
VNIVERSIDAD NACIONAL AVTNOMA DE MEXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA
DE LA COMPUTACIÓN

# ESTUDIO DEL COMPORTAMIENTO DE APLICACIONES MULTIMEDIA SOBRE IEEE 802.11e

T E S I S

QUE PARA OBTENER EL GRADO DE:

**MAESTRA EN INGENIERÍA DE LA COMPUTACIÓN**

**P R E S E N T A**

**MARTHA MARÍA MONTES DE OCA CÁLIZ**

**DIRECTOR DE TESIS:**
**DR. JAVIER GÓMEZ CASTELLANOS**

**MÉXICO, D.F** **2008**

A Carito,
a Tony, al Alex, a *Angie* y al *Fabi*.

# AGRADECIMIENTOS

"¡Estas cosas tendrán que ser!
Surgirá una raza más excelsa
de lo que el mundo ha conocido hasta ahora,
con la llama de la libertad en su alma y
la luz del conocimiento en sus ojos…"

*John Addicton*

Es mucha la gente a quien debo mi más sincero agradecimiento, estoy segura que cada uno de ellos sabe que quisiera incluirlos aquí, para dejar constancia escrita de tal sentimiento. A todos ellos les doy las gracias.

A lo largo de mi existencia, ha habido personajes que me han dejado una enseñanza tan, pero tan especial, que sin ellos, no sería quien soy: Caro, Migue y Lichín, les agradezco profundamente.

# Índice

# Índice de figuras

# Índice de tablas

# Capítulo 1 Introducción

## 1.1 Introducción

No ha pasado mucho tiempo desde que Internet llegó a nuestras vidas. La velocidad con la que se ha incrementado la aparición de nuevas tecnologías, ha permitido a su vez el desarrollo de nuevas aplicaciones. Dichas aplicaciones ahora son más demandantes en el uso del ancho de banda y aun más sensibles a los posibles retardos.

Durante los últimos años hemos sido testigos activos del gran crecimiento que han tenido las comunicaciones inalámbricas, las cuales han llegado para proporcionarnos dos grandes características: movilidad y flexibilidad en el acceso a redes. En México, por ejemplo, ya es algo común encontrar un equipo inalámbrico con conexión a Internet en algún hogar.

Aunado a este avance, ha crecido también el uso de aplicaciones multimedia, como la videoconferencia, la descarga de videos bajo demanda, radio por Internet, etc. No obstante el auge de estas aplicaciones no ha venido acompañado a la par de algún mecanismo que nos asegure un eficiente acceso a estos servicios, es decir, no se garantiza la calidad del servicio (QoS, *Quality of Service*). Poco se ha avanzado en este tema, si bien es cierto que podemos tener acceso a determinado ancho de banda, también es cierto que aún no es posible "asegurar" una adecuada disponibilidad de los recursos en el transcurso de toda la comunicación.

Las comunicaciones inalámbricas proveen múltiples posibilidades en cuanto a cobertura se refiere, es decir, podemos mencionar desde una comunicación vía *Bluetooth*, hasta sistemas que pueden cubrir casi todo el planeta tierra, como son aquellos basados en tecnología satelital. En esta tesis nos dedicaremos a la tecnología conocida como WiFi (*Wireless-Fidelity*), la cual hace referencia a un conjunto de estándares para redes inalámbricas definidos por el grupo IEEE[1] 802.11.

En el caso de las redes inalámbricas, las cuales se caracterizan por tener una gran dependencia de las condiciones del entorno y su limitada eficiencia en escenarios con múltiples estaciones móviles, el lograr calidad de servicio representa todo un reto, por demás interesante. Hasta el día de hoy las redes inalámbricas siguen careciendo del soporte para la calidad del servicio. Sin embargo, el grupo de trabajo IEEE 802.11e ha presentado diversas mejoras al actual estándar 802.11, con el objetivo de maximizar el rendimiento de la red de acceso, ajustando ciertos mecanismos para permitir una priorización de flujos de tráfico entre aplicaciones y estaciones.

---

[1] **IEEE**: Se refiere al Instituto de Ingenieros Eléctricos Electrónicos (*Institute of Electrical and Electronics Engineers*), la cual es una asociación técnico-profesional mundial sin fines de lucro, dedicada a la estandarización de nuevas tecnologías.

## 1.2 Objetivos

En esta tesis se plantea como objetivo general **estudiar el mecanismo de priorización de flujos de tráfico**, el cual es propuesto por el grupo de trabajo IEEE 802.11e; **comprobar de manera práctica, qué tanto mejora la calidad de servicio proporcionada a las aplicaciones de transmisión de voz y video con respecto al protocolo IEEE 802.11**, el cual no implementa categorías de acceso o priorización de flujos de tráfico.

Para soportar el objetivo principal, será necesario **estudiar cómo se comporta el tráfico de video y de voz al ser enviado por un medio inalámbrico con determinada prioridad**.

La necesidad de la movilidad en las oficinas así como la tendencia actual de tener una oficina en todo lugar o una oficina móvil, ha ocasionado que la flexibilidad en las comunicaciones sea indispensable. Además, también se ha vuelto necesario que las transmisiones de tráfico multimedia tengan cierta prioridad, por sobre el resto de los envíos para que su calidad no se vea disminuida. Por lo anterior, estudiar el comportamiento del tráfico de video y de voz cuando es asignada determinada prioridad a los nodos que lo envían, tiene un impacto importante para corroborar que el estándar 802.11e garantiza una mejora en la calidad de servicio.

Para lograr el objetivo, en este trabajo se llevarán a cabo diversas pruebas con distintos escenarios que incluyan, obviamente, estaciones móviles que soporten IEEE 802.11e y estaciones móviles que no lo soporten.

## 1.3 Estructura de la tesis

A continuación se presenta un breve resumen de cada capítulo de esta tesis con la intención de facilitar la localización de información dentro de la misma.

En el capítulo 2, se analiza la situación actual de las redes inalámbricas, para ello se profundiza en el funcionamiento del protocolo estándar 802.11, haciendo énfasis en la capa MAC[2]; se describen las dos funciones de acceso: DCF y PCF, así como las limitantes que tiene éste estándar para proporcionar calidad de servicio.

En el capítulo 3, se estudia la propuesta de IEEE 802.11e y al igual que en el capítulo 2 se analiza la capa MAC, las mejoras realizadas a ésta para el soporte de calidad de servicio y también se identifican las posibles limitantes.

En el capítulo 4, se menciona brevemente la herramienta de simulación utilizada para la realización de las pruebas; el tipo de escenarios que se utilizarán y la forma en que el 802.11e asigna calidad de servicio a los nodos.

En el capítulo 5, se explicará cómo se inyectó tráfico real al simulador y la herramienta utilizada para el manejo y configuración de las máquinas de virtuales.

En el capítulo 6, se elabora un resumen de los resultados obtenidos, el análisis de éstos y en el capítulo 7 se presentan las conclusiones.

[2] **MAC**: (*Medium Access Control*, Capa de Acceso al Medio). Capa que consiste en conjunto de reglas que determinan cómo acceder al medio, enviar datos y resolver las colisiones que se presenten.

# Capítulo 2 Redes IEEE 802.11

## 2.1 Familia IEEE 802

El estándar IEEE 802.11 es un miembro de la familia IEEE 802 y hace referencia a un conjunto de especificaciones para tecnología de redes inalámbricas. Dichas especificaciones se centran en las dos capas inferiores del modelo OSI ya que incorporan componentes físicos y de enlace de datos. En la figura 2.1 se muestran las relaciones existentes entre los diversos componentes de la familia IEEE 802, el estándar IEEE 802.11 y el lugar que ocupan en el modelo OSI.

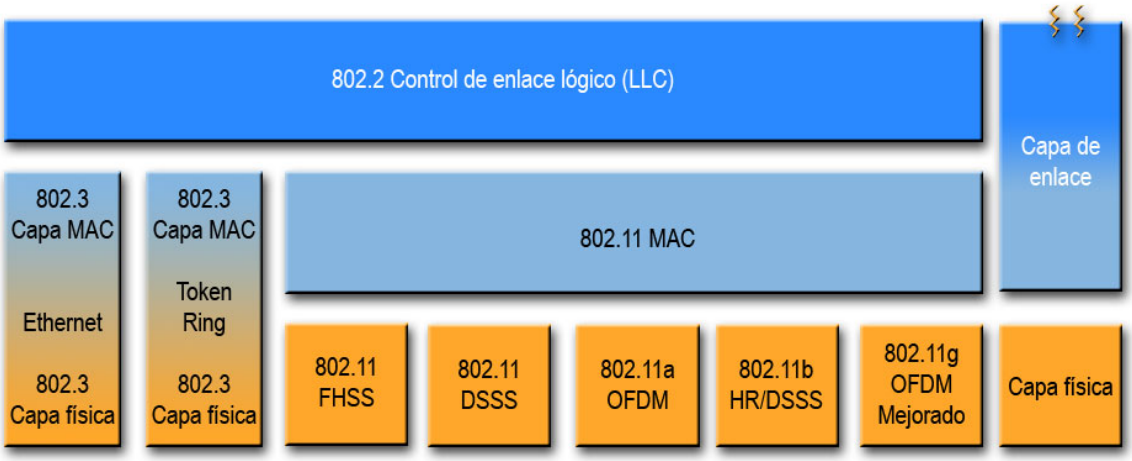



**Figura 2.1 La familia IEEE 802 y el estándar 802.11**

Todas las redes del tipo IEEE 802 tienen una capa MAC y una capa física (PHY). La capa MAC es un conjunto de reglas que determinan cómo acceder al medio, enviar datos y resolver las colisiones que se presenten. Los detalles de la transmisión y recepción los proporciona la capa física (PHY).

Las especificaciones individuales en la serie 802 se identifican por un segundo número. Por ejemplo, 802.3 es la especificación para las redes que utilizan la técnica de Acceso múltiple con escucha de portadora y detección de colisión (CSMA/CD, *Carrier Sense Multiple Access/Collision Detection*) a las cuales se les conoce como Ethernet. Otro ejemplo es 802.2, el cual especifica una capa de enlace común, el Control de Enlace Lógico (LLC, *Logical Link Control*). Un ejemplo más es 802.11, el cual revisaremos a continuación.

## 2.2 IEEE 802.11

IEEE 802.11 es, a su vez, otro conjunto de estándares y fue aprobado inicialmente por el IEEE en el año 1997. Dicho conjunto contempla todos los aspectos de una red inalámbrica (*Wireless LAN)*. En la tabla 2.1 se muestran el nombre y una breve descripción de cada uno de esos estándares.

- **Puntos de acceso (AP, *Access Point*)**: Son los dispositivos encargados de transmitir los paquetes o tramas entre las estaciones (en caso de que la red sea de infraestructura, la cual revisaremos más adelante), también se encarga de convertir las tramas *ethernet* (802.3) a tramas *wireless* (802.11) y viceversa para su entrega al equipo correspondiente (esto en caso de que el punto de acceso este conectado a una red *Ethernet*).
- **Medio inalámbrico**: Para mover las tramas de una estación a otra, el estándar utiliza un medio inalámbrico. La capa física que se utiliza es de radio frecuencia (RF, *Radio Frequency*).

**Tabla 2.1 Familia de estándares IEEE 802.11**

| Grupo de trabajo | Características |
|---|---|
| 802.11a | 5 GHz OFDM PHY |
| 802.11b | 2.4 GHz CCK PHY |
| 802.11c | *Bridging* 802.11 |
| 802.11d | Itinerancia internacional |
| 802.11e | Calidad de servicio (***Quality of Service, QoS***) |
| 802.11f | Interacción entre puntos de acceso |
| 802.11g | 2.4 GHz OFDM PHY |
| 802.11h | Anexos regulatorios a 5 GHz |
| 802.11i | Seguridad |
| 802.11j | Especificaciones japonesas a 5GHz |
| 802.11k | Medida de recursos de radio |
| 802.11m | Mantenimiento |
| 802.11n | PHY de gran capacidad |
| 802.11p | *Handover* en vehículos |
| 802.11r | *Roaming* rápido |
| 802.11s | Redes malladas |
| 802.11t | Predicción de rendimiento en redes inalámbricas |
| 802.11u | Uso conjunto con otras redes no-802 |
| 802.11v | Gestión de redes inalámbricas |
| 802.11x | Seguridad |

El estándar 802.11 permite el acceso a redes móviles y para conseguir esto, la capa MAC incorpora diversas opciones adicionales, de las cuales sólo se revisarán las necesarias para la comprensión de este trabajo.

### 2.2.1 Conceptos básicos

Para adentrarnos en la arquitectura de las redes IEEE 802.11, es necesario mencionar tres puntos clave y así entender mejor su funcionamiento:

- Elementos básicos de una red inalámbrica
- Tipos de redes inalámbricas
- El medio inalámbrico y las bandas ISM

#### 2.2.1.1 Elementos básicos de una red inalámbrica

La redes 802.11 están compuestas por tres componentes físicos, que se resumen en la figura 2.2

- **Estaciones**: Dispositivos informáticos con interfaces de red inalámbricas.

sí. Los BSS son de dos tipos, ilustrados en la figura 2.3


**Figura 2.2 Elementos básicos de una red inalámbrica**

#### 2.2.1.2 Tipos de redes inalámbricas

La base de una red IEEE 802.11 se le conoce como Conjunto de servicios básicos (BSS, *Basic Service Set*), el cual es simplemente un grupo de estaciones que se comunican entre


**Figura 2.3 Tipos de redes inalámbricas**

- **Independientes**: (IBSS, *Independent BSS*). Las estaciones en un IBSS se comunican directamente entre sí, por lo que deben encontrarse dentro del alcance directo de la comunicación. A este tipo de red también se le conoce como red ***Ad-hoc***.
- **De infraestructura**: Se distinguen de las redes independientes porque utilizan un dispositivo conocido como punto de acceso. Así, si una estación móvil en una BSS de infraestructura necesita comunicarse con una segunda estación móvil, la comunicación debe tener dos saltos. Primero, la estación móvil origen transfiere la trama al punto de acceso. Segundo, el punto de acceso transfiere la trama a la estación de destino.

#### 2.2.1.3 El medio inalámbrico y las Bandas ISM

La comunicación en una red 802.11, se caracteriza por utilizar las ondas de radio como soporte de dicha comunicación. A diferencia de lo que ocurre con las redes cableadas, no requieren de un cable o una fibra óptica que sirva de guía.

Los dispositivos inalámbricos utilizan una banda de frecuencia; cada banda tiene un ancho asociado que es simplemente la cantidad de espacio de frecuencia en la banda [GAST]. Existen bandas de frecuencia etiquetadas como ISM (*Industrial, Scientific and Medical*, Industrial, científico y médico). Dichas bandas permiten su utilización sin necesidad de licencia, siempre y cuando los dispositivos que las utilicen cumplan con determinadas restricciones de potencia. El estándar 802.11 funciona en las bandas ISM junto con muchos otros dispositivos. Por ejemplo los teléfonos inalámbricos comunes funcionan también en las bandas ISM, los dispositivos 802.11b y 802.11g funcionan dentro de la banda ISM de 2.4 GHz, mientras que los dispositivos 802.11a funcionan en la banda de los 5GHz.

#### 2.2.1.4 Ventajas y desventajas de las redes inalámbricas

Antes de comenzar con el detalle del funcionamiento del estándar 802.11, es relevante indicar algunas ventajas y desventajas de las redes inalámbricas. Las principales ventajas son:

- Movilidad y sencillez en la reubicación de terminales, así como la rapidez de instalación.
- La solución inalámbrica resuelve la instalación de una red en aquellos lugares donde el cableado resulta inviable, por ejemplo en edificios históricos o en grandes construcciones industriales, donde la realización de canaletas para cableado podría dificultar el paso de transportes, así como en situaciones que impliquen una gran movilidad de los terminales del usuario.

Y entre las desventajas encontramos:

- Las restricciones que puedan tener debido a que el medio utilizado es un canal gratuito y compartido.
- Debido a que tienen menor velocidad que las redes Ethernet, la calidad de servicio en este tipo de redes es un tema que aún se encuentra en desarrollo.
- Las redes inalámbricas tienen que autenticarse de forma robusta, para evitar su uso por parte de usuarios no autorizados y las conexiones autenticadas tienen que cifrarse sólidamente, para evitar la intercepción e inserción de tráfico por partes no autorizadas.

### 2.2.2 Funcionamiento de IEEE 802.11

IEEE 802.11 es un estándar inspirado en las redes cableadas, IEEE 802.3 (Ethernet), que sigue la misma técnica democrática para acceder al medio pero utilizando enlaces de radio. Por ello, similar a IEEE 802.3, IEEE 802.11 utiliza un esquema de Acceso múltiple con escucha de portadora (CSMA, *Carrier Sense Multiple Access*) para controlar el acceso al medio de transmisión; la diferencia radica en que las colisiones ocurridas en 802.3 consumen capacidad de transmisión porque son detectadas (CSMA/CD, *Collision Detection*), y en 802.11 en lugar de detectar dichas colisiones, son evitadas con CSMA/CA (*Collision Avoidance*), como consecuencia de esto IEEE 802.11 dificulta la implementación de calidad de servicio pues las prestaciones alcanzadas dependen de tres puntos:

- Número de usuarios presentes en la red
- Condiciones del medio
- Interferencia

Como resultado de la era multimedia, la gestión de la calidad de servicio adquirió mucha importancia, por ello la IEEE creó un grupo de trabajo para estudiar el soporte de calidad de servicio (QoS, *Quality of Service*) en 802.11. A finales del año 2005 y como resultado de este estudio, se obtuvo un nuevo estándar conocido como IEEE 802.11e, del cual se hablará más adelante.
Ya se mencionó que el protocolo 802.11 es un estándar de comunicaciones inalámbricas que define la capa física y la de enlace. En la figura 2.4 se muestra la arquitectura general del IEEE 802.11

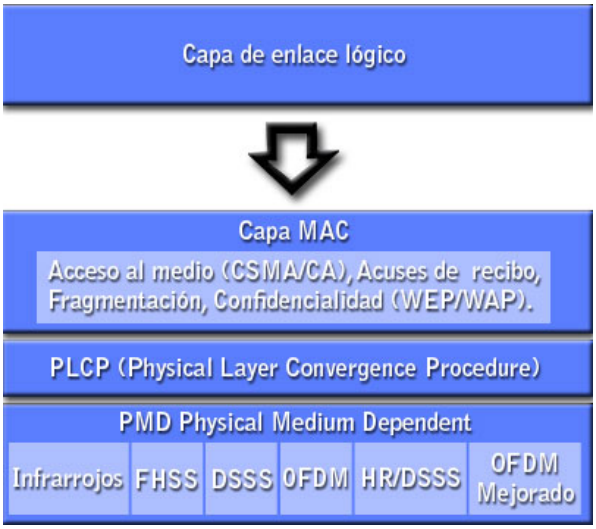



**Figura 2.4 Arquitectura general del IEEE 802.11**

De esta arquitectura sólo revisaremos el funcionamiento de la capa MAC, pues es en ella donde se ha realizado la modificación para lograr ofrecer la calidad de servicio que es el tema de interés de este trabajo.

#### 2.2.2.1 Capa MAC

En las redes cableadas es común suponer que al enviar una trama, ésta alcanzará su destino. Los enlaces de radio son diferentes pues están sujetos a ruidos e interferencias, por ello los dispositivos 802.11 tienen que suponer que va a existir una interferencia y deben tomarla en cuenta. Otro aspecto que también puede afectar a las transmisiones es el desvanecimiento de la señal por múltiples rutas, esto quiere decir que el nodo puede moverse hacia un punto en el cual ya no tenga la suficiente capacidad de recepción.

Por lo anterior, en IEEE 802.11 todas las tramas transmitidas deben tener un acuse de recibo positivo. Si alguna parte de la transferencia falla, la trama se considerará perdida. En la figura 2.5 se ilustra la operación que debe realizarse. Ésta, en su totalidad, debe ser completada con éxito de lo contrario se considerará fallida. El estándar 802.11 define dos funciones para realizar el acceso al medio inalámbrico (figura 2.6):

- Función de Coordinación Distribuida (*Distributed Coordination Function,* DCF*)*
- Función de Coordinación Centralizada (*Point Coordination Function*, PCF)

##### 2.2.2.1.1 Función de Coordinación Distribuida – DCF (*Distribution Coordination Function*)

DCF es una función que proporciona un acceso compartido al medio entre dispositivos con la misma capa física mediante el uso de un protocolo basado en Acceso Múltiple con Escucha de Portadora (CSMA, *Carrier Sense Multiple Access*) con evasión de colisiones (CA, *Collision Avoidance*). Todas las estaciones deben incluir obligatoriamente este mecanismo, a diferencia del mecanismo PCF que es opcional.

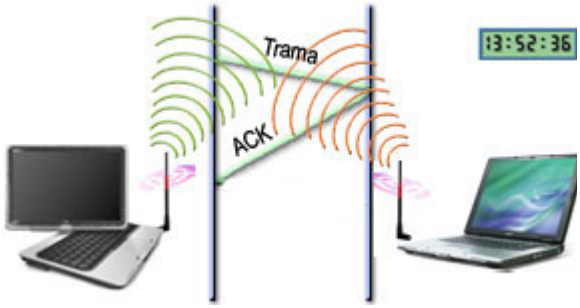



**Figura 2.5 Acuse de recibo positivo**


**Figura 2.6 Funciones de acceso al medio**

Las funciones de escucha de portadora se utilizan para determinar si el medio se encuentra disponible. La idea principal es que cualquier estación que quiera enviar tramas, antes de hacerlo, debe escuchar el canal y comprobar que el medio está libre por un periodo de tiempo (IFS, *Inter Frame Space*). La duración de este periodo varía, pero la utilizada justo antes de una transmisión en condiciones normales se llama DIFS (IFS de DCF, DIFS).

En 802.11 existen dos tipos de funciones que llevan a cabo dicha tarea (escuchar antes de transmitir):

- **Función de escucha de portadora física**: Es proporcionada a través de la capa física en cuestión, por lo que depende del medio. Esta función se ve afectada con el problema del nodo oculto [GAST], por lo que no puede proporcionar toda la información necesaria, razón por la cual se requiere también de la siguiente función.

- **Función de escucha de portadora virtual**: Esta es proporcionada por un vector de asignación de red (NAV, *Network Allocation Vector*). Las tramas 802.11 transportan un campo de duración que se puede utilizar para reservar el medio para cierto periodo de tiempo. NAV es un cronometrador que indica, en microsegundos, la cantidad de tiempo que se va a reservar el medio. Una vez que alguna estación actualice su NAV, debe contar hacia atrás desde el NAV hacia 0. Cuando el NAV es distinto de 0, la función de escucha de portadora virtual indica que el medio está ocupado; cuando el NAV llega a 0, la función de escucha de portadora virtual indica que el medio está libre. Con esto las estaciones pueden asegurar que no se interrumpen las operaciones atómicas. Este es el mecanismo que permite evitar las colisiones

Una vez que una estación consigue acceso al medio, ésta puede transmitir la trama de información (MSDU[1]). La estación que recibe la trama espera un periodo de tiempo llamado SIFS (*Short* IFS, IFS corto) para transmitir la confirmación (ACK[2]). La duración del periodo SIFS es más corta que DIFS, por ello la trama de confirmación tiene mayor prioridad para acceder al medio. Además de esta manera se asegura que ninguna otra estación podrá comenzar una transmisión antes que la confirmación. Si éste no es recibido justo después de un periodo SIFS, se intenta una retransmisión hasta que el número de retransmisión supera determinado umbral o el tiempo de vida de la MSDU expira. En este caso la trama de información MSDU sería descartada.

En la figura 2.7 se ilustra el siguiente ejemplo:

El remitente gana el acceso al medio y envía una trama RTS (*Request To Send*, Solicitud de envío) y espera un tiempo SIFS, el receptor le contesta con una trama CTS (*Clear To Send*, Libre para enviar), para indicarle que puede enviar sus datos. Mientras tanto, otra estación está escuchando el medio y le llega el NAV del RTS por lo que actualiza su NAV con el nuevo valor, después escucha el NAV del CTS y vuelve a actualizar su valor, esta estación no podrá transmitir pues su función de escucha de portadora virtual le

---

[1] **MSDU**: *MAC Service Data Unit Delivery,* es el servicio de la capa MAC encargado de la entrega de las unidades de datos.

[2] **ACK**: *ACKOWLEDGEMENT*, en español significa acuse de recibo.

indicará que el medio está ocupado. El remitente envía sus datos, espera un SIFS y el receptor le envía una trama ACK para indicar que sí recibió la trama de datos. En la figura se aprecia muy bien cómo el NAV protege la secuencias RTS/CTS y Datos/ACK.

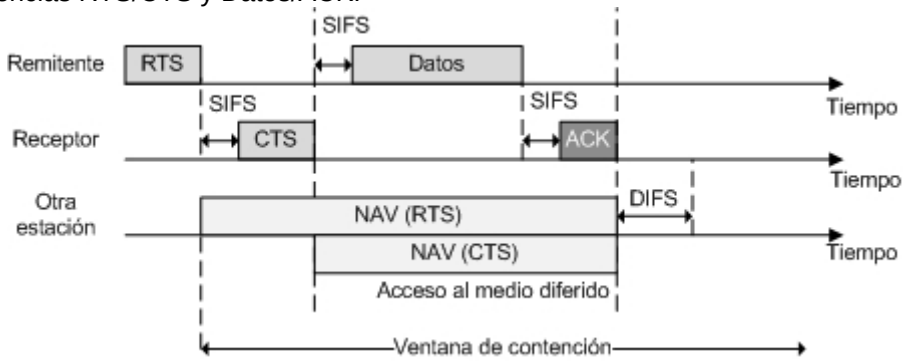



**Figura 2.7 Función de escucha de portadora virtual**

Antes de continuar con la explicación de DCF, es importante mencionar el algoritmo de demora exponencial conocido también como *backoff.* Este método es utilizado para resolver la contención entre diferentes estaciones que quieren acceder al medio, el método requiere que cada estación seleccione un número seudo aleatorio entre [CWmin,CWmax] donde:

$$CW_{min} \leq CW_{max} \leq 255$$

Inicialmente CWmin=0 y CWmax=31, si ocurre una colisión, la trama se intentará enviar otra vez, pero el rango de la ventana de contención cambiará de acuerdo a lo siguiente:

$$CW_{min,new} = 2 * CW_{min,old} + 1$$

Con base en la expresión anterior, los tamaños de la ventana de contención, son siempre uno menos que una potencia de 2 (31, 63, 127, 255, etc.) El tamaño de la ventana de contención está limitado por la capa física. Por ejemplo la capa física DSSS (*Direct-Sequence Spread-Spectrum*), limita la ventana de contención a 1023 ranuras o slots de transmisión.

En general todas las transmisiones que utilizan DCF siguen dos reglas básicas:

- Si el medio ha estado libre más tiempo que el DIFS, la transmisión se puede iniciar inmediatamente. Bajo esta regla pueden darse dos casos:
  - Si la trama anterior se recibió sin errores, el medio tiene que estar libre durante, al menos, el tiempo DIFS.
  - Pero si contenía errores, el medio tiene que estar libre durante el tiempo DIFS + el tiempo resultante de la ejecución del procedimiento de *backoff*, explicado anteriormente.
- Si el medio está ocupado, la estación tiene que esperar a que el canal quede libre. 802.11 se refiere a esa espera como acceso diferido (*Access deferral*); si se difiere el acceso, la estación espera a que el medio esté libre para el DIFS y se prepara a realizar el procedimiento *backoff* y de este modo seleccionar la cantidad de tiempo seudo aleatorio que deberá esperar.

El estándar 802.11 define un segundo mecanismo de acceso llamado PCF, pero este es opcional, por lo que los productos 802.11 no están obligados a implementarlo. El PCF está diseñado para ofrecer cierta calidad de servicio. Aunque es opcional, lo revisaremos más adelante pues representa un antecedente muy importante para el 802.11e. En la figura 2.8 podemos observar el modelo de funcionamiento del mecanismo DCF.

#### 2.2.2.1.2 Función de Coordinación Centralizada - PCF

Para soportar aplicaciones que requieran servicio en tiempo real, el estándar 802.11 incluye una segunda función de coordinación para proporcionar un método de acceso al medio inalámbrico distinto a DCF.

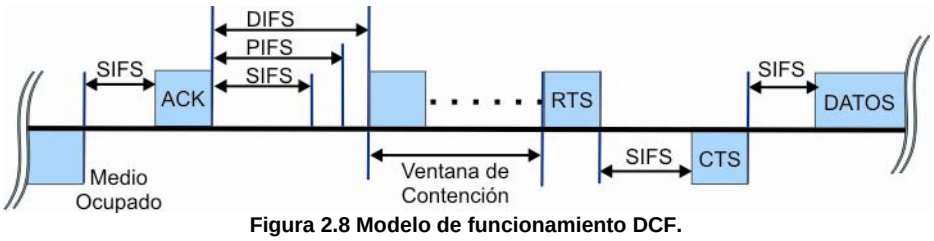



**Figura 2.8 Modelo de funcionamiento DCF.**

En esta función aparece un nuevo elemento llamado punto de coordinación (PC, *Point Coordinator*) el cual será el responsable de priorizar el acceso al medio de determinadas estaciones y normalmente está situado en el punto de acceso. Nos referiremos a él como punto de coordinación o PC. Antes de detallar cómo funciona PCF es importante mencionar lo siguiente:

El estándar 802.11 define dos tipos de periodos para el envío de mensajes, el periodo de contienda (CP, *Contention Period)* y el periodo libre de contienda (CFP*, Contention Free Period*). El primer periodo, como su nombre lo indica, es un tiempo en el cual las estaciones compiten entre sí para poder ganar el acceso al medio usando el mecanismo DCF. El segundo periodo está libre de tal contención, durante este periodo, el punto de coordinación será el que se encargue de controlar y otorgar el acceso al medio, usando un mecanismo basando en sondeo (*Polling*), PCF. Los periodos CP y CFP pueden alternarse como se muestra en la figura 2.9

Se puede configurar el tamaño de un período sin contención y durante éste, las estaciones asociadas pueden transmitir datos, sólo cuando se lo permita el punto de coordinación. De alguna manera, el acceso sin contención en PCF se parece a los protocolos de sistemas de red basados en *token,* en este caso la trama de sondeo que envía el coordinador de punto ocuparía el lugar del *token*.

**Funcionamiento de PCF**

Como ya se mencionó, si se requiere la entrega sin contención, se tiene que utilizar PCF. Éste es una parte opcional de la especificación 802.11 por lo que no es necesario que se implemente en todos los productos. Sin embargo, el IEEE diseñó PCF para que las estaciones que implementen sólo la función de coordinación distribuida, DCF, puedan interactuar con los puntos de coordinación.

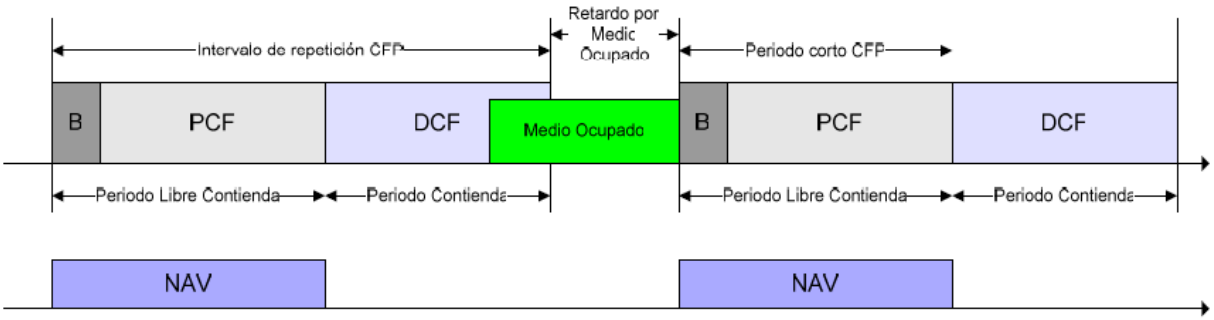



**Figura 2.9 Esquema de funcionamiento con periodos alternos.**

La figura 2.9 muestra una transferencia que utiliza PCF y DCF. Cuando se utiliza PCF, el tiempo del medio se divide entre los periodos CFP y CP. El acceso al medio en el primer caso está controlado por PCF mientras que el acceso al medio en el segundo caso está controlado por DCF. El periodo de contención tiene que ser lo bastante largo como para que se produzca la transferencia de, al menos, una trama de tamaño máximo y su acuse de recibo correspondiente. Periodos alternativos de servicios sin contención y servicios con contención se repiten a intervalos regulares, lo que se denominan intervalos de repetición sin contención.

Al inicio del periodo sin contención, el punto de acceso transmite una trama *Beacon*. Un componente de la trama *Beacon* es la duración máxima del periodo sin contención, *CFPMaxDuration*. Todas las estaciones que reciben la *Beacon,* establecen el NAV en la duración máxima, para bloquear el acceso al medio inalámbrico basado en DCF. Como seguridad adicional para evitar interferencia, todas las transmisiones sin contención se separan sólo por el espacio entre tramas corto (SIFS), y el espacio entre tramas PCF conocido como PIFS. Ambos son más cortos que el espacio entre tramas DCF (DIFS), por lo que ninguna estación basada en DCF puede obtener acceso al medio utilizando DCF.

Cuando el punto de acceso ha obtenido el control del medio inalámbrico, busca cualquier estación asociada en una lista de sondeos para las transmisiones de datos. Como ya se dijo, durante CFP las estaciones pueden transmitir sólo si el punto de acceso solicita la transmisión con una trama de sondeo.

El sistema de sondeo comienza cuando el punto de acceso envía una trama *CF-Poll,* la cual es una licencia o permiso para transmitir*,* a una de las posibles estaciones. Si el punto de acceso tiene alguna trama pendiente de envío, éste podría utilizar una trama de datos incorporando una trama *CF-Poll*, lo que resultaría en una trama *Datos+CF-Poll***.** La estación sondeada puede responder también con datos junto a una trama *CF-ACK* (*Datos+CF-Ack*), o simplemente con una trama *CF-ACK* si no desea enviar más información. Una vez que el intercambio de tramas con una estación termina, el punto de acceso envía el *CF-Poll* a otra estación que estuviese en la lista de estaciones sondeables. Cuando el punto de acceso ha terminado con todas las estaciones de la lista, o una vez que la duración del CFP ha expirado, transmite por difusión una trama *CF-End* anunciando el final del ciclo CFP. Sólo se pueden transmitir múltiples tramas si el punto de acceso envía múltiples peticiones de sondeo. En la figura 2.10 se ilustra el funcionamiento de PCF.

La duración mínima del periodo de contención es el tiempo requerido para transmitir y acusar recibo de una trama de tamaño máximo. Sin embargo, el servicio basado en contención puede sobrescribir el final del periodo de contención en el principio del periodo sin contención. Esto es, cuando el servicio basado en contención (CP) se ejecuta pasado el inicio esperado del periodo libre de contención (CFP), éste se reduce, como se muestra en la figura 2.11.

Cuando el periodo sin contención (CFP) se reduce, se permite completar el intercambio de tramas existente antes de que se transmita la *Beacon* que anuncia el inicio de la operación sin contención. El periodo sin contención se reduce en la cantidad de la demora. El servicio sin contención no finaliza más tarde que la duración máxima del punto de inicio esperado, que se conoce como Tiempo de Transmisión de Beacon de Destino (TBTT, *Target Beacon Transmission Time*). Como se mencionó anteriormente, PCF maneja soporte para calidad de servicio, sin embargo tiene las siguientes limitaciones.

#### 2.2.2.1.3 Limitaciones de PCF para el soporte de Calidad de Servicio

Debido a los problemas en el mecanismo de acceso PCF el grupo 802.11 ha trabajado en desarrollar mejoras para el soporte de calidad de servicio. Algunos de estos problemas son los retardos impredecibles de las tramas de *beacon* y la duración desconocida de los periodos de transmisión de las estaciones en el periodo de contienda (CP).

El responsable del envío de las tramas *beacon* a intervalos regulares, TBTT, es el punto de coordinación. Sin embargo, esta trama sólo puede transmitirse cuando el medio ha sido detectado como vacío por un periodo PIFS. Con base en el estándar 802.11, las estaciones pueden comenzar sus transmisiones aun cuando la trama MSDU (MAC *Service Data Unit*) enviada no está acabada antes de la llegada del TBTT. En función de si el medio está vacío u ocupado durante TBTT, se podría producir un retardo de la trama *beacon*. El retraso provocado de esta forma sobre TBTT fijará el retardo de la transmisión de MSDUs que tienen que ser enviadas en el ciclo CFP. Este hecho podría afectar severamente a la calidad de servicio ya que introduce un retardo impredecible en cada ciclo CFP. En el peor de los casos (mayor MSDU, fragmentación, RTS/CTS activadas) se podrían llegar a alcanzar retardos de algunos milisegundos.

Otro problema con el mecanismo PCF es que el tiempo de transmisión de las estaciones en el periodo CP es desconocido. Una estación que ha sido sondeada por el punto de coordinación tiene la posibilidad de enviar una MSDU que podría ser fragmentada y de una longitud arbitraria, hasta un máximo de 2304 bytes (2312 en caso de usar algún cifrado). Además, al existir diferentes esquemas de modulación y codificación, la duración del envío de las MSDU después del sondeo no está bajo control del punto de coordinación, lo cual reduce la calidad de servicio proporcionada a otras estaciones durante el resto del periodo de CFP.

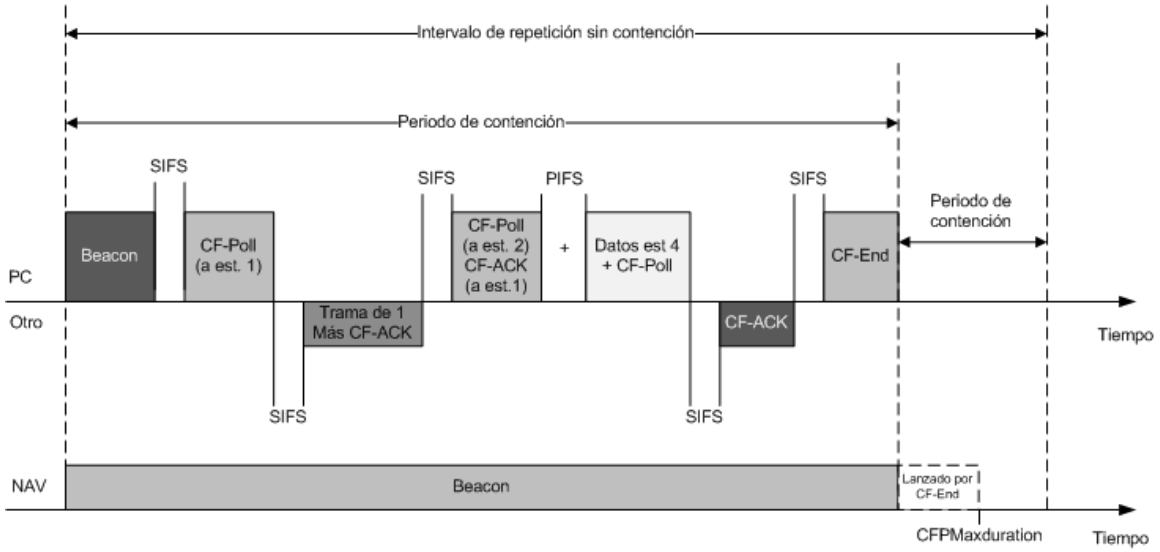



**Figura 2.10 Funcionamiento de PCF**

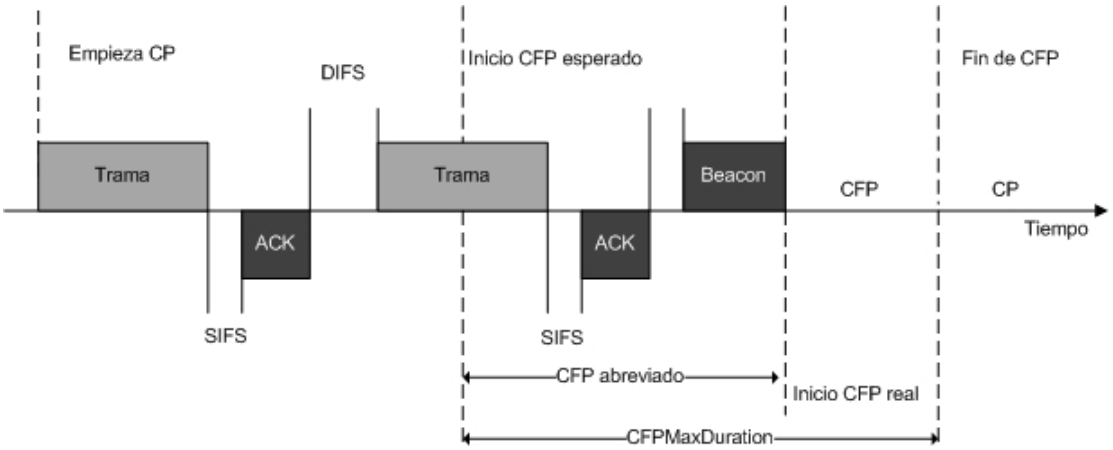



**Figura 2.11 Utilización de Datos+CF+Ack y Datos+CF+Poll**

Adicional a esto, en el estándar no se ha definido ninguna interfaz de gestión para controlar las operaciones PCF. Esto ocasiona que sea imposible configurar alguna política PCF según los requisitos de protocolos de capas superiores tales como *Diffserv*[3] o *Intserv*[4], mismos que serán explicados en el capítulo 3. Entonces no existe ningún mecanismo para comunicar los requisitos de calidad de servicio de las estaciones al punto de acceso, algo que es fundamental para optimizar el rendimiento del algoritmo PCF en el punto de coordinación.

Estos problemas no resueltos en PCF, son los que guían las actividades del grupo IEEE 802.11 para mejorar el protocolo.

[3] **Diffserv**: Se refiere a los servicios diferenciados, los cuales proporcionan un método que intenta garantizar la calidad de servicio en redes de gran tamaño. La idea básica de este método es clasificar el tráfico en varios flujos.

[4] **Intserv**: Se refiere a los servicios en donde hay una reservación anticipada de recursos.

# Capítulo 3 IEEE 802.11e y Calidad de Servicio

## 3.1 Calidad de Servicio

El concepto de calidad de servicio (QoS, *Quality of Service*) se refiere a proporcionar un nivel de servicio "adecuado" a cada tipo de tráfico, es decir, una tecnología de comunicación ofrece calidad de servicio si ésta puede garantizar la transmisión de cierta cantidad de datos de un determinado tipo de tráfico, si esto ocurre entonces se está ofreciendo un buen servicio. Hoy en día, sobre todo para determinado tipo de tráfico se requiere que la información llegue:

- A una cierta tasa de bits (*Throughput*)
- En un determinado tiempo (*Delay* o Retardo)
- Con una variación determinada (*Jitter*)
- Con cierta pérdida de paquetes
- Con una cantidad máxima de errores

Con base en lo anterior y en términos cuantitativos, el perfil de un tipo de tráfico se determina básicamente por cinco valores: *Throughput*, *Delay*, *Jitter*, pérdida de paquetes y errores recibidos.

Adicionalmente a estos valores, existen también dos factores importantes para garantizar la calidad de servicio a los tipos de tráfico que así la requieren, como la voz y el video. Estos factores son la cantidad de ancho de banda y la gestión del uso que se haga de él. Como ya sabemos, el medio es un recurso finito que ha de ser compartido por todas las aplicaciones que transmitan datos en la red, el uso que se hace de ésta o la prioridad que se asigne a cada uno de los paquetes transmitidos, repercutirá de forma determinante en aquellas aplicaciones más sensibles al tiempo de transmisión y recepción. Surge entonces la necesidad de priorizar aquellos paquetes que deban ser transmitidos en "tiempo real". En este punto, la calidad de servicio (QoS) juega un papel fundamental.

Existen diversas técnicas para ofrecer calidad de servicio, como son:

- **Clasificación de tráfico**: Esta técnica controla el tráfico en "enlaces individuales". Por ejemplo, IEEE 802.1p/Q emplea dos bytes adicionales en la trama para indicar un determinado nivel de prioridad.

- **Fragmentación de tráfico:** En esta técnica, los paquetes grandes se dividen en otros de menor tamaño, para evitar que los paquetes de voz se vean obligados a esperar demasiado tiempo en las colas de los equipos de ruteo, antes de ser transmitidos.

- **Gestión de ancho de banda:** Su objetivo es ofrecer a las aplicaciones de red calidad de servicio de extremo a extremo. En este grupo se tiene a *IntServ* y *DiffServ*.

- **Control de la congestión:** Indica reglas que establecen cómo deben gestionarse las colas de tráfico en los nodos de la red.

### 3.1.1 Tipos de flujos de información

Así como es importante conocer las diversas técnicas que nos permiten lograr cierta calidad de servicio en la información transmitida, también es relevante mencionar que existen distintos tipos de flujos de información relacionados con el tema de calidad de servicio.

- ***Conversational (Conversacional)***: Básicamente se caracteriza por mantener un bajo retardo en la transmisión de información, de forma que se identifica muy claramente con los tradicionales servicios de telefonía. Dentro de esta clasificación entrarían igualmente aplicaciones de audio-conferencia (VoIP) o videoconferencia sobre IP.

- ***Streaming ()***: Aquí el retardo no es un factor tan determinante, pero sí lo es el lograr una relación de flujo constante entre origen y destino. El mejor ejemplo es el video bajo demanda.

- ***Interactive***: En este caso se presta especial importancia al retardo de ida y vuelta y a la tasa de error. Se trata de aplicaciones de acceso remoto a sistemas interactivos, donde resulta especialmente importante lograr una transmisión sin errores.

- ***Background***: Se caracteriza por una tasa de error nula pero sin restricciones respecto del retardo extremo a extremo. En este tipo de tráfico podríamos localizar aplicaciones como el correo electrónico, la transmisión de archivos, servicios no interactivos, etc.

Como ya se ha mencionado, la calidad de servicio hace referencia a la capacidad de la red de proporcionar el nivel de servicio adecuado a cada tipo de tráfico. Para ordenar el tema de la calidad de servicio en las redes inalámbricas, la IEEE estableció el estándar IEEE 802.11e para garantizar la calidad de servicio. Y aquí surge una cuestión interesante:

Se mencionó que el problema de garantizar la calidad de la transmisión de voz o de video, depende de dos factores, el ancho de banda y la gestión de su uso; sin embargo las redes inalámbricas, por su naturaleza, tienen una limitante importante en ese ancho de banda, por lo que la complejidad del reto para lograr calidad de servicio se incrementa ya que se tiene un ancho de banda limitado y se debe gestionar de una mejor forma el uso que se hace de éste.

Para lograr calidad de servicio en las transmisiones de datos, el estándar IEEE 802.11e básicamente lleva a cabo una clasificación de tráfico, para lo cual utiliza cuatro categorías que se revisarán más adelante.

## 3.2 Modelos de servicios

Existen algunos modelos de servicio cuyo objetivo es proponer una solución para el soporte de calidad de servicio durante toda la transmisión del flujo de la información. A continuación se hablará acerca de dos de ellos: *Intserv* y *Diffserv.*

### 3.2.1 *Intserv:* Servicios Integrados

La idea básica de este modelo es la pre-reservación de los recursos de los diferentes equipos que participan en la transmisión de la información, con el objetivo de ofrecer soporte a las aplicaciones en tiempo real. Para lograr esto se utilizan dos elementos:

- Una arquitectura que permita la pre-reservación de recursos de los equipos.
- Un protocolo que permita a las aplicaciones transmitir sus requisitos a estos equipos. Por ejemplo el protocolo RSVP (*Resource Reservation Protocol*).

Así por ejemplo, cuando una aplicación requiere iniciar una comunicación debe realizar una petición de recursos, dicha petición atravesará todos los nodos que formen el trayecto para el flujo de información, y en función de los recursos disponibles, dicha petición o reserva será aceptada o rechazada. En la figura 3.1 se ilustra este procedimiento. Aquí se pueden observar dos funciones básicas: gestión de recursos y control de admisión.

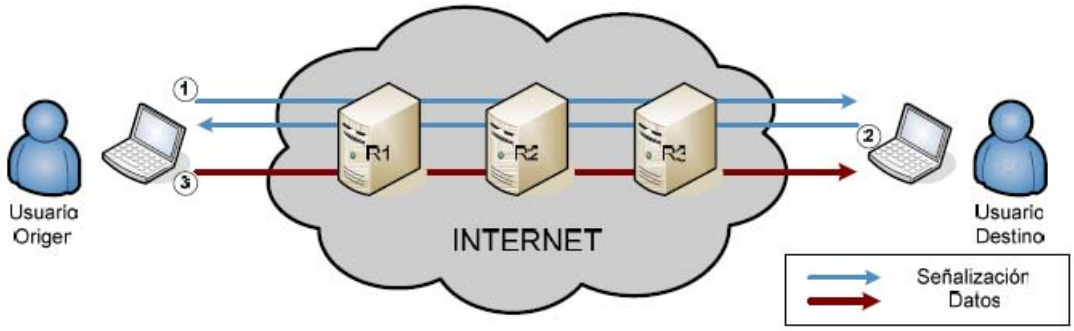



**Figura 3.1 Ejemplo de pre-reservación de recursos: *Intserv***

En el modelo *Intserv* el tráfico es clasificado en:

- **Elástico**: Es un tipo de tráfico en el cual el retardo que sufren las diferentes tramas entre fuente y destino no afecta de forma substancial al servicio, por ejemplo el tráfico web, correo electrónico, servicio de ftp.
- **Inelástico**: Es un tipo de tráfico muy sensible al retardo que requiere de un ancho de banda mínimo para que el servicio no se vea afectado, por ejemplo audio y videoconferencias, video bajo demanda.

Y los servicios ofrecidos son:

- **Servicio garantizado**: En éste se permite reservar un ancho de banda mínimo extremo a extremo y también se puede limitar el retado máximo de las tramas. La implementación de este servicio se realiza a través de un protocolo de reserva como RSVP.
- **Servicio *Best-Effort***: Es utilizado cuando la reserva de recursos no resulte exitosa debido a la falta de recursos disponibles en la red. Por ello no es necesario hacer una reserva de recursos. Éste es adecuado para el tipo de tráfico elástico.
- **Servicio de carga controlada**: Este servicio ofrece una calidad comparada a la de una red que no está congestionada, es decir en general un buen tiempo de respuesta pero sin garantías estrictas por lo que eventualmente puede tener grandes retardos.

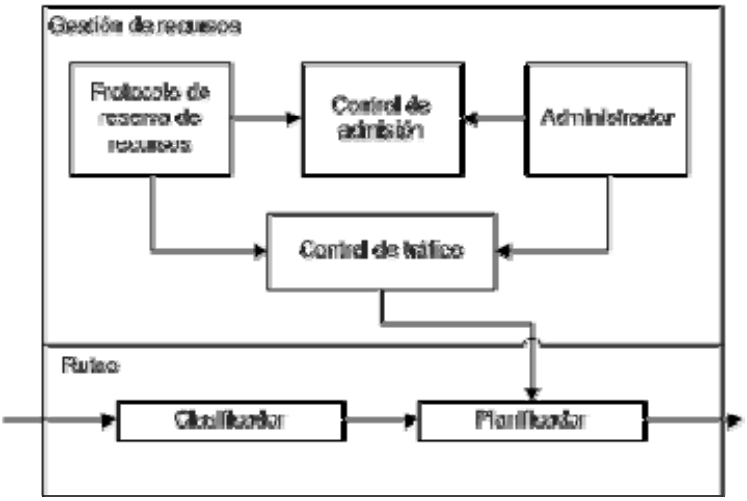



**Figura 3.2 Funciones de *Intserv*.**

En la figura 3.2 se muestra de forma resumida las funciones del modelo *Intserv*.

### 3.2.1.1 Protocolo de Reserva. RSVP

Como ya se mencionó, RSPV es un protocolo que permite a las aplicaciones de una red de paquetes obtener una calidad de servicio determinada para los flujos de datos que la atraviesan. Se utiliza para indicar a los ruteadores de la red las necesidades de calidad de servicio de los usuarios y de sus aplicaciones. Los mensajes RSVP se envían en paralelo con los paquetes IP con número de protocolo 46 y su función es doble: por una parte, indica los recursos que se desea reservar y, por otra, describir el perfil de los paquetes a los que se quiere aplicar la reserva.
El protocolo RSVP lleva a cabo una reserva de cierta cantidad de ancho de banda, para ello lleva el registro de una lista dinámica de equipos intermedios, (la mayoría de las veces estos equipos son de ruteo), de extremo a extremo de la comunicación, de tal forma que si no es posible soportar determinada calidad de servicio mínima requerida en cada uno de los recursos de la red ubicados entre origen y destino, a través de los cuales pasará el tráfico de datos, el solicitante recibirá un mensaje de error, indicando que la comunicación no puede llevarse a cabo.

En general se definen dos tipos de servicio:

- **Garantizado**: el retardo máximo ofrecido por la red y el ancho de banda solicitado es fijo. Apto para aplicaciones que descarten un paquete si éste no ha llegado antes del instante de utilización.
- **Predictivo**: el retardo máximo es aproximado y la pérdida de paquetes debe ser muy baja o nula. Es adecuado para aplicaciones que se adapten a variaciones en el retardo a costa de la calidad.

El tipo de servicio más adecuado para aplicaciones de tiempo real es el servicio garantizado, aunque es mucho más complejo de implementar.

El protocolo define dos sentidos para la transferencia de mensajes de señalización, uno del origen al destino (sentido *downstream*) y el otro del destino a la fuente (sentido *upstream*). La figura 3.3 muestra el funcionamiento de este protocolo. Para establecer una comunicación entre dos extremos debe haber recursos en ambos sentidos. Esta reserva de recursos se efectúa a través de dos mensajes básicos de RSVP. Antes de enviar cualquier otro tipo de información, el origen indica un perfil del tráfico que desea enviar en términos de ancho de banda, retardo y *jitter* junto con sus márgenes y variaciones dentro del campo *TSpec* del mensaje PATH. El objetivo de este mensaje es solicitar la reserva de recursos en el sentido descendente en cada *router* de la red que atraviesa.

Esta reserva de recursos debe confirmarse en cada ruteador mediante el envío, en sentido ascendente, de un mensaje RESV que, además del TSpec, contiene una indicación del tipo de servicio requerido (campo *RSpec, Request Specification*) y una notificación de si la reserva se ha llevado a cabo con éxito o no (campo *FilterSpec*, *Filter Specification*). Estos indicadores forman un descriptor de flujo que caracteriza a la reserva de recursos en cuestión. Cuando el último *router* (aquel que se encuentra más cerca del origen) recibe el mensaje RESV, responde con un mensaje de confirmación al destino y la reserva se da por concluida. Sin embargo, esta reserva de recursos será válida, sólo si la congestión y el retardo que introduzcan los *routers* no RSVP no es significativa.

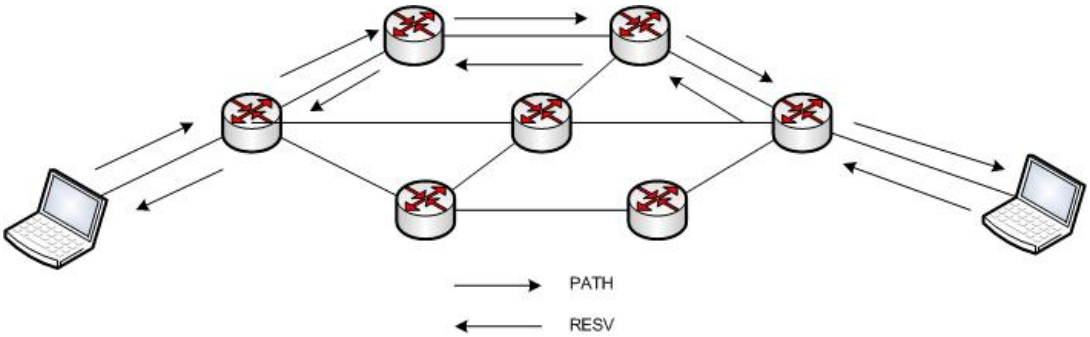



**Figura 3.3 Calidad de servicio extremo a extremo en RSVP**

Las solicitudes de reserva establecidas mediante mensaje PATH tienen un tiempo de vida (*timeout*), si este tiempo vence se llevará a cabo una liberación de recursos con los mensajes PATHTEAR y REVSTEAR, respectivamente.

Por otro lado RSVP también ofrece información sobre errores. En el sentido ascendente, el mensaje PATHERR notifica que se ha producido un error de procesamiento de un mensaje PATH, aunque no modifica el estado de los nodos que atraviesan. En cuanto al sentido descendente, el mensaje RESVERR indica que ha habido un error en un mensaje RESV o que se ha interrumpido una reserva.

### 3.2.2 *Diffserv*: Servicios Diferenciados

Este modelo propone una solución para el soporte de calidad de servicio basada en la priorización de tipo de tráfico. En este modelo también se realiza una especie de "reserva de recursos" en los nodos intermedios, en esta reserva de recursos las aplicaciones no hacen ninguna petición de recursos sino que simplemente marcan el tipo de tráfico generado para que reciba un tratamiento específico en función de la clase a la que pertenezca.

En este modelo la red trata de cumplir los requerimientos de calidad de servicio prometidos a una aplicación con base en la información contenida en cada paquete, en lugar de emplear señalización explícita con el *router* antes de proceder al envío de información.

Para diferenciar los tipos de servicio entre sí se utiliza una marca específica en cada paquete. Esta marca depende de la versión del protocolo IP que se esté utilizando.

En IPv4, por ejemplo, se establece el campo ToS (*Type of service,* Tipo de servicio) de la cabecera. Del total de los 8 bits del campo ToS, los tres primeros bits indican la prioridad relativa del paquete según seis clases de servicio distintas. Los otros cinco bits representan las características del servicio en términos de retardo mínimo, *throughput* máximo, fiabilidad máxima y coste mínimo excepto el último bit, que siempre es nulo (MBZ, *Must Be Zero,* Debe ser cero). El funcionamiento básico del modelo de calidad de servicio que se utiliza en IEEE 802.11e toma ideas de los modelos mencionados.

## 3.3 IEEE 802.11e: Arquitectura y funcionamiento

Debido a la problemática que presenta el estándar IEEE 802.11 para soportar calidad de servicio, el IEEE creó un grupo cuyo principal objetivo sería realizar modificaciones sobre el 802.11 para lograr cierto nivel de calidad de servicio. Fue de esta forma como el grupo de trabajo IEEE 802.11e definió un conjunto de novedades publicadas en 1999.

Para entender estas novedades es importante describir la nomenclatura que se utilizará:

- **QSTA (*QoS Enhanced Station*)**: Se refiere a la estación que soporta calidad de servicio. También se les conoce como estaciones 802.11e
- **STA (*Station*)**: Son las estaciones que no soportan calidad de servicio.
- **QAP (QoS *Access Point*)**: Punto de acceso que soporta calidad de servicio. Normalmente en el QAP reside el coordinador central que en 802.11e se conoce como *Hybrid Coordinator* (HC), este concepto es similar a PCF.
- **AP (*Access Point*)**: Punto de acceso que no soporta calidad de servicio.
- **QBSS (QoS BSS)**: Conjunto de servicios que soportan calidad de servicio. Un QBSS incluye un HC en el punto de acceso.

Dentro de IEEE 802.11e se pueden distinguir dos grupos funcionales:

- Funciones de acceso al canal
- Gestión de especificación de tráfico (TSPEC).

### 3.3.1 Capa MAC IEEE 802.11e

La extensión 802.11e define una nueva función de coordinación llamada función de coordinación híbrida (HCF, *Hybrid Coordination Function*), la cual se emplea para el conjunto de servicios básicos con soporte de QoS (QBSS). La función HCF define dos modos de operación:

- **EDCA (*Enhanced Distributed Channel Access*)***:* Acceso a canal distribuido mejorado, el cual consiste en una función de acceso basada en contienda y que funciona de forma concurrente junto al segundo modo de operación llamado HCCA.
- **HCCA (HCF *Controlled Channel Access*)**: Acceso a canal controlado HCF, el cual se basa en un mecanismo de sondeo controlado por el coordinador híbrido (HC, *Hybrid Coordinator*). Este coordinador se encuentra situado en el punto de acceso (QAP). Este modo de acceso incluye una técnica de sondeo (*Polling*).

Ambas funciones (EDCA y HCCA) de acceso mejoran o extienden la funcionalidad de los métodos de acceso originales (DCF y PCF). La primera función de acceso, EDCA, fue diseñada para soportar la priorización de tráficos, tal como hace *Diffserv*, mientras que HCCA soporta tráfico parametrizado, de la misma forma que *Intserv*.

HCCA y EDCA operan concurrentemente, durante los periodos de contención (*Contention Period*, CP) la EDCA gestiona el acceso al canal, mientras que la HCCA es la más utilizada durante los periodos libres de contención (*Contention Free Period*, CFP). Por ello, en 802.11e puede haber dos fases de operación en cierto periodo de tiempo y se conoce como *SuperFrame*. Un SuperFrame consiste en un periodo de contención (CP) y un periodo libre de contención (CFP). EDCA es usado únicamente en CP mientras que HCCA es usado en ambos periodos, CP y CFP. La función HCF combina métodos de PCF y DCF, razón por la cual es llamada híbrido.

El concepto básico de estas funciones de acceso al canal es la oportunidad de transmisión (TXOP, *Transmisión Opportunity*). Un TXOP es un intervalo de tiempo limitado durante el cual una QSTA tiene derecho a transmitir una serie de tramas. El periodo TXOP se define a través de un tiempo de inicio y una

duración máxima. Si el periodo TXOP se obtiene usando el acceso a canal basado en contienda, entonces recibirá el nombre de EDCA-TXOP. Si por el contrario, se obtiene a través de HCCA se conocerá como HCCA-TXOP. La diferencia entre ambas es que la duración del periodo EDCA-TXOP se controla a través del QAP y se transmite al resto de estaciones QSTA en las tramas de *beacon* junto con otros parámetros relacionados con EDCA, mientras que la duración del periodo HCCA-TXOP se transmite a las estaciones QSTA directamente por el HC como parte de la trama *QoS CF-Poll*, la cual garantiza el periodo HCCA-TXOP (ver figura 3.4), más adelante se explicará la trama *QoS CF-Poll*.

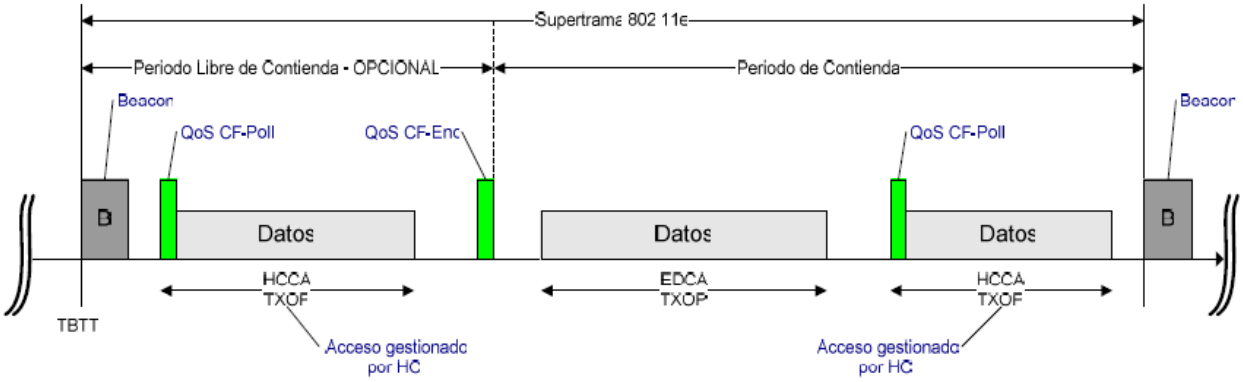



**Figura 3.4 Esquema de funcionamiento de HCF**

Como se comentó, en el estándar original IEEE 802.11, el envío de las tramas de confirmación es obligatorio para cada trama recibida correctamente. En 802.11e estas tramas de confirmación ahora son opcionales. Así cuando se usa una política basada en no utilizar confirmaciones, la capa MAC no deberá enviar mensajes ACK por cada trama recibida correctamente. Es cierto que la fiabilidad de este tráfico se verá reducida, pero mejorará el rendimiento general de la capa MAC para tráfico sensible al retardo, tal como sucede con VoIP, donde la información tiene un valor durante un periodo de tiempo muy corto.

La opción de trabajar sin confirmaciones también introduce severos requisitos de tiempo real, ya que si no es necesario esperar la trama de confirmación, entonces la siguiente trama a transmitir debe estar preparada en un tiempo SIFS (que es el tiempo más corto) desde el final de la anterior transmisión.
A continuación analizaremos con mayor detalle la función de acceso distribuido EDCA.

### 3.3.2 Acceso al canal mejorado, EDCA

La DCF descrita en el estándar 802.11 no es capaz de garantizar ningún tipo de calidad de servicio. La EDCA incluye un mecanismo de priorización de tráfico que mejora el original DCF para proporcionar soporte de calidad de servicio. Esta priorización se consigue introduciendo cuatro categorías de acceso (AC, *Access Category*) que permiten el envío de tráfico asociado a prioridades de usuario, tal como lo define el estándar IEEE 802.1D En la tabla 3.1 se resumen las prioridades relativas y la tabla de mapeo entre 802.1D y las categorías de acceso 802.11e.

**Tabla 3.1 Mapeo de prioridad de usuario a Categoría de Acceso**

| Prioridad | Prioridad 802.1D | Descripción 802.1D | Categoría Acceso 802.11e | Descripción 802.11e |
|---|---|---|---|---|
| Menor | 1 | Background | AC_BK | Background |
| … | 2 | Background | AC_BK | Background |
| … | 0 | Best Effort | AC_BE | Best Effort |
| … | 3 | Excelent Effort | AC_BE | Best Effort |
| … | 4 | Carga controlada | AC_VI | Video |
| … | 5 | Video | AC_VI | Video |
| … | 6 | Voz , Video | AC_VO | Voz |
| Mayor | 7 | Señalización Red | AC_VO | Voz |

En la figura 3.5 se muestra el mapeo entre 802.1D, 802.11e y 802.11.

Cada categoría de acceso dispone de su propia cola de transmisión caracterizada por determinados parámetros. La priorización entre las diferentes categorías se consigue configurando adecuadamente los parámetros de cada cola de acceso. Podemos ver un esquema de funcionamiento del sistema de categorías de acceso en la figura 3.6. Los parámetros de mayor interés son los siguientes:

- **Número de Espacio Arbitrario entre Tramas (AIFSN, *Arbitrary Inter-Frame Space Number*)**: se corresponde con el intervalo mínimo desde que el medio físico se detecta como vacío hasta que se comienza la transmisión.
- **Ventana de Contienda (CW, *Contention Window*)**: un número aleatorio se escoge en este rango para lanzar el mecanismo de espera (*backoff*). Se manejan dos valores: CWmin y CWmax.
- **Límite de Oportunidad de Transmisión (*TXOP limit*)**: es la duración máxima durante la cual una QSTA puede transmitir tras haber obtenido el TXOP.
- **Factor de persistencia (PF, *Persistence Factor*)**: es la cantidad que se emplea para determinar el incremento de la ventana de contienda de una categoría de acceso, una vez que se ha detectado una colisión. El tráfico de mayor prioridad tendrá un valor menor que el tráfico de prioridad más baja.

En la tabla 3.2 se muestra el conjunto de parámetros básicos para cada categoría y los valores recomendados por default.

**Tabla 3.2 Parámetros y valores por default**

| | AC_VO | AC_VI | AC_BE | AC_BK |
|---|---|---|---|---|
| AIFSN | 2 | 2 | 3 | 7 |
| CWmin | 3 | 7 | 15 | 15 |
| CWmax | 7 | 15 | 1023 | 1023 |

Como ya se mencionó, la calidad de servicio soportada en EDCA es provista por la introducción de categorías de acceso (*ACs*) y la múltiple independencia de las estaciones 802.11e. Los *MSDUs* son entregados por las estaciones 802.11e, cada una de estas estaciones son priorizadas usando el conjunto de parámetros específicos a cada categoría. Como se vio en la tabla 3.2 hay cuatro categorías, AC_VO (voz), AC_VI (video), AC_BE (*best-effort*), AC_BK (*background*). El conjunto de parámetros EDCA define las prioridades para el acceso al medio.

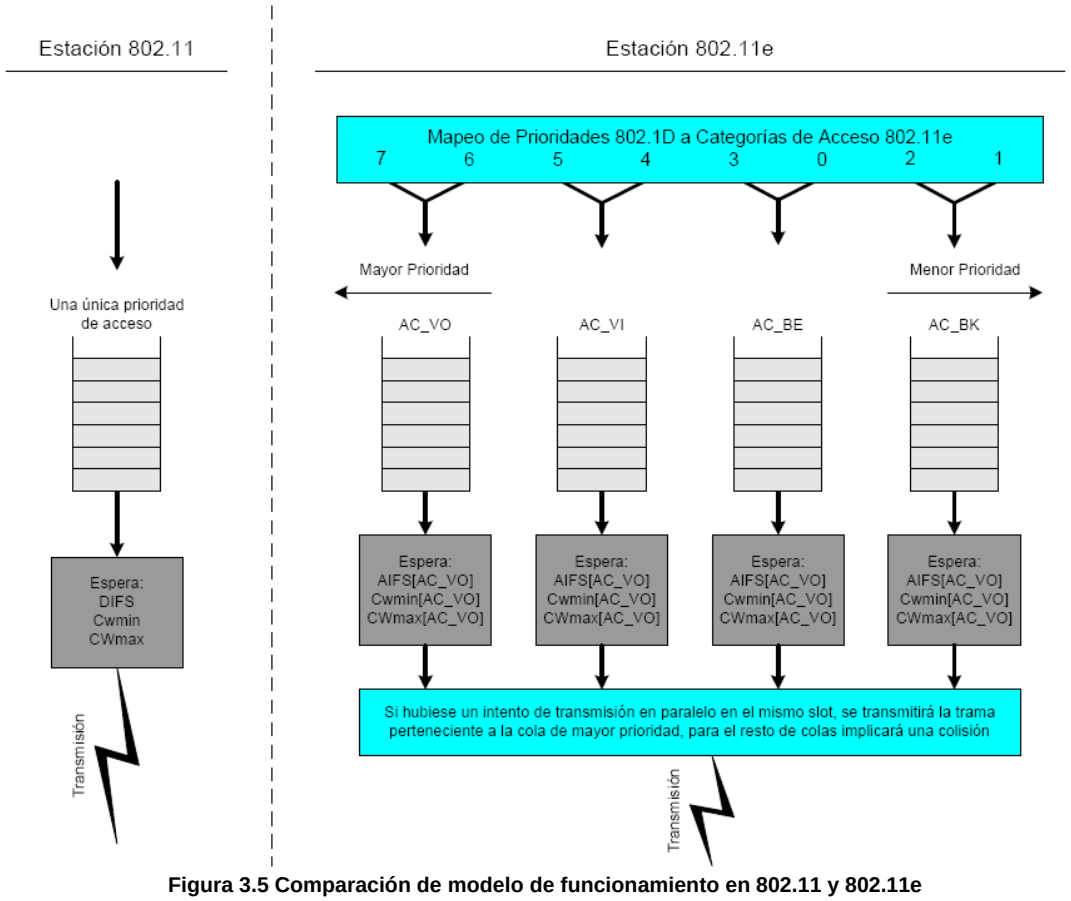



**Figura 3.5 Comparación de modelo de funcionamiento en 802.11 y 802.11e**

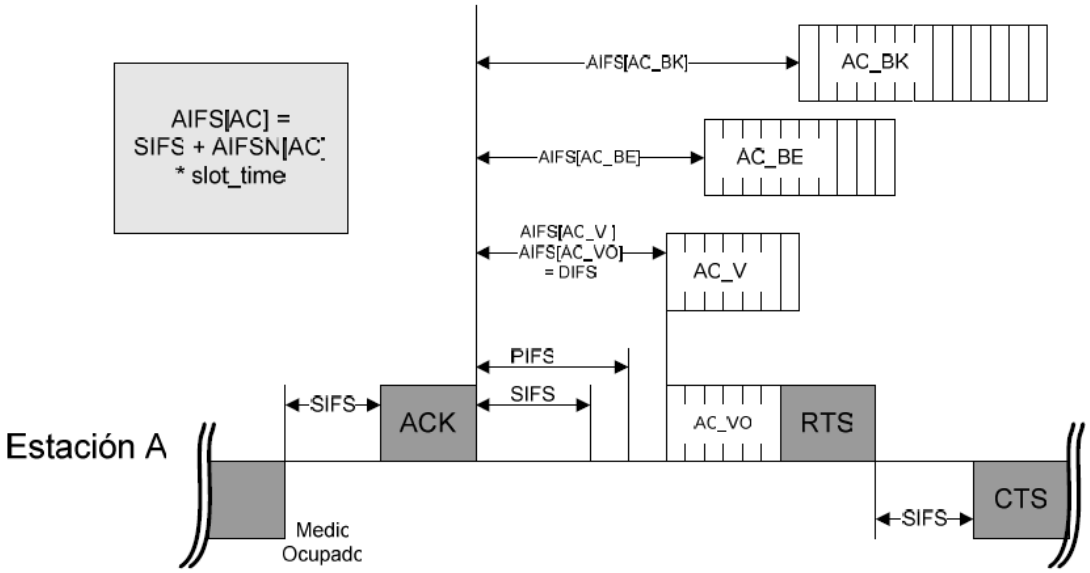



**Figura 3.6 Modelo de funcionamiento de capa MAC 802.11e**

Cuando los datos llegan al punto de acceso, la capa MAC de 802.11e se encarga de clasificar adecuadamente dichos datos y envía la MSDU a la cola correspondiente. Entonces los bloques de información (MSDU) de las diferentes colas (AC), compiten internamente por el EDCA-TXOP. El algoritmo de contienda interno calcula la espera (*backoff*), independientemente para cada cola (AC), según los parámetros descritos: AIFSN, CW y un número aleatorio. El mecanismo de espera es similar al de DCF y la cola con el menor *backoff* ganará la competición interna. La expresión para el cálculo de este tiempo está definida por:

**AIFS[AC] = SIFS + AIFSN[AC] * aSlotTime, AIFSN[AC] ≥ 2**

Donde:

AIFSN[AC] -> Lo podemos observar en la tabla 3.2
SIFS -> 16 µs.
aSlotTime -> 9 µs. (Duración de un *slot*).

AIFSN[AC] debe ser seleccionada por el HC, el parámetro *aSlotTime* define la duración de un slot. El más pequeño AIFSN[AC] representa la mayor prioridad del acceso al medio.

La cola (AC) vencedora, competirá externamente por el acceso al medio inalámbrico. El algoritmo de contienda externo no se ha modificado significativamente comparado con DCF, excepto que en DCF el *backoff* y los tiempos de espera son fijos para un medio físico concreto, mientras que en 802.11e estos son variables y se configuran adecuadamente según la cola (AC) correspondiente.

A través de un ajuste adecuado de los parámetros de las colas (AC), el rendimiento del tráfico de diferentes colas puede ser ajustado y se puede lograr la priorización de tráfico. Esto requiere un punto de coordinación central (QAP), para mantener un conjunto común de parámetros en las colas y garantizar así un acceso justo entre las diferentes estaciones que componen la red (QBSS). De igual forma, para lograr ajustar la asimetría existente entre el tráfico de subida (QSTA a QAP) y de bajada (QAP a QSTA), un conjunto separado de parámetros EDCA se define exclusivamente para el QAP.

Cada QSTA implementa hasta 4 colas de transmisión con parámetros de calidad de servicio independientes que determinarán su prioridad. Se ejecuta un algoritmo de *backoff* parametrizado para adaptarse a cada AC. Durante el periodo de contienda, cada TC compite por una oportunidad de transmisión y tras comprobar que el canal ha permanecido inactivo durante un tiempo denominado AIFS, da comienzo un algoritmo de *backoff*. El AIFS es, al menos, de igual duración que el DIFS y cada AC tiene asignado un valor que se asemeja de manera independiente en cada AC. Tras esperar un AIFS, cada *backoff* empieza una cuenta que acaba al alcanzar el número aleatorio de slots distribuido uniformemente en el intervalo [0,CWi].

En la figura 3.7 se muestran los parámetros de la EDCA.

Al igual que ocurre en DCF, cuando el medio está ocupado antes de que el contador llegue a cero, el *backoff* congela la cuenta y espera hasta que el medio esté libre durante un tiempo AIFS antes de reiniciar la cuenta descendente.

El tamaño mínimo de la ventana de contención, *CWmin*[AC] es otro parámetro dependiente de cada categoría. El valor inicial para el contador del *backoff* es un número aleatorio tomado de un intervalo definido por la ventana de contención, similar a DCF. La más pequeña es *CWmin*[AC], la cual proporciona la más alta prioridad para acceder al medio. Una gran diferencia entre 802.11 y 802.11e EDCA en términos de la regla de descuento del *backoff* es la siguiente:

- El primer descuento *backoff* ocurre al final del intervalo AIFSN[AC].
- La transmisión de un *frame* es iniciada en el momento en que el contador *backoff* llega a cero.

La probabilidad de colisión se incrementa cuando *CWmin*[AC] es muy pequeña y si hay más de una estación 802.11e operando en la misma categoría en el QBSS.

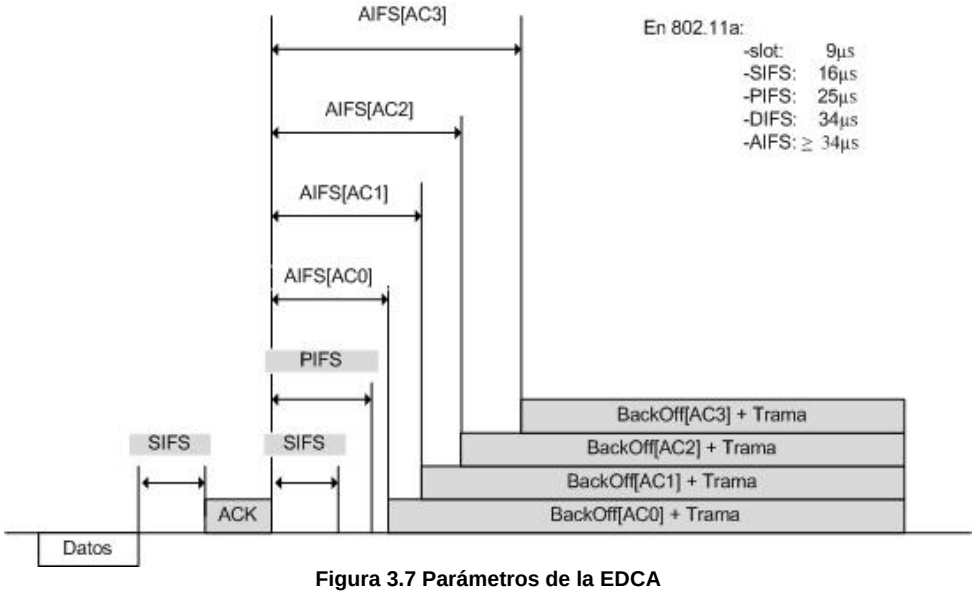



**Figura 3.7 Parámetros de la EDCA**

Tras un intento fallido de transmisión, el valor de la ventana de contienda de cada categoría de acceso se calcula en función de un factor de persistencia para dicha categoría de acceso y el valor anterior de la ventana de contienda. Este nuevo valor será mayor que el anterior con el fin de reducir la probabilidad de una nueva colisión. En la aproximación de 802.11 este factor de persistencia es 2, es decir, que tras cada colisión, la ventana de contienda se duplica. Sin embargo, en 802.11e se emplea la siguiente expresión:

$$CW[AC]nuevo \geq (CW[AC]anterior + 1)PF[AC] - 1$$

El nuevo valor de ventana de contienda nunca podrá exceder de un valor máximo para cada categoría de acceso.

La ventana de contención nunca excede el valor de CWmax[AC]. Este parámetro también es definido como parte del conjunto de parámetros EDCA. El más pequeño valor de CWmax[AC], proporciona la mayor prioridad para acceder al medio. Sin embargo, una CWmax[AC] con un valor pequeño puede incrementar la probabilidad de colisión.

Si los contadores de dos o más AC pertenecientes a una única QSTA expiran simultáneamente, se produce una colisión virtual que será resuelta por el planificador de cada QSTA, concediendo la TXOP a la AC con mayor prioridad. Sin embargo, todavía hay una probabilidad de que la trama transmitida colisione en el medio inalámbrico con la trama de otra estación.

La AC ganadora competirá entonces, por el uso del canal inalámbrico. El algoritmo de contienda externo es el mismo que el de DCF, salvo por el hecho de que los tiempos de espera y *backoff* no son constantes para un nivel físico dado. Una posible implementación consiste en ajustar adecuadamente los parámetros de la AC, de esta forma es posible optimizar el comportamiento de la red y la prioridad para cada tipo de tráfico. Para ello, es necesario un coordinador central que mantenga un conjunto común de parámetros de AC que garantice el acceso al medio justo para todas las QSTA de una QBSS. Además, para luchar contra la

asimetría de los sentidos de la comunicación, el QAP emplea valores de los parámetros de la EDCA específicos que tienen en cuenta las diferencias entre los canales ascendente y descendente.

El protocolo 802.11e también define un máximo tiempo de vida para un MSDU por categoría, esto quiere decir que hay un tiempo máximo para que un MSDU espere en la capa MAC. Una vez que el máximo tiempo de vida ha pasado desde que el *frame* llegó a la capa MAC, el *frame* es desechado sin haber sido transmitido. Esta característica es muy útil, ya que en las transmisiones en tiempo real el retardo de un paquete puede retrasar el envío de otros paquetes, por ello es mejor desecharlo, finalmente la pérdida de un paquete no es tan significativa en este tipo de aplicaciones.

Una vez que una estación 802.11e ha obtenido la oportunidad de transmitir, TXOP, la estación 802.11e puede continuar entregando más de un MSDU consecutivamente durante el mismo TXOP, el cual puede durar lo que indique el parámetro TXOPlimit[AC].

### 3.3.3 Acceso a canal controlado, HCCA

HCCA es un componente de HCF que soporta calidad de servicio basado en parametrización. Hereda alguna de las reglas de PCF e introduce algunas extensiones. De igual forma que en PCF, HCCA proporciona acceso basado en sondeo (*Polling*) al medio inalámbrico, pero, a diferencia del primero, el sondeo puede tener lugar en el periodo CP y la planificación de paquetes se basa en los perfiles TSPECs (Especificaciones de tráfico) admitidos.

En efecto la HCF permite al HC comenzar unos períodos libres de contienda durante el transcurso de un CP, siempre y cuando el medio de transmisión haya permanecido libre al menos en un PIFS. Este mecanismo es mucho más flexible que la PCF tradicional.

Durante el período de contienda CP, cada TXOP empieza cuando el medio está disponible según las reglas de EDCA, (es decir, un AIFS más el tiempo aleatorio resultante de la ejecución del algoritmo de *backoff* o bien cuando la estación 802.11e recibe una trama especial de encuesta del HC (QoS CF-Poll). El HC puede enviar la trama QoS CF-Poll después de esperar un PIFS y sin necesidad de ejecutar el algoritmo de *backoff* y por tanto, utilizar las TXOP en su beneficio aprovechándose en el acceso al medio. Durante el CFP, el HC envía un QoS CF-Poll a una QSTA concreta, a la que concede una TXOP, especificando el tiempo de inicio y la duración máxima. Puesto que, durante este tiempo, ninguna otra QSTA intentará ganar el acceso al medio, al recibir la trama de encuesta la QSTA podrá transmitir toda la información de que se disponga. Es importante mencionar que durante el periodo CFP la estaciones 802.11e no podrán acceder al medio, a menos que ellas hayan sido encuestadas explícitamente por el HC; solo el HC podrá ofrecer una oportunidad para transmitir (TXOP).

Tras recibir la trama de encuesta (QoS CF-Poll), se espera que la QSTA empiece a transmitir información antes de que transcurra un SIFS, si no lo hace, el HC tomará el control del medio pasando un PIFS y preguntará a otra QSTA. Un esquema de como éste permite utilizar el medio de manera muy eficiente durante el periodo libre de contienda. El CFP termina al transcurrir el tiempo indicado en la trama de *beacon* o con una trama CF-End.
Durante el tiempo TXOP, una estación encuestada puede transmitir múltiples *frames*, con un tiempo SIFS entre cada trama consecutiva, enviará tantos *frames* como se los permita el tiempo máximo que le fue otorgado para transmitir (TXOPlimit).

La figura 3.8 ilustra un ejemplo de un *superframe* que incluye un CFP y un CP. El *superframe* inicia con una trama *beacon* transmitida por el HC, en la figura se indica con el número 1. Durante el CFP (la primer parte del *superframe*), la estación sólo transmite el tiempo que le fue permitido por el QoS CF-poll y que le fue enviado desde el HC. El número 2 indica la transmisión de un MSDU fragmentado en el CFP. El CFP termina con el envío del *frame* CF-End, transmitido por el HC, señalado el número 3. Durante el siguiente periodo CP, todas las estaciones intentan transmitir a través del acceso al medio basado en contención del HCF (EDCA). Los EDCA-TXOPs son obtenidos a través de la contención. Dos EDCA-TXOPs son indicados en el número 4. Durante el CP, el HC también puede encuestar a una estación, esto es distinto a como ocurría en PCF. Los siguientes dos EDCA-TXOPs, el HC encuesta a la estación para reservar un TXOP durante el cual un fragmento de MSDU es transmitido, como se indica en 5.

El estándar 802.11e define un protocolo de acceso aleatorio que resuelve las colisiones rápidamente y que recibe el nombre de contienda controlada. El objetivo de este proceso es que el HC sepa qué nodos necesitan ser preguntados, en qué instantes y por cuánto tiempo para lo cual las QSTA envían actualizaciones al HC.

La contienda controlada permite a las QSTA solicitar la reserva de TXOP al HC enviando peticiones de recursos (RR, *Resource Request*). Este proceso tiene lugar durante el intervalo de contienda controlada que comienza cuando el HC envía una trama de control específica. Esta trama fuerza a las estaciones 802.11 convencionales a inicializar su temporizador NAV y por tanto, a permanecer en silencio hasta que la contienda controlada termine. Además, define un cierto número de oportunidades de contienda controlada disponibles, (separadas por SIFS) y una máscara que indica a qué categorías de acceso se asignarán los recursos solicitados. Cada QSTA con tráfico en cola para cada una de las AC indicadas en la máscara, escoge una oportunidad y transmite una trama de RR en la que se especifica la AC solicitada y la duración de la TXOP o el tamaño de la cola de AC solicitada. El HC emite una trama de reconocimiento para que las estaciones solicitantes puedan detectar colisiones durante la contienda controlada.

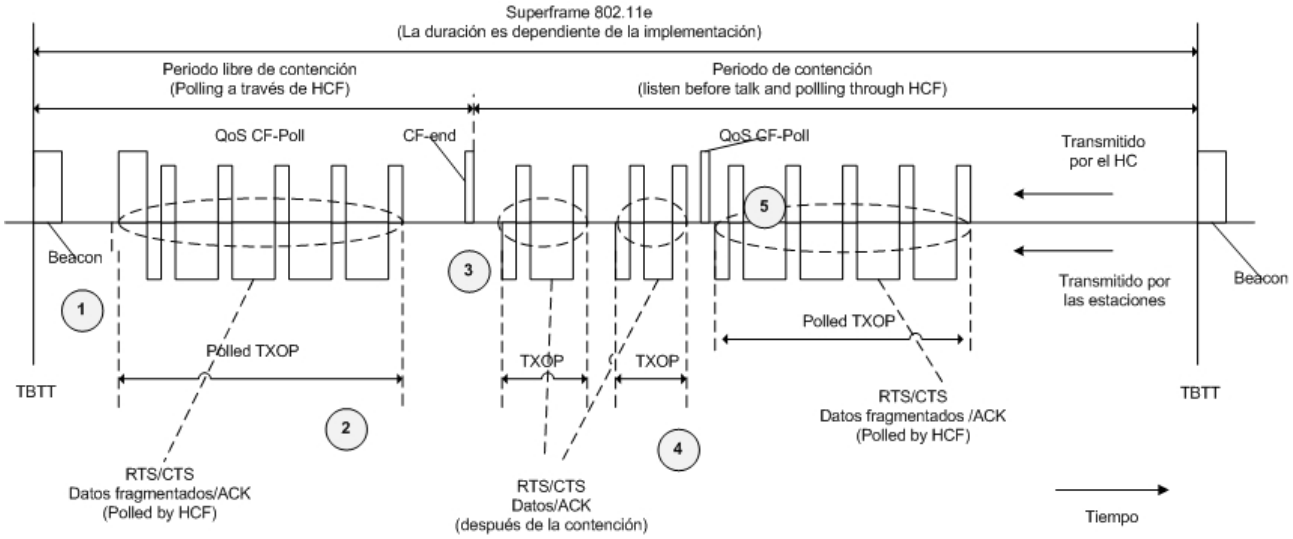



**Figura 3.8 Funcionamiento de 802.11e**

La HCF también incluye la señalización para negociar los parámetros de QoS de un flujo de datos específico, para lo cual introduce nuevo subtipos de trama. La trama QoS CF-Poll concede una HCCA-TXOP la cual acabará cuando se recibe una trama QoS-End procedente de un QAP o cuando la transferencia de información haya terminado. Todos estos tipos de tramas mejoran la eficiencia del nivel de MAC a costa de introducir una mayor complejidad.

La reservación de las oportunidades de transmisión (TXOPs) puede ser retrasada debido a la duración de un EDCA-TXOP, como se ilustra en la figura 3.9. Como ya sabemos el HC controla la duración máxima de los EDCA-TXOPs dentro de su QBSS a través del TXOPlimit[AC], para cada categoría vía el *beacon*. Cuando el tiempo para la entrega de un MSDU es muy pequeño, una extensión del tiempo será requerida. Aun cuando la entrega del MSDU no sea finalizada, el HC puede iniciar el envío de las tramas QoS CF-Poll y ocasionar una colisión como se muestra en la figura 3.9

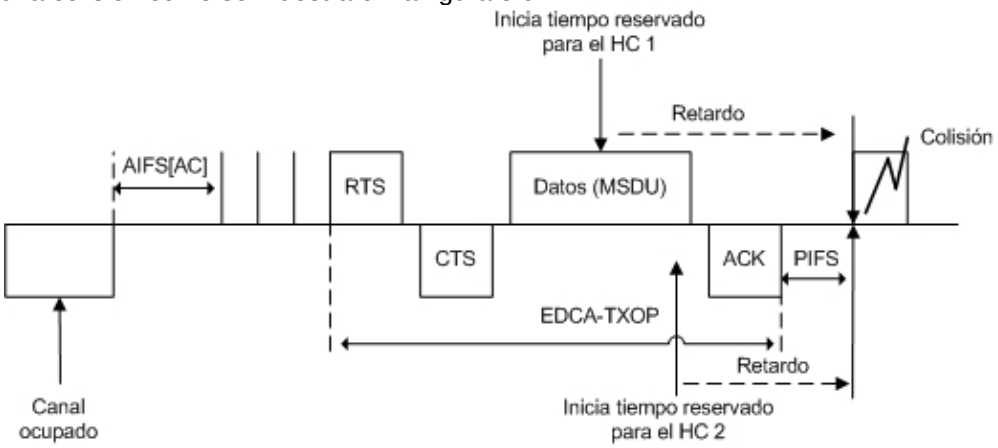



**Figura 3.9 Retraso en EDCA**

### 3.3.4 Especificaciones de tráfico - TSPEC

La especificación de tráfico (TSPEC), es el dispositivo de gestión de flujos de tráfico definido por el estándar 802.11e que proporciona un enlace de gestión entre protocolos de QoS de capas superiores, tales como

*Intserv* y *Diffserv*, con las funciones de acceso al canal de 802.11e. Esta especificación describe las características de los flujos de tráfico, tales como el tamaño de los paquetes, el caudal o el retardo. La negociación TSPEC proporciona un mecanismo para el control de la admisión, establecimiento, ajuste, y eliminación de flujos de tráfico.

El control de admisión es especialmente importante debido al limitado ancho de banda disponible en el medio inalámbrico. Este control permite evitar congestiones de tráfico que podrían llevar a anular enlaces ya establecidos o a una degradación del rendimiento general. El estándar 802.11e especifica el uso de TSPEC para este propósito tanto en EDCA como en HCCA.

# Capítulo 4 NS-2

## 4.1 *Network Simulator 2 (Simulador* de Redes versión 2*)*

NS-2 es un simulador de eventos discretos[1] orientado a redes que surgió a finales de los años 80. Incorpora capacidades de ruteo (*routing)* y *multicast*[2] en redes estructuradas e inalámbricas.

La razón principal por la que se escogió el simulador NS-2, para llevar a cabo este trabajo, es porque es una herramienta libre, muy completa y en constante desarrollo. Actualmente dicho simulador es muy utilizado en el ambiente académico para llevar a cabo diversas investigaciones. Es importante mencionar que, otra razón por la que se seleccionó el simulador NS-2, es porque se puede trabajar con 802.11e.

NS-2 está escrito en C++[3], usa OTcl[4] como *front-end*[5]. El simulador presenta dos jerarquías, la jerarquía compilada escrita en C++ y la jerarquía interpretada que corresponde a OTcl. Ambas se encuentran estrechamente relacionadas entre sí, ya que cada objeto presente en la jerarquía compilada encuentra su símil en la jerarquía interpretada.

En este trabajo se realizaron una serie de simulaciones, con el objetivo de imitar el funcionamiento de un sistema real durante un intervalo de tiempo. Para lograr simulaciones cuyos resultados sean confiables, NS-2 utiliza C++, pues es rápido para las ejecuciones y OTcl para la creación y configuración de escenarios. También utiliza *tclcl* para unir ambas cosas. En la figura 4.1 se resume la arquitectura del NS-2.

## 4.2 Creación de simulaciones

Para crear una simulación es necesario primero crear un objeto simulador con la instrucción:

**set ns [new Simulator]**.

Después hay que diseñar y crear la topología con la cual se trabajará, para esto es necesario crear los nodos y las ligas que formarán parte de dicha topología.

---

[1] La simulación de redes basada en eventos discretos, se refiere a que el avance de la variable tiempo depende de la temporización no continua de los eventos.
[2] ***Multicast***: Multidifusión. Es el envío de la información en una red a múltiples destinos simultáneamente
[3] **C++:** Es un lenguaje de programación diseñado a mediados de los años 1980, por *Bjarne Stroustrup*, como extensión del lenguaje de programación c.
[4] **OTcl**: Es una extensión orientada a objetos de TCL, creada por David Wethereall. Es utilizado en el simulador de redes NS-2
[5] ***Front-end***: Es la parte del software que interactúa con los usuarios. Es lo que visualiza el usuario.

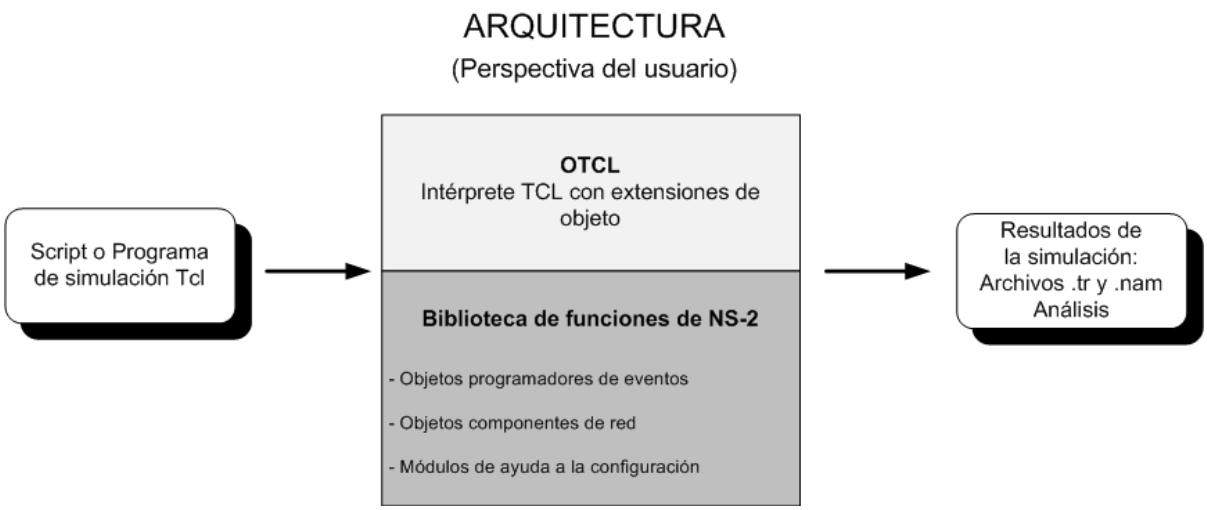
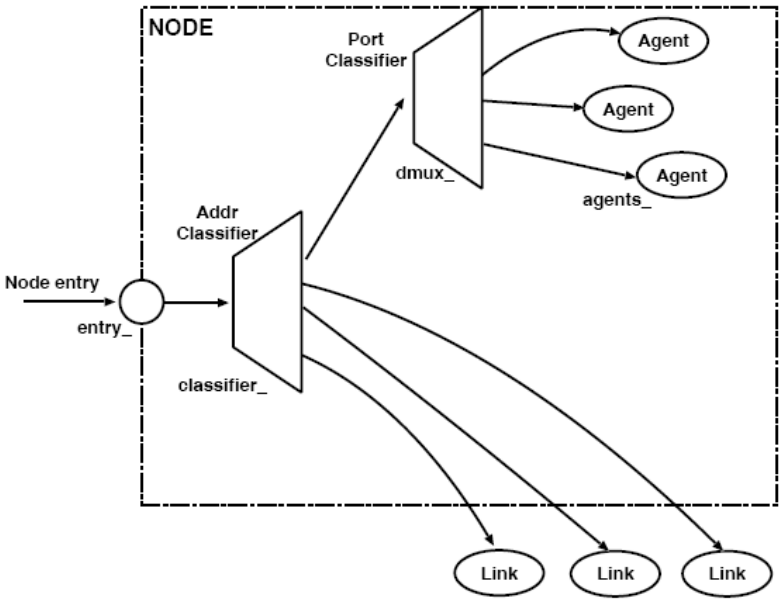



**Figura 4.1 Arquitectura general del NS-2**

### 4.2.1 Nodos

El objeto nodo puede crearse de la siguiente forma: **set n0 [$ns node]**




La estructura básica de un nodo consiste de dos objetos *TclObjects*: un clasificador de direcciones (*address classifier*) (classifier_) y un clasificador de puerto (*port classifier*) (dmux_). La función de estos clasificadores es, distribuir los paquetes de entrada hacia el agente correcto o el enlace de salida. Todos los nodos tienen al menos los siguientes componentes (La figura 4.2 muestra la arquitectura básica de un nodo):

- Una dirección o *id_*, se incrementa de uno en uno.
- Una lista de vecinos (*neighbor_*).
- Una lista de agentes (*agent_*).
- Un identificador de tipo de nodo (*nodetype_*).
- Un módulo de ruteo.

**Figura 4.2 Arquitectura básica de un nodo [VINT].**

Por default los nodos en NS-2 son construidos de tipo *unicast*[6] y son los nodos que se utilizaron para este trabajo.

### 4.2.2 Creación de escenarios

Es necesario crear una topología de red para trabajar de manera formal con un proyecto. Por ejemplo, en la figura 4.3 se ilustra una topología de 2 nodos y en la tabla 4.1 se muestra la programación correspondiente. Las líneas 1 y 2 crean los nodos, la línea 3 establece una conexión tipo Ethernet, a una velocidad de 1Mbps y un retardo (*delay*) de 10ms.

---

[6] ***Unicast***: Es un envío de información desde un único emisor a un único receptor. Se contrapone al *multicast*.

En esta topología nada sucede porque no hay envío de ningún tipo de tráfico. Para poner más interesante la cuestión vamos a generar tráfico.

### 4.2.3 Agentes, aplicaciones y generadores de tráfico

Los agentes más usados en NS-2 son UDP y TCP; ambos fueron usados en este trabajo. Las aplicaciones más comunes y los generadores de tráfico provistos por NS-2 son:

```
1 set n0 [$ns node]
2 set n1 [$ns node]
3 $ns duplex-link $n0 $n1 1Mb 10ms
DropTail
```

**Tabla 4.1 Creación de 2 nodos**

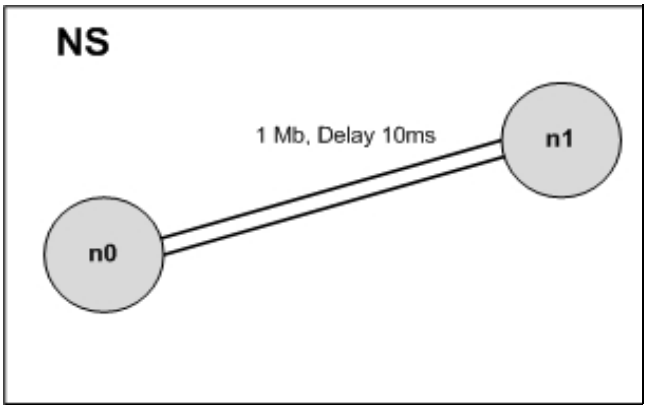



**Figura 4.3 Topología de dos nodos**

- FTP : *Application*/FTP
- Tráfico constante: *Application*/*Traffic*/CBR
- Registro de trazas: *Application*/*Traffic*/*Trace*

El tráfico UDP será enviado del nodo 1 al nodo 0. En el siguiente ejemplo (tabla 4.2) se crean dos nodos (líneas 1 y 2), se crea un agente tipo UDP (línea 3) y se relaciona con el nodo 0 (línea 4), por lo que el nodo 0 será el encargado de generar el tráfico tipo UDP. Se crea entonces un generador de tráfico ($cbr) y se adjunta al agente UDP (líneas 5 y 6), al generador se le indica que el tamaño de los paquetes que va a generar es de 1000 bytes, estos paquetes serán enviados cada 0.1 segundos (líneas 7 y 8).

Las siguientes líneas crean un agente nulo, el cual actuará como el nodo receptor de los datos generados por el nodo 0; este agente se adjunta al nodo 1 (líneas 10 y 11). Es necesario crear una conexión física entre los nodos, seguiremos usando Ethernet, es decir, conectaremos los nodos a través de un "cable" (línea 12), la transmisión se inicia en el segundo 0.5 y termina en el minuto 4.5 (líneas 14 y 15).

Para ejecutar la aplicación el NS-2 toma como entrada un script en OTcl como el que se muestra en la tabla 4.2. Como ya se mostró, en este script se define físicamente la configuración de la red (nodos, conexiones entre nodos), los protocolos que serán usados y definiciones específicas de aplicaciones usando tipos de conexiones.

```
1 set n0 [$ns node]
2 set n1 [$ns node]

3 set udp0 [new Agent/UDP]
4 $ns attach-agent $n0 $udp0

5 set cbr0 [new Application/Traffic/CBR]
6 $cbr0 attach-agent $udp0
7 $cbr0 set packet_size_ 1000
8 $cbr0 set interval_ 0.1

10 set null0 [new Agent/Null]
11 $ns attach-agent $n1 $null0
12 $ns connect $udp0 $null0
14 $ns at 0.5 "$cbr0 start"
15 $ns at 4.5 "$cbr0 stop"
```

**Tabla 4.2 Creación de nodos, agentes y tráfico**

El script es un archivo con extensión .tcl. Para hacerlo correr se debe ejecutar *ns ejemplo1.tcl* desde la línea de comandos y esto creará un archivo que contendrá la salida del análisis, un archivo de extensión .nam y un archivo de extensión .tr si así se configuró en el script.

### 4.2.4 Revisión de resultados con NAM (*Network Animator*)

Para simular los resultados de manera gráfica se utiliza una herramienta llamada NAM (*Network Animator*). NAM es una herramienta de animación basada en Tcl/TK[7] y sirve para ver la simulación de la trazas de la red y del tráfico real. Nam recibe como entrada el archivo con extensión .nam resultante de la simulación y lleva a cabo de manera gráfica el comportamiento de la red que se le haya especificado.

### 4.2.5 Nodos inalámbricos

En NS, el *MobileNode* es el objeto básico con funciones de movimiento, habilidad para transmitir y recibir en un medio inalámbrico. Técnicamente el *MobileNode* es derivado de clase base *Node*

Al inicio de una simulación inalámbrica se requiere definir el tipo o valores para cada uno los componentes mencionados. Adicionalmente, se necesita definir otro parámetro como el tipo de antena, el modelo de radio-propagación, el protocolo de ruteo, etc. En la tabla 4.3 se hace una breve descripción de cada variable. En este caso se utiliza un arreglo para definir estas variables: val().

En las redes inalámbricas es necesario configurar los nodos antes de crearlos, para configurarlos se realiza lo que se muestra en la tabla 4.5 y se pueden utilizar los valores definidos en el arreglo val().

```
set val(chan)    Channel/WirelessChannel  ;# tipo de canal
set val(prop)    Propagation/TwoRayGround ;# modelo de radio-propagación
set val(ant)     Antenna/OmniAntenna      ;# tipo de antena
set val(ll)      LL                       ;# tipo de capa de link
set val(ifq)     Queue/DropTail/PriQueue  ;# tipo de interface de cola
set val(ifqlen)  50                       ;# cantidad máx. de paquetes en la cola
set val(netif)   Phy/WirelessPhy          ;# tipo de interface de red
set val(mac)     Mac/802_11               ;# tipo MAC
set val(rp)      DSDV                     ;# protocolo de ruteo,DSR,TORA,AODV,NOAH
set val(nn)      2                        ;# número de nodos inalámbricos
```

**Tabla 4.3 Definición de variables**

### 4.2.6 Registro de eventos

Es posible registrar todos los eventos que ocurren en la simulación, generando un archivo de texto con toda la información. La instrucción para permitir el registro de los eventos en redes inalámbricas (líneas 1 y 2) se muestra en la tabla 4.4.

```
1 $ns trace-all [open salida.tr w]
2 $ns namtrace-all [open salida.nam w]
```

**Tabla 4.4 Registro de eventos**

La diferencia entre el archivo .nam y .tr es muy sencilla: .nam es el formato que lee el programa NAM y .tr es un formato más amigable si se requiere un análisis. Desde el punto de vista de contenido son similares, pero difieren sólo en el formato. Por lo tanto, este trabajo se enfocó en el análisis de los archivos .tr. En la figura 4.4 se describe brevemente la estructura del archivo.

## 4.3 NS-2 y 802.11e

Una vez instalado el NS-2, para lograr que funcione 802.11e, es necesario instalar un parche que modificará básicamente el funcionamiento de la capa MAC.

El parche seleccionado para realizar las simulaciones de diferentes escenarios en este trabajo, es el creado por el Grupo de Redes de Telecomunicación (TKN) de la Universidad Politécnica de Berlín (http://www.tkn.tu-berlin.de/research/802.11e_ns2/), y ha sido desarrollado por ***Sven Wiethölter*** **y** ***Christian Hoene*****.** Los autores tienen diferentes publicaciones, donde presentan esta herramienta a la comunidad científica y validan el correcto funcionamiento del mismo. A partir de su publicación, este parche ha sido

[7] **Tcl/TK**: Lenguaje interpretado de programación visual, basado en *widgets*, que genera código 100% portable. Ha sido desarrollado por la empresa *Sun Microsystems*.

utilizado por toda la comunidad científica, para la elaboración de múltiples artículos como herramienta de simulación de entornos inalámbricos 802.11e.

El parche para el soporte de 802.11e se encuentra disponible en http://sourceforge.net/projects/ieee80211e-ns2/. El software mencionado es un modelo de simulación sólo para el modo EDCF o EDCA, descrito en el capítulo 3, mismo que forma parte del estándar 802.11e y que funciona con la versión 2.28 del simulador NS-2.

```
$ns_ node-config -addressingType flat or hierarchical or expanded
                 -adhocRouting   $val(rp)
                 -llType          $val(ll)
                 -macType        $val(mac)
                 -propType       $val(prop)
                 -ifqType        $val(ifq)
                 -ifqLen         $val(ifqlen)
                 -phyType        $val(netif)
                 -antType        $val(ant)
                 -channelType    $val(chan)
                 -topoInstance   $topo
                 -agentTrace     ON o OFF
                 -routerTrace    ON o OFF
                 -macTrace       ON o OFF
                 -movementTrace  ON o OFF
```

**Tabla 4.5 Configuración de nodos inalámbricos**

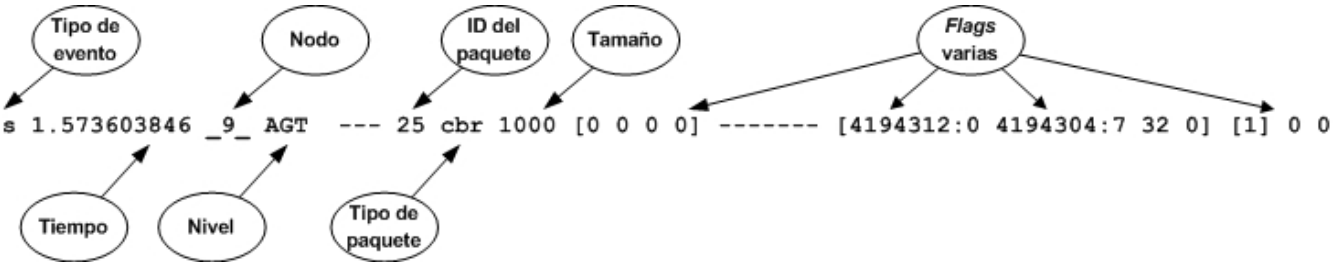



**Figura 4.4 Estructura del archivo .tr**

Este parche tiene la característica de que los parámetros de acceso a la capa de enlace 802.11e (Cwmax, Cwmin, AIFS, TXOP) deben cambiarse antes de que inicie la simulación.

Después de haber instalado el parche, podremos hacer uso de las diferentes colas de acceso proporcionadas por el estándar 802.11e para el acceso al medio. La configuración de los parámetros que se encargan del comportamiento de estas colas viene descrita en el archivo:

```
~/ns-allinone-2.28/ns-2.28/mac/802_11e/priority.tcl
```

Este archivo se reproduce en la tabla 4.6. Este archivo puede ser modificado según las necesidades, para adaptar los parámetros de configuración de las colas. Sin embargo las modificaciones son manuales, es decir, cada vez que se cambien, es necesario volver a compilar el ns y generar un nuevo ejecutable.

Para indicar la prioridad asignada a determinado nodo, es necesario indicarle cuál será la prioridad con la que trabajará. En la tabla 4.7, las líneas 2 y 3 le indican al agente que pertenece a la categoría o clase 1 y que llevará la prioridad 1. Esto en términos de calidad de servicio en 802.11e, indica que el tráfico a transmitir es tráfico de video, por lo que requiere de prioridad sobre el resto del tráfico.

```
# 802.11b parameters (default EDCA parameter set), aCWmin=31, aCWmax=1023
proc priority { ifq_name } {
```

```
   upvar $ifq_name ifq

    # parameters for Queue 0
    $ifq Prio 0 PF 2
    $ifq Prio 0 AIFS 2
    $ifq Prio 0 CW_MIN 7           ;# (aCWmin+1)/4 - 1
    $ifq Prio 0 CW_MAX 15          ;# (aCWmin+1)/2 - 1
    $ifq Prio 0 TXOPLimit 0.003264

    #parameters for Queue 1
    $ifq Prio 1 PF 2
    $ifq Prio 1 AIFS 2
    $ifq Prio 1 CW_MIN 15          ;# (aCWmin+1)/2 - 1
    $ifq Prio 1 CW_MAX 31          ;# aCWmin
    $ifq Prio 1 TXOPLimit 0.006016

    #parameters for Queue 2
    $ifq Prio 2 PF 2
    $ifq Prio 2 AIFS 3
    $ifq Prio 2 CW_MIN 31         ;# aCWmin
    $ifq Prio 2 CW_MAX 1023       ;# aCWmax
    $ifq Prio 2 TXOPLimit 0

    #parameters for Queue 3
    $ifq Prio 3 PF 2
    $ifq Prio 3 AIFS 7
    $ifq Prio 3 CW_MIN 31         ;# aCWmin
    $ifq Prio 3 CW_MAX 1023       ;# aCWmax
    $ifq Prio 3 TXOPLimit 0
}
```

**Tabla 4.6 Archivo de configuración de las categorías de acceso en 802.11e (Priority.tcl)**

```
1 set src_udp0 [new Agent/UDP]
2 $src_udp0 set class_ 1
3 $src_udp0 set prio_ 1
4 $src_udp0 set packetSize_ 1500
...
```

**Tabla 4.7 Establecimiento de prioridad 1 a un agente**

# Capítulo 5 Emulación

## 5.1 Emulación

La emulación se refiere a la técnica de introducir al simulador NS-2 el tráfico real de la red. Para lograr esto, se introducen al NS-2 una serie de objetos especiales; dichos objetos deben ser capaces de introducir tráfico real en el NS y de inyectar tráfico desde el simulador a la red.

Hay dos principales usos, cada uso depende de si el simulador aparece para la estación final como un *ruteador* o si aparece como alguna otra estación. En el primer caso, el tráfico real puede pasar a través del simulador (esto será transparente para los puntos finales) y puede ser afectado por los objetos de la simulación o por otro tráfico en la red. En el segundo caso, el simulador puede incluir algún generador de tráfico que se comunica con las entidades del mundo real. Actualmente el primer caso está más desarrollado que el segundo. En este trabajo se usó el primer caso.

La emulación puede ser dividida en dos modos:

- **Modo opaco (*opaque mode*)**: [VINT] El simulador trata los paquetes de la red como paquetes sin interpretar. En particular, los campos reales del protocolo no son manipulados por el simulador. En este modo, los paquetes pueden ser eliminados (*dropped*), retrasados (*delayed*) o duplicados, pero como no se lleva a cabo ningún procesamiento en cuanto al contenido de los paquetes, instrucciones como la siguiente no podrán realizarse: “Elimina el paquete TCP que es la retransmisión de la secuencia 4070”.

- **Modo protocolo (*protocol mode*)**: [VINT] Los paquetes de datos pueden ser interpretados o generados por el simulador.

La interface entre el simulador y la red está compuesta por una colección de objetos incluyendo los agentes “tap” y los objetos de red. Los agentes tap o interfaces tipo tap, ayudan a incrustar el tráfico real en la red en paquetes simulados y viceversa. Los objetos de red son instalados en los agentes tap y provistos de un punto de entrada para el envío y recepción del tráfico real. La figura 5.1 ilustra cómo estos objetos son usados para la emulación. Cuando se utiliza la emulación, se debe usar una versión especial del sistema planificador (*system scheduler*) llamada *RealTime scheduler*. Este planificador ayuda a manejar eventos en tiempo real. A continuación se muestran dos figuras, 5.2 y 5.3, que ilustran los dos modos de operación.

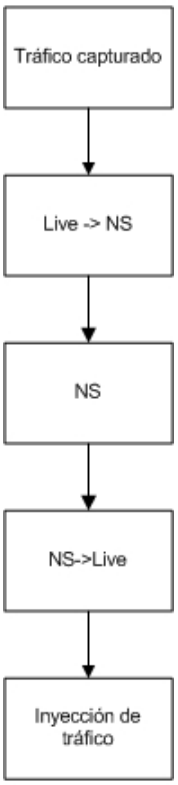



**Figura 5.1 Interacción del emulador con el simulador**

En la figura 5.2 se muestra que el simulador actúa como *ruteador*, permitiendo que el tráfico real pase a través del NS sin que éste sea manipulado. El paquete NS contiene un apuntador al paquete de red. Los paquetes de red pueden ser descartados o eliminados (*dropped*), retrasados (*delayed*), re-ordenados o duplicados por el simulador. El modo opaco es útil cuando el objetivo principal es

evaluar el comportamiento de las implementaciones de red en el mundo real pero sobre todo cuando éstas, están sujetas a condiciones adversas.

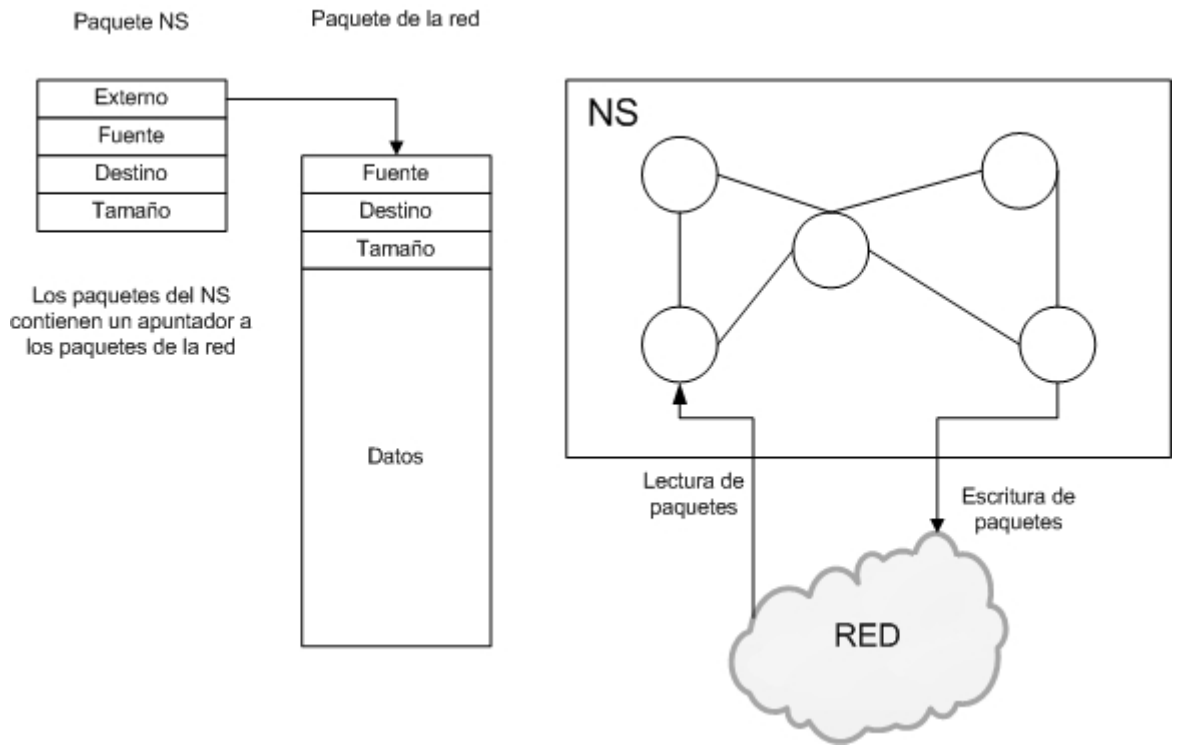



**Figura 5.2 Modo opaco: Los paquetes son pasados a través del simulador sin ser interpretados**

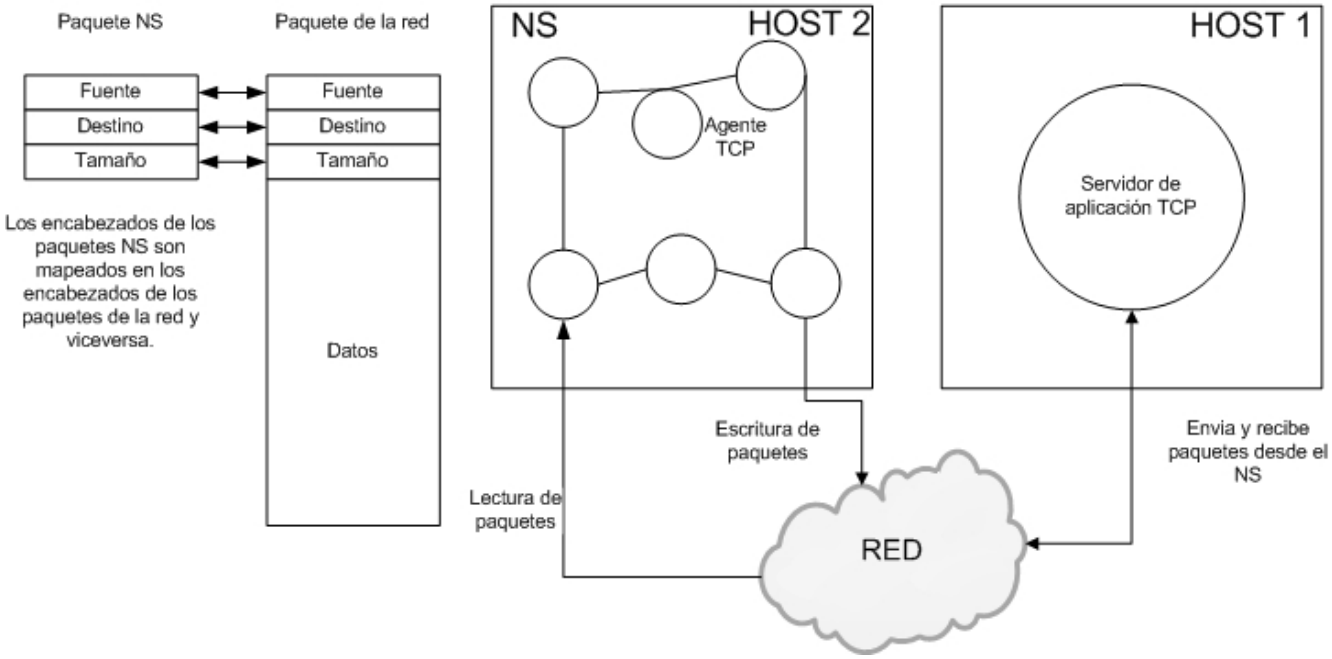



**Figura 5.3 Los paquetes son generados por el agente TCP que interactúa transparentemente con un servidor real de TCP.**

La figura 5.3 muestra que el simulador es usado como un punto para generar tráfico TCP. Un agente TCP dentro del NS, interactúa con un servidor de tráfico TCP real y puede recibir datos desde una aplicación

externa. NSE[1] permite soportar agentes tipo ICMP, ARP y TCP nat. El modo protocolo puede ser usado para probar una aplicación de inicio a fin.

## 5.2 Instalación y configuración del NSE

Se requiere el archivo parche, este puede bajarse en:
http://www-ivs.cs.uni-magdeburg.de/EuK/forschung/projekte/nse/index.shtml

Una vez instalado el NS se debe hacer lo mostrado en la tabla 5.1

```
1. [home@host]$ cd ns-allinone_version/ns-version
2. [home@host]$ patch -p0 < ~/ns2uml/ns2emulation/ns2emulation.diff
3. [home@host]$ ./configure
4. [home@host]$ make clean
5. [home@host]$ make
6. [home@host]$ su
7. Con esto debe aparecer un archivo ejecutable nse en el directorio
```

**Tabla 5.1 Procedimiento para instalar el NSE**

## 5.3 Creación de interfaces y agentes TAP

La clase *TapAgent* es una clase derivada de la clase *Agent*. Como tal, la clase *TapAgent* es capaz de generar paquetes en el simulador con valores asignados arbitrariamente dentro de los encabezados comunes del NS. El agente tipo TAP manipula los campos del tamaño y tipo de paquete en el encabezado. Los paquetes que son inyectados en el simulador quedan registrados en el archivo .tr como paquetes tipo *LIVE*. Cada agente TAP puede tener asociado al menos un objeto de red y más de un agente tap puede ser instanciado para un mismo nodo en la simulación.

## 5.4 Objetos de red

Los objetos de red proveen el acceso a la red real. Hay varias formas de crear los objetos de red, dependiendo de la capa del protocolo al que quiere accederse. Los objetos de red tienen un punto de entrada a la red real, por ejemplo *link*, *raw*, IP, UDP, etc., y con un modo de acceso particular (*read-only*, *write-only* or *read-write*). Algunos objetos de red proveen facilidades particulares como filtrado de tráfico, acceso en modo promiscuo, lo cual puede lograrse con el objeto de red PCAP/BFP o el objeto UDP/IP *multicast*. Actualmente son tres los objetos de red soportados:

- **Objeto PCAP/BPF**: Provee una librería para la captura de los paquetes conocida como ***libpcap***[2] .La captura de *frames* la realiza a nivel de la capa de enlace de datos. Este objeto también aporta la posibilidad de leer y escribir paquetes en formato de la salida de *tcpdump*[3].

- **Objeto IP**: Este objeto provee acceso crudo (*raw*) al protocolo IP. La implementación de este objeto hace uso de un *socket raw*[4].

---

[1] NSE: *Network Simulator Emulation*, es la versión del NS para poder trabajar con tráfico real.
[2] **Libpcap**: Es una librería que permite en modo promiscuo, capturar tráfico real de las interfaces de red. Más información en http://ee.lbl.gov/
[3] **Tcpdump**: Es una herramienta en línea de comandos cuya utilidad principal es analizar el tráfico que circula por la red.
[4] **Socket raw**: Un *socket raw* permite recibir o envíar el datagrama crudo sin incluir cabeceras del nivel de enlace, aunque en el presente trabajo no se utiliza esta capacidad de *socket raw*

- **Objeto UDP/IP**: Estos objetos proporcionan accesos a la implementación de UDP a través del soporte para multicast IP y operaciones de grupo. Este objeto aún se encuentra en desarrollo.

El objeto que se utilizó en este trabajo es PCAP/BPF. A continuación se muestra en la tabla 5.2 un ejemplo en el que se ilustra el uso de este objeto, de los agentes y la configuración en general.

En la línea 1 se declara la variable 'me' y se le asigna el valor de la IP del equipo desde donde se está corriendo la simulación. En la línea 2 se crea el objeto 'ns' necesario para correr cualquier simulación. La línea 3 específica que el objeto ns utilizará *RealTime* para poder trabajar con tráfico real en vivo. Las líneas 4 y 5 declaran objetos ($bpf0 y $bpf1) de red del tipo PCAP/BPF.

```
1   set me "10.0.1.1"
2   set ns [new Simulator]
3   $ns use-scheduler RealTime

4   set bpf0 [new Network/Pcap/Live]
5   set bpf1 [new Network/Pcap/Live]
6   $bpf0 set promisc_ true
7   $bpf1 set promisc_ true

8   set ipnet [new Network/IP]
9   set nd0 [$bpf0 open readonly eth0]
10  set nd1 [$bpf1 open readonly eth1]
11  $ipnet open writeonly

12  set notme "(not ip host $me)"
13  set notbcast "(not ether broadcast)"
14  set ftp "and port ftp-data"

15  set f0len [$bpf0 filter "(ip dst host bit) and $notme and $notbcast"]
16  set f1len [$bpf1 filter "(ip src host bit) and $notme and $notbcast"]

17  puts "filter lengths: $f0len (bpf0), $f1len (bpf1)"
18  puts "dev $nd0 has address [$bpf0 linkaddr]"
19  puts "dev $nd1 has address [$bpf1 linkaddr]"

20  set a0 [new Agent/Tap]
21  set a1 [new Agent/Tap]
22  set a2 [new Agent/Tap]

23  puts "install nets into taps..."
24  $a0 network $bpf0
25  $a1 network $bpf1
26  $a2 network $ipnet

27  set node0 [$ns node]
28  set node1 [$ns node]
29  set node2 [$ns node]
30  $ns simplex-link $node0 $node2 10Mb 10ms DropTail
31  $ns simplex-link $node1 $node2 10Mb 10ms DropTail

32  $ns attach-agent $node0 $a0
33  $ns attach-agent $node1 $a1
34  $ns attach-agent $node2 $a2
35  $ns connect $a0 $a2
36  $ns connect $a1 $a2
37  puts "okey"
38  $ns run
```

**Tabla 5.2 Ejemplo de emulación**

Las líneas 6 y 7 le indican a los objetos que deben funcionar en modo promiscuo. La línea 8 crea un objeto de red del tipo IP. Las líneas 9 y 10 relacionan los objetos de red $bpf0 y $bpf1 con las interfaces eth0 y

eth1, especificando también que el modo será únicamente de lectura y lo asignan a las variables $nd0 y nd1. En la línea 11 se indica que el objeto de red $ipnet se trabajará en modo de lectura. Las líneas 12-14 declaran y asignan valores a variables de texto que se usarán más adelante. En las líneas 15-16 se declaran y aplican los filtros al tráfico, por ejemplo se escuchará todo el tráfico excepto el que no esté dirigido a la IP declarada en la variable $me y el tráfico de broadcast[5]. Las líneas 17-19 sólo envían mensajes de texto a la pantalla.

Las líneas 20-22 crean agentes TAP ($a0, $a1 y $a3). La línea 23 envía el mensaje de que los agentes tap serán instalados en los objetos de red. En las líneas 24-26 se instalan los agentes TAP en los objetos de red a través de instrucción 'network'.

Las líneas 27-29 crean 3 nodos (node0, node1 y node2). En las líneas 30-31 se crea la interconectividad, como es un ejemplo de red cableada (Ethernet), en la línea 30 se declara un "cable" tipo *half-duplex* entre el nodo 0 y el nodo 2 con un ancho de banda de 10 Mbps y un retardo de 10 ms. De forma similar de declara la interconectividad entre el nodo 1 y nodo 2 quedando entonces como se muestra en la figura 5.4.

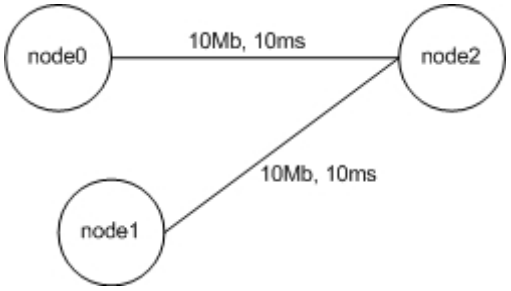



**Figura 5.4 Diagrama de red**

Las líneas 32-34 adjuntan cada uno de los nodos a cada agente TAP. La línea 35 realiza la conexión del agente 0 ($a0) con el agente 2 ($a2) y la línea 36 conecta el agente 1 (a1) con el agente 2 ($a2). Las línea 37 escribe un mensaje en pantalla señalando que todo está correcto y configurado. Finalmente la línea 38 indica la ejecución de la simulación.

## 5.5 Configuración del escenario utilizado en este trabajo

Como se mencionó, el objeto tipo PCAP/BPF junto con la librería *libpcap,* fueron utilizados para conectar el simulador con el tráfico real. Para crear el escenario de configuración de este trabajo se utilizaron, en general, dos máquinas virtuales; una sirvió de servidor de video y la otra como receptor de dicho video.

### 5.5.1 Máquinas virtuales usadas

### UML (User-Mode Linux)

*User Mode Linux* (Linux en Modo Usuario) es el *Kernel*[6] de Linux portado a su propia interfaz de llamadas al sistema. Provee una especie de máquina virtual, que corre Linux como un proceso dentro de otro *kernel* de Linux.
Linux en modo de usuario provee una máquina virtual que puede tener más recursos virtuales de hardware y software que el equipo de cómputo que lo corre. El almacenamiento en disco de la máquina virtual se aloja en un archivo en la máquina huésped. Es importante aclarar que los procesos dentro de UML corren nativamente, en el procesador de la máquina huésped.

### *VMware Workstation*

[5] **Broadcast**: Se refiere a la difusión y es un modo de transmisión de información en el que el nodo emisor envía información a una multitud de nodos receptores de manera simultánea, sin necesidad de reproducir la misma transmisión nodo por nodo.
[6] **Kernel**: Es el núcleo del sistema operativo Linux y es el encargado de que software y el hardware del equipo de cómputo puedan trabajar coordinadamente.

***VMware*** es un sistema de virtualización por software. Un sistema virtual por software, es un programa que simula un sistema físico (un ordenador, un hardware) con unas características de hardware determinadas. Cuando se ejecuta dicho programa, proporciona un ambiente de ejecución similar al de un ordenador físico, excepto en el puro acceso físico al hardware simulado, con CPU (puede ser más de uno), BIOS, tarjeta gráfica, memoria RAM, tarjeta de red, sistema de sonido, conexión USB, disco duro (pueden ser más de uno), etc. La versión de *VMware* que se utilizó en este trabajo se llama ***VMware Worstation 6.0*.**

***VMware Worstation*** es uno de los más utilizados pues permite la emulación en plataformas PC x86, esto permite que cualquier usuario con una computadora de escritorio o laptop pueda emular tantas máquinas virtuales como los recursos de hardware lo permitan. Esta versión es una aplicación que se instala dentro de un sistema operativo (*host*) como un programa estándar, de tal forma que las máquinas virtuales corren dentro de esta aplicación, existiendo un aprovechamiento restringido de recursos. La imagen 5.5 muestra la consola de *VMware Workstation* y la 5.6 ilustra una instancia de *Windows XP* sobre *Ubuntu* 7.4.

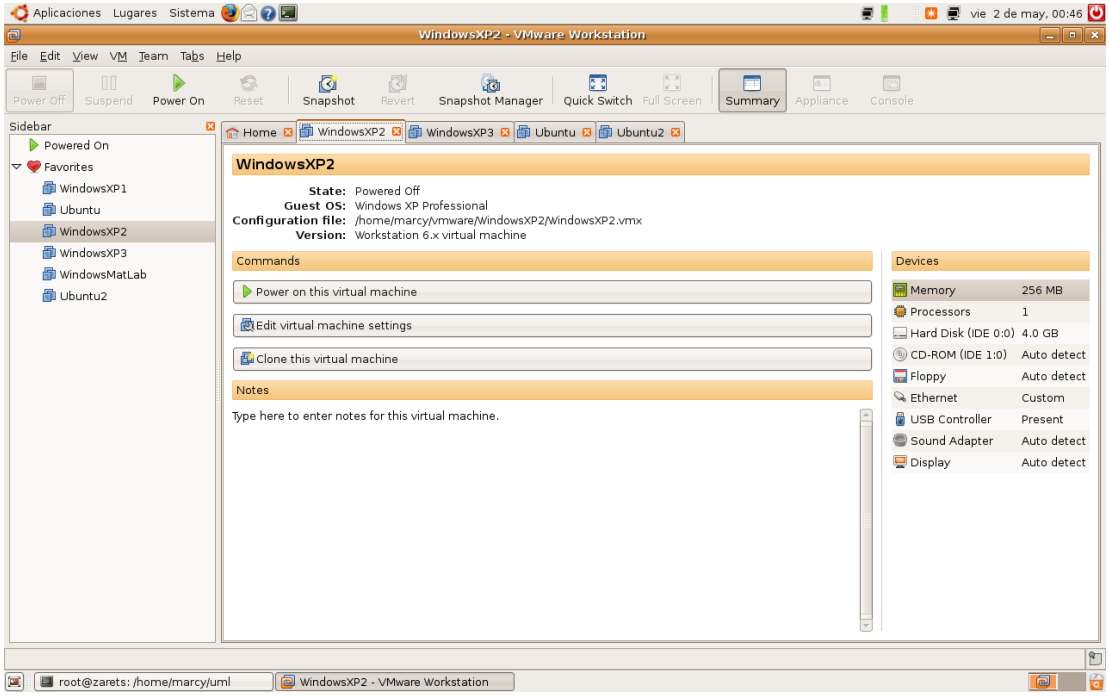



**Figura 5.5 Consola de *VMware Workstation***

## 5.5.2 Interfaces TAP

Una interfaz TAP es un dispositivo de red virtual. El procedimiento para crear las interfaces se muestra en la tabla 5.3. La línea 1 crea la interfaz tap0, la línea 2 le asigna una IP a la interfaz, la línea 4 indica la IP que se le va a configurar a la máquina virtual y la línea 6 asocia la IP que será asignada a la máquina virtual, con la interfaz de la máquina base.

```
1  tunctl -u root
2  ifconfig tap0 192.168.0.20 netmask 255.255.255.0
3  echo 1 > /proc/sys/net/ipv4/ip_forward
4  route add -host 192.168.0.30 dev tap0
5  echo 1 > /proc/sys/net/ipv4/conf/tap0/proxy_arp
6  arp -Ds 192.168.0.30 eth1 pub
```

**Tabla 5.3 Procedimiento para crear interfaces TAP**

Una vez creada la interfaz TAP, se inicia la máquina virtual; como IP se le debe configurar la que está en la instrucción de la línea 4 ó 6.

## 5.5.3 Servidor de video VideoLAN

La herramienta utilizada para el envío y recepción del tráfico de video fue VideoLAN ( http://www.videolan.org ). VideoLAN es una solución de software para transmisión de vídeo (ver la figura

5.7), desarrollada por estudiantes de *Ecole Centrale Paris* ( http://www.ecp.fr ) y desarrolladores de todo el mundo, dentro de GNU *General Public License*[7]. VideoLAN está diseñado para transmitir vídeo MPEG[8] en redes de datos.

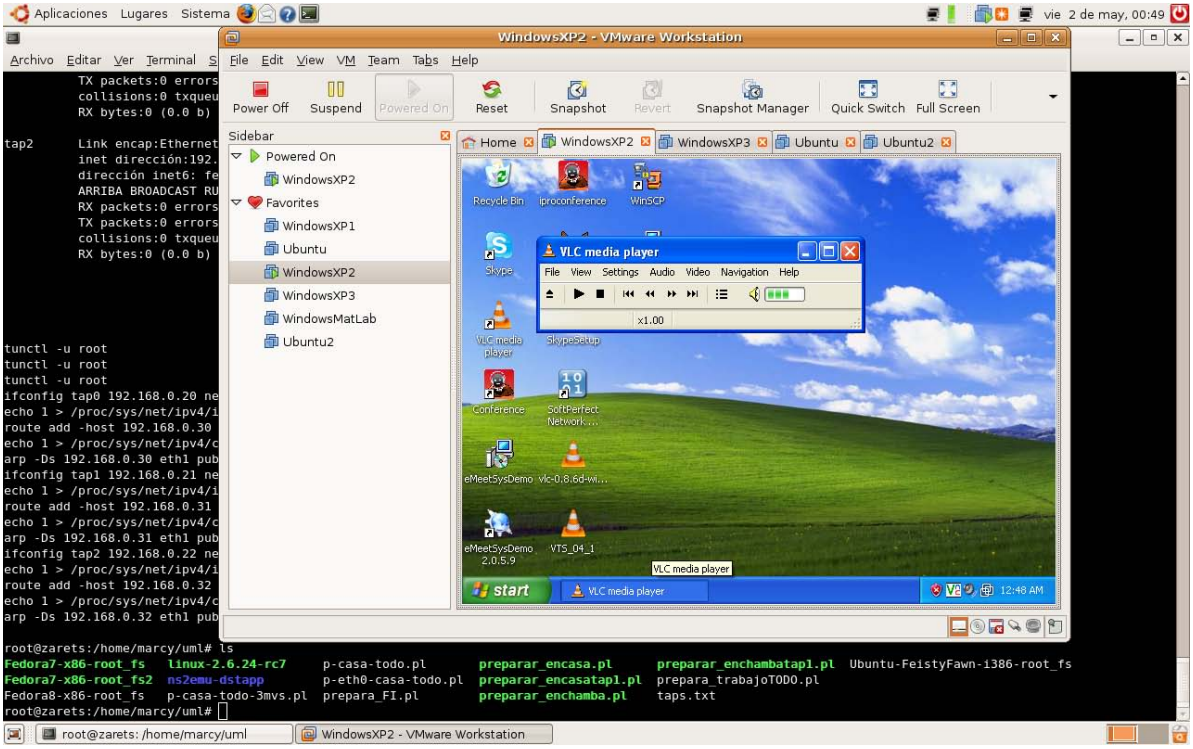



**Figura 5.6 Windows XP corriendo sobre Ubuntu 7.04**

La solución VideoLAN incluye :

- **VLS** (Servidor VideoLAN), el cual puede transmitir archivos MPEG-1, MPEG-2 y MPEG-4, DVDs, canales digitales de satélite, canales digitales de televisión terrestre y video en vivo sobre la red en *unicast* o *multicast*,

- **VLC** (Cliente y Servidor VideoLAN), el cual puede ser usado como servidor para transmitir archivos MPEG-1, MPEG-2 y MPEG-4, DVDs y video en vivo sobre la red en unicast o multicast; o usado como cliente para recibir, decodificar y visualizar flujos MPEG sobre varios sistemas operativos.

El utilizado en este trabajo fue VLC, pues puede funcionar como cliente y servidor.

---

[7] **GNU *General Public License***: La licencia pública general de GNU (GNU GPL), es una licencia creada por la *Free Software Foundation* (Fundación de Software Libre) a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

[8] **MPEG (*Moving Picture Experts Group*)**: Se refiere al Grupo de Expertos de Imágenes en Movimiento conocido comúnmente como MPEG y es un grupo de trabajo encargado de desarrollar estándares de codificación de audio y video.

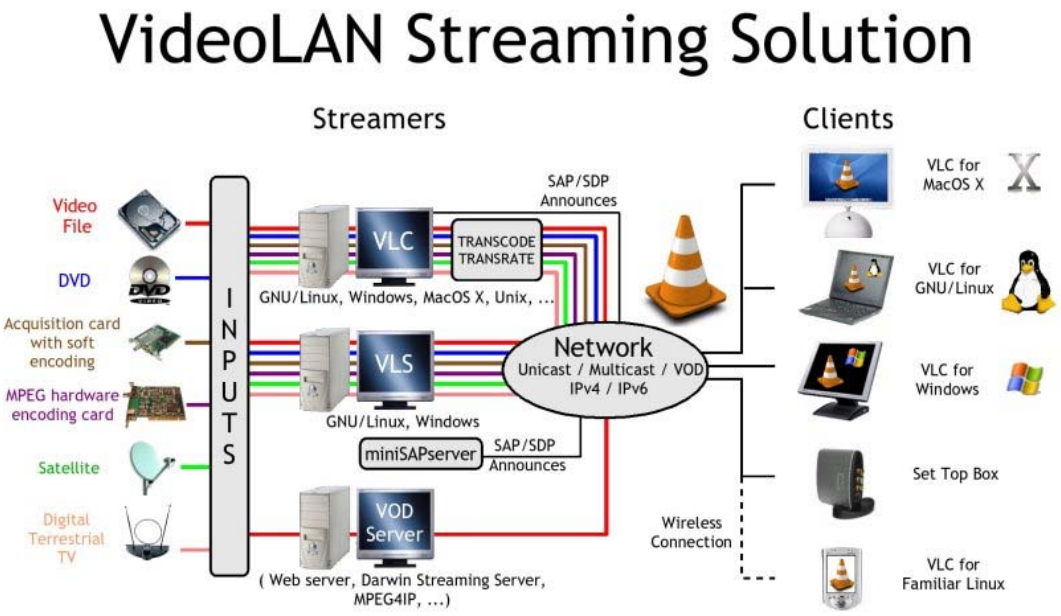



**Figura 5.7 VideoLAN. [VIDEOLAN]**

### 5.5.4 Skype

Es un software para realizar llamadas sobre Internet. El éxito de esta herramienta reside en la gran compresión que utiliza, sin que ésta afecte prácticamente a la calidad de la transmisión de voz. El funcionamiento de *Skype* consiste básicamente en establecer una conexión con un clúster de servidores (servidores redundantes) de *Skype* para iniciar sesión, en la cual se devuelve la lista de contactos. Cuando se inicia una llamada se establece una conexión directa con la persona, eliminando así el consumo de ancho de banda utilizado por la voz en los servidores de *Skype* e incrementando la seguridad, al ser una conexión directa.

# Capítulo 6 Experimentos y Resultados

## 6.1 Experimentos

El tipo de tráfico utilizado en los experimentos fue de dos tipos:

**UDP (*User Datagram Protocol,* Protocolo de Datagramas de Usuario)**: Es un protocolo del nivel de transporte, basado en el intercambio de datagramas. Permite el envío de datagramas a través de la red, sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera. Tampoco tiene confirmación, ni control de flujo, por lo que los paquetes pueden adelantarse unos a los otros; no se sabe si los paquetes han llegado correctamente porque no hay confirmación de entrega o recepción.

**TCP (*Transmission Control Protocol*, Protocolo de Control de Transmisión)**: Es un protocolo orientado a conexión, a nivel de la capa de transporte. Este protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron. Proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina, a través del concepto de puerto.

Los experimentos realizados se dividieron en cinco grupos, con las siguientes características:

1. Con tráfico tipo UDP-CBR[1] generado por el simulador. Con 802.11 y 802.11e. Prioridad asignada al nodo 1. Ver figura 6.1.
2. Con tráfico tipo TCP, FTP[2] generado por el simulador. Con 802.11 y 802.11e. Prioridad asignada al nodo 1. Ver figura 6.1.
3. Con tráfico tipo TCP, FTP generado por el simulador. Con 802.11e. Prioridad asignada a los nodos 1, 2, 3, 4, 5, 6, 7, 8, 9 y 10. Ver figura 6.1.
4. Con tráfico real de video (*VideoLAN*) generado por las máquinas virtuales y tráfico generado por el simulador. Con 802.11e.
5. Con tráfico real de voz (*Skype*), generado por las máquinas virtuales y tráfico generado por el simulador. Con 802.11e.

Los aspectos que se midieron fueron:

- *Throughput* por terminal
- *Delay* por terminal
- *Drops* por terminal

El escenario utilizado se muestra en la figura 6.1. Los parámetros configurados fueron:

- Paquetes de tamaño 100 , 512, 1024, 1500 (bytes); una simulación para cada tamaño de paquete.
- Duración de cada simulación: 500 segundos.

---

[1] **CBR (*Constant Bit Rate,* Tasa de Transmisión Constante**): Se refiere a una característica del canal de transmisión en que la etapa de codificación conserva constante la tasa de bits enviada.

[2] **FTP (*File Transfer Protocol*, Protocolo de Transferencia de Archivos)**: Es un protocolo de transferencia de archivos entre sistemas conectados a una red TCP basado en arquitectura cliente – servidor.

- Intervalo para envío de paquetes: 1 segundo.
- La configuración para los nodos inalámbricos se muestra en la tabla 6.1

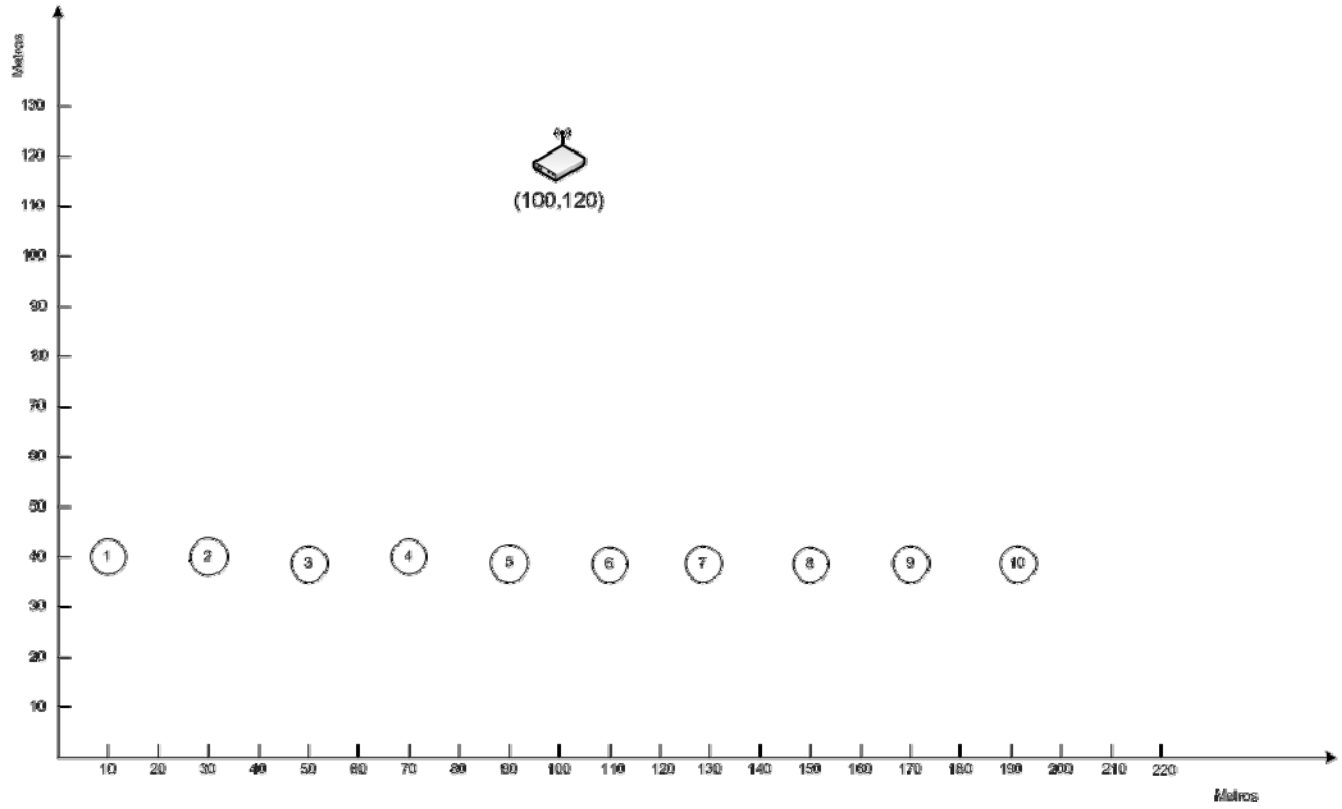



**Figura 6.1 Escenario utilizado**

```
set opt(chan)       Channel/WirelessChannel
set opt(prop)       Propagation/TwoRayGround
set opt(netif)      Phy/WirelessPhy
set opt(mac)        Mac/802_11 o Mac/802.11e
set opt(ifq)        Queue/DropTail/PriQueue
set opt(ll)         LL
set opt(ant)        Antenna/OmniAntenna
set opt(x)          670      ;# X dimension of the topography
set opt(y)          670      ;# Y dimension of the topography
set opt(ifqlen)     50       ;# max packet in ifq
set opt(seed) 0.0
#set opt(tr)        sal.tr   ;# trace file
set opt(lm)         "off"    ;# log movement

$ns_ node-config -adhocRouting  NOAH \
                    -llType   $opt(ll) \
                    -macType  $opt(mac) \
                    -ifqType  $opt(ifq) \
                    -ifqLen   $opt(ifqlen) \
                    -antType  $opt(ant) \
                    -propType $opt(prop) \
                    -phyType  $opt(netif) \
                    -channel [new $opt(chan)] \
                    -topoInstance $topo \
                   -wiredRouting OFF \
                    -agentTrace   ON \
                    -routerTrace  OFF \
                    -macTrace     OFF \
                    -movementTrace OFF
```

```
# El generador de tráfico UDP-CBR para 802.11
TestSuite instproc create-udp-traffic {id src dst start pkt} {
    global null_
    $self instvar ns_
    set udp_($id) [new Agent/UDP]
    $ns_ attach-agent $src $udp_($id)
    set null_($id) [new Agent/LossMonitor]
    $ns_ attach-agent $dst $null_($id)
    set cbr_($id) [new Application/Traffic/CBR]
    $cbr_($id) set packetSize_ $pkt
    $cbr_($id) set interval_ 1.0
    $cbr_($id) set random_ 1
    $cbr_($id) attach-agent $udp_($id)
    $ns_ connect $udp_($id) $null_($id)
    $ns_ at $start "$cbr_($id) start"
    puts [$cbr_($id) set packetSize_]
    puts [$cbr_($id) set interval_]

}

#El generador de tráfico UDP-CBR para 802.11e
TestSuite instproc create-udp-traffic {id src dst start pkt clase} {
    puts "Voy a crear trafico udp $pkt"
    $self instvar ns_
    set udp_($id) [new Agent/UDP]
    $udp_($id) set class_ $clase
    $udp_($id) set prio_ $clase
    $ns_ attach-agent $src $udp_($id)
    set null_($id) [new Agent/LossMonitor]
    $ns_ attach-agent $dst $null_($id)
    set cbr_($id) [new Application/Traffic/CBR]
    $cbr_($id) set packetSize_ $pkt
    $cbr_($id) set interval_ 1.0
    $cbr_($id) set random_ 1
    $cbr_($id) attach-agent $udp_($id)
    $ns_ connect $udp_($id) $null_($id)
    $ns_ at $start "$cbr_($id) start"
    puts [$cbr_($id) set packetSize_]
    puts [$cbr_($id) set interval_]

}
```

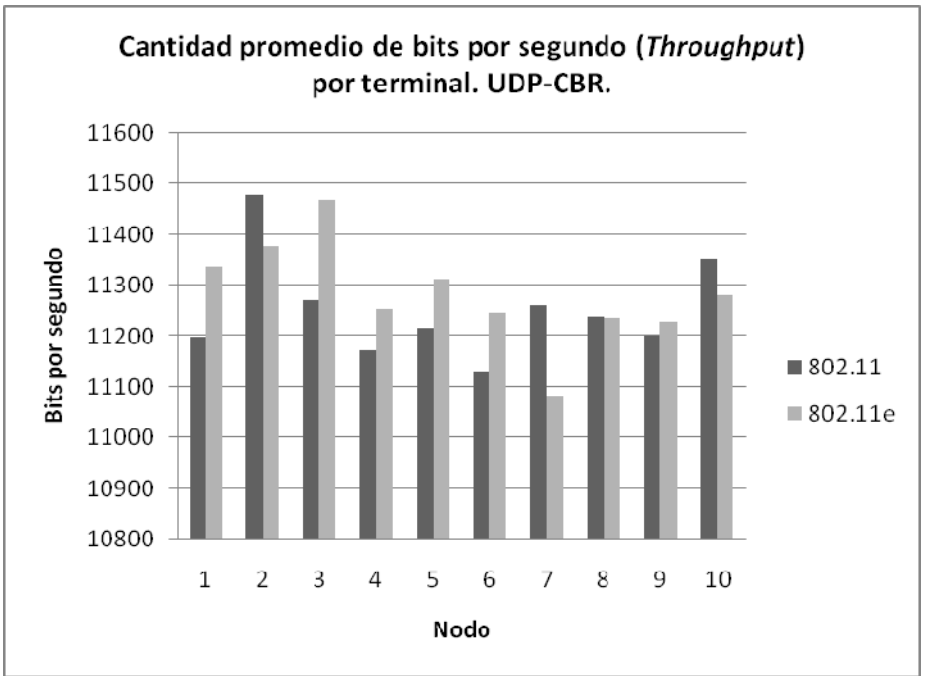
**Tabla 6.1 Configuración de los nodos**

Los resultados resumidos son los siguientes:

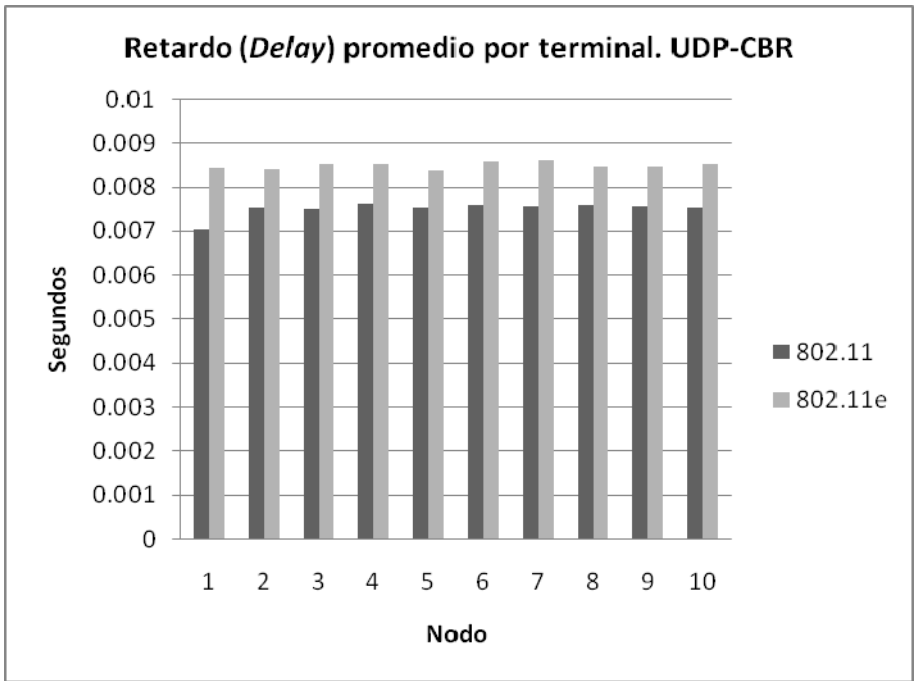
### 6.1.1 Experimentos con tráfico tipo UDP-CBR generado por el simulador. Con 802.11 y 802.11e.

#### Prioridad asignada al nodo 1.

En las gráficas 1 y 2 se puede apreciar que la diferencia entre 802.11 y 802.11e no es muy significativa y que cuando se utilizó 802.11e y se le asignó prioridad al nodo 1, no hay un incremento muy representativo; de hecho tiene, en promedio, más *throughput* el nodo 3 al cual no le fue asignada prioridad de envío.




**Gráfica 1**



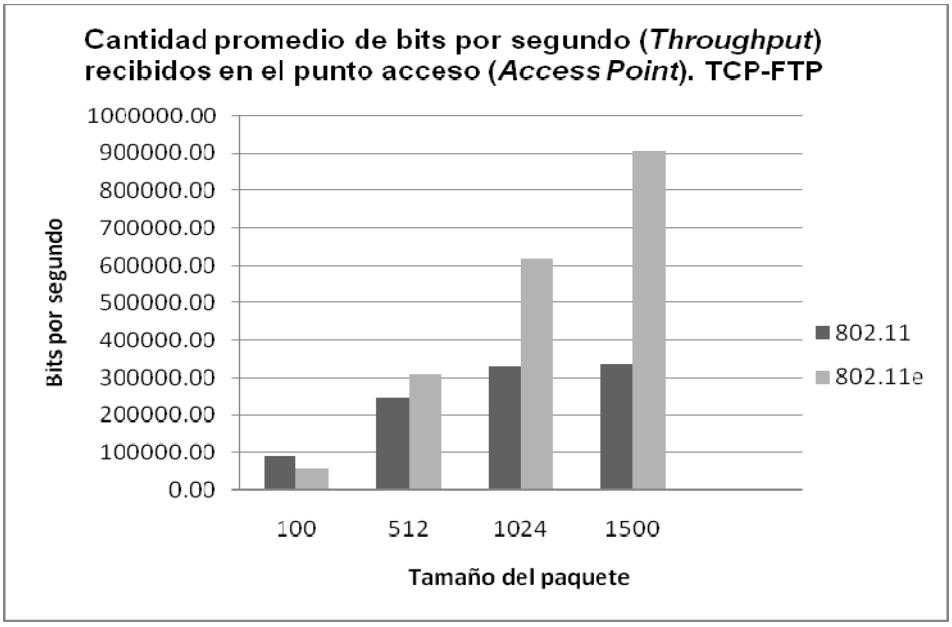

**Gráfica 2**

### 6.1.2 Experimentos con tráfico tipo TCP-FTP generado por el simulador. Con 802.11 y 802.11e.

#### Prioridad asignada al nodo 1.

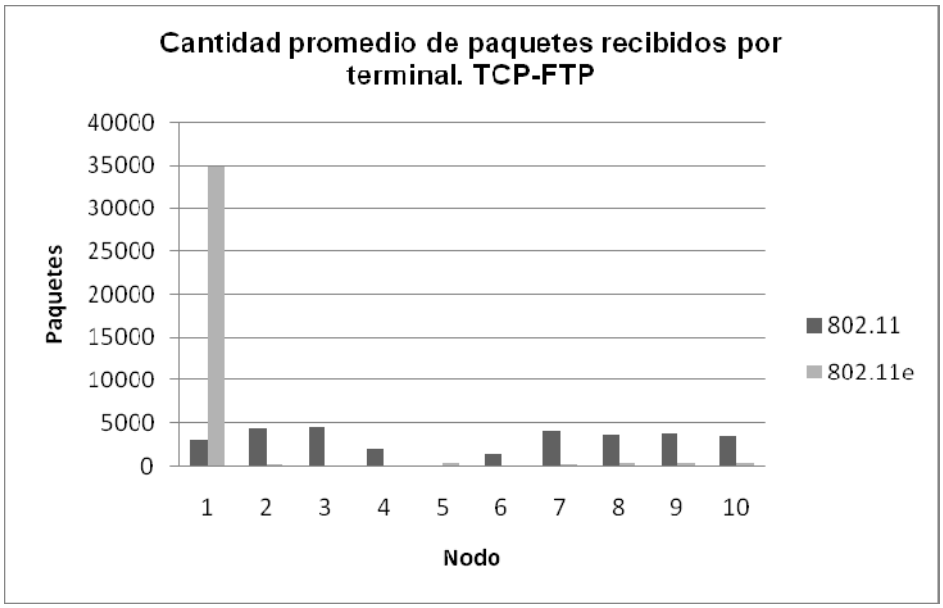
Al utilizar tráfico tipo TCP-FTP y comparar el desempeño de 802.11 y 802.11e, se encontraron los siguientes resultados:

En la gráfica 3 se ilustra el *throughput* recibido en el *Access Point*, se puede apreciar la mejora cuando se utiliza 802.11e. La prioridad sólo está asignada al nodo 1.



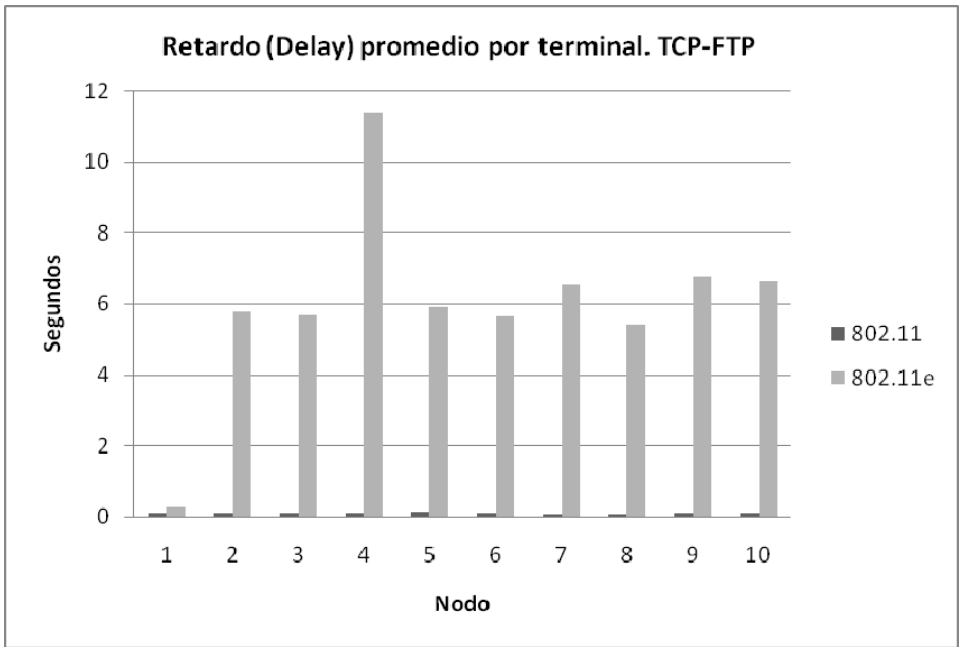

**Gráfica 3**

En la gráfica 4 se muestra la cantidad de paquetes recibidos utilizando tráfico TCP-FTP y comparando 802.11 y 802.11e. La asignación de la calidad de servicio es al nodo 1. Se observa cómo la mejora de 802.11e respecto a 802.11 se aprecia mejor con el tráfico tipo TCP-FTP.




**Gráfica 4**

La gráfica 5 muestra el retardo por terminal. Al utilizar 802.11e y asignar la prioridad al nodo 1, éste es el que tiene menos retardo que el resto. Si se utiliza 802.11, el retardo es similar en todos los nodos. El retardo es más alto en 802.11e porque hay un nodo al que se le está asignando prioridad por sobre el resto.




**Gráfica 5**

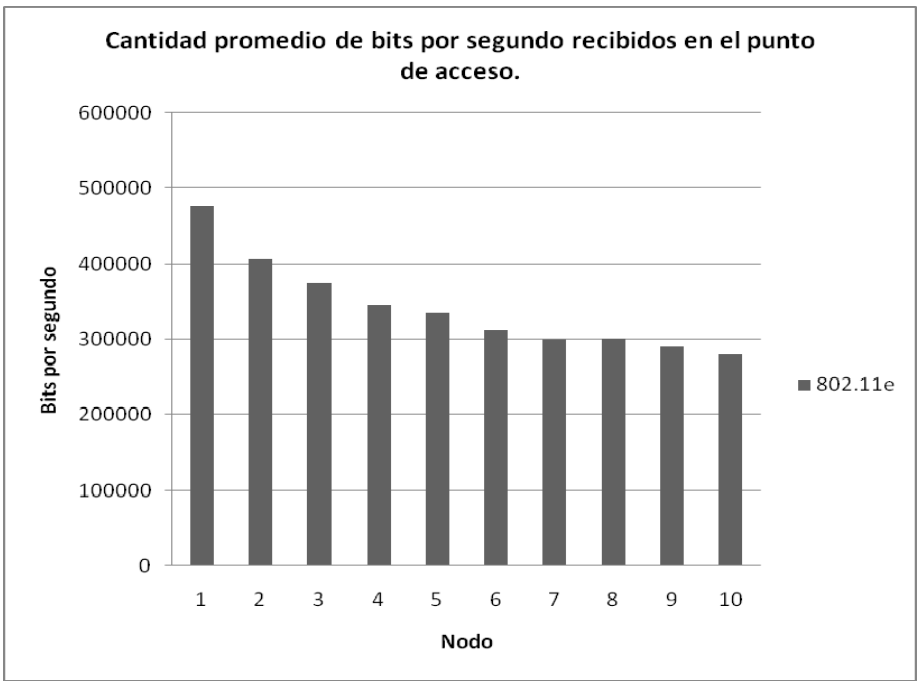
### 6.1.3 Experimentos con tráfico tipo TCP-FTP generado por el simulador. Con 802.11e. Prioridad asignada a los nodos 1, 2, 3, 4, 5, 6, 7, 8, 9 y 10.

La gráfica 6 ilustra el *Throughput* promedio que recibe el punto de acceso y que viene de cada uno de los diez nodos de la simulación. Entre menos nodos tengan prioridad será mayor el *throughput* recibido en el AP.

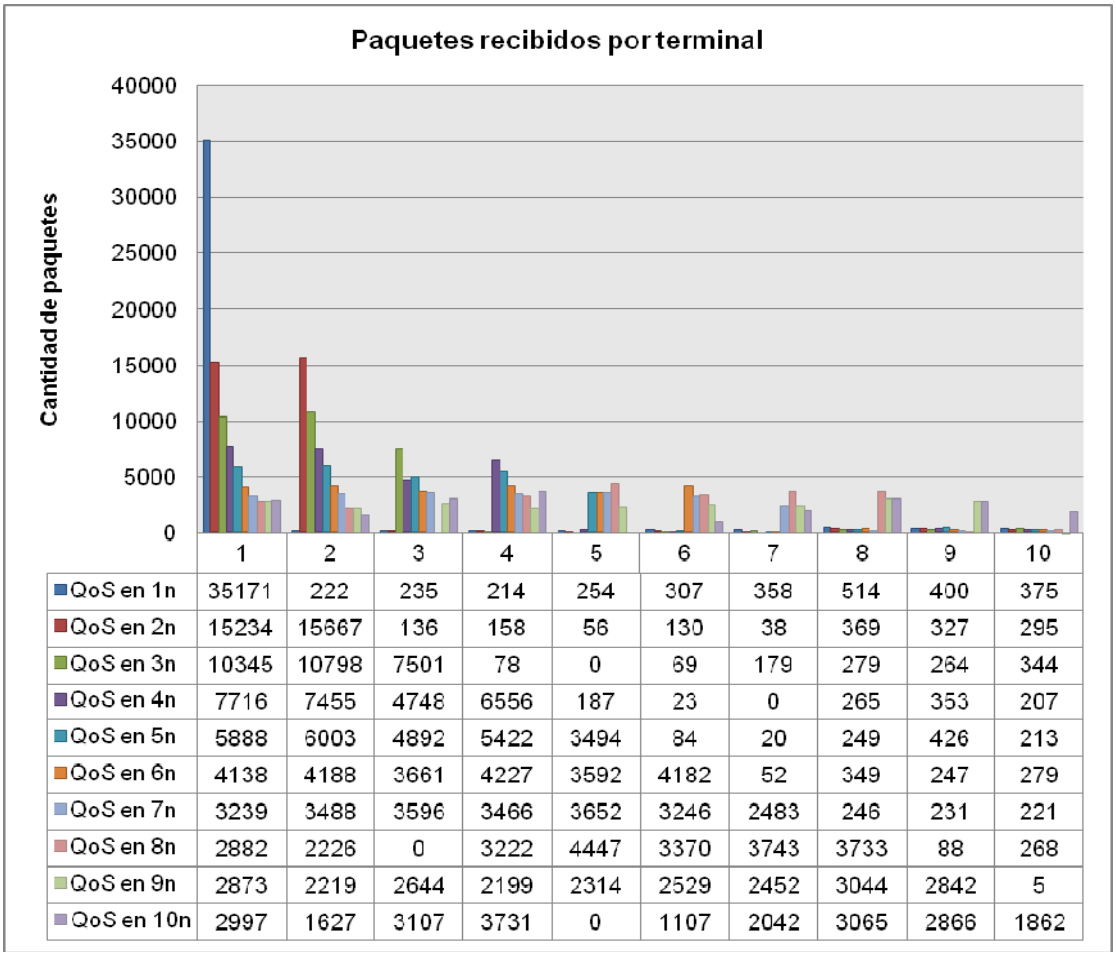
La gráfica 7 muestra la cantidad promedio de paquetes enviados/recibidos por cada terminal. Para obtener estos resultados se corrieron 40 simulaciones en 10 grupos de 4, en cada grupo se varió el tamaño el paquete en 100, 512, 1024 y 1500. También se incrementó la cantidad de nodos a los que se asignó calidad de servicio, así por ejemplo, en las primeras cuatro simulaciones se asignó la calidad al nodo 1, en las siguientes cuatro se asignó calidad de servicio al nodo 1 y al nodo 2 y así sucesivamente, hasta tener 10 nodos con calidad de servicio en las últimas 4 simulaciones.

En el gráfico 7 se observa que cuando se le da prioridad al nodo 1, la calidad de servicio es significativamente alta respecto al resto de los otros nodos. Cuando se incrementa en 2, el número de los nodos a los que se asigna calidad, la cantidad de paquetes que envía cada uno de estos nodos disminuye a la mitad respecto a la simulación en la que sólo se tenía un nodo con QoS, pero ambos nodos siguen teniendo prioridad respecto a los otros ocho.

Entre más sean los nodos a los que se les asigne calidad de servicio, la gráfica tiende a tener menos “picos”, es decir, la cantidad de paquetes que envían se va haciendo más equitativa, esto es normal porque si de los 10 nodos que se tienen en la simulación a los 10 se les asigna la misma prioridad, todos tendrán la misma oportunidad de acceder al medio.



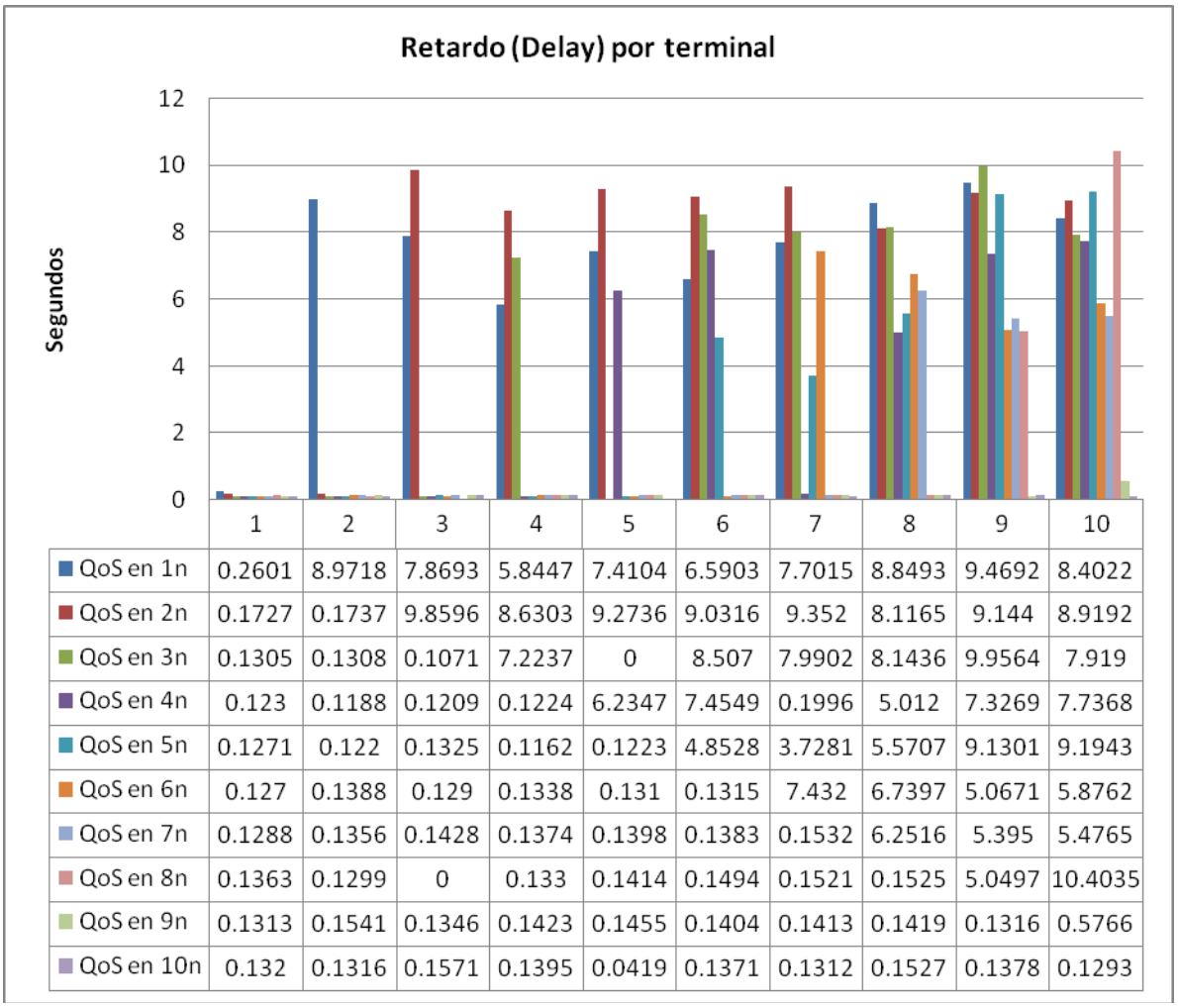

**Gráfica 6**




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| QoS en 1n | 35171 | 222 | 235 | 214 | 254 | 307 | 358 | 514 | 400 | 375 |
| QoS en 2n | 15234 | 15667 | 136 | 158 | 56 | 130 | 38 | 369 | 327 | 295 |
| QoS en 3n | 10345 | 10798 | 7501 | 78 | 0 | 69 | 179 | 279 | 264 | 344 |
| QoS en 4n | 7716 | 7455 | 4748 | 6556 | 187 | 23 | 0 | 265 | 353 | 207 |
| QoS en 5n | 5888 | 6003 | 4892 | 5422 | 3494 | 84 | 20 | 249 | 426 | 213 |
| QoS en 6n | 4138 | 4188 | 3661 | 4227 | 3592 | 4182 | 52 | 349 | 247 | 279 |
| QoS en 7n | 3239 | 3488 | 3596 | 3466 | 3652 | 3246 | 2483 | 246 | 231 | 221 |
| QoS en 8n | 2882 | 2226 | 0 | 3222 | 4447 | 3370 | 3743 | 3733 | 88 | 268 |
| QoS en 9n | 2873 | 2219 | 2644 | 2199 | 2314 | 2529 | 2452 | 3044 | 2842 | 5 |
| QoS en 10n | 2997 | 1627 | 3107 | 3731 | 0 | 1107 | 2042 | 3065 | 2866 | 1862 |

**Gráfica 7.**

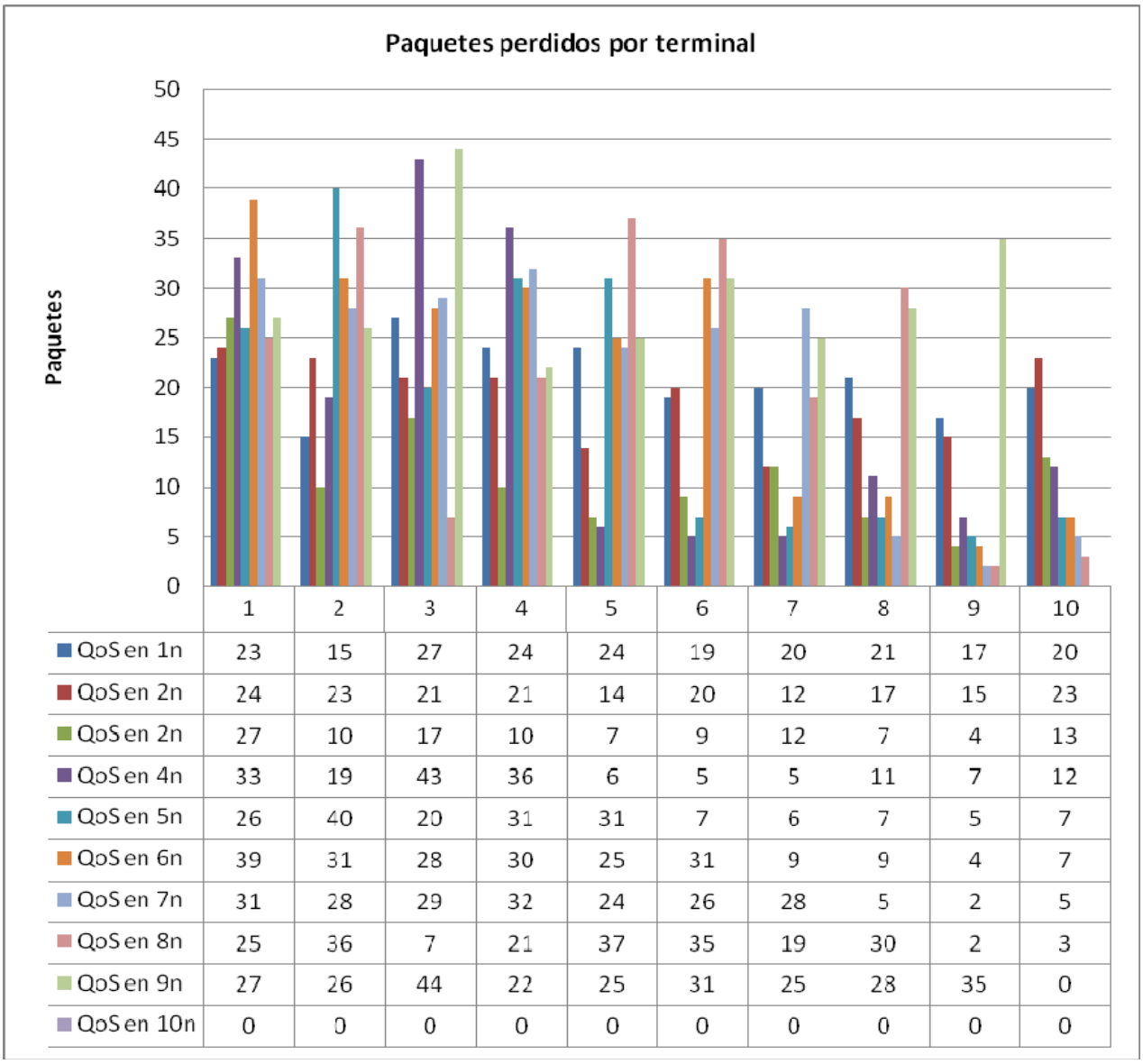
La gráfica 8, muestra el retardo promedio de los paquetes por terminal. Similar al gráfico anterior, cuando la prioridad es asignada sólo al nodo 1, el menor retardo es para el nodo 1. Conforme se va incrementando la cantidad de nodos con prioridad, el retardo se hace menor para cada uno de dichos nodos.




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| QoS en 1n | 0.2601 | 8.9718 | 7.8693 | 5.8447 | 7.4104 | 6.5903 | 7.7015 | 8.8493 | 9.4692 | 8.4022 |
| QoS en 2n | 0.1727 | 0.1737 | 9.8596 | 8.6303 | 9.2736 | 9.0316 | 9.352 | 8.1165 | 9.144 | 8.9192 |
| QoS en 3n | 0.1305 | 0.1308 | 0.1071 | 7.2237 | 0 | 8.507 | 7.9902 | 8.1436 | 9.9564 | 7.919 |
| QoS en 4n | 0.123 | 0.1188 | 0.1209 | 0.1224 | 6.2347 | 7.4549 | 0.1996 | 5.012 | 7.3269 | 7.7368 |
| QoS en 5n | 0.1271 | 0.122 | 0.1325 | 0.1162 | 0.1223 | 4.8528 | 3.7281 | 5.5707 | 9.1301 | 9.1943 |
| QoS en 6n | 0.127 | 0.1388 | 0.129 | 0.1338 | 0.131 | 0.1315 | 7.432 | 6.7397 | 5.0671 | 5.8762 |
| QoS en 7n | 0.1288 | 0.1356 | 0.1428 | 0.1374 | 0.1398 | 0.1383 | 0.1532 | 6.2516 | 5.395 | 5.4765 |
| QoS en 8n | 0.1363 | 0.1299 | 0 | 0.133 | 0.1414 | 0.1494 | 0.1521 | 0.1525 | 5.0497 | 10.4035 |
| QoS en 9n | 0.1313 | 0.1541 | 0.1346 | 0.1423 | 0.1455 | 0.1404 | 0.1413 | 0.1419 | 0.1316 | 0.5766 |
| QoS en 10n | 0.132 | 0.1316 | 0.1571 | 0.1395 | 0.0419 | 0.1371 | 0.1312 | 0.1527 | 0.1378 | 0.1293 |

**Gráfica 8**

La gráfica 9 muestra la cantidad promedio de paquetes perdidos por terminal.




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| QoS en 1n | 23 | 15 | 27 | 24 | 24 | 19 | 20 | 21 | 17 | 20 |
| QoS en 2n | 24 | 23 | 21 | 21 | 14 | 20 | 12 | 17 | 15 | 23 |
| QoS en 2n | 27 | 10 | 17 | 10 | 7 | 9 | 12 | 7 | 4 | 13 |
| QoS en 4n | 33 | 19 | 43 | 36 | 6 | 5 | 5 | 11 | 7 | 12 |
| QoS en 5n | 26 | 40 | 20 | 31 | 31 | 7 | 6 | 7 | 5 | 7 |
| QoS en 6n | 39 | 31 | 28 | 30 | 25 | 31 | 9 | 9 | 4 | 7 |
| QoS en 7n | 31 | 28 | 29 | 32 | 24 | 26 | 28 | 5 | 2 | 5 |
| QoS en 8n | 25 | 36 | 7 | 21 | 37 | 35 | 19 | 30 | 2 | 3 |
| QoS en 9n | 27 | 26 | 44 | 22 | 25 | 31 | 25 | 28 | 35 | 0 |
| QoS en 10n | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Gráfica 9**

### 6.1.4 Con tráfico real de video (*VideoLAN*), generado por las máquinas virtuales y tráfico generado por el simulador. Con 802.11e

La estructura utilizada para estas pruebas se muestra en la figura 6.2.

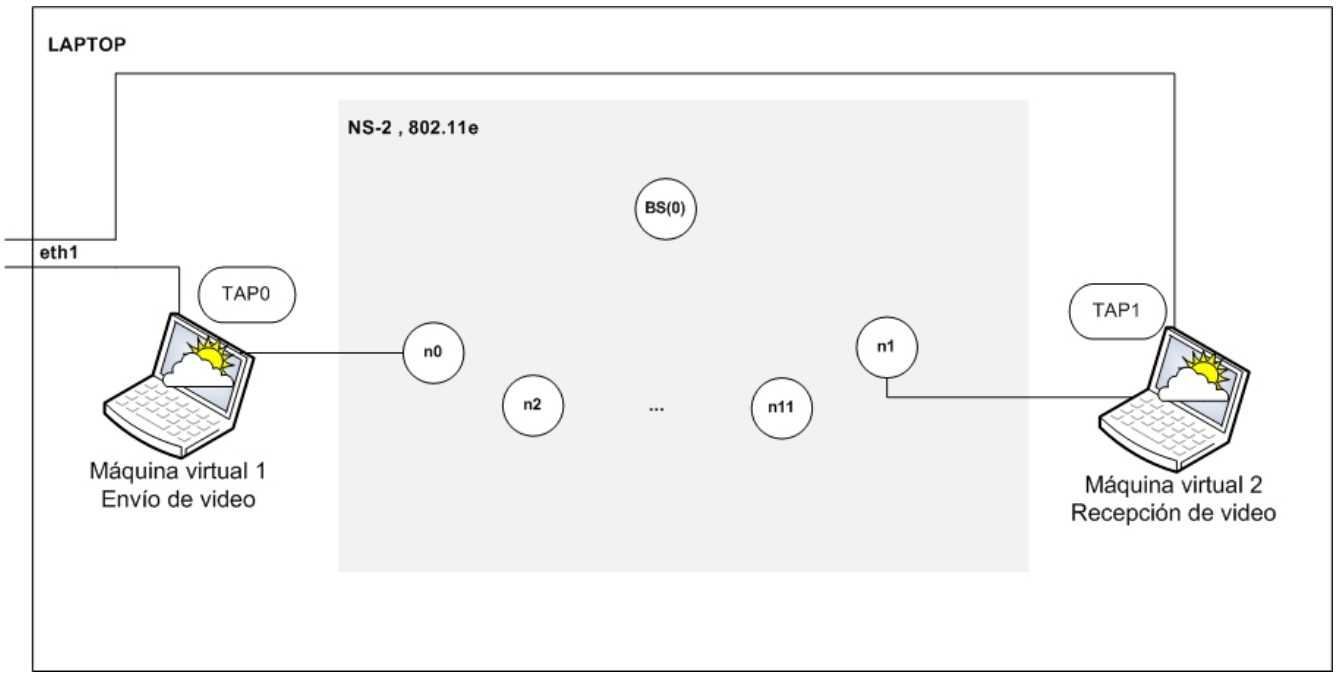



**Figura 6.2 Estructura utilizada**

Se realizaron 2 simulaciones con una duración de 500 segundos, cada una, la configuración utilizada fue:

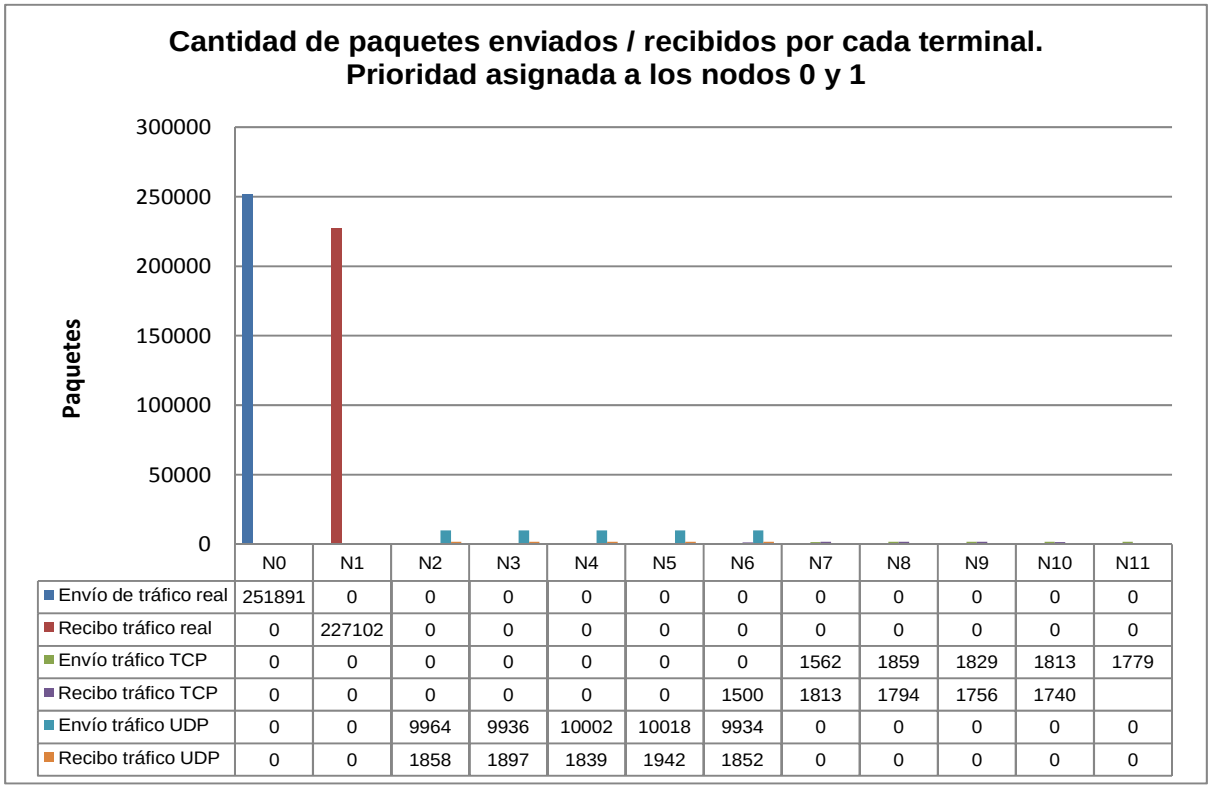
- **Simulación 1**:
  a. Nodo 0 y nodo 1 con calidad de servicio para manejo de video: prio_ 0, class_ 0.
  b. 5 conexiones de tipo TCP-FTP con calidad de servicio prio_ 3, class_ 3. (Nodos 7-11)
  c. 5 conexiones de tipo UDP-CBR con calidad de servicio prio_ 3, class_ 3. (Nodos 2 -6)
- **Simulación 2**:
  a. Nodo 0 y nodo1 con calidad de servicio menor: prio_ 3, class_ 3.
  b. 5 conexiones de tipo TCP-FTP con calidad de servicio superior: prio_ 0, class_ 0. (Nodos 7-11)
  c. 5 conexiones de tipo UDP-CBR con calidad de servicio prio_ 3, class_ 3. (Nodos 2 -6)

Los resultados obtenidos de la simulación 1, son los siguientes:

La tabla 6.2 muestra la cantidad de paquetes enviados por terminal. Como se puede observar la calidad de servicio está asignada al nodo 0. De forma gráfica, estos resultados también se aprecian en la gráfica 10.

**Tabla 6.2 Cantidad de paquetes enviados por terminal**

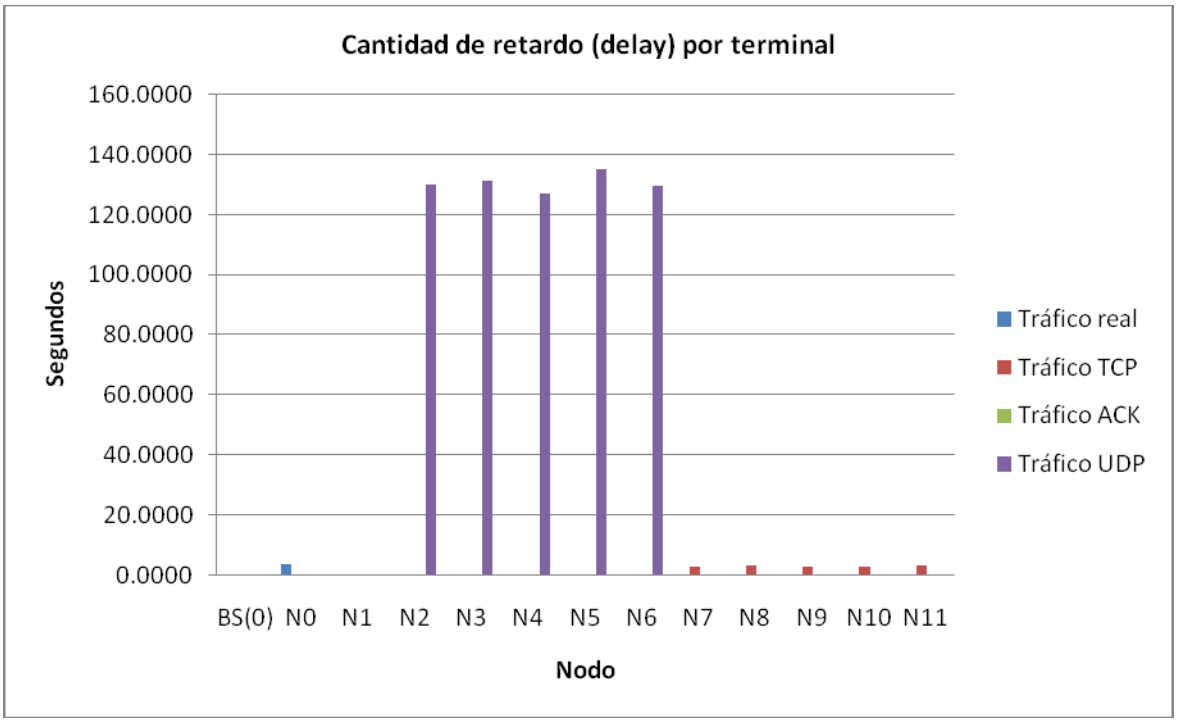
| Tipo de tráfico | Nodo | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BS(0) | N0 | N1 | N2 | N3 | N4 | N5 | N6 | N7 | N8 | N9 | N10 | N11 |
| Tráfico real | 0 | 251891 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Tráfico TCP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1562 | 1859 | 1829 | 1813 | 1779 |
| Tráfico UDP | 0 | 0 | 0 | 9964 | 9936 | 10002 | 10018 | 9934 | 0 | 0 | 0 | 0 | 0 |




| | N0 | N1 | N2 | N3 | N4 | N5 | N6 | N7 | N8 | N9 | N10 | N11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Envío de tráfico real | 251891 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Recibo tráfico real | 0 | 227102 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Envío tráfico TCP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1562 | 1859 | 1829 | 1813 | 1779 |
| Recibo tráfico TCP | 0 | 0 | 0 | 0 | 0 | 0 | 1500 | 1813 | 1794 | 1756 | 1740 | |
| Envío tráfico UDP | 0 | 0 | 9964 | 9936 | 10002 | 10018 | 9934 | 0 | 0 | 0 | 0 | 0 |
| Recibo tráfico UDP | 0 | 0 | 1858 | 1897 | 1839 | 1942 | 1852 | 0 | 0 | 0 | 0 | 0 |

**Gráfica 10**

La gráfica 10 muestra la cantidad de paquetes enviados y recibidos por terminal. Es importante mencionar que el nodo 0 envía paquetes al nodo 1.




**Gráfica 11**

**Tabla 6.3 Cantidad de retardo por terminal**

| Nodo | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **BS(0)** | **N0** | **N1** | **N2** | **N3** | **N4** | **N5** | **N6** | **N7** | **N8** | **N9** | **N10** | **N11** |
| 0 | 3.4975 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.5051 | 3.0302 | 2.6555 | 2.6601 | 2.8965 |
| 0.0064 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 129.9263 | 131.1614 | 127.1151 | 135.3822 | 129.5962 | 0 | 0 | 0 | 0 | 0 |

La gráfica 11 muestra la cantidad de retardo por cada terminal. Se puede obervar que el nodo 0 tiene un retardo similar al de las conexiones TCP (n7 – n11) pero la cantidad de paquetes enviados por el nodo 0 es superior a la de paquetes enviados por las conexiones TCP. La tabla 6.3 muestra el detalle de los datos.

Como se puede apreciar, el nodo 0 tiene mayor cantidad de paquetes enviados, menor cantidad de paquetes perdidos, en proporción al tráfico enviado y menor retardo. Con esto nos aseguramos, que en efecto, se le está asignando calidad de servicio a diferencia de los demás nodos.

Para las pruebas de video, se utilizó un video tipo MPEG con una duración de 30 minutos para enviarlo de una máquina a otra. Uno de los objetivos principales de este trabajo, es estudiar cómo se comporta el tráfico de video al ser enviado por un medio inalámbrico con determinada prioridad. Observamos en la gráfica que el tráfico real tuvo prioridad respecto a los otros tráficos, para medir la calidad del video recibido se creó una escala subjetiva que nos indicará qué tan bien o qué tan mal se ve el video y cuál es la calidad del audio . A continuación se muestra dicha escala.

**Tabla 6.4 Escala de evaluación**

| **Calidad del video y audio recibidos** | **Descripción.** |
|---|---|
| Excelente (E) | Se ve y se escucha excelente. |
| Muy bien (MB) | Se ve y se escucha muy bien; a veces hay un poco de eco. |
| Bien (B) | Se escucha bien y a veces el video se distorsiona un poco. |
| Aceptable (A) | Se entiende el audio aunque el video sufra algunas distorsiones. |
| Mal (M) | Se entiende el audio, en ocasiones hay eco y el video se distorsiona demasiado |
| Muy mal (MM) | No se escucha bien, es decir, a veces no se entiende y el video se distorsiona. |
| Pésimo (P) | No se entiende nada y hay partes del video que no se ven. |

Con base en esta escala se evalúo el video recibido en la simulación 1, resultando **Muy Bien** y en ocasiones **Bien** pues el video se distorsiona un poco.

En la simulación 2, la calidad de servicio fue asignada a los nodos que transmiten tráfico TCP-FTP y al nodo que transmite tráfico real de video-audio se le asignó una calidad de servicio menor. Los resultados fueron los siguientes:

En la gráfica 12 se ilustra que, al quitar la calidad de servicio al nodo 0, la cantidad de paquetes enviados disminuyó enormemente. Aquí quien envía más paquetes es el punto de acceso (BS, *Base Station*), el tipo de tráfico es ACK y son las confirmaciones a los paquetes TCP recibidos. La tabla 6.5 muestra el detalle de los números.

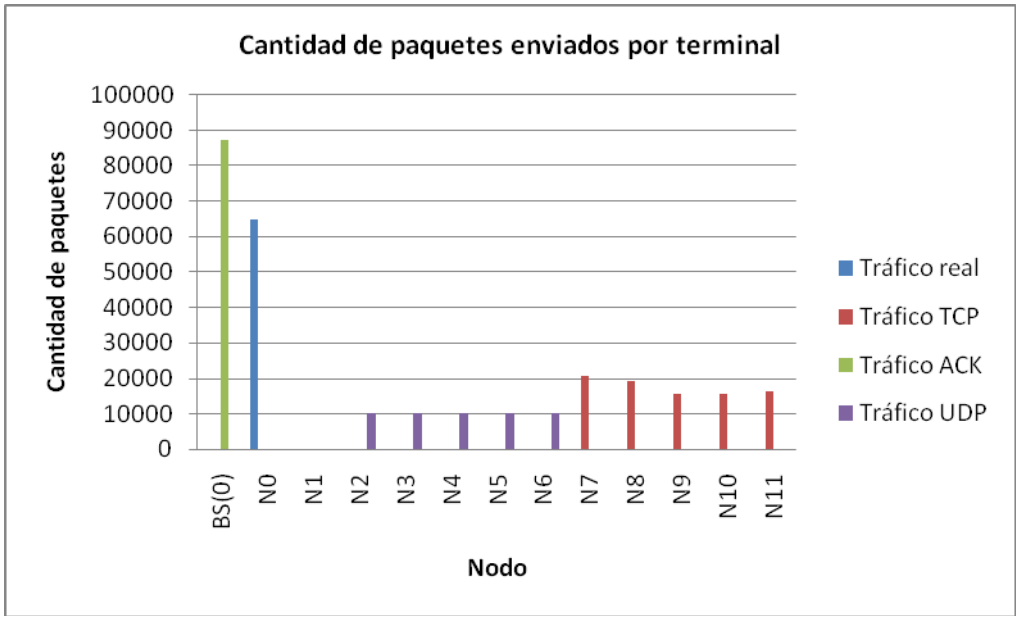
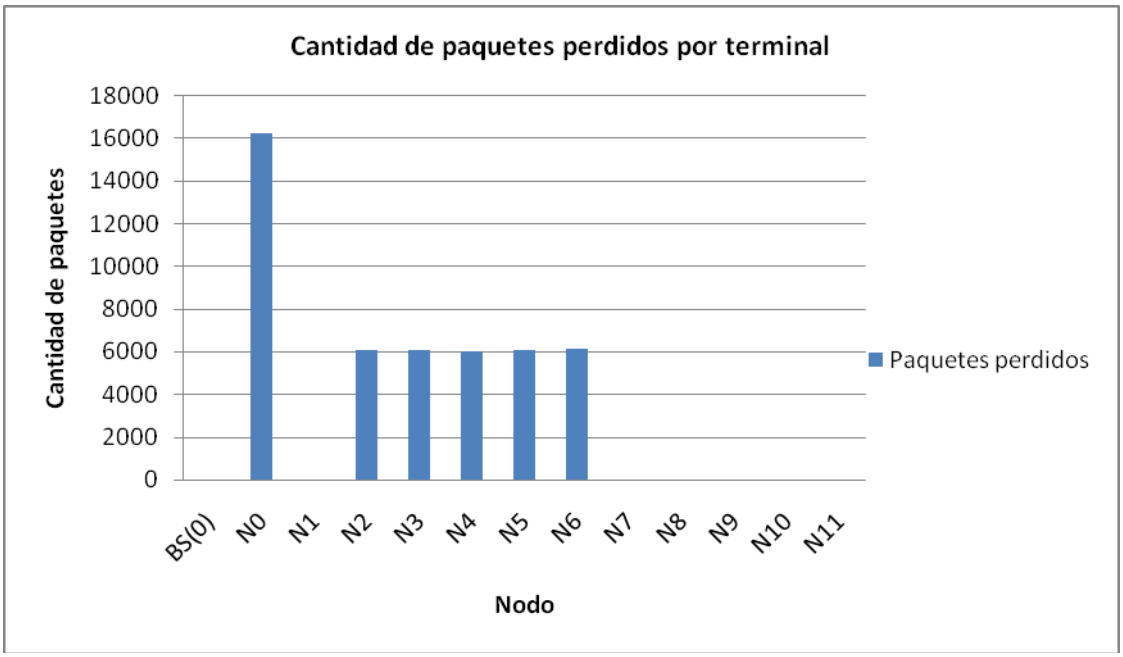



**Gráfico 12**

**Tabla 6.5 Cantidad de paquetes enviados por terminal**

| Nodo | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **BS(0)** | **N0** | **N1** | **N2** | **N3** | **N4** | **N5** | **N6** | **N7** | **N8** | **N9** | **N10** | **N11** |
| 0 | 65000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20789 | 19095 | 15731 | 15679 | 16245 |
| 87280 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 9990 | 10034 | 10034 | 10042 | 10008 | 0 | 0 | 0 | 0 | 0 |



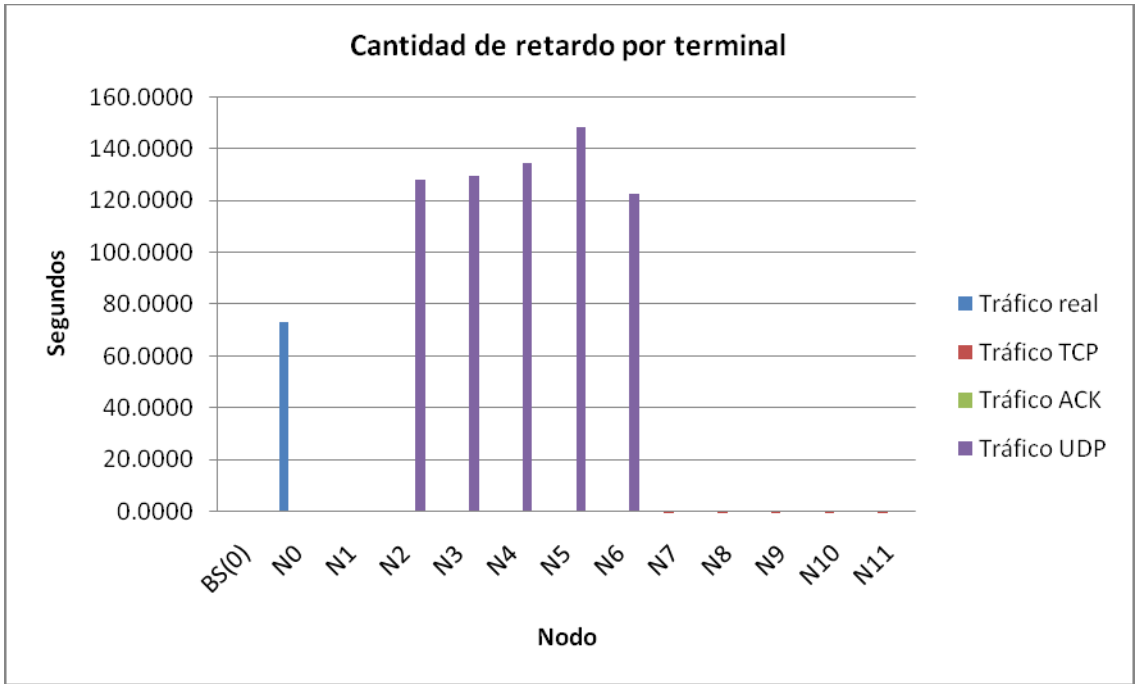

**Gráfica 13**

**Tabla 6.6 Cantidad de paquetes perdidos por terminal**

| Nodo | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **BS(0)** | **N0** | **N1** | **N2** | **N3** | **N4** | **N5** | **N6** | **N7** | **N8** | **N9** | **N10** | **N11** |
| 49 | 16258 | 0 | 6102 | 6105 | 6049 | 6122 | 6149 | 24 | 32 | 26 | 31 | 34 |

La gráfica 13 también muestra la cantidad de paquetes recibidos por terminal, como puede observarse, el nodo que pierde más paquetes es el nodo 0, el cual, en esta simulación, ya no tiene calidad de servicio. Los nodos que menos pierden paquetes son los nodos que enviaron tráfico TCP-FTP (nodo 7 al nodo 11) y fue a los nodos
a los que se les asignó calidad de servicio.

La gráfica 14 muestra el retardo por terminal, aquí los nodos más afectados son los que transmiten tráfico UDP-CBR y el nodo 0, ninguno de estos nodos tiene asignada calidad de servicio. La tabla 6.7 muestra los números.




**Gráfica 14**

| Cantidad de retardo (delay) por terminal | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Nodo | | | | | | | | | | | | |
| **BS(0)** | **N0** | **N1** | **N2** | **N3** | **N4** | **N5** | **N6** | **N7** | **N8** | **N9** | **N10** | **N11** |
| 0.0000 | 73.0651 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0303 | 0.0312 | 0.0246 | 0.0261 | 0.0254 |
| 0.4108 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 0.0000 | 0.0000 | 0.0000 | 127.9975 | 129.3827 | 134.2367 | 148.3571 | 122.4183 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

**Tabla 6.7**

De lo anterior se resume que en la simulación 2, al quitar la calidad de servicio al nodo 0, disminuyó la cantidad de paquetes enviados por éste, se incrementó la cantidad de paquetes perdidos y también se incrementó el retardo.

Con base en la escala de evaluación mostrada en la tabla 6.4, el video enviado desde el nodo 0 hacia el nodo 1 se evalúo entre **Mal** y **Muy Mal**, esto debido a que el audio se entrecorta mucho, ocasionando que

ciertas partes no se entiendan; el video se distorsiona demasiado y en ocasiones pareciera que está en cámara lenta.

Con esto, podemos concluir que se cumple la misión del estándar 802.11e, la cual consiste en garantizar la calidad de servicio de las transmisiones, a las que se asigne prioridad. Mediante la serie resultados explicados anteriormente, queda demostrado que el estándar 802.11e sí ofrece una mejora en la calidad del servicio, respecto al original 802.11

Es importante mencionar que estos fueron resultados obtenidos a través del simulador, el cual tiene una serie de limitaciones, como que no es posible configurar parámetros de acceso individual para cada dispositivo; el cambio de configuración no es dinámico; sólo está implementada la función EDCA y no la función HCCA, entre otros. Cada uno de estos aspectos nos abre una vía para un nuevo entorno de trabajo.

### 6.1.5 Con tráfico real de voz (*Skype*), generado por las máquinas virtuales y tráfico generado por el simulador. Con 802.11e.

La estructura utilizada para estas pruebas es la misma de la figura 6.2.

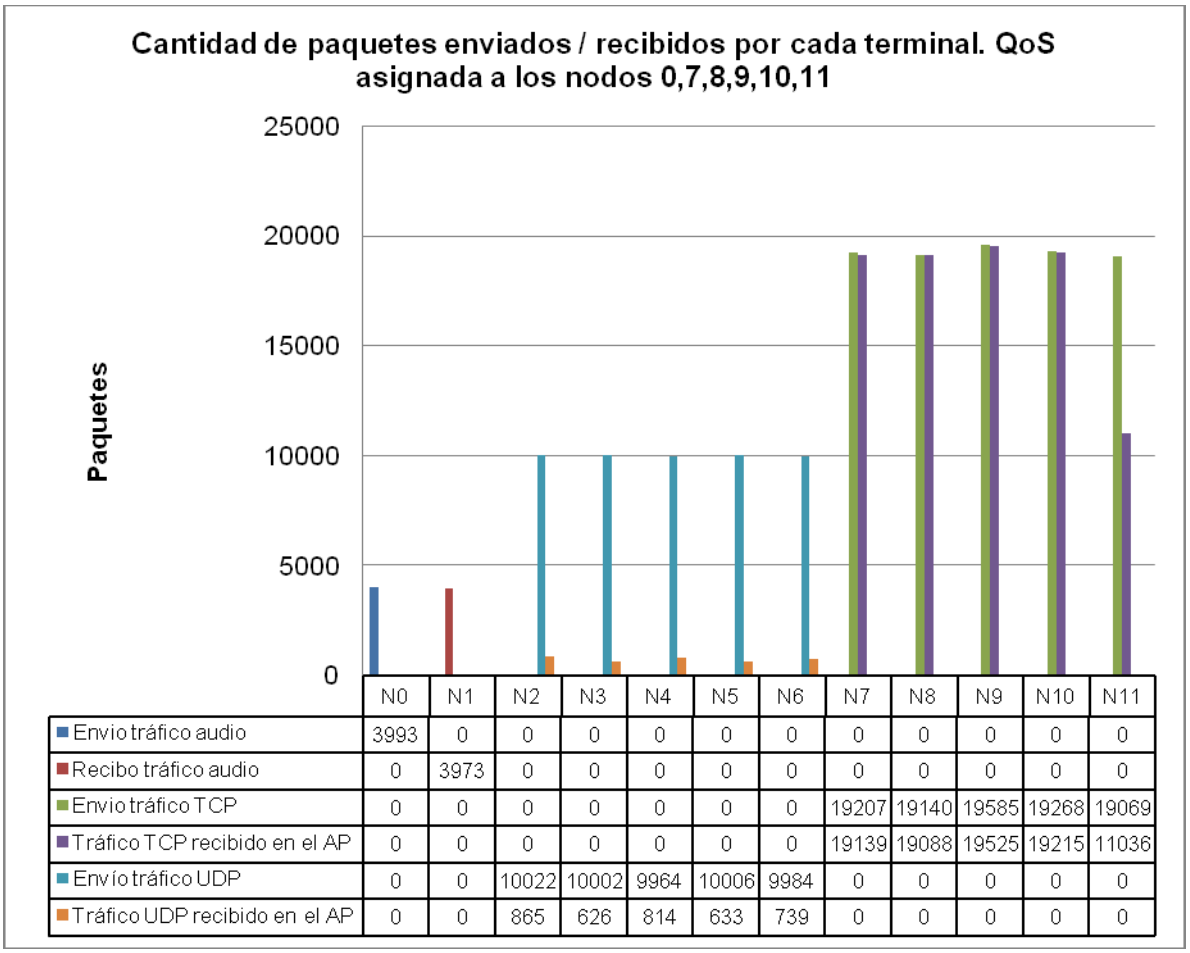
Se realizaron 3 simulaciones con una duración de 500 segundos, cada una, la configuración utilizada fue:

- **Simulación 1**:
  a. Calidad de servicio asignada a los nodos que transmiten voz a través de Skype, prio_0 y class_0 (Nodo 0 y 1).
  b. 5 conexiones de tipo TCP-FTP con calidad de servicio prio_ 0, class 0. (Nodos 7-11).
  c. 5 conexiones de tipo UDP-CBR con calidad de servicio prio_ 3, class_ 3. (Nodos 2 -6).

- **Simulación 2**:
  d. Calidad de servicio prio_ 3 y class 3 asignada los nodos que transmiten voz a través de Skype, Nodo 0 y 1).
  e. 5 conexiones de tipo TCP-FTP con calidad de servicio prio_ 0, class 0. (Nodos 7-11).
  f. 5 conexiones de tipo UDP-CBR con calidad de servicio prio_ 3, class_ 3. (Nodos 2 -6).

- **Simulación 3**:
  g. Calidad de servicio prio_ 0 y class 0 asignada los nodos que transmiten voz a través de Skype, Nodo 0 y 1).
  h. 5 conexiones de tipo TCP-FTP con calidad de servicio prio_ 0, class 0. (Nodos 7-11).
  i. 5 conexiones de tipo UDP-CBR con calidad de servicio prio_ 0, class_ 0. (Nodos 2 -6)

Los resultados obtenidos de la simulación 1, son los siguientes:



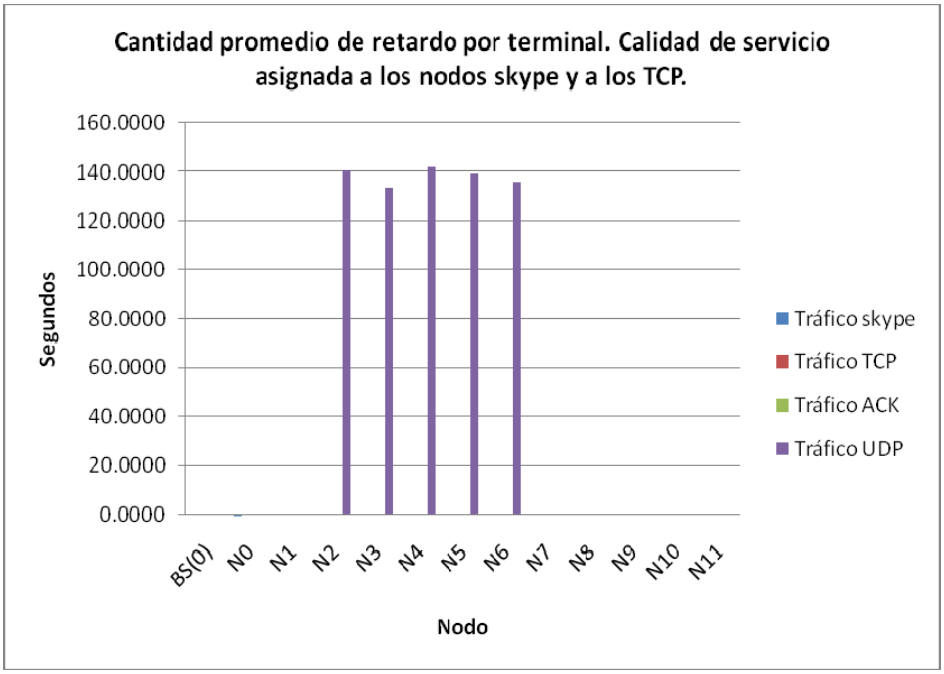

| | N0 | N1 | N2 | N3 | N4 | N5 | N6 | N7 | N8 | N9 | N10 | N11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Envio tráfico audio | 3993 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Recibo tráfico audio | 0 | 3973 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Envio tráfico TCP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 19207 | 19140 | 19585 | 19268 | 19069 |
| Tráfico TCP recibido en el AP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 19139 | 19088 | 19525 | 19215 | 11036 |
| Envío tráfico UDP | 0 | 0 | 10022 | 10002 | 9964 | 10006 | 9984 | 0 | 0 | 0 | 0 | 0 |
| Tráfico UDP recibido en el AP | 0 | 0 | 865 | 626 | 814 | 633 | 739 | 0 | 0 | 0 | 0 | 0 |

**Gráfica 15**

La gráfica 15 muestra la cantidad de paquetes enviados y recibidos por terminal. Como se puede observar, la calidad de servicio está asignada al nodo 0 y a las conexiones de TCP. Es importante mencionar que, la cantidad de paquetes que envia el nodo 0 (N0) es menor a los que envía cualquier nodo tipo TCP (N7 – N11) porque recordemos que el nodo 0 está transmitiendo tráfico de voz.

En la gráfica 15 también se muestra la cantidad de paquetes recibidos por cada terminal. Podemos apreciar que, la calidad asignada al nodo que transmite audio, en este ejemplo, el nodo 0 envía 3993 paquetes y el receptor, el nodo 1, recibe 3973. Si consideramos la escala de evaluación, mostrada en la tabla 6.5, se puede decir que la calidad con que se recibió el tráfico de audio fue muy aceptable, es decir, se escucha muy bien aunque en algunas ocasiones se percibió un poco de eco.

La gráfica 16 y la tabla 6.8 muestran la cantidad promedio de retardo por cada terminal. Se aprecia claramente que el mayor retardo lo tienen las transmisiones a las que se asigna una calidad de servicio menor, que en este caso es a los nodos con tráfico UDP.



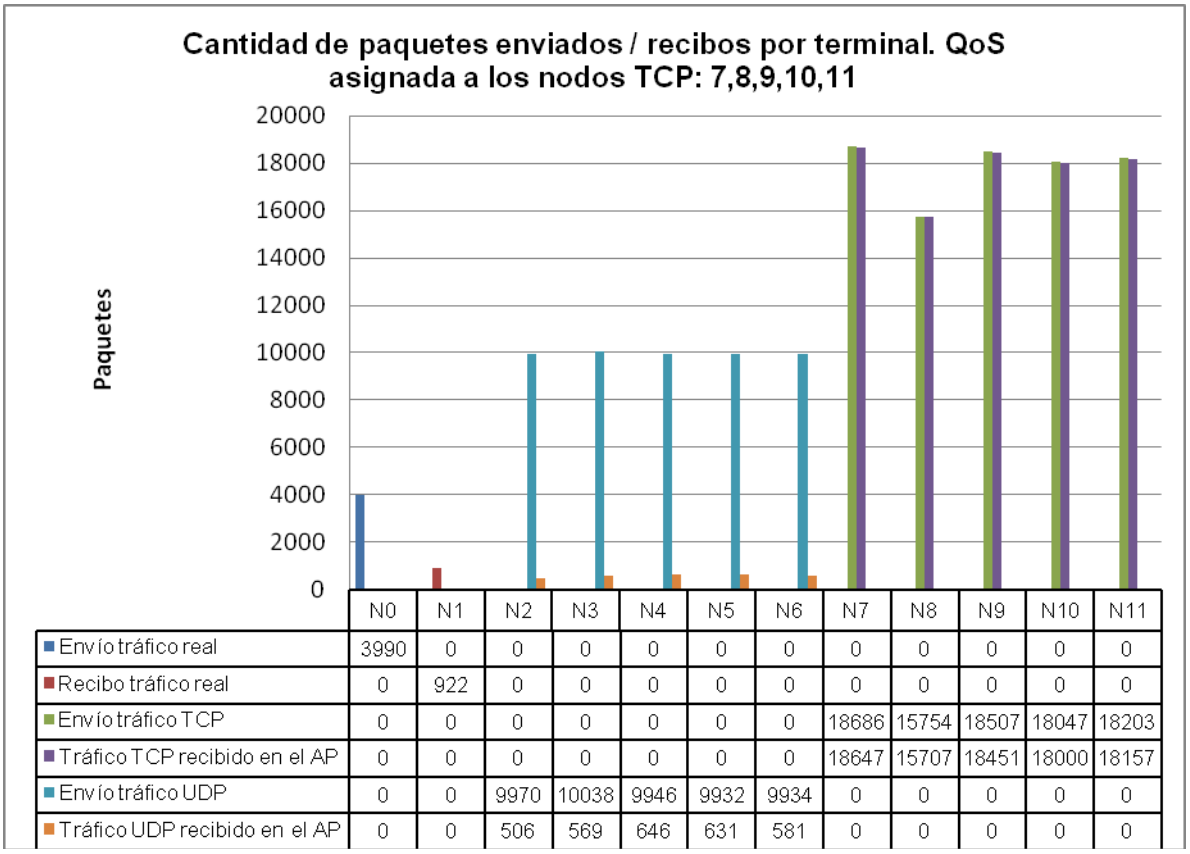

**Gráfica 16**

**Tabla 6.8 Cantidad promedio de retardo por terminal**

| Nodo | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **BS(0)** | **N0** | **N1** | **N2** | **N3** | **N4** | **N5** | **N6** | **N7** | **N8** | **N9** | **N10** | **N11** |
| 0.0000 | 0.0109 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0265 | 0.0265 | 0.0273 | 0.0268 | 0.0263 |
| 0.3854 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 0.0000 | 0.0000 | 0.0000 | 140.8266 | 133.4819 | 142.0114 | 139.1554 | 136.1600 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

En la simulación 2, los resultados fueron los siguientes:

La gráfica 17 muestra la cantidad de paquetes enviados y recibidos por cada terminal, la calidad de servicio fue asignada sólo a los nodos con tráfico TCP. Se puede apreciar que el nodo 0, con tráfico de audio, envía 3990 paquetes, cifra similar a la enviada en la anterior simulación, la diferencia se aprecia en la misma gráfica, pues el nodo 1, que es el nodo receptor, sólo recibe 922 paquetes; además de que el retardo es muy alto. Con base en la escala de evaluación, el tráfico de audio recibido fue **Muy Mal**, el audio se entendió por periodos cortos, hubo mucho eco y el retraso fue muy alto.



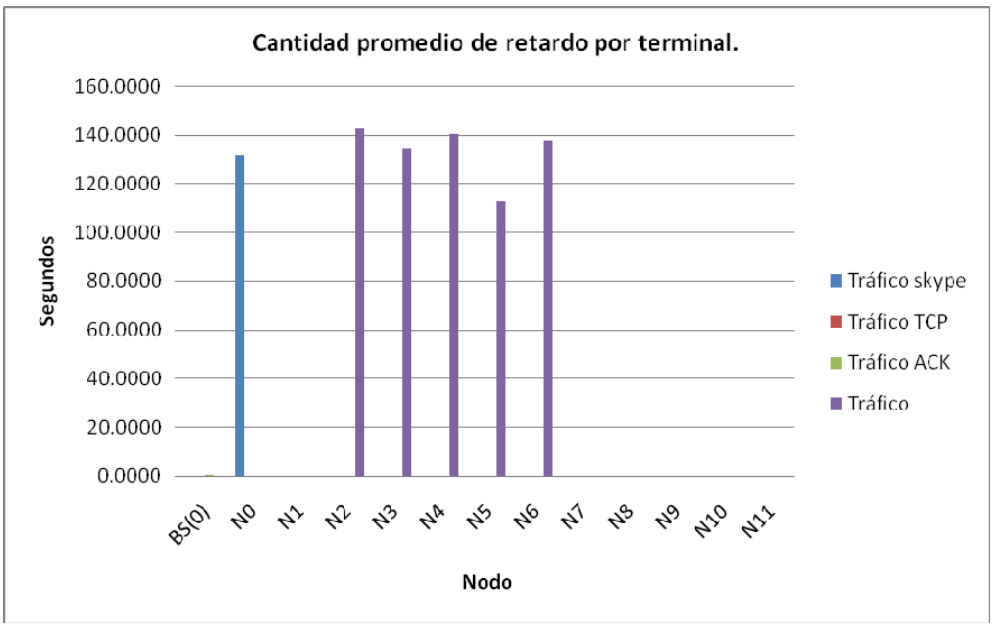

| | N0 | N1 | N2 | N3 | N4 | N5 | N6 | N7 | N8 | N9 | N10 | N11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Envío tráfico real | 3990 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Recibo tráfico real | 0 | 922 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Envío tráfico TCP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18686 | 15754 | 18507 | 18047 | 18203 |
| Tráfico TCP recibido en el AP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18647 | 15707 | 18451 | 18000 | 18157 |
| Envío tráfico UDP | 0 | 0 | 9970 | 10038 | 9946 | 9932 | 9934 | 0 | 0 | 0 | 0 | 0 |
| Tráfico UDP recibido en el AP | 0 | 0 | 506 | 569 | 646 | 631 | 581 | 0 | 0 | 0 | 0 | 0 |

**Gráfica 17**

La gráfica 18 y la tabla 6.9 muestran la cantidad promedio de retardo por cada terminal.

**Tabla 6.9 Cantidad promedio de retardo por terminal**

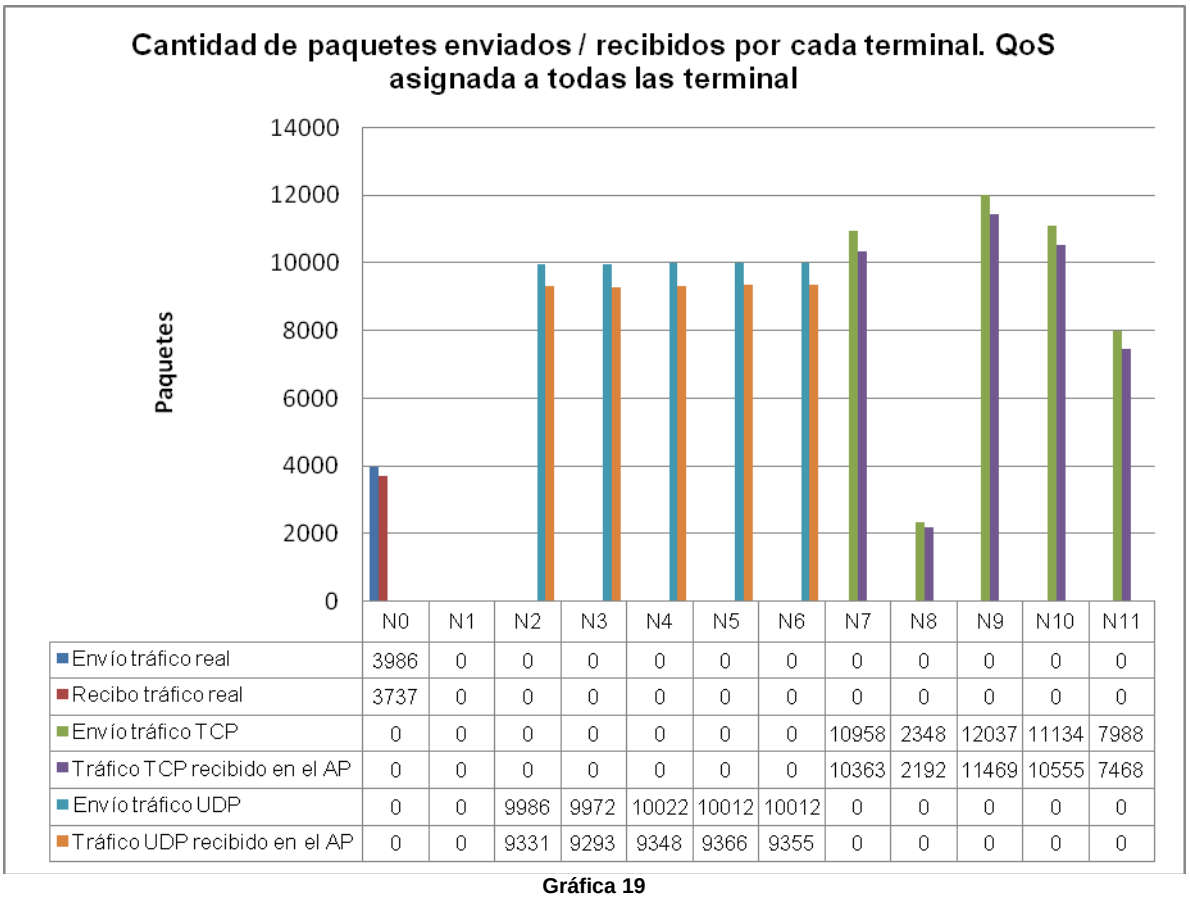
| Nodo | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **BS(0)** | **N0** | **N1** | **N2** | **N3** | **N4** | **N5** | **N6** | **N7** | **N8** | **N9** | **N10** | **N11** |
| 0.0000 | 131.3378 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0245 | 0.0250 | 0.0249 | 0.0244 | 0.0250 |
| 0.4183 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 0.0000 | 0.0000 | 0.0000 | 142.6131 | 134.3615 | 140.3210 | 112.4865 | 137.8540 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |




**Gráfica 18**

Los resultados de la simulación 3 se muestran a continuación:

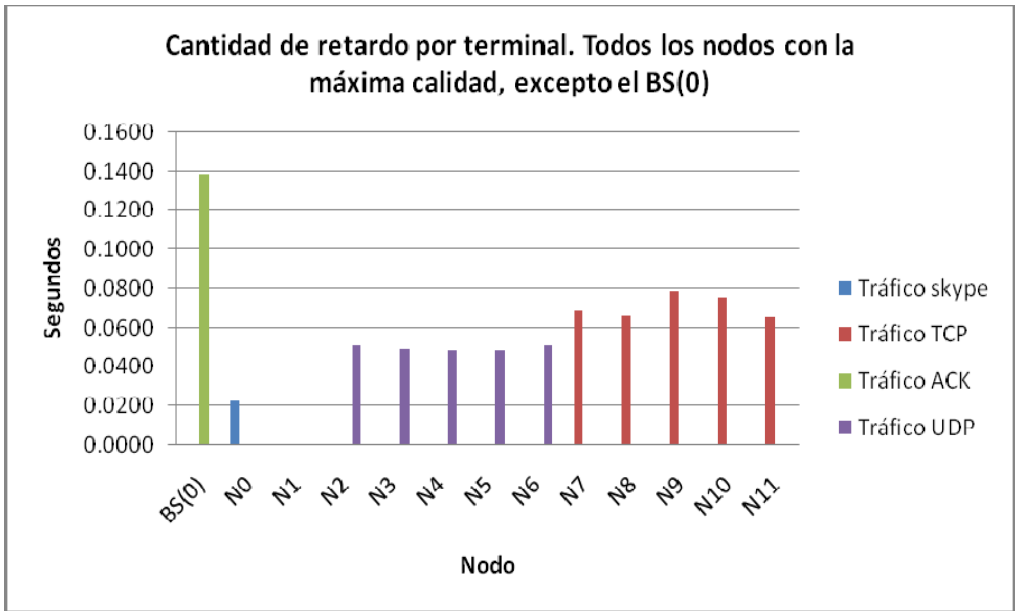
La gráfica 19 muestra la cantidad de paquetes enviados y recibidos por terminal; en esta simulación la calidad de servicio fue asignada a todos los nodos, por ello las cantidades se equilibran.




| | N0 | N1 | N2 | N3 | N4 | N5 | N6 | N7 | N8 | N9 | N10 | N11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Envío tráfico real | 3986 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Recibo tráfico real | 3737 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Envío tráfico TCP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10958 | 2348 | 12037 | 11134 | 7988 |
| Tráfico TCP recibido en el AP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10363 | 2192 | 11469 | 10555 | 7468 |
| Envío tráfico UDP | 0 | 0 | 9986 | 9972 | 10022 | 10012 | 10012 | 0 | 0 | 0 | 0 | 0 |
| Tráfico UDP recibido en el AP | 0 | 0 | 9331 | 9293 | 9348 | 9366 | 9355 | 0 | 0 | 0 | 0 | 0 |

**Gráfica 19**

La gráfica 20 y la tabla 6.10 muestran la cantidad de retardo por cada terminal.




**Gráfica 20**

**Tabla 6.10 Cantidad promedio de retardo por terminal**

| Nodo | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **BS(0)** | **N0** | **N1** | **N2** | **N3** | **N4** | **N5** | **N6** | **N7** | **N8** | **N9** | **N10** | **N11** |
| 0.0000 | 0.0230 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0685 | 0.0659 | 0.0789 | 0.0753 | 0.0652 |
| 0.1384 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 0.0000 | 0.0000 | 0.0000 | 0.0512 | 0.0492 | 0.0484 | 0.0484 | 0.0509 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

En esta simulación el tráfico de audio se recibió de manera aceptable, es decir, se escuchó entre **Bien** y **Muy bien**, hubo poco eco y poco retardo.

# Capítulo 7 Conclusiones

## 7.1 Conclusiones

Una de las contribuciones importantes de este trabajo, es que se ha llevado a cabo una revisión de las tecnologías de redes inalámbricas, realizando una valoración cuantitativa y cualitativa de su funcionamiento. Se realizó un análisis exhaustivo del recién liberado estándar IEEE 802.11e, el cual ofrece la posibilidad de soportar cierto nivel de calidad de servicio en las redes inalámbricas. Como se mencionó, este estándar sugiere algunos cambios en la capa de enlace, en específico en la capa MAC.

De igual forma, en esta tesis se revisaron las principales propuestas de calidad de servicio a nivel de red, mencionando de forma resumida las técnicas de Servicios Diferenciados (*DiffServ*) y Servicios Integrados (*Intserv*), pues de éstas se han tomado algunas ideas para implementar la calidad de servicio en las redes inalámbricas, las cuales son especialmente sensibles a las condiciones del entorno y altamente ineficientes en determinados escenarios.

Para la realización de los experimentos y obtención de resultados, fue necesario elegir un entorno de simulación adecuado, que ofreciera la posibilidad de modelar estaciones inalámbricas 802.11e. En este sentido, se eligió el simulador de redes NS-2 y una extensión desarrollada por la Universidad Politécnica de Berlín.

Se llevaron a cabo una serie de simulaciones con distintos tipos de tráfico, incluyendo transmisión de tráfico real de video y de voz. Con los resultados obtenidos, se demostró que hay un incremento importante en la calidad de servicio a los nodos que se les proporciona prioridad, esto sin afectar demasiado a los nodos que no la tienen. Para ello se instaló y configuró el NS-2 con soporte para IEEE 802.11e. Además, con el objetivo de usar escenarios reales y casos prácticos, como el envío de video y de audio, se utilizó la técnica de las máquinas virtuales para la generación de tráfico real y creación de escenarios de interés. El escenario con el que se trabajó se muestra en el capítulo 6.

Como se mencionó, la calidad de servicio es lograda a través de la asignación de prioridades, a los nodos, para transmitir; cuando a los nodos que transmiten video se les asigna una prioridad alta, la calidad con la que se recibe el audio y el video se ve incrementada de forma considerable respecto a cuando no se asigna calidad de servicio alguna; es decir, si se le asigna calidad de servicio (prio_ 0, class 0), se recibe el video con **Muy Buena** calidad, es decir, el audio se escucha muy bien y el video también se ve muy bien. Cuando no se asigna calidad de servicio y el medio es compartido con otros nodos que transmiten tráfico tipo TCP, tanto la calidad del video y del audio recibido es **Muy Mala**, se incrementa la pérdida de paquetes y hay mucho eco en el audio. Algo similar ocurre cuando se transmite audio; si se

asigna calidad de servicio (prio_ 0, class 0), la calidad con que se recibe es muy aceptable, si se retira la prioridad de servicio, la calidad es **Muy Mala** y en general no se entiende.

Los resultados de las simulaciones han demostrado como el estándar IEEE 802.11e, ofrece una mejora significativa en el manejo de la calidad de servicio para transmisiones que son muy susceptibles frente a los retardos y perdidas de los paquetes.

## 1.2 Trabajos futuros

El desarrollo del presente trabajo puede continuar hacia diferentes líneas de investigación, como también se mencionó, en el NS-2 sólo esta implementada la función EDCA, pero no HCCA; por lo que resultará necesario llevar a cabo la implementación de ésta. La versión de NS con la que se trabajó, es la NS-2.28, al finalizar este trabajo, ya existe la versión NS-2.31, entonces resulta necesario también, trabajar en la actualización de los parches para poder trabajar IEEE 802.11e con la última versión.

Por otro lado, ha salido el estándar IEEE 802.11n, el cual permite lograr una velocidad de transmisión de hasta 100 Mbps o más, por lo que resultaría por demás interesante, investigar cómo se implementa la calidad de servicio en 802.11n.

Si bien es cierto que, el simulador NS-2 ayudó mucho para la obtención de resultados, también es cierto que tiene muchas limitaciones que pueden ser mejoradas con el trabajo académico.

# Referencias


[CHOI1] S. Choi. *Emerging IEEE 802.11e WLAN for Quality of Service(QoS) Provisionin.* SK Telecom Telecommun. Rev., vol. 12, no. 6, Dec 2002. pp. 894-906

[CHOI2] S. Choi, J. Prado, S. Shankar, S. Mangold. *IEEE 802.11e contention based channel access (EDCF) performance evaluation*. IEEE International Conference on Communications, 2003, Vol 2, pages 1151-1156

[CRANLEY1] Nicola Cranley, Mark Davis*. Performance Evaluation of Video Streaming with Background Traffic over IEEE 802.11 WLAN Networks*. Canada, 2005.

[CRANLEY2] Nicola Cranley, Mark Davis*. Performance Analysis of Network-level QoS with Encoding Configurations for Unicast Video Streaming over IEEE 802.11WLAN Networks*. Canada, 2005.

[GAST] Matthew S. Gast. *802.11 Wireless Networks: the Definitive Guide*. O'Really & Associates, USA, 2nd edition, 2002

[GREIS] M. Greis. *Tutorial for the network simulator ns*. http://www.isi.edu/nsnam/ns/tutorial/index.html

[HUIDOBRO] José M. Huidobro, David Roldán. Comunicaciones en redes WLAN: Wifi, VoIP, Multimedia, Seguridad. Ed. Limusa, México 2006.

[KUANG] Tianbo Kuang, Carey Williamson. *RealMedia Streaming Performance on an IEEE 802.11 Wireless LAN*. Department of Computer Science, University of Calgary, Canada 2004.

[MANGOLD] Stefan Mangold, Sunghyun Choi, Guido R. Hiertz, Bernhard Walke. *Analysis of IEEE 802.11e for QoS support in Wireless LANs*. IEEE 2003.

[NETPERF] http://www.netperf.org

[TAO] Zhifeng Tao, Shivendra Panwar. *Throughput and Delay Analysis for the IEEE 802.11e Enhanced Distibuted Channel Access*. IEEE, Brooklyn, NY, 2006.

[UML] UML User Guide. http://user-mode-linux.sourceforge.net

[VIDEOLAN] http://www.videolan.org

[VINT] VINT Project, Kevin Fall, Kannan Varadhan. *The NS Manual (formerly NS notes and Documentation)*. US Berkeley, Xerox PARC, April 2008.

[WALL] Larry Wall, Randal L. Schwartz, Tom Christiansen and Stephen Potter. *Programming Perl*. Nutshell Handbook, O'Really & Associates, USA, 2nd edition, 1996

[WANG] Yubing Wang, Mark Claypool,Sheng Zuo, "*An Empirical Study of RealVideo Performance Across the Internet*", San Francisco, USA 2001.

[WIETHOLTER1] S. Wietholter, C. Hoene, "*Design and Verification of an IEEE 802.11e EDCF Simulation Model in ns-2.26*", Berlin, November 2003, TKN Technical Report TKN03-19, http://www.tkn.tu-berlin.de/research/802.11e_ns2/techreport.pdf

[WIETHOLTER2] S. Wietholter, C. Hoene, A. Wolisz, "*PerceptualQuality of Internet Telephony over 802.11e Supporting Enhanced DCF and Contention Free Bursting*", Berlin, September 2004, TKN Technical Report, TKN-04-11, http://www.tkn.tuberlin.de/research/802.11e_ns2/techreport2.pdf

[ZHU] Hua Zhu, Ming Li, Imrich Chlamtac, and B. Prabharakan. *A survey of Quality of Service in IEEE 802.11 Networks.* China 2001.