frontera y una vez realizado todo este proceso pasa al router núcleo.

## Router Núcleo

Current node information:	
Number of generated	0
Number of sent	0
Number of forwarded	28
Number of received	0
Number of dropped	0
Number of lost	0
Number of sent	0
Number of	1232
Number of received	0
Number of dropped	0
Minimal packet	44
Maximal packet size:	44
Average packet	44

Figura 5.9: Datos del router núcleo, protocolo AODV y modelo diffServ

Current node information:	
Number of	0
Number of sent	0
Number of forwarded	0
Number of received	0
Number of dropped	0
Number of lost	0
Number of sent	0
Number of	0
Number of received	0
Number of dropped	0
Minimal packet	0
Maximal packet size:	0
Average packet	0

Figura 5.10: Datos del router núcleo, protocolo DSDV y modelo diffServ

Current node information:	
Number of	1
Number of sent	1
Number of forwarded	63
Number of	1
Number of dropped	0
Number of lost	0
Number of	56
Number of	7436
Number of	68
Number of dropped	0
Minimal packet	48
Maximal packet	240
Average packet	118.3077

Figura 5.11: Datos del router núcleo, protocolo DSR y modelo diffServ

El router núcleo va en contraste con el router frontera, ya que definitivamente este actúa como la columna vertebral de toda la red y en este caso los paquete que llegan se envía en su siguiente salto de acuerdo con el comportamiento por salto (PHB), actuando en forma de amortiguadores de un router y ancho de banda de enlace que son compartidos entre las clases que compiten el tráfico. Por su parte los routers núcleo ya no necesitan mantener información de estado para los pares origen-destino, sino que en el caso de que algún dato estuviera inconsistente en el paquete pues éste se reenviará y encaminará para garantizar escalabilidad en la comunicación.

## 5.7. Conclusiones

Para saber si existe Calidad de Servicio en la red Ad Hoc se ha procedido ha analizar cinco indicadores de acuerdo a la simulación realizada, como es el caso del rendimiento promedio, variación de retardo promedio, retardo promedio extremo a extremo, paquetes borrados y paquetes perdidos, los cuales han sido calculados a través de los paquetes recibidos, mediante la herramienta de simulación NS2. Determinando que todos estos parámetros funcionan de mejor manera con el modelo de Servicios Diferenciados y con protocolos reactivos en este caso el AODV.

Se deduce que este modelo proporciona mejores beneficios porque el flujo de paquetes enviados es clasificado en una cierta clase de tráfico para asignar la prioridad que tiene cada paquete y cada enlace de salida en un cierto intervalo de tiempo y de esa manera evitar la congestión de la red, así mismo reenvía paquetes en el caso de que

existiese inconsistencia de la red o recursos.

Por su parte los routers frontera y núcleo son los componentes principales del modelo DiffServ ya que priorizan los paquetes, los reenvían, asignan la ruta más corta, etc., ayudando de esta manera a tener una comunicación eficiente en sus conexiones, como ejemplo se ha tomado la conexión desde el nodo 15 al nodo 60, en donde el router frontera recibe los paquetes desde el nodo 15, revisando inconsistencias o errores y si es necesario borrando y etiquetando en la cabecera de cada paquete estos datos, luego estos paquetes pasan al router núcleo el cual está encargado de reenviar, encaminar y asignar una ruta más corta, y son enviados al siguiente router frontera que hace los procesos antes mencionados, para luego pasar la información al nodo destino, cabe recordar que estos datos son tomados de acuerdo a los paquetes enviados.

# Capítulo 6

## Conclusiones

En el presente trabajo de investigación se ha podido recabar la información teórica necesaria para poder entender el alto nivel de importancia que tienen las redes Ad Hoc en la sociedad y que en sí éstas vienen a cubrir las deficiencias que otras redes quizá no han podido satisfacer. Por lo tanto la mayoría de las empresas lo que necesita es una red que soporte todo el tráfico y que garantice una comunicación eficiente, que la diferencie y la priorice a fin de tener transmisión acorde a cada tipo de tráfico.

Luego de haber analizado y desarrollado la *Simulación De QoS En Una Red Ad Hoc*, mediante la herramienta de simulación Network Simulation en su versión 2.34, con la que se ha trabajado para crear escenarios y hacer estudios comparativos del comportamiento de los protocolos y modelo de simulación que se ha utilizado como es el de Servicios Diferenciados, se pudo establecer las siguientes conclusiones específicas:

- Se diseñó la estructura del escenario de la red a simular con 100 nodos móviles, en donde 2 de ellos son nodos frontera y 1 nodo central, lo mismos que actúan como puente entre el envío y recepción de paquetes, cabe destacar que dentro de esta estructura se establecieron 10 conexiones diferentes
- Mediante la investigación que se llevó a cabo sobre los simuladores de redes se optó por trabajar con el Network Simulator en su versión 2.34, la cual de manera eficaz ayudó a arrojar datos y gráficas importantes

- El rendimiento, un indicador que evalúa la eficiencia de funcionamiento a partir de la capa de red ya que desde ahí se hace el direccionamiento lógico y determinación de una posible ruta, hasta la capa de aplicación que nos ofrece la posibilidad de interactuar con los datos o servicios enviados, por lo tanto en la simulación realizada se puede destacar que existe del 10 al 20 % de diferencia promedio en su rendimiento con el modelo DiffServ
- La variación del retardo de los paquetes o jitter debe ser baja para lograr una buena comunicación y en nuestro caso el jitter promedio más bajo es de 0.0045 % el mismo que tiene la configuración del modelo diffserv y el protocolo AODV
- La finalidad del retardo extremo a extremo o Delay es emplear el menor tiempo posible entre el viaje de un paquete desde la fuente al destino, por cuanto lo recomendable es que sea menos de 150 ms entre los nodos extremos para tráfico en tiempo real. Por lo que en nuestro caso se tiene el 0,000081758 % de tiempo empleado para la transmisión de cada paquete con el modelo DiffServ
- El indicador de paquetes borrados se analiza de acuerdo a la cantidad de fallos o inconsistencia en la red y sus recursos, el 1580,14 % es el menor número de paquetes borrados que se dió utilizando DiffServ
- En lo que se refiere a paquetes perdidos el codec por defecto requiere que sea menor del 1 % para evitar errores y en este caso se tiene el 0 %, debido a que DiffServ consta de routers que ayudan a reenviar paquetes y a organizar de mejor manera la comunicación entre sus nodos
- Es necesario que los protocolos de encaminamiento se adapten rápidamente a los cambios constantes de la topología de la red para poder mantener una ruta entre los nodos fuente y destino y según los escenarios que se han simulado se concluye que hay mejores beneficios con el protocolo reactivo AODV
- Generalmente podemos concluir que mediante los cinco indicadores se ha podido comprobar que el modelo de servicios diferenciados si proporciona mejores ventajas al momento de llevar a cabo una comunicación entre varios nodos, sin importar el grado de movimiento, topología, disponibilidad de recursos y aun más

sin utilizar un punto de acceso, sino establecer una comunicación directa.

## 6.1. Recomendaciones

Concluyendo esta investigación se puede recomendar de manera general lo siguiente:

- Es recomendable seguir incursionando en la investigación de redes inalámbricas debido al vertiginoso avance de la tecnología móvil e inalámbrica, para estar al tanto de las noticias y proyectos relacionados con la misma, así como también para poder beneficiarnos al máximo de la utilidad que éstas ofrecen
- Se recomienda el estudio e implementación de sistemas para garantizar calidad de servicio en redes inalámbricas pues se tiene poca información del tema, lo cual serviría como una aportación a esta línea de investigación
- Además también se recomienda que en un futuro, se profundice y se lleve a la práctica los conocimientos acerca de tecnologías, protocolos y redes inalámbricas, dentro del pensum de estudios de la carrera de Ingeniería en Sistemas, debido a las dificultades que se presentan ya sea por falta de recursos, equipos y espacio

## 6.2. Trabajos futuros

Las redes Ad Hoc en la actualidad se encuentran en auge si de utilización se trata, por lo tanto existen muchas apartados de estas que aún no han sido explorados, los campos en los que se puede trabajar son muy amplios y aquí mencionamos algunas líneas que se pueden continuar a partir de nuestro proyecto:

- La implementación de un protocolo de QoS, en la herramienta de simulación NS2 y simular la red con los dos modelos de QoS, Servicios Integrados y Servicios Diferenciados
- Diseñar e implementar una red Ad Hoc con calidad de Servicios ya que en la actualidad solo se han llevado a cabo hasta la simulación
- Presentar e implementar estrategias de seguridad en una red Ad Hoc a fin de garantizar la confiabilidad de los datos

# Bibliografía

- [1] Zygmunt J. Haas y Deng Liang Jing y Ben y Panagiotis Papadimitratos y Sajama S. Redes ad hoc inalámbricas, 2002.
- [2] Zhao Jun-dom. Mobile ad hoc de redes: una tecnología esencial para pervasive computing. In *Actas de las Conferencias Internacionales de información-tecnología y la Info-net*, page 316 a 321, 2001.
- [3] Carlos García Ignacio Soto José Ignacio Moreno Ivan Vidal. Servicios de valor añadido en redes móviles ad-hoc. departamento de ingeniería telemática universidad carlos iii de madrid. [http://www.it.uc3m.es/ividal/articulos/telecom03\\_adhoc.pdf](http://www.it.uc3m.es/ividal/articulos/telecom03_adhoc.pdf).
- [4] Chenxi Zhu y M. Scott Corson. calidad de servicio de enrutamiento para redes móviles ad hoc, flarion technologies, bedminster, new jersey 07921, 2002.
- [5] Zhili Sun. University of Surrey. Satellite networking - principles and protocols, 2005.
- [6] R. Braden y D. Clark y Grupo R. Braden y Shenker S. y Shenker Scott y Lixia Zhang y Deborah Estrin y Sugih Jamin. servicios integrados en la arquitectura de internet : una visión general. 1994.
- [7] R. Zhang Braden y L. y S. Berson y S. Herzog y S. Jamin. Resource reservation protocol ( rsvp) - versión 1 de la especificación funcional, 1996.
- [8] Hector Raimondo y Javier Días Gustavo Mercado. Calidad de servicio en redes ip.pdf.
- [9] Ramjee Prasad. School of Electrical Engineering Sudhir Dixit and Computer Science. Wireless ip and building the mobile internet. Artech House, 2002.

- [10] Administrador de sistemas operativos. <http://www.adminso.es/wiki/index.php/3.2arquitecturad> Madrid 2008.
- [11] Prasant Mohapatra y Li Jian Gui y Chao. Qos en redes móviles ad hoc. universidad de california, escuela de ciencias de la computación, Diciembre 2002.
- [12] H. Xiao K.C. Chua and K.G. Seah. Relative service differentiation for mobile ad hoc networks, iee wireless communications and networks conference, new,, Orleans, USA, 2003.
- [13] K. Wu and J. Harms. "qos support in mobile ad-hoc networks", crossing boundaries- the gsa journal of university of alberta, vol. 1, no. 1,, Noviembre 2001.
- [14] Matías Grossglauser Tse y David. La movilidad aumenta la capacidad de las redes inalámbricas ad-hoc. *IEEE / ACM Transactions on Networking*, 10:477 a 486, 2002.
- [15] Jinyang Li y Charles Blake y Douglas SJ De Couto y Douglas De SJ y Hu Hu Couto y IMM Lee y Robert Morris. Capacidad de las redes inalámbricas ad hoc. page 61 a 69, 2001.
- [16] P. A. Farrington y Nembhard HB y Sturrock DT y GW Evans y Chang Xinjie. Network simulations with opnet. network technology research center school of eee, nanyang technological university singapore 639798, 1999.
- [17] Witold Drytkiewicz y Sroka Steffen y Handziski Vlado y H Vlado y Andreas Köpke y Karl Holger. Un marco para la movilidad omnet + +. technische university de berlin, 2003.
- [18] Andrés Lagar Cavilla andreslc@cs.toronto.edu. Department of Computer Science University of Toronto. Manet extensiones para ns2.
- [19] Elena Zamarro Parra. Generación de escenarios de redes bluetooth compatibles con el simulador ns-2(pdf). Madrid 2008.
- [20] The Network Simulator NS2. <http://www.isi.edu/nsnam/ns/doc/index.html>.
- [21] Trace graph. <http://www.tracegraph.com/>.

- [22] G. Mamais, M. Markaki, G. Politis, and I. S. Venieris. Efficient buffer management and scheduling in a combined intserv and diffserv architecture: A performance study. In *in Second International Conference on ATM*, pages 236–242, 1999.
- [23] Yasser L. Morgan and Thomas Kunz. Pylon: 1 an architectural framework for ad-hoc qos interconnectivity with access domains. carleton university, ottawa, ont., canada k1s 5b6.
- [24] Elizabeth M. Royer y Toh Chai Keong. A review of current routing protocols for ad-hoc mobile wireless networks. dept. of electrical y computer engineering-university of california, santa barbara y georgia institute of technology, atlanta.
- [25] Charles E.Perkins y Pravin Bhagwat IBM. Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers. watson research center y computer science departament, university of maryland.
- [26] Charles E. Perkins y Elizabeth M. Royer. Ad- hoc on-demand de enrutamiento vector distancia. In *EN LOS PROCESOS DE LA 2 ° TALLER IEEE SOBRE APLICACIONES MÓVILES DE INFORMÁTICA DE SISTEMAS Y*, pages 90–100, 1997.
- [27] David B. Johnson y David A. Maltz y Broch Josh. Dsr : El dinámico fuente protocolo de enrutamiento de multi- hop redes inalámbricas ad hoc. In *En ad hoc una red , editado por Charles E. Perkins , Capítulo 5*, pages 139–172. Addison-Wesley, 2001.

# Apéndice A

## Script de simulación

A continuación se presentará el fichero .tcl que se ha usado para la simulación en el NS2, el mismo que está configurado con el modelo de calidad de servicio diffServ, protocolo de enrutamiento AODV y 10 conexiones, ahí podemos darnos cuenta de todos los parámetros y configuraciones que se ha utilizado para que la comunicación se de mejor manera.

```

# =====
# Define options
# =====
set val(chan) Channel/WirelessChannel ;# tipo de canal
set val(prop) Propagation/TwoRayGround ;# modelo de propagacion
set val(netif) Phy/WirelessPhy ;# tipo de interface
set val(mac) Mac/802_11 ;# tipo de protocolo MAC
set val(ifq) Queue/DropTail/PriQueue ;# tipo de encolamiento
set val(ll) LL ;# tipo de capa de enlace
set val(ant) Antenna/OmniAntenna ;# modelo de antena
set val(ifqlen) 500 ;# max numero de paquetes en cola
set val(nn) 100 ;# número de nodos móviles
set val(rp) AODV ;# protocolo de enrutamiento
set val(x) 1400 ;# X dimension de la topografia
set val(y) 600 ;# Y dimension de la topografia
set testTime 500 ;# Duration of simulation
set packetSize 1000 ;# Tamaño de paquetes
set rate0 500000 ;# Tasa de envio
set rate1 500000
set rate2 500000
set rate3 500000
set rate4 500000
set rate5 500000
set rate6 500000
set rate7 500000
set rate8 500000
set rate9 500000
set cbs0 3000
set cbs1 3000
set cbs2 3000
set cbs3 3000
set cbs4 3000
set cbs5 3000
set cbs6 3000
set cbs7 3000
set cbs8 3000
set cbs9 3000
set cir0 1000000
set cir1 1000000
set cir2 1000000
set cir3 1000000
set cir4 1000000
set cir5 1000000
set cir6 1000000
set cir7 1000000
set cir8 1000000
set cir9 1000000
set val(seed) 0.0

```

```

# =====
# Main Program
# =====
Phy/WirelessPhy set CPTHresh_ 10.0
Phy/WirelessPhy set CSThresh_ 5.011872e-12
Phy/WirelessPhy set RXThresh_ 1.02054e-10
Phy/WirelessPhy set Rb_ 11*1e6
Phy/WirelessPhy set bandwidth_ 11Mb
Phy/WirelessPhy set Pt_ 0.031622777
Phy/WirelessPhy set freq_ 2.437e9
Phy/WirelessPhy set L_ 1.0

Mac/802_11 set dataRate_ 11Mb
Mac/802_11 set basicRate_ 1Mb

# Inicializar variables globales
set ns_ [new Simulator]

set topo [new Topography]

# Abre el fichero de trazas para nam
set tracefile [open out1.2.tr w]
Sns_ trace-all $tracefile

set namtrace [open out1.2.nam w]
Sns_ namtrace-all-wireless $namtrace $val(x) $val(y)

# set up topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

#Crear el objeto God
create-god [expr $val(nn)]

#Usamos el nuevo formato de traza
Sns_ use-newtrace

#Configuración de los nodos
Sns_ node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -channelType $val(chan) \
    -topoInstance $topo \
    -wiredRouting OFF \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace OFF \
    -movementTrace OFF \

```

```

    for {set i 0} {$i < $val(nn)} {incr i} {
        set node_($i) [$ns_ node]
        $node_($i) random-motion 0;# enable random motion
    }

# Configuración de colores para los nodos
$ns_ color 0 red
$ns_ color 1 blue
$ns_ color 2 chocolate
$ns_ color 3 pink
$ns_ color 4 brown
$ns_ color 5 tan
$ns_ color 6 gold
$ns_ color 7 purple
$ns_ color 8 orange
$ns_ color 9 yellow
$ns_ color 10 green

$node_(3) color "blue"
$node_(3) shape "circle"

$node_(98) color "blue"
$node_(98) shape "circle"

$node_(15) color "brown"
$node_(15) shape "circle"

$node_(60) color "brown"
$node_(60) shape "circle"

$node_(19) color "red"
$node_(19) shape "circle"

$node_(53) color "red"
$node_(53) shape "circle"

$node_(23) color "chocolate"
$node_(23) shape "circle"

$node_(68) color "chocolate"
$node_(68) shape "circle"

$node_(44) color "tan"
$node_(44) shape "circle"

$node_(52) color "tan"
$node_(52) shape "circle"

$node_(34) color "pink"
$node_(34) shape "circle"

$node_(78) color "pink"
$node_(78) shape "circle"

```

```

$node_(41) color "orange"
$node_(41) shape "circle"

$node_(63) color "orange"
$node_(63) shape "circle"

$node_(48) color "yellow"
$node_(48) shape "circle"

$node_(81) color "yellow"
$node_(81) shape "circle"

$node_(38) color "green"
$node_(38) shape "circle"

$node_(93) color "green"
$node_(93) shape "circle"

$node_(21) color "purple"
$node_(21) shape "circle"

$node_(99) color "purple"
$node_(99) shape "circle"

# Asignando las coordenadas a los nodos
$node_(0) set X_ 300.0
$node_(0) set Y_ 290.0
$node_(1) set X_ 450.0
$node_(1) set Y_ 410.0

for {set i 2} {$i < $val(nn) } {incr i} {
    $node_($i) set X_ 250.0
    $node_($i) set Y_ 250.0
    $node_($i) set Z_ 0.0
}

#Movilidad de los nodos
$ns_ at 0.1 "$node_(1) set dest 615.0 297.0 5.0"
$ns_ at 0.1 "$node_(0) set dest 700.0 297.0 5.0"
$ns_ at 0.1 "$node_(2) set dest 785.0 297.0 5.0"

$ns_ at 0.1 "$node_(3) set dest 05.0 05.0 5.0"
$ns_ at 0.1 "$node_(4) set dest 105.0 05.0 5.0"
$ns_ at 0.1 "$node_(5) set dest 205.0 05.0 5.0"
$ns_ at 0.1 "$node_(6) set dest 305.0 05.0 5.0"
$ns_ at 0.1 "$node_(7) set dest 405.0 05.0 5.0"
$ns_ at 0.1 "$node_(8) set dest 505.0 05.0 5.0"

$ns_ at 0.1 "$node_(9) set dest 55.0 78.0 5.0"
$ns_ at 0.1 "$node_(10) set dest 155.0 78.0 5.0"
$ns_ at 0.1 "$node_(11) set dest 255.0 78.0 5.0"
$ns_ at 0.1 "$node_(12) set dest 355.0 78.0 5.0"
$ns_ at 0.1 "$node_(13) set dest 455.0 78.0 5.0"

```

\$ns\_ at 0.1 "\$node\_(14) setdest 05.0 151.0 5.0"  
\$ns\_ at 0.1 "\$node\_(15) setdest 105.0 151.0 5.0"  
\$ns\_ at 0.1 "\$node\_(16) setdest 205.0 151.0 5.0"  
\$ns\_ at 0.1 "\$node\_(17) setdest 305.0 151.0 5.0"  
\$ns\_ at 0.1 "\$node\_(18) setdest 405.0 151.0 5.0"  
\$ns\_ at 0.1 "\$node\_(19) setdest 505.0 151.0 5.0"

\$ns\_ at 0.1 "\$node\_(20) setdest 55.0 224.0 5.0"  
\$ns\_ at 0.1 "\$node\_(21) setdest 155.0 224.0 5.0"  
\$ns\_ at 0.1 "\$node\_(22) setdest 255.0 224.0 5.0"  
\$ns\_ at 0.1 "\$node\_(23) setdest 355.0 224.0 5.0"  
\$ns\_ at 0.1 "\$node\_(24) setdest 455.0 224.0 5.0"

\$ns\_ at 0.1 "\$node\_(25) setdest 05.0 297.0 5.0"  
\$ns\_ at 0.1 "\$node\_(26) setdest 105.0 297.0 5.0"  
\$ns\_ at 0.1 "\$node\_(27) setdest 205.0 297.0 5.0"  
\$ns\_ at 0.1 "\$node\_(28) setdest 305.0 297.0 5.0"  
\$ns\_ at 0.1 "\$node\_(29) setdest 405.0 297.0 5.0"  
\$ns\_ at 0.1 "\$node\_(30) setdest 505.0 297.0 5.0"

\$ns\_ at 0.1 "\$node\_(31) setdest 55.0 370.0 5.0"  
\$ns\_ at 0.1 "\$node\_(32) setdest 155.0 370.0 5.0"  
\$ns\_ at 0.1 "\$node\_(33) setdest 255.0 370.0 5.0"  
\$ns\_ at 0.1 "\$node\_(34) setdest 355.0 370.0 5.0"  
\$ns\_ at 0.1 "\$node\_(35) setdest 455.0 370.0 5.0"

\$ns\_ at 0.1 "\$node\_(36) setdest 05.0 443.0 5.0"  
\$ns\_ at 0.1 "\$node\_(37) setdest 105.0 443.0 5.0"  
\$ns\_ at 0.1 "\$node\_(38) setdest 205.0 443.0 5.0"  
\$ns\_ at 0.1 "\$node\_(39) setdest 305.0 443.0 5.0"  
\$ns\_ at 0.1 "\$node\_(40) setdest 405.0 443.0 5.0"  
\$ns\_ at 0.1 "\$node\_(41) setdest 505.0 443.0 5.0"

\$ns\_ at 0.1 "\$node\_(42) setdest 55.0 516.0 5.0"  
\$ns\_ at 0.1 "\$node\_(43) setdest 155.0 516.0 5.0"  
\$ns\_ at 0.1 "\$node\_(44) setdest 255.0 516.0 5.0"  
\$ns\_ at 0.1 "\$node\_(45) setdest 355.0 516.0 5.0"  
\$ns\_ at 0.1 "\$node\_(46) setdest 455.0 516.0 5.0"

\$ns\_ at 0.1 "\$node\_(47) setdest 05.0 589.0 5.0"  
\$ns\_ at 0.1 "\$node\_(48) setdest 155.0 589.0 5.0"  
\$ns\_ at 0.1 "\$node\_(49) setdest 355.0 589.0 5.0"  
\$ns\_ at 0.1 "\$node\_(50) setdest 505.0 589.0 5.0"

\$ns\_ at 0.1 "\$node\_(51) setdest 1395.0 589.0 5.0"  
\$ns\_ at 0.1 "\$node\_(52) setdest 1295.0 589.0 5.0"  
\$ns\_ at 0.1 "\$node\_(53) setdest 1145.0 589.0 5.0"  
\$ns\_ at 0.1 "\$node\_(54) setdest 995.0 589.0 5.0"  
\$ns\_ at 0.1 "\$node\_(55) setdest 895.0 589.0 5.0"

\$ns\_ at 0.1 "\$node\_(56) setdest 1345.0 516.0 5.0"  
\$ns\_ at 0.1 "\$node\_(57) setdest 1245.0 516.0 5.0"  
\$ns\_ at 0.1 "\$node\_(58) setdest 1145.0 516.0 5.0"

```

$ns_ at 0.1 "$node_(59) setdest 1045.0 516.0 5.0"
$ns_ at 0.1 "$node_(60) setdest 945.0 516.0 5.0"

$ns_ at 0.1 "$node_(61) setdest 1395.0 443.0 5.0"
$ns_ at 0.1 "$node_(62) setdest 1295.0 443.0 5.0"
$ns_ at 0.1 "$node_(63) setdest 1195.0 443.0 5.0"
$ns_ at 0.1 "$node_(64) setdest 1095.0 443.0 5.0"
$ns_ at 0.1 "$node_(65) setdest 995.0 443.0 5.0"
$ns_ at 0.1 "$node_(66) setdest 895.0 443.0 5.0"

$ns_ at 0.1 "$node_(67) setdest 1345.0 370.0 5.0"
$ns_ at 0.1 "$node_(68) setdest 1245.0 370.0 5.0"
$ns_ at 0.1 "$node_(69) setdest 1145.0 370.0 5.0"
$ns_ at 0.1 "$node_(70) setdest 1045.0 370.0 5.0"
$ns_ at 0.1 "$node_(71) setdest 945.0 370.0 5.0"

$ns_ at 0.1 "$node_(72) setdest 1395.0 297.0 5.0"
$ns_ at 0.1 "$node_(73) setdest 1295.0 297.0 5.0"
$ns_ at 0.1 "$node_(74) setdest 1195.0 297.0 5.0"
$ns_ at 0.1 "$node_(75) setdest 1095.0 297.0 5.0"
$ns_ at 0.1 "$node_(76) setdest 995.0 297.0 5.0"
$ns_ at 0.1 "$node_(77) setdest 895.0 297.0 5.0"

$ns_ at 0.1 "$node_(78) setdest 1345.0 224.0 5.0"
$ns_ at 0.1 "$node_(79) setdest 1245.0 224.0 5.0"
$ns_ at 0.1 "$node_(80) setdest 1145.0 224.0 5.0"
$ns_ at 0.1 "$node_(81) setdest 1045.0 224.0 5.0"
$ns_ at 0.1 "$node_(82) setdest 945.0 224.0 5.0"

$ns_ at 0.1 "$node_(83) setdest 1395.0 151.0 5.0"
$ns_ at 0.1 "$node_(84) setdest 1295.0 151.0 5.0"
$ns_ at 0.1 "$node_(85) setdest 1195.0 151.0 5.0"
$ns_ at 0.1 "$node_(86) setdest 1095.0 151.0 5.0"
$ns_ at 0.1 "$node_(87) setdest 995.0 151.0 5.0"
$ns_ at 0.1 "$node_(88) setdest 895.0 151.0 5.0"

$ns_ at 0.1 "$node_(89) setdest 1345.0 78.0 5.0"
$ns_ at 0.1 "$node_(90) setdest 1245.0 78.0 5.0"
$ns_ at 0.1 "$node_(91) setdest 1145.0 78.0 5.0"
$ns_ at 0.1 "$node_(92) setdest 1045.0 78.0 5.0"
$ns_ at 0.1 "$node_(93) setdest 945.0 78.0 5.0"

$ns_ at 0.1 "$node_(94) setdest 1395.0 05.0 5.0"
$ns_ at 0.1 "$node_(95) setdest 1295.0 05.0 5.0"
$ns_ at 0.1 "$node_(96) setdest 1195.0 05.0 5.0"
$ns_ at 0.1 "$node_(97) setdest 1095.0 05.0 5.0"
$ns_ at 0.1 "$node_(98) setdest 995.0 05.0 5.0"
$ns_ at 0.1 "$node_(99) setdest 895.0 05.0 5.0"

# Movilidad de los nodos a la mitad de la simulacion
$ns_ at 280.0 "$node_(3) setdest 505.0 589.0 5.0"
$ns_ at 280.0 "$node_(98) setdest 1195.0 05.0 5.0"
$ns_ at 280.0 "$node_(15) setdest 505.0 297.0 5.0"

```

```

$ns_ at 280.0 "$node_(60) setdest 995.0 443.0 5.0"
$ns_ at 280.0 "$node_(19) setdest 55.0 224.0 5.0"
$ns_ at 280.0 "$node_(53) setdest 895.0 589.0 5.0"
$ns_ at 280.0 "$node_(23) setdest 05.0 297.0 5.0"
$ns_ at 280.0 "$node_(68) setdest 1045.0 370.05.0"
$ns_ at 280.0 "$node_(44) setdest 55.0 370.0 5.0"
$ns_ at 280.0 "$node_(52) setdest 895.0 151.0 5.0"
$ns_ at 280.0 "$node_(34) setdest 355.0 589.0 5.0"
$ns_ at 280.0 "$node_(78) setdest 1045.0 78.0 5.0"
$ns_ at 280.0 "$node_(41) setdest 105.0 443.0 5.0"
$ns_ at 280.0 "$node_(63) setdest 945.0 224.0 5.0"
$ns_ at 280.0 "$node_(48) setdest 405.0 443.0 5.0"
$ns_ at 280.0 "$node_(81) setdest 895.0 297.0 5.0"
$ns_ at 280.0 "$node_(38) setdest 305.0 443.0 5.0"
$ns_ at 280.0 "$node_(93) setdest 1295.0 297.05.0"
$ns_ at 280.0 "$node_(21) setdest 255.0 78.0 5.0"
$ns_ at 280.0 "$node_(99) setdest 895.0 443.0 5.0"

```

```

$ns_ at 280.0 "$node_(50) setdest 05.0 05.0 5.0"
$ns_ at 280.0 "$node_(96) setdest 995.0 05.0 5.0"
$ns_ at 280.0 "$node_(30) setdest 105.0 151.0 5.0"
$ns_ at 280.0 "$node_(65) setdest 945.0 516.0 5.0"
$ns_ at 280.0 "$node_(20) setdest 505.0 151.0 5.0"
$ns_ at 280.0 "$node_(55) setdest 1145.0 589.0 5.0"
$ns_ at 280.0 "$node_(25) setdest 355.0 224.0 5.0"
$ns_ at 280.0 "$node_(70) setdest 1245.0 370.05.0"
$ns_ at 280.0 "$node_(31) setdest 255.0 516.0 5.0"
$ns_ at 280.0 "$node_(88) setdest 1295.0 589.05.0"
$ns_ at 280.0 "$node_(49) setdest 355.0 370.0 5.0"
$ns_ at 280.0 "$node_(92) setdest 1345.0 224.05.0"
$ns_ at 280.0 "$node_(37) setdest 505.0 443.0 5.0"
$ns_ at 280.0 "$node_(82) setdest 1195.0 443.05.0"
$ns_ at 280.0 "$node_(40) setdest 155.0 589.0 5.0"
$ns_ at 280.0 "$node_(77) setdest 1045.0 224.05.0"
$ns_ at 280.0 "$node_(39) setdest 205.0 443.0 5.0"
$ns_ at 280.0 "$node_(73) setdest 945.0 78.0 5.0"
$ns_ at 280.0 "$node_(11) setdest 155.0 224.0 5.0"
$ns_ at 280.0 "$node_(66) setdest 895.0 05.0 5.0"

```

```

# Configuración de nombres y colores para los nodos
$ns_ at 1.0 "$node_(3) color Blue"
$ns_ at 1.0 "$node_(3) label Source1"
$ns_ at 1.0 "$node_(98) color blue"
$ns_ at 1.0 "$node_(98) label Destination1"
$ns_ at 1.0 "$node_(15) color brown"
$ns_ at 1.0 "$node_(15) label source2"
$ns_ at 1.0 "$node_(60) color Brown"
$ns_ at 1.0 "$node_(60) label Destination2"
$ns_ at 1.0 "$node_(19) color Red"
$ns_ at 1.0 "$node_(19) label Source3"
$ns_ at 1.0 "$node_(53) color red"
$ns_ at 1.0 "$node_(53) label Destination3"
$ns_ at 1.0 "$node_(23) color chocolate"

```

```

$ns_ at 1.0 "$node_(23) label source4"
$ns_ at 1.0 "$node_(68) color Chocolate"
$ns_ at 1.0 "$node_(68) label Destination4"
$ns_ at 1.0 "$node_(44) color tan"
$ns_ at 1.0 "$node_(44) label Source5"
$ns_ at 1.0 "$node_(52) color Tan"
$ns_ at 1.0 "$node_(52) label Destination5"
$ns_ at 1.0 "$node_(34) color pink"
$ns_ at 1.0 "$node_(34) label Source6"
$ns_ at 1.0 "$node_(78) color Pink"
$ns_ at 1.0 "$node_(78) label Destination6"
$ns_ at 1.0 "$node_(41) color orange"
$ns_ at 1.0 "$node_(41) label source7"
$ns_ at 1.0 "$node_(63) color Orange"
$ns_ at 1.0 "$node_(63) label Destination7"
$ns_ at 1.0 "$node_(48) color Yellow"
$ns_ at 1.0 "$node_(48) label Source8"
$ns_ at 1.0 "$node_(81) color yellow"
$ns_ at 1.0 "$node_(81) label Destination8"
$ns_ at 1.0 "$node_(38) color green"
$ns_ at 1.0 "$node_(38) label source9"
$ns_ at 1.0 "$node_(93) color Green"
$ns_ at 1.0 "$node_(93) label Destination9"
$ns_ at 1.0 "$node_(21) color purple"
$ns_ at 1.0 "$node_(21) label Source10"
$ns_ at 1.0 "$node_(99) color Purple"
$ns_ at 1.0 "$node_(99) label Destination10"

$ns_ at 1.0 "$node_(1) label Edge1"
$ns_ at 1.0 "$node_(2) label Edge2"
$ns_ at 1.0 "$node_(0) label Core"

# Configuración de los routers edge y core
$ns_ simplex-link $node_(1) $node_(0) 10Mb 5ms dsRED/edge
$ns_ simplex-link $node_(0) $node_(1) 10Mb 5ms dsRED/core
$ns_ simplex-link $node_(0) $node_(2) 10Mb 5ms dsRED/core
$ns_ simplex-link $node_(2) $node_(0) 10Mb 5ms dsRED/edge

$ns_ duplex-link-op $node_(1) $node_(0) orientright
$ns_ duplex-link-op $node_(0) $node_(2) orientright

set qE1C [[$ns_ link $node_(1) $node_(0)] queue]
set qE2C [[$ns_ link $node_(2) $node_(0)] queue]
set qCE1 [[$ns_ link $node_(0) $node_(1)] queue]
set qCE2 [[$ns_ link $node_(0) $node_(2)] queue]

# Establecer parámetros de DS RED desde Edge1 a Core
$qE1C meanPktSize $packetSize
$qE1C set numQueues_ 1
$qE1C set NumPrec 2
$qE1C addPolicyEntry [$node_(3) id] [$node_(98) id] TokenBucket 10 $cir0 $cbs0
$qE1C addPolicyEntry [$node_(15) id] [$node_(60) id] TokenBucket 10 $cir1 $cbs1
$qE1C addPolicyEntry [$node_(19) id] [$node_(53) id] TokenBucket 10 $cir2 $cbs2

```

```

SqE1C addPolicyEntry [$node_(23) id] [$node_(68) id] TokenBucket 10 $cir3 $cbs3
SqE1C addPolicyEntry [$node_(44) id] [$node_(52) id] TokenBucket 10 $cir4 $cbs4
SqE1C addPolicyEntry [$node_(34) id] [$node_(78) id] TokenBucket 10 $cir5 $cbs5
SqE1C addPolicyEntry [$node_(41) id] [$node_(63) id] TokenBucket 10 $cir6 $cbs6
SqE1C addPolicyEntry [$node_(48) id] [$node_(81) id] TokenBucket 10 $cir7 $cbs7
SqE1C addPolicyEntry [$node_(38) id] [$node_(93) id] TokenBucket 10 $cir8 $cbs8
SqE1C addPolicyEntry [$node_(21) id] [$node_(99) id] TokenBucket 10 $cir9 $cbs9
SqE1C addPolicerEntry TokenBucket 10 11
SqE1C addPolicerEntry TokenBucket 20 21
SqE1C addPolicerEntry TokenBucket 30 31
SqE1C addPolicerEntry TokenBucket 40 41
SqE1C addPolicerEntry TokenBucket 50 51
SqE1C addPolicerEntry TokenBucket 60 61
SqE1C addPolicerEntry TokenBucket 70 71
SqE1C addPolicerEntry TokenBucket 80 81
SqE1C addPolicerEntry TokenBucket 90 91
SqE1C addPolicerEntry TokenBucket 100 101
SqE1C addPHBEntry 1 0 0
SqE1C addPHBEntry 11 0 1
SqE1C addPHBEntry 2 0 0
SqE1C addPHBEntry 21 0 1
SqE1C addPHBEntry 3 0 0
SqE1C addPHBEntry 31 0 1
SqE1C addPHBEntry 4 0 0
SqE1C addPHBEntry 41 0 1
SqE1C addPHBEntry 5 0 0
SqE1C addPHBEntry 51 0 1
SqE1C addPHBEntry 6 0 0
SqE1C addPHBEntry 61 0 1
SqE1C addPHBEntry 7 0 0
SqE1C addPHBEntry 71 0 1
SqE1C addPHBEntry 8 0 0
SqE1C addPHBEntry 81 0 1
SqE1C addPHBEntry 9 0 0
SqE1C addPHBEntry 91 0 1
SqE1C addPHBEntry 10 0 0
SqE1C addPHBEntry 101 0 1
SqE1C configQ 0 0 20 40 0.02
SqE1C configQ 0 1 10 20 0.10

```

# Establecer parámetros de DS RED desde Edge2 a Core

```

SqE2C meanPktSize $packetSize
SqE2C set numQueues_ 1
SqE2C setNumPrec 2
SqE2C addPolicyEntry [$node_(98) id] [$node_(3) id] TokenBucket 10 $cir0 $cbs0
SqE2C addPolicyEntry [$node_(60) id] [$node_(15) id] TokenBucket 10 $cir1 $cbs1
SqE2C addPolicyEntry [$node_(53) id] [$node_(19) id] TokenBucket 10 $cir2 $cbs2
SqE2C addPolicyEntry [$node_(68) id] [$node_(23) id] TokenBucket 10 $cir3 $cbs3
SqE2C addPolicyEntry [$node_(52) id] [$node_(44) id] TokenBucket 10 $cir4 $cbs4
SqE2C addPolicyEntry [$node_(78) id] [$node_(34) id] TokenBucket 10 $cir5 $cbs5
SqE2C addPolicyEntry [$node_(63) id] [$node_(41) id] TokenBucket 10 $cir6 $cbs6
SqE2C addPolicyEntry [$node_(81) id] [$node_(48) id] TokenBucket 10 $cir7 $cbs7
SqE2C addPolicyEntry [$node_(93) id] [$node_(38) id] TokenBucket 10 $cir8 $cbs8

```

```

SqE2C addPolicyEntry [$node_(99) id] [$node_(21) id] TokenBucket 10 Scir9 Scbs9
SqE2C addPolicerEntry TokenBucket 10 11
SqE2C addPolicerEntry TokenBucket 20 21
SqE2C addPolicerEntry TokenBucket 30 31
SqE2C addPolicerEntry TokenBucket 40 41
SqE2C addPolicerEntry TokenBucket 50 51
SqE2C addPolicerEntry TokenBucket 60 61
SqE2C addPolicerEntry TokenBucket 70 71
SqE2C addPolicerEntry TokenBucket 80 81
SqE2C addPolicerEntry TokenBucket 90 91
SqE2C addPolicerEntry TokenBucket 100 101
SqE2C addPHBEntry 100 0
SqE2C addPHBEntry 110 1
SqE2C addPHBEntry 200 0
SqE2C addPHBEntry 210 1
SqE2C addPHBEntry 300 0
SqE2C addPHBEntry 310 1
SqE2C addPHBEntry 400 0
SqE2C addPHBEntry 410 1
SqE2C addPHBEntry 500 0
SqE2C addPHBEntry 510 1
SqE2C addPHBEntry 600 0
SqE2C addPHBEntry 610 1
SqE2C addPHBEntry 700 0
SqE2C addPHBEntry 710 1
SqE2C addPHBEntry 800 0
SqE2C addPHBEntry 810 1
SqE2C addPHBEntry 900 0
SqE2C addPHBEntry 910 1
SqE2C addPHBEntry 1000 0
SqE2C addPHBEntry 1010 1
SqE2C configQ 0 0 20 40 0.02
SqE2C configQ 0 1 10 20 0.10

```

# Establecer parámetros de DS RED desde Core a Edge1

```

SqCE1 meanPktSize $packetSize
SqCE1 set numQueues_ 1
SqCE1 setNumPrec 2
SqCE1 addPHBEntry 100 0
SqCE1 addPHBEntry 110 1
SqCE1 addPHBEntry 200 0
SqCE1 addPHBEntry 210 1
SqCE1 addPHBEntry 300 0
SqCE1 addPHBEntry 310 1
SqCE1 addPHBEntry 400 0
SqCE1 addPHBEntry 410 1
SqCE1 addPHBEntry 500 0
SqCE1 addPHBEntry 510 1
SqCE1 addPHBEntry 600 0
SqCE1 addPHBEntry 610 1
SqCE1 addPHBEntry 700 0
SqCE1 addPHBEntry 710 1
SqCE1 addPHBEntry 800 0

```

```

SqCE1 addPHBEntry 8 1 0 1
SqCE1 addPHBEntry 9 0 0 0
SqCE1 addPHBEntry 9 1 0 1
SqCE1 addPHBEntry 10 0 0 0
SqCE1 addPHBEntry 10 1 0 1
SqCE1 configQ 0 0 20 40 0.02
SqCE1 configQ 0 1 10 20 0.10

# Establecer parámetros de DS RED desde Core a Edge2
SqCE2 meanPktSize $packetSize
SqCE2 set numQueues_ 1
SqCE2 setNumPrec 2
SqCE2 addPHBEntry 10 0 0
SqCE2 addPHBEntry 11 0 1
SqCE2 addPHBEntry 20 0 0
SqCE2 addPHBEntry 21 0 1
SqCE2 addPHBEntry 30 0 0
SqCE2 addPHBEntry 31 0 1
SqCE2 addPHBEntry 40 0 0
SqCE2 addPHBEntry 41 0 1
SqCE2 addPHBEntry 50 0 0
SqCE2 addPHBEntry 51 0 1
SqCE2 addPHBEntry 60 0 0
SqCE2 addPHBEntry 61 0 1
SqCE2 addPHBEntry 70 0 0
SqCE2 addPHBEntry 71 0 1
SqCE2 addPHBEntry 80 0 0
SqCE2 addPHBEntry 81 0 1
SqCE2 addPHBEntry 90 0 0
SqCE2 addPHBEntry 91 0 1
SqCE2 addPHBEntry 100 0 0
SqCE2 addPHBEntry 101 0 1
SqCE2 configQ 0 0 20 40 0.02
SqCE2 configQ 0 1 10 20 0.10

# Configuración de conexiones con tráfico CBR desde la fuente hacia el destino
set udp0 [new Agent/UDP]
$ns_ attach-agent $node_(3) $udp0
set cbr0 [new Application/Traffic/CBR]
$scbr0 attach-agent $udp0
$scbr0 set packet_size_ $packetSize
$udp0 set packetSize_ $packetSize
$scbr0 set rate_ $rate0
set null0 [new Agent/Null]
$ns_ attach-agent $node_(98) $null0
$ns_ connect $udp0 $null0

# Configuración de conexiones con tráfico CBR desde la fuente hacia el destino
set udp1 [new Agent/UDP]
$ns_ attach-agent $node_(15) $udp1
set cbr1 [new Application/Traffic/CBR]
$scbr1 attach-agent $udp1
$scbr1 set packet_size_ $packetSize

```

```

$udp1 set packetSize_ $packetSize
$scbr1 set rate_ $rate1
set null1 [new Agent/Null]
$ns_ attach-agent $node_(60) $null1
$ns_ connect $udp1 $null1

# Configuración de conexiones con tráfico CBR desde la fuente hacia el destino
set udp2 [new Agent/UDP]
$ns_ attach-agent $node_(19) $udp2
set cbr2 [new Application/Traffic/CBR]
$scbr2 attach-agent $udp2
$scbr2 set packet_size_ $packetSize
$udp2 set packetSize_ $packetSize
$scbr2 set rate_ $rate2
set null2 [new Agent/Null]
$ns_ attach-agent $node_(53) $null2
$ns_ connect $udp2 $null2

# Configuración de conexiones con tráfico CBR desde la fuente hacia el destino
set udp3 [new Agent/UDP]
$ns_ attach-agent $node_(23) $udp3
set cbr3 [new Application/Traffic/CBR]
$scbr3 attach-agent $udp3
$scbr3 set packet_size_ $packetSize
$udp3 set packetSize_ $packetSize
$scbr3 set rate_ $rate3
set null3 [new Agent/Null]
$ns_ attach-agent $node_(68) $null3
$ns_ connect $udp3 $null3

# Configuración de conexiones con tráfico CBR desde la fuente hacia el destino
set udp4 [new Agent/UDP]
$ns_ attach-agent $node_(44) $udp4
set cbr4 [new Application/Traffic/CBR]
$scbr4 attach-agent $udp4
$scbr4 set packet_size_ $packetSize
$udp4 set packetSize_ $packetSize
$scbr4 set rate_ $rate4
set null4 [new Agent/Null]
$ns_ attach-agent $node_(52) $null4
$ns_ connect $udp4 $null4

# Configuración de conexiones con tráfico CBR desde la fuente hacia el destino
set udp5 [new Agent/UDP]
$ns_ attach-agent $node_(34) $udp5
set cbr5 [new Application/Traffic/CBR]
$scbr5 attach-agent $udp5
$scbr5 set packet_size_ $packetSize
$udp5 set packetSize_ $packetSize
$scbr5 set rate_ $rate5
set null5 [new Agent/Null]
$ns_ attach-agent $node_(78) $null5
$ns_ connect $udp5 $null5

```

```

# Configuración de conexiones con tráfico CBR desde la fuente hacia el destino
set udp6 [new Agent/UDP]
Sns_ attach-agent $node_(41) $udp6
set cbr6 [new Application/Traffic/CBR]
Scbr6 attach-agent $udp6
Scbr6 set packet_size_ $packetSize
Sudp6 set packetSize_ $packetSize
Scbr6 set rate_ $rate6
set null6 [new Agent/Null]
Sns_ attach-agent $node_(63) $null6
Sns_ connect $udp6 $null6

# Configuración de conexiones con tráfico CBR desde la fuente hacia el destino
set udp7 [new Agent/UDP]
Sns_ attach-agent $node_(48) $udp7
set cbr7 [new Application/Traffic/CBR]
Scbr7 attach-agent $udp7
Scbr7 set packet_size_ $packetSize
Sudp7 set packetSize_ $packetSize
Scbr7 set rate_ $rate7
set null7 [new Agent/Null]
Sns_ attach-agent $node_(81) $null7
Sns_ connect $udp7 $null7

# Configuración de conexiones con tráfico CBR desde la fuente hacia el destino
set udp8 [new Agent/UDP]
Sns_ attach-agent $node_(38) $udp8
set cbr8 [new Application/Traffic/CBR]
Scbr8 attach-agent $udp8
Scbr8 set packet_size_ $packetSize
Sudp8 set packetSize_ $packetSize
Scbr8 set rate_ $rate8
set null8 [new Agent/Null]
Sns_ attach-agent $node_(93) $null8
Sns_ connect $udp8 $null8

# Configuración de conexiones con tráfico CBR desde la fuente hacia el destino
set udp9 [new Agent/UDP]
Sns_ attach-agent $node_(21) $udp9
set cbr9 [new Application/Traffic/CBR]
Scbr9 attach-agent $udp9
Scbr9 set packet_size_ $packetSize
Sudp9 set packetSize_ $packetSize
Scbr9 set rate_ $rate9
set null9 [new Agent/Null]
Sns_ attach-agent $node_(99) $null9
Sns_ connect $udp9 $null9

# Cuando termina la simulación
for {set i 0} {$i < $val(nn)} {incr i} {
    Sns_ at 500.0 "$node_($i) reset";
}

```

```
$ns_ at 500.0 "stop"  
$ns_ at 500.01 "puts \"NS EXITING...\"; $ns_ halt"
```

```
proc stop {} {  
    global ns_ tracefile namtrace  
    $ns_ flush-trace  
    close $tracefile  
    close $namtrace  
    exec nam out1.2.nam &  
    exit 0  
}
```

```
$qE1C printPolicyTable  
$qE1C printPolicerTable  
$qE1C printPolicyTable  
$qE1C printPolicerTable  
$qE1C printPolicyTable  
$qE1C printPolicyTable  
$qE1C printPolicerTable  
$qE1C printPolicyTable  
$qE1C printPolicerTable  
$qE1C printPolicyTable
```

```
$ns_ at 225.0 "$cbr0 start"  
$ns_ at 225.0 "$cbr1 start"  
$ns_ at 225.0 "$cbr2 start"  
$ns_ at 225.0 "$cbr3 start"  
$ns_ at 225.0 "$cbr4 start"  
$ns_ at 225.0 "$cbr5 start"  
$ns_ at 225.0 "$cbr6 start"  
$ns_ at 225.0 "$cbr7 start"  
$ns_ at 225.0 "$cbr8 start"  
$ns_ at 225.0 "$cbr9 start"  
$ns_ at $stestTime "$cbr0 stop"  
$ns_ at $stestTime "$cbr1 stop"  
$ns_ at $stestTime "$cbr2 stop"  
$ns_ at $stestTime "$cbr3 stop"  
$ns_ at $stestTime "$cbr4 stop"  
$ns_ at $stestTime "$cbr5 stop"  
$ns_ at $stestTime "$cbr6 stop"  
$ns_ at $stestTime "$cbr7 stop"  
$ns_ at $stestTime "$cbr8 stop"  
$ns_ at $stestTime "$cbr9 stop"  
$ns_ at [expr $stestTime + 1.0] "finish"
```

```
$ns_ run
```

## A.1. Resultados obtenidos con el TraceGrahp

Options		Network information	
Simulation information:			
Simulation length in seconds:	275	Simulation End2End delays in	
Number of	100	Minimal delay	0.008175761 (15,60,86221)
Number of sending nodes:	100	Maximal delay	27.98306021 (48,81,129177)
Number of receiving nodes:	24	Average	8.011947947
Number of generated packets:	97380		
Number of sent packets:	-41718		
Number of forwarded packets:	92570		
Number of dropped packets:	158014		
Number of lost packets:	0		
Minimal packet size:	32		
Maximal packet size:	1020		
Average packet size:	700.7032		
Number of sent bytes:	-87688680		
Number of forwarded bytes:	92317144		
Number of dropped bytes:	161065248		
Packets dropping nodes:	1 2 3 4 5 6 7 8 9 10 1		
Current node information:			
Number of generated packets:	381		
Number of sent packets:	381		
Number of forwarded packets:	1547		
Number of received	2225		
Number of dropped packets:	0		
Number of lost packets:	0		
Number of sent bytes:	18288		
Number of forwarded bytes:	1550612		
Number of received bytes:	1582956		
Number of dropped bytes:	0		
Minimal packet size:	32		
Maximal packet size:	1020		
Average packet size:	758.9347		

Figura A.1: Información del comportamiento de la red con traceGrahp

En la figura B.1, se muestra la información del comportamiento de toda la red con 10 conexiones, modelo DiffServ y protocolo AODV, este es un ejemplo de como se ha obtenido todos los datos para llevar a cabo la comparación tanto de protocolos como de modelos de calidad de servicio.