UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO-CAMPUS SUR

FACULTAD DE INGENIERÍA DE SISTEMAS

ANÁLISIS Y SIMULACIÓN DE LA CALIDAD DE SERVICIO (QoS) SOBRE REDES INALÁMBRICAS CON IPV6, PARA EQUIPOS DE VIDEOCONFERENCIA EN LA EMPRESA METROTEK ECUADOR S.A.

TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS

EDISON RICARDO SUNTAXI SANTAMARÍA

DIRECTOR: ING. JORGE LÓPEZ

Quito, Febrero 2013

DECLARACIÓN

Yo, Edison Ricardo Suntaxi Santamaría, declaro bajo juramento que el trabajo aquí descrito es de mi autoría, que no ha sido previamente presentada por ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mi derecho de propiedad intelectual correspondiente a este trabajo a la Universidad Politécnica Salesiana, según lo establecido por la Ley de Propiedad intelectual, por su reglamento y por la normativa institucional vigente.

_____

Edison Ricardo Suntaxi Santamaría

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por el estudiante Edison Ricardo Suntaxi Santamaría bajo mi dirección.

Ing. Jorge E. López L.

Director de Tesis

AGRADECIMIENTO

Agradezco a Dios por permitirme culminar una etapa más en mi vida, guiándome por el camino correcto, llenándome de paciencia, inteligencia, felicidad a lo largo de estos años de estudio.

En segundo lugar quiero agradecer a toda mi familia, en especial a mi mamá y me hermana, que han sido mi soporte y apoyo, durante todos mis años de estudio y el desarrollo del proyecto.

Finalmente quiero dar un gran agradecimiento al Ing. Jorge López, director de tesis, por brindarme su ayuda, paciencia durante la realización de este proyecto.

Ricardo Suntaxi

DEDICATORIA

El desarrollo de este proyecto se la dedicó a mis padres Alejandro y Beatriz, por haber creído en mí durante todos estos años de estudio, por inculcarme valores que siempre estarán presentes en mi vida, y por brindarme todo su amor.

A mis hermanos Alejandro, Mónica, Verónica, a mi cuñada Marianela, por brindarme todo su apoyo, cuando lo necesitaba.

A mis sobrinos Cinthya, María José, Diego y en especial a Elian, que ha sido mi amigo y compañero, en los momentos difíciles de mi vida.

RESUMEN

El crecimiento que han tenido las redes inalámbricas por su bajo costo y movilidad, además de la aparición del protocolo de Internet versión 6, han permitido el desarrollo de aplicaciones avanzadas, para la transmisión de video, voz y datos simultáneamente, convirtiendo así a las redes tradicionales en redes multimediales. El manejo de nuevos tipos de tráfico requieren un tratamiento especial, por lo cual, los administradores de red se han visto en la necesidad de aplicar técnicas de Calidad de Servicio que permitan a los usuarios administrar adecuadamente estos servicios, esto se logra mediante la utilización de los métodos que se encuentran en el estándar 802.11e. Entre las más utilizadas en la actualidad por disminuir el tiempo y costo en las comunicaciones entre dos o más estaciones es la videoconferencia que es el objetivo principal de este proyecto. Por este motivo las empresas, han visto la necesidad de instalar este tipo sistemas dentro de sus redes en especial en ambientes inalámbricos, ya que los usuarios acceden a la información desde cualquier lugar mediante el uso de dispositivos móviles.

A través de la herramienta de simulación network simulator, se creará un escenario inalámbrico, en el cual, sus estaciones transmitirán datos simulando una videoconferencia. Está investigación tiene como objetivo dar a conocer mediante el análisis de los resultados conseguidos de la simulación, que beneficios obtendrá la red al momento de aplicar técnicas de priorización de tráfico durante el período de transmisión

ÍNDICE DE CONTENIDOS

CAPITULO I

CAPITULO II

ANEXOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ABREVIATURAS

QoS.- Calidad de Servicio.

IEEE.- Instituto de Ingenieros Eléctricos y Electrónicos

VoIP.- Voz sobre IP.

Ad hoc.- Red inalámbrica descentralizada.

IETF.- Fuerza de Tareas de Ingeniería en Internet

EUI-64.- Extended Unique Identifier de 64 bits

DHCPv6.- Protocolo de Configuración Dinámica de Hosts para IPv6.

NAT-PT.- Traducción de Dirección de Red- Traducción de Protocolo.

SIT.- Transición Simple de Internet.

Wi-Fi: Fidelidad Inalámbrica

BSS.- Conjunto de Servicios Básicos.

BSA.- Área de Servicio Básico.

IBSS.- Conjunto Básico de Servicios Independientes.

EBSS.- Conjunto de Servicios Básicos Extendido.

DS.- Sistema de distribución.

ESS.- Conjunto de Servicio Extendido.

PMD.- Dependiente del Medio Físico.

PLCP.- Procedimiento de Convergencia de Capa Física.

FHSS.- Espectro Ensanchado por Salto de Frecuencia.

DSSS.- Espectro Ensanchado por Secuencia Directa.

LLC.- Control de Enlace Lógico.

MAC.- Control de Acceso al Medio.

CSMA/CA.- Acceso Múltiple con Escucha de Portadora y Detección de Colisión.

CSMA/CA.- Acceso Múltiple por Detección de Portadora con Detección de Colisiones

DCF.- Función de Coordinación Distribuida

PCF.- Función de Coordinación Puntual

RTS.- Solicitud de Envio

CTS.- Listo para Transmitir

IFS.- Intervalo entre Tramas

SSID.- Identificador de Conjunto de Servicios.

RSVP.- Protocolo de Reserva de Recursos.

WEP.- Privacidad Equivalente a Cableado

HCF.- Función de Coordinación Hibrida

EDCA.- Acceso al Canal Distribuido Mejorado

HCCA.- Acceso Controlado al Canal HCF

TXOP.- Oportunidad de Trasmisión

MSDU.- Unidad de Servicio de Datos de la Capa MAC

HC.- Coordinador Híbrido

AIFS.- Espacio entre Tramas Arbitrario.

CAPITULO I

1. INTRODUCCIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

Las redes, en especial las inalámbricas, han tenido un crecimiento exponencial y tecnológico durante los últimos años. Esto debido a que la información que se maneja llega de forma continua, brindando una amplia comodidad a las personas y facilitando la comunicación dentro de su entorno laboral.

Uno de los principales motivos por lo que las redes inalámbricas han tenido gran aceptación es su bajo costo; puesto que elimina todo el cableado Ethernet y las conexiones físicas entre los dispositivos. Lo que permite conectividad y movilidad a los usuarios de la red, de la misma manera. En consecuencia, esta sobredemanda devino en la escasez de direcciones IPV4. Debido a estas limitaciones de adecuación y funcionamiento en las aplicaciones actuales; se requirió la creación del protocolo IPV6.

Dentro de estos avances tecnológicos, aparece la videoconferencia; que para su funcionamiento necesita de equipos que internamente manejen la codificación de video y voz; además, de requerir una dirección IP. Esto presupone un gran ancho de banda para evitar la presencia de retardo o pérdida de paquetes durante la transmisión. Por esto, es frecuente que muchas empresas suspendan el suministro de Internet a los demás usuarios de la red,

para abastecer con toda la capacidad del ancho de banda a las videoconferencias.

El ofrecer QoS[1] a redes inalámbricas, se presenta como un gran desafío para todo administrador de red. Por este motivo, la IEEE[2] desarrolló el estándar 802.11e, el cual nos ofrece un conjunto de técnicas y métodos, para la aplicación de QoS; y, así disminuir los problemas que se pueden presentar en red inalámbrica.

Uno de los problemas prioritarios de las videoconferencias es el retardo en la recepción de imagen y sonido. Por ejemplo, al enviar un mail, un retardo de 30 segundos es imperceptible; sin embargo, en una aplicación de tiempo real, un retraso de tan solo 5 segundos se considera un fracaso. En este contexto, el desarrollo de esta tesis tiene como objetivo, dar a conocer los beneficios que se obtiene al aplicar QoS sobre redes durante la transmisión de datos.

1.2 JUSTIFICACIÓN DEL PROYECTO

Con las redes inalámbricas accedemos a la información sin necesidad de estar conectados físicamente a un lugar específico. El emplear QoS es esencial para el éxito de aplicaciones en tiempo real, en especial la videoconferencia que es el punto principal de este trabajo; aunque, se puede mencionar otras aplicaciones como VoIP[3] y la telemedicina, estas aplicaciones además de necesitar un gran ancho de banda, requieren un servicio diferenciado, para garantizar que en la transmisión de datos no exista pérdida de paquetes.

[1] **QoS Calidad de Servicio.-** es la capacidad de dar un buen servicio para ciertas aplicaciones tales como la transmisión de vídeo o voz.
[2] **IEEE.-** Instituto de Ingenieros Eléctricos y Electrónicos.
[3] **VoIP: Voz sobre IP.-** Conjunto de recursos que hacen posible que la señal de voz viaje a través de Internet empleando un protocolo IP.

En las redes actuales se manejan aplicaciones que trabajan en tiempo real como son: la telemedicina y la videoconferencia, las cuales presentan calidad de audio y video poco satisfactoria. En general, este tipo de aplicaciones transmiten información de manera continua, en la cual no debe existir la presencia de factores que alteren el flujo de datos, es por eso, que al aplicar QoS manejaremos prioridades sobre los diferentes tipos de tráfico que viajan a través de la red.

En este trabajo se plantea analizar el estándar IEEE 802.11e, que describe diferentes técnicas de QoS. A través de la simulación de una red inalámbrica IPv6, utilizaremos la priorización del tráfico, que es una de forma de aplicar QoS. Con los resultados obtenidos se observarán los beneficios de utilizar estos métodos durante la transmisión de información, por consiguiente, se evitaría perdida de paquetes y retardo al momento de realizar una videoconferencia.

1.3 OBJETIVOS

1.3.1 Objetivo General

Realizar un análisis y simulación de la Calidad de Servicio (QoS) aplicadas sobre redes inalámbricas IPV6, para equipos móviles de video conferencia para la empresa METROTEK ECUADOR S.A.

1.3.2 Objetivos Específicos

- Conocer el fundamento teórico de la Calidad de Servicio en el estándar IEEE 802.11e, con el fin de realizar una configuración que permita un funcionamiento óptimo de la redes wireless.

- Conocer y determinar los beneficios de implementar QoS en las redes inalámbricas.

- Desarrollar scripts para creación de un escenario de simulación, el cual constará de una red inalámbrica, donde se aplicará técnicas de calidad de servicio, durante el envío y recepción de paquetes entre los nodos.

- Analizar y comparar los resultados logrados de las pruebas realizadas con las redes inalámbricas, a una de ellas, se aplicó calidad de servicio durante el proceso de simulación en la red wireless.

- Demostrar con los datos obtenidos, la importancia de aplicar QoS sobre cualquier tipo de red, sea esta cableada o inalámbrica, al momento de utilizar aplicaciones de tiempo real como por ejemplo videoconferencia.

1.4 ALCANCE DEL PROYECTO

Para obtener conocimiento de IPV6 y todo lo referente a la Calidad de Servicio (QoS), se estudiará el estándar 802.11e, por lo cual, podremos determinar las normas que debemos cumplir para la creación de una red inalámbrica en el Simulador NS-2.

Para el desarrollo de este proyecto, se elaborará scripts, en los cuales se creará una red tipo ad-hoc[4] IPv6, que por ser un tipo red inalámbrica sus nodos estarán en movimiento, así, recrearemos situaciones de la vida real, es especial enfocados al uso de tablets o dispositivos móviles, en los cuales para realizar videoconferencia se instalaría un software específico.

El proceso de simulación se lo realizará en dos tipos de escenarios, los cuales se diferencian únicamente por la utilización de QoS en uno de ellos. Se trabajará especialmente con tráfico de video, como resultado se obtendrá datos como: perdida de paquetes, consumo de ancho de banda, tiempo de retardo.

Al concluir este proyecto y realizado el respectivo análisis de los resultados obtenidos de las simulaciones, la empresa METROTEK ECUADOR S.A., estará en la capacidad de ofrecer a sus clientes asesoría en lo referente a la utilización de las técnicas de QoS dentro de sus redes al momento de realizar no solo videoconferencia, si no también telefonía IP.

[4] **Ad-hoc: Red inalámbrica descentralizada.-** La red es ad hoc porque cada nodo está preparado para reenviar datos a los demás.

CAPITULO II

2. IPv6

2.1 INTRODUCCIÓN

El protocolo IPv4 en la última década comenzó a presentar cierto tipo de problemas; uno de los principales fue la escasez de direcciones IP, puesto que IPv4 ofrece un espacio para direccionamiento de 32 bits, y debido al crecimiento no solo de las redes comunes como por ejemplo las redes LAN, WAN, etc; sino que además se sumó las redes de telefonía celular, las cuales requieren de una IP fija, ya para el funcionamiento de cierto tipo de aplicaciones. Además al momento de crear el protocolo IPv4 no tomaron en cuenta los avances tecnológicos que hubo en la última década como son el tráfico de video y audio en tiempo real, además Ipve no ofrece mecanismos de seguridad con respecto a los datos transmitidos.

Desde los años 90 la IETF[5] tomando en cuenta los problemas que comenzaron a presentarse sobre todo el del agotamiento de direcciones IP, se crea un nuevo protocolo de internet conocido como IPv6 o también como *"IPng (Next Generation Internet Protocol)"*[6]. A IPv6 se lo considera como evolución del protocolo IPv4, ya que se mantienen el objetivo y conceptos generales del protocolo predecesor. Las principales características que presente el IPv6 son las siguientes:

[5] **IETF:** organismo que se encarga de la estandarización de los protocolos de Internet
[6] **Fuente:** http://www.rau.edu.uy/ipv6/queesipv6.htm

- **Mayor espacio para direcciones:** IPv6 ofrece un tamaño de dirección de 128 bits a diferencia de IPv4 que solamente ofrecía 32 bits para direccionamiento
- **Simplificación del formato de cabecera:** en IPv6 se simplifican los encabezados de los datagramas, solo se presentan 7 campos a diferencia de los 14 que tiene IPv4. Con esta mejora los routers procesan de manera más rápido el trato que deben recibir los paquetes, por ende los router no realizan la fragmentación
- **Autoconfiguración de los nodos:** los nodos se configuran automáticamente, es decir, se auto asignan una dirección IPv6 mediante un mecanismo conocido como "*stateless address configuration*".
- **Mejora compatibilidad para Calidad de Servicio (QoS) y Clase de Servicio (CoS).**- en el campo Flow Label se define como se controla y el trato que se debe dar a los paquetes dentro de una transmisión
- **Movilidad:** es una de las características más sobresalientes que presenta IPv6, permite que dispositivos móviles como celulares o tablets cambien de forma dinámica sus puntos de acceso.
- **Mayor Seguridad:** A través de IPsec, permite la autenticación y encriptación del propio protocolo, beneficiando a las aplicaciones.

2.2 CABECERA IPv6

La estructura de la cabecera IPv6 es la siguiente:

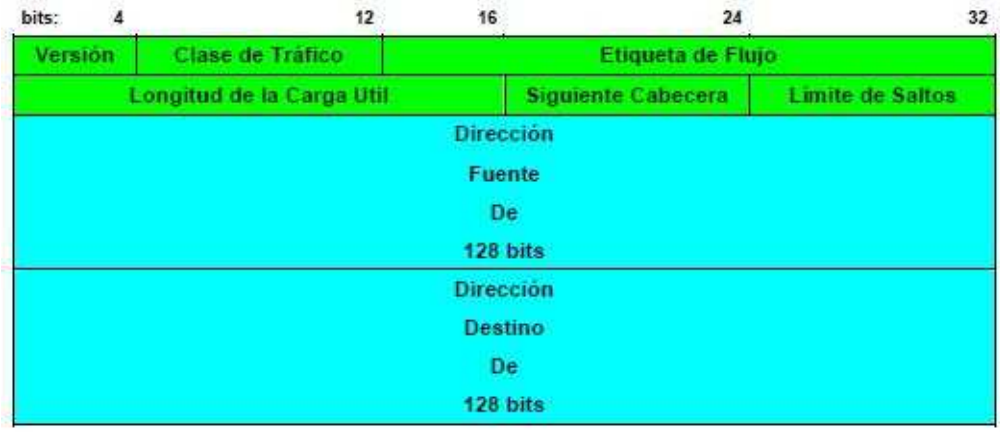

Figura 2.1: Formato Cabecera IPv6[7].

- ***Versión (4 bits):*** Número que indica la versión IP = 6
- ***Clase de Tráfico (8 bits):*** Especifica la clase o prioridad del paquete IPv6. Remplaza al campo Tipo de Servicio de IPv4.
- ***Etiqueta de Flujo (20 bits):*** Sirve para para identificar un flujo de paquetes, con lo cual los routers, pueden de manera más rápida el tipo de trato que se debe dar a cada paquete.
- ***Longitud del paquete (16 bits):*** Indica el tamaño total del paquete, en la cual se incluyen la cabecera y los datos, este campo es esencial puesto que en IPv6 existen campos opcionales que van en la cabecera.
- ***Siguiente Cabecera (8 bits):*** Indica a parte de la cabecera fija, cual es la siguiente cabecera en el paquete, o el protocolo de capa superior (TCP o UDP).
- ***Límite de Saltos (8 bits):*** Su función es muy similar al TTL de IPv4, indica el máximo número de saltos que puede realizar el paquete. Cuyo valor es de 8 bits el cual disminuye por cada vez que pasa por un router, si el valor llega a 0 este es descartado.
- ***Dirección de Origen (128 bits):*** Indica la dirección de origen del emisor de paquete.

[7] **Fuente:** Daniel Andrés, Protocolo IP6, 25/01/2012, disponible en: <http://protocoloip6.wikispaces.com>

- ***Dirección de Destino (128 bits):*** Indica la dirección de destino del paquete.

2.3 DIRECCIONAMIENTO IPv6

A las direcciones IPv6 se las representan, en 8 campos de 16 bits cada uno, los cuales se encuentran separados por dos puntos (:) como se muestra a continuación:

x:x:x:x:x:x:x:x.

A cada campo se lo representa con 4 caracteres hexadecimales (0-f), lo que genera una dirección IPv6 de la siguiente forma:

fe80::41f3:f4de:d278:7550

Las direcciones IPv6 son asignadas a las interfaces y no a los nodos; es por eso, si un host o nodo tiene varias interfaces el mismo tendrá varias direcciones unicast. Las direcciones de *broadcast* en IPv6 no existen, su función es sustituida por las direcciones *multicast*.

La máscara de red que maneja IPv4, en IPv6 es reemplazada por la longitud de prefijo o simplemente prefijo, que cumple una función similar, con este valor decimal se puede identificar la ruta de encaminado de un host.

Dirección IPV6/ longitud de prefijo

Dónde:

- **Dirección IPv6:** en la dirección de cada host
- **Longitud de prefijo:** indica los primeros bits desde la izquierda que identifican la red, mientras que los bits restantes identifican nuestro host.

2.3.1 Tipos y Formato de Direcciones IPv6

2.3.1.1 Direcciones Unicast IPv6

Es utilizada para identificar a una sola interfaz, es decir que si se envía un paquete a un host que también tiene una dirección unicast, el paquete es enviado al nodo que está identificado por esa dirección. Existen tres tipos de direcciones Unicast:

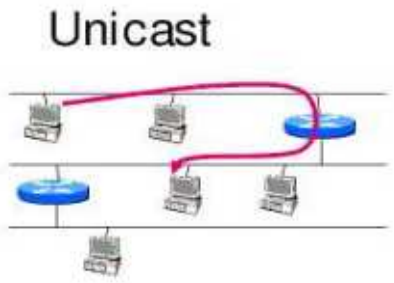

Figura 2.2: Dirección Unicast[8]

2.3.1.1.1 Direcciones Unicast Globales (Global Address) Ipv6

Utilizadas para el tráfico IPv6 a través de Internet, son similares a las direcciones públicas que manejamos en IPv4, son las únicas que se pueden rutear globalmente.

Figura 2.3: Estructura Dirección Unicast Global[9]

[8] **Fuente:** SÁNCHEZ Belén RAMÓN Luis, *Redes de Banda Ancha*, 25/01/2013, disponible en: <http://www.monografias.com/trabajos-pdf/redes-banda-ancha/redes-banda-ancha.pdf >

[9] **Fuente:** Franklin, IPV6 , 25/01/2013, disponible en: <http://modeloosi-franklin.blogspot.com/2010/11/ipv6.html>

- ***Prefijo:*** representa los 48 bits con lo cual los ISP's estructuran a la dirección de una manera jerarquica.
- ***ID de Subred:*** nos sirve como identificador de una subred dentro de una área. A diferencia del prefijo, la *id* de subred es estrucuturada por los administradores.
- ***ID interfaz:*** sirve para identificar a la interfaz de nuestro host, este identificador debe estar compuesto por 64 bits.

2.3.1.1.2 Direcciones Unicast Locales Únicas (Unique Local address) IPv6

Se asemeja a las direcciones privadas de IPv4, estas direcciones permiten que, dentro de una red compuesta por subredes los nodos se comuniquen, estas direcciones a diferencia de las unicast globales no tienen enrutamiento hacia Internet, a continuación se detalla su estructura:

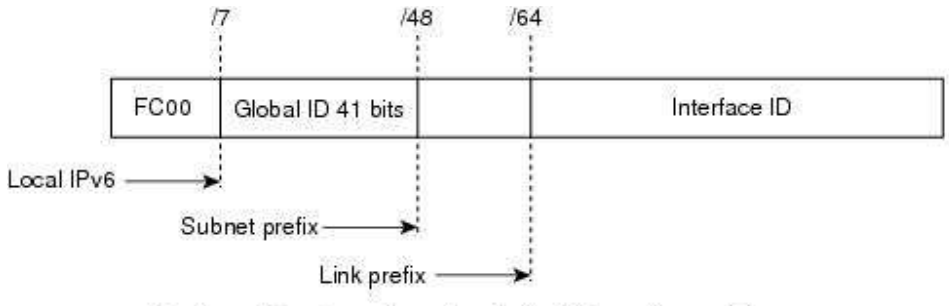

Figura 2.4: Estructura Dirección Unicast Local[10]

- ***Prefijo.-*** FEC0::/7 es el rango del prefijo en el que trabajan todas las direcciones unicast únicas locales.
- ***Global ID.-*** identificador usado para crear un prefijo global único.
- ***Sunet ID.-*** permite la identificación de subredes dentro de una red.
- ***Interface Identifier.-*** identifica a una interfaz dentro de una determinada subred.

[10] **Fuente:** CISCO, Configuring IPv6, 25/01/2013, disponible en: <http://www.cisco.com/en/US/docs/switches/datacenter/sw/4_2/nx-os/unicast/configuration/guide/l3_ipv6.html>

Estas direcciones sustituyeron a las Direcciones Unicast de Sitio Local.

2.3.1.1.3 Direcciones Unicast de Enlaces Locales (Link local address) IPv6

Son direcciones unicast que se configura de manera automática, al momento que un host se vincula a una red, el formato de una dirección de enlace local es el siguiente.

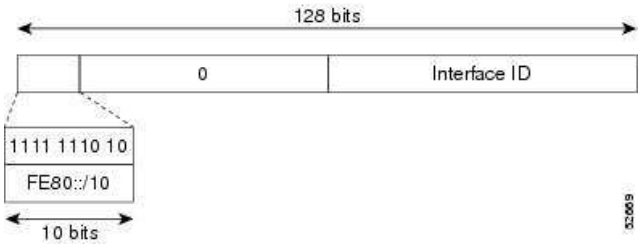

Figura 2.5: Estructura Dirección Unicast de Enlace Local[11]

Dónde:

- fe80 es el prefijo de enlace local
- Identificador del interfaz que se obtienen a partir de sus direcciones MAC

En la figura 2.6 se puede observar el procedimiento para obtener el identificador de interfaz, el cual representa un nuevo estándar para el direccionamiento de las interfaces de red. El identificador está compuesto por 64 bits que están distribuidos de la siguiente manera: 24 bits para el fabricante y los 40 restantes representan el número de serie de las interfaces.

[11] **Fuente:** CISCO, Op. Cit.

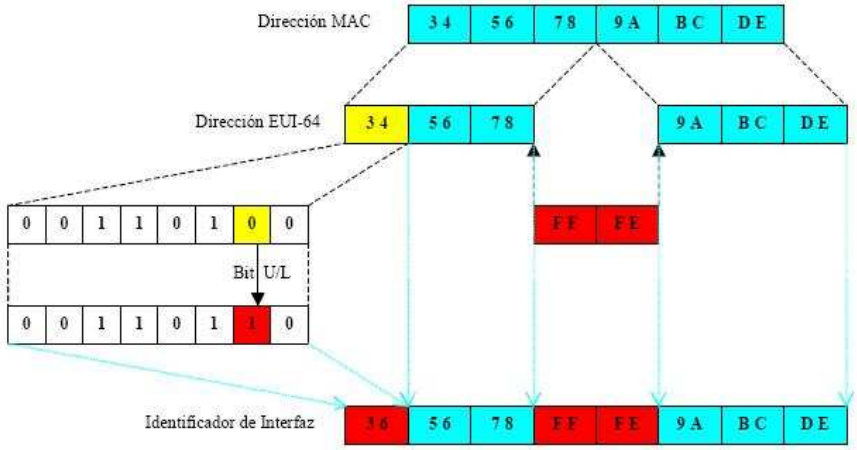

Figura 2.6: Identificador de Interface EUI-64[12]

El identificador de interfaz es utilizado, para verificar que las interfaces dentro de un enlace sean únicas.

2.3.1.2 Direcciones Anycast IPv6

Son aquellas direcciones que identifican a un conjunto de interfaces de uno o varios nodos, es decir, los paquetes enviados a una dirección anycast, son reenviados a través de las diferentes rutas a la dirección anycast más cercana, el router de la red debe conocer la dirección y la distancia de la interfaz.

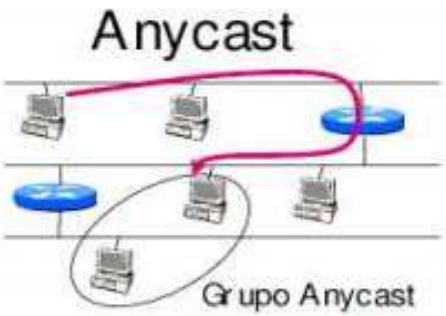

Figura 2.7: Dirección Anycast[13]

[12] **Fuente:** PÉREZ Ricardo, IPv6: Teoría y Práctica (Parte 1), 25/01/2013, disponible en: <http://ricardoperez.ingenieriaupoli.net/2012/10/28/ipv6-teoria-y-practica-parte-1/>
[13] **Fuente:** SÁNCHEZ Belén RAMÓN Luis, Op. Cit.

La estructura de una dirección anycast es la siguiente:

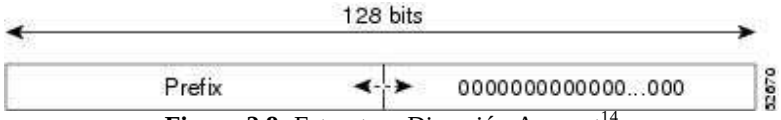

Figura 2.8: Estructura Dirección Anycast[14]

Las direcciones anycast utilizan una porción del espacio asignado a las direcciones unicast, así, un paquete anycast no se puede distinguir fácilmente de otro proveniente de una interfaz unicast, es por esto que, la interfaz debe estar configurada para identificar una dirección anycast. Una dirección unicast se transforma en anycast, si ésta es asignada a más de una interfaz.

Esta clase de direcciones no debe utilizarse como una dirección de origen en una transmisión IPv6, por lo cual, son utilizadas para implementar los siguientes tipos de mecanismos.

- ***Comunicación con el servidor más cercano.-*** permite establecer comunicación con el servidor más cercano dentro de la red.
- ***Descubrimiento de Servicios.-*** no es necesario especificar el DNS, proxy, etc., al configurar un nodo con IPv6.
- ***Movilidad.-*** los nodos pueden comunicarse con un solo router a la vez de los que se encuentran disponibles en la red.

2.3.1.3 Direcciones Multicast IPv6

Una dirección multicast en IPv6, sirve para identificar a un grupo de interfaces, es decir, si un paquete es enviado a una dirección multicast, el mismo será entregado a todas las interfaces que tengan la dirección de destino.

[14] **Fuente:** CISCO, Op. Cit.

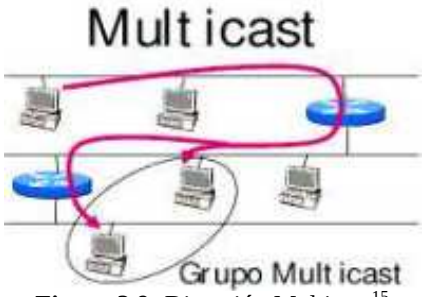

Figura 2.9: Dirección Multicast[15]

La figura 2.10 representa la estructura de una dirección multicast.

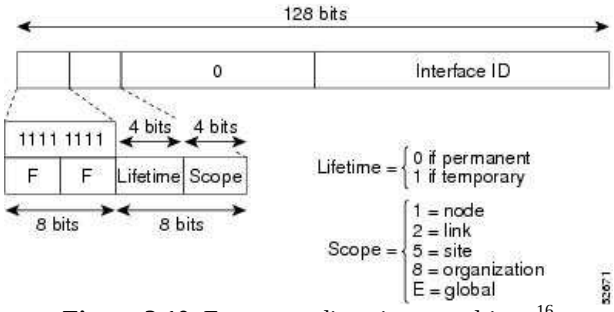

Figura 2.10: Estructura direcciones multicast[16]

- Los primeros 8 bits tienen el valor de 1, son los prefijo FF, que caracteriza a las direcciones multicast.
- ***Lifetime* o Tiempo de vida.-** este campo puede tener cualquiera de los siguientes valores:
  - Si es 0 nos indica que es una dirección permanente
  - Si es 1 nos indica que es una dirección temporal.
- ***Scope* o Ámbito.-** utilizado para limitar el alcance del grupo multicast y puede tener los siguientes valores hexadecimales:

[15] **Fuente:** SÁNCHEZ Belén RAMÓN Luis, Op. Cit.
[16] **Fuente:** CISCO, Implementing IPv6 Multicast, 25/01/2013, disponible en: <http://www.cisco.com/en/US/docs/ios-xml/ios/ipv6/configuration/15-2s/ip6-multicast.html,>

0	Reservado
1	Nodo Local
2	Enlace Local
3	Subred Local
4	Admin Local
5	Sitio Local
8	Organización Local
E	Ámbito Global
F	Reservado

Tabla 2.1: Significado de bits de ámbito en Multicast[17]

Tomado como referencia del documento Multicast IPv6[18]; en la tabla 2.1 se detallan únicamente los valores hexadecimales significativos, utilizados para la creación de la dirección multicast.

2.3.2 Características de las Direcciones IPv6

- Las direcciones IPv6 son asignadas a las interfaces lógicas.
- Una sola interfaz puede tener asignadas varias direcciones IPv6, las cuales pueden ser multicast, anycast y unicast.
- El ámbito de acción que tienen las direcciones IPv6, puede ser de tres tipos:
    - De enlace local
    - De sitio local
    - Global.

2.3.3 Representación de Direcciones IPv6

Según el Autor de la Tesis *Evolución de las Redes Fijas del Protocolo IPv4 a IPv6 en Guatemala,* para representar direcciones y prefijos IPv6, se deberá cumplir las siguientes normas.

---

[17] **Fuente:** El Autor, tomado como referencia del Documento: Multicast IPv6, CICILEO Guillermo PALET Jordi, 25/01/2013 disponible en: <http://lacnic.net/documentos/lacnicxii/presentaciones/flip6/05_Jordi.pdf>

[18] Idem.

- Para representar de direcciones IPv6, debemos tener en cuenta la siguiente estructura:

x:x:x:x:x:x:x

La x representa a un valor hexadecimal de 16 bits, perteneciente a la dirección IPv6. A continuación se muestra un ejemplo de una dirección:

3ffe:3328:0041:0003:250:4ff:fe5c:b3f4

Para simplificar a las direcciones IPv6 se puede suprimir los ceros que están ubicados a la izquierda de cada grupo de 16 bits, como se muestra a continuación:

3ffe:3328:41:3:250:4ff:fe5c:b3f4

- Podría darse el caso, de que direcciones IPv6 tengan varios bits con el valor de “cero”, en este caso podremos reemplazar con “::” a toda esa cadena consecutiva de ceros.

3234:328:0:0:0:7f8:fb8c:d2f4

Se podrá representar así:

3234:328::7f8:fb8c:d2f4

Pero esto no puede ser usado más de una ocasión dentro de una dirección IPv6:

- En ambientes mixtos existen direcciones IPv6 que contienen a direcciones IPv4 como se muestra a continuación:

0:0:0:0:0:FFFF:128.12.13.14

De manera abreviada tendría la siguiente forma:

::FFFF:128.12.13.14

2.3.3.1 Representación de los prefijos de direcciones IPv6

"La representación de los prefijos en direcciones IPv6 es similar a la que se utiliza IPv4 con CIDR[19]."[20]

Dirección IPv6/longitud de prefijo

En donde:

- ***Dirección IPv6:*** puede ser cualquier tipo de dirección IPv6 de las vistas anteriormente..
- ***Longitud de prefijo:*** valor decimal con el cual se representan a los bits ubicados a la izquierda de la dirección corresponden al prefijo:

fe80::41f3:f4de:d278:7550/48

La dirección IPv6 es:

fe80::41f3:f4de:d278:7550

Y la longitud del prefijo es:

48

2.4 AUTOCONFIGURACIÓN DE DIRECCIONES IPv6

Con la autoconfiguración el host adquiere una dirección del tipo enlace local y realiza una revisión, de todas las direcciones que pertenecen al enlace para comprobar que ésta se única, con lo cual adquiere información para ser autoconfigurada, en el caso de necesitar una dirección IP esta podrá ser obtenida a través de dos mecanismos:

---

[19] **CIDR:** Enrutamiento interdominios sin clases

[20] **Fuente:** SANTIZO Robert, *Evolución de las Redes Fijas del Protocolo IPv4 a IPv6 en Guatemala*, Tesis USCG Facultad de Ingenierías Escuela de Ingenierías en Ciencias y Sistemas, Guatemala, 25/01/2013, disponible en <http://biblioteca.usac.edu.gt/tesis/08/08_0183_CS.pdf>

- ***Autoconfiguración de direcciones sin estado:*** en este método el host no necesita ser configurado manualmente, los routers pueden trabajar con una configuración mínima, y no es necesario la presencia de servidores adicionales. El host generará su propia dirección mediante la combinación de la adquirida localmente y de mensajes enviados por él o los routers.

  Esta dirección estaría conformada por el prefijo que nos asigna el router, y el identificador de interfaz que proviene de cada host. Sí, en el enlace no existiera ningún router, el host generará una dirección IPv6 de enlace local, que le permitirá la comunicación dentro de ese enlace.

  Cuando la interfaz del host esta activada, su autoconfiguración seguirá los siguientes pasos:

  1. Creación de una dirección de enlace local.
  2. Comprobación de la dirección creada para descartar, que este duplicada en el mismo enlace.
  3. Si no existe direcciones duplicadas, se establece la conectividad a nivel de IP, permitiendo asignar dicha dirección a la interfaz.
  4. Si la dirección es para un host, los routers mostrarán que procedimiento deber seguir el nodo.
  5. El procedimiento de autoconfiguración mediante DHCPv6 se la realiza únicamente si no existe ningún router dentro del enlace.
  6. Si dentro del enlace existen routers, estos indicarán el mecanismo a utilizar para la obtención de una dirección IPv6.

- ***Autoconfiguración de direcciones con estado (DHCPv6*** [21] ***):*** a diferencia del mecanismo anterior, éste requiere de un servidor para que el host puede obtener su dirección IPv6, el cual envía un mensaje que contiene la dirección IPv6 e información del host, y también la dirección del DNS. En los servidores existen tablas que contienen las direcciones IPv6 que han sido asignadas a cada host.

  Al utilizar un servidor DHCPv6 para la asignación de direcciones, se abaratan los costos de la administración de los host y beneficia a los administradores de red, debido a que todo está centralizado, permitiendo que la asignación de recursos sea más rápida y exacta.

Antes de asignar direcciones IPv6 a las interfaces, los nodos verifican a través del algoritmo de detección de direcciones duplicadas, evitando así, la duplicidad de direcciones dentro del mismo enlace. Este algoritmo puede ser utilizado por cualquier tipo de direcciones, no importa el mecanismo utilizado para la obtención de ésta.

"La autoconfiguración está diseñada para host, no para routers, lo cual no implica que parte de la autoconfiguración de los routers también pueda ser realizada automáticamente. Además los routers, también deben realizar el algoritmo de detección de direcciones duplicadas."[22]

[21] **DHCPv6.-** Protocolo de Configuración Dinámica de Hosts para IPv6
[22] **Fuente:** PALET, Jordi, Tutorial de IPv6: INTRODUCCIÓN disponible en: < http://long.ccaba.upc.es/long/050Dissemination_Activities/jordi_palet_tutorialipv6introduccion.pdf>

2.5 MECANISMOS DE TRANSICIÓN

Con la escasez de direcciones IPv4, la transición a Ipv6 a empezado en los últimos años de manera progresiva, con lo que se espera que estos dos protocolos coexistan varios años más, con el fin, de que la integración y la transición sea imperceptible para los usuarios

Debido a estos problemas fueron creados mecanismos de transición que son un conjunto de métodos y de protocolos que se implementan tanto a los hosts como a los routers IPv6, para que de esta manera sean compatibles con el protocolo IPv4. Existen 3 mecanismos transición que son:

2.5.1 Dual Stack

Este mecanismo consiste en hacer trabajar de manera sincrónica a los protocolos IPv4 e Ipv6 en los nodos que componen la red, cada uno de estos nodos tendrán establecida una dirección IPv4 y una IPv6, permitiendo asegurar la conectividad entre todos los nodos de la red independientemente del tipo de protocolo que estén utilizando. Más que un mecanismo de transición es un mecanismo de integración.

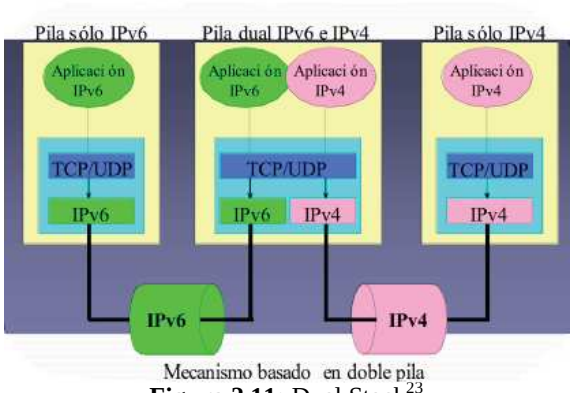

Figura 2.11: Dual Stack[23]

[23] **Fuente:** CASTELLANOS, Germán y MENDOZA, Mario, IPv4 e IPv6, 25/01/2013, disponible en: <http://manejo2010.wikispaces.com/ipv4+y+ipv6>

2.5.2 Túneles

Los túneles son procesos en los cuales son encapsulados información de un protocolo como por ejemplo IPv4 dentro de un paquete del protocolo IPv6. Este método es muy utilizado en redes donde ya se conoce las rutas de tráfico.

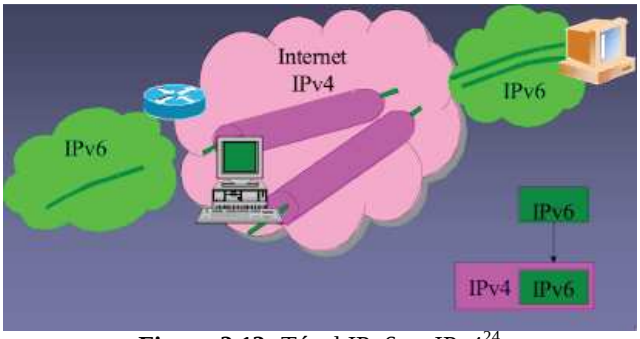

Figura 2.12: Túnel IPv6 en IPv4[24]

La figura 2.12 se observan como viaje dentro en una trama IPv6, paquetes Ipv4. Los requisitos necesarios para realizar el *tunneling* es conocer la dirección IP de origen y destino que son utilizados en la encapsulación y además las direcciones de origen y destino para entablar la conexión. Existen dos tipos de túneles:

2.5.2.1 Túneles Configurados

Esta tipo de túnel es utilizado por proveer conexiones más estables y seguras entre routers, puede ser unidireccional o bidireccional, la dirección del destino se obtiene a través de información de la configuración del túnel del nodo que va a encapsular el paquete.

[24] Idem.

El objetivo principal de los túneles configurados es de interconectar dos redes que pueden ser IPv4 o IPv6 para la transmisión de tráfico IPv6, utilizando para ello una infraestructura IPv4, en cada nodo se configura las direcciones IPv6 e IPv4.

2.5.2.2 Túneles Automáticos

Los túneles automáticos permiten a los nodos IPv6/IPv4 utilizar una infraestructura IPv4 de manera similar a los túneles configurados, para esto utilizan un encapsulamiento IPv6 over IPv4; en donde, la dirección de destino utilizada por el túnel IPv4, se la determina en función de la dirección IPv4 agregada en los últimos 32 bits de la dirección IPv6 destino, la cual deberá ser compatible con IPv4.

Los túneles automáticos que utilizan direcciones IPv6 compatibles con IPv4, son utilizados para establecer conexiones entre los routers de la red. La inicialización y finalización de estos túneles dura tanto como dura la comunicación.

La Figura 2.13 se observa como están formadas las direcciones, en donde los últimos 32 bits pertenecen a la dirección IPv4 del nodo destino.

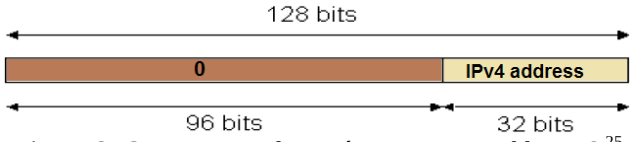

Figura 2.13: Estructura dirección IPv4 compatible IPv6.[25]

[25]**Fuente:** El Autor

A continuación se observa cómo está configurado un túnel automático

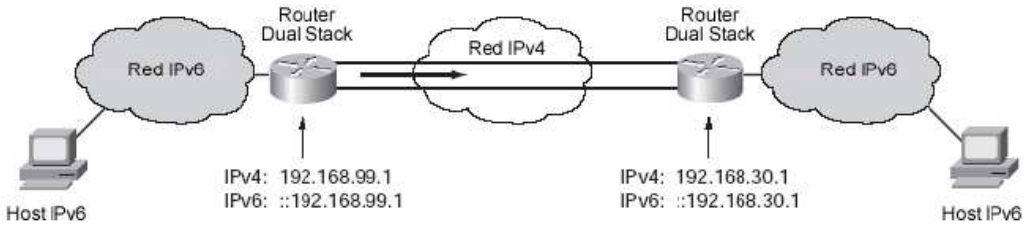

Figura 2.14: Túnel Automático compatible con IPv4[26]

2.5.2.3 Túneles 6to4

Este mecanismo fue diseñado para que exista convivencia entre IPv6 e IPv4, es utilizado para la conexión entre varias redes, en las cuales existe al menos una conexión que trabaje con una red IPv4.

En cada sitio donde va a implementar *tunneling,* debe existir por lo menos una dirección IPv6 y un router que sea *dual-stack*, en el cual su interfaz, debe estar configurada una dirección 6to4. Con esto tendremos que dentro de la red trabajaremos con ruteo utilizando el protocolo IPv6 y para comunicarnos fuera de la red trabajaremos con el protocolo IPv4.

En la figura 2.15, se muestra el funcionamiento del túnel 6to4, las ventajas de utilizar este tipo de túnel es que son imperceptibles a nivel IPv6, por lo que no afecta el funcionamiento de la red, son túneles dinámicos

[26] **Fuente:** Jessica Barrera, Edgar Guerra, Implementación de Tunneling entre redes IPv4 e IPv6, 25/01/2013, disponible en: <http://repositorio.espe.edu.ec/handle/21000/406>

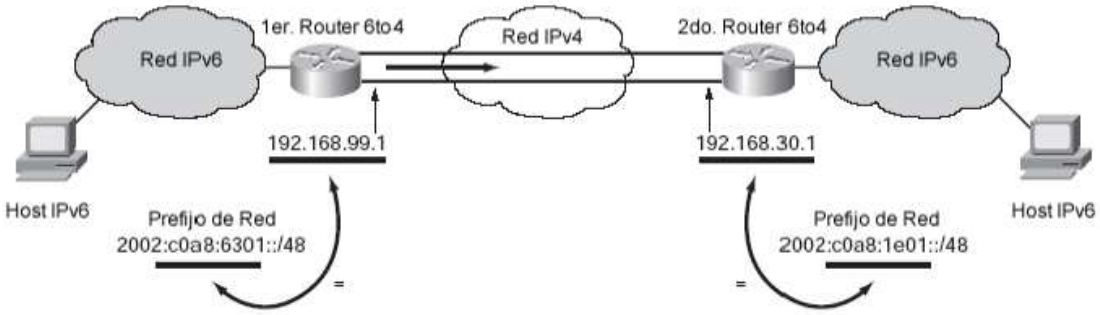

Figura 2.15: Túnel Automáticos 6to4[27]

2.5.2.4 Tunnel Broker

Un host dual stack, que pertenece a una red IPv4, utiliza un broker para que establezca la comunicación entre los servidores del túnel y el cliente, el trabajo que realiza el broker es hallar un servidor de túnel que esté disponible para la comunicación, una vez encontrado el servidor, se solicita la configuración del túnel, esta puede ser información de autenticidad direcciones IP, servidores DHCP, etc., esto se lo envía al cliente en forma de script que al ejecutarlo establece un túnel IPv6 en IPv4, al servidor *tunnel broker*.

2.5.3 Mecanismos de Translación

Un mecanismo de translación permite comunicar a hosts que funcionan solo con IPv6 con hosts que trabajan solamente con IPv4, uno de los mecanismos más conocidos en NAT-PT[28], el cual se encarga de convertir directamente paquetes IPv6 en paquetes IPv4 y viceversa, es transparente con lo cual la conexión de los nodos no se ve afecta en ningún momento. En este caso necesitaremos configurar un router para que realice la transformación.

[27] Idem.
[28] **NAT-PT.-** Traducción de Red-Traducción de Protocolo

2.6.3.1 Características

- Permite la comunicación de nodos que solo tienen soporte para IPv6, con otros nodos que solo tienen soporte para IPv4 y que se encuentran ubicados en redes distintas.
- Utiliza un dispositivo dedicado que realiza la traducción de direcciones IPv4 a IPv6 y viceversa. El traductor de direcciones es denominado IPv6NAT y es similar en su lógica al traductor de direcciones IPv4 NAT pero no idéntico. IPv4 NAT traduce una dirección IPv4 a otra dirección IPv4. IPv6 NAT traduce una dirección IPv4 a una dirección IPv6 y viceversa.
- El traductor de protocolo, utiliza el mecanismo de traducción SIT[29] para traducir un paquete IPv4 en un paquete equivalente en formato IPv6 y viceversa. NAT-PT reside dentro del router frontera de la red interna IPv6 con la red IPv4 denominada Internet.
- Utiliza una pila de direcciones IPv4 para asignarlos a los nodos que tienen soporte solo para IPv6. Dichas direcciones deben ser únicas, deben ser direcciones IPv4 públicas y no privadas.
- Este mecanismo requiere por lo menos una dirección IPv4 pública para la red IPv6 que desea comunicación con redes IPv4.
- Utiliza una tabla de mapeo que contiene el vínculo de las direccionesIPv6 de los nodos internos de la red IPv6 con las direcciones IPv4 de los nodos externos a la red y viceversa para proveer un ruteo transparente.

[29] **SIT: Transición Simple de Internet.-** conjunto de mecanismos de protocolos diseñados para una transición simple a IPv6

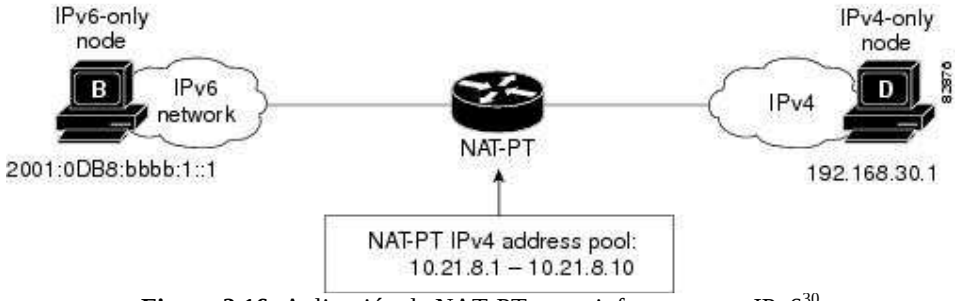

Figura 2.16: Aplicación de NAT-PT a una infraestructura IPv6[30]

[30] **Fuente:** CISCO, Implementing NAT-PT for IPv6, 25/01/2013, http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-nat_trnsln_ps6350_TSD_Products_Configuration_Guide_Chapter.html

CAPITULO III

3. REDES WIRELESS: EL ESTÁNDAR 802.11 Y 802.11E

3.1. INTRODUCCIÓN

El protocolo 802.11 es un conjunto de estándares, también conocido por su nombre comercial *Wi-Fi*[31]; que fue desarrollado en 1997 por la IEEE. El cual, especifica el uso de la capa física y la de enlace de datos del modelo OSI, determinando su funcionamiento dentro de una WLAN. Pertenece a la familia del estándar 802, debido a esto se lo conoce como Ethernet Inalámbrica. En la Figura 3.1 se puede observar a esta familia.

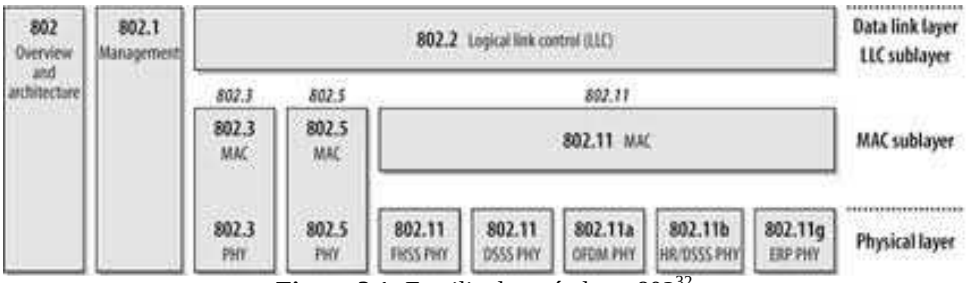

Figura 3.1: Familia de estándares 802[32]

El 802.11 original, define una interfaz entre el usuario y punto de acceso, a través de vía aérea, trabajaba con velocidades de 1 y 2 Mbps, con un sistema de seguridad WEP, que transmite en una frecuencia de 2.4 GHz, actualmente el estándar 802.11 presenta seis técnicas de transmisión.

---

[31] **Wi-Fi:** Fidelidad Inalámbrica

[32] **Fuente:** Carlos Benedito Guerrero, WLAN, parte 1, 27/01/2013, disponible en: <http://vidateleco.wordpress.com/2009/04/13/wlan-parte-1/>

En el estándar 802.11, aparece la el concepto de BSS[33], que es una red conformada por dos o más estaciones inalámbricas, las cuales se reconoce entre ellas, estableciendo una conexión para el intercambio de información, esta conexión, también puede realizarse a través de un punto de acceso. Estas comunicaciones se las realiza dentro de un área conocida como BSA[34]. Los BSS brindan soporte a los tipos de redes en la que se clasifica:

- ***Redes Independientes:*** es conocida también IBSS[35], en este tipo de BSS la comunicación de los nodos es de forma directa. Generalmente estas redes están conformadas por un número pequeño de hosts, los cuales estan configurados de una manera específica durante un período corto de tiempo, normalmente son utilizadas para realizar videoconferencia; aquí, los usuarios una vez iniciada la conferencia generan un IBSS para intercambiar información, finalizada la reunión la red se cancela. A las IBSS se las conoce también como redes Ad-hoc.

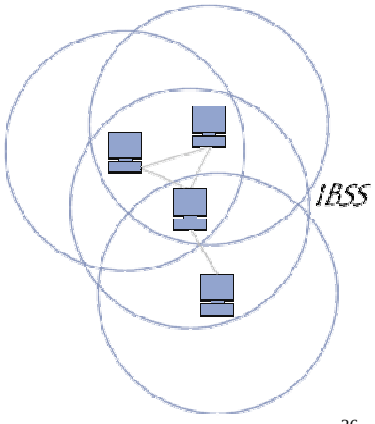

Figura 3.2: Red IBSS o ad-hoc[36]

[33] **BSS: Conjunto de Servicios Básicos.-** conjunto de normas IEEE para una red inalámbrica que tiene sólo un único punto de acceso inalámbrico.
[34] **BSA: Área de Servicio Básico.-** es la zona donde se comunican las estaciones de una misma BSS, se definen dependiendo del medio de transmisión.
[35] **Conjunto Básico de Servicios Independientes.-** es un sistema de estaciones que se están conectadas vía inalámbrica de manera uno a uno.
[36] **Fuente:** Kioskea, Modos de Funcionamiento Wifi, 27/01/2013, disponible en: <http://es.kioskea.net/contents/wifi/wifimodes.php3>

- ***Redes de Infraestructura:*** a diferencia de las IBSS, en las redes de infraestructura se utiliza un punto de acceso en donde estarán centralizadas todas las comunicaciones de los hosts que la conforman. De esta manera sin un host que se encuentra fuera del radio de cobertura de otro, podrá transmitir información a través del punto de acceso utilizando un DS[37]. Por lo que, los hosts deberán asociarse con un único punto de acceso para transmitir datos, tomando en cuenta la distancia en la que se encuentra de su posición. En la Figura 3,2 se observa una red de .infraestructura.

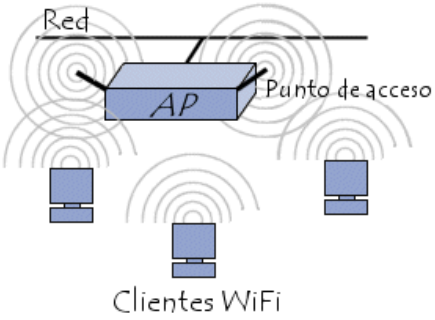

Figura 3.3: Red de Infraestructura BSS[38]

Para que la cobertura de la red sea mayor, se pueden interconectar varias BSS, mediante un DS, la conexión puede ser de forma cableada o inalámbrica. A este conjunto de BSS se los conoce como ESS[39], la desventaja que presenta este tipo de redes es que existe la presencia de solapamiento entre las señales de los diferentes puntos de accesos, por lo que se recomienda que estos trabajen en diferentes frecuencias para evitar interferencia, y así las estaciones puedan movilizarse por toda el área de cobertura, sin preocuparse de perder la conexión. Cuando una estación requiere cambiarse a un nuevo punto de

[37] **DS: Sistema de Distribución.-** es el medio por el cual se comunican un punto de acceso con otro punto de acceso para intercambiar tramas entre estaciones de sus respectivos BSS.
[38] **Fuente:** Kisokea, Op. Cit.
[39] **ESS: Conjunto de Servicio Extendido.-** Es un conjunto de varios BSS que se comunican a través de un DS.

acceso, ésta debe desasociarse del punto de acceso al que está conectado. En la figura 3.4 se puede ver una red ESS.

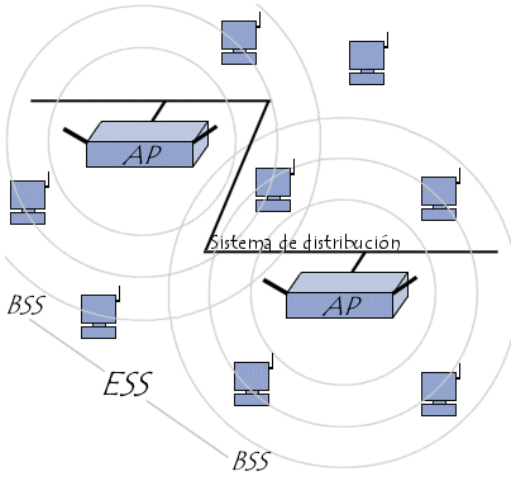

Figura 3.4: Conjunto de Servicio Básico Extendido (EBSS)[40]

3.2 CAPA FÍSICA

El estándar 802.11 divide a la capa física en dos subcapas como se muestra en la figura 3.5:

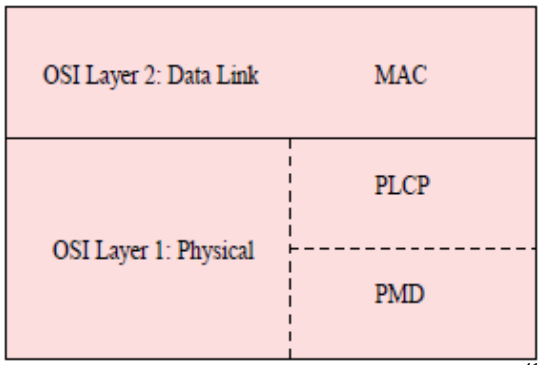

Figura 3.5: Arquitectura lógica de una red 802.11[41]

[40] **Fuente:** Kioskea, Op. Cit.
[41] **Fuente:** Javier Cañas, Introducción a las Redes Inalámbricas 802.11, 27/01/2013, disponible en: <http://www.microalcarria.com/descargas/documentos/Wireless/wireless.pdf>

- ***La subcapa PMD[42]:*** se encarga de la modulación y codificación de la señal, así, determina el modo en el que los datos viajarán a través de un medio inalámbrico.
- ***La subcapa PLCP[43]:*** adapta las tramas que son generadas por la subcapa MAC, al formato de la subcapa PDM, para que ésta los transforme en paquetes los cuales serán transmitidos.

3.2.1 MÉTODOS DE TRANSMISIÓN

El estándar 802.11 define varias técnicas de transmisión para ser implementadas en redes inalámbricas, las más utilizadas son las que trabajan con el concepto de *spread spectrum,* las cuales son:

- ***FHSS[44]:*** este método utiliza saltos de frecuencia, por lo que, el estándar 802.11 ha dividido la banda de 2.4 GHz en 79 subcanales de 1 MHz de ancho, evitando el solapamiento entre los subcanales. Con FHSS se tiene una alta inmunidad a la interferencia y al ruido. Esta técnica consiste en transmitir una parte de la información en una frecuencia durante un lapso tiempo, el cual si es superior a 400 ms, ésta cambia la frecuencia para seguir transmitiendo; así, cada trama es transmitida en diferentes frecuencias. En la Figura 3.6 se puede observar cómo trabaja FHSS.

---

[42] **PMD: Dependiente del Medio Físico.-** se encarga de la transmisión y recepción de datos de las diferentes estaciones

[43] **PLCP: Procedimiento de Convergencia de Capa Física.-** se encargar de realizar la comunicación entre la capa Mac y la subcapa PMD.

[44] **FHSS: Espectro Ensanchado por Salto de Frecuencia.-** es una técnica de modulación el que la señal se emite sobre una serie de radiofrecuencias aleatorias, saltando de frecuencia en frecuencia y en cada subcanal transmite una porción de los datos.

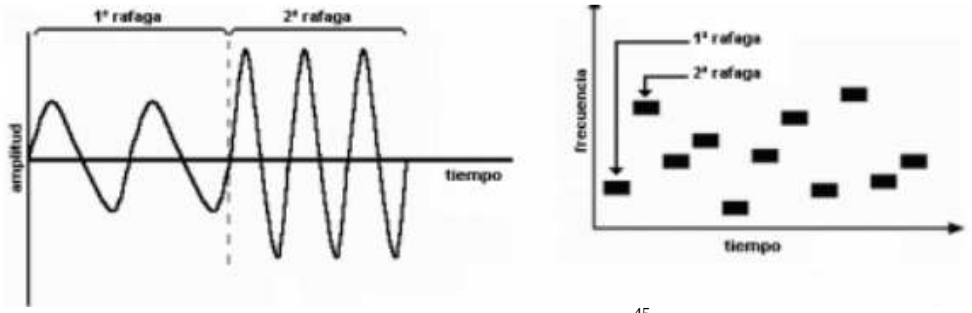

Figura 3.6: Salto de Frecuencia[45]

- ***DSSS[46]:*** trabaja en la banda de los 2.4 GHz, este método utiliza un código de pseudo-ruido, generado localmente para codificar la señal digital a transmitir, éste se ejecuta varias veces en frecuencias más altas, que la señal original. El receptor al recibir esta señal la decodifica utilizando un código similar al que utilizó el emisor.

3.3 CAPA DE ENLACE

La capa de enlace de acuerdo al estándar 802.11, está dividida en dos subcapas:

- ***Subcapa LLC[47]:*** su funcionamiento es igual a la subcapa LLC que es usada por la redes Ethernet del estándar 802.3. Se encarga de especificar, como los datos van a ser enviados por el medio de transmisión, además tiene la función del control de errores, control de flujo y de proporcionar una interfaz a la subcapa MAC y a la capa de Red.

---

[45] **Fuente:** Enrique de Miguel Ponce y Otros, Redes Inalámbricas: IEEE 802.11, 27/01/2013, disponible en: <http://multingles.net/docs/Manual%20-%20Redes%20WiFi%20inalambricas.pdf>

[46] **DSSS: Espectro ensanchado por secuencia directa.-** es uno de los métodos de codificación de canal para transmisión de señales digitales sobre ondas radiofónicas.

[47] **LLC: Control de Enlace Lógico.-** define la forma en que los datos son transferidos sobre el medio físico, proporcionando servicio a las capas superiores.

- ***Subcapa MAC***[48]***:*** proporciona control de acceso equitativo al medio inalámbrico compartido, además de delimitar las tramas, y proporcionar direccionamiento.

  CSMA/CA[49] es el protocolo de acceso al medio que se utiliza, que a diferencia de CSMA/CD[50], las estaciones tienen que comprobar si el medio está disponible antes de empezar a transmitir. Esto significa que una red 802.11 usa un esquema de acceso distribuido, es decir, nadie tiene el control total de la red, y cada estación deberá emplear el mismo método y así, obtener acceso al medio, para empezar a transmitir.

  Existen dos tipos de funcionamiento:

  - ***DCF***[51]***:*** es cuando una estación puede transmitir y recibir datos de protocolo a nivel MAC a través del medio inalámbrico.
  - ***PCF***[52]***:*** las estaciones acceden al medio inalámbrico coordinadas por un punto de coordinación que encuesta a los nodos si desean o no transmitir

  En la figura 3.7 se observa los modos de funcionamiento que existen en la capa MAC.

---

[48] **MAC: Control de Acceso al Medio.-** es la encargada de la topología lógica de la red y del método de acceso a ésta.
[49] **CSMA/CA: Acceso múltiple con escucha de portadora y Detección de Colisiones.-** las estaciones esperan a transmitir si el canal se encuentra ocupado. Cuando el canal está libre, la estación comienza a transmitir. Inmediatamente, esa estación es capaz de comprobar si se está produciendo una colisión, por lo que puede abortar ese envío de forma casi instantánea
[50] **CSMA/CD:** Acceso múltiple por Detección de Portadora con Detección de Colisiones.
[51] **DCF:** Función de Coordinación Distribuida
[52] **PCF:** Función de Coordinación Puntual

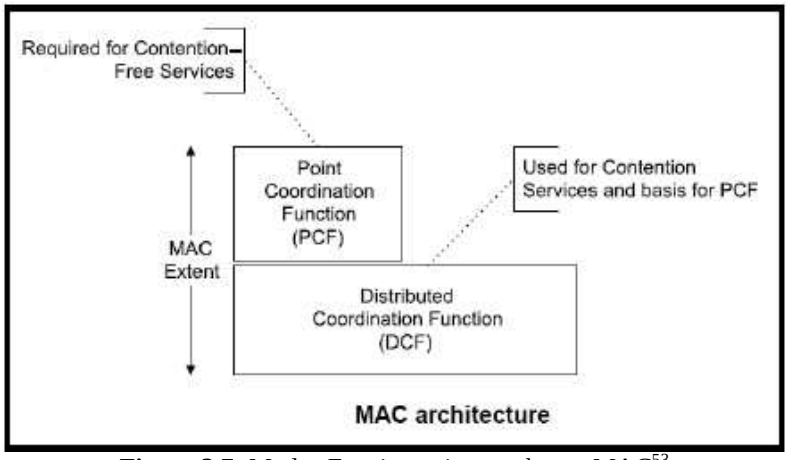

Figura 3.7: Modos Funcionamiento subcapa MAC[53]

3.3.1 Problema del nodo oculto

En una red Ethernet, las estaciones utilizan la recepción de tramas transmitidas para implementar la función CSMA/CD, en las redes wireless existen la presencia de fronteras difusas, donde existen nodos que no se pueden comunicar con otros.

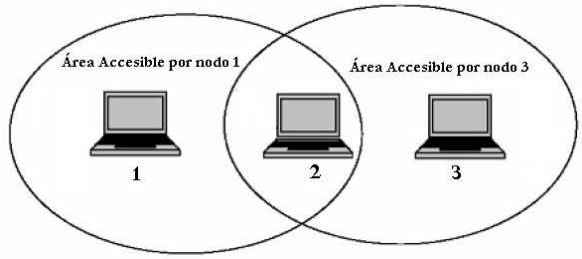

Figura 3.8: Problema del nodo oculto[54]

En la figura 3.8, el nodo 2 se puede comunicar con el nodo 1 y 3, pero los nodos 1 y 3 no pueden hacerlo directamente; este se debe a el nodo 3 está "oculto", desde la vista del nodo 1 y por lo tanto que los dos nodos transmitan al

[53] **Fuente:** Nelson Pozo, *Estudio y diseño de una red Lan inalámbrica con Calidad de Servicio, para voz y datos en el Colegio de Ingenieros Geólogos, Minas y Petróleos (CIGMYP), empleando los estándares IEEE 802.11g, IEEE 802.11e*, Tesis E.P:N Facultad de Ingeniería Eléctrica y Electrónica, 27/01/2013, disponible en <http://bibdigital.epn.edu.ec/handle/15000/1316>

[54] **Autor:** El Tesista, tomado del Trabajo de Investigación de: Javier Cañas, disponibles en: <http://www.microalcarria.com/descargas/documentos/Wireless/wireless.pdf>

mismo tiempo es posible, en cuyo caso el momento de realizar la trasmisión ambos mensajes colisionarán, ésta colisión es local al nodo 2, la colisión que provocan los nodos ocultos; son difíciles de detectar en redes inalámbricas, para prevenir colisiones se introducen dos nuevos mensajes:

- ***RTS***[55]***:*** reserva el medio para la transmisión y silencia a la estación que lo escucha
- ***CTS*** [56] ***:*** la estación receptora responde con un CTS, silencia las estaciones vecinas.

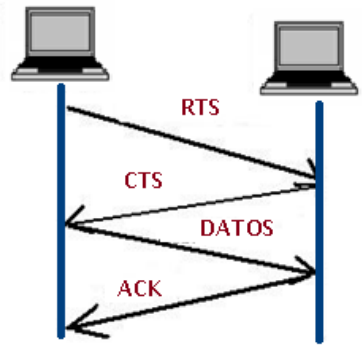

Figura 3.9: Intercambio completo de paquetes con RTS/CTS[57]

Una vez que se ha completado el intercambio RTS/CTS, el nodo 1 puede trasmitir sin preocuparse de la interferencia ocasionada por los nodos ocultos. Al generar el receptor mensajes CTS, éste silencia a los nodos ocultos que están fuera del rango de la estación que transmite. El intercambio RTS/CTS consume una parte de la capacidad del canal, por esta razón se usa sólo en ambientes que tienen una contención significativa en la transmisión, en este caso el intercambio RTS/CTS sólo se realiza para paquetes cuyo tamaño supera el *RTS threshold*[58]*,* si los paquetes son cortos, se envían sin intercambio de RTS/CTS.

---

[55] **RTS: Solicitud de Envío.-** es una señal enviada por un dispositivo de comunicaciones, para verificar si el otro dispositivo está listo para recibir datos.
[56] **CTS: Listo para Transmitir.-** señal de autorización que da el dispositivo para transmitir
[57] **Fuente:** El Tesista, Op. Cit.
[58] **RTS threshold.-** tamaño del paquete en el cual un punto de acceso emite una solicitud de envío (RTS). El rango es de 0 a 2347 bytes. El valor definido por defecto es 2347.

3.3.2 Modos de Acceso y Diagramas de Tiempo

El acceso al medio inalámbrico es controlado por funciones de coordinación, entre estas se encuentra DCF, que es la base del estándar CSMA/CA la cual proporciona una función acceso al medio, de manera similar a Ethernet, antes de transmitir verifica que el medio está en silencio. Para evitar colisiones, DCF utiliza *backoff* [59] aleatorio después de cada trama, también en algunas circunstancias puede utilizar CTS/RTS para reducir las probabilidades de que exista una colisión.

3.3.3. Tiempo entre Tramas

Los nodos luego de verificar que el medio este libre y establecer la comunicación, deben esperar un intervalo de tiempo para empezar a transmitir. El estándar 802.11, determina cuatro tipos de tiempos:

- ***SIFS (Short Interframe Space).-*** es utilizado para transmitir tramas con mayor prioridad, entre las que se encuentran los ACK's. luego que ha transcurrido un intervalo de tiempo SIFS, las transmisiones que tengas prioridad serán las primeras en acceder al medio.
- ***PIFS (PCF Interframe Space).-*** Los hosts que necesiten transmitir podrán hacerlo luego de haber transcurrido un tiempo PIFS, este tiempo se calcula aplicando la siguiente fórmula:

PIFS = SIFS + slotTime[60]

[59] **Backoff.-** algoritmo con el que se intenta evitar sobrecargar la red, realizando retransmisiones una vez que la red está saturada.

[60] **Fuente:** OLVER, Emilio, *Análisis de mecanismos de calidad de servicio para aplicaciones multimedia en IEEE 802.11e*, Tesis U.A.M Departamento de Ingeniería Eléctrica Área: Redes y Telecomunicaciones,México, 29 Mayo 2012, http://mcyti.izt.uam.mx/archivos/Tesis/Generacion2009/ICR_EmilioOlvera.pdf

- ***DIFS (DFC Interframe Space).-*** es el tiempo mínimo que el medio debe estar disponible, para empezar a transmitir:

DIFS = SIFS + 2*slotTime[61]

- ***EIFS (Extended Interframe Space).-*** Este tiempo es el más largo de todos, pues es usado para la retrasmisión de tramas en caso de existir un error en la transmisión.

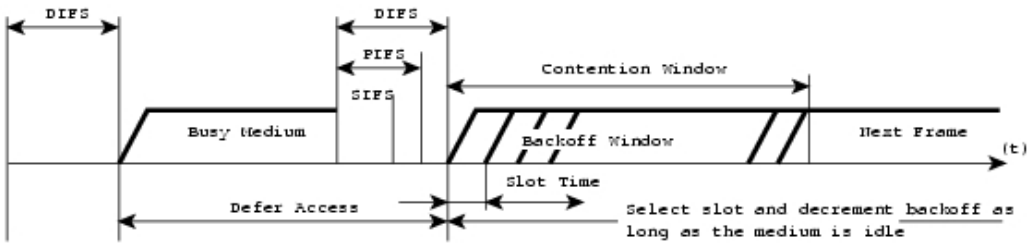

Figura 3.10: Espacio entre tramas[62]

La figura 3.10 muestra la relación entre estos tiempos de espaciado de trama.

3.3.4 CSMA/CA

La función principal de este protocolo es la escuchar antes de transmitir, para reducir la posibilidad de que se produzca una colisión entre dos o más estaciones que desean intercambiar información simultáneamente, este protocolo realiza un retardo a la hora de la transmisión, una vez que se ha comprobado que el medio está disponible.

---

[61] **Fuente:** Idem.
[62] **Fuente:** Paul Vracken, IEEE 802.11 Medium Access Control (MAC), 27/01/2013, disponible en: <http://www.wirelesscommunication.nl/reference/chaptr01/wrlslans/80211_page2.htm>

3.4 TRAMAS 802.11

El estándar 802.11 define tres tipos de tramas que se pueden categorizar de acuerdo a sus funciones:

- ***Administración.-*** permite establecer y mantener la comunicación entre nodos inalámbricos.
- ***Control.-*** brindan asistencia a estaciones inalámbricas mientras intercambian información
- ***Datos.-*** transportan información entre los nodos y los puntos de acceso.

Todas las tramas tienen la misma composición:

- ***Cabecera MAC (MAC Header).-*** incluye información de control, dirección MAC, etc.
- ***Cuerpo de la Trama (Frame body).-*** campo de longitud variable que contiene información específica de la trama.
- ***Secuencia de chequeo de Trama (FCS).-*** contiene un campo de 32 bits conocido como Código de redundancia cíclica (CRC).

3.5 CALIDAD DE SERVICIO

Es un conjunto de tecnologías que garantizan la transmisión de cierta cantidad de datos en un determinado intervalo de tiempo. QoS es la capacidad que tiene una red para proveer un buen servicio a los distintos tipos de tráfico mediante la asignación de prioridades, garantizando un mínimo nivel de servicio. Con la aparición de aplicaciones como son el *streaming*, voz sobre IP,

videoconferencia, etc.; la necesidad de implementar técnicas de calidad de servicio se ha vuelto más evidente.

Al momento de diseñar una red, el ofrecer a los usuarios y a sus aplicaciones, las condiciones necesarias para una buena transmisión es muy importante. Con la aparición de nuevas aplicaciones, es inevitable que existan mecanismos con los cuales los administradores de la red garanticen un buen funcionamiento aplicando QoS. Con una correcta administración de los recursos de la red, la transmisión de datos no se vería afectada. Uno de los objetivos principales de una administrador es la de proveer QoS a los usuarios de la red.

A continuación se indica algunas de las situaciones en las cuales sería conveniente dar QoS:

- Para dar prioridad a ciertas aplicaciones de nivel crítico en la red.
- Para maximizar el uso de la infraestructura de la red.
- Para proveer una mejor performance a aplicaciones sensitivas al retardo como son las de voz y video.
- Para responder a cambios en los flujos del tráfico de red.

La calidad de servicio se puede definir como el proceso de entrega de datos de forma fiable. Cuando el administrador de la red aplica QoS, adquiere el control sobre los diferentes parámetros que definen la Calidad de Servicio:

- ***Delay (latencia).-*** es la diferencia que existe entre el tiempo, que una señal es transmitida y el tiempo que está llega a su destino. En aplicaciones como el video, es necesario que exista un retardo acotado para obtener una buena calidad en la aplicación

- ***Jitter (variación en el retardo).-*** es la diferencia entre el tiempo en que un paquete llega a su destino y el tiempo que se cree que arribará el paquete a su destino, esto puede ser causado por varios factores como la presencia de congestión en la red, las condiciones del medio en el caso de redes inalámbricas. Un ejemplo de jitter se muestra en la Figura 3.11

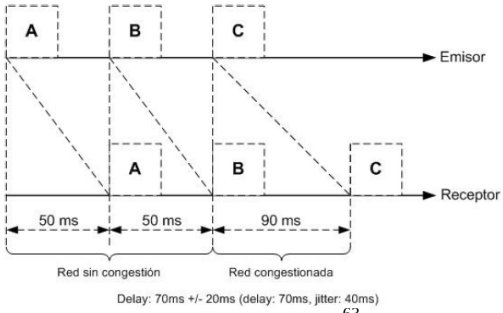

Figura 3.11: Jitter[63]

Los paquetes A y B llegan al destino cada 50 milisegundos pero el paquete C tarda 90 milisegundos; 40 milisegundos más de retardo que los dos paquetes anteriores lo que provoca un jitter de 40 milisegundos.

- ***Packet Loss (perdida de paquetes).-*** es cuando los paquetes no llegan a su destino a causa de diferentes factores como por ejemplo: la congestión de la red y las interferencias que se presentan en los medios inalámbricos, etc.
- ***Bandwidth (ancho de banda).-*** es la cantidad de paquetes que se envían mediante una conexión de red en un período de tiempo determinado, el ancho de banda está dado en bits.

[63] **Fuente:** Ana María Narváez y Mónica Flores, "Estudio de Calidad de Servicio en la coexistencia entre nodos WLAN 802.11b, g y e", 27/01/2013, disponible en <http://hdl.handle.net/123456789/616>

3.6 MODELOS DE SERVICIO

Existen tres tipos de modelos de servicio, los cuales describen un conjunto de capacidades que provee la red, a determinados tipo de tráfico desde su origen hasta su destino. Estos niveles se los detalla a continuación.

3.6.1 Servicio de mejor esfuerzo

Servicio que la red provee cuando hace todo lo posible para entregar un paquete a su destino, en un escenario donde no se puede garantizar que esto ocurra. Cualquier aplicación podrá enviar datos cuando lo necesite, sin la necesidad de pedir autorización o notificar a la red. Las aplicaciones Ftp y Http utilizan este modelo de servicio, pero no para aplicaciones en tiempo real, las cuales necesitan un tratamiento especial.

3.6.2 Servicios Integrados

Propone reservar recursos en los diferentes equipos que integran la ruta por la que viajara la información. Con este modelo se pretende ofrecer mejor soporte para aplicaciones en tiempo real. Este modelo define dos elementos los cuales son: una arquitectura en donde los elementos de la red pueden reservar recursos de conmutación y el protocolo RSVP[64] que se encarga de realizar de forma dinámica la reserva de los recursos solicitados por la aplicación.

Así cuando una aplicación desea establecer una comunicación, lo primero que realizará es una petición de recursos, la cual pasa por todos los nodos que conforman ese tramo para el flujo de información, y dependiendo de los recurso disponibles esta será aceptada o rechazada.

---

[64] **RSVP: Protocolo de Reserva de Recursos.-** protocolo de la capa de transporte diseñado para reservar recursos de una red bajo la arquitectura de servicios integrados, protocolo de control de internet.

3.6.3 Servicios Diferenciados

Fue creado para evitar los problemas de escalabilidad que presentaba los servicios integrados, propone la priorización del tráfico para el soporte de QoS. Este modelo incluye un conjunto de herramientas para la clasificación y mecanismos de cola para determinar la prioridad del tráfico de una aplicación sobre el resto en la red. Cuenta con los enrutadores de bordes para realizar esta clasificación

Esta clasificación se la puede realizar a través de la dirección de red, protocolo, puertos. Las ventajas de utilizar servicios diferenciados es que los enrutadores procesan la información de manera más rápida, se elimina el tráfico de señalización y el almacenamiento.

3.7 HERRAMIENTAS DE CALIDAD DE SERVICIO

Las herramientas de QoS que los administradores tienen disponibles se encuentran dentro de la siguiente clasificación:

3.7.1 Marcado y clasificación

Un paquete que entra en un router primero debe pasar por los procesos de marcado y clasificación, para conocer qué prioridad tiene, y diferenciarlo entre los distintos tráficos; y así, poder tratarlo de acuerdo al tipo de tráfico al que pertenecen.

- ***Marcado:*** es una forma de señalar con un identificador interno a un paquete, que luego puede ser usado para el criterio de filtrado y

traducción. Con este mecanismo se puede utilizar las reglas de traducción

- ***Clasificación:*** una vez marcado los paquetes son agrupados, el grupo al que pertenece cada uno de los paquetes depende del valor con el que fueron marcados para darles el tratamiento señalado.

3.7.2 Policing and shaping

- ***Policing.-*** proceso con el que se descarta paquetes en un flujo de datos, que al detectar tráfico excesivo los elimina, permitiendo mantener el flujo de datos dentro de los límites establecidos

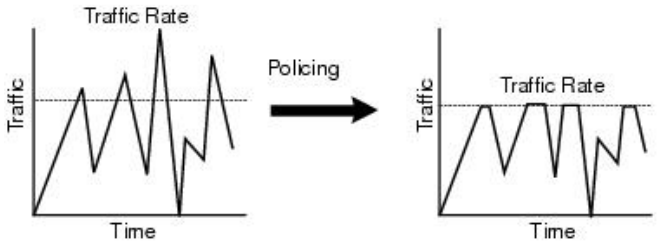

Figura 3.12: Policing[65]

- ***Shaping:*** retarda paquetes dentro de un flujo de tráfico, en este caso el exceso de paquetes no es eliminado sino que es aplazado, intentando no sobrepasar el límite establecido.

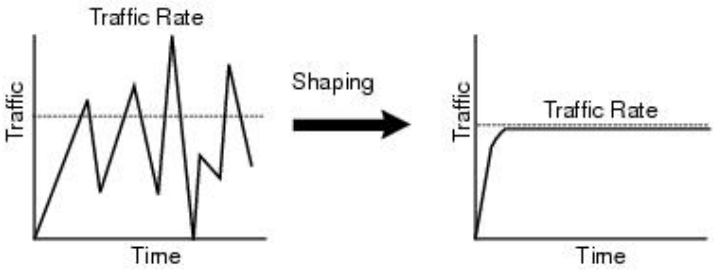

Figura 3.13: Shaping[66]

[65] **Fuente:** CISCO Systems, Configuring Quality of Service, 27/01/2013, disponible en: <http://www.cisco.com/en/US/docs/security/security_management/cisco_security_manager/security_manager/4.1/user/guide/rtqos.html>

3.8 EL PROTOCOLO 802.11e

Con los avances tecnológicos y la aparición de aplicaciones en tiempo real como son la telemedicina, videoconferencia, transmisión de video, juegos online, e-learning, las cuales necesitan ciertos requerimientos para su correcto funcionamiento, esto se logra al aplicar técnicas de QoS a las redes inalámbricas, como lo describe el protocolo 802.11e .

El protocolo MAC del estándar IEEE 802.11 no provee ninguna forma de diferenciar los distintos tipos de tráficos, todos son tratados de la misma forma. A diferencia del 802.11e, en que se puede diferenciar que estaciones están o no utilizando QoS. El estándar 802.11e fue desarrollado para corregir los problemas que se presentan a la hora de brindar QoS en el estándar 802.11

El estándar 802.11e está conformado por un conjunto de técnicas, con las cuales podemos priorizar el tráfico, prevenir colisiones, evitar retraso de los paquetes; logrando mejorar el desempeño de las aplicaciones en tiempo real con garantías de QoS. Para ello implementa una nueva función en la capa MAC llamada *HCF*[67], con dos tipos de acceso EDCA[68] y HCCA[69], la primera fue diseñada para priorizar el tráfico, como lo hace los servicios diferenciados, y la segunda fue creada para soporta tráfico de la misma manera que los servicios integrados.

---

[66] **Fuente:** Idem.
[67] **HCF: Función de Coordinación Híbrida.-** permite sondear las estaciones mediante un período libre de contenciones.
[68] **EDCA: Acceso al Canal Distribuido Mejorado.-** diseñado para soportar la priorización de tráfico.
[69] **HCCA: Acceso Controlado al Canal HCF.-** diseñado para soportar tráfico parametrizado.

3.8.1 EDCA

Diseñado para mejorar el mecanismo DCF del estándar 802.11, proporcionando un método de acceso distribuido, con el que se diferencian los servicios entre diferentes tipos de tráfico. EDCA clasifica al tráfico en cuatro diferentes categorías de acceso como se ilustra en la Figura 3.14. El conjunto de parámetros EDCA incluye el tamaño mínimo de la ventana de contención (CWmin), el tamaño máximo de ventana de contención (CWmax), espacio arbitrario entre tramas (AIFS[70]), y el límite de oportunidades de transmisión (TXOPlimit).

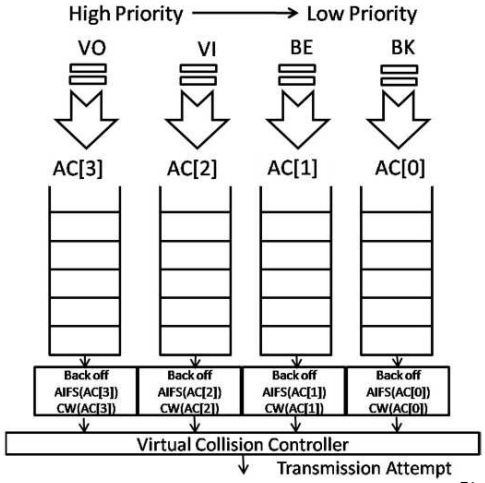

Figura 3.14. Descripción del proceso EDCA[71]

La Figura 3.15 muestra las operaciones de EDCA en el estándar 802.11e. Para lograr diferenciar, EDCA asigna la prioridad más alta a las categorías de acceso con menor CWmin, CWmax, y AIFS, para influir en el momento de acceder al medio.

---

[70] **AIFS: Espacio entre Tramas Arbitrario.-** tiempo que debe esperar una estación hasta poder decrementar su contador de backoff y que depende de la cola de tráfico.

[71] **Fuente:** Wen-Ping LAi y En-Chen Liou, A novel cross-layer desing using comb-shaped quadratic packet mapping for video delivery over 802.11e adhoc networks, 27/01/2013/ diponible en : <http://www.readcube.com/articles/10.1186/1687-1499-2012-59>

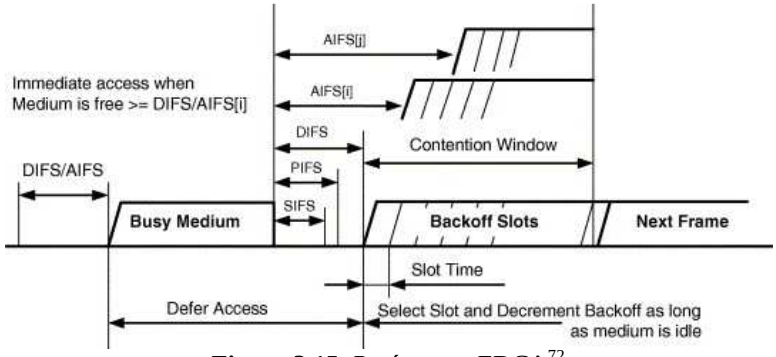

Figura 3.15: Parámetros EDCA[72]

- ***Espacio Arbitrario entre tramas (AIFS).-*** es el intervalo de tiempo mínimo disponible que debe tener el medio antes de empezar a transmitir. El valor del AIFS se lo obtiene con la siguiente formula:

$$\mathbf{AIFS(AC)= SIFS + AIFSN(AC)*SlotTime}^{73}$$

Dónde:

  - ***SlotTime y SIFS:*** representan parámetros conocidos de DCF correspondientes al tiempo de una ranura y el intervalo más corto entre tramas,
  - ***AIFSN[AC]:*** es el número de espacio entre tramas arbitrario y determina la prioridad. La prioridad más alta corresponde al valor más pequeño.

- ***Ventana de Contención (CW).-*** un número aleatorio para lanzar el mecanismo de espera aleatorio (*backoff*).
- ***Límite de Oportunidad de Transmisión (límite TXOP).-*** es la duración máxima que tiene una estación para poder transmitir

[72] **Fuente:** Hwangnam Kim, QoS provisioning in IEEE 802.11-complaint networks: Past, present and future, 27/01/2013, disponible en:
<http://www.sciencedirect.com/science/article/pii/S1389128606002726>

[73] **Fuente:** OLVER, Emilio, *Análisis de mecanismos de calidad de servicio para aplicaciones multimedia en IEEE 802.11e*, Tesis U.A.M Departamento de Ingeniería Eléctrica Área: Redes y Telecomunicaciones,México, 27/01/2013, disponible en:
<http://mcyti.izt.uam.mx/archivos/Tesis/Generacion2009/ICR_EmilioOlvera.pdf>

El estándar 802.11e señala que la capa MAC, es la encargada de clasificar los datos y de enviar un MSDU[74] a sus respectivas colas, las cuales compiten por acceder al EDCA-TXOP[75], a partir de aquí pueden ocurrir los siguientes casos:

- ***El medio está libre y hay una sola cola con información para transmitir:*** en este caso la EDCA-TXOP es adjudicada a esta cola y podrá transmitir la información almacenada.
- ***El medio está ocupado:*** tiempo que espera una trama para comprobar si el medio está disponible para transmitir, caso contrario tendrá que esperar nuevamente.

3.8.2 HCCA

Es un mecanismo de acceso al medio centralizado para proveer calidad de servicio basado en la parametrización HC[76], utiliza esta función para que las estaciones tengan acceso al medio, luego de detectar que el canal se encuentra libre durante un tiempo PIFS.

HCCA hereda reglas de PCF, introduce también algunas extensiones, el concepto principal de HCCA es la CAP[77], que consiste en un intervalo de tiempo formado por la concatenación de TXOPs-HCCA. Otra función importante de HCCA es el de control de admisión, en la que, una estación sin QoS para solicitar nuevo flujo de tráfico, envía un mensaje al HC, el cual está compuesto por los siguientes parámetros:

- Tasa de datos promedio
- Tamaño nominal de la MSDU

---

[74] **MSDU.- Unidad de Servicio de Datos de la capa MAC**.- recibe de la subcapa LLC una pila de protocolos

[75] **EDCA-TXOP.-** periodo que es obtenido utilizando el acceso al canal que se basa en la contención.

[76] **HC: Coordinador Híbrido.-** es un dispositivo encargado de decidir a qué estación le toca transmitir. Normalmente coincide con el QAP

[77] **CAP:** Fase de acceso controlada.

- Tasa mínima en la capa física
- Retardo
- Máximo intervalo de servicio

3.8.3 MEJORAS 802.11e MAC

Además de proveer las funcionalidades de EDCA y HCCA detalladas anteriormente, el estándar 802.11e contiene otras mejoras en la capa MAC.

- ***Ráfaga Libre de Contención (CFB).-*** Es el procedimiento en el que múltiples tramas pueden ser transmitidas en un TXOP obtenido mediante EDCA, siguiendo ciertas reglas.

  En los gráficos siguientes, se observa las diferencia que hay entre una red sin CFB, la Figura 3.16 y con CFB en la Figura 3.17.

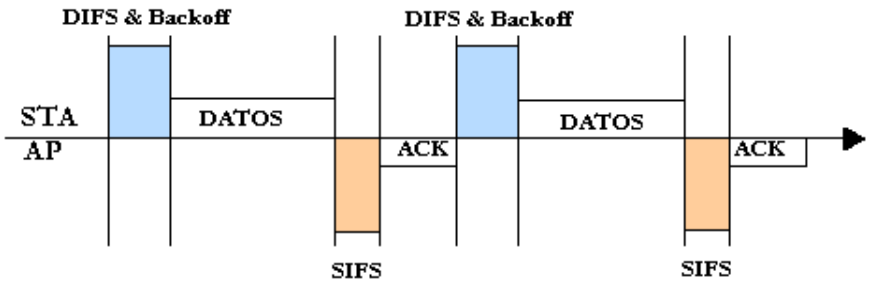

Figura 3.16: Red sin CFB aplicado[78]

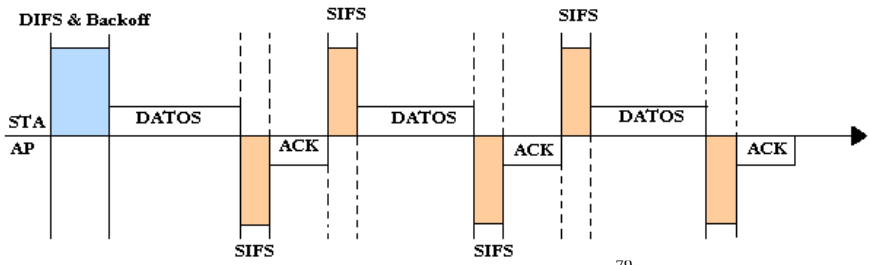

Figura 3.17: Red con CFB aplicado[79]

[78] **Fuente:** El Autor, tomado como referencia del artículo de :Tim Godfrey, Iside 802.11e: Making QoS Reality over WLAN Connections, 27/01/2013, disponible en: <http://www.design-reuse.com/articles/6865/inside-802-11e-making-qos-a-reality-over-wlan-connections.html>

- **Protocolo de enlace directo (DLP).-** permite que una estación envíe tramas de forma directa a otra estación que pertenece al mismo BSS, sin la necesidad de comunicarse a través de un punto de acceso.

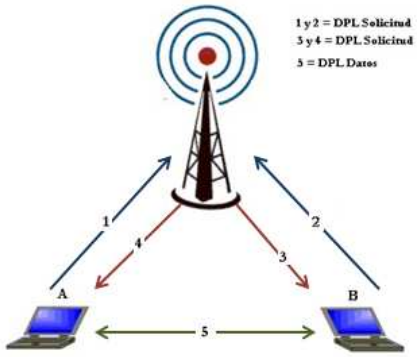

Figura 3.18: Proceso del protocolo de enlace directo[80]

- **Nuevas reglas de acuse de recibo.-** El estándar 802.11 enunciaba que; "para cada trama unicast se requiere una trama de control ACK"[81]. El estándar 802.11e agrega dos nuevas opciones las cuales son incorporadas en el campo de control de cada una de las tramas:

    - ***Sin ACK.-*** con este método no se necesita él envió de ACK'S con el objetivo de mejorar el desempeño de las aplicaciones.
    - **Bloque ACK.-** incrementa la eficiencia mediante el envío de un arreglo de varias ACK para varias tramas.

---

[79] **Fuente:** Idem.
[80] **Fuente:** Idem
[81] **Fuente:** MARRONE, Luis , y otros, **"***Nuevas reglas de acuse de recibo*", *Tecnologías Wireless y Movilidad en IPv4/IPv6* 1era Edición, Editorial de la Universidad Nacional de la Plata, Buenos Aires-Argentina, 2011

CAPITULO IV

4. DESARROLLO DE SCRIPT PARA LA SIMULACION

4.1 INTRODUCCIÓN

Debido al limitante acceso a equipos físicos, para realizar cualquier tipo de práctica referente a la creación o administración de una red, medir el ancho de banda, etc. El uso de herramientas de simulación nos ayuda a entender de una mejor manera, todo lo que puede ocurrir en una red.

El objetivo principal de este proyecto es dar conocer, los beneficios que se obtiene al utilizar herramientas para aplicar QoS sobre una red inalámbrica tipo Ad-hoc, como se vio en el Capítulo 2, una red Ad-hoc no necesita de un punto de acceso para que los host puedan transmitir información entre sí.

El NS-2 Simulator se utilizará como herramienta para el desarrollo de la parte práctica de la tesis, él cual es un software muy potente para la creación y simulación de redes inalámbricas. Con el simulador se crearán dos escenarios, que estarán conformados por una red ad-hoc, además, a uno de ellos se aplicará QoS. Luego de la simulación y con los datos obtenidos se podrá medir el desempeño de la red en cada escenario.

4.2 DESCRIPCIÓN NS-2

NS-2 es un simulador gratuito que suministra todo el código fuente. En el Anexo 1 se citará la dirección web de donde se puede descargar el software, ejemplos y manuales para su aprendizaje. El Ing. Miguel Herrera autor del libro: "NS2-Network Simulator", define al simulador de la siguiente manera:

> Network Simulator es un simulador de eventos discretos creado por la universidad de Berkeley para modelar redes de tipo IP. En la simulación se toma en cuenta lo que es la topología de la red y el tráfico de paquetes que posee la misma, con el fin de crear una especie de diagnóstico que muestre el comportamiento que se obtiene al tener una red con ciertas características.[82]

NS-2 permite simular redes Ethernet e inalámbricas, locales o vía satélite con una gran cantidad de protocolos de la capa de aplicación (ftp, cbr, http, etc.), de la capa de transporte (TCP, UDP, etc…) o de la capa de enlace de datos (como el MAC del tipo CSMA/CA); además permite trabajar en modo unicast o multicast y utilizar varios algoritmos para la planificación de colas. El NS-2 es una herramienta tan potente y a la vez configurable que resulta útil tanto en entornos de investigación como en entornos educativos.

4.2.1 Módulos Principales del NS-2

Los módulos principales del NS-2 que se han utilizado en este proyecto son el simulador ns, el xgraph y el nam.

---

[82] **Fuente:** GUANOCHANGA, Vinicio, *Empleo de la herramienta computacional ns2 para simular el comportamiento de una red de telecomunicaciones móviles celulares cuando se utiliza el protocolo ip móvil v6 (mipv6) en aplicaciones de voz*, 2009, Tomado de HERRERA, Miguel, NS2-NETWORK SIMULATOR

4.3.1.1 Simulador ns

Este simulador dispone de un núcleo principal, donde están definidos todos los protocolos, que está desarrollado en C++. En cambio, los scripts donde se configuran los escenarios de la simulación se deben programar en el lenguaje OTcl, que como C++ también es un lenguaje de programación orientado a objetos. Por lo tanto, para desarrollar en el simulador NS-2 es necesario programar en los dos lenguajes.

El simulador se invoca tecleando: *./ns nom_fichero.tcl* en la línea de comandos. Las simulaciones en el NS2 son scripts programados por el usuario, en donde, define la topología y características de los enlaces de los nodos que conformarán la red. Los resultados obtenidos en la simulación se pueden valorar utilizando las herramientas nam o xgraph que se explicarán a continuación.

4.3.1.2 Nam (Network Animator)

Herramienta que permite representar gráficamente la red diseñada. Además, permite visualizar dinámicamente los resultados de la simulación realizada. Se puede ejecutar de dos formas: escribiendo en la línea de comandos: *./nam nom_fichero.nam.*o también se puede ejecutar directamente dentro del código del archivo de simulación .tcl.

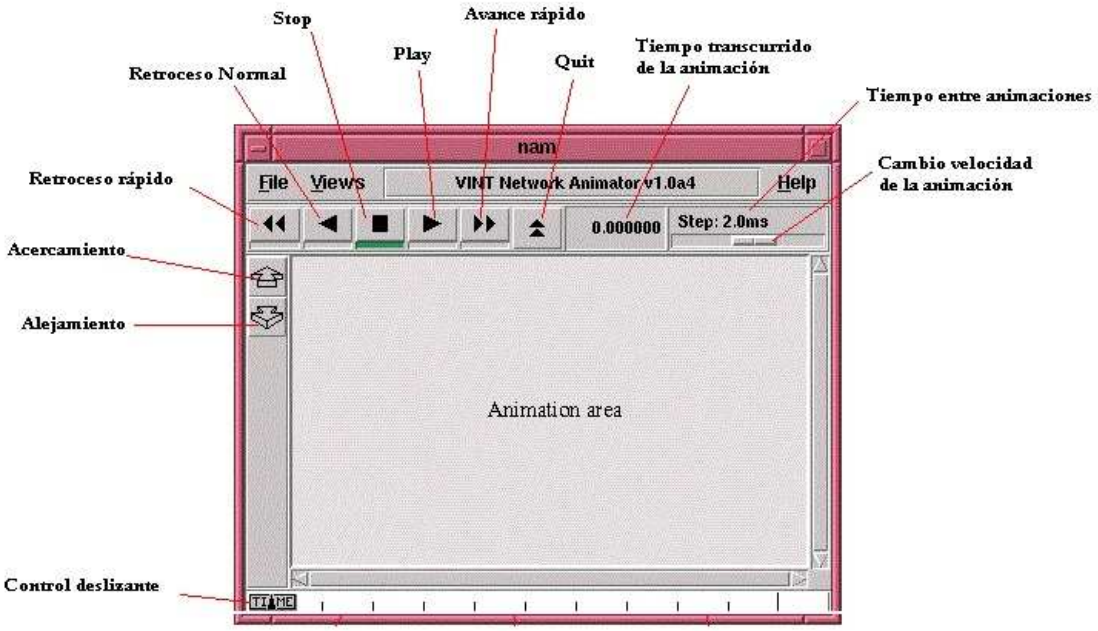

Figura 4.1: Nam[83]

En la figura 4.1 se observan las opciones de visualización del nam y se detallan a continuación:

- Retroceso rápido: la simulación se va retrocediendo multiplicando el paso del tiempo por 25.
- Retroceso normal: la simulación se retrocede según el paso del tiempo.
- Stop: detiene la simulación.
- Avance normal: se inicia la animación o la hace continuar si estaba pausada.
- Avance rápido: la simulación se va avanzando multiplicando el paso del tiempo por 25.
- Tiempo: indica el instante en el que se encuentra la simulación.
- Paso del tiempo: da idea de la velocidad de la simulación.
- Zoom: para aumentar o disminuir la simulación.
- Tamaño de los nodos, permite variar el tamaño de los nodos.
- Indicador de tiempo: da idea del tiempo transcurrido de la simulación.

[83] **Fuente:** El Autor, tomado como referencia del Tutorial NS2 creado por Marc Greis

- Flujo del enlace: si se pulsa un enlace y se selecciona la opción Graph se puede durante qué tiempo viajará información por ese enlace en ambas direcciones y la información que se pierde.

4.3.1.3 Xgraph

Xgraph es utilizado para generar gráficos bidimensionales de los datos generados en los archivos *.tr*. Esta herramienta ha sido de gran ayuda en este proyecto para poder visualizar las gráficas de los paquetes perdidos y delay obtenidas en las simulaciones. Se ejecuta tecleando en la línea de comandos: *./xgraph nom_fichero.tr.* Como el archivo .nam el xgraph también se la puede ejecutar desde el código del archivo de simulación.

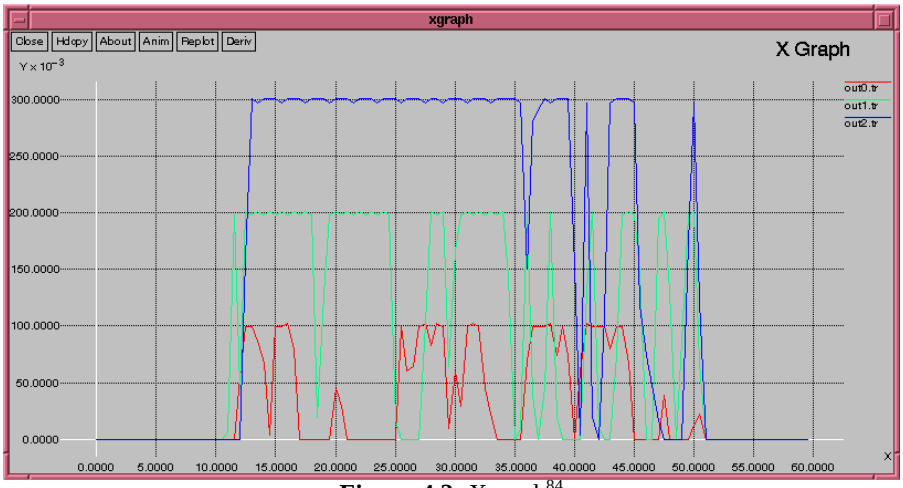

Figura 4.2: Xgraph[84]

[84] **Fuente:** Marc Greis, Tutorial NS2, 28/01/2013, disponible en: <http://www.isi.edu/nsnam/ns/tutorial/index.html>

4.4 MODIFICACIONES AL NS-2

El NS-2 trabaja con el estándar 802.11, pero, para desarrollar lo propuesto en este proyecto es necesario realizar modificaciones al código fuente del simulador, además se necesita instalar la librería 802.11e, así el simulador podrá trabajar con el estándar 802.11e.

4.4.1 Descripción Librería 802.11e

Para el desarrollo de esta tesis se utilizó la librería 802.11e, que fue elaborada por el Grupo de Redes de Telecomunicaciones de la Universidad Técnica de Berlín, en la cual, fue incorporado el manejo de prioridades desde el Simulador NS-2, por lo cual, este software es utilizado para el realizar investigaciones orientadas a la calidad de servicio.

Según los autores utilizaron la capa MAC previamente establecida este simulador, y la convirtieron en una capa MAC multidimensional, como resultado, apareció la librería 802.11e, la cual está conformada por varios archivos, pero, el principal es el *priority.tcl,* que es el encargado de manejar la prioridad del tráfico, la cual se la asocia con el campo *prio_field*, dentro de la estructura *hdr_ip*, que viaja en la cabecera de cada paquete. Más adelante se explicará el funcionamiento del campo *prio* dentro de nuestros scripts.

4.4.2 Modificaciones al código del NS-2

Como se explicó anteriormente para que el NS-2 trabaje con el estándar 802.11e, se descargará la librería, la cual se encuentra disponible en la web de los autores, el archivo es Patch for 802.11 model (ns-2.28)[85].

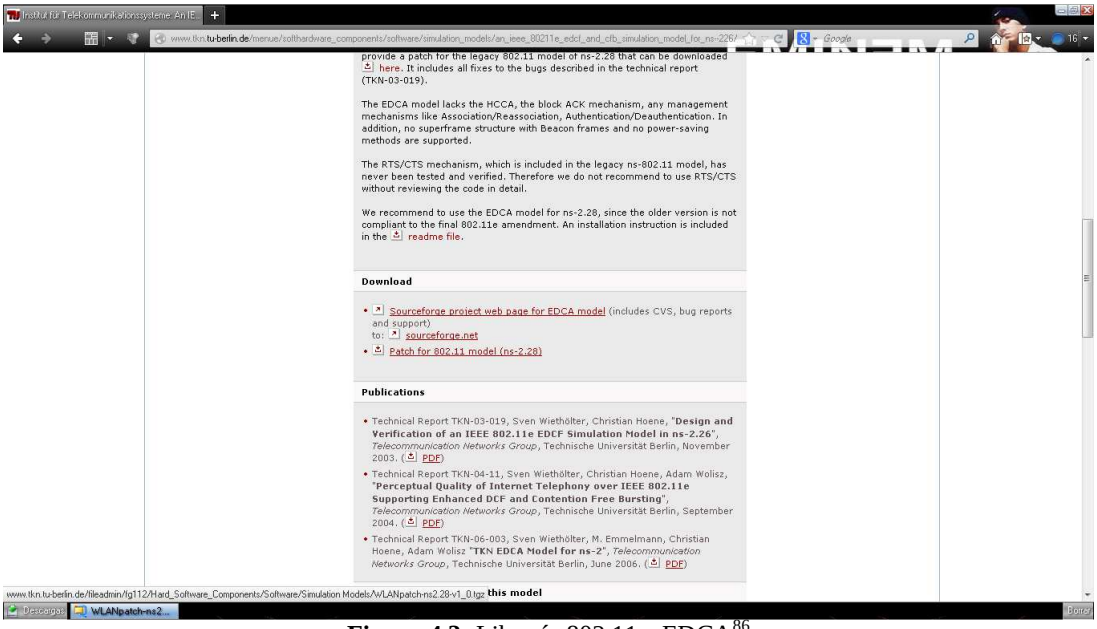

Figura 4.3: Librería 802.11e EDCA[86]

Luego de instalar nuestro simulador como se indica en el Anexo 1, y haber descargado la librería; para comenzar con la modificación del código, nos dirigimos al directorio donde se encuentra instalado el NS-2: */ns-allinone-2.28/ns-2.8*

[85] **Fuente:** http://www.tkn.tu-berlin.de/menue/softhardware_components/software/simulation_models/an_ieee_80211e_edcf_and_cfb_simulation_model_for_ns--226/

[86] **Fuente:** El Autor

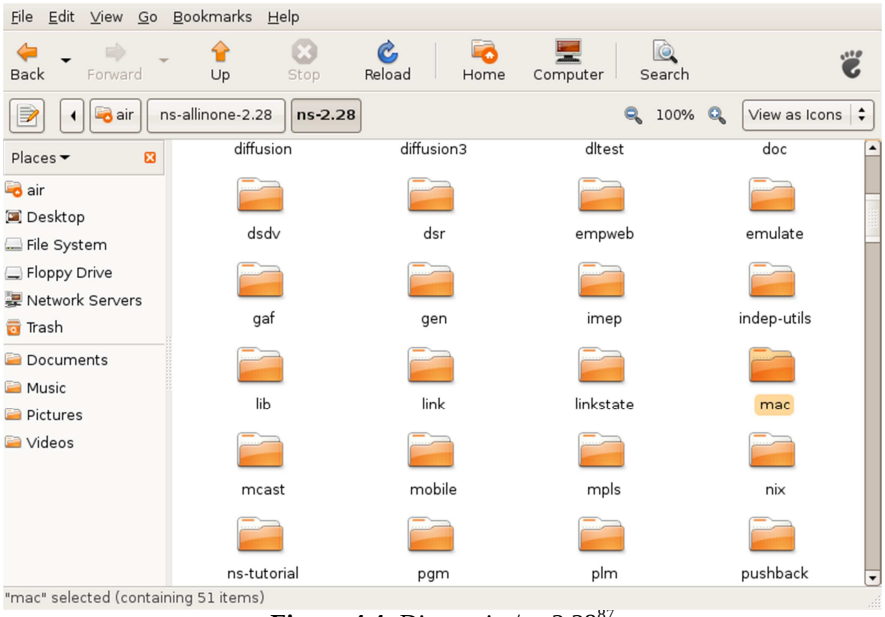

Figura 4.4: Directorio /ns-2.28[87]

Dentro de la carpeta *ns-2.28* encontraremos un directorio con el nombre mac, que debemos respaldarlo antes de comenzar con la modificaciones, luego procedemos a descomprimir la librería dentro del mismo.

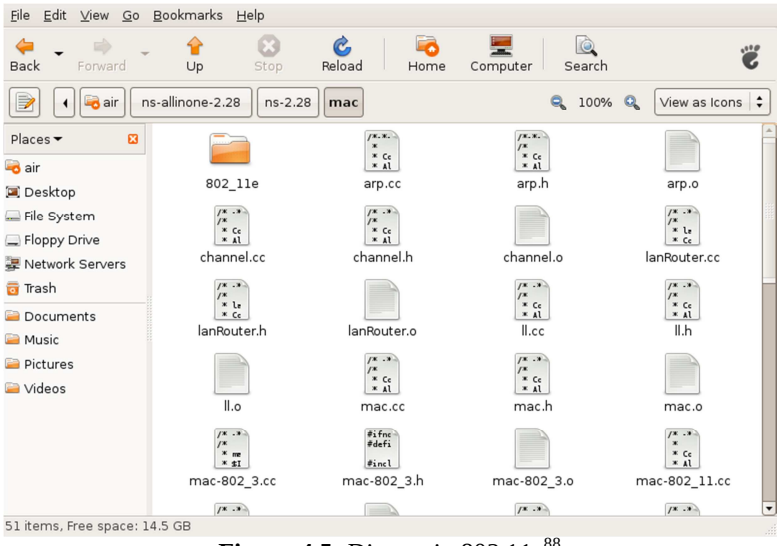

Figura 4.5: Directorio 802.11e[88]

[87] **Idem**
[88] **Fuente:** El Autor

Ya descomprimido el archivo dentro del directorio *./mac*, aparece una carpeta con el nombre 802.11e. En el carpeta */ns-2.28*, encontraremos un archivo con el nombre *Makefile.in*, que será el primero en ser modificado, el principal cambio que se hizo en este archivo fue el de instanciar el estándar 802.11e para poder utilizarlo en la simulación. En el Anexo 2 se mostrarán todo el código que fue agregado. Parte de las modificaciones su puede observar en la Figura 4.6

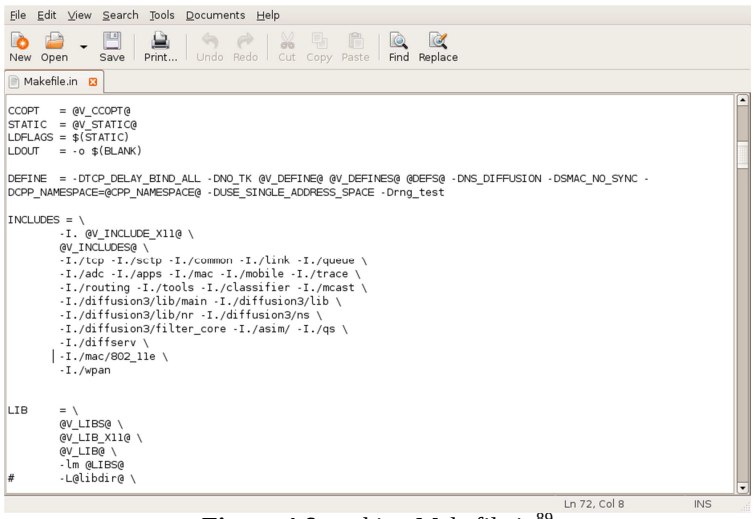

Figura 4.6: archivo Makefile.in[89]

Terminado la modificación de este archivo nos dirigimos al directorio *./tcl/lib*, en que encontraremos un archivo con el nombre de: *ns-lib.tcl.* Los creadors de la librería excluyen algunas líneas de programación, puesto que si no se lo hace al momento de ejecutar una simulación, el NS-2 no sabe con qué estándar va a trabajar, como se muestra en la Figura 4.7

[89] **Fuente:** El Autor

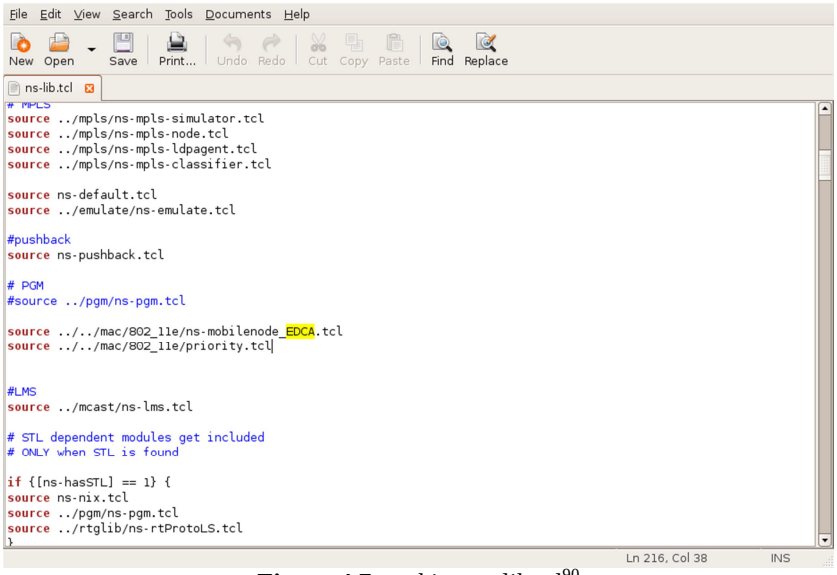

Figura 4.7: archivo ns-lib.tcl[90]

Nuevamente nos ubicamos en el directorio *./tcl/lib,* pero esta vez, se modificará el archivo *ns-default.tcl;* aquí vamos a inicializar variables como el tamaño del paquete, el tiempo de espera, el número de protocolos de enrutamiento, que trabajarán explicitamente con el estándar 802.11e. ver Figura 4.8

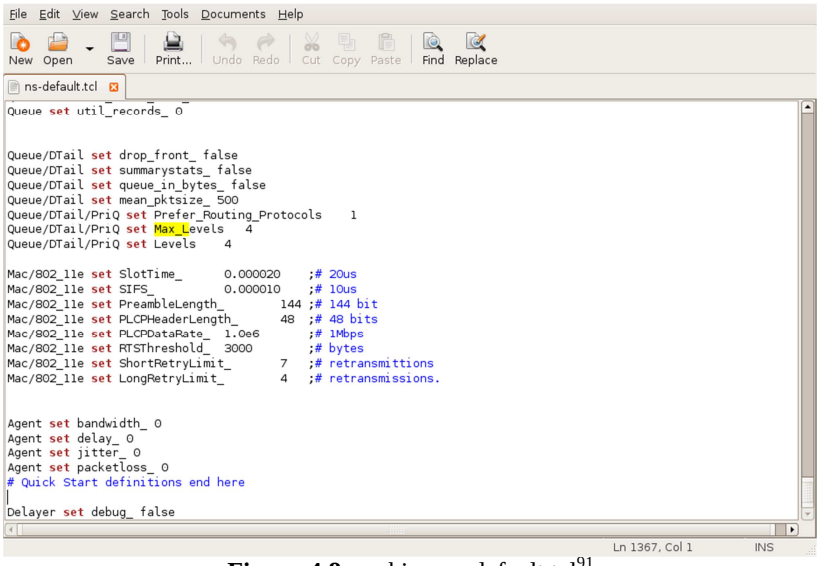

Figura 4.8: archivo ns-default.tcl[91]

[90] Idem
[91] **Fuente:** El Autor

Terminado de habilitar las nuevas variables, no dirigimos al archivo *./tcl/lan/ns-mac.tcl*, adaptaremos las características de la MAC/802.11 al estándar 802.11e. Como se observa en la Figura 4.9

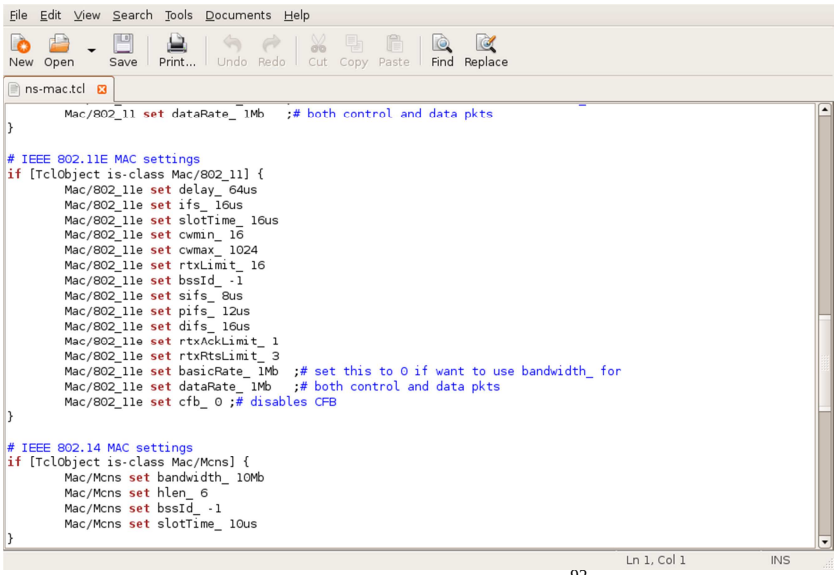

Figura 4.9: archivo ns-mac.tcl[92]

En el archivo *wireless-phy.h* se determinara el estatus del canal con el que se trabajara en las simulaciones. Como se muestra en la Figura 4.10

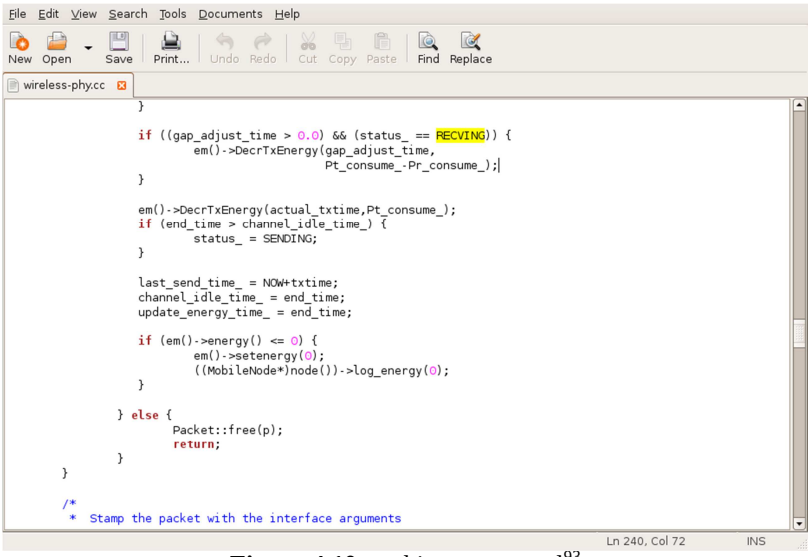

Figura 4.10: archivo ns-mac.tcl[93]

[92] **Fuente:** El Autor
[93] Idem.

En el Anexo 2 se puede observar todas las líneas de programación que fueron utilizadas para adaptar el NS-2 simulator al protocolo 802.11e

4.4.3 Transformar un script 802.11 en 802.11e

Luego de terminar con las modificaciones del punto anterior, ya podremos crear simulaciones que trabajen con el estándar 802.11e, para ello se deberá realizar el siguiente procedimiento dentro de nuestro script.

Lo primero que debemos hacer es indicar al simulador que se va a cambiar el tipo de Mac para trabajar con el estándar 802.11e, y además se indicará el tipo de cola, para que el simulador soporte colas con prioridad. Esto se hace modificando las variables ifq (tipo de cola) y mac (tipo de MAC):

```
set val (ifq)        Queue/DTail/PriQ;        #Tipo de Cola
set val(mac)         Mac/802.11e;             #Tipo de MAC
```

Dentro de los agentes de tráfico se debe indicar la prioridad que se tiene dentro de la simulación. Esto se indica con la variable prio_; después de la declaración del agente. Los valores de la prioridad están en el rango de 0 y 3, puesto que el estándar 802.11e establece solamente 4 colas de prioridad, en este caso siendo 0 el de mayor prioridad y 3 el de menor.

```
set udp_1 [new Agent/UDP]        #Creacion de Agente UDP
udp_1 set prio_ 0                #Estableciendo prioridad
```

4.5 PROCESO DE SIMULACIÓN

Para la realización de las simulaciones se deben definir ciertos parámetros dentro del escenario propuesto, como son, el modelo de propagación en el medio inalámbrico, las estaciones y los distintos tipos de tráfico.

4.5.1 Descripción del escenario

La simulación propuesta consta de dos escenarios, para la creación de los escenarios se utilizó el estándar IEEE 802.11, el cual detalla las características que debe cumplir una red inalámbrica. Como uno de los objetivos es la aplicación de QoS, esto se lo hará a uno de los escenarios, para ello utilizaremos las especificaciones del estándar 802.11e, específicamente la priorización de tráfico.

La red diseñada será inalámbrica tipo Ad-hoc que consta de 8 estaciones inalámbricas; que generarán tráfico en el lapso de 40 segundos. El objetivo principal es observar el comportamiento que presentan los diferentes tipos de tráficos en los dos escenarios, que se diferencian únicamente en la priorización de tráfico que se utilizará en uno de ellos, con lo cual se estará aplicando QoS sobre esa red, visto en el capítulo 3. La topología de nuestra red se puede observar en la Figura 4.11.

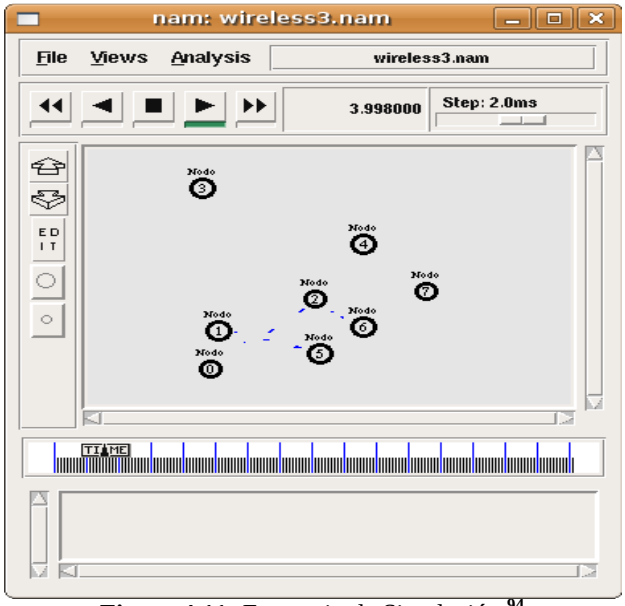

Figura 4.11: Escenario de Simulación [94]

La red manejará diferentes tipos de tráfico como son el ftp, VoIP, Video, siendo este último el objetivo principal de nuestro análisis. Como se explicará más adelante en este capítulo, los agentes de tráfico con los que trabajaremos para la generación de tráfico durante la simulación, serán TCP/FTP, TCP/CBR, UDP/CBR, además los nodos utilizarán el protocolo DSDV como protocolo de enrutamiento. En el Anexo 3 se adjuntará los scripts utilizados para la simulación.

A continuación se detallará las partes más importantes que conforman los scripts que vamos a utilizar para la simulación.

[94] **Fuente:** El Autor

4.5.1.1 Parámetros del canal inalámbrico

Los parámetros que se deben definir para el medio inalámbrico son comunes para todas las estaciones y son: el tipo de canal, el modelo de propagación, protocolo de capa MAC, el tipo de encolamiento, la característica de radiación de las antenas y protocolo de enrutamiento.

```
set val(chan)           Channel/WirelessChannel ;     # Tipo de canal
set val(prop)           Propagation/TwoRayGround ;    # Modelo de propagación
set val(netif)          Phy/WirelessPhy ;             # Tipo de interfaz
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)         50 ;                          # Max. número de paquetes
                                                      de cola
set val(nn)             9 ;                           # Numero de nodos móviles
set val(adhocRouting)   DSDV ;                        # Protocolo de enrutamiento
```

El tipo de canal, es lo que define si es alámbrico o inalámbrico, de acuerdo a esta selección se establecerán los parámetros del medio. Se utilizará dos modelos de encolamiento para el escenario que trabaja sin QoS se utiliza *DropTail* y para el segundo escenario con QoS utiliza el modelo *DTail* el cual utiliza cuatro buffers de diferente prioridad, en donde cada uno de ellos maneja su propio tráfico y algoritmo de *backoff*. La longitud de cada uno de los *buffers* es de 50 paquetes, esto significa que los paquetes excedentes serán descartados.

El protocolo de enrutamiento utilizado es el DSDV, en este protocolo los nodos vecinos se van enviando mensajes de enrutamiento, internamente se construye una tabla de enrutamiento en la que se van actualizando los cambios. Si se da la situación de que llega un paquete del que no se conoce el destino, se envía a los nodos vecinos un mensaje de solicitud de ruta y se retiene el paquete hasta que no se obtiene la respuesta.

A continuación se configuran los parámetros de la interfaz inalámbrica estableciendo la capacidad del canal correspondiente a 11 Mbps, la frecuencia ISM de 2.4 GHz y la potencia de 30 mW.

```
Phy/WirelessPhy      set CPThresh_ 10.0                  # Collision Threshold
Phy/WirelessPhy      set CSThresh_ 2.0e-14               # Carrier Sense Power
Phy/WirelessPhy      set RXThresh_ 1.77827941e-13# Receive Power Threshold
Phy/WirelessPhy      set bandwidth_ 11Mb                 # Capacidad del canal
Phy/WirelessPhy      set Pt_ 0.031622777                 # Potencia de transmisión
Phy/WirelessPhy      set freq_ 2.472e9                   # Frecuencia de transmsion
                                                         2.472GHz
Phy/WirelessPhy           set L_ 1                       # Perdida por trayectoria
```

Finalmente se configura los valores para la capa MAC. La velocidad de los datos por defecto en NS-2 es de 2 Mbps, por tanto, se debe cambiar este valor a 11 Mbps.

```
Mac/802_11           set dataRate_ 11.0e6                # Parámetros capa MAC
Mac/802_11           set basicRate_ 1.0e6                # Funcionamiento 802.11
```

4.5.1.2 Creación del escenario

En el caso de este proyecto o para simulaciones de redes inalámbricas de cualquier tipo se debe definir al inicio los parámetros antes mencionados. Una vez definidos, lo siguiente es crear el objeto del simulador mediante la siguiente línea de código:

```
set ns_ [new Simulator]
```

Creado el objeto, lo siguiente es crear los ficheros donde se guardarán las trazas y resultados de la simulación, estos archivos tienen extensión .tr y .nam

con los cuales podemos observar la simulación, y se los define de la siguiente manera:

```
set tracefd [open wireless3.tr w]
$ns_ trace-all $tracefd
$ns_ use-newtrace
```

La primera línea nos indica que se creará un archivo wireless3.tr y se le asigna un identificador tracefd; en la segunda línea se escribirán todos los datos de la simulación dentro del archivo, en la tercera línea se indica que el simulador utilizará el nuevo formato de trazas.

```
set namtrace [open wireless.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
```

El procedimiento es similar al descrito con el fichero de trazas, se crea un fichero wireless3.nam, cuyo identificador es namtrace; a continuación se almacenarán los datos de la simulación en este fichero.

A continuación se debe crear el área en el que se ejecutará la simulación, esto se consigue con una instancia, el procedimiento es parecido cuando creamos una nueva simulación.

```
set topo [new Topography]
$topo load_flatgrid $val (x) $val(y)
```

Como en el caso anterior, una vez definida la topología cualquier procedimiento que se refiera a ella debe empezar con $topo.

4.5.1.3 Creación de los nodos inalámbricos

Para la creación de los nodos móviles, se estableció los parámetros del canal inalámbrico a cada uno de ellos. Si la red que se va a simular es pequeña se puede crear uno a uno los nodos como se muestra a continuación, en cambio sí, se trabaja con redes que esté formada por más de 10 nodos, el trabajo se simplificará utilizando un bucle, previo a esto siempre se debe definir el número de nodos con el que se realizará la simulación.

```
# Create God
create-god $val(nn)
```

El objeto GOD[95] es utilizado para mantener la información global del estado de la red y sus nodos.

```
$ns node-config -adhocRouting $val(adhocRouting) \  #Protocolo de enrutamiento
-llType $val(ll) \                                  #Capa de enlace
-macType $val(mac) \                                #Capa Mac
-ifqType $val(ifq) \                                #Tipo de cola
-ifqLen $val(ifqlen) \                              #Núm. de paquetes en cola
-antType $val(ant) \                                #Tipo de antena
-propType $val(prop) \                              #Tipo de propagación
-phyType $val(netif) \                              #Tipo de interfaz
-channel $chan \                                    #Tipo de canal
-topoInstance $topo \                               #Tipo de topología
-wiredRouting ON \
-agentTrace ON \
-routerTrace OFF \
-macTrace OFF \
-movementTrace OFF

#Creacion de nodo de forma individual

set node_(nn) [$ns node]
$node_ (nn) random-motion 0
```

[95] **GOD.-** General Operations Director

Nota: al momento de crear los nodos de forma individual, NS-2 los guardará en un arreglo es decir que siempre el número del primer nodo será 0

```
set node_(0) [$ns node]
$node_ (0) random-motion 0
…….
set node_(nn -1) [$ns node]                    #donde nn es el número total de nodos
$node_ (nn -1) random-motion 0

#Creación de nodo utilizando bucles

for {set i 0} {$i < $num_mobile_nodes} {incr i} {     #Bucle para creación de nodos
set wl_node_($i) [$ns node]
$wl_node_($i) random-motion 0 ;                       # movimiento aleatorio
                                                      deshabilitado
puts "wireless node $i created ..."                   #Muestra mensaje
}
```

Después de la creación de los nodos, se debe especificar la posición que ocupará cada uno de ellos en un sistema de referencia, en el cual lo más importante es la distancia de separación entre los nodos. Estableciendo las coordenadas (x, y, z) para cada nodo, en este caso bidimensional (x, y, z=0), así por ejemplo:

```
$wl_node_(0) set X_ 5.0          #Posición en el eje X
$wl_node_(0) set Y_ 5.0          #Posición en el eje Y
$wl_node_(0) set Z_ 0.0          #Posición en el eje Z
```

Luego de haber deshabilitado el movimiento aleatorio con la sentencia random-motion 0, se les puede establecer un patrón de movimiento, con la sentencia setdest de la siguiente manera:

$ns_ at $time "$node_(i) setdest <x> <y> <velocidad>".

Esta sentencia expresa la siguiente: dentro del tiempo total de simulación a partir del momento $*time*, el nodo *i* se desplaza hasta las coordenadas (x, y), a una *velocidad* en m/s. a continuación se muestra una de las sentencias *setdest* de nuestro proyecto.

$ns_ at 1 "$node_(0) setdest 150 200 0.50"

Cuando transcurrió 1 segundo el nodo 0 comenzará a desplazarse a la nueva posición a una velocidad de 0.5m/s.

4.5.1.4 Creación y asociación de los agentes de tráfico

Los nodos por si solos no generan ni reciben tráfico, para hacerlo hay que crear los agentes. Una vez creados estos agentes se deben asociar a distintos nodos. De la implementación de estos agentes dependerán los protocolos de las capas de transporte y superiores. A continuación se detallarán los agentes utilizados para la simulación.

- **Agentes TCP/Sink.-** agente encargado de enviar los ACK's correspondientes a los paquetes que se van recibiendo de su agente emisor asociado.

set src_tcp4 [new Agent/TCP]	#Agente TCP origen
$src_tcp4 set class_ 2	#Diferenciar flujo de trafico
$src_tcp4 set fid_ 4	#ID del flujo de datos
$src_tcp4 set packetSize_ 1000	#Tamañono de paquete
set dst_tcp4 [new Agent/TCPSink]	#Agente TCP Destino
$ns_ attach-agent $wl_node_(2) $src_tcp4	#Vínculo del Agente TCP con el nodo origen
$ns_ attach-agent $wl_node_(0) $dst_tcp4	#Vínculo del Agente TCP con el nodo destino
$ns_ connect $src_tcp4 $dst_tcp4	#Vínculo entre origen y destino

```
set app4 [new Application/FTP]          #Creación de aplicación
$app4 set packetSize_ 1000              #Tamaño de paquete
$app4 attach-agent $src_tcp4            #Vínculo de la aplicación con el nodo
                                        destino
$ns_ at 3.0 "$app4 start"               #Iniciación de la aplicación
```

En este ejemplo se ha creado un agente TCP para el nodo 2 y para el nodo 0, un agente tipo *TCPSink*, ambos nodos inalámbricos se interconectan. El Agente *TCPSink* es receptor TCP, al nodo generador de tráfico se le asocia la fuente de tráfico del tipo FTP, algunas características se las puede dejar por defecto, o se las puede asignar un valor como por ejemplo el campo *class_*; o el campo *packetsize_*.

- **Agentes UDP.-** el transmisor es UDP y el receptor puede ser simplemente un agente *LossMonitor*. A los agentes también les podemos definir el protocolo de la capa de aplicación, a continuación se mostrará el protocolo utilizado por este agente en la simulación.

```
set src_udp3 [new Agent/UDP]            #Agente UDP origen
$src_udp3 set class_ 2                  #Diferenciación del flujo de tráfico
$scr_udp3 prio_ 0                       #Tipo de prioridad
$src_udp3 set fid_ 3                    #ID del flujo de datos
set dst_udp3 [new Agent/LossMonitor]    #Agente UDP destino
$ns_ attach-agent $wl_node_(7) $src_udp3  #Vínculo de agente UDP con el nodo
                                        origen
$ns_ attach-agent $wl_node_(2) $dst_udp3  #Vínculo de agente UDP con el nodo
                                        destino
$ns_ connect $src_udp3 $dst_udp3        #Vínculo entre origen y destino
set app3 [new Application/Traffic/CBR]  #Creación de la aplicación
$app3 set packetSize_ 1400              #Tamaño de paquete
$app3 set rate_ 3000Kb                  #Velocidad de transmisión
$app3 attach-agent $src_udp3            #Vínculo de la aplicación con el nodo
                                        destino
$ns_ at 5.0 "$app3 start"               #Inicio de la aplicación
```

Para el nodo 7 se crea un agente UDP, y para el nodo 2 se crea un agente tipo *LossMonitor,* él cual es un agente receptor con el cual podremos obtener algunas estadísticas sobre los paquetes recibidos y perdidos. Al nodo 2 se le asocia una fuente de tráfico tipo CBR, en la que fijamos la longitud de los paquetes a 1400 bytes, con una velocidad de transmisión de 3MB o 3000 Kb.

Dentro de la creación de los agentes, para utilizar la prioridad de tráfico, se debe instanciar una nueva variable llamada *prio_,* en la cual se establecerá el tipo de prioridad trabajará un determinado paquete durante la simulación. Transcurridos todos los eventos a los nodos se les comunica la finalización de la simulación con la siguiente sentencia.

```
for {set i} {$i < $val(nn)} [incr i]{
$ns_ at $val(stop).0 "$node_($i) reset";
}
```

4.5.1.5 Creación de los ficheros de traza para Xgraph

Para la creación de los ficheros debemos declarar nuevos ficheros .tr como lo hicimos al inicio cuando creamos el archivo wireless3.tr. La información de los nuevos ficheros se representará en unos gráficos similares al fichero de traza Nam. A continuación unos ejemplos:

```
set voz_ [open voz.tr w]
set video_ [open video.tr w]
```

Donde voz_ y video_ serán los nombres de los ficheros.

```
proc record {} {
global sink0 video_ voz_                    #variables
set ns_ [Simulator instance]                #obtenemos un instancia del
                                            simulador
set time 0.005                              #tiempo de muestreo
set bw0 [$sink0 set byte_]                  #cuantos bytes ha recibido
set now [$ns_ now]                          #obtenemos el tiempo actual
```

```
puts $f0 "$now [expr $bw0/$time*8/1000000]"      #calculamos el ancho de banda en MBits/s y escribimos en el fichero
$sink1 set bytes_ 0                              #reiniciamos los valores
$ns at [expr $now+$time] "record"                #reiniciamos el procedimiento
```

Con el código mostrado anteriormente se calcula el *goodput*[96]. El agente *LossMonitor* almacena en la variable *byte_* los bytes que se han recibido, los cuales se transforman a bits y se dividen entre la porción de tiempo definida en la variable time y se multiplica por 1M, estos datos se escriben en el fichero .tr creado, una vez escrito el valor, se resetea el valor de bytes_ y se vuelve a llamar a la función record.

4.5.1.6 Proceso final de programación para la simulación

Para terminar la simulación se utiliza el procedimiento *finish*, el cual se encargará de cerrar todos los ficheros de salida que se han abierto al inicio de la simulación y de ejecutar, si es necesario, alguna aplicación para el análisis de resultados. A continuación un ejemplo:

```
Proc finish {} {
        global ns tracefd namtrace voz_
        close $tracefd
        close $voz_
        exec nam wireless3.nam &
        exec xgraph voz_.tr
        exit 0
}
```

Se puede observar como primero se cierran todos los ficheros de trazas del NAM y de NS-2 y luego se ejecuta el fichero .nam y también se puede ejecutar el archivo .tr; y por último se ejecuta exit para salir de la aplicación. Finalizado el proceso y como última línea se agrega.

[96] **Goodput.-** es la cantidad de bits recogidos por el receptor en un intervalo de tiempo

$ns_ run

4.5.1.7 Inicio de la simulación

Escrita la última línea de programación en nuestra script, la cual indica nuestra script se ejecutar, al momento de nosotros a través de la consola digitamos el nombre de nuestra aplicación, anteponiendo *ns,* con lo cual se indica que es un script del programa NS2, como se muestra en la figura 4.12.

Figura 4.12: Ejecución de Programa mediante Línea de Comandos[97]

Después de escribir en la consola, la línea para ejecutar nuestra aplicación, tecleamos el botón *enter*, y se desplegarán en la pantalla las ventanas se se muestran en la Figura 4.13.

[97] **Fuente:** El Autor

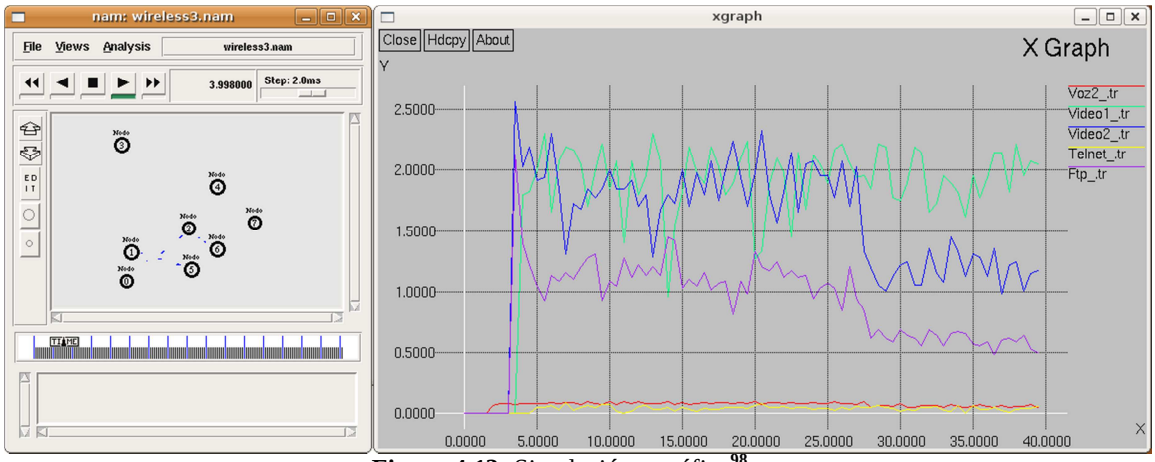

Figura 4.13: Simulación y gráfica[98]

Una de ellas es nuestra archivo *.nam* en la cual se apreciará la simulación de nuestra red, mientras que la otra ventana se observará las gráficas de nuestra simulación que se obtienen de nuestra archivo *.tr*. En la figura 4.14 se observa una parte del archivo wireless4a.tr, que se genera durante nuestra simulación.

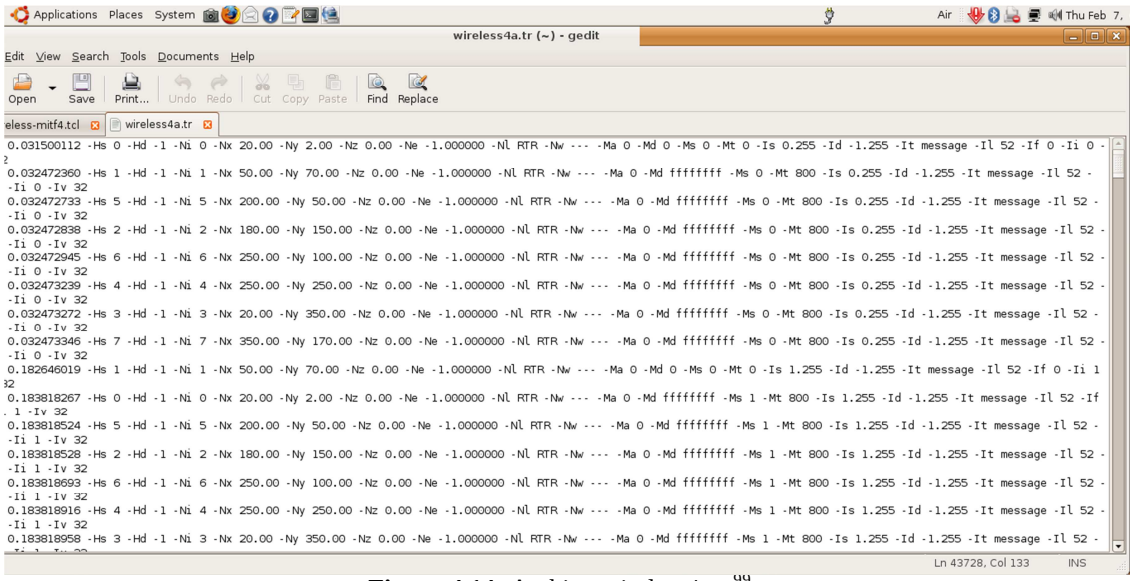

Figura 4.14: Archivo wireless4a.tr[99]

[98] **Idem.**
[99] **Fuente:** El Autor

CAPITULO V

5. SIMULACIÓN: OBTENCIÓN Y ANÁLISIS DE RESULTADOS

5.1 INTRODUCCIÓN

En este capítulo se analizarán los resultados obtenidos de las simulaciones, como se mencionó, el objetivo principal de este trabajo, es observar el desempeño que tuvieron las diferentes aplicaciones, pero sobre todo analizar el comportamiento que tuvo la aplicación de video, en los dos escenarios descritos en el Capítulo 4.

Para lo cual se realizaron dos pruebas, en la que modificamos la velocidad de transmisión de la aplicación de video, las cuales fueron de 1.5Mbits/s y 4.5Mbits/s, con lo cual pretendíamos utilizar todo el ancho de banda solo con esta aplicación, una de las técnicas para aplicar QoS sobre una red, es priorizar el tráfico, para nuestra simulación la aplicación de video tuvo la prioridad más alta.

El análisis de los resultados se enfocará más, en los datos obtenidos con el video, principalmente analizaremos lo referente a la perdida de paquetes y el delay; los cuales son factores que al momento de realizar una videoconferencia, afectan el desempeño de la misma.

5.2 SIMULACIÓN

5.2.1 Simulación N°1.- Red wireless con y sin QoS, velocidad de transmisión de video 1.5 Mbits/s

La única aplicación que varía su velocidad de transmisión será la de video, las otras aplicaciones tendrán velocidades que son asignadas por defecto por el simulador, en esta primera simulación se trabajó con un velocidad de transmisión para video de 1.5Mbits, en este escenario no aplicamos QoS, en la Figura 5.1 se observa el desempeño de las aplicaciones utilizadas en nuestra red.

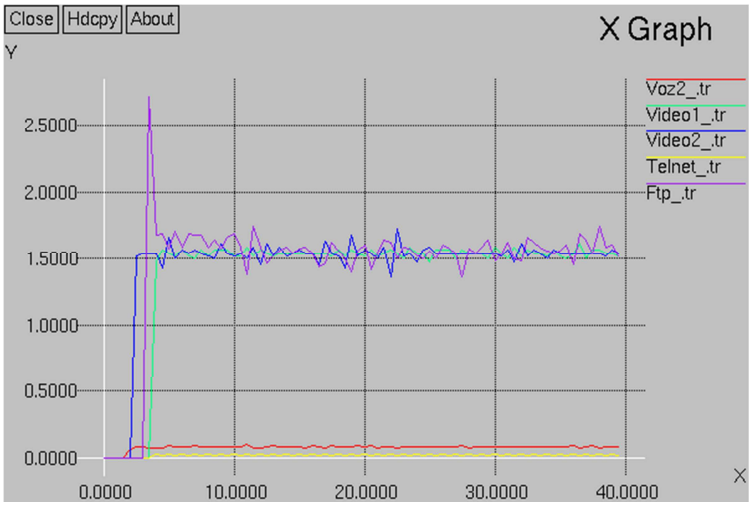

Figura 5.1: Red aplicada QoS y Video trabajando a 1.5Mbits/s[100]

Los datos obtenidos de la simulación se detallarán en la Tabla 5.1.

	VoIP	Video_1	Video_2	FTP	CBR
Tipo de Tráfico	UDP/CBR	UDP/CBR	UDP/CBR	TCP/FTP	TCP/CBR
Origen	0	1	6	2	4
Destino	3	5	2	0	6
Paquetes env.	1915	4876	5089	6809	184
Paquetes recib.	1915	4875	5088	6808	184
Paquetes perd.	0	1	1	1	0
Retardo (ms)	9.142	8.993	9.735	1.35690	7.15

Tabla 5.1: Datos obtenido de la Simulación N°1 sin QoS.[101]

[100] **Fuente:** El Autor
[101] **Idem**

Luego se haber analizado las tabla con los resultados de la primer simulación, se observa que existe la presencia de perdida de paquetes en alguna de las aplicaciones, que están trabajando en la red, aunque la cantidad de perdida no es muy elevada, esto se debe principalmente a que la velocidad de transmisión con la que trabajó el video es de 1.5Mbits, la cual no consume el ancho de banda en la que puede trabajar las redes inalámbricas según el estándar 802.11.

La Figura 5.2 corresponde al desempeño de las aplicaciones ya utilizadas en el escenario anterior la diferencia es, aquí se aplicó QoS, es decir se priorizó el tráfico de todas las aplicaciones, las aplicaciones de video tienen la prioridad más alta como se explicó en el Capítulo 3. Es por eso que su desempeño fue mejor que las aplicaciones de la simulación anterior.

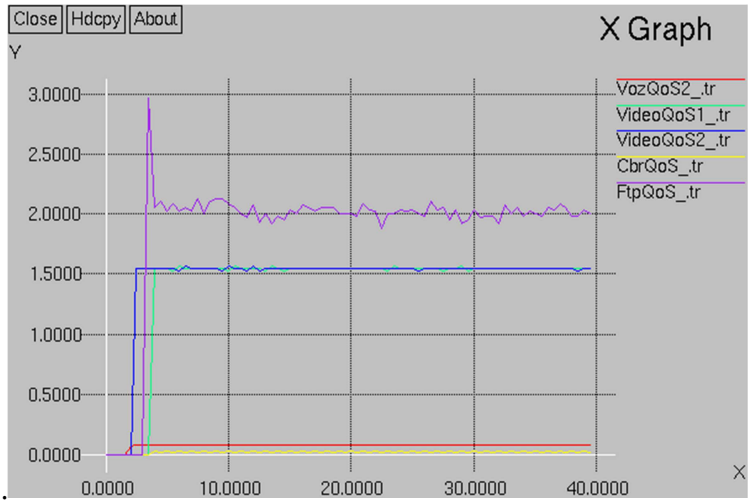

Figura 5.2: Red aplicada QoS y Video trabajando a 1.5Mbits/s[102]

En la tabla 5.3 se aprecia los resultados del gráfico anterior. En comparación con la tabla 5.1, la pérdida de paquetes disminuyó, las aplicaciones de Voz y una de las aplicaciones de Video ya no tienen perdida de información a

[102] **Fuente:** El Autor.

diferencia del escenario del primer escenario, se puede ver que al aplicar QoS, mejora considerablemente el desempeño de las aplicaciones ya mencionadas.

	VoIP	Video_1	Video_2	FTP	CBR
Tipo de Tráfico	UDP/CBR	UDP/CBR	UDP/CBR	TCP/FTP	TCP/CBR
Origen	0	1	6	2	4
Destino	3	5	2	0	6
Paquetes env.	1920	4889	5890	8718	0
Paquetes recib.	1920	4889	5889	8717	0
Paquetes perd.	0	0	0	1	0
Retardo (ms)	2.2547	2.4156	2.6344	22.804	0

Tabla 5.2: Datos obtenido de la Simulación N°1 con QoS.[103]

5.2.2 Simulación N° 2.- Red wireless con y sin QoS, velocidad de transmisión de video 4.5Mbits/s

La topología de las próximas simulaciones será la misma con que hemos venido trabajando la única diferencia es la velocidad de transmisión con la que trabajan las aplicaciones de video, en este caso utilizaremos una velocidad 4.5Mbits/s, que en el Ecuador, no es posible trabajar a esta velocidad, pero por motivos de estudio la utilizamos para ver el desempeño de la red.

La figura 5.3 se observa el comportamiento de la red, cuando esta no está utilizando técnicas de calidad de servicio, se observa las aplicaciones no tienen un desempeño, ya que ninguna logra, cierta estabilidad al momento de transmitir, esto debe ser producto, perdida de paquetes, cuellos de botella, saturación del ancho de banda de la red.

[103] **Fuente:** El Autor.

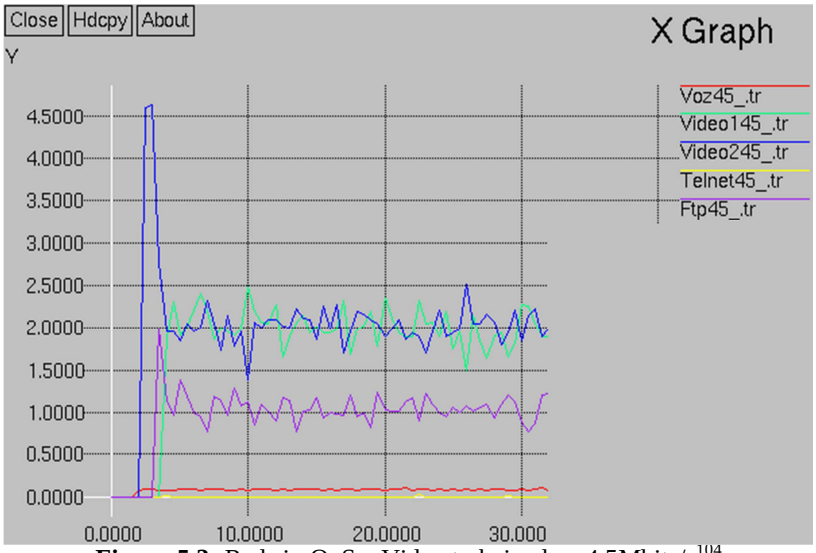

Figura 5.3: Red sin QoS y Video trabajando a 4.5Mbits/s[104]

En la tabla 5.3, se observan los valores obtenido de las aplicaciones durante la simulación.

	VoIP	Video_1	Video_2	FTP	CBR
Tipo de Tráfico	UDP/CBR	UDP/CBR	UDP/CBR	TCP/FTP	TCP/CBR
Origen	0	1	6	2	4
Destino	3	5	2	0	6
Paquetes env.	1917	6422	6778	4548	13
Paquetes recib.	2900	4183	4371	4547	13
Paquetes perd.	0	2239	2407	1	0
Retardo (ms)	14.369	28.056	26.889	18.427	8.6617

Tabla 5.3: Datos obtenido de la Simulación N°2 sin QoS.
Autor: El Tesista

Al igual que el primer escenario que trabajo sin QoS, los resultados de esta simulación son muy similares puesto que se observa la perdida de paquetes, el retardo que se presenta en las dos aplicaciones de video es muy elevado para este tipo de aplicaciones, esto se debe a que no logran alcanzar los 4.5Mbits/s que necesitan para su buen funcionamiento, esto resultado al momento de realizar una videoconferencia es desastroso, puesto momento de presentarse un 1.3% de perdida de paquetes, la imagen comienza a pixelarse.

[104] **Fuente:** El Autor.

En la figura 5.4 se observa la gráfica de la última simulación realizada la cual trabajo con QoS y la velocidad de las aplicaciones de video fue 4.5Mbits/s similar muestra el resultado de la simulación anterior pero sin aplicar QoS, la aplicaciones TCP tienen el peor desempeño, por tener baja prioridad en par acceder al medio.

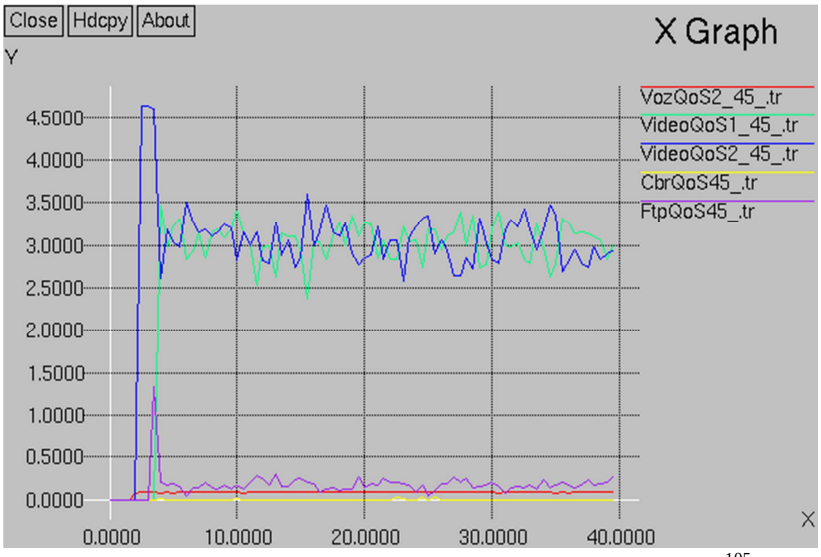

Figura 5.4: Red aplicada QoS y Video trabajando a 4.5Mbits/s[105]

La tabla 5.4 muestra los datos obtenidos de las aplicaciones durante la simulación.

	VoIP	Video_1	Video_2	FTP	CBR
Tipo de Tráfico	UDP/CBR	UDP/CBR	UDP/CBR	TCP/FTP	TCP/CBR
Origen	0	1	6	2	4
Destino	3	5	2	0	6
Paquetes env.	1920	9651	12277	800	25
Paquetes recib.	1920	8146	8913	799	25
Paquetes perd.	0	1505	3587	1	0
Retardo (ms)	8.290	242.683	300.698	90.0997	32.2583

Tabla 5.4: Datos obtenido de la Simulación N°2 con QoS.[106]

[105] **Fuente:** El autor
[106] Idem.

5.3 ANÁLISIS DE RESULTADOS

A continuación se analizarán los diferentes escenarios simulados y los resultados obtenidos en cada uno de ellos, en este punto nos enfocaremos más en el análisis del tráfico de video, dado que este es el punto principal por la que se desarrolló esta tesis. El análisis se lo llevará a cabo en los escenarios donde la velocidad de transmisión de video fue variando, comenzando por 1.5Mbits/s y por último 4.5Mbits/s.

5.3.1 Simulación N° 1

Se van a mostrar las dos gráfica de la simulación cuando la velocidad de transmisión de video fue de 1.5Mbits/s, a comparar los resultado obtenidos en lo referente a la perdida de paquetes durante la transmisión.

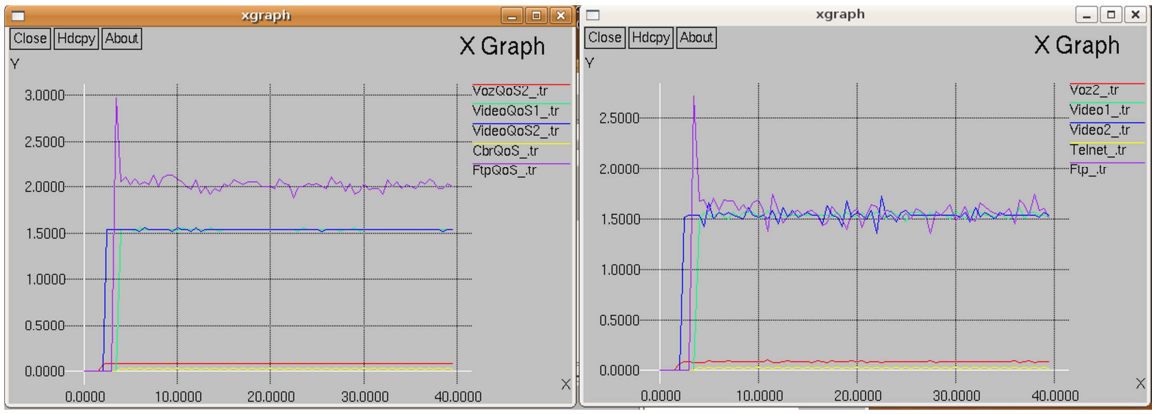

Figura 5.5: Comparación de Escenarios, conexión de video 2Mbits/s[107]

Como se puede observar en la figura 5.5, en el Escenario del lado izquierdo de la imagen se puede observar que el funcionamiento de las aplicaciones de video son las que tienen el mejor funcionamiento respecto de las otras aplicaciones esto debido, que, al momento de ponerles la prioridad más alta tuvieron acceso al medio de forma inmediata, con lo cual, la transmisión no

[107] **Fuente:** El Autor.

presento ningún problema, a diferencia de la imagen de la derecha en donde se observa que aplicaciones no logran mantener su velocidad de transmisión, se puede observar las líneas que representas a estas aplicaciones son muy irregulares, esto se debe a que ningún tráfico tiene prioridad sobre otro y todos quieren acceder al medio al mismo tiempo.

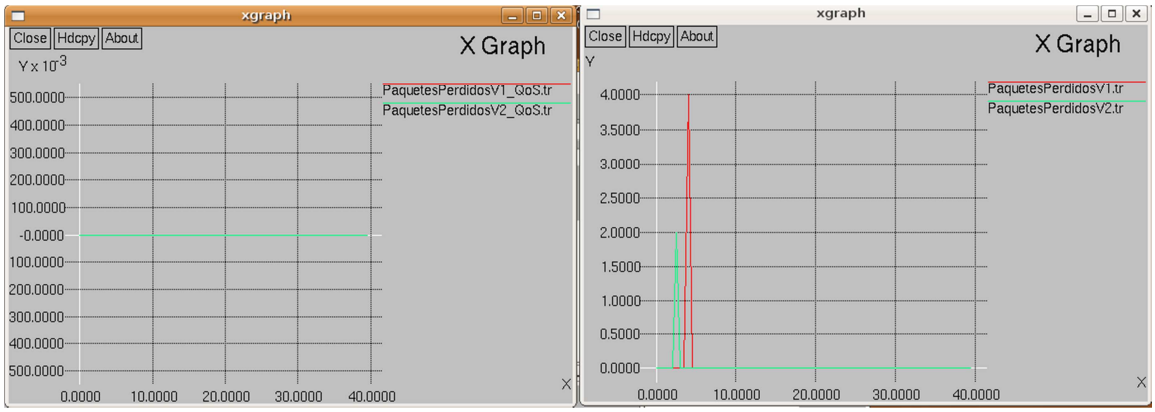

Figura 5.6: Comparación perdida de paquetes[108]

En la figura 5.6 se muestra de forma gráfica el resultado de la perdida de paquetes de las aplicaciones de video, en la imagen de la derecha observamos que las dos aplicaciones de video, presentan perdida de paquetes, los valores de la misma lo puede observar en la tabla 5.1; mientras que en el lado izquierdo de la imagen observamos que no existe perdida de paquetes, con lo cual se puede concluir que la utilización de QoS sobre redes, mejora el desempeño de las aplicaciones sobre todo de la que trabajan en tiempo real.

[108] **Fuente:** El Autor.

5.3.2 Simulación N° 2

En la vida real no existe ninguna aplicación hasta el día de hoy que utilice tanto ancho de banda, el objetivo de utilizar una velocidad de 4.5Mbits/s, fue conocer cuál sería el comportamiento de la aplicaciones en la red.

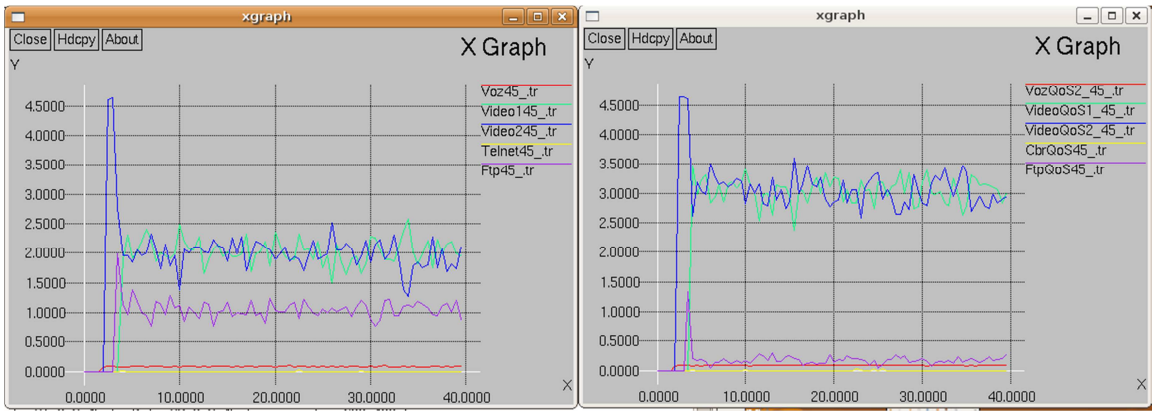

Figura 5.7: Comparación de Escenarios, conexión de video 4.5Mbits/s[109]

En la figura 5.7 se presentan las gráficas de esta simulación, como se puede apreciar son muy parecidas son muy parecidas en lo referente al tráfico; a diferencia de las simulaciones anteriores existen, perdida de paquetes, delay en lo concerniente transmisión de video, a pesar de haber dado prioridad como se señaló en el capítulo 4, en el escenario del lado derecho. Con el video trabajando a 4.5Mbits/s se puede observar que la red está saturada y debido a esto las conexiones de video no logran obtener el ancho de banda deseado, un claro ejemplo es cuando en una misma red hay varios equipos que están al mismo tiempo están viendo videos, los problemas que se presentan en el ejemplo citado es que ninguno de los videos fluye o se carga a la velocidad deseada.

[109] **Fuente:** El Autor.

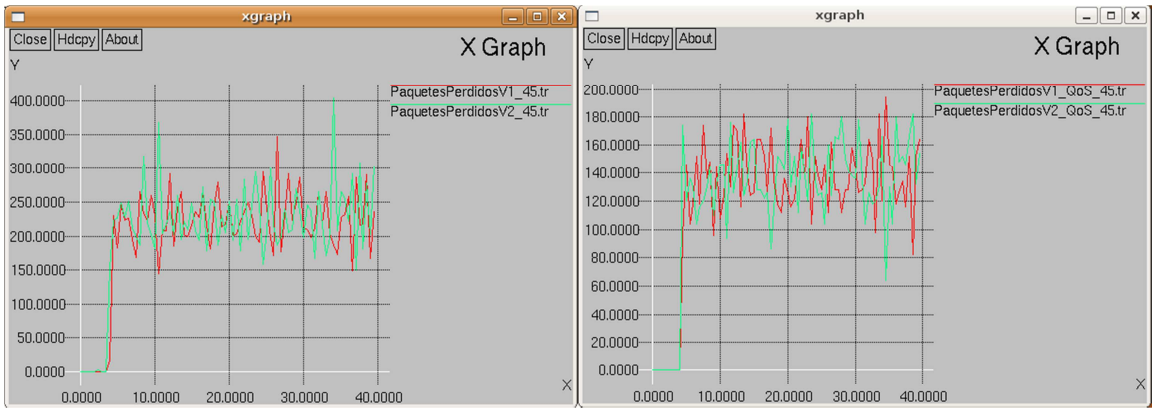

Figura 5.13: Comparación perdida de paquetes[110]

En la figura 5.13 se observa gráficamente que existe pérdida de paquetes en los dos escenarios, aunque en el escenario del lado derecho, se aplicó QoS la pérdida de paquetes disminuyó, aunque no es muy notable la diferencia, el porcentaje de perdida de paquetes del escenario del lado izquierdo es 30% mayor al escenario donde aplicamos QoS.

[110] **Fuente:** El Autor.

CAPITULO VI

6. CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

- Al momento de utilizar programas de simulación, no refleje exactamente lo que puede ocurrir en la realidad, pero el uso de los simuladores como son Packet Tracer, en este caso el NS-2, no ayudado a comprender el funcionamiento de los diferentes protocolos que se puede aplicar en una red.

- El estándar IEEE 802.11e, está orientado a proveer QoS, mediante el manejo de prioridades de acuerdos a las diversos tipos de tráfico, permitiendo disminuir los retardos en las comunicaciones inalámbricas, favoreciendo de esta manera las transmisiones en tiempo real. De esta forma se complementa al estándar IEEE 802.11, volviendo a las transmisiones inalámbricas seguras, confiables y accesibles.

- El suministrar QoS a redes del tipo Ad Hoc, que manejan diferentes tipos de tráfico, los cuales deben recibir tratamiento diferenciado por parte de los administradores de la red, sí, a esto se le suma la competencia por el acceso al medio y las consecuentes colisiones, no quedan dudas de la necesidad de brindar QoS. Los resultados obtenidos en este proyecto sirven para confirmarlo, al aumentar la carga de la red, los tráficos que tienen prioridades mayores pueden seguir obteniendo los requerimientos

solicitados mientras que los de menor prioridad disminuyen considerablemente su rendimiento.

- El protocolo de enrutamiento DSDV, es utilizado para redes pequeñas por su buen desempeño, es por eso que fue utilizado para la elaboración de las simulación, obteniendo como resultados, tiempos bajos en el retardo, menor perdida de paquetes.

- Para la utilización de aplicaciones en tiempo real, como es la videoconferencia la utilización de QoS es imprescindible. Como se señaló en el capítulo 3, cuando se trabaja sin prioridad de tráfico se presentan grandes retardos y pérdidas con el incremento de tráfico en la red. A diferencia que si priorizamos cierto tipo de trafico como por ejemplo la videoconferencia, se puede garantizar su correcto funcionamiento.

6.2 RECOMENDACIONES

- Es importante implementar lo propuesto en este trabajo en una situación real para obtener resultados más exactos de los que se obtuvieron con el NS-2 que es una herramienta de simulación.

- En escenarios donde diferentes tipos de tráfico lo cuales deben coexistan con el video se recomienda trabajar con QoS debido a que se deben diferenciar los paquetes de video dándoles prioridad, y evitando así una saturación de la red.

- Para aplicar QoS sobre una red inalámbrica, es necesario realizar un estudio y una planificación, para determinar cuáles son los requerimientos de la red, y las prioridades que se deben dar a los diferentes tipos de tráfico con los que se trabajan, para garantizar un mejor servicio.

- Para un trabajo futuro se sugiere realizar trabajar con los otros tipos de protocolos de enrutamiento utilizando el mismo escenario de simulación para analizar el comportamiento de la red y los resultados en lo que se refiere a perdida de paquetes, delay, etc.

- Se recomienda para un trabajo futuro analizar el comportamiento de una red mixta, dando mayor prioridad al tráfico TCP y que al UDP, para ver observar el comportamiento que tienen las aplicaciones en este tipo de redes.

ANEXO 1

INSTALACIÓN DEL NS-2

El simulador NS-2 ofrece dos versiones una para Microsoft Windows y la otra para Linux, su funcionamiento es mejor en este último. Por este motivo se decidió instalar el simulador en Ubuntu 8.4.

NS-2 se puede instalar de dos formas una es paquete por paquete es decir primero el simulador, luego el nam con el que vemos las animaciones y por el ultimo el xgraph otra de las herramientas con las que podemos observar las gráficas de nuestra simulación; y la otra opción es hacer la instalación todo a la vez, para el desarrollo de este proyecto utilizó la segunda manera de instalación, para descargarla nos dirigimos a la página web del programa NS2[111], elegimos en este caso la versión ns-allinone-2.28.tar.gz.

Una vez descargado el paquete realizamos los siguientes pasos:

1 Antes de comenzar en sí con la instalación del NS-2 debemos previamente descargar los siguientes paquetes:

- **build-essential**: *"es un paquete que contiene herramientas necesarias para la creación, compilación e instalación de programas"*[112]*.*
- **automake**: Herramienta con la que se generan archivos Makefiles que son compatibles con los estándares GNU.

[111] **Página Oficial de descarga.-** http://www.isi.edu/nsnam/ns/
[112] **Build-essential.-** http://www.nireleku.com/2011/09/instalar-las-build-essential-en-ubuntu/

- **autoconf**: Produce guiones del intérprete de comandos que configuran automáticamente el código fuente
- **libxmu-dev**: proporciona un conjunto de utilidades para las librerías que vamos a utilizar.

Mediante línea de comandos digitamos lo siguiente para que se instalen los paquetes ya mencionados.

sudo apt-get install build-essential autoconf automake libxmu-dev

2 Una vez instalado los paquetes, y previamente descargado el simulador digitamos la siguiente línea para descomprimirlo.

tar -xzvf ns-allinone-2.28.tar.gz

Nota: La ubicación del simulador la decide el usuario, para el desarrollo de esta tesis el software se lo ubicó en la carpeta ./home

3 Después de haber extraído el programa, nos dirigimos al directorio donde se encuentra, mediante línea de comando.

cd ns-allinone-2.28

4 Ya dentro del directiorio …./ns-allinone-2.28, digitamos la siguiente línea para que comience la instalación.

```
./install
```

5 Una vez terminada la instalación, el programa nos pide que ajustemos las ubicaciones de las diferentes aplicaciones que requiere el simulador. Dado que el NS-2 se puede ejecutar desde cualquier ubicación, el sistema operativo requiere ubicar los archivos y librerías de NS-2 y alojar su ubicación en el archivo **bash.bashrc,** para así poder ser utilizado desde cualquier parte. Para ello debemos ingresar al archivo bash de la siguiente manera:

```
cd
gedit ~/.bashrc
```

Dentro del archivo agregamos las siguientes sentencias:

```
# LD_LIBRARY_PATH
OTCL_LIB=/home/air/ns-allinone-2.28/otcl-1.11
NS2_LIB=/home/air/ns-allinone-2.28/lib
X11_LIB=/usr/X11R6/lib
USR_LOCAL_LIB=/usr/local/lib
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$
X11_LIB:$USR_LOCAL_LIB

# TCL_LIBRARY
TCL_LIB=/home/air/ns-allinone-2.28/tcl8.4.11/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB
```

```
# PATH
XGRAPH=/home/air/ns-allinone-2.28/bin:/home/air/ns-allinone-
2.28/tcl8.4.11/unix:/home/air/ns-allinone-2.28/tk8.4.11/unix
NS=/home/air/ns-allinone-2.28/ns-2.28/
NAM=/home/air/ns-allinone-2.28/nam-1.11/
PATH=$PATH:$XGRAPH:$NS:$NAM
```

Nota: la dirección **/home/air/** debe ser reemplazada por la ubicación de donde se encuentra el programa**.**

6 A continuación procedemos a salir del editor donde agregamos las líneas, guardando los cambios del archivo, para que el archivo se actualice escribimos lo siguiente en el terminal:

```
source ~/.bashrc
```

7 Por último se valida la instalación del NS-2:

```
cd ns-allinone-2.28/ns-2.28
./validate
```

ANEXO 2

INSTALACIÓN DE LA LIBRERÍA 802.11e

Para el desarrollo de este proyecto utilizó como guía, el documento y la librería 802.11e desarrollada por el Grupo de Redes de Telecomunicaciones de la Universidad Técnica de Berlín, la cual permite trabajar al NS2 Simulator con el estándar 802.11e. En el Capítulo 4 se dio a conocer la página web donde se puede descargar esta librería.

A lo largo de este Anexo se observará el código que se añadió a cada uno de los archivos que mencionados en el Capítulo 4. Todo este procedimiento se lo puede observar en el documento que viene junto con la librería al momento de descargarlos

1. Cambiar el archivo Makefile.in en ns-allinone-2,28/ns-2.28/.Escribir en la sección:

   - INCLUDES:

     -I./mac/802_11e ****

   - En OBJ_CC:

     mac/802_11e/mac-802_11e.o mac/802_11e/priq.o ****

     mac/802_11e/d-tail.o mac/802_11e/mac-timers_802_11e.o ****

   - En NS_TCL_LIB borrar:

     tcl/lib/ns-mobilenode.tcl ****

   - Escribir en NS_TCL_LIB:

     mac/802_11e/ns-mobilenode_EDCA.tcl ****

***mac/802_11e/priority.tcl ***

2. En el archivo ns-2.28/ns-2.28/tcl/lib/ns-lib.tcl, realizar las siguientes modificaciones:

- Borrar de source list:

  source ns-mobilenode.tcl

- Escribir en source list:

  source ../../mac/802_11e/ns-mobilenode_EDCA.tcl

  source ../../mac/802_11e/priority.tcl

3. Agregar en el archivo ns-2.28/ns-2.28/tcl/lib/ns-default.tcl, lo siguiente:

: ***Queue/DTail set drop_front_ false***

Queue/DTail set summarystats_ false

Queue/DTail set queue_in_bytes_ false

Queue/DTail set mean_pktsize_ 500

Queue/DTail/PriQ set Prefer_Routing_Protocols 1

Queue/DTail/PriQ set Max_Levels 4

Queue/DTail/PriQ set Levels 4

Mac/802_11e set SlotTime_ 0.000020 ;# 20us

Mac/802_11e set SIFS_ 0.000010 ;# 10us

Mac/802_11e set PreambleLength_ 144 ;# 144 bit

Mac/802_11e set PLCPHeaderLength_ 48 ;# 48 bits

Mac/802_11e set PLCPDataRate_ 1.0e6 ;# 1Mbps

Mac/802_11e set RTSThreshold_ 3000 ;# bytes

Mac/802_11e set ShortRetryLimit_ 7 ;# retransmittions

Mac/802_11e set LongRetryLimit_ 4 ;# retransmissions.

4. Agregar al archivo tcl/lan/ns-mac.tcl, las siguientes sentencias.

if [TclObject is-class Mac/802_11e] {

...

copy settings of MAC/802.11

(which are contained in this file) into this section

and at an "e" at the end of the "Mac/802_11" terms

...

Mac/802_11e set cfb_ 0 ;# disables CFB

}

Según los creadores de la librería, se deberá copiar toda la estructura del *if [TclObject is-class Mac/802_11]* que trabaja con el 802.11 y modificarlo para que trabaje con el estándar 802.11e, añadiendo la letra *e* al final de *Mac/802_11.*

5. En el archivo mac/wireless-phy.h, realizar el siguiente cambio:

enum ChannelStatus {IDLE, RECV, SEND};

a

enum ChannelStatus {IDLE, RECVING, SENDING};

6. Reemplazar en el archivo mac/wireless-phy.cc, lo siguiente:

RECV y SEND (por RECVING y SENDING).

7. En el archivo common/packet.h, escribimos lo que se detalla a continuación:

#define HDR_MAC802_11E(p) ((hdr_mac802_11e *)hdr_mac::access(p))

8. Una vez terminadas con todas las modificaciones, nos dirigimos al terminal y escribimos lo siguiente, y mediante línea de comandos nos ubicamos en el directorio donde está instalado nuestro simulador.
/home/air/ns-allinone-2.28/ns-2.28/ sudo ./configure; make clean; make depend; make

Ejecutada esta instrucción podremos comenzar a trabajar con el estándar 802.11e en nuestro simulador.

ANEXO 3

CÓDIGO DE PROGRAMACIÓN PARA NUESTRA SIMULACIÓN

- **ESCENARIO 1.-** el siguiente script fue utilizado para la simulación de la red sin aplicar calidad de servicio, con conexión de video de 1.5Mbits/s.

```
set val(chan)        Channel/WirelessChannel    ;# Tipo de Canal
set val(prop)        Propagation/TwoRayGround   ;# Modelo de propagacion
set val(netif)       Phy/WirelessPhy            ;# Tipo de interfaz
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)      50                         ;# Maximo numero de paquetes de cola
set val(nn)          8                          ;# Numero de nodos moviles
set val(rp)          DSDV                       ;# Protocolo de enrutamiento
set val(x)           400
set val(y)           400


Phy/WirelessPhy set CPThresh_ 10.0
Phy/WirelessPhy set CSThresh_ 2.0e-14
Phy/WirelessPhy set RXThresh_ 1.77827941e-13
Phy/WirelessPhy set bandwidth_ 11Mb
Phy/WirelessPhy set Pt_ 0.031622777
Phy/WirelessPhy set freq_ 2.472e9
Phy/WirelessPhy set L_ 1

#Parámetros de 802.11

Mac/802_11 set CWMin_ 31
Mac/802_11 set CWMax_ 1023
Mac/802_11 set SlotTime_ 0.000020
Mac/802_11 set SIFS_ 0.000010
Mac/802_11 set PreambleLength_ 144
Mac/802_11 set ShortPreambleLength_ 72
Mac/802_11 set PreambleDataRate_ 1.0e6
Mac/802_11 set PLCPHeaderLength_ 48
Mac/802_11 set PLCPDataRate_ 1.0e6
Mac/802_11 set ShortPLCPDataRate_ 2.0e6
Mac/802_11 set RTSThreshold_ 3000
```

```
Mac/802_11 set ShortRetryLimit_ 7
Mac/802_11 set LongRetryLimit_ 4
Mac/802_11 set dataRate_ 11.0e6
Mac/802_11 set basicRate_ 1.0e6

#Tamano de paquetes

Agent/TCP set packetSize_ 1400
Agent/UDP set packetSize_ 1400

# Inicializacion de variables globales

set ns_   [new Simulator]
$ns_ color 0 BLUE

set tracefd     [open wireless3.tr w]
$ns_ trace-all $tracefd

set Voz2 [open Voz2_.tr w]
set Video1_ [open Video1_.tr w]
set Video2_ [open Video2_.tr w]
set Telnet_ [open Telnet_.tr w]
set Ftp_ [open Ftp_.tr w]


# *** Paquetes Perdidos ***
set f6 [open pktlz1.tr w]
set f7 [open pktlz2.tr w]

# *** Delay ***
set f10 [open delayz1.tr w]
set f11 [open delayz2.tr w]

set namtrace [open wireless3.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
$ns_ use-newtrace


# set up topography object
set topo       [new Topography]

$topo load_flatgrid $val(x) $val(y)

# Create God
create-god $val(nn)

# Creacion del Canal
set chan_1_ [new $val(chan)]

# Configuracion nodo
$ns_ 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) \
                -topoInstance $topo \
                -wiredRounting ON \
                -agentTrace ON \
                -routerTrace ON \
                -macTrace OFF \
                -movementTrace OFF \
                -channel $chan_1_


set node_(0) [$ns_ node]
set node_(1) [$ns_ node]
set node_(2) [$ns_ node]
set node_(3) [$ns_ node]
set node_(4) [$ns_ node]
set node_(5) [$ns_ node]
set node_(6) [$ns_ node]
set node_(7) [$ns_ node]

$node_(0) random-motion 0
$node_(1) random-motion 0
$node_(2) random-motion 0
$node_(3) random-motion 0
$node_(4) random-motion 0
$node_(5) random-motion 0
$node_(6) random-motion 0
$node_(7) random-motion 0

#
# Coordenadas para los nodos
#
$node_(0) set X_ 20.0
$node_(0) set Y_ 2.0
$node_(0) set Z_ 0.0

$node_(1) set X_ 50.0
$node_(1) set Y_ 70.0
$node_(1) set Z_ 0.0

$node_(2) set X_ 180.0
$node_(2) set Y_ 150.0
$node_(2) set Z_ 0.0

$node_(3) set X_ 20.0
$node_(3) set Y_ 350.0
$node_(3) set Z_ 0.0

$node_(4) set X_ 250.0
$node_(4) set Y_ 250.0
$node_(4) set Z_ 0.0

$node_(5) set X_ 200.0
$node_(5) set Y_ 50.0
$node_(5) set Z_ 0.0
```

```
$node_(6) set X_ 250.0
$node_(6) set Y_ 100.0
$node_(6) set Z_ 0.0

$node_(7) set X_ 350.0
$node_(7) set Y_ 170.0
$node_(7) set Z_ 0.0

$node_(0) label "Nodo"
$node_(1) label "Nodo"
$node_(2) label "Nodo"
$node_(3) label "Nodo"
$node_(4) label "Nodo"
$node_(5) label "Nodo"
$node_(6) label "Nodo"
$node_(7) label "Nodo"


#Generacion de movimiento de los nodos

$ns_ at 4.0 "$node_(3) setdest 90.0 200.0 35.0"
$ns_ at 3.0 "$node_(0) setdest 90.0 160.0 25.0"
$ns_ at 3.0 "$node_(1) setdest 10.0 200.0 25.0"
$ns_ at 3.5 "$node_(7) setdest 70.0 50.0 25.0"
$ns_ at 3.5 "$node_(5) setdest 10.0 80.0 25.0"

# Generacion de Traficos entre nodos

# Conexiones TCP/FTP

set tcp1 [new Agent/TCP]
$tcp1 set class_ 2
$tcp1 set fid_ 4
$tcp1 set packetSize_ 1000
set sink5 [new Agent/TCPSink]
$ns_ attach-agent $node_(2) $tcp1
$ns_ attach-agent $node_(0) $sink5
$ns_ connect $tcp1 $sink5
set ftp1 [new Application/FTP]
$ftp1 set packetSize_ 1000
$ftp1 attach-agent $tcp1
$ns_ at 3.0 "$ftp1 start"

# Conexiones de Telnet

set tcp2 [new Agent/TCP]
$tcp2 set class_ 2
$tcp2 set fid_ 5
$tcp2 set packetSize_ 400
set sink4 [new Agent/TCPSink]
$ns_ attach-agent $node_(4) $tcp2
$ns_ attach-agent $node_(6) $sink4
$ns_ connect $tcp2 $sink4
set telnet2 [new Application/Telnet]
$telnet2 set packetSize_ 400
$telnet2 set interval_ 0.100
```

```
$telnet2 attach-agent $tcp2
$ns_ at 4.5 "$telnet2 start"

#Conexiones de Voz#


set udp2 [new Agent/UDP]
set sink1 [new Agent/LossMonitor]
$udp2 set class_ 2
$udp2 set fid_ 1
$ns_ attach-agent $node_(0) $udp2
$ns_ attach-agent $node_(3) $sink1
$ns_ connect $udp2 $sink1
set app2 [new Application/Traffic/CBR]
$app2 set packetSize_ 160
$app2 set interval_ 0.020
$app2 attach-agent $udp2
$ns_ at 2.0 "$app2 start"

#Conexiones de Video

set udp3 [new Agent/UDP]
set sink2 [new Agent/LossMonitor]
$udp3 set class_ 2
$udp3 set fid_ 2
$ns_ attach-agent $node_(1) $udp3
$ns_ attach-agent $node_(5) $sink2
$ns_ connect $udp3 $sink2
set app3 [new Application/Traffic/CBR]
$app3 set packetSize_ 1400
$app3 set rate_ 1500Kb
$app3 attach-agent $udp3
$ns_ at 4.0 "$app3 start"

set udp4 [new Agent/UDP]
set sink3 [new Agent/LossMonitor]
$udp4 set class_ 2
$udp4 set fid_ 3
$ns_ attach-agent $node_(6) $udp4
$ns_ attach-agent $node_(2) $sink3
$ns_ connect $udp4 $sink3
set app4 [new Application/Traffic/CBR]
$app4 set packetSize_ 1400
$app4 set rate_ 1500Kb
$app4 attach-agent $udp4
$ns_ at 5.0 "$app4 start"

for {set i 0} {$i < $val(nn)} {incr i} {
            $ns_ initial_node_pos $node_($i) 30
}

# Finalizacion de simulacion

for {set i 0} {$i < $val(nn) } {incr i} {
   $ns_ at 60.0000 "$node_($i) reset";
}
```

```
#inicializacion de variables para el cálculo de los paquetes perdidos y delay

set holdtime 0
set holdseq 0

set holdtime1 0
set holdseq1 0

set holdtime2 0
set holdseq2 0

set holdtime3 0
set holdseq3 0

set holdrate1 0
set holdrate2 0
set holdrate3 0
set holdrate4 0

$ns_ at 0.0 "record"
proc record {} {
global sink0 sink1 sink2  sink3 sink4 sink5 Voz1_ Voz2_ Video1_ Video2_ Telnet_ Ftp_  f4 f5 f6 f7
holdtime holdseq holdtime1 holdseq1 holdtime2 holdseq2 holdtime3 holdseq3 f8 f9 f10 f11 holdrate1
holdrate2 holdrate3 holdrate4
set ns_ [Simulator instance]
set time 0.50
set bw0 [$sink0 set bytes_]
set bw1 [$sink1 set bytes_]
set bw2 [$sink2 set bytes_]
set bw3 [$sink3 set bytes_]
set bw4 [$sink4 set bytes_]
set bw5 [$sink5 set bytes_]

set bw6 [$sink0 set nlost_]
set bw7 [$sink1 set nlost_]
set bw8 [$sink2 set nlost_]
set bw9 [$sink3 set nlost_]

set bw10 [$sink0 set lastPktTime_]
set bw11 [$sink0 set npkts_]


set bw12 [$sink1 set lastPktTime_]
set bw13 [$sink1 set npkts_]

set bw14 [$sink2 set lastPktTime_]
set bw15 [$sink2 set npkts_]

set bw16 [$sink3 set lastPktTime_]
set bw17 [$sink3 set npkts_]

set now [$ns_ now]
set j [expr $now/$time]
set distance [expr $j*$time]
```

```
set i [expr $now/$time]
set distance1 [expr $i*$time]

set k [expr $now/$time]
set distance2 [expr $k*$time]

set l [expr $now/$time]
set distance3 [expr $l*$time]

set m [expr $now/$time]
set distance4 [expr $m*$time]

set n [expr $now/$time]
set distance5 [expr $n*$time]

puts $Voz1_ "$distance [expr $bw0/$time*8/1000000]"
puts $Voz2_ "$distance1 [expr $bw1/$time*8/1000000]"
puts $Video1_ "$distance2 [expr $bw2/$time*8/1000000]"
puts $Video2_ "$distance3 [expr $bw3/$time*8/1000000]"
puts $Telnet_ "$distance4 [expr $bw4/$time*8/1000000]"
puts $Ftp_ "$distance5 [expr $bw5/$time*8/1000000]"

# Record Packet Loss Rate in File

puts $f4 "$now [expr $bw6/$time]"
puts $f5 "$now [expr $bw7/$time]"
puts $f6 "$now [expr $bw8/$time]"
puts $f7 "$now [expr $bw9/$time]"

# Record Packet Delay in File

        if { $bw11 > $holdseq } {
                puts $f8 "$now [expr ($bw10 - $holdtime)/($bw11 - $holdseq)]"
        } else {
                puts $f8 "$now [expr ($bw11 - $holdseq)]"
        }

        if { $bw13 > $holdseq1 } {
                puts $f9 "$now [expr ($bw12 - $holdtime1)/($bw13 - $holdseq1)]"
        } else {
                puts $f9 "$now [expr ($bw13 - $holdseq1)]"
        }

        if { $bw15 > $holdseq2 } {
                puts $f10 "$now [expr ($bw14 - $holdtime2)/($bw15 - $holdseq2)]"
        } else {
                puts $f10 "$now [expr ($bw15 - $holdseq2)]"
        }

        if { $bw17 > $holdseq3 } {
                puts $f11 "$now [expr ($bw16 - $holdtime3)/($bw17 - $holdseq3)]"
        } else {
                puts $f11 "$now [expr ($bw17 - $holdseq3)]"
        }

$sink0 set bytes_ 0
```

```
$sink1 set bytes_ 0
$sink2 set bytes_ 0
$sink3 set bytes_ 0
$sink4 set bytes_ 0
$sink5 set bytes_ 0

$sink0 set nlost_ 0
$sink1 set nlost_ 0
$sink2 set nlost_ 0
$sink3 set nlost_ 0

set holdtime $bw8
set holdseq $bw9

$ns_ at [expr $now+$time] "record"
}

$ns_ at 60.0000 "finish"
$ns_ at 60.0003 "puts \"NS EXITING...\" ; $ns_ halt"
proc finish {} {
   global ns_ tracefd namtrace Voz2_ Video1_ Video2_ Telnet_ Ftp_ f5 f6 f7 f9 f10 f11
   $ns_ flush-trace
          close $tracefd
          close $Voz1_
          close $Voz2_
          close $Video1_
          close $Video2_
          close $Telnet_
          close $Ftp_


          close $f5
          close $f6
          close $f7
          close $f9
          close $f10
          close $f11

exec nam wireless3.nam &
exec xgraph Voz1_.tr Voz2_.tr Video1_.tr Video2_.tr Telnet_.tr Ftp_.tr -geometry 800x400 &
exec xgraph pktlv1.tr pktlv2.tr pktlz1.tr pktlz2.tr -geometry 800x400 &
exec xgraph delayv1.tr delayv2.tr delayz1.tr delayz2.tr -geometry 800x400 &
exit 0
}
puts "Starting Simulation..."

$ns_ run
```

- **Escenario 2.-** el siguiente script es similar al anterior en lo que se refiere a generación de tráfico, pero en este se aplica QoS.

```
set val(chan)         Channel/WirelessChannel    ;#Channel Type
set val(prop)         Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)        Phy/WirelessPhy            ;# network interface type
set val(mac)          Mac/802_11e                ;# MAC type
set val(ifq)          Queue/DTail/PriQ    ;# interface queue type
set val(ll)           LL                         ;# link layer type
set val(ant)          Antenna/OmniAntenna        ;# antenna model
set val(ifqlen)       50                         ;# max packet in ifq
set val(nn)           8                          ;# number of mobilenodes
set val(rp)           DSDV                       ;# routing protocol
set val(x)                    600
set val(y)                    600

Phy/WirelessPhy set CPThresh_ 10.0
Phy/WirelessPhy set CSThresh_ 2.0e-14
Phy/WirelessPhy set RXThresh_ 1.77827941e-13
Phy/WirelessPhy set bandwidth_ 11Mb
Phy/WirelessPhy set Pt_ 0.031622777
Phy/WirelessPhy set freq_ 2.472e9
Phy/WirelessPhy set L_ 1

#Parámetros de 802.11
Mac/802_11 set CWMin_ 31
Mac/802_11 set CWMax_ 1023
Mac/802_11 set SlotTime_ 0.000020
Mac/802_11 set SIFS_ 0.000010
Mac/802_11 set PreambleLength_ 144
Mac/802_11 set ShortPreambleLength_ 72
Mac/802_11 set PreambleDataRate_ 1.0e6
Mac/802_11 set PLCPHeaderLength_ 48
Mac/802_11 set PLCPDataRate_ 1.0e6
Mac/802_11 set ShortPLCPDataRate_ 2.0e6
Mac/802_11 set RTSThreshold_ 3000
Mac/802_11 set ShortRetryLimit_ 7
Mac/802_11 set LongRetryLimit_ 4
Mac/802_11 set dataRate_ 11.0e6
Mac/802_11 set basicRate_ 1.0e6

#Parámetros de 802.11e

Mac/802_11e set dataRate_ 11.0e6
Mac/802_11e set basicRate_ 1.0e6
Mac/802_11e set backoff_mode_ 1
Mac/802_11e set RTSThreshold_ 3000
Mac/802_11e set PreambleLength_ 72

#Tamaño de paquetes
Agent/TCP set packetSize_ 1400
Agent/UDP set packetSize_ 1400
```

```
# Inicializacion de variables globales

set ns_               [new Simulator]

set tracefd     [open wireless3qos.tr w]
$ns_ trace-all $tracefd


set Voz2_ [open Voz2qos_.tr w]
set Video1_ [open Video1qos_.tr w]
set Video2_ [open Video2qos_.tr w]
set Telnet_ [open Telnetqos_.tr w]
set Ftp_ [open Ftpqos_.tr w]

# *** Paquetes Perdidos ***
set f6 [open pktlz1qos.tr w]
set f7 [open pktlz2qos.tr w]

# *** Delay ***
set f10 [open delayz1qos.tr w]
set f11 [open delayz2qos.tr w]

set namtrace [open wireless3qos.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

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

# Create God
create-god $val(nn)

# Create channel #1 and #2
set chan_1_ [new $val(chan)]

# Configuracion de nodos
$ns_ 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) \
                 -topoInstance $topo \
                 -wiredRounting ON \
                 -agentTrace ON \
                 -routerTrace OFF \
                 -macTrace ON \
                 -movementTrace OFF \
                 -channel $chan_1_

set node_(0) [$ns_ node]
set node_(1) [$ns_ node]
```

```
set node_(2) [$ns_ node]
set node_(3) [$ns_ node]
set node_(4) [$ns_ node]
set node_(5) [$ns_ node]
set node_(6) [$ns_ node]
set node_(7) [$ns_ node]


$node_(0) random-motion 0
$node_(1) random-motion 0
$node_(2) random-motion 0
$node_(3) random-motion 0
$node_(4) random-motion 0
$node_(5) random-motion 0
$node_(6) random-motion 0
$node_(7) random-motion 0

# Coordenadas para la simulacion

$node_(0) set X_ 20.0
$node_(0) set Y_ 2.0
$node_(0) set Z_ 0.0

$node_(1) set X_ 50.0
$node_(1) set Y_ 70.0
$node_(1) set Z_ 0.0

$node_(2) set X_ 180.0
$node_(2) set Y_ 150.0
$node_(2) set Z_ 0.0

$node_(3) set X_ 20.0
$node_(3) set Y_ 350.0
$node_(3) set Z_ 0.0

$node_(4) set X_ 250.0
$node_(4) set Y_ 250.0
$node_(4) set Z_ 0.0

$node_(5) set X_ 200.0
$node_(5) set Y_ 50.0
$node_(5) set Z_ 0.0

$node_(6) set X_ 250.0
$node_(6) set Y_ 100.0
$node_(6) set Z_ 0.0

$node_(7) set X_ 350.0
$node_(7) set Y_ 170.0
$node_(7) set Z_ 0.0


$ns_ at 4.0 "$node_(3) setdest 90.0 200.0 35.0"
$ns_ at 3.0 "$node_(0) setdest 90.0 160.0 25.0"
$ns_ at 3.0 "$node_(1) setdest 10.0 200.0 25.0"
$ns_ at 3.5 "$node_(7) setdest 70.0 50.0 25.0"
```

```
$ns_ at 3.5 "$node_(5) setdest 10.0 80.0 25.0"

# Generacion de Tráfico entre nodos

# Coneciones TCP/FTP
set tcp1 [new Agent/TCP]
$tcp1 set bandwidth_ 1000
$tcp1 set prio_ 3
$tcp1 set class_ 2
$tcp1 set fid_ 4
$tcp1 set packetSize_ 1000
set sink5 [new Agent/TCPSink]
$ns_ attach-agent $node_(2) $tcp1
$ns_ attach-agent $node_(0) $sink5
$ns_ connect $tcp1 $sink5
set ftp1 [new Application/FTP]
$ftp1 set packetSize_ 1000
$ftp1 attach-agent $tcp1
$ns_ at 3.0 "$ftp1 start"

# Conexiones de Telnet

set tcp2 [new Agent/TCP]
$tcp2 set delay_ 1000
$tcp2 set prio_ 2
$tcp2 set class_ 2
$tcp2 set fid_ 5
$tcp2 set packetSize_ 400
set sink4 [new Agent/TCPSink]
$ns_ attach-agent $node_(4) $tcp2
$ns_ attach-agent $node_(6) $sink4
$ns_ connect $tcp2 $sink4
set telnet2 [new Application/Telnet]
$telnet2 set packetSize_ 400
$telnet2 set interval_ 0.100
$telnet2 attach-agent $tcp2
$ns_ at 4.5 "$telnet2 start"


set udp2 [new Agent/UDP]
set sink1 [new Agent/LossMonitor]
$udp2 set delay_ 150
$udp2 set prio_ 0
$udp2 set class_ 2
$udp2 set fid_ 1
$ns_ attach-agent $node_(0) $udp2
$ns_ attach-agent $node_(3) $sink1
$ns_ connect $udp2 $sink1
set app2 [new Application/Traffic/CBR]
$app2 set packetSize_ 160
$app2 set interval_ 0.020
$app2 attach-agent $udp2
$ns_ at 2.0 "$app2 start"
```

```
#Conexiones de Video

set udp3 [new Agent/UDP]
set sink2 [new Agent/LossMonitor]
$udp3 set delay_ 200
$udp3 set prio_ 1
$udp3 set class_ 2
$udp3 set fid_ 2
$ns_ attach-agent $node_(1) $udp3
$ns_ attach-agent $node_(5) $sink2
$ns_ connect $udp3 $sink2
set app3 [new Application/Traffic/CBR]
$app3 set packetSize_ 1400
$app3 set rate_ 1500Kb
$app3 attach-agent $udp3
$ns_ at 4.0 "$app3 start"

set udp4 [new Agent/UDP]
set sink3 [new Agent/LossMonitor]
$udp4 set delay_ 200
$udp4 set prio_ 1
$udp4 set class_ 2
$udp4 set fid_ 3
$ns_ attach-agent $node_(6) $udp4
$ns_ attach-agent $node_(2) $sink3
$ns_ connect $udp4 $sink3
set app4 [new Application/Traffic/CBR]
$app4 set packetSize_ 1400
$app4 set rate_ 1500Kb
$app4 attach-agent $udp4
$ns_ at 5.0 "$app4 start"

for {set i 0} {$i < $val(nn)} {incr i} {
        $ns_ initial_node_pos $node_($i) 30
}

#inicializacion
set holdtime 0
set holdseq 0

set holdtime1 0
set holdseq1 0

set holdtime2 0
set holdseq2 0

set holdtime3 0
set holdseq3 0

set holdrate1 0
set holdrate2 0
set holdrate3 0
set holdrate4 0


$ns_ at 0.0 "record"
```

```
proc record {} {
global sink0 sink1 sink2  sink3 sink4 sink5 Voz1_ Voz2_ Video1_ Video2_ Telnet_ Ftp_ f4 f5 f6 f7
holdtime holdseq holdtime1 holdseq1 holdtime2 holdseq2 holdtime3 holdseq3 f8 f9 f10 f11 holdrate1
holdrate2 holdrate3 holdrate4
set ns_ [Simulator instance]
set time 0.50
set bw0 [$sink0 set bytes_]
set bw1 [$sink1 set bytes_]
set bw2 [$sink2 set bytes_]
set bw3 [$sink3 set bytes_]
set bw4 [$sink4 set bytes_]
set bw5 [$sink5 set bytes_]

set bw6 [$sink0 set nlost_]
set bw7 [$sink1 set nlost_]
set bw8 [$sink2 set nlost_]
set bw9 [$sink3 set nlost_]

set bw10 [$sink0 set lastPktTime_]
set bw11 [$sink0 set npkts_]

set bw12 [$sink1 set lastPktTime_]
set bw13 [$sink1 set npkts_]

set bw14 [$sink2 set lastPktTime_]
set bw15 [$sink2 set npkts_]

set bw16 [$sink3 set lastPktTime_]
set bw17 [$sink3 set npkts_]

set now [$ns_ now]
set j [expr $now/$time]
set distance [expr $j*$time]

set i [expr $now/$time]
set distance1 [expr $i*$time]

set k [expr $now/$time]
set distance2 [expr $k*$time]

set l [expr $now/$time]
set distance3 [expr $l*$time]

set m [expr $now/$time]
set distance4 [expr $m*$time]

set n [expr $now/$time]
set distance5 [expr $n*$time]

#Record Throughput

        puts $Voz2_ "$distance1 [expr $bw1/$time*8/1000000]"
        puts $Video1_ "$distance2 [expr $bw2/$time*8/1000000]"
        puts $Video2_ "$distance3 [expr $bw3/$time*8/1000000]"
        puts $Telnet_ "$distance4 [expr $bw4/$time*8/1000000]"
        puts $Ftp_ "$distance5 [expr $bw5/$time*8/1000000]"
```

```
# Record Paquetes perdidos

        puts $f4 "$now [expr $bw6/$time]"
        puts $f5 "$now [expr $bw7/$time]"
        puts $f6 "$now [expr $bw8/$time]"
        puts $f7 "$now [expr $bw9/$time]"

# Record Delay

    if { $bw11 > $holdseq } {
        puts $f8 "$now [expr ($bw10 - $holdtime)/($bw11 - $holdseq)]"
    } else {
        puts $f8 "$now [expr ($bw11 - $holdseq)]"
    }

    if { $bw13 > $holdseq1 } {
        puts $f9 "$now [expr ($bw12 - $holdtime1)/($bw13 - $holdseq1)]"
    } else {
        puts $f9 "$now [expr ($bw13 - $holdseq1)]"
    }

    if { $bw15 > $holdseq2 } {
        puts $f10 "$now [expr ($bw14 - $holdtime2)/($bw15 - $holdseq2)]"
    } else {
        puts $f10 "$now [expr ($bw15 - $holdseq2)]"
    }

    if { $bw17 > $holdseq3 } {
        puts $f11 "$now [expr ($bw16 - $holdtime3)/($bw17 - $holdseq3)]"
    } else {
        puts $f11 "$now [expr ($bw17 - $holdseq3)]"
    }


$sink0 set bytes_ 0
$sink1 set bytes_ 0
$sink2 set bytes_ 0
$sink3 set bytes_ 0
$sink4 set bytes_ 0
$sink5 set bytes_ 0

$sink0 set nlost_ 0
$sink1 set nlost_ 0
$sink2 set nlost_ 0
$sink3 set nlost_ 0

set holdtime $bw8
set holdseq $bw9

$ns_ at [expr $now+$time] "record"
}

# Finalizacion de la simulacion
#
for {set i 0} {$i < $val(nn) } {incr i} {
```

```
   $ns_ at 60.0000 "$node_($i) reset";
}
$ns_ at 60.0000 "finish"
$ns_ at 60.0003 "puts \"NS EXITING...\" ; $ns_ halt"
proc finish {} {
   global ns_ tracefd namtrace Voz1_ Voz2_ Video1_ Video2_ Telnet_ Ftp_ f5 f6 f7 f9 f10 f11
   $ns_ flush-trace
          close $tracefd
          close $Voz2_
          close $Video1_
          close $Video2_
          close $Telnet_
          close $Ftp_

          close $f4
          close $f5
          close $f6
          close $f7

          close $f8
          close $f9
          close $f10
          close $f11

exec nam wireless3qos.nam &
exec xgraph Voz2qos_.tr Video1qos_.tr Video2qos_.tr Telnetqos_.tr Ftpqos_.tr -geometry 800x400 &
exec xgraph pktlv1qos.tr pktlv2qos.tr pktlz1qos.tr pktlz2qos.tr -geometry 800x400 &
exec xgraph delayv1qos.tr delayv2qos.tr delayz1qos.tr delayz2qos.tr -geometry 800x400 &
exit 0
}

puts "Starting Simulation..."

$ns_ run
```

BIBLIOGRAFÍA

FUENTES DE INTERNET

3Com University, “Wireless LAN Fundamentals”, Disponible en Web: <http://es.scribd.com/doc/57717208/4/IEEE-802-11-Capa-Fisica>.

6SOS,”El Protocolo IPv6”, España, Disponible en Web: <http://www.6sos.org/documentos/6SOS_El_Protocolo_IPv6_v4_0.pdf>.

CISCO, “Configuring Quality of Sevice”, Disponible en Web: <http://www.cisco.com/en/US/docs/security/security_management/cisco_security_manager/security_manager/4.1/user/guide/rtqos.html>.

FERNANDEZ Fernando, “IPv6: La próxima generación de Internet”, Madrid-España, Disponible en Web: <http://mundopc.net/ipv6-la-proxima-generacion-de-internet/>.

GREIS Marc, “Tutorial for Network Simulator”, E.E.U.U, Disponible en Web: <http://www.isi.edu/nsnam/ns/tutorial>.

MILLÁN Ramón, “El protocolo IPv6”, España, Disponible en Web: <http://www.ramonmillan.com/tutoriales/ipv6_parte1.php>.

Oracle, “IPv6 en profundidad”, Disponible en Web: <http://docs.oracle.com/cd/E19957-01/820-2981/6nei0r181/index.html>.

PADILLA John y BECERRA Yazmin, “Manual Practicas con NS-2”, Colombia, Disponible en Web: <http://git.upbbga.edu.co>.

PALET Jordi, “Tutorial de IPv6: INTRODUCCIÓN”, Madrid, Disponible en Web: <
http://long.ccaba.upc.es/long/050Dissemination_Activities/jordi_palet_tutorialipv6introduccion.pdf>.

See-my.ip.com, “Direccionamiento IPv6 (RFC2373)”, Disponible en Web: <http://www.see-my-ip.com/tutoriales/protocolos/ipv6_direccionamiento.php>.

SIMAL Tomás, “Redes Wifi”, España, Disponible en Web: <http://recursostic.educacion.es/observatorio/web/es/cajon-de-sastre/38-cajon-de-sastre/961-monografico-redes-wifi>.

VALLE, Fernando, “Coexistencia de Redes WLAN & WPAN”, Disponible en Web: < http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/valle_i_lf/capitulo1.pdf>.

VERA Juan. “Instalar ns2 en Ubuntu”, Perú, Disponible en Web: <http://blog.pucp.edu.pe/item/37506/instalar-ns2-en-ubuntu>.

VILLANUEVA Félix, “Simulador ns2. Principios básicos”, Disponible en Web: <http://crysol.org/es/node/224>.

FUENTES BIBLIOGRÁFICAS

BARRERA Paola y GUERRA Mauricio, *Implementacion de Tunneling entre redes IPv4 e IPv6 para la empresa "NETXPERTS CONSULTING S.A."*, Tesis ESPE Facultad de Sistemas e Informática, Sangolquí, 13 Junio 2005, Disponible en Web: <http://repositorio.espe.edu.ec/handle/21000/406>

BARRIONUEVO Evelin y TAMAYO Viviana, *Análisis del desempeño de una red con tecnología WiFi para largas distancias en un ambiente rural de la Región Sierra*, Tesis ESPE Facultad de Eléctrica y Electrónica, Sangolquí, 2011, Disponible en Web: <http://repositorio.espe.edu.ec/handle/21000/2934>

GRACÍA Guillermo, *Análisis de la calidad de servicio QoS en IPV6, utilizando la arquitectura INSIGNIA y la tecnología de red BRAIN*, Tesis EPN Facultad de Electrónica y Telecomunicaciones, Quito, Junio 2007, Disponible en Web: <http://bibdigital.epn.edu.ec/handle/15000/4166>

GUANOCHANGA Galo, *Empleo de la herramienta computacional NS2 para simular el comportamiento de una red de telecomunicaciones móviles celulares cuando se utiliza el protocolo IP Móvil V6 (MIPV6) en aplicaciones de Voz*, Tesis EPN Facultad de Electrónica y Telecomunicaciones. Quito. Julio 2009, Disponible en Web: < http://bibdigital.epn.edu.ec/handle/15000/1639>

SANCHEZ Olga, *Implementación de un modelo de canal inalámbrico para redes 802.11 bajo el simulador ns-2*, Tesis Universidad Politécnica de Catalunya, España, 5 Septiembre 2005, Disponible en Web: <http://hdl.handle.net/20993.1/3784>.

SANCHO Nicolás, *Comprobación del estándar IEEE 802.11n utilizando un punto de acceso (AP AIRONET 1250 de CISCO)*, Tesis Escuela Politécnica Nacional, Quito, Septiembre 2009, Disponible en Web: < http://bibdigital.epn.edu.ec/bitstream/15000/1858/1/CD-2426.pdf>.

SANTIZO Alejandro, *Evolución de Redes Fijas del Protocolo IPv4 aIPv6 en Guatemala,* Tesis USAC Facultad de Ingeniería Escuela de Ingeniería en Ciencia y Sistemas, Septiembre 2003, Disponible en Web: < http://biblioteca.usac.edu.gt/tesis/08/08_0183_CS.pdf>

ZAVALA Angélica, *Estudio de QoS Sobre WLAN Utilizando el Estándar 802.11e Aplicado a transmisiones de Sistemas Multimediales en Tiempo Real*, Tesis ESPOCH Faculta de Informática y Electrónica, Riobamba - Ecuador. Disponible en Web < http://hdl.handle.net/123456789/328>.