EVALUACIÓN DEL DESEMPEÑO DE REDES 802.11P/WAVE EN LA TRANSMISIÓN DE DATOS, VOZ Y VIDEO IP

ANEXOS

ANTONI GABRIEL CAICEDO BASTIDAS
JUAN MANUEL MARTÍNEZ OJEDA

UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
Departamento de Telecomunicaciones
Grupo I+D Nuevas Tecnologías En Telecomunicaciones – GNTT
Línea de Investigación: Gestión Integrada de Redes, Servicios y Arquitecturas de Telecomunicaciones
POPAYÁN
2011

ANEXO A

CONCEPTOS

A.1 CSMA/CA

El protocolo de acceso múltiple por detección de portadora con prevención de colisiones, CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*), permite a una estación censar un canal de comunicaciones con el fin de detectar si este se encuentra ocupado, debido a que se está efectuando alguna transmisión, o si está libre; y si lo encuentra ocupado, espera un intervalo de tiempo denominado intervalo de backoff y procede a testear de nuevo, para intentar el envío de un paquete que ha llegado a la capa MAC de la estación en el momento en que el canal este libre [1]. Con este mecanismo se pretende evitar las colisiones antes de que sucedan; porque en una red inalámbrica con topología ad hoc resulta extremadamente difícil detectar las colisiones en el medio, como lo es el aire; además, prevenirlas constituye un proceso más eficiente.

Cuando una estación desea enviar una trama, primero debe censar el medio. Si el canal está inactivo durante un intervalo de tiempo igual o mayor que un espacio entre tramas (DIFS, *Distributed Interframe Space*), entonces la estación puede enviar la trama; en caso contrario la estación permanece a la espera de poder transmitir sondeando el medio. En el momento en que dicha estación percibe que el medio vuelve a estar inactivo, espera un DIFS más; si el medio se ocupa durante este intervalo de tiempo, la estación vuelve a quedar a la espera de que el medio se desocupe sondeándolo; sin embargo, si el medio continúa inactivo, la estación pasa a esperar un tiempo aleatorio llamado tiempo de backoff [2]. Un ejemplo de este proceso se puede observar en la figura A-1 [3].

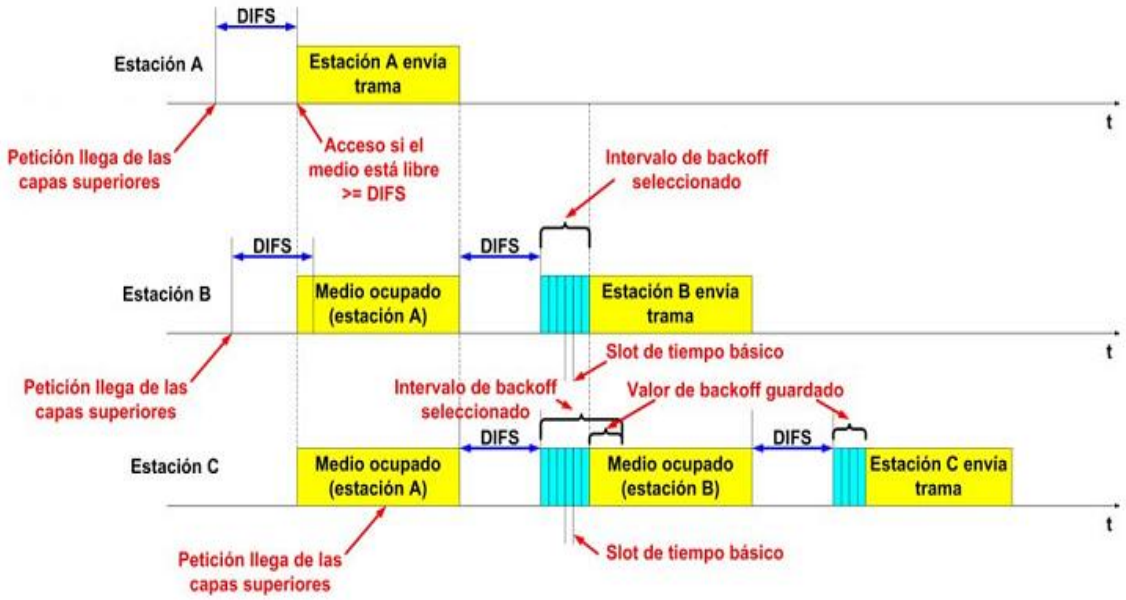


Figura A-1 Proceso de sondeo en CSMA/CA.

El intervalo de tiempo de backoff se calcula de la forma siguiente:

$$T_{backoff} = f(0,CW) \times T_{slot}$$

Donde T_{slot} representa el slot de tiempo escogido según la capa física, CW representa la ventana de contención (*Contention Window*) y f(0,CW) es un numero entero pseudoaletorio determinado a partir de una distribución uniforme en el intervalo [0, CW].

El intervalo de backoff se usa para inicializar un temporizador de backoff, el cual es decrementado por cada estación mientras el medio esté libre y se detiene el decremento si se detecta la transmisión de otra estación. Si el medio permanece libre durante un intervalo de backoff, el temporizador de backoff se decrementa en una unidad el conjunto acumulado de slots de tiempo transcurrido. Cuando el temporizador de backoff expira, la estación accede inmediatamente al medio y transmite la trama; pero si el medio aún está ocupado antes de que el temporizador de backoff haya expirado, la estación salva el valor del tiempo de backoff residual (calculado como el intervalo de backoff escogido menos el tiempo de backoff consumido), y dicha estación debe esperar a que el medio se libere, y hasta que eso ocurre, la estación debe esperar un DIFS y agregarle el tiempo de backoff residual que había salvado.

Entre los escenarios que se pueden crear, puede suceder que dos o más estaciones inicien a transmitir en el mismo slot de tiempo y se produzca una colisión. Con el fin de reducir la probabilidad de colisiones consecutivas, la estación que experimente una colisión duplica su CW después de haber intentado transmitir sin éxito, hasta alcanzar un valor máximo (max CW). Las estaciones que han provocado una colisión entre tramas esperan un DIFS y después vuelven a seleccionar un nuevo intervalo de backoff teniendo en cuenta que el valor de CW se ha duplicado [3][4].

Para el caso en el que el número de intentos fallidos es alto, existe un número máximo de reintentos de retransmisión para cada trama, de tal forma que si el número de colisiones que ha sufrido una estación al intentar transmitir una trama ha alcanzado este límite, entonces la estación deja automáticamente de intentar la transmisión de esa trama. Después de una transmisión exitosa, la CW se reinicializa a min CW, el valor inicial de CW.

Cuando una estación ha logrado transmitir una trama de datos exitosamente, deberá esperar que se le envíe un reconocimiento positivo o ACK (Acknowledgement) siempre y cuando la trama de datos enviada se ha recibido correctamente (refiérase a la figura A-2 [2]). La estación receptora enviará el ACK a la estación emisora después de haber transcurrido un intervalo de tiempo SIFS (*Short Inter-Frame Space*), que tiene una duración menor que un DIFS, durante el cual, la demás estaciones no pueden acceder al medio y de esta manera se prevenga una colisión entre tramas [2].

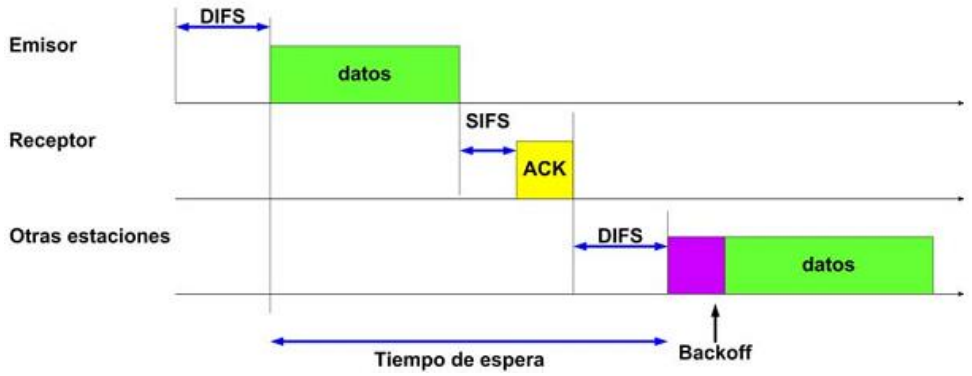


Figura A-2 Ejemplo de una transmisión de trama exitosa.

Además de las ventajas de comunicación que el protocolo CSMA/CA brinda para las redes inalámbricas bajo la norma 802.11, y de las que de ella se derivan como 802.11p, en una topología Ad Hoc o Vanet; estas pueden encontrarse con alguno de los siguientes problemas, que el protocolo CSMA/CA no es capaz de prevenir por sí solo:

A.1.1 Problema del terminal escondido: se genera cuando dos nodos que se hallan fuera de su alcance radial, el uno respecto al otro, intentan enviar al mismo instante información al mismo nodo receptor, pudiéndose producir una colisión de los datos que no es detectable por CSMA/CA, ver figura A-3 [4]. Este problema produce ineficiencia en cuanto a retardo y ancho de banda. Para evitar colisiones, es necesario que todos los nodos vecinos del receptor tengan conocimiento de si el canal se encuentra libre u ocupado. Para esto es utilizado el protocolo de reconocimiento *Handshake* para reservar el canal, mediante el cual el nodo emisor envía una trama RTS (*Request To Send*) indicando al nodo receptor que desea enviar datos, el nodo receptor puede permitir la comunicación enviando una trama CTS (*Clear To Send*). Cuando las tramas RTS y CTS se transmiten en forma broadcast, todos los nodos en su alcance radial quedan informados de que el medio estará ocupado y dejarán de transmitir, evitando colisiones. A pesar de ello, continúa siendo posible que dos estaciones transmitan sus respectivas tramas RTS al mismo tiempo, de forma que colisionen en el nodo receptor, pero esto se considera de menos gravedad que la colisión de datos [4].

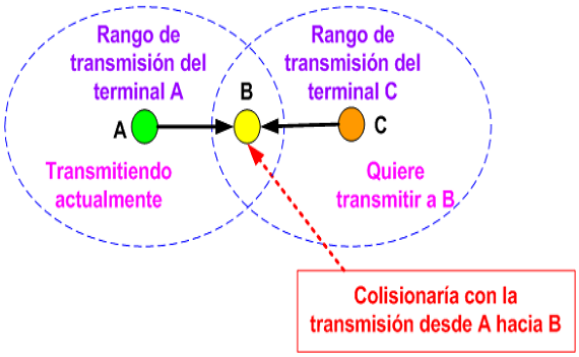


Figura A-3 Problema del terminal escondido.

El análisis que se acaba de efectuar es correcto si el alcance y el rango de transmisión es el mismo (sin que exista rango de interferencia), pero realmente no lo es. Cuando se habla de alcance, se debe definir tres rangos (desde el punto de vista del nodo transmisor):

- Rango de transmisión: Rango dentro del cual un paquete es recibido correctamente.

- Rango sensible a la portadora: Rango dentro del cual el transmisor comprueba si el medio está libre (poniéndose en modo receptor) y por lo tanto puede ser usado para enviar información.

- Rango de interferencia: Rango dentro del cual las estaciones receptoras serán interferidas por el transmisor y podrían sufrir una pérdida de datos.

En términos ideales, la opción de transmitir RTS/CTS puede eliminar la mayor parte de las interferencias, sin embargo, hay estudios [4] que demuestran que la opción de transmitir RTS/CTS no consigue resolver enteramente el problema del terminal escondido. En teoría, cuando una estación A transmite un RTS para poder enviar tramas a una estación B, la estación B transmitirá un CTS para autorizar la transmisión de tramas y al mismo tiempo cualquier nodo vecino que esté dentro de su rango de transmisión y escuche el medio, al recibir la trama CTS se detendrá hasta que sea "seguro" un intento de transmisión. Pero aún puede existir un problema y es cuando un nodo puede estar fuera del rango de cobertura de la estación B, de forma que no recibirá la CTS, por ejemplo, el nodo C en la figura A-4; pero sí puede encontrarse dentro del rango de interferencia del receptor (estación B), ver figura A-4, de tal manera que si la estación B está recibiendo un paquete y si el terminal escondido C decide en ese momento iniciar una transmisión, se producirá una colisión; pero este problema en un sistema tan cambiante como en el entorno vehicular, puede suceder frecuentemente.

Para solucionar el problema se ha propuesto [4] que se envíe un CTS únicamente si la potencia de recepción del RTS se halla por encima de un determinado umbral, porque eso significará que el nodo que ha enviado el RTS no está muy distante y así la zona de interferencia será menor y la calidad del enlace será mayor. El umbral escogido debe ser mayor que el umbral que un nodo requiere para poder recibir con éxito un paquete.

El problema de utilizar este mecanismo es que así se reduce el rango de transmisión efectivo.

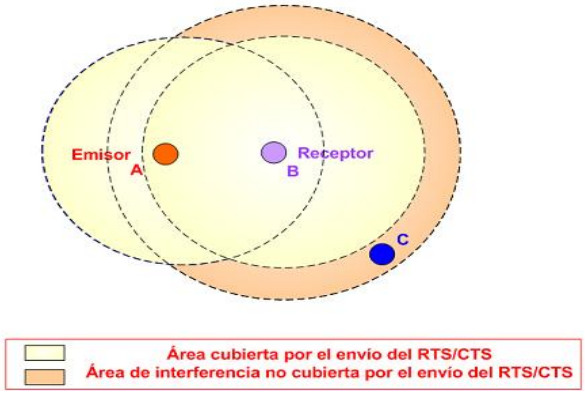


Figura A-4 Cobertura del protocolo basado en RTS/CTS.

A.1.2 Problema del terminal expuesto: Es un problema complementario del anterior y se crea cuando una estación B quiere enviar sus tramas a una estación A y al mismo tiempo una estación C decide enviar sus tramas a una estación D, como en la figura A-5.

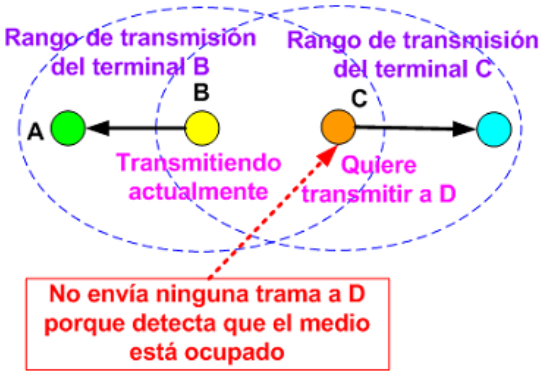


Figura A-5 Problema del terminal expuesto.

Pero C es un terminal expuesto a B; C escucha el medio y comprueba que está ocupado porque se están enviando tramas de la estación B a la A; entonces C cree que si decide transmitir en esos momentos se producirá una colisión y evita hacerlo. De este modo no se hace efectiva la transmisión de paquetes de la estación C a la D, cuando sí se podía hace con éxito, siempre y cuando C estuviera fuera del alcance de A. Una solución al problema del terminal expuesto consiste en utilizar antenas directivas [5].

A.2 EDCA

EDCA o DCF Mejorado está basado en prioridad, utiliza valores de IFS nuevos y diferenciación basada en CW. Este modo de acceso introduce el concepto de calidad de servicio (QoS, *Quality of Service*), dentro del esquema que propone el estándar IEEE 802.11. Este modo se definió en la norma 802.11e y aunque esta norma define dos conceptos de calidad de servicio, se ha escogido explicar solo uno de ellos por la importancia que tiene dentro del trabajo de grado.

A.2.1 Calidad de servicio con prioridad: Su principal tarea es la identificar la prioridad de una trama de datos con relación a otra. Se definen ocho Categorías de Tráfico (*Traffic Categories,* TC) las cuales son un conjunto de tramas de datos que se diferencian del resto mediante un Identificador de Categoría de Tráfico (*Traffic Category Identification,* TCID) de acuerdo a un mapeo predefinido de prioridades. Este valor varía desde 0 hasta 7.

Como ya es conocido, las aplicaciones de datos, video y audio tienen requerimientos de transmisión diferentes. Sin embargo, en el estándar original, IEEE802.11, con la función de coordinación distribuida (DCF, Distributed Coordination Function), todas las estaciones y flujos de datos tienen la misma prioridad de acceso al medio, es decir no hay manera de favorecer el cumplimiento de los parámetros de calidad para cada servicio debido a la ausencia de un mecanismo que brinde prioridad en el acceso al medio según el tipo de paquete. En otras palabras, una estación no tiene la capacidad de diferenciar sus propios paquetes y por lo tanto carece de la capacidad de manejar QoS diferentes.

El acceso distribuido o basado en contienda denominado EDCA (*Enhanced Distributed Channel Access*). Parte del modo DCF, pero expandiéndolo en varias formas; una de estas es que cada estación implementa hasta cuatro categorías de acceso (AC) independientes, cada una está asociada a un determinado tipo de tráfico. El nombre que cada AC indica el tipo de aplicación de la misma: *Background* (BK), Best Effort (BE), Video (VI) y Voz (VO), y cada AC tiene un valor de prioridad o categoría de tráfico, ver según tabla A-1.

Tabla A-1 Mapeo de las Categorías de Tráfico sobre las Categorías de Acceso.

Nivel de prioridad	Categoría de Tráfico (TC)	Categoría de Acceso (AC)	Tipo de Tráfico
Más baja	1	AC_BK	Background
↓	2	AC_BK	Background
	0	AC_BE	Best Effort
	3	AC_BE	Best Effort
	4	AC_VI	Video
	5	AC_VI	Video
	6	AC_VO	Voz
Más alta	7	AC_VO	Voz

Si durante el intervalo de backoff se produce una colisión interna entre dos ACs de la misma estación, llamada colisión virtual, la aplicación con mayor prioridad transmitirá la trama al medio físico, mientras que la otra aplicación se comportará como si hubiese colisionado. Por otra parte, el proceso de backoff también se ve modificado, por la introducción de unos parámetros, diferentes en cada AC, que afectan el comportamiento del mismo. Estos parámetros se los puede observar en la figura A-6:

A.2.2 TXOP (Transmission Oportunity): Cada vez que una estación ha logrado acceder al medio, tiene el privilegio a utilizar el canal durante un tiempo menor o igual al TXOP. Esto permite que se puedan transmitir varias tramas durante ese intervalo de tiempo, existiendo la posibilidad de realizar una configuración por cada trama, o una única configuración final, esta alternativa es más eficiente en ausencia de errores, como se muestra en [6]. Si el valor de este parámetro es igual a cero se asume que la estación transmisora utilizará el canal el tiempo necesario para transmitir una única trama (como en DCF). Si el tiempo necesario para transmitir una trama supera el valor indicado por este parámetro, la trama debe ser fragmentada.

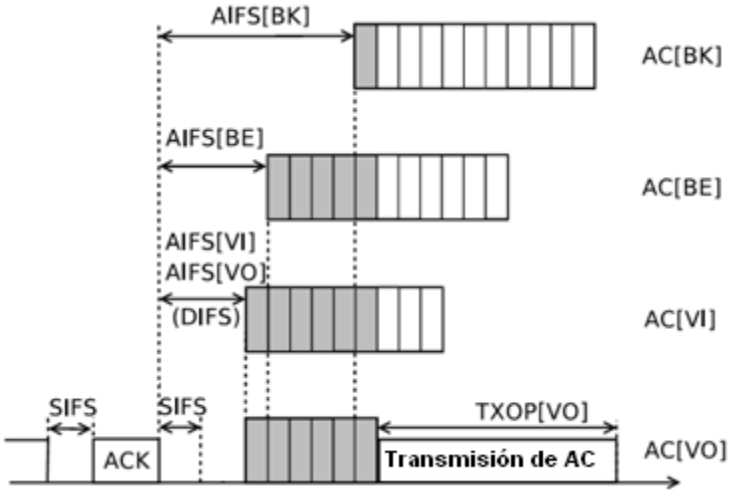


Figura A-6 EDCA – Parámetros.

A.2.3 CW (Contention Window): El funcionamiento es similar al mecanismo DCF, salvo por una diferencia: tanto el valor de CWmin como el de CWmax puede ser diferente para cada clase de tráfico. Ver tabla A-2 [7].

A.2.4 AIFS (Arbitration InterFrame Space): En el modo DCF, cuando hay un slot de tiempo ocupado las estaciones deben esperar DIFS hasta el siguiente decremento de su contador de backoff. En el modo EDCA, el tiempo que debe esperar a una estación hasta decrementar dicho contador es diferente para cada cola de tráfico, y viene determinado por este parámetro.

Este parámetro toma la forma:

AIFS = SIFS + AIFSN x (Slot-Time), con AIFSN ≥ 2.

Donde AIFSN corresponde al Número Arbitrario de Espacio Inter-Trama (AIFSN, *Arbitration Interframe Space Number*) y cuyo valor es un número entero que depende de la categoría de acceso. Ver tabla A-2 [7].

Tabla A-2 Valores por defecto de AIFSN y CW para los estándares IEEE 802.11x.

	AC_VO [0]	**AC_VI [1]**	**AC_BE [2]**	**AC_BK [3]**
AIFSN	2	2	3	7
CWmin	3	7	15	15
CWmax	7	15	1023	1023

En la capa MAC, WAVE también utiliza el método de acceso de IEEE802.11, basado en CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). Para mitigar el problema del terminal oculto, WAVE mantiene el intercambio de mensajes RTS/CTS (Request-To-Send/Clear-To-Send), aunque se encuentra deshabilitado en el canal CCH por transmitir siempre en modo broadcast.

ANEXO B

B.1 MANEJO DEL SIMULADOR NCTUns 6.0 EN 802.11p

En este apartado se explicará el proceso de instalación y uso de la herramienta de simulación NCTUns (National Chiao Tung University Network Simulator), desarrollada por el profesor Shie-Yuan Wang. Además se presenta los recursos hardware y software necesarios para ello y se documenta una guía básica para familiar al lector con el modo de uso a través de un ejemplo.

Debido a que NCTUns 6.0 es construido para que funcione bajo el reingreso de kernel y que lo haga en sistemas operativos Linux solamente, no todos los sistemas operativos de Unix cumplen con los requisitos necesarios para ello, por lo que su creador recomienda Red Hat Fedora 12 con kernel 2.6.31.6 [1]. Una vez instalado este sistema operativo se deben instalar los paquetes de G++ y aceptar las recomendaciones del instalador.

Los requerimientos hardware mínimos recomendados son:

- Procesador de 1.6 GHz de 32 bits.
- Memoria RAM 512 Mb.
- Espacio Libre de 300 Mb en Disco Duro.

El software de instalación puede descargarse desde su sitio oficial, http://NSL.csi.nctu.edu.tw/nctuns.html, y una vez instalado en el equipo, se modificará el MBR creando la opción para el reingreso de kernel desde NCTUns. Esto se consigue bajo la ejecución del archivo install.sh incluido en los instaladores, lo que a su vez creará un directorio bajo el nombre por defecto de “nctuns” ubicado en /usr/local, y que contendrá subdirectorios como “bin,” ”etc,” ”tools,” ”BMP,” y “lib”. La carpeta “bin” contiene los programas ejecutables de la interfaz gráfica de usuario y los motores de simulación, denominados nctunsclient, *dispatcher*, *coordinator* y nctunsse respectivamente. En “tools” se alojarán los ejecutables de varias aplicaciones y herramientas para la configuración de los dispositivos que soporta el simulador. El directorio de “etc” contiene los archivos de configuración necesarios por el programa de GUI, *dispatcher* y *coordinator*, llamados mdf.cfg, *dispatcher*.cfg y *coordinator*.cfg respectivamente. Por su parte, “BMP” contiene todos lo archivos *.bmp utilizados por el programa de GUI y que sirven para mostrar varios iconos de dispositivos y botones de control. Finalmente, “lib” almacena las librerías utilizadas por los motores de simulación, como son protocolos e implementaciones.

Antes de arrancar el simulador se debe configurar las variables de NCTUNSHOME. Para ello debe ejecutarse “setnv NCTUNSHOME /usr/local/nctuns” y escribir el comando “export NCTUNSHOME=/usr/local/nctuns/”. A su vez, se deben cargar las librerías mediante los comandos “setenv LD_LIBRARY_PATH /usr/local/nctuns/” y export LD_LIBRARY_PATH =/usr/local/nctuns/”. Y posteriormente se deben ejecutar los programas: *dispatcher*, *coordinator*, con permisos de Administrador, y nctunsclient como usuario, en este orden estrictamente y se obtendrá la interfaz del simulador tal como se observa en la Figura B -1.

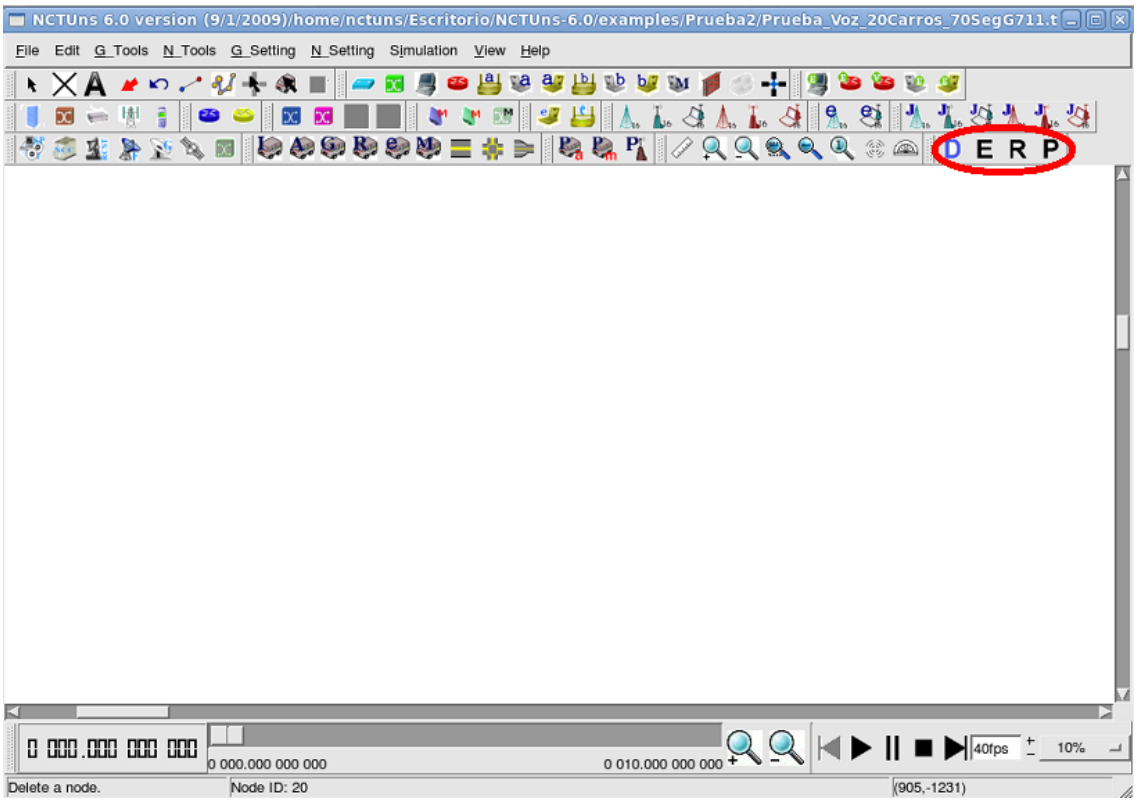


Figura B-1. Interfaz gráfica de NCTUns 6.0

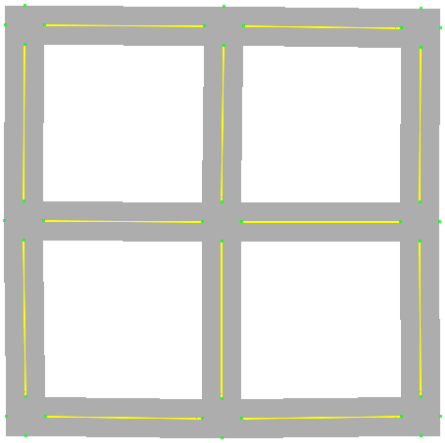
En la figura anterior se ha resaltado el área donde aparecen los botones D, E, R y P, los cuales sirven para realizar todos los procesos de: dibujo de topología (Draw Topology), edición (Edit Property), simulación (Run Simulation) y acción (Play Back) respectivamente. La función de simulación requiere activar el botón R y además activar la acción de "run" que está situada en el menú "simulation", otros requerimientos para la simulación ya aparecen configurados por defecto, por lo que no se explicaran, para mayor información refiérase al manual de usuario de Nctuns 6.0 [8].

En Draw Topology se puede adicionar y eliminar nodos y/o enlaces, en síntesis organizar la topología de red; en Edit Property se pueden agregar las propiedades y especificaciones de cada nodo, así como las aplicaciones; en Run Simulation se puede correr, pausar, continuar, detener, abortar, desconectar y reconectar una simulación; en Play Back se observa la animación creada una vez finalizada la simulación, donde puede observarse el flujo de los paquetes a lo largo de la red simulada, permitiendo cambios en la escala de tiempo y fps (frames per second) enviados por la pantalla.

Entre los distintos dispositivos y objetos con los cuales se pueden realizar simulaciones, se encuentran los utilizados en este trabajo de grado, como los pertenecientes a la tecnología 802.11p y otros (ver Figura B-2.):

NCTUns 6.0 permite la construcción de la red vial en donde se moverán los vehículos. Las siguientes Figuras muestran los tipos de vías utilizadas para construir la red vial [1]:

1. Segmento General.



2. Intersección.

Los dispositivos de generación de tráfico y recepción utilizados en este trabajo de grado son:

3. Host.

4. OBU con interface 802.11p.

5. RSU con interface 802.11p.

6. Router

7. AP con interface 802.11b.

8. OBU con interface 802.11b.

Para comenzar con el posicionamiento de los OBU y la RSU se debe primero, escoger el número de OBUs y ubicarlos en la red vial y las RSU al lado de la carretera, luego de procede a interconectar todos los nodos cableados y posteriormente se define el perfil de los OBU así:

Un perfil indica las características de conducción del vehículo, es decir, su velocidad, aceleración y desaceleración que tendrá durante la simulación.

La herramienta permite configurar hasta 5 perfiles con diferentes características. A continuación, en la figura B-3 se muestra la ventana de configuración de los perfiles:

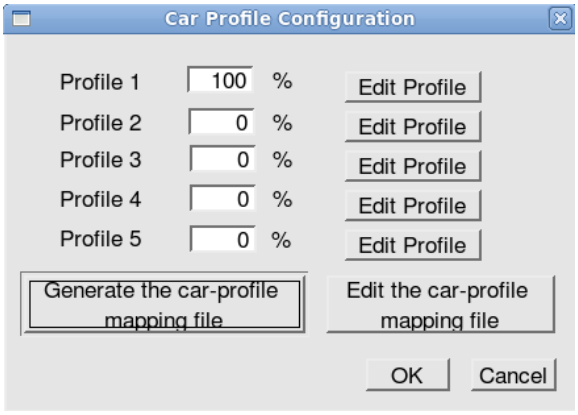


Figura B-3 Ventana de configuración de perfiles.

La herramienta permite la edición de los perfiles a través del botón editar perfiles en donde sale la ventana representada en la Figura B-4.

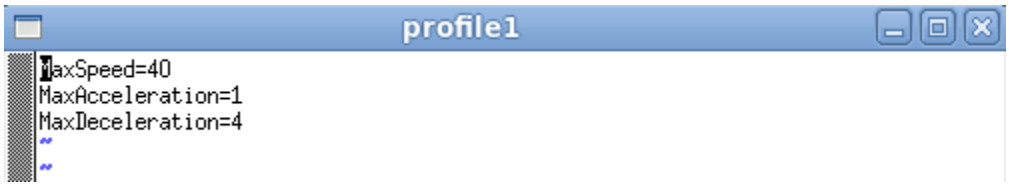


Figura B-5 Ventana de edición de perfil.

La herramienta también permite cambiar el perfil a un OBU según las necesidades del usuario mediante la opción de edición del perfil (edit profile) mostrada en la Figura B-5.

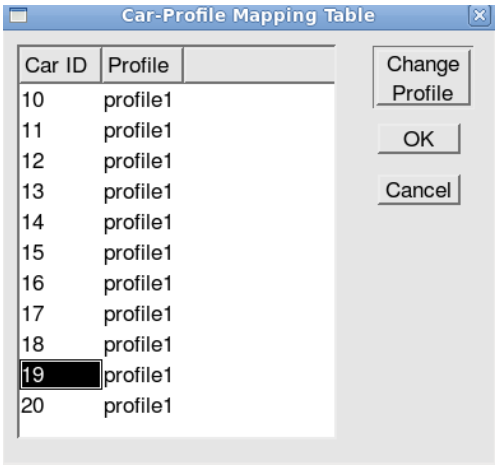


Figura B-5 Ventana de mapeo de los perfiles.

Luego de tener configurado los perfiles de los OBU, se sigue con la configuración del host, quien es el encargado de generar el tráfico que se va a enviar por la red.

Nctuns 6.0 presenta varios comandos para la generación del tráfico, a modo de ejemplo se presentara solo los usados en las simulaciones del presente trabajo de grado, los cuales son:

stg y stcp, para el envío; rtg y rtcp, para la recepción de los paquetes de información.

Sintaxis de los comandos:

1. **stcp** -p ### Dirección IP destino: Generador de tráfico

2. **rtcp** -p ###: Receptor de tráfico.

En donde -p ### es el número del puerto por donde en envía o recibe información.

3. **stg** -modos Dirección IP del destino -opciones

-modos:

-t Duración (seg) Genera tráfico TCP con una duración en segundos determinada por el usuario.

-u Tamaños de la carga útil de datos (bytes) Duración (seg): Se transmite tráfico UDP con una carga útil dada por un tiempo específico.

-i XXX.cfg Genera tráfico de acuerdo a las características del archivo de configuración XXX.conf

-opciones:

-p YYY Envía tráfico por el puerto YYY

Formato del archivo XXX.conf:

_____

type: ###: Protocolo ### ya sea TCP o UDP

start_time: ###: Tiempo ### expresado en segundos desde el cual se comienza la transmisión del primer paquete.

on-off: ###: Número de líneas entre la línea actual y la línea que contiene la cadena de caracteres “end” que será repetida ### veces.

on: [time: XXX o packet YYY] [distribución de tiempo entre transmisiones consecutivas]

[distribución de tamaño de paquetes]: Indica que el tráfico debe ser generado basándose en un intervalo de tiempo específico con una determinada distribución de tiempo y de tamaño de paquetes.

end

_____

[time:XXX] indica que se debe generar tráfico durante XXX segundos

[packet:YYY] indica que se deben generar YYY paquetes únicamente

De las anteriores configuraciones solo se debe escoger una.

La distribución de tiempo entre transmisiones consecutivas puede ser:

[const XXX]: indica que el tiempo que transcurre entre el envío de paquetes consecutivos es un valor fijo de XXX segundos.

[uniform XXX, YYY]: el tiempo transcurrido entre el envío de paquetes consecutivos varía entre una valor mínimo igual a XXX segundos y un máximo igual a YYY segundos.

[exponential XXX, YYY, ZZZ]: el tiempo de envío entre paquetes consecutivos es una función exponencial con una media igual a XXX, un valor mínimo de YYY segundos y un máximo de ZZZ segundos.

La distribución del tamaño del paquete puede ser:

[const XXX]: indica que el tamaño de los paquetes es un valor XXX fijo.

[uniform XXX, YYY]: indica que el tamaño de los paquetes generados es función de una distribución uniforme con un valor mínimo de XXX bytes y un máximo de YYY bytes.

[exponential XXX, YYY, ZZZ]: el tamaño de los paquetes generados es función de una distribución exponencial con una media de XXX bytes, un valor mínimo de YYY bytes y un máximo de ZZZ bytes.

off: time: ###: el generador de tráfico deja de transmitir por un período de ### segundos.

Nota para tráfico UDP:

stg soporta cuatro tipos de distribuciones de tiempo para el tráfico UDP:

(1) const XXX: Tasa Constante de Bits (Constant Bit Rate, CBR) con distribución de tiempo de XXX segundos

(2) uniform XXX YYY: Distribución con un tiempo mínimo de XXX bytes y un máximo de YYY bytes.

(3) exponential XXX YYY ZZZ: Distribución con promedio XXX, mínimo YYY y máximo ZZZ (Flujos de paquetes con una distribución de Poisson)

(4) greedy: tiempo aleatorio

stg soporta tres tipos de tamaños de paquetes (en bytes) para el tráfico UDP:

(1) const XXX: constante
(2) uniform XXX YYY: valor mínimo XXX y máximo YYY
(3) exponential XXX YYY ZZZ: promedio XXX, valor mínimo YYY y máximo ZZZ.

Nota: El máximo tamaño de los datos no puede ser mayor a los 1500 bytes de la Máxima Unidad de Transferencia (Maximun Transfer Unit, MTU) Ethernet.

4. **rtg** -tipo -opciones

-tipo -t Conexión TCP
-u Conexión UDP

-opcione]

-p ### La estación escucha por el puerto número ### (por defecto 3000)
-o XXX.log Registra estadísticas de delay y pérdida de paquetes en un archivo de nombre XXX.log.
-w YYY.log Registra el throughput en un archivo de nombre YYY.log

La configuración del Host seria la presentada en la figura B-6, que de forma similar se presenta en cualquier terminal generador de tráfico de la herramienta Nctuns 6.0.

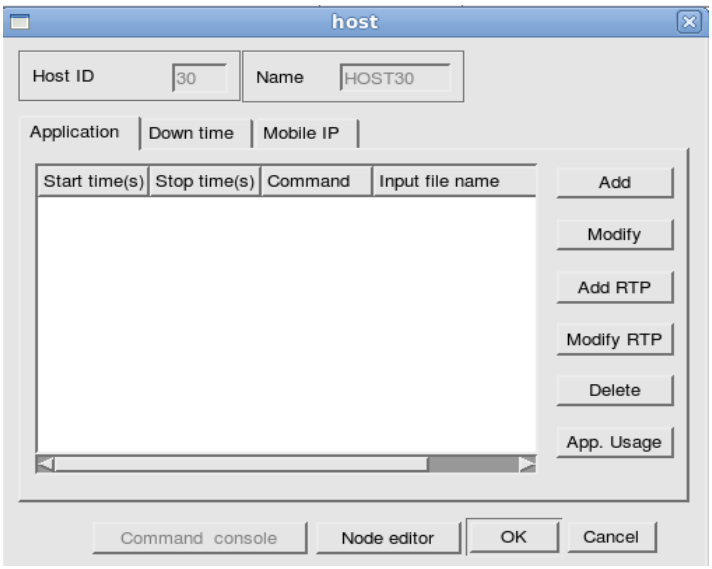


Figura B-6 Ventana de configuración de Host.

En la ventana anterior se presentan las opciones para agregar (Add), modificar (Modify) y borrar (Delete) las aplicaciones para generar el trafico requerido, al presionar el botón Add o Modify se despliega la venta siguiente (ver figura B-7).

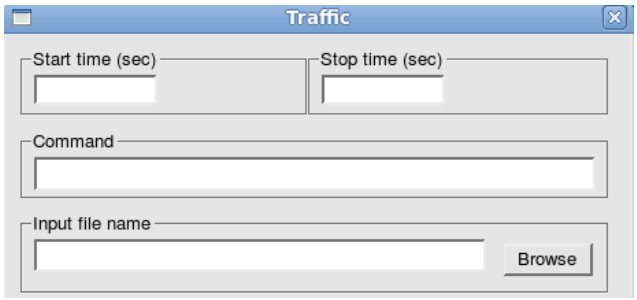


Figura B-7 Ventana de configuración de Tráfico.

Aquí se agrega el tráfico a generar, primero se tiene los campos Start time y Stop time en los cuales se establece el intervalo de tiempo durante el cual se generara el tráfico, dentro del tiempo que durara la simulación, en el campo Command se configura el comando a utilizar como se explicó anteriormente y finalmente en el campo Input file name se da la ruta del archivo de configuración, si es el caso.

Los dispositivos como host, RSU, etc., los nodos inalámbricos OBUs IEEE 802.11p también presentan una interfaz de configuración de atributos, y al entrar en su “Node Editor” se podrá observar la constitución de este nodo, en especial los módulos basados en 802.11p tanto en los OBUs y RSU que funcionen bajo este tipo de tecnología (ver Figuras B-6, B-7, B-8 y B-9).

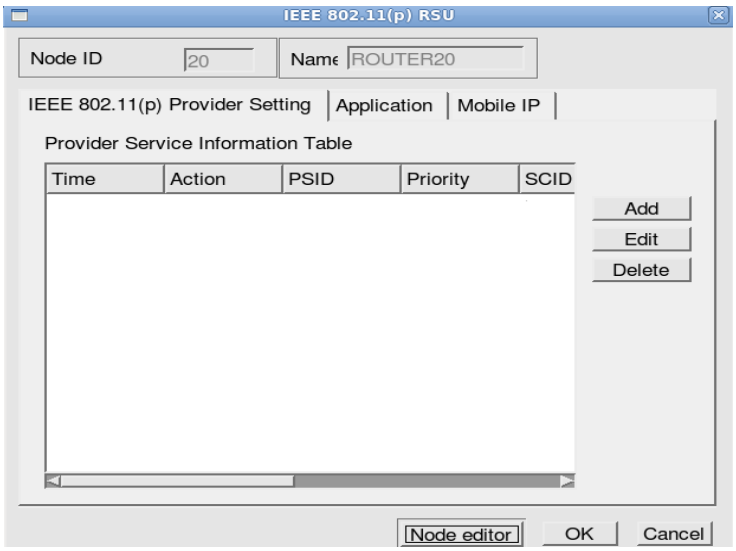


Figura B-8 Ventana de configuración de atributos de RSU.

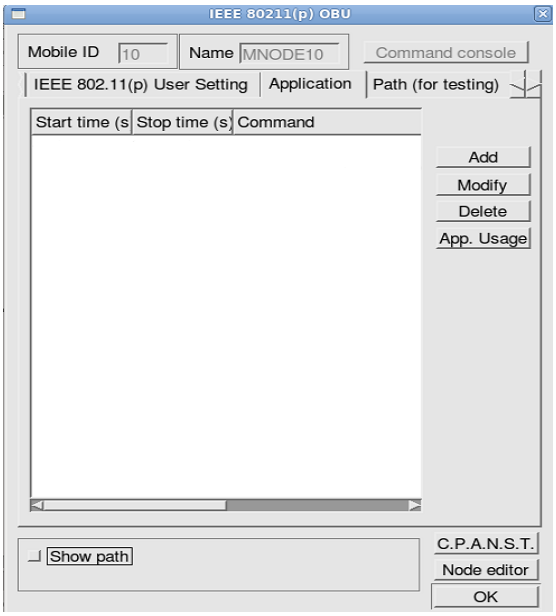


Figura B-9 Ventana de configuración de atributos de OBU.

Luego se pasa al modo de Run Simulation por lo que debe irse al menú Simulation -> Run (ver Figura B-10).

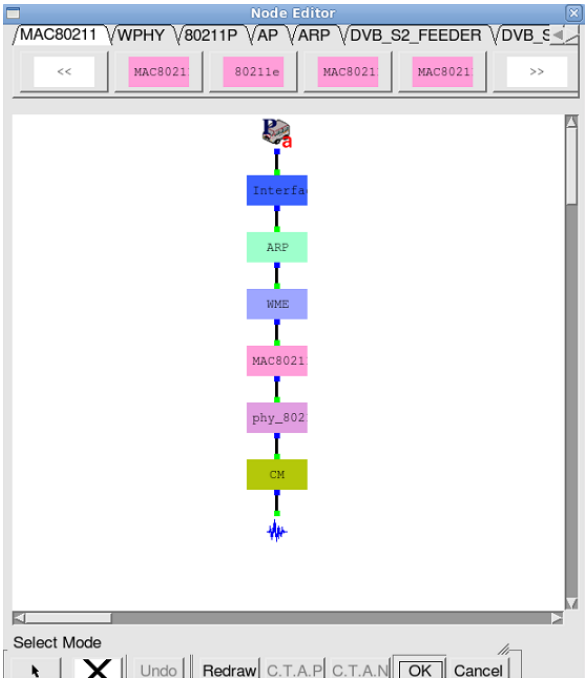


Figura B-10 Ventana de configuración de node del OBU.

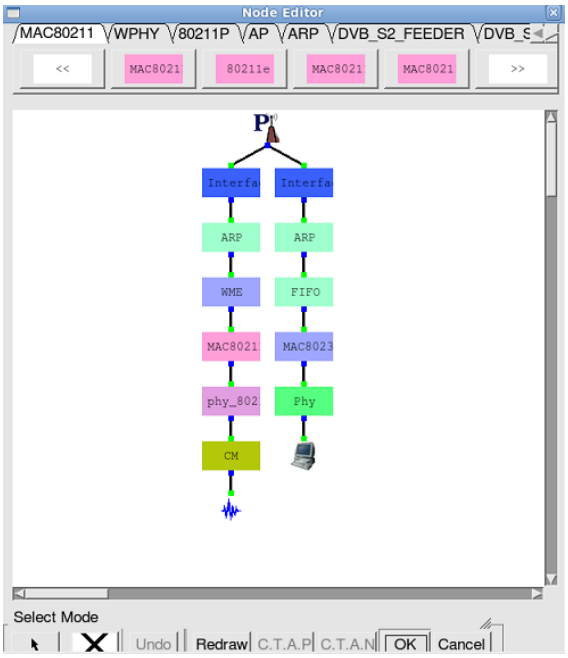


Figura B-11 Ventana de configuración de nodo del RSU.

Finalmente, en el modo Play Back puede observarse la animación creada como resultado de la simulación donde podrá observase el flujo de información entre los diferentes nodos (ver Figura B-11).

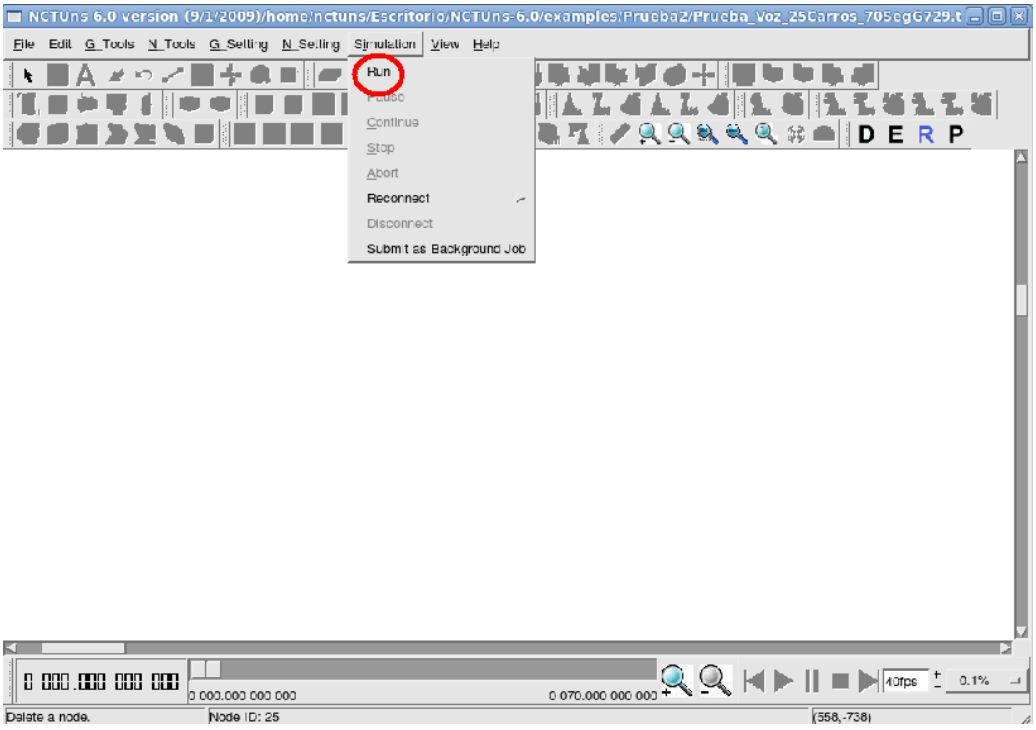


Figura B-11 Activación de Simulación en NCTUns 6.0

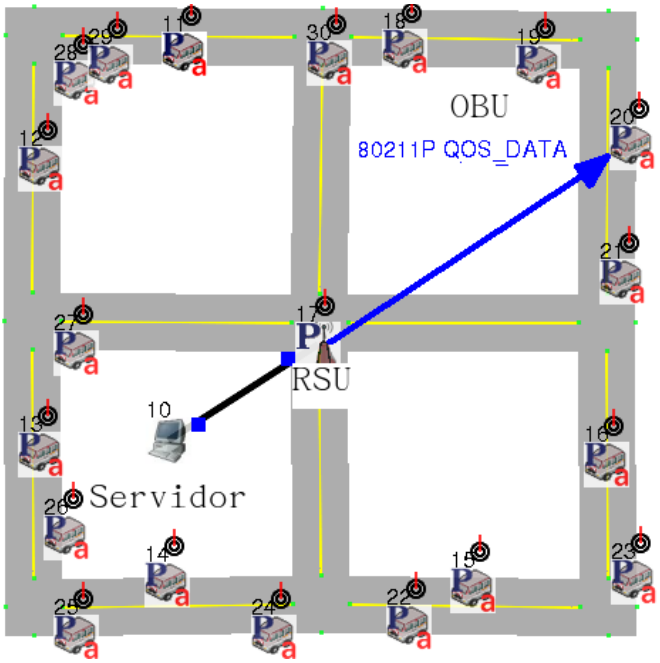


Figura B-12 Resultados vistos mediante el Play Back de NCTUns 6.0

B.2 RECOLECCIÓN DE DATOS A TRAVÉS DEL SIMULADOR NCTUns 6.0

Una vez culminada la etapa de simulación, NCTUns almacena los resultados obtenidos en una carpeta llamada Nombre_de_Archivo.results según la ruta que el usuario haya configurado, por ejemplo: `/usr/local/nctuns/Nombre_de_Archivo.results`.

Estos resultados se encuentran en formato de texto plano por lo que la apertura del archivo puede realizarse desde cualquier procesador de texto convencional.

Para la recolección de los resultados generados por los comandos descritos anteriormente es necesario ingresar el comando **rtg -u -w archivo1.log -o archivo2.log** en la estación receptora, de esta manera la estación está en la capacidad de capturar todo el tráfico UDP dirigido hacia ella, registrar el *throughput* en el archivo denominado "archivo1.log" y de igual manera, almacenar la información relacionada con la pérdida de paquetes y el retardo en el archivo llamado "archivo2.log". Como se mencionó anteriormente, estos archivos se encuentran en la carpeta Nombre_de_Archivo.results en la ruta que el usuario haya configurado previamente.

En la Figura B-13 se muestra como está compuesto el archivo1.log

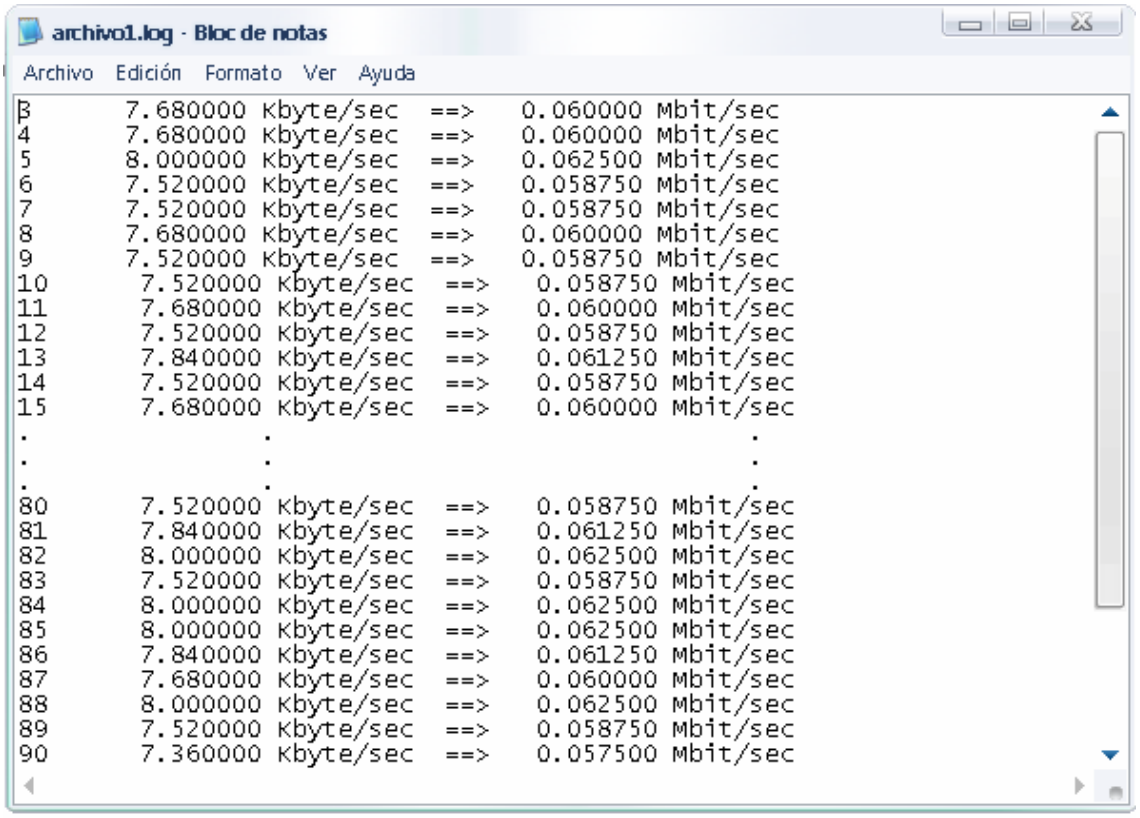


Figura B-13. Representación del parámetro *throughput* mediante el comando rtg.

En la anterior figura se puede apreciar la variación del *throughput* en KBps y su equivalencia en Mbps en intervalos de tiempo de un segundo. La manera en que la pérdida de paquetes y el retardo son representados en el archivo2.log se muestra en la Figura B- 11. Cabe anotar que hasta el momento se han obtenido resultados tomando al tiempo como variable sobre el eje *x*.

A pesar de que la longitud de este archivo (líneas de código) es muy extensa, este cuenta con un resumen estadístico al final del mismo, el cual es de gran ayuda en el proceso de análisis de datos puesto que brinda una idea general sobre el comportamiento de la red de acuerdo a la configuración de ciertos parámetros de entrada. Sin embargo, la información detallada que proporciona este archivo es necesaria en el momento de la generación de gráficas mediante el uso de Microsoft Excel.

A partir de lo anterior se deduce que se requiere de dos archivos (archivo1.log, archivo2.log) cuyos datos sirven como información de origen para el eje *y* en Microsoft Excel.

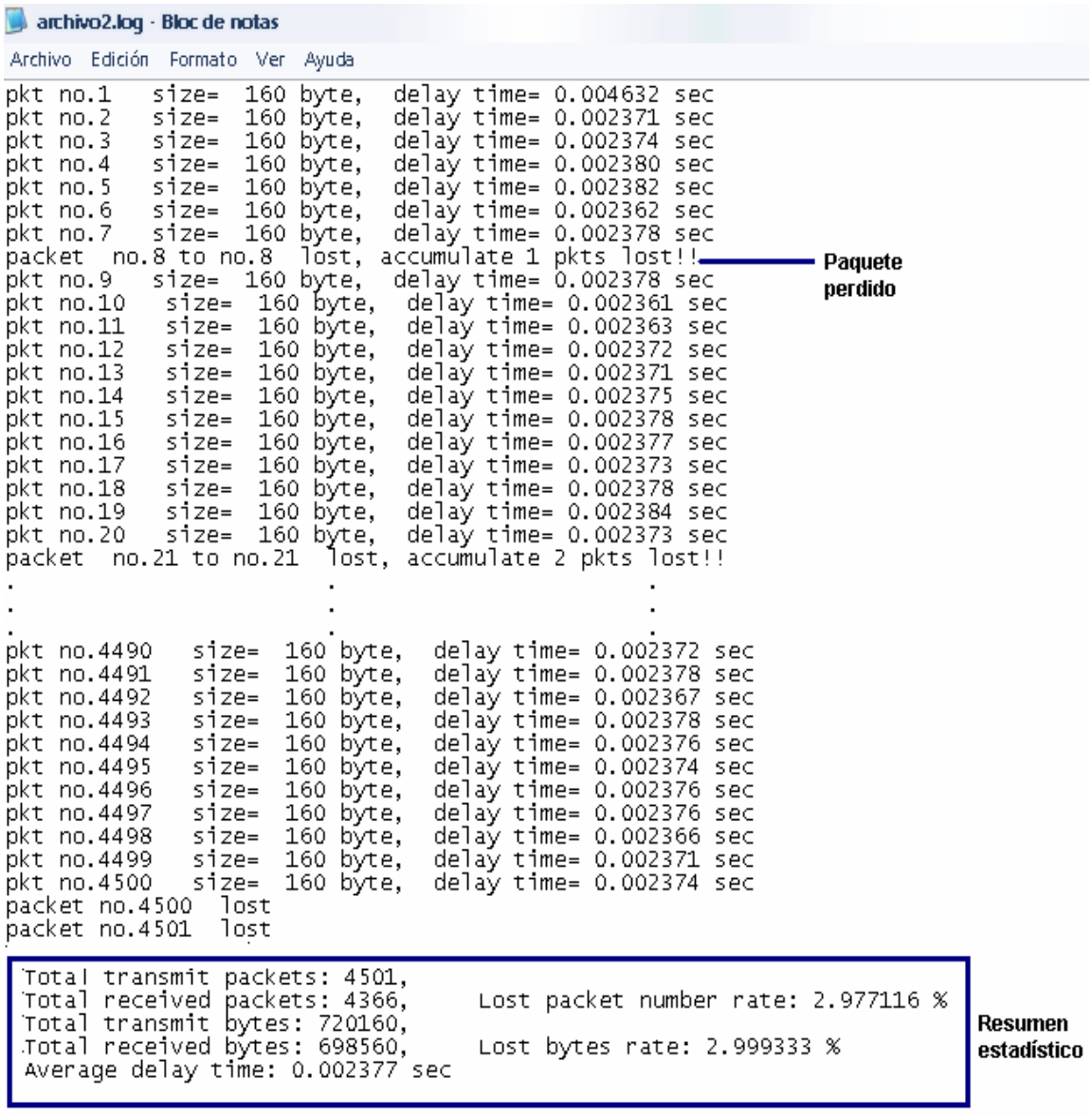


Figura B2. Representación de los parámetros retardo y pérdida de paquetes mediante el comando rtg.

Adicionalmente, es necesario considerar el procedimiento de recolección de datos al variar el número de estaciones o la distancia convirtiéndose esta información en un parámetro de entrada (Eje *x*) para los escenarios que así lo requieran. Como se describió anteriormente, los archivos de importancia son los generados por el comando **rtg** salvo que en esta ocasión, se dará origen a los mismos dos archivos (archivo1.log, archivo2.log) por cada variación en el número de estaciones. Es decir, para analizar el comportamiento de un *códec* frente a la variación del número de clientes o la distancia, se debe observar el comportamiento de los datos contenidos en los archivos "archivo1.log, archivo2.log" en cada carpeta para posteriormente concatenar tales resultados y representarlos en una gráfica en común.

REFERENCIAS

[1] A. León, I. Widjaja, “Redes de Comunicación: Conceptos fundamentales y arquitecturas básicas”, ed. Mc Graw-Hill, 2002. [Consultado: Junio 2010].

[2] J. H. Schiller, “Mobile Communications”, ed. Addison Wesley Professional, 2000. [Consultado: Julio 2010].

[3] D. Remondo, "Tutorial on Wireless Ad Hoc Networks," Conf. on Performance Modelling and Evaluation of Heterogeneous Networks,” *Ilkley*, West Yorkshire, U.K., July 26-28, 2004. [Consultado: Marzo 2010]

[4] K. Xu, M. Gerla, S. Bae, “Effectiveness of RTS/CTS Handshake in IEEE 802.11 Based Ad Hoc Networks”, Ad Hoc Networks Journal, Volume 1, Issue 1, July 2003, pp. 107 – 123. [Consultado: Abril 2010].

[5] C-K Toh, “Ad Hoc Mobile Wireless Networks”, Prentice Hall, 2002. [Consultado: Abril 2010]

[6] Y. Xiao y J. Rosdahl. Performance Analysis and Enhancement for the Current and Future IEEE 802.11 MAC Protocols. SIGMOBILE Mob. Comput. Commun, pp 6–19, 2003. [Consultado: Mayo 2010].

[7] Estándar IEEE 802.11e, “IEEE Standard for Information technology-Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. Amendment 8: Medium Access Control (MAC) - Quality of Service Enhancements”, 2005. Disponible en: http://standards.ieee.org/getieee802/download/802.11e-2005.pdf [Consultado: Marzo 2010].

[8] The GUI User Manual for the NCTUns 6.0, Disponible en : http://nsl10.csie.nctu.edu.tw/support/documentation/GUIManual.pdf [Cosultado: Agosto 2010].