

ANEXO B

MANEJO DEL SIMULADOR NCTUns 3.0

En este documento se examinarán asuntos concernientes a la instalación y uso de la herramienta de simulación NCTUns (National Chiao Tung University Network Simulator), creada por el profesor Shie-Yuan Wang. Se expondrán los recursos hardware y software necesarios para ello y, para facilitar su comprensión, se explicará su modo de uso a través de un ejemplo.

Debido a que NCTUns se basa en una metodología de simulación de reingreso de kernel y que sólo funciona en sistemas Linux, no todos los sistemas operativos de Unix cumplen con los requisitos necesarios para ello, por lo que su creador recomienda sistemas creados por Debian y Red Hat, en especial Red Hat Fedora Core 4.0 con kernel 2.6.11 [1]. Una vez instalado este sistema operativo se deben instalar los paquetes de G++ y deshabilitar el archivo SELinux.

En cuanto a los requerimientos hardware, se recomienda, como recursos mínimos, un procesador de 1.6 GHz de 32 bits, 512 Mb en RAM y 300 Mb libres en disco duro disponibles sólo para su instalación.

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 de “bin” contendrá los programas ejecutables de la interfaz gráfica de usuario, *dispatcher*, *coordinator*, 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 los 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.

Para arrancar el simulador, deben activarse los programas de *dispatcher*, *coordinator* y *nctunsclient*, en dicho orden, por lo que antes deben configurarse la variable de NCTUNSHOME. Para ello debe ejecutarse “setenv 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/”. Ahora pueden correrse los programas de *dispatcher*, *coordinator* y *nctunsclient* ubicados en /usr/local/nctuns/bin, como se mencionó antes, a lo cual surgirá la interfaz del simulador tal como se observa en la Figura A1.

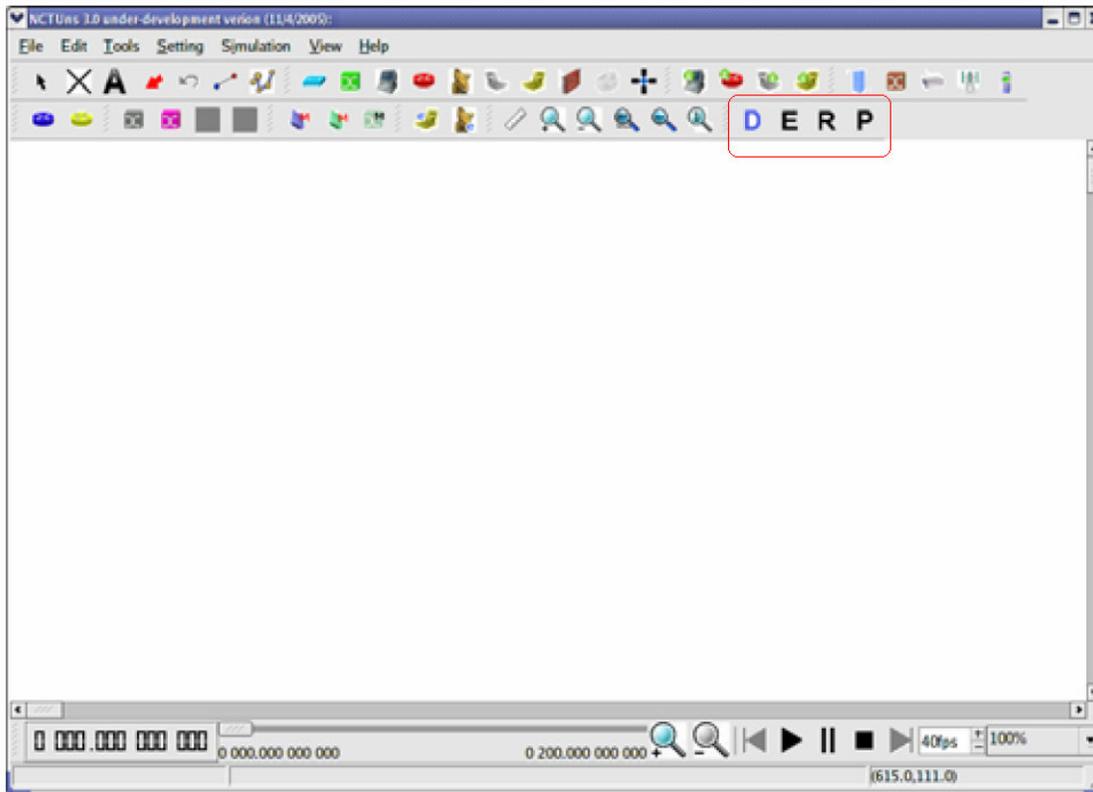


Figura A1. Interfaz gráfica de NCTUns 3.0

Como se observa en el área resaltada de la Figura A1, los botones D, E, R y P, se refieren a los módulos de dibujo (**Draw Topology**), editor (**Edit Property**), arranque (**Run Simulation**) y acción (**Play Back**), respectivamente. Sin embargo, el programa no funcionará si se opera bajo una sesión de root, por lo que debe establecerse otra cuenta de usuario y su respectivo pasword en Settings -> Simulation (ver Figura A2).

Entre los distintos dispositivos con los cuales se pueden realizar simulaciones se encuentran (ver Figura A3.):

1. host, 2. hub, 3. switch, 4. router, 5. AP WLAN, 6. nodo móvil WLAN (modo ad-hoc), 7. nodo móvil WLAN (modo infraestructura), 8. host externo 9. router externo, 10. nodo externo móvil WLAN (modo ad-hoc), 11. nodo externo móvil WLAN (modo infraestructura), 12. GGSN GPRS, 13. pseudo switch GPRS, 14. SGSN GPRS, 15. estación base GPRS, 16. teléfono GPRS, 17. router de frontera con soporte para DiffServ, 18. router interno con soporte para DiffServ, 19. circuito de conmutación para redes ópticas, 20. burst switch óptico, 21. AP para redes en malla, 22. AP con STP para redes en malla, 23. switch multipasarela para redes en malla, 24. nodo móvil 802.11e, 25. AP 802.11e.

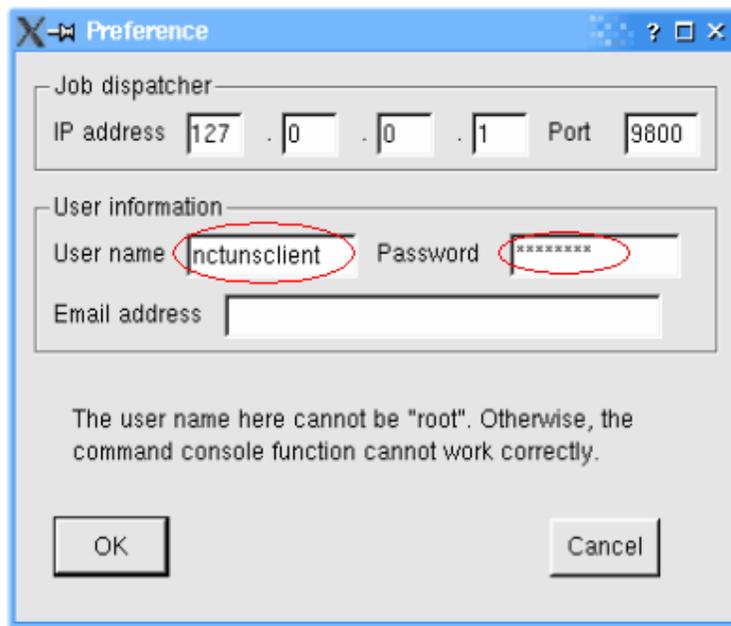


Figura A2. Información de usuario.



Figura A3. Iconos de los diferentes dispositivos soportados por NCTUns 3.0

El modo **Draw Topology** permite la adición y eliminación de nodos y/o enlaces; en el modo **Edit Property** se pueden agregar las propiedades y especificaciones de cada nodo, así como sus aplicaciones, sin embargo, no pueden realizarse cambios en la topología por lo que se debe regresar al modo anterior; en el modo **Run Simulation** se puede correr/pausar/continuar/detener/abortar/desconectar/reconectar una simulación, pero para ella debe guardarse el modelo de simulación; por último, el modo **Play Back** permite observar 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.

EDITOR DE TOPOLOGÍAS

Contiene los cuatro modos mencionados anteriormente: **Draw Topology**, **Edit Property**, **Run Simulation** y **Play Back**. A través del primero de ellos se pueden agregar todos los elementos pertenecientes a cierta topología de red, incluidos nodos como los que se muestran en la Figura A3, así como conexiones cableadas, trayectorias de equipos móviles, obstáculos, incluir etiquetas, etc. Pero es mediante el modo **Edit Property** que pueden configurarse los atributos de los nodos de red.

Para el caso de un *host*, la ventana de la Figura A4 aparecerá cuando se de doble click sobre él, en la cual podrán adicionarse los comandos que dictaminen los atributos del mismo. A través del botón “App. Usage” (ver Figura A5) se obtiene información sobre las distintas aplicaciones que puede ejecutar este terminal.

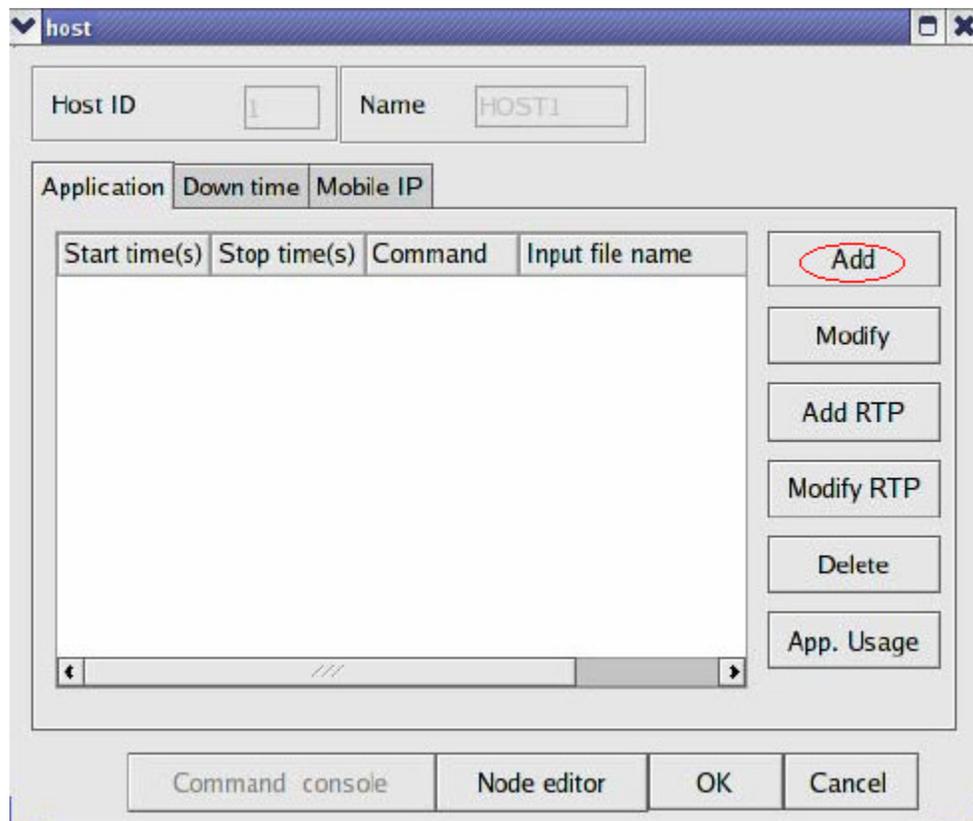


Figura A4. Ventana de atributos de un *host*.

Para agregar una de las aplicaciones indicadas en la ventana de manual de usuario, se presiona el botón “Add” lo cual dará lugar a la ventana indicada en la siguiente Figura A6.

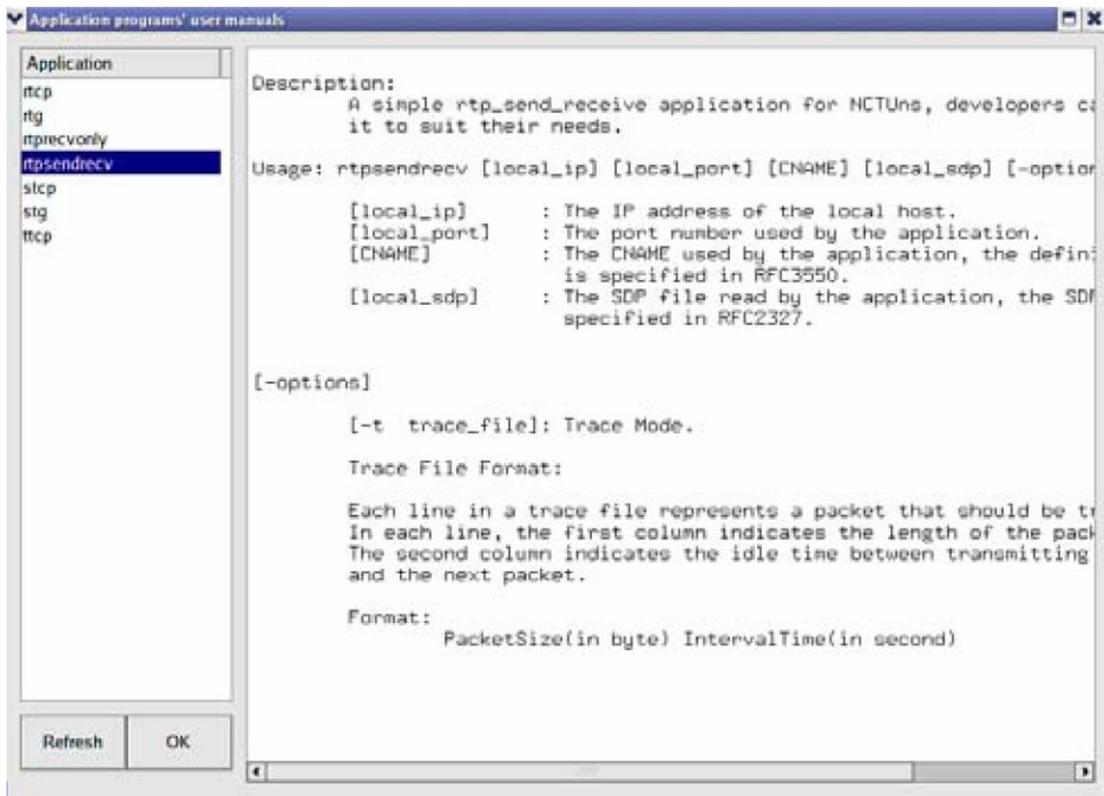


Figura A5. Ventana de manual de usuario sobre aplicaciones programadas.

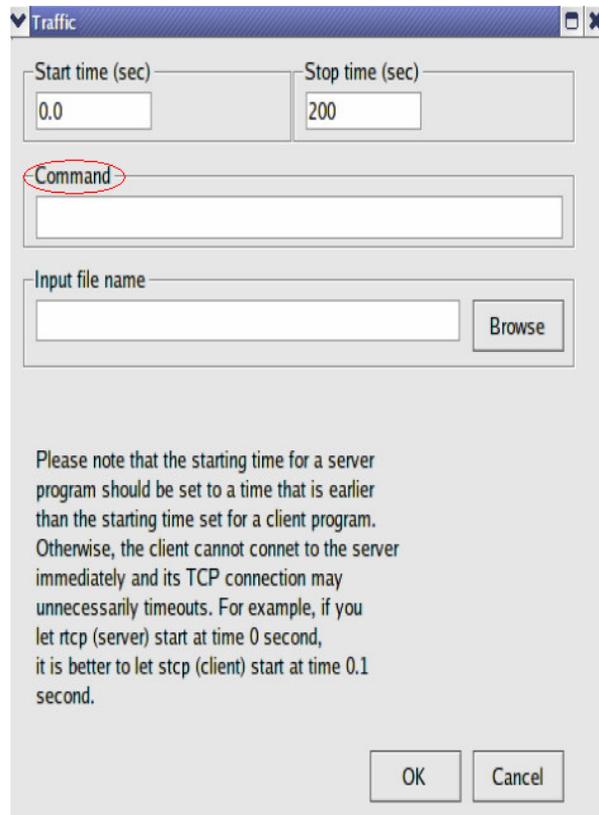


Figura A6. Ventana de comandos de tráfico.

Una vez se agregan los nodos, el usuario puede modificar los atributos de ese nodo haciendo doble clic sobre el icono. En particular, la herramienta de simulación NCTUNS en su versión 3.0 trae consigo un conjunto de alternativas para generar tráfico a través de la red en cuestión. El primer paso es ingresar a la ventana de configuración de tráfico haciendo clic en el botón “Add”:

Su uso se basa en el ingreso de un comando o programa de aplicación en el campo “Command” en donde adicionalmente el usuario está en capacidad de modificar los tiempos de inicio y fin de la generación de tráfico. Esto se muestra en la Figura A6. Si en algún momento el usuario desea agregar un nuevo comando, modificar o borrar uno ya existente sólo es necesario hacer click sobre los botones “Add”, “Modify” o “Delete” respectivamente.

Entre los comandos utilizados en el trabajo de grado para generar tráfico se encuentran:

Comandos:

- **ttcp**

Sintaxis:

ttcp -t [-opciones] dirección IP host destino: Genera tráfico

ttcp -r [-opciones]: Recibe tráfico

Opciones comunes a ambos comandos:

-u hace uso de UDP (por defecto TCP)

-p *YYY* envía o recibe tráfico por el puerto *YYY*

Opciones específicas de la opción -r:

-w *XXX.log* almacena en un archivo de nombre “*XXX.log*” un registro del throughput por segundo

Ejemplo 1:

ttcp -t -p 8000 1.0.1.2 : Este comando es característico de una estación que genera tráfico encapsulado sobre TCP en el puerto 8000 con destino la estación con dirección IP 1.0.1.2. Nótese que al no incluir la opción -u, se genera automáticamente tráfico TCP.

ttcp -r -p 8000: La estación que contenga este comando registrará todo el tráfico UDP que reciba por el puerto 8000 proveniente de cualquier estación.

- **stg (generador de tráfico), rtg (receptor de tráfico)**

Sintaxis:

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.cfg

[-opciones]

- p YYY Envía tráfico por el puerto YYY

Formato del archivo XXX.cfg:

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 un 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 del paquete no puede ser mayor a los 1500 bytes de la Máxima Unidad de Transferencia (*Maximun Transfer Unit*, MTU) Ethernet.

Ejemplo 2:

Un ejemplo de generación del *códec* G.711¹ haciendo uso de la opción.

Ejemplo.cfg

```
type: udp
start_time: 1
on-off: 1
on:   time: 90   const 0.02   length: const   160
end
```

En este caso se genera tráfico con una longitud de trama de 160 bytes con una distribución de tiempo de 0.02 segundos es decir 20ms (const 0.02) por un tiempo de 90 segundos (time) a partir del primer segundo de haber iniciado la simulación (start_time).

Es así como el comando `stg -i Ejemplo.cfg -p 5004 1.0.1.5` envía tráfico por el puerto 5004 hacia la estación con dirección IP 1.0.1.5 de acuerdo a la configuración especificada en el archivo ejemplo.cfg

- **rtg**

Sintaxis:

rtg [-tipo] [-opciones]

[-tipo] -t Conexión TCP
 -u Conexión UDP

[-opciones]

-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

Ejemplo 3:

`rtg -u -w archivo1.log -o archivo2.log`

En este caso la estación receptora generará archivos con nombres archivo1.log archivo2.log que contendrán las estadísticas relacionadas con el tráfico UDP entrante por cualquier puerto (no se ha especificado el número del puerto).

- **stcp, rtcp**

¹ *Códec* con carga útil de 160 bytes y tamaño de muestra de 20 ms.

Sintaxis:

stcp -p ### Dirección IP destino : Generador de tráfico

rtcp -p ###: Receptor de tráfico.

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

De manera similar pueden editarse tanto *hubs*, *switch*, *router*, AP y WANs (ver Figuras A7, A8, A9, A10 y A11 respectivamente).

Para el caso de estaciones inalámbricas, estas cuentan con la opción de trabajar tanto en modo ad-hoc, en el cual es posible ingresar instrucciones de trayecto; como en modo infraestructura, en el que puede asignarse direcciones IP, así como la creación de subredes, etc. Y en ambos casos pueden agregarse obstáculos para el estudio de entornos *indoor*, por ejemplo, en los cuales pueda evaluarse la implicación de los mismos (ver Figura A12).

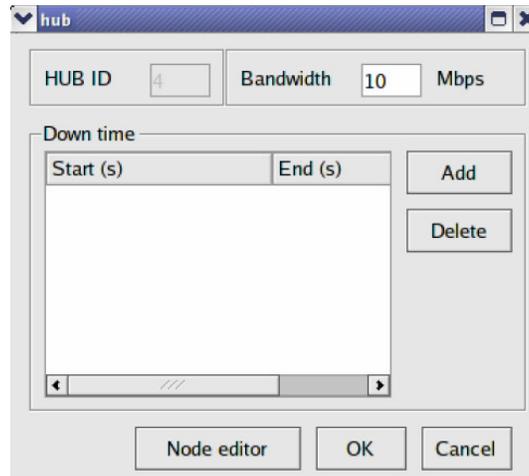


Figura A7. Ventana de atributos de un *hub*

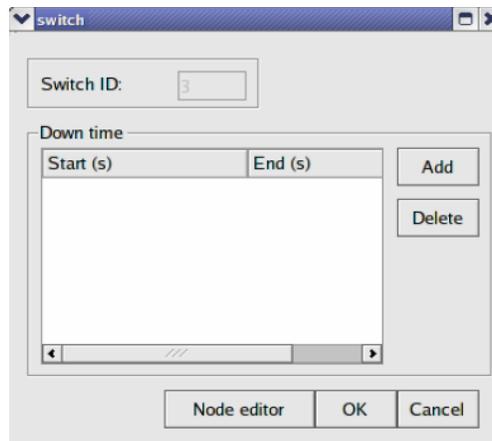


Figura A8. Ventana de atributos de un *switch*.

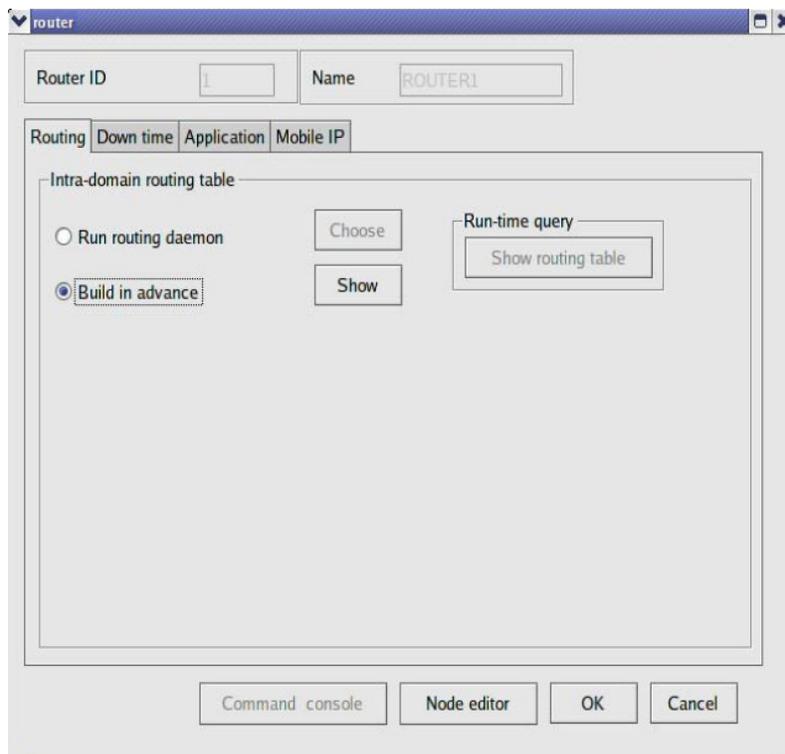


Figura A9. Ventana de atributos de un *router*.

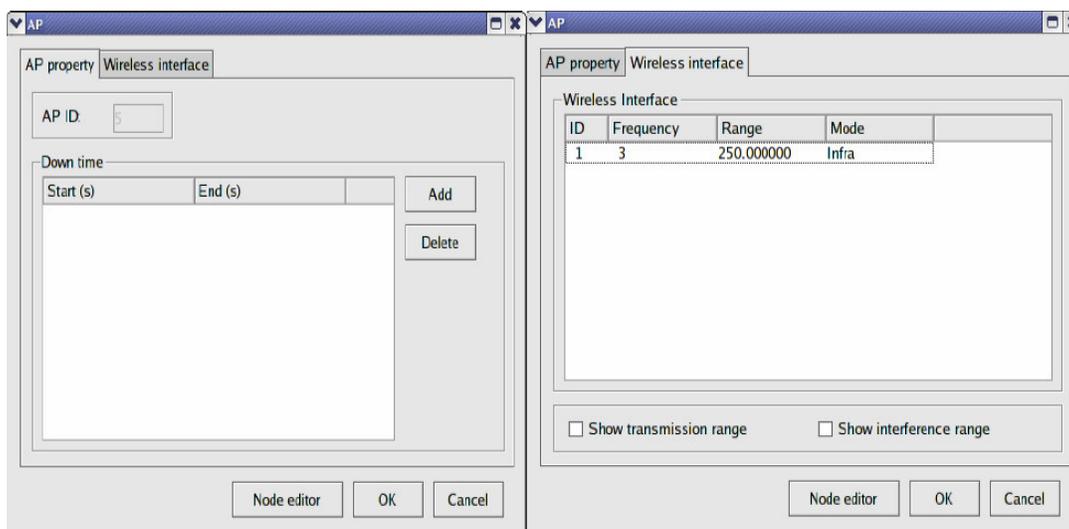


Figura A10. Ventana de atributos de un AP.



Figura A11. Ventana de atributos de una WAN.

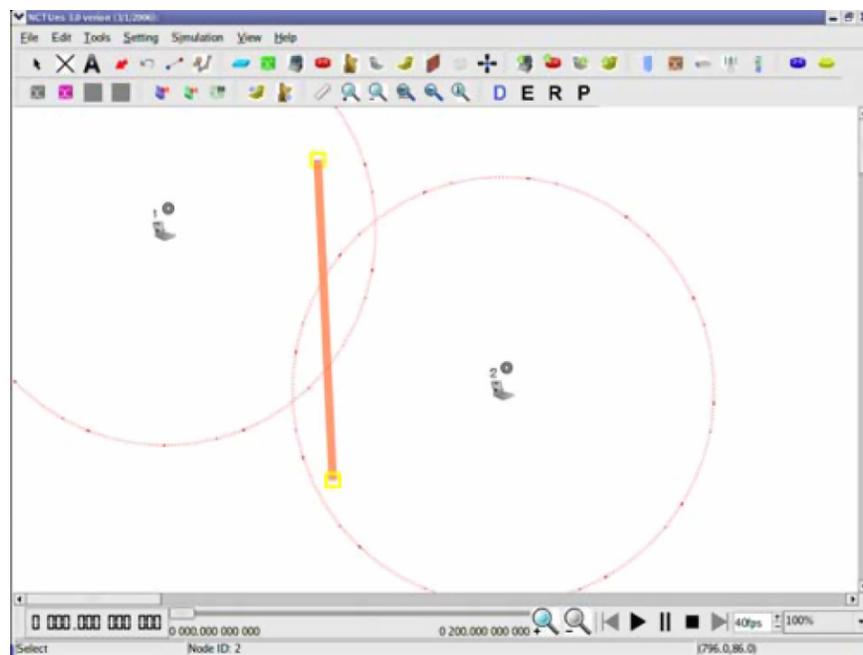


Figura A12. Topología ad-hoc con obstáculo de por medio.

A semejanza de los dispositivos como *host*, *switches*, etc., los nodos inalámbricos IEEE 802.11e 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 avanzados 802.11e tanto en las estaciones y APs que funcionen bajo este tipo de tecnología (ver Figuras A13, A14 y A15).

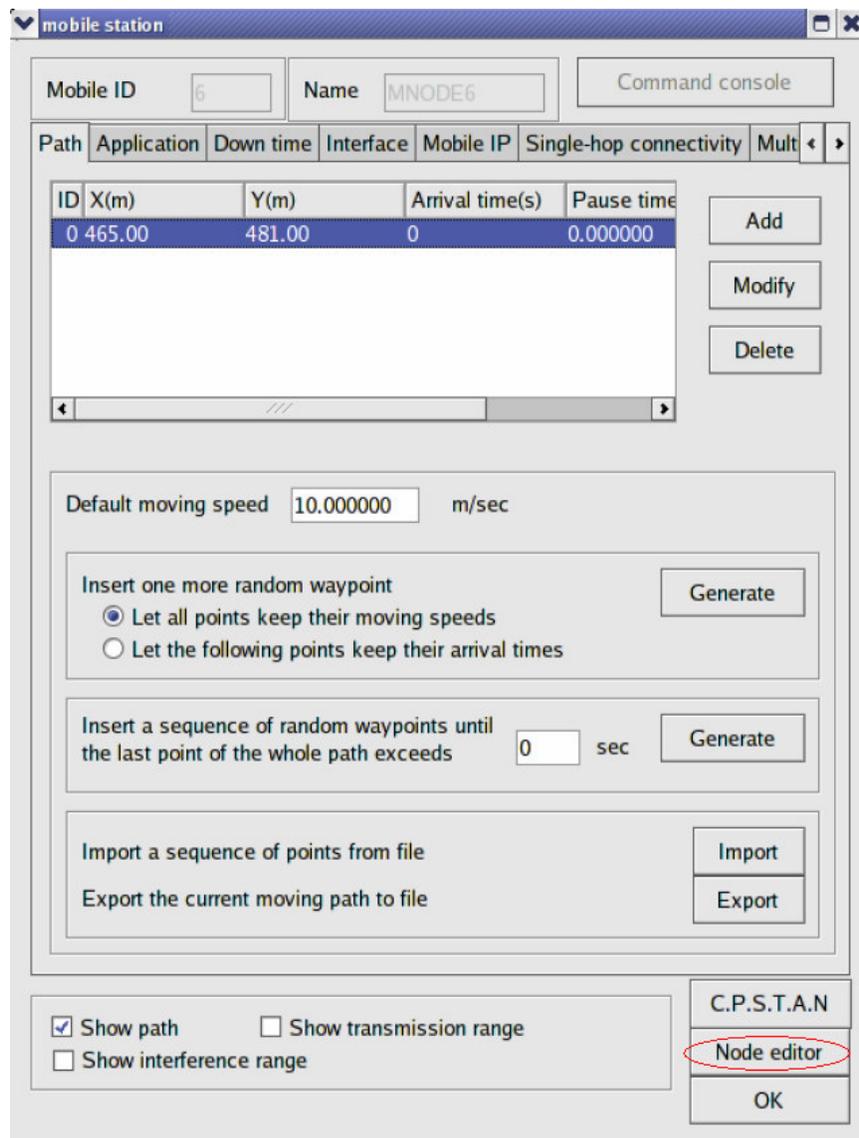


Figura A13. Ventana de atributos de una estación móvil.

Que mejor manera para observar el comportamiento de esta herramienta de simulación que mediante un ejemplo. Por ello, a continuación se expondrá paso a paso la simulación de una pequeña red inalámbrica que soporte QoS en la que se evalúe la función de acceso al medio EDCA.

La topología que se utiliza se indica en la Figura A16 en la que aparece resaltados el icono que permite las conexiones cableadas entre los dispositivos que así lo requieran y enfatiza el modo **Draw Topology** bajo el cual se realiza la inserción de nodos y conexiones como se mencionó anteriormente.

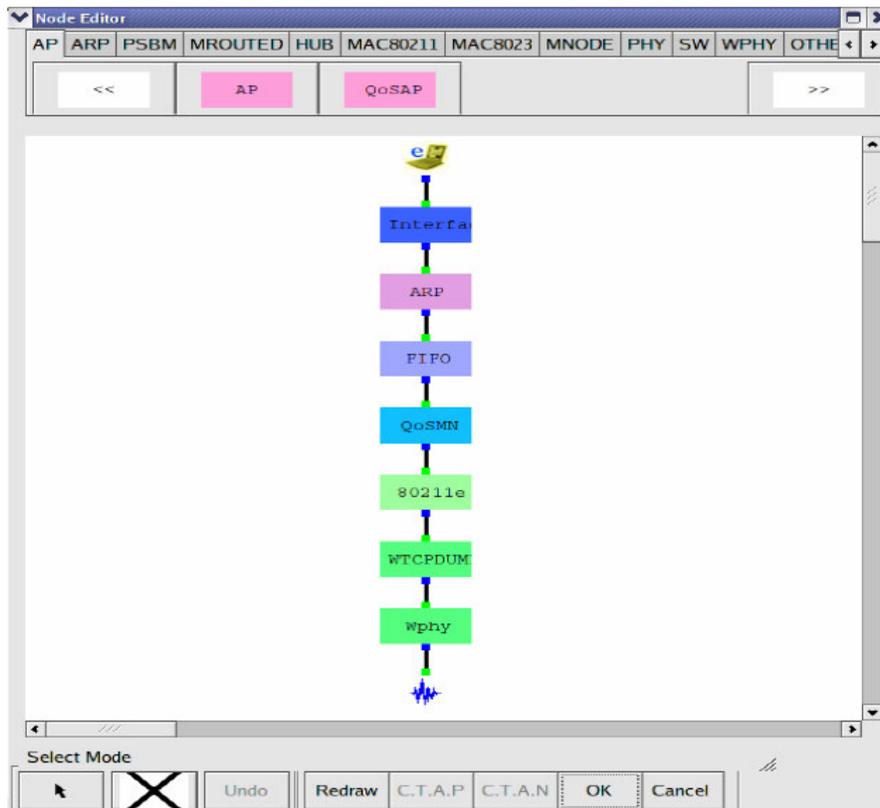


Figura A14. Ventana de editor de nodo de una estación inalámbrica 802.11e

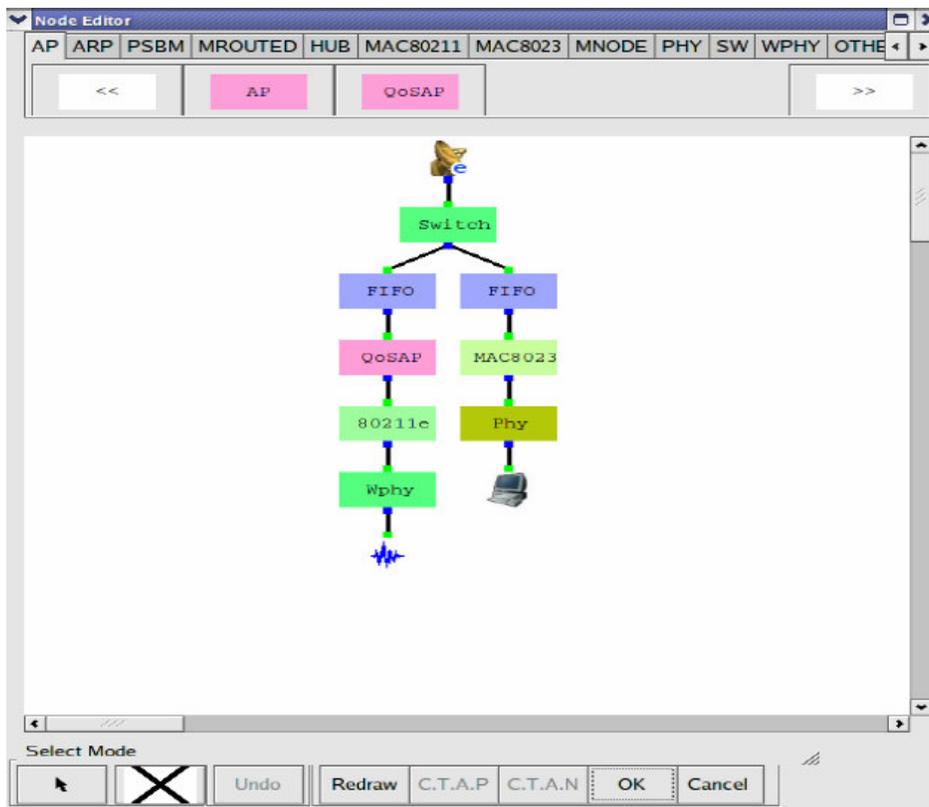


Figura A15. Ventana de editor de nodo de un AP 802.11e

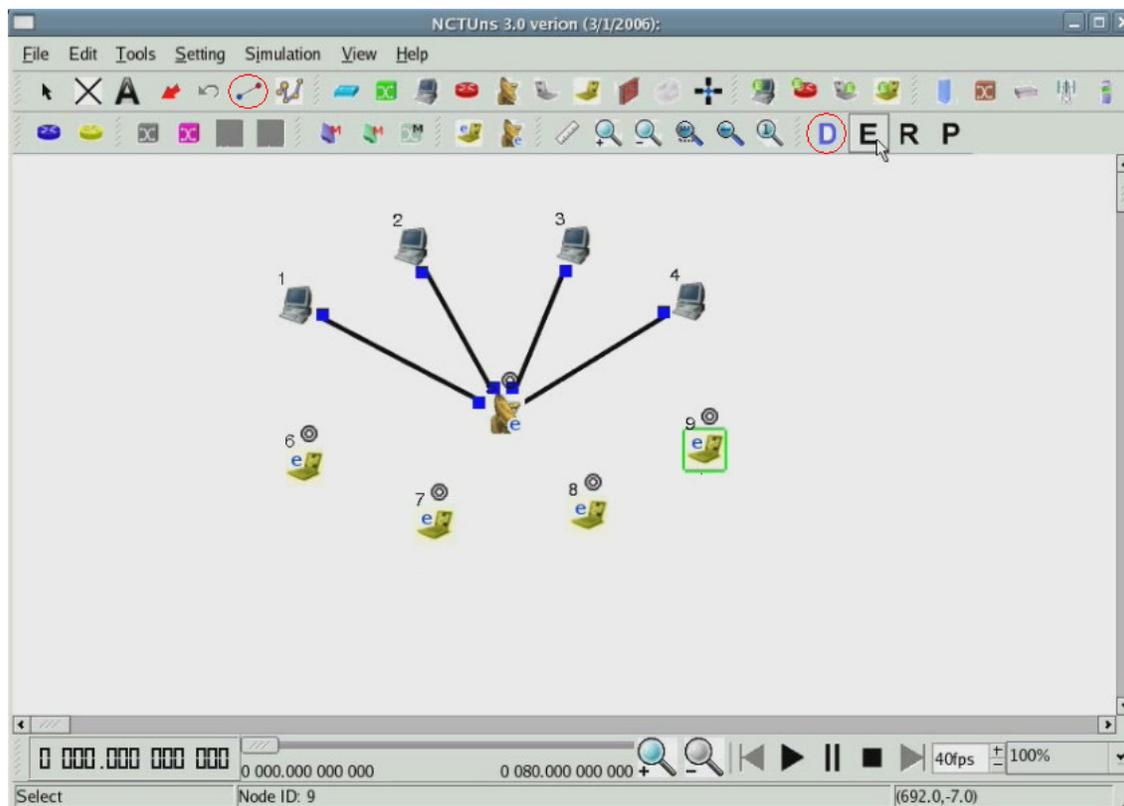


Figura A16. Topología de red.

Cada vez que se inserta un terminal inalámbrico 802.11e debe establecerse su prioridad (ver Figura A17). Para este caso los nodos 6, 7, 8 y 9 tendrán prioridad 1, 3, 5 y 7 respectivamente. Como se sabe, las prioridades de estos tipos de terminales varían de 0 a 7 representando el tipo de tráfico del cual se trata según se indica en la Tabla A1:

Tabla A1. Tipos de tráfico según 802.1p

Priority	UP (Same as 802.1D user priority)	802.1D designation	AC	Designation (informative)
Lowest ↓ Highest	1	BK	AC_BK	Background
	2	—	AC_BK	Background
	0	BE	AC_BE	Best Effort
	3	EE	AC_BE	Best Effort
	4	CL	AC_VI	Video
	5	VI	AC_VI	Video
	6	VO	AC_VO	Voice
	7	NC	AC_VO	Voice

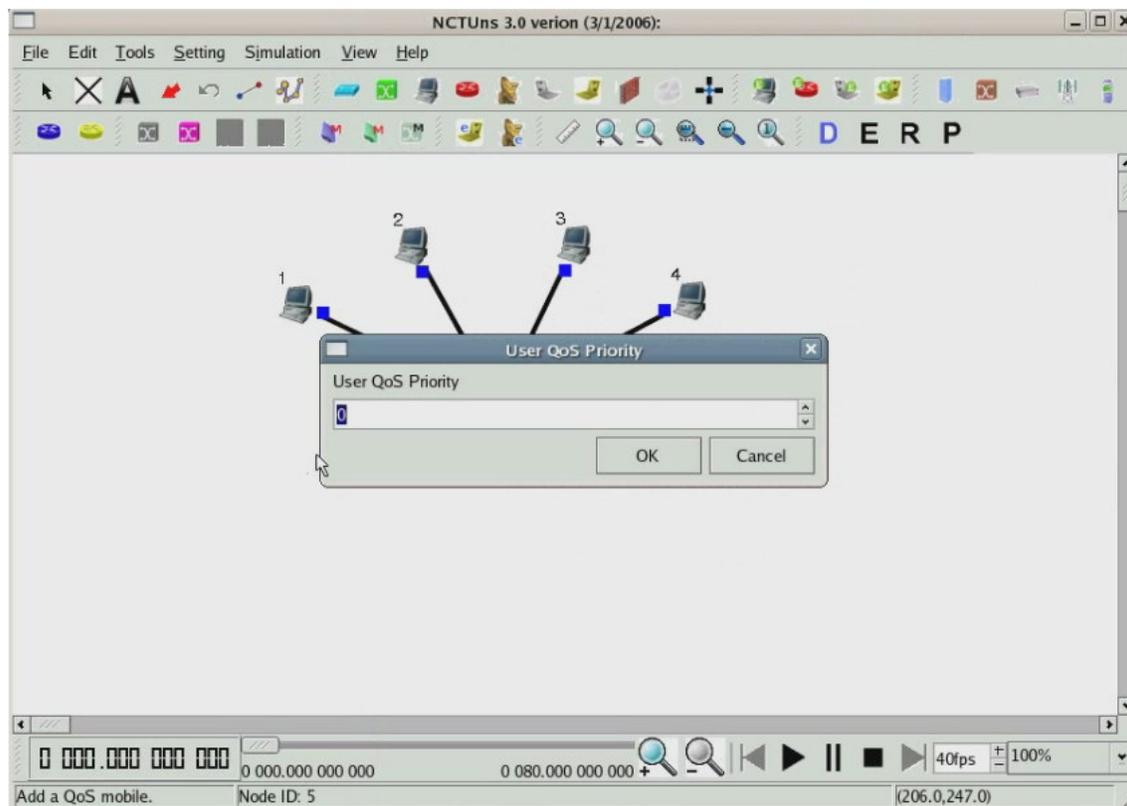


Figura A17. Prioridad QoS de terminales 802.11e.

Como dato adicional cabe mencionar que para el caso en el que el número de nodos a agregar resulte demasiado grande, este simulador, bajo su modo **Draw Topology** ofrece la opción de crear matrices de nodos, así como la ubicación de nodos en posiciones aleatorias, a través del comando² `Tools -> 802.11(e) Mobile Nodes -> Insert 802.11(e) Mobile Nodes` a lo cual aparecerá una ventana donde puede configurarse la dimensión de la matriz de nodos a ser insertada y especificarse el espaciamiento entre las estaciones que la conforman (ver Figura A18). A modo de ejemplo, para la creación de una matriz de nodos con dimensiones 3x4, con un espaciamiento entre nodos de 50 m, el resultado sería el mostrado en la Figura A19.

Volviendo al ejemplo original, para pasar al modo **Edit Property** se debe guardar el archivo de simulación tal como se aprecia en la Figura A20.

Luego, se va al menú `Edit -> Select All 802.11(e) Mobile Nodes` para seleccionar todos los nodos 802.11e y asignar direcciones IP a los mismos a través del menú `Tools -> 802.11(e) Mobile Nodes -> Generate 802.11(e) Mobile Nodes'IP and MAC Address (Need to select some Duch nodes in the Edit Property)`. Ahora pueden agregarse los comandos pertinentes de cada nodo por lo que se da doble click sobre el dispositivo de interés, dando a su paso la aparición de una nueva ventana donde podrá modificarse su configuración.

² Si se desea agregar nodos 802.11 ya sea en modo infraestructura o ad-hoc, el comando a utilizar es `Tools -> WLAN Mobile Nodes -> Insert WLAN Mobile Nodes`.

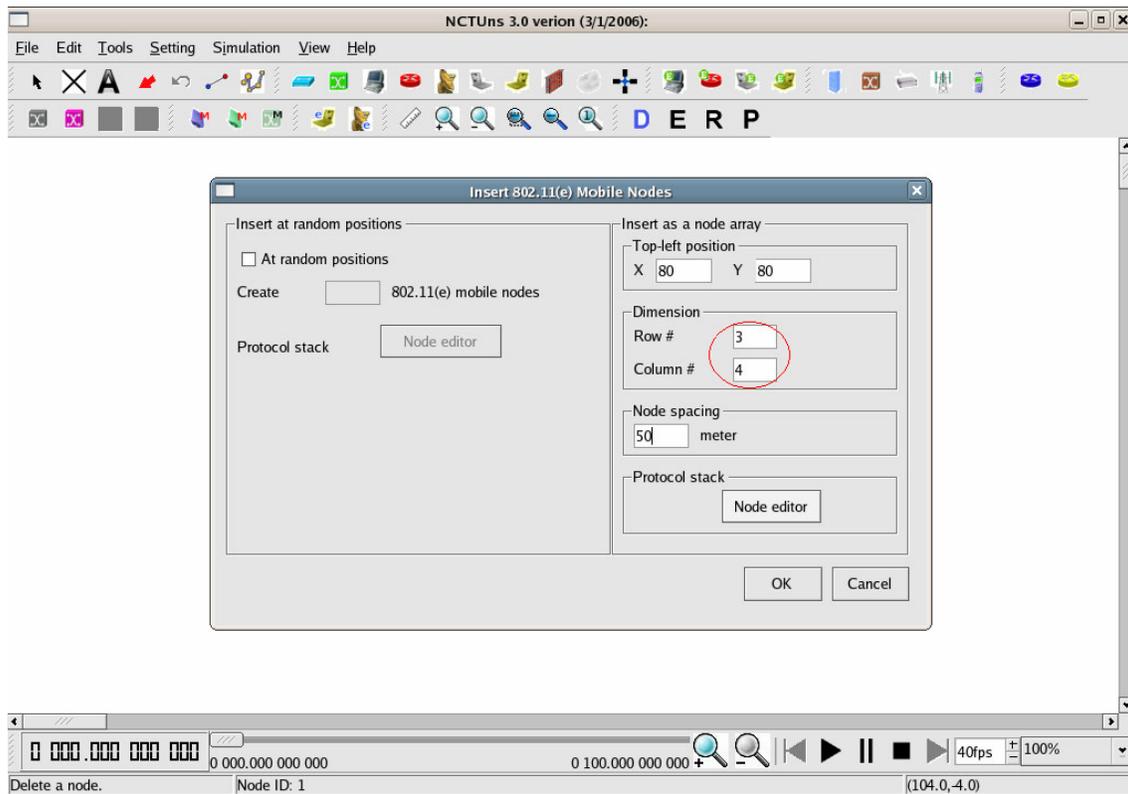


Figura A18. Inserción de múltiples nodos.

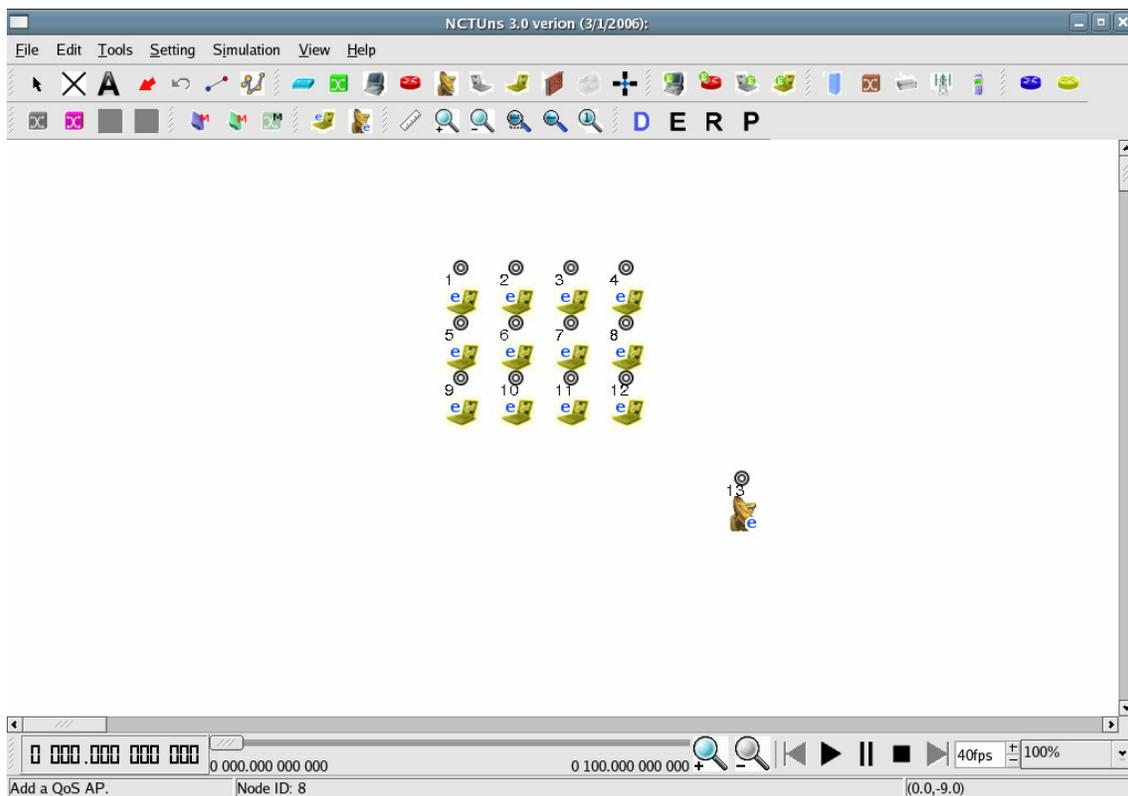


Figura A19. Matriz de 3x4 nodos.

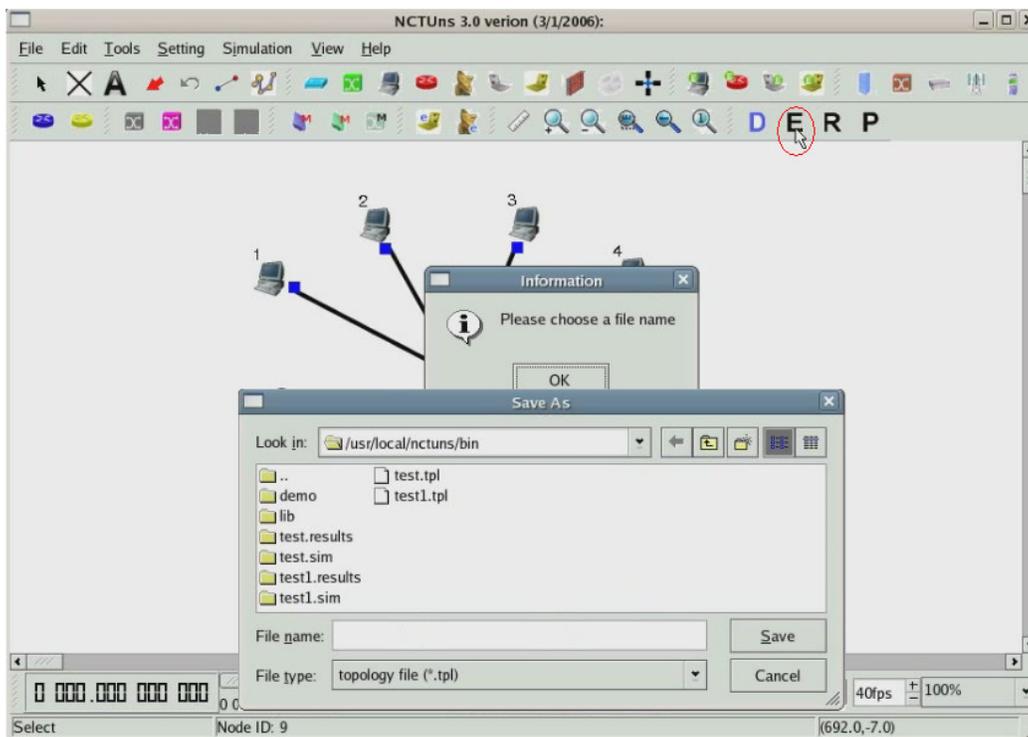


Figura A20. Modo Edit Property.

La Figura A21 señala en la pestaña de “Applications” sobre el botón “Add” el comando “*ttcp -t -u -s -p 8000 1.0.1.1*” asignado al nodo 6, donde la dirección IP al final del comando corresponde a la del nodo 1. Así mismo, se configura a los nodos 8, 9 y 10 con las direcciones IP de los nodos 2, 3 y 4 respectivamente. Con “*ttcp -t*” se indica al simulador que tales nodos tendrán la tarea de emitir paquetes hacia las direcciones IP señaladas a través de los puertos indicados con “-p # de puerto”. Para mayor información se puede remitir al botón “App. Usage” el cual contiene el significado de los comandos y su apropiado uso.

Para los nodos 1, 2, 3 y 4 se realiza de manera similar, sólo que estos serán configurados de modo que reciban la información que fue enviada hacia ellos (“*ttcp -r*”) y por el número de puerto asignado (“-p # de puerto”), tal como se muestra en la Figura A22.

Luego se pasa la modo de **Run Simulation** por lo quede irse al menú *Simulation* -> *Run* (ver Figura A23).

Finalmente, en el modo **Play Back** puede observarse la animación creada como resultado de la simulación donde podrá observarse de hecho que aquel nodo con categoría más alta ganará el acceso al medio, tal como lo estipula EDCA, mientras que las demás estaciones se verán obligadas a retrasar sus transmisiones (ver Figura A24).

Para descargar éste, y otros simuladores como NS-2 y Opnet IT Academic Guru Edition, remítase a [2], [3] y [4].

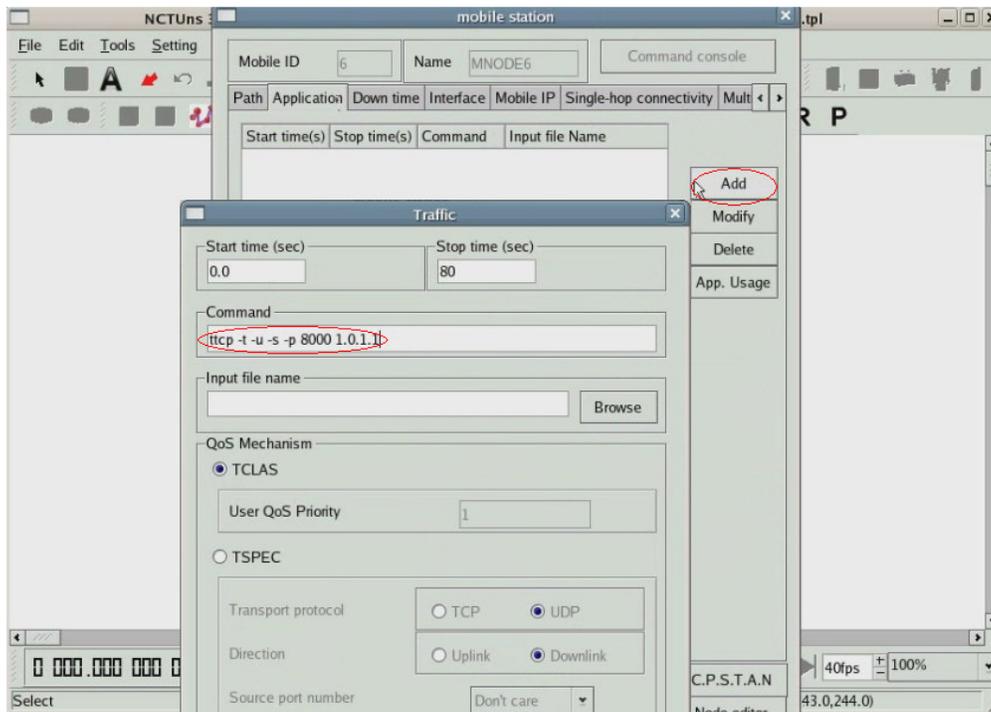


Figura A21. Inserción de comandos en los nodos inalámbricos de red.

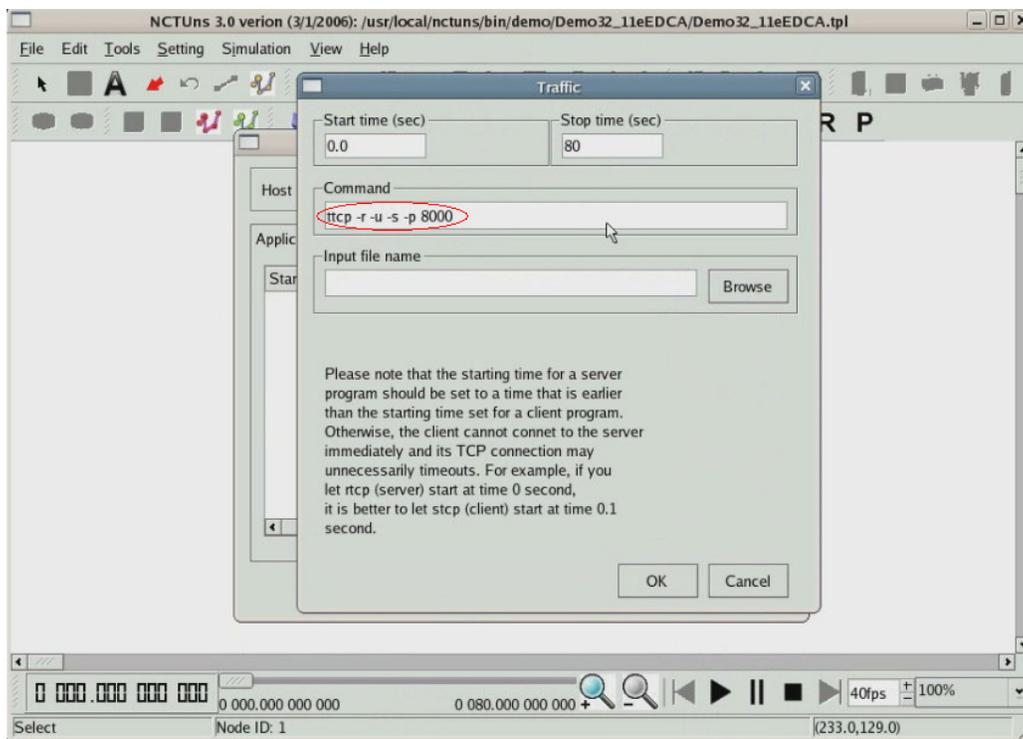


Figura A22. Inserción de comandos en los nodos cableados de red.

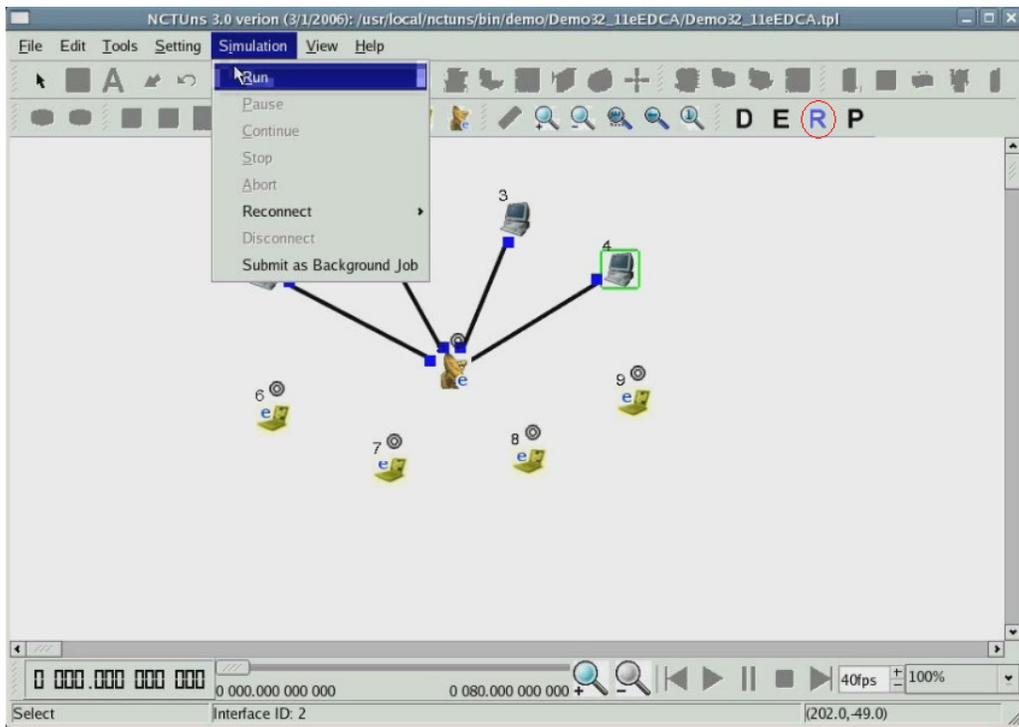


Figura A23. Simulación de la topología de red.

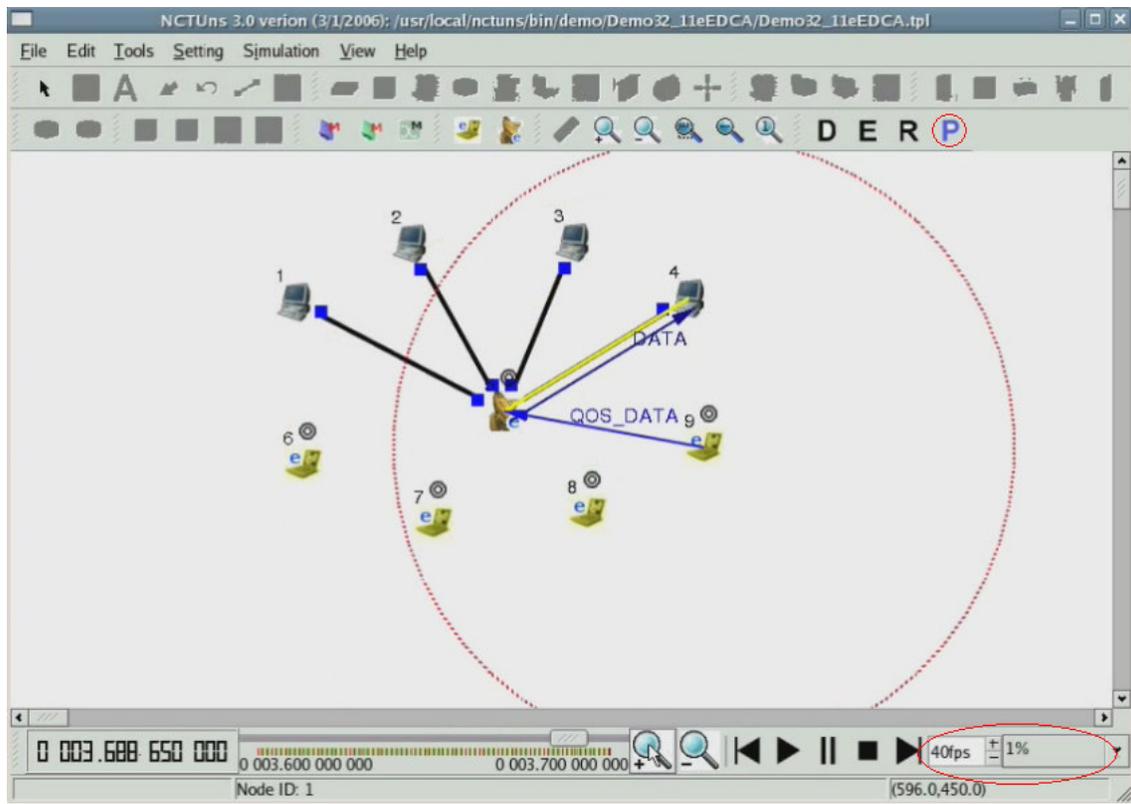


Figura A24. Resultados vistos mediante el Play Back de NCTUns 3.0

REFERENCIAS

- [1]. Wang, Shie-Yuang "The GUI User Manual for the NCTUns 3.0 Network Simulator and Emulator ". 2006.
- [2]. <http://nsl10.csie.nctu.edu.tw/products/nctuns/download/download.php>
- [3]. http://sourceforge.net/project/showfiles.php?group_id=149743&package_id=169584&release_id=450422
- [4]. http://enterprise37.opnet.com/4dcgi/DOWNLOAD_HOME