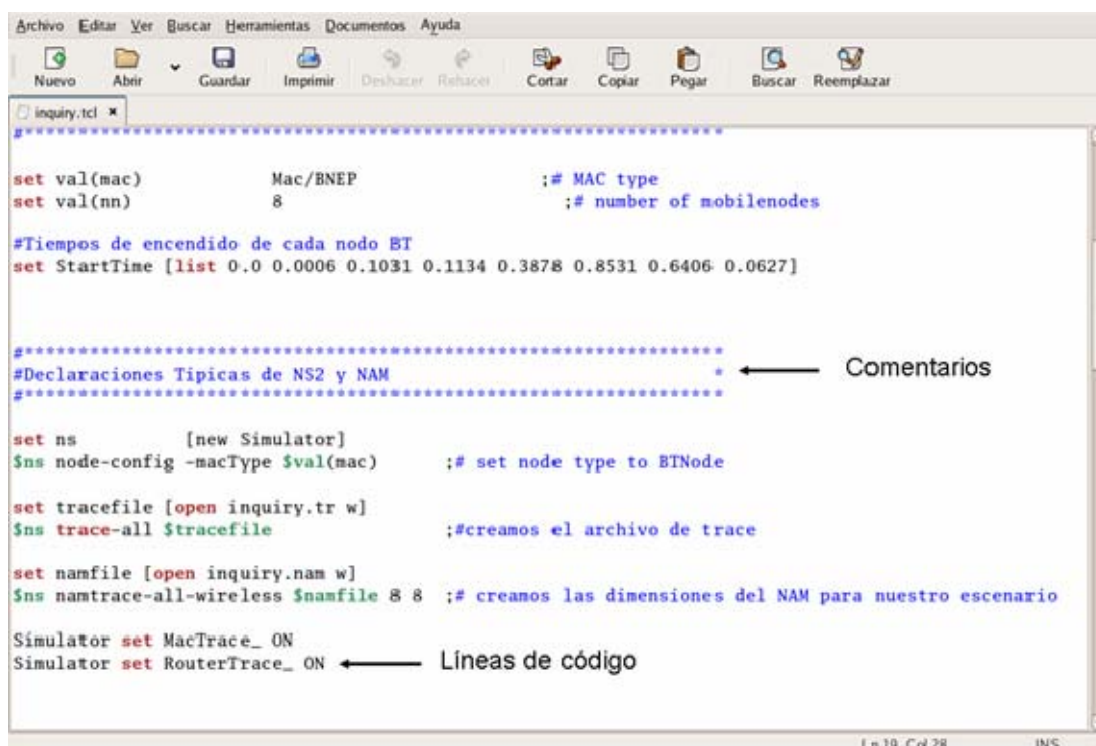


APÉNDICE A.

Guías para comenzar a trabajar con NS2, UCBT y NAM.

NS2 es un simulador de redes de código abierto en el que se pueden realizar diferentes escenarios de redes tanto alámbricas como inalámbricas. Este simulador utiliza comandos que son escritos en un archivo llamado script, donde se especifican todas las variables, nodos, dispositivos, conexiones, tipos de tráfico, entre otros.

La siguiente figura es un ejemplo de este archivo:



```
Archivo  Editar  Ver  Buscar  Herramientas  Documentos  Ayuda
Nuevo  Abrir  Guardar  Imprimir  Deshacer  Rehacer  Cortar  Copiar  Pegar  Buscar  Reemplazar
inquiry.tcl
#=====
set val(mac)          Mac/BNEP          ;# MAC type
set val(nn)           8                  ;# number of mobilenodes

#Tiempos de encendido de cada nodo BT
set StartTime [list 0.0 0.0006 0.1031 0.1134 0.3878 0.8531 0.6406 0.0627]

#=====
#Declaraciones Tipicas de NS2 y NAM * ← Comentarios
#=====

set ns                [new Simulator]
$ns node-config -macType $val(mac)      ;# set node type to BTNode

set tracefile [open inquiry.tr w]
$ns trace-all $tracefile              ;# creamos el archivo de trace

set namfile [open inquiry.nam w]
$ns namtrace-all-wireless $namfile 8 8 ;# creamos las dimensiones del NAM para nuestro escenario

Simulator set MacTrace_ ON
Simulator set RouterTrace_ ON ← Líneas de código
```

Script para simulación en NS2

La creación de un script es sencilla y para ello solo necesitamos escribir las sentencias en lenguaje tcl y siguiendo la estructura adecuada en cualquier editor de texto. Una de las ventajas de usar el lenguaje tcl es que no se necesita definir el tipo de variable que se utilizaran, es por ello que si revisamos un archivo tcl no encontraremos tipos de datos entero, reales, etc. Lo único que debemos definir son los agentes que se utilizaran en la simulación. La declaración de los agentes y la estructura que se debe seguir al momento de la creación de un script se detallan a continuación:

set variable_simulador [new Simulator]

esta sentencia realiza la declaración de la variable que identificará la instancia del simulador o también conocida como variable del simulador. Hay que tener en cuenta que si se utilizan constantes dentro del script, éstas deben ser declaradas con sus respectivos valores justo antes de especificar esta sentencia. Lo podemos notar en la figura anterior.

set variable_trace [open nombre_archivo.tr w]

\$variable_simulador trace-all \$variable_trace

estas líneas crean el archivo de trazas disponible para escritura, además le indican al simulador que todas las trazas generadas se deben almacenar en el archivo que se creó anteriormente.

A continuación se procede a la creación de los nodos con su respectivas características, dependiendo del tipo de librería que se esta utilizando, seguido por la definición de los enlaces, tipo de tráfico y declaración de colas en caso de necesitarse en la simulación. Luego procedemos a definir la organización de los eventos. La explicación de esta sección se detallará más adelante en esta guía.

Para finalizar todo script debe realizarse un procedimiento en el cual se detallarán ciertas acciones que se deben ejecutar antes de salir del simulador. Este procedimiento se detalla a continuación:

```
proc finish {} {  
    global node variable_simulador variable_trace variable_nam  
    $ variable_simulador flush-trace  
    $node(0) print-all-stat  
    close $ variable_nam  
    close $ variable_trace  
    exec nam nombre_archivo.nam &  
    exit 0  
}  
$ns run
```

Procedimiento de finalización

En este procedimiento se ejecutan varias acciones, entre las principales están el cierre de los archivos *.nam* y *.tr* que se abrieron al inicio de la simulación, para ello al inicio del procedimiento se definen las variables que representan a estos archivos como globales.

También podemos realizar ciertas acciones especiales para ciertos nodos así como también escribir sentencias que se ejecutarán en la línea de comando del entorno donde se ejecute la simulación. Para este caso es una ventana de terminal muy parecida a la ventana de DOS en los sistemas operativos Windows, esto se logra utilizando el comando *exec*. Por último especificamos que al terminar de procesar el contenido del archivo tcl, la ventana de terminal permanezca abierta por el comando *exit 0*.

Una vez que tenemos listo nuestro script procedemos a la ejecución de este archivo. Para ejecutar una simulación, primero debemos abrir una ventana de terminal donde escribimos la siguiente sentencia:

```
ns nombreachivo.tcl >> nombreachivo.out
```

```

root@localhost:usr/local/ns-allinone-2.29/ns-2.29/bluetooth/test/tesis/ejemplos
Archivo Editar Ver Terminal Solapas Ayuda
[root@localhost ejemplos]# ls
3f_mod.tcl   prueba_inq.tcl   scat-form-law.tcl   simulacion2.tcl   Test_Inq.tcl
Inqmod.tcl   prueba.tcl       sim2.tcl            simulacion3.tcl   transfer.tcl
Inq.tcl      scat2pico(1).tcl sim3.tcl            simulacion4.tcl
piconet.tcl  scat2pico.tcl    simulacion1.tcl     simulacion5.tcl
[root@localhost ejemplos]# ns simulacion1.tcl >> simulacion1.out

```

Pantalla de terminal con sentencias para ejecutar simulación.

Para lograr esto el archivo que contiene la simulación que se va a ejecutar, en este caso identificado como *nombearchivo*, debe estar contenido en la misma carpeta que indica la ruta en la pantalla del terminal.

```

root@localhost:usr/local/ns-allinone-2.29/ns-2.29/bluetooth/test/tesis/ejemplos
Archivo Editar Ver Terminal Solapas Ayuda
[root@localhost ejemplos]# ls
3f_mod.tcl   prueba_inq.tcl   scat-form-law.tcl   simulacion2.tcl   Test_Inq.tcl
Inqmod.tcl   prueba.tcl       sim2.tcl            simulacion3.tcl   transfer.tcl
Inq.tcl      scat2pico(1).tcl sim3.tcl            simulacion4.tcl
piconet.tcl  scat2pico.tcl    simulacion1.tcl     simulacion5.tcl
[root@localhost ejemplos]# ns simulacion1.tcl >> simulacion1.out
Inq time: 0.003412 5 num: 1 ave: 0.003412 5
Inq time: 0.006224 10 num: 2 ave: 0.003112 5
0 0.00 (0/0.00) 0.000 (0.00/723200.00) d: 0.00 0.00 l: 1.00 1.00 1.00
1 0.00 (0/0.00) 0.000 (0.00/723200.00) d: 0.00 0.00 l: 1.00 1.00 1.00
2 0.00 (0/0.00) 0.000 (0.00/723200.00) d: 0.00 0.00 l: 1.00 1.00 1.00
3 0.00 (0/0.00) 0.000 (0.00/723200.00) d: 0.00 0.00 l: 1.00 1.00 1.00
4 0.00 (0/0.00) 0.000 (0.00/723200.00) d: 0.00 0.00 l: 1.00 1.00 1.00
[root@localhost ejemplos]# █

```

Pantalla de terminal en un instante después de terminar la simulación

Una vez que el simulador termina de procesar el script, pasamos a la revisión de los resultados, estos se guardan en archivos con extensión *.tr* denominados archivos de traza y *.out* denominados archivos de salida. La siguiente figura es un ejemplo del archivo de trazas, el archivo *sim1.tr*:

```

sim1.tr (/usr/local/ns-allinone-2.29/ns-2.29/ucbt-0.99.1/test/tesis/ejemplos) - gedit
Archivo Editar Ver Buscar Herramientas Documentos Ayuda
Nuevo Abrir Guardar Imprimir Deshacer Rehacer Cortar Copiar Pegar Buscar Reemplazar
sim1.tr x
s 0.100233500 _0_ MAC --- 0 bb 8 [0:0--1:0 ID 0 10390323 06 0 0 068 0:0]
s 0.101171000 _0_ MAC --- 0 bb 8 [0:0--1:0 ID 0 10390323 53 0 0 068 1:0]
s 0.101483500 _0_ MAC --- 0 bb 8 [0:0--1:0 ID 0 10390323 69 0 0 068 2:0]
s 0.102421000 _0_ MAC --- 0 bb 8 [0:0--1:0 ID 0 10390323 37 0 0 068 3:0]
s 0.102733500 _0_ MAC --- 0 bb 8 [0:0--1:0 ID 0 10390323 08 0 0 068 5:0]
s 0.103046000 _2_ MAC --- 0 bb 45 [2:0-0:0 FH 0 10390323 36 0 0 366 4:0]
s 0.103671000 _0_ MAC --- 0 bb 8 [0:0--1:0 ID 0 10390323 55 0 0 068 6:0]
s 0.103983500 _0_ MAC --- 0 bb 8 [0:0--1:0 ID 0 10390323 71 0 0 068 7:0]
s 0.104921000 _0_ MAC --- 0 bb 8 [0:0--1:0 ID 0 10390323 39 0 0 068 8:0]
s 0.105233500 _0_ MAC --- 0 bb 8 [0:0--1:0 ID 0 10390323 10 0 0 068 9:0]
s 0.105858500 _4_ MAC --- 0 bb 45 [4:0-0:0 FH 0 10390323 74 0 0 366 10:0]

```

Archivo sim1.tr visualizado en el editor de texto.

Estos archivos de trazas siguen un formato específico, para poder ser analizados se necesita conocer el significado de cada uno de los parámetros que se encuentran en las

líneas de este archivo. El formato de estas trazas comienza con uno de cuatro caracteres. Este es seguido por una bandera ó pares de valores los cuales son detallados a continuación.

Evento	Abreviatura	Tipo	Valor
Eventos Normales	r: Recibe d: Caído e: Error +: En cola -: Sale de cola	%g %d %d %s %d %s %d %d.%d %d.%d %d %d	
		doble	tiempo
		entero	Nodo fuente
		entero	Nodo destino
		caracteres	Nombre del paquete
		entero	Tamaño del paquete
		caracteres	Banderas
		entero	ID para el flujo
		entero	Dirección de la fuente
		entero	Dirección del destino
		entero	Numero de secuencia
		entero	ID único del paquete

Formatos de trazas normales.

La tabla que lista la información adicional de trazas inalámbricas no tiene la columna abreviatura, esta información es añadida al final de las trazas inalámbricas regulares.

Evento	Tipo	Valor
Trazos TCP	%d 0x%x %d %d	
	entero	Número de Ack
	hexadecimal	Banderas
	entero	Longitud de cabeceras
	entero	Longitud de la dirección del puerto

Formato de trazas adicionales.

Evento	Abreviatura	Bandera	Tipo	Valor
Eventos inalámbricos	s: Enviado r: Recibido d: Caído f: Reexpedir	-t	doble	tiempo (* para ajustes globales)
		-Ni	entero	ID del nodo
		-Nx	doble	Coordenada X del nodo
		-Ny	doble	Coordenada Y del nodo
		-Nz	doble	Coordenada Z del nodo
		-Ne	doble	Nivel de energía del nodo
		-NI	caracteres	Nivel de trazo de la red (AGT, RTR, MAC, bb, etc.)
		-Nw	caracteres	Razón de la caída
		-Hs	entero	ID del nodo fuente del salto
		-Hd	entero	ID del nodo destino del salto , -1, -2

		-Ma	hexadecimal	Duración
		-Ms	hexadecimal	Dirección Ethernet de la fuente
		-Md	hexadecimal	Dirección Ethernet del destino
		-Mt	hexadecimal	Tipo de Ethernet
		-P	caracteres	Tipo de paquete (arp, dsr, imep, tora, etc.)
		-Pn	caracteres	Tipo de paquete (cbr, tcp)

Formato de trazas inalámbricas.

Otra de las fuentes donde podemos obtener información es el archivo de salida de la simulación. La siguiente figura muestra un ejemplo de este archivo de salida, el archivo *simulacion1.out*:

```

1 start page scan. clkn:4897236
1 : not connected.
2 start page scan. clkn:486945
2 : not connected.
3 start page scan. clkn:829007
3 : not connected.
4 start page scan. clkn:15580335
4 : not connected.
2 start page scan. clkn:486947
2 : not connected.
2 at 0.000174 CHANGE ST STANDBY -> INQUIRY_SCAN clkn:486948 clk:0
3 start page scan. clkn:829007
3 : not connected.
3 at 0.000400 CHANGE ST STANDBY -> INQUIRY_SCAN clkn:826008 clk:0
4 start page scan. clkn:15580335
4 : not connected.
4 at 0.000622 CHANGE ST STANDBY -> INQUIRY_SCAN clkn:15580336 clk:0
1 start page scan. clkn:4897239
1 : not connected.
1 at 0.000913 CHANGE ST STANDBY -> INQUIRY_SCAN clkn:4867240 clk:0
0 start page scan. clkn:13727063
0 : not connected.
0 at 0.001171 CHANGE ST STANDBY -> INQUIRY_SCAN clkn:13727064 clk:0
0 at 0.012109 CHANGE ST INQUIRY_SCAN -> STANDBY clkn:13727099 clk:0
0 at 0.012421 CHANGE ST STANDBY -> PAGE_SCAN clkn:13727100 clk:0
0 at 0.023359 CHANGE ST PAGE_SCAN -> STANDBY clkn:13727135 clk:0
0 at 0.100000 CHANGE ST STANDBY -> INQUIRY clkn:13727380 clk:0
t 0 INQ 0:0-:0 ID 0 10390323 06 0 0 c 0 068 0.100233 0:0 Q 13727381 0

```

Archivo simulacion1.out visualizado en el editor de texto.

Este archivo tiene un formato específico según la librería que se esté utilizando para simular los escenarios, en este caso la simulación corresponde a la de una red Bluetooth, por lo cual este archivo tiene un formato acorde con la librería UCBT.

Las trazas más importantes son las que hacen referencias a los eventos de transmisión y recepción. Estos son identificados por medio de la letra con la que empieza cada traza para el caso de transmisión “t” y para recepción “r”. Los formatos de estos eventos se detallarán en la siguiente tabla:

Evento	Tipo	Bandera	Valor
Trazas archivo de salida	%c %d %s %d:%d-%d:%d		
	carácter	stat	Estado de transmisión
	entero	ad	Identificador del nodo
	caracteres	st	Estado del nodo
	entero	sender	Identificador del emisor
	entero	srcTxSlot	Slot en emisión

entero	receiver	Identificador del receptor
entero	dstTxSlot	Slot en recepción
%s %d %d %d %.02d %d %d c:%d %d %.03d %f %d:%d %s %d %d		
caracteres	packet_type_str_short(type)	Tipo del paquete
entero	lt_addr_	Identificador del nodo
entero	ac	
entero	fs_	Secuencia de salto de frecuencia
entero	arqn	
entero	seqn	
entero	transmitCount	
entero	size	Tamaño de paquete
real	ts()	Tiempo
entero	pid	Tamaño de la carga útil
entero	seqno	Secuencia
caracteres	comment()	Comentario
doble	clk	
entero	extinfo	

Formato del archivo de salida.

El NAM es la herramienta que utilizaremos para la visualización de las simulaciones que se ejecuten con el NS2, siempre y cuando la librería que se utilice tenga compatibilidad para crear el archivo que es interpretado por el NAM, archivos *.nam*, los cuales poseen todos los detalles de la simulación que han sido compiladas por el simulador.

A continuación tenemos un ejemplo de un archivo *.nam*.

```

sim1.nam *
V -t * -v 1.0a5 -a 0
W -t * -x 670 -y 670
A -t * -n 1 -p 0 -o 0xffffffff -c 31 -a 1
A -t * -h 1 -m 2147483647 -s 0
v -t 0.000172999993083 -e sim_annotation 0.000172999993083 0 2 en 0.0001740000000000 STANDBY ->
INQUIRY_SCAN
n -t 0.000172999993083 -s 2 -S DLABEL -l INQUIRY_SCAN -L STANDBY
v -t 0.000399000011384 -e sim_annotation 0.000399000011384 1 3 en 0.0004000000000000 STANDBY ->
INQUIRY_SCAN
n -t 0.000399000011384 -s 3 -S DLABEL -l INQUIRY_SCAN -L STANDBY
v -t 0.000621000013780 -e sim_annotation 0.000621000013780 2 4 en 0.0006220000000000 STANDBY ->
INQUIRY_SCAN
n -t 0.000621000013780 -s 4 -S DLABEL -l INQUIRY_SCAN -L STANDBY
v -t 0.000912000017706 -e sim_annotation 0.000912000017706 3 1 en 0.0009130000000000 STANDBY ->
INQUIRY_SCAN
n -t 0.000912000017706 -s 1 -S DLABEL -l INQUIRY_SCAN -L STANDBY
v -t 0.0011699999944977 -e sim_annotation 0.0011699999944977 4 0 en 0.0011710000000000 STANDBY ->
INQUIRY_SCAN
n -t 0.0011699999944977 -s 0 -S DLABEL -l INQUIRY_SCAN -L STANDBY
v -t 0.012108599767089 -e sim_annotation 0.012108599767089 5 0 en 0.0121085000000000 INQUIRY_SCAN -
> STANDBY
n -t 0.012108599767089 -s 0 -S DLABEL -l STANDBY -L INQUIRY_SCAN
v -t 0.012421100400388 -e sim_annotation 0.012421100400388 6 0 en 0.0124210000000000 STANDBY ->
PAGE_SCAN
n -t 0.012421100400388 -s 0 -S DLABEL -l PAGE_SCAN -L STANDBY
v -t 0.023358600214124 -e sim_annotation 0.023358600214124 7 0 en 0.0233585000000000 PAGE_SCAN ->
STANDBY
n -t 0.023358600214124 -s 0 -S DLABEL -l STANDBY -L PAGE_SCAN

```

Archivo sim1.nam visualizado en el editor de texto.

El archivo *.nam* está conformado por un conjunto de líneas las cuales poseen parámetros, el formato general para un traza del NAM es una letra seguida por uno o más pares de valores. En la siguiente tabla se detallan estos parámetros, hay que tener en cuenta que no todos estos serán usados siempre:

Evento	Abrev.	Bandera	Tipo	Valor
Nodo	n	-t	tiempo	Tiempo
		-s	entero	Nodo ID
		-u	doble	Velocidad X
		-U	doble	Velocidad X
		-V	doble	Velocidad Y
		-v	Forma	Forma (circle, box, hexagon)
		-c	color	Color
		-z	doble	Tamaño de nodo
		-a	entero	Dirección
		-x	doble	Posición X
		-y	doble	Posición Y
		-Z	doble	Posición Z (0)
		-i	color	Color de la etiqueta
		-b	caracteres	Etiqueta
		-l	caracteres	Etiqueta
		-o	color	Color previo
		-S	caracteres	Estado (UP, DOWN, COLOR)
		-L	caracteres	Etiqueta previa
		-p	caracteres	Posición de la etiqueta
		-P	caracteres	Posición previa de la etiqueta
		-i	color	Color del interior de la etiqueta
		-I	color	Color previo del interior de la etiqueta
		-e	color	Color de la etiqueta
		-E	color	Color previo de la etiqueta
-T	doble	Duración del movimiento		
-w	bandera	Nodo inalámbrico		
Enlace	l	-t	tiempo	Tiempo
		-s	entero	ID de la fuente
		-d	entero	ID del destino
		-r	doble	Tasa de transmisión
		-D	doble	Retardo
		-h	doble	Longitud
		-O	orientación	orientación
		-b	caracteres	Etiqueta
		-c	color	Color
		-o	color	Color previo
		-S	caracteres	Estado (UP, DOWN)
		-l	caracteres	Etiqueta
		-L	caracteres	Etiqueta previa
		-e	color	Color de la etiqueta
		-E	color	Color previo de la etiqueta
Paquetes	h: Salto r: Recibe d: Línea caída	-t	tiempo	Tiempo
		-s	entero	ID de la fuente
		-d	entero	ID del destino

	+: En cola -: Sale de cola	-e	entero	Alcance
		-a	entero	ID color atributo del paquete
		-i	entero	ID
		-l	entero	Energía
		-c	caracteres	Conversación
		-x	comentario	Comentario
		-p	caracteres	Tipo de paquete
		-k	caracteres	Tipo de paquete
		-y	comentario	
		-S	entero	
		-m	entero	
		-f	entero	
Sesión	E: En cola D: Sale de cola P: Caída	-t	tiempo	Tiempo
		-s	entero	ID de la fuente
		-d	entero	ID del destino
		-e	entero	Alcance
		-a	entero	Atributo
		-i	entero	ID
		-l	entero	Energía
		-c	caracteres	Conversación
		-x	comentario	Comentario
		-p	caracteres	Tipo de paquete
		-k	caracteres	Tipo de paquete
		Agente	a	-t
-s	entero			ID de la fuente
-d	entero			ID del destino
-x	bandera			Quitar Agente
-n	caracteres			Nombre de Agente
Característica	f	-t	tiempo	Tiempo
		-s	entero	ID de la fuente
		-d	entero	ID del destino
		-x	bandera	Quitar característica
		-T	caracter	Tipo
		-n	caracteres	Nombre
		-a	caracteres	Agente
		-v	caracteres	Valor
-o	caracteres	Valor previo		
Grupo	G	-t	tiempo	Tiempo
		-n	caracteres	Nombre
		-i	entero	ID del nodo
		-a	entero	ID del grupo
		-x	bandera	Quitar de grupo
Enlaces LAN	L	-t	tiempo	Tiempo
		-s	entero	ID de la fuente
		-d	entero	ID del destino
		-o	orientación	orientación
		-O	orientación	orientación
Marcas a nodos	m	-t	tiempo	Tiempo
		-n	caracteres	Nombre
		-s	entero	ID del nodo
		-c	caracteres	Color

		-h	caracteres	Forma (circle, square, hexagon)
		-X	bandera	Quitar marca
Evento de enrutamiento	R	-t	tiempo	Tiempo
		-s	entero	ID de la fuente
		-d	entero	ID del destino
		-g	entero	Grupo Multicast
		-p	paquete fuente	ID del paquete fuente ó *
		-n	bandera	Caché negativo
		-x	bandera	Ruta expirada
		-T	doble	Tiempo para expirar
		-m	caracteres	Modo (IIF ó OIF)
		Rango inalámbrico	W	-t
-x	entero			X
-y	entero			Y
Ejecutar expresión tcl	v	-t	tiempo	Tiempo
		-e	expresión tcl	expresión tcl
Jerarquía de configuración de direcciones -- solo en inicialización	A	-t	tiempo	Tiempo
		-n	entero	Jerarquía
		-p	entero	Puerto de turno
		-o	hexadecimal	Mascara del puerto
		-c	entero	Multicast de turno
		-a	entero	Mascara Multicast
		-h	entero	Jerarquía
		-m	entero	Nodo de turno
configuración de la tabla de colores -- solo inicialización	c	-t	tiempo	Tiempo
		-i	entero	ID
		-n	caracteres	Color
Creación de cola de paquetes -- solo inicialización	q	-t	tiempo	Tiempo
		-s	entero	ID de la fuente
		-d	entero	ID del destino
		-a	orientación	orientación
Esquema LAN	X	-t	tiempo	Tiempo
		-n	caracteres	Nombre
		-r	doble	Tasa
		-D	doble	Retardo
		-o	orientación	orientación
		-O	orientación	orientación

Formatos de los trazas del NAM.

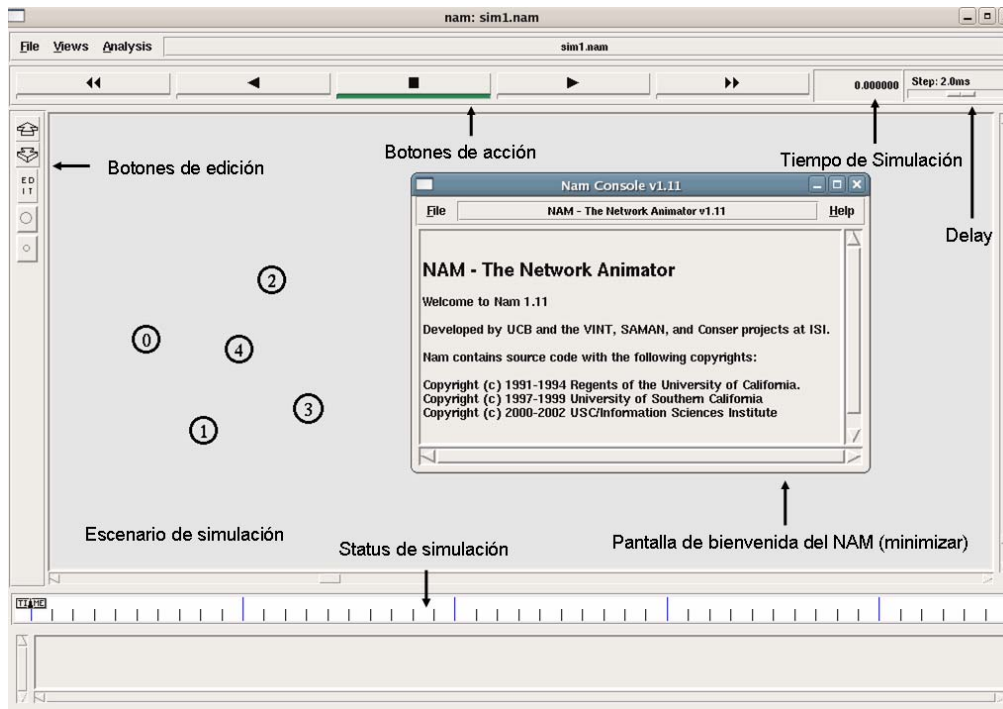
Para eventos en paquetes (entradas que empiezan con “h”, “r”, “d”, “+”, or “-“), los campos después de “-x” tienen el siguiente formato:

Evento	Tipo	Valor
Traza de nodo		{%s.%s %s.%s %d %s %s }
	caracteres	Dirección del nodo fuente
	caracteres	Puerto del nodo fuente
	caracteres	Dirección del nodo destino
	caracteres	Puerto del nodo destino
	entero	Numero de secuencia

	caracteres	Banderas
	caracteres	Nombre del paquete

Formatos de las trazas adicionales del NAM.

Toda la información contenida en el archivo *.nam* se traduce en la ejecución de una simulación de forma gráfica en la ventana del NAM, en intervalos de tiempos ajustables y con opciones novedosas parecidas a la de un reproductor para poder ver y manipular dicha simulación. En la siguiente figura podemos apreciar la pantalla principal del NAM para una simulación en particular en la cual se detallan las partes principales que la conforman.



Pantalla del NAM.

Dentro de NS2 se han creado varias librerías para poder simular escenarios específicos que no están contemplados por el simulador como lo son las simulaciones de redes Bluetooth para lo cual nos valemos de la librería UCBT de la Universidad de Cincinnati, la cual nos da los mejores resultados para la simulación de este tipo de redes. Sin embargo, para lograr que trabaje con el NAM desarrollamos una adaptación de esta librería en su versión 0.9.9.1. A continuación, se explican los comandos necesarios dentro del script para utilizar UCBT:

set val (mac) Mac/BNEP

esta sentencia indica que los nodos de la simulación trabajarán con la especificación Bluetooth.

set variable _nam [open nombre_archivo.nam w]

esta sentencia crea un archivo de texto con extensión NAM disponible para la escritura.

\$variable_simulador namtrace-all-wireless \$variable_nam x y

esta sentencia define que el escenario será de tipo wireless y con las dimensiones de las variables "x" y "y", además enlaza esta variable a la instancia del simulador.

Simulator set MacTrace_ON

Simulator set RouterTrace_ON

estas sentencias habilitan la escritura de paquetes de tipo MAC y de enrutamiento en el archivo de extensión NAM.

set node(id_nodo) [\$variable_simulador node id_nodo]

esta sentencia especifica como se crea un nodo en el NS2.

\$node(id_nodo) rt tipo_enrutamiento

esta sentencia indica el tipo de enrutamiento usado en la simulación, sea este estático (ManualRT) o dinámico (AODV)

[\$node(id_nodo) set l2cap_] set ifq_limit_ tamaño_cola

esta sentencia especifica el tamaño máximo del número de paquetes L2CAP que puede atender cada nodo.

\$node(id_nodo) inqscan 4096 2048

\$node(id_nodo) pagescan 4096 2048

estas sentencias especifican los tiempos promedio que un nodo debe permanecer en los procedimientos de inquiry y paging.

\$node(id_nodo) pos x y

esta sentencia indica las coordenadas del nodo en el escenario de simulación.

Los tipos de transferencia de paquetes que soporta la librería UCBT son TCP y UDP, la definición de estos la detallamos a continuación:

set var_tcp [new Agent/TCP]

\$variable_simulador attach-agent \$node(tx_node) \$var_tcp

estas sentencias crean un agente TCP y se lo añade con un nodo que será el transmisor en una transferencia TCP.

set var_ftp [new Application/FTP]

\$var_ftp attach-agent \$var_tcp

estas sentencias crean una aplicación FTP sobre el enlace TCP.

set var_ack [new Agent/TCPSink]

\$variable_simulador attach-agent \$node(rx_node) \$var_ack

estas sentencias establecen el nodo que recibirá los paquetes TCP y que a su vez enviará los paquetes ACK al nodo transmisor.

\$variable_simulador connect \$var_tcp \$var_ack

esta sentencia realiza la conexión entre los nodos trasmisor y receptor.

set var_queue [new Queue/DropTail]

\$var_queue set limit_ 20

estas sentencias definen e indica el número de paquetes que pueden estar almacenados en la cola de espera.

Sentencias utilizadas para la transferencia de paquetes UDP:

```
set var_udp [new Agent/UDP]
```

```
$variable_simulador attach-agent $node(tx_node) $var_udp
```

estas sentencias crean un agente UDP y se lo añade con un nodo que será el transmisor en una transferencia UDP.

```
set var_cbr [new Application/Traffic/CBR]
```

```
$var_cbr attach-agent $var_udp
```

estas sentencias crean una aplicación CBR sobre el enlace UDP.

```
set var_null [new Agent/Null]
```

```
$variable_simulador attach-agent $node(rx_node) $var_null
```

estas sentencias establecen el nodo que recibirá los paquetes UDP.

```
$variable_simulador connect $var_udp $var_null
```

esta sentencia realiza la conexión entre los nodos transmisor y receptor.

Para la organización de los eventos se utilizan los siguientes comandos:

```
$variable_simulacion at t1 "$node(master) make-bnep-connection $node(slave)  
pack_master pack_slave var_queue
```

esta sentencia establece la conexión maestro - esclavo y se indica el tipo de paquetes, en el caso de paquetes UDP el *var_queue* es reemplazado por *NONE*. Esta línea se ejecutara en el tiempo correspondiente a *t1*.

```
$variable_simulacion at t2 "$node(master) make-br $node(bridge) pack_master  
pack_bridge var_queue
```

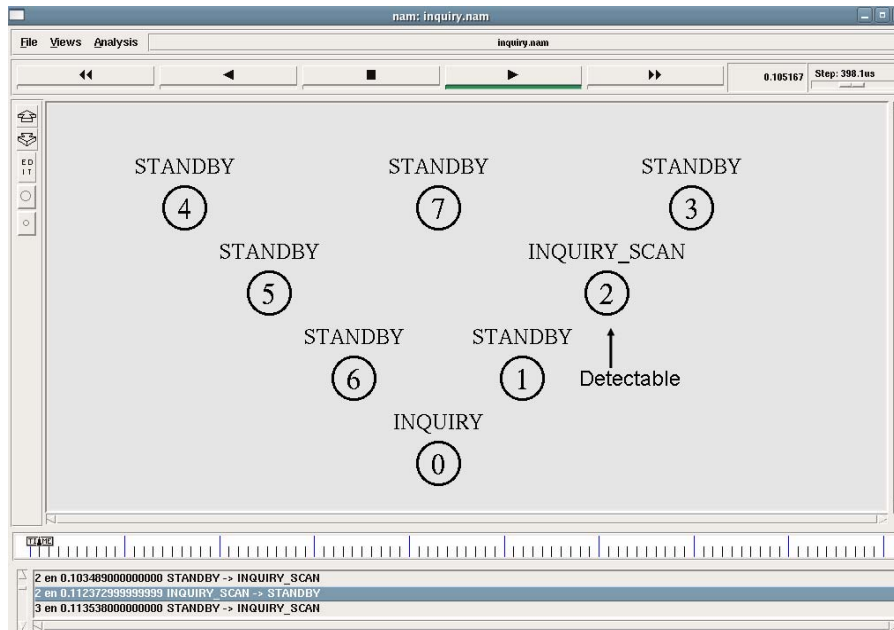
esta sentencia establece una conexión entre ambos nodos, especificando que el nodo "bridge" podrá aceptar múltiples conexiones activas, en el caso de paquetes UDP el *var_queue* es reemplazado por *NONE*.

Una vez que terminamos de definir todos los eventos de la simulación se debe llamar al procedimiento de finalización.

```
$ variable_simulacion at t3 "finish"
```

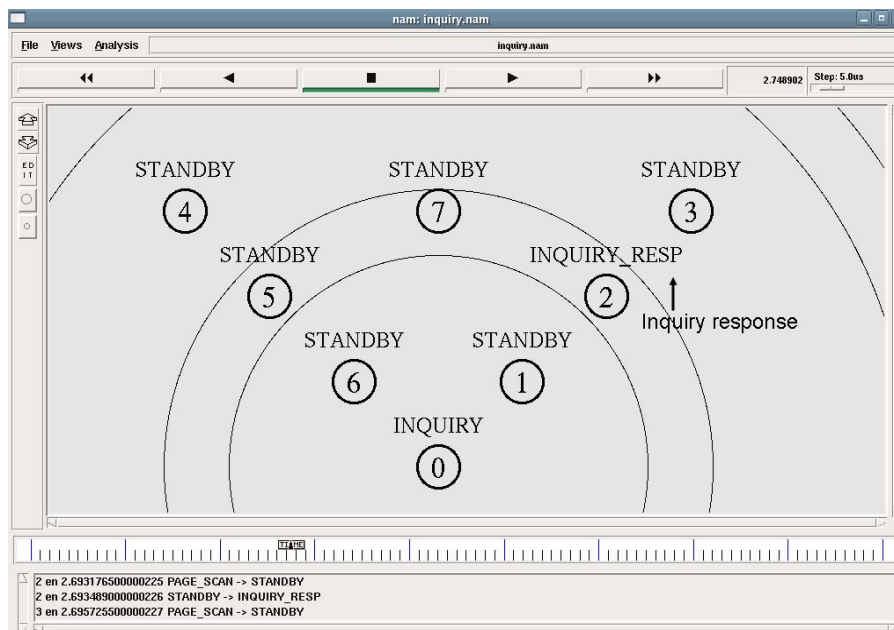
esta sentencia invoca en el tiempo *t3*, el cual hace referencia a la finalización de la simulación, al procedimiento de finalización "finish".

En las siguientes figuras se muestran los procedimientos básicos de redes Bluetooth utilizando las adaptaciones realizadas a la librería UCBT. Primero encontramos el procedimiento de detección (descubrimiento) de nodos dentro del área de cobertura a lo que se conoce con el nombre de *inquiry*.



Nodo 0 en procedimiento de Inquiry.

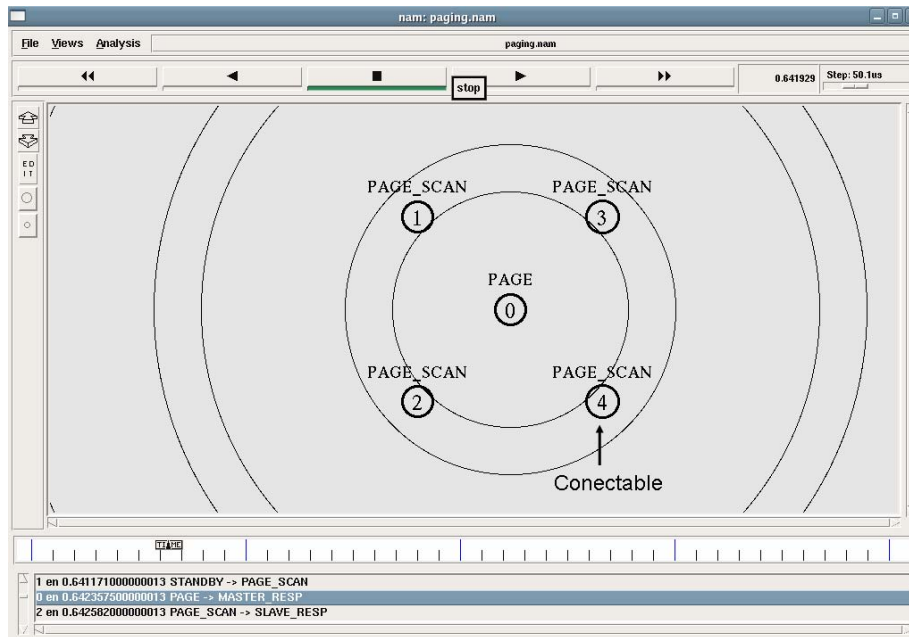
Podemos apreciar a varios dispositivos en una formación en “V” donde los dispositivos que se encuentren en el estado de INQUIRY_SCAN son aquellos que podrán ser detectados por el dispositivo que se encuentra en el estado de INQUIRY.



Nodo 2 envía una respuesta al procedimiento de Inquiry del Nodo 0.

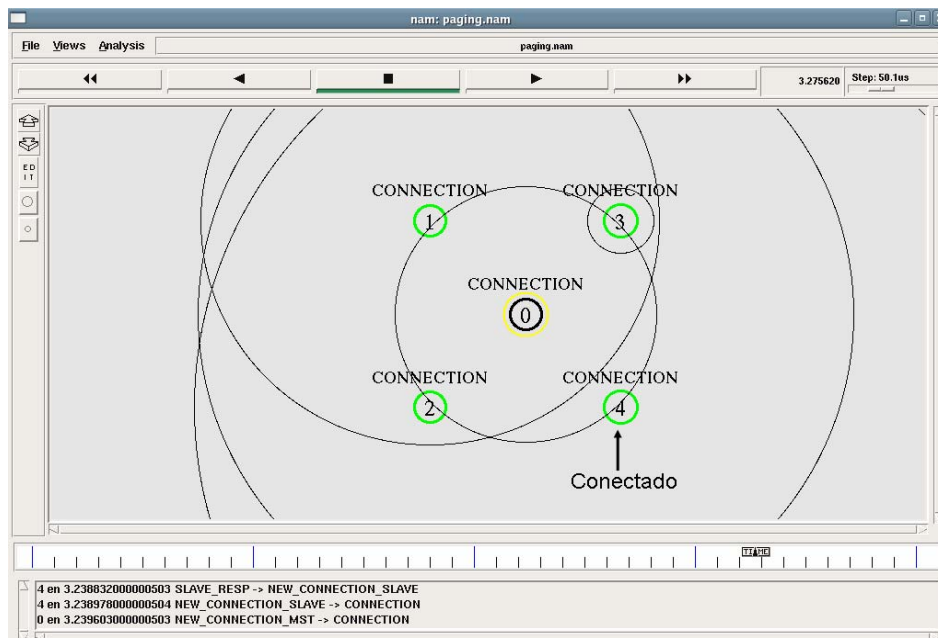
Una vez que el dispositivo que está buscando a otros (el que se encuentra en INQUIRY), envía peticiones a los dispositivos que pueden ser detectados (en INQUIRY_SCAN), estos envían una respuesta para que sean registrados y se confirme que están dentro del área de cobertura para poder luego conectarse entre sí. Cuando se han detectado los dispositivos que estarán próximos a ser interconectados, se realiza el procedimiento de conexión (Paging), Dentro de este procedimiento los dispositivos

pueden estar en 2 estados, el de PAGE_SCAN (conectable) y el de PAGE (buscando conectar dispositivos).



Nodo 0 en procedimiento de Page.

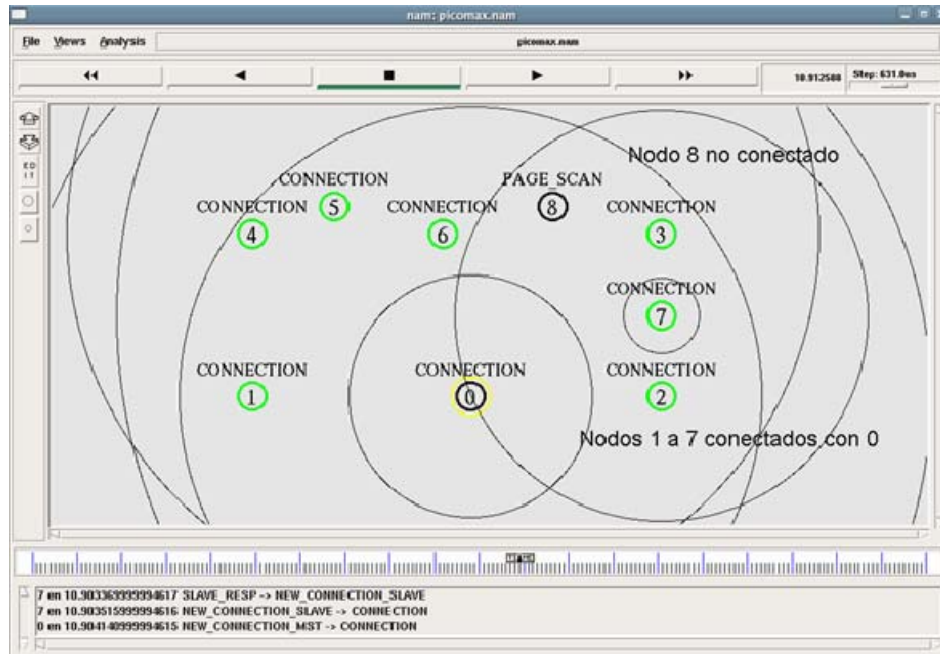
Para que los dispositivos entren en el estado de CONNECTION (conectado) deben primero realizar un par de peticiones y sincronizaciones entre los dispositivos que están participando, una vez que se efectuó la respuesta del PAGE, se empezara a intercambiar comandos LMP para establecer una piconet (forma básica de conexión en dispositivos Bluetooth).



Nodo 1, 2, 3, 4 conectados al Nodo 0.

Podemos apreciar como los nodos que intercambiaron las respuestas de PAGE y los comandos LMP necesarios para establecer la conexión son aquellos que tienen coloración verde y con el estado CONNECTION (conectado), cabe recalcar que en este

procedimiento se designan roles entre los dispositivos, el dispositivo que inicio con el PAGE es aquel que tendrá el role de MASTER (maestro, el dispositivo que podrá interconectar a más nodos dentro de la misma red), y aquellos que enviaron la respuesta tendrán el role de SLAVES (esclavos, nodos sujetos al esquema que imponga el dispositivo maestro).



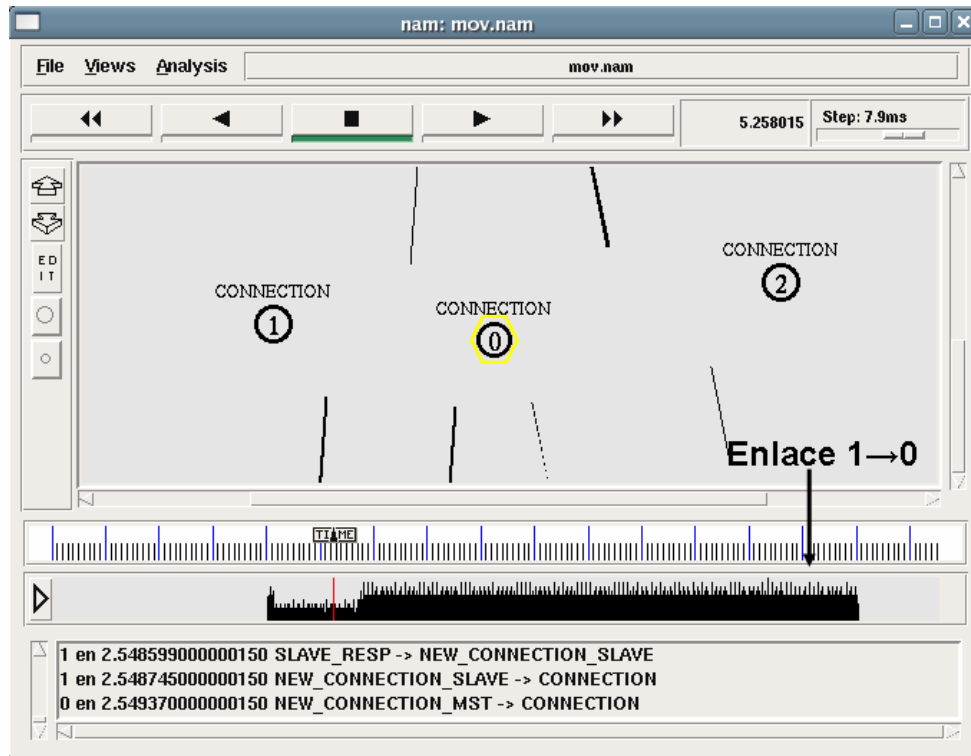
Formación de una piconet.

En la figura anterior tenemos una piconet con 8 nodos conectados, los nodos del 1 al 7 conectados con al nodo 0 que es el maestro de esta piconet (por tal motivo lleva una marca circular amarilla), y vemos que el nodo 8 no se encuentra conectado pero fue detectado y es conectable puesto que está en el estado de PAGE_SCAN, este nodo no se logra conectar debido a que la especificación Bluetooth define que una piconet puede tener hasta 7 esclavos activos conectados a un maestro.

En este momento los nodos conectados al maestro están listos para efectuar transferencias de paquetes entre esclavos o entre esclavos y el maestro en caso de que se hubiese detallado eso en la simulación.

Una de las herramientas que nos brinda el NAM y la cual podemos utilizar durante el tiempo que transcurra la simulación es la visualización de la ocupación de los enlaces entre los nodos que se encuentren conectados. Esta opción se puede activar dando click derecho a un costado de uno de los nodos que interviene en la transferencia de paquetes en dirección hacia el otro nodo conectado.

Al realizar esta acción nos saldrá un menú contextual de donde seleccionamos la opción "Graph" y por último seleccionamos el enlace que nos interese visualizar ya que nos permite obtener gráficos de los enlaces en ambos sentidos. La siguiente figura nos muestra la activación de esta opción en la ventana del NAM y nos muestra la ocupación del enlace entre el nodo 0 y el nodo 1 en el sentido del nodo 1 hacia el nodo 0.



Visualización del momento en el que el enlace esta ocupado.

APÉNDICE B.

Prácticas Didácticas de Simulación de Redes Bluetooth.

Estos apéndices de prácticas didácticas tienen el propósito de guiar al estudiante al desarrollo de las simulaciones de redes Bluetooth. Antes de empezar a trabajar con estas prácticas el estudiante debe estar familiarizado con los tutoriales de NS2, NAM y UCBT, descritos en el apéndice A.

B.1 Práctica No. 1 “Simulación de procedimientos básicos para la conexión de dispositivos Bluetooth basados en NS2/UCBT/NAM”

Práctica No.1

Simulación de procedimientos básicos para la conexión de dispositivos Bluetooth basados en NS2/UCBT/NAM

1. OBJETIVOS

- Orientar a los estudiantes al manejo de las simulaciones de redes Bluetooth utilizando el simulador de redes NS2.
- Conocer y visualizar los procedimientos que están incluidos en la formación de redes Bluetooth.
- Crear una topología de red Bluetooth básica (Piconet).
- Verificar el número máximo de enlaces activos dentro de una piconet.

2. INTRODUCCION

En esta práctica se aprenderán varios procedimientos básicos para establecer una comunicación vía Bluetooth, dentro de los cuales se mencionan: el Inquiry o Descubrimiento, el Paging o Emparejamiento, y el intercambio de paquetes necesarios para poder establecer una comunicación entre dispositivos.

UCBT es una librería compatible con el simulador de redes NS2. El objetivo de esta librería es utilizar los nodos creados en NS2 y adaptarlos para que funcionen dentro de la especificación Bluetooth, en lo que se

refiere a conexión, transferencia de archivos, rango de cobertura, movilidad, entre otros.

3. MARCO TEORICO

Bluetooth es una norma que define un standard global de comunicación inalámbrica, que posibilita la transmisión de voz y datos entre diferentes equipos mediante un enlace por radiofrecuencia. Los dispositivos Bluetooth operan en una frecuencia de radio que se encuentra entre los 2.4 y 2.48GHz, con la posibilidad de realizar hasta 1.600 saltos por segundo entre las 79 frecuencias soportadas en intervalos de 1MHz. Cada unidad incluye una radio, un controlador de enlaces de banda base y el software para la administración de los enlaces y flujo de datos.

Los usuarios tienen la opción de dos potencias de señal: un nivel de baja potencia para distancias de hasta 10 metros, y un nivel de alta potencia de hasta 100 metros de distancia para los puntos de acceso. Los dispositivos Bluetooth pueden conectarse simultáneamente hasta siete aparatos más, sin incluir el dispositivo maestro. La velocidad máxima de transferencia de datos es de aproximadamente 720 Kbps por canal.

4. MATERIALES Y EQUIPOS

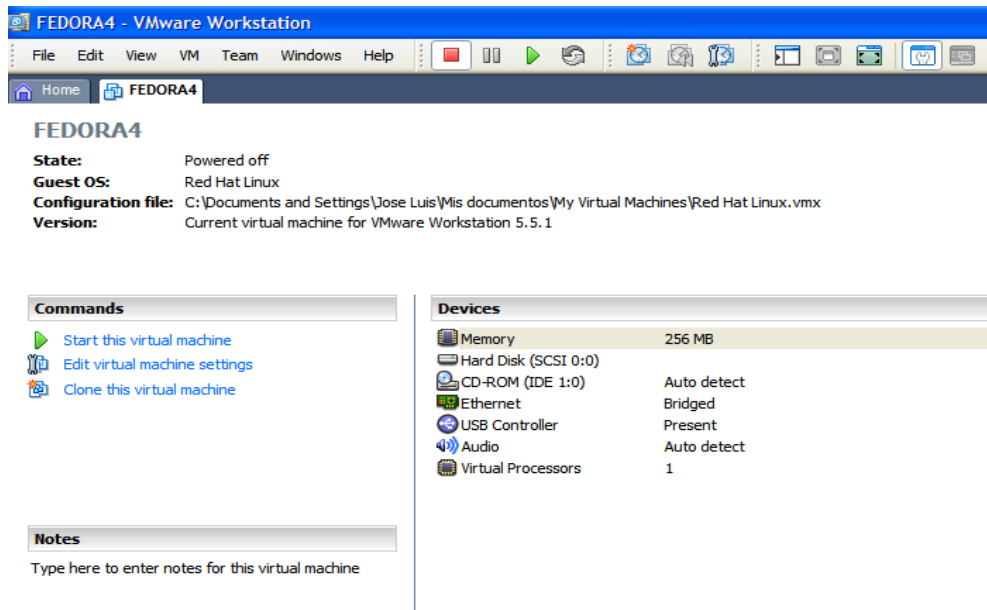
- PC con Fedora Core 4 y Network Simulator 2: Network Animator, librería UCBT modificada.

5. PROCEDIMIENTOS

Abriremos el emulador de sistemas operativos “VMware Workstation”, para esto haga doble click en el icono **VMware Workstation 5.5.1** en el escritorio o vaya a la siguiente dirección:

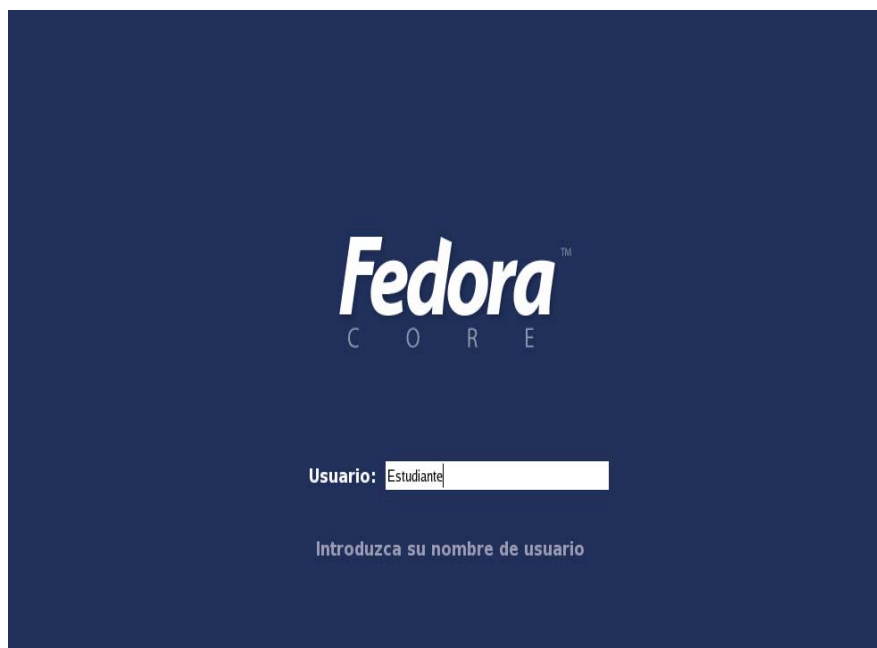
Inicio->Programas->VMware->VMware Workstation 5.5.1

aparecerá la siguiente ventana:

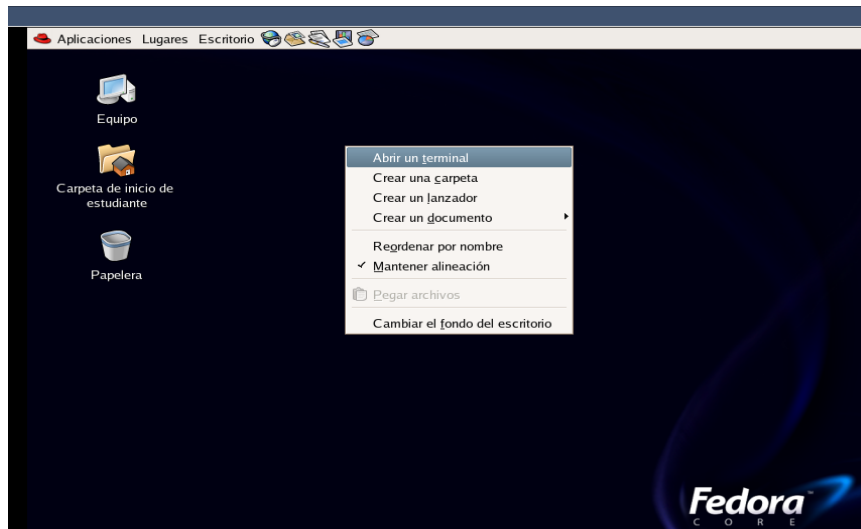


desde aquí haga click en “*Start this virtual machine*” para abrir el sistema operativo “Fedora Core 4”.

En la pantalla de inicio de sesión de Fedora Core 4 proceda a digitar en el campo usuario lo siguiente: Estudiante, como se muestra a continuación:



Luego, en el campo contraseña, proceda a digitar lo siguiente: Estudiante, y al finalizar presione “*Enter*”. Dentro del escritorio abrimos una ventana de terminal dando click derecho sobre el escritorio.

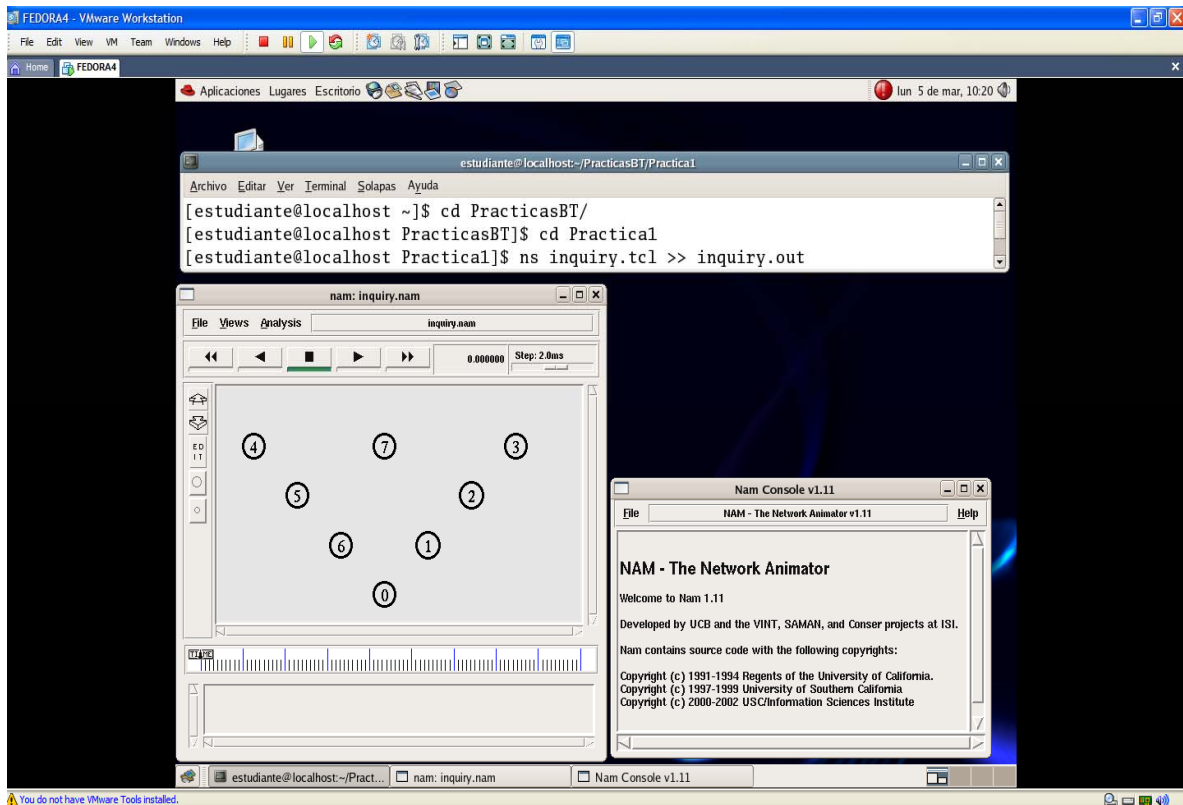


5.1 Simular un procedimiento de Inquiry y observar los cambios de estados de los nodos.

El procedimiento de Inquiry permite a un dispositivo ser descubierto por otros dispositivos que se encuentren en su rango y a su vez determina las direcciones y sus respectivos clocks. Se involucra un dispositivo (la fuente) el cual envía paquetes de inquiry (estado inquiry) y después recibe la respuesta correspondiente. El dispositivo que recibe los paquetes de inquiry (el destino) debe encontrarse en el sub-estado de inquiry scan para poder recibir dichos paquetes. Después el dispositivo destino entrará al sub-estado de inquiry response y enviará una respuesta de inquiry a la fuente. Luego que el procedimiento de inquiry se ha completado, se establece la conexión mediante el procedimiento de Paging.

Una vez que tengamos la pantalla del terminal, ejecutaremos las siguientes sentencias:

- `cd PracticasBT`
- `cd Practical`
- `ns inquiry.tcl >> inquiry.out`



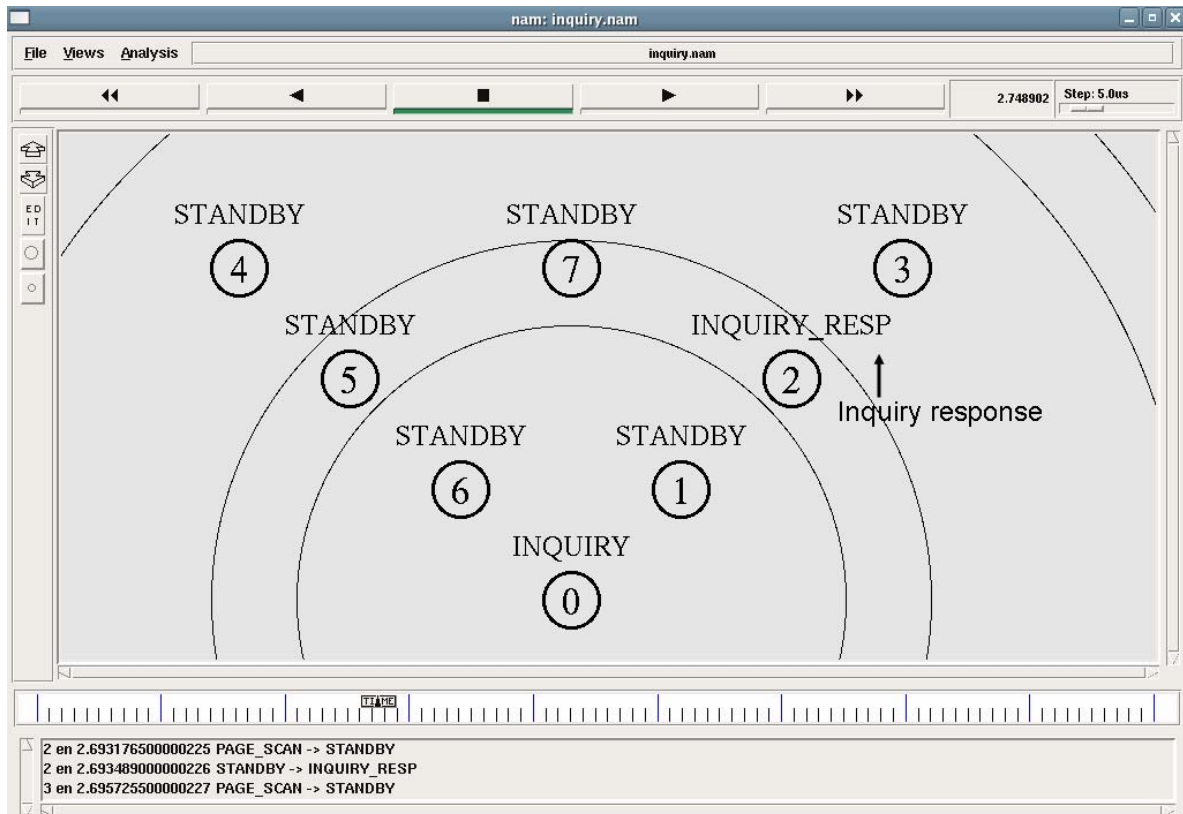
Al presionar *play* en la presente simulación se ejecutarán los estados de descubrimiento de dispositivos en los nodos 0 y 7, durante un tiempo determinado. El comando que se utilizará es el siguiente:

\$ns at 0.1 "\$node(0) inquiry 5 5"

Lo que implica que el nodo 0 permanecerá en un estado de inquiry durante 5x1.28seg. a menos que reciba 5 inquiry responses de dispositivos diferentes. Para el nodo 7 se ejecutará la siguiente sentencia:

\$ns at 6.0 "\$node(7) inquiry 2 5"

En la siguiente figura se puede apreciar el momento en el cual el nodo 2 envía un inquiry response hacia el nodo 0.



Al finalizar esta simulación conteste las siguientes preguntas:

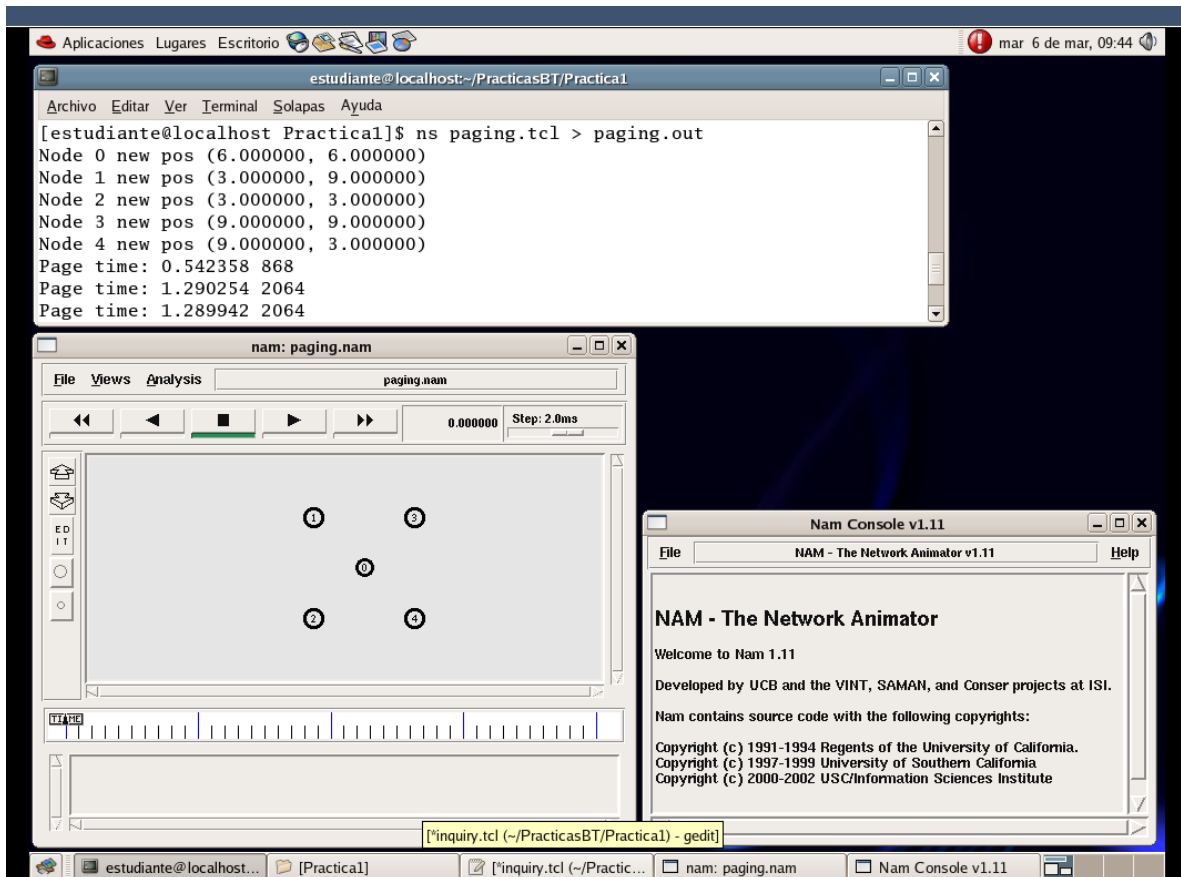
- ¿Cuáles son los 5 nodos que envían “*inquiry responses*” al nodo 0 y en que tiempo ocurre cada respuesta?
- ¿Cuánto tiempo se toma el nodo 0 durante el proceso de inquiry?
- ¿En que tiempo termina el nodo 7 su procedimiento de Inquiry?
- ¿Cuáles son los nodos que envían “*inquiry response*” al nodo 7?
- ¿Cuanto tiempo se toma el nodo 7 durante el proceso de Inquiry?

5.2 Simular un procedimiento de Paging y observar los roles que adquiere los nodos.

Una vez que conocemos los nodos disponibles para conectarse, se realiza el procedimiento de conexión o Paging, donde los nodos intercambiarán información con respecto a su posición y su frecuencia de reloj CLK para proceder a sincronizarse a un esquema dominante. El nodo que se impone en este esquema se denominará “maestro” y los que lo seguirán se llamarán “esclavos”.

En la ventana del terminal proceda a ejecutar la siguiente sentencia:

- `ns paging.tcl >> paging.out`

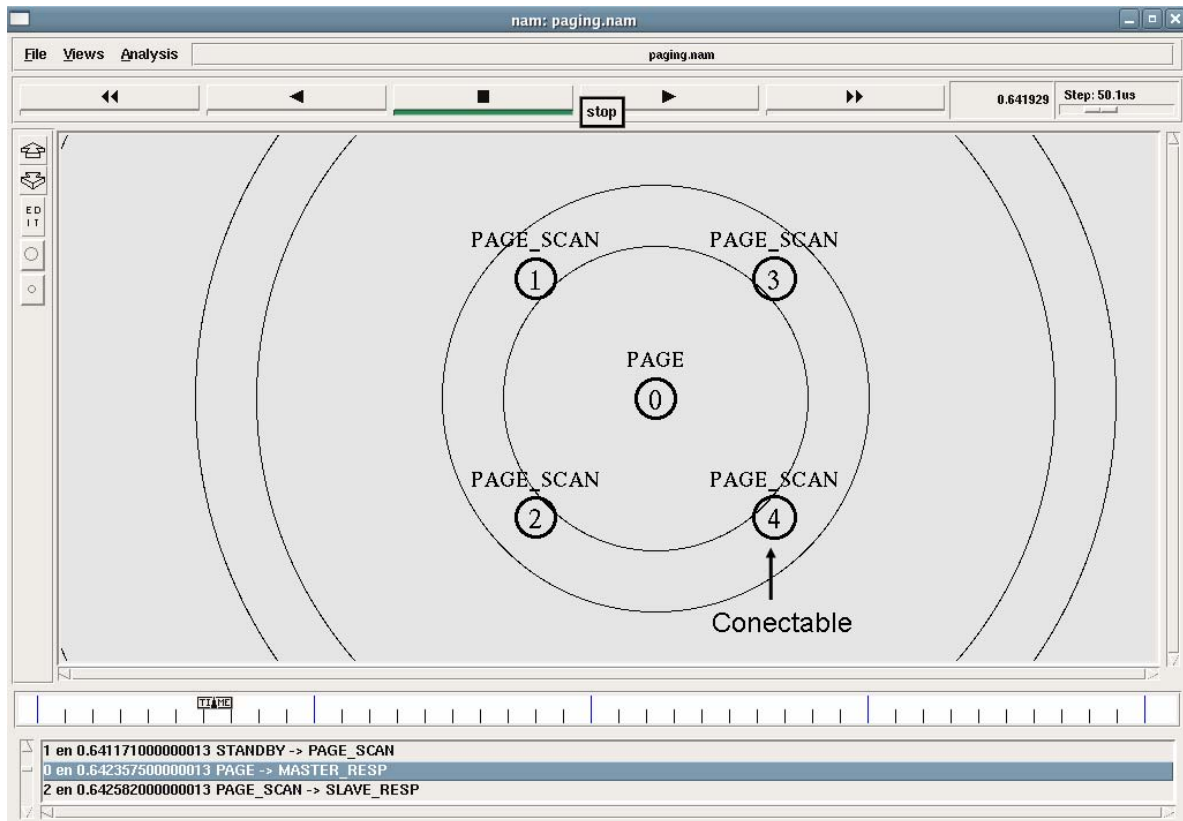


Al presionar el botón *play* en la presente simulación, podemos observar que se realizara el procedimiento de conexión de dispositivos, paging, dando como resultado un dispositivo maestro y varios dispositivos esclavos. Esto se lo consigue con el siguiente comando:

\$ns at 0.1 "\$node(0) make-bnep-connection \$node(1)"

Este comando permite establecer una conexión hasta la capa BNEP dentro de la pila de protocolos Bluetooth entre el nodo 0 y el nodo 1. Ambos dispositivos entrarán en el procedimiento de paging y de page scan respectivamente, realizando intercambios de paquetes necesarios para la conexión de ambos dispositivos.

En la figura se pueden apreciar los estados de *page* y *page scan* respectivamente:



Al finalizar esta simulación conteste las siguientes preguntas:

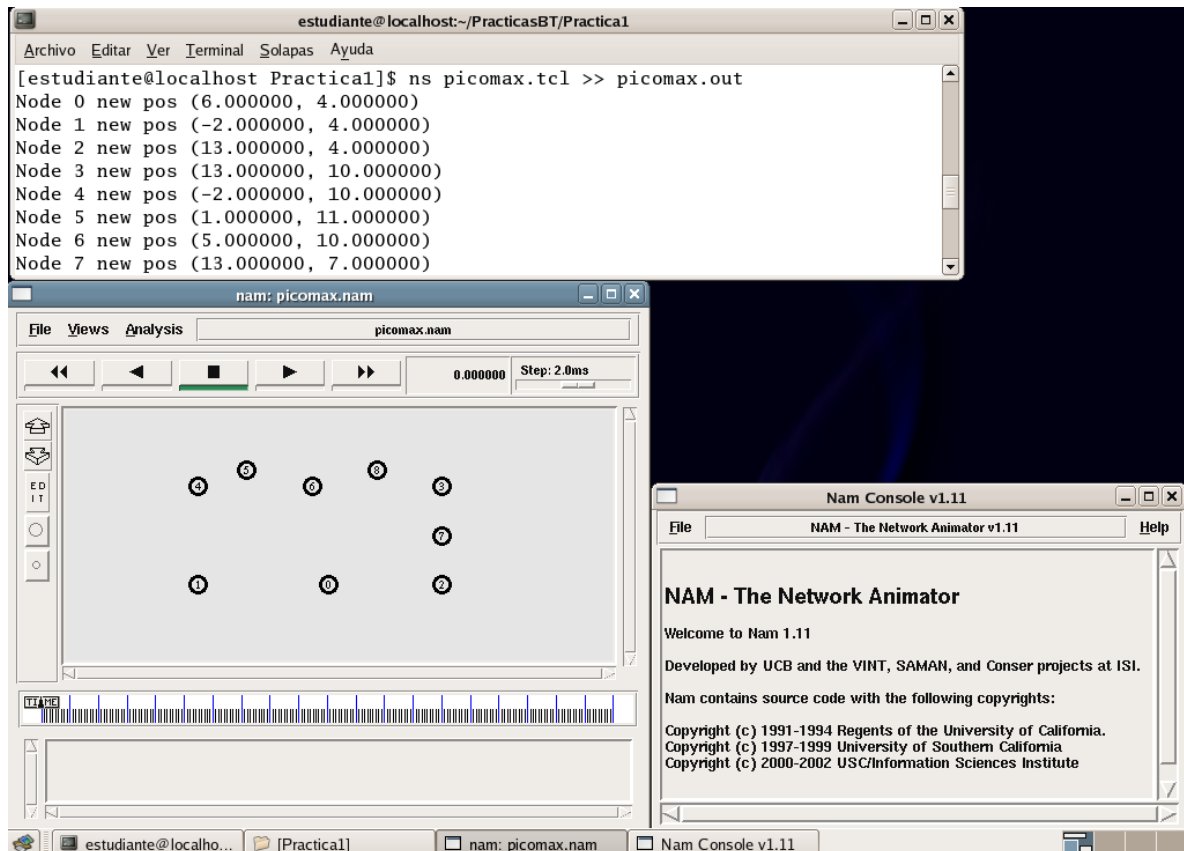
- ¿Qué tipo de paquetes envía y recibe el nodo 0?
- ¿Qué tipo de paquetes envían y reciben los otros nodos?
- ¿Qué nodos logran el rol de maestro?
- ¿Qué nodos logran el rol de esclavo?
- ¿Por cuáles estados deben de pasar los nodos hasta llegar al estado de conexión? Haga una tabla con cada nodo.

5.3 Simular una piconet con la cantidad máxima de nodos conectados y con una transferencia de archivos bajo una aplicación TCP.

Una vez concluido el procedimiento de paging, los dispositivos tomaron roles específicos dentro de una conexión común llamada piconet. Una piconet consta de un único dispositivo maestro al cual se conectan varios dispositivos esclavos. El número máximo de dispositivo esclavos “activos” dentro de una piconet es de 7. Una vez conectados, los nodos pueden intercambiar archivos según aplicaciones que se demanden entre dispositivos. Estas aplicaciones pueden ser basadas en el protocolo TCP o UDP según la aplicación.

En la ventana del terminal proceda a ejecutar la siguiente sentencia:

- `ns picomax.tcl >> picomax.out`



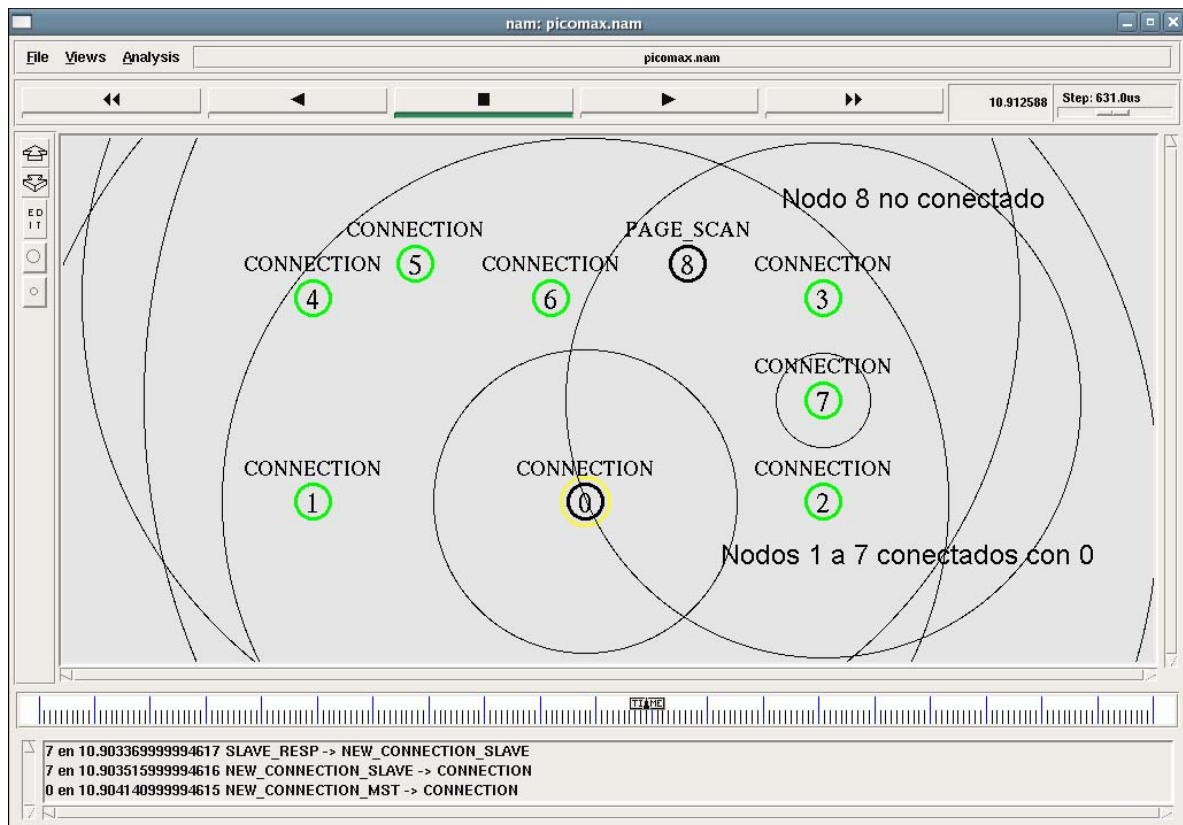
Al presionar el botón *play* en la presente simulación, podemos observar la formación de una piconet con el número máximo de nodos esclavos activos (7). Además habrá una transferencia de tipo TCP entre los nodos 1 y 2 que se la consigue con las siguientes líneas de código:

```

$ns at 0.1 "$node(0) make-bnep-connection $node(1) DH3 DH3 noqos $ifq"
$ns at 0.2 "$node(0) make-bnep-connection $node(2)"
$ns at 12 "$ftp0 start"
  
```

Estas líneas de código permiten establecer una aplicación de tipo FTP entre los nodos 1 y 2 mediante una transferencia de archivos de tipo TCP. Antes de empezar el traspaso de datos, se debe realizar una conexión hasta la capa BNEP entre el nodo maestro 0 y los nodos esclavos 1 y 2, respectivamente.

En la siguiente figura podemos apreciar la formación de una piconet:



Al finalizar esta simulación conteste las siguientes preguntas:

- Explique la función del nodo con marca amarilla y los nodos con marca verde.
- Explique que sucede con el nodo 8.
- En la transferencia de paquetes entre los nodos 1 y 2, explique por qué el tráfico pasa por el nodo 0.

BIBLIOGRAFIA

1. The VINT Project, The ns Manual, UC Berkeley, LBL, USC/ISI, and Xerox PARC, Kevin Fall (kfall@ee.lbl.gov), Kannan Varadhan kannan@catarina.usc.edu), Editores; Enero 8, 2003.
2. BluetoothSIG, Specifications Documents, <http://www.bluetooth.com/Bluetooth/Learn/Technology/Specifications/> revisada el 4 de Enero 2006.
3. D. Agrawal, Q. Wang, UCBT Bluetooth Extension for NS2 at the University of Cincinnati, <http://www.ececs.uc.edu/~cdmc/ucbt/ucbt.html> revisada el 15 de Enero de 2006.

Práctica No.2

Transferencia de paquetes en diferentes topologías de redes Bluetooth basados en NS2/UCBT/NAM

1. OBJETIVOS

- Examinar escenarios móviles de redes Bluetooth y verificar sus rangos de cobertura.
- Conocer el funcionamiento de los nodos PMPs y su importancia en las redes Bluetooth.
- Mostrar el intercambio de roles de dos dispositivos conectados a una piconet.
- Formar una topología de red Bluetooth avanzada (Scatternet).

2. INTRODUCCIÓN

En esta práctica se aprenderán escenarios avanzados de redes Bluetooth, donde tenemos topologías con gran número de nodos y varias aplicaciones simultáneas, así como el intercambio de archivos entre nodos bajo aplicaciones FTP y CBR y procedimientos de intercambio de roles entre dispositivos conectados.

En la librería UCBT se ha especificado como distancia máxima entre dispositivos Bluetooth cerca de 10 metros a la redonda, es decir la cobertura máxima de cada nodo será esa distancia. El objetivo es utilizar los nodos con las potencias mínimas, teniendo en consideración que por el hecho de ser dispositivos móviles, estos requieren un sistema de ahorro de energía.

3. MARCO TEORICO

Un dispositivo Bluetooth puede atender más de una conexión a la vez, y esto lo logra porque cuando se establece una piconet se usan ciertos canales en un esquema de salto de frecuencia, por lo que cada piconet tendrá un

esquema propio. Estos dispositivos que participan en más de una piconet a la vez se los conoce como nodos PMP (Participante en Múltiples Piconets), estos nodos suelen desempeñar tareas importantes cuando de interconectar piconets se trata, es decir, forman un enlace o puente para la interconexión entre piconets. A lo que se conoce con el nombre de scatternet.

Muchas veces dentro del establecimiento de una piconet o incluso de una scatternet, los dispositivos requieren hacer un cambio de roles para poder mejorar su topología y establecer una conexión con la menor cantidad de problemas. A este procedimiento se lo conoce con el nombre de Role-Switch.

Un nodo puente puede tener roles de maestro o de esclavo, según como se realice la interconexión. Debido a esto y otros factores, las tasas de datos en este tipo de topología pueden disminuir según sea el nivel de interferencia o la cantidad de dispositivos y aplicaciones simultáneas que se atiendan. Lo bueno de este tipo de topología es que de alguna manera puede ampliar la cobertura entre dispositivos que se encuentren muy lejanos a los 10 metros.

4. MATERIALES Y EQUIPOS

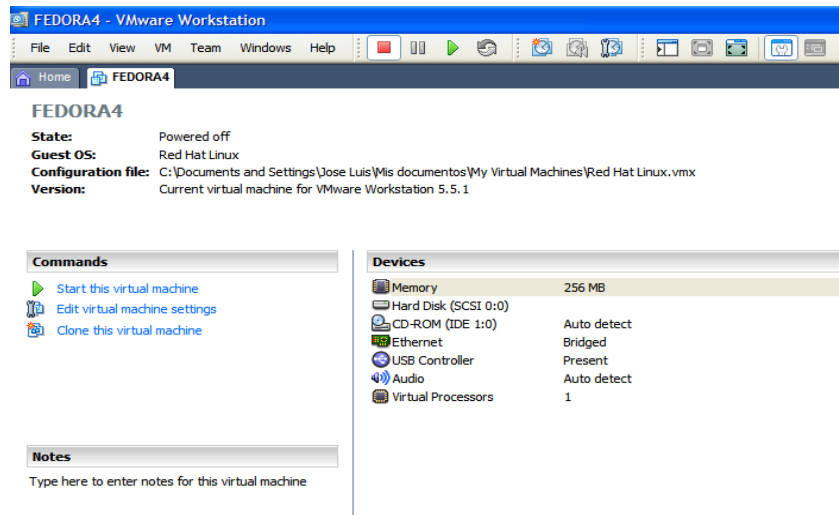
- PC con Fedora Core 4 y Network Simulator 2: Network Animator, librería UCBT modificada.

5. PROCEDIMIENTOS

Abriremos el emulador de sistemas operativos “VMware Workstation”, para esto haga doble click en el icono **VMware Workstation 5.5.1** en el escritorio o vaya a la siguiente dirección:

Inicio->Programas->VMware->VMware Workstation 5.5.1

aparecerá la siguiente ventana:

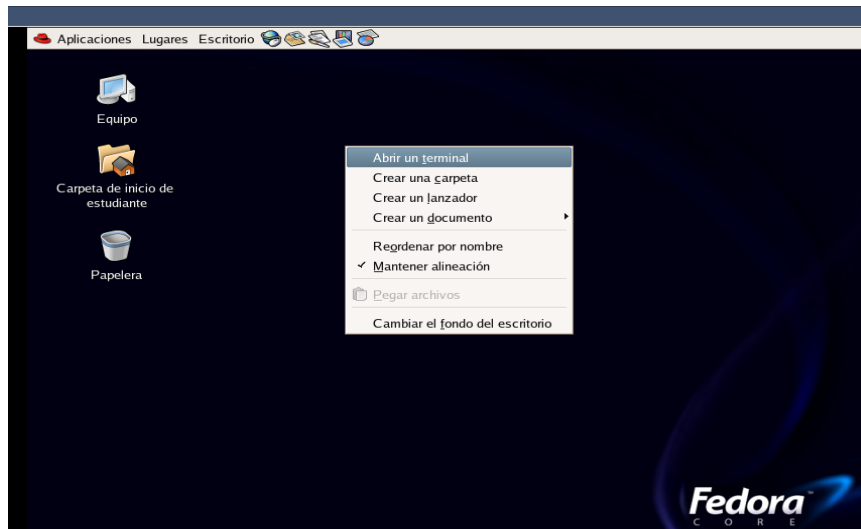


desde aquí haga click en “*Start this virtual machine*” para abrir el sistema operativo “Fedora Core 4”.

En la pantalla de inicio de sesión de Fedora Core 4 proceda a digitar en el campo usuario lo siguiente: Estudiante, como se muestra a continuación:



Luego, en el campo contraseña, proceda a digitar lo siguiente: Estudiante, y al finalizar presione “*Enter*”. Dentro del escritorio abrimos una ventana de terminal dando click derecho sobre el escritorio.

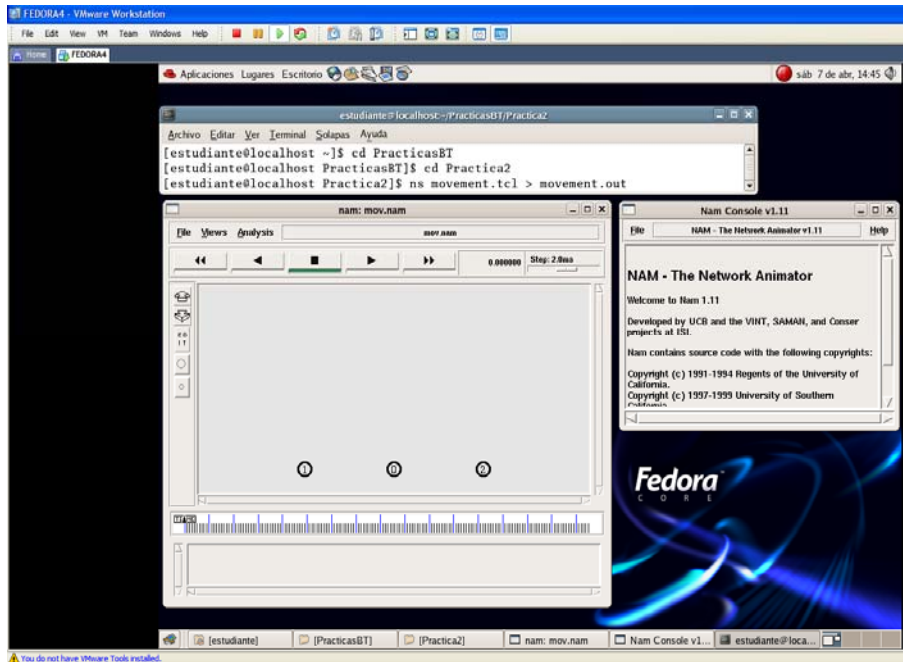


5.1 Simular una piconet con nodos móviles y analizar los rangos de cobertura.

En todo momento, los dispositivos conectados en una piconet deben estar dentro de un espacio físico en el cual puedan comunicarse, es decir, logren un intercambio efectivo de paquetes de información y control. A este espacio físico se lo denomina área de cobertura. Por definición, todos los nodos tienen un área de cobertura de 10 metros para potencias bajas, debido a esto, cuando los dispositivos han establecido una piconet significa que entre el dispositivo maestro y los dispositivos esclavos existen como máximo una distancia de 10 metros.

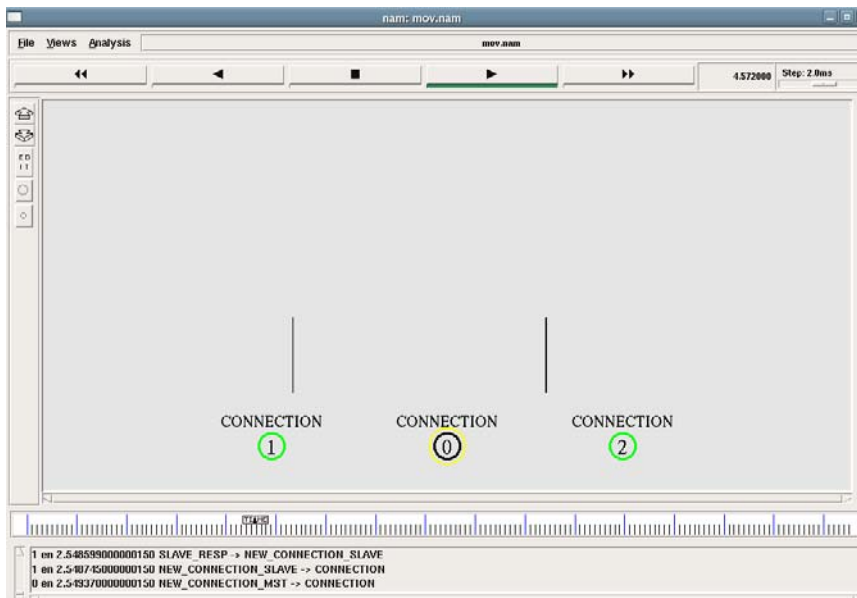
Una vez que tengamos la pantalla del terminal, ejecutaremos las siguientes sentencias:

- `cd PracticasBT`
- `cd Practica2`
- `ns movement.tcl >> movement.out`



Al presionar *play* en la presente simulación se ejecutará una piconet entre los nodos 0, 1 y 2 durante un tiempo determinado. Los comandos que se utilizarán son los siguientes:

\$ns at 1.0 "\$node(0) make-bnep-connection \$node(2) DH5 DH3 noqos \$ifq1"
\$ns at 1.1 "\$node(0) make-bnep-connection \$node(1) DH5 DH3 noqos \$ifq"
\$ns at 4.0 "\$ftp0 start"
\$ns at 4.1 "\$ftp1 start"



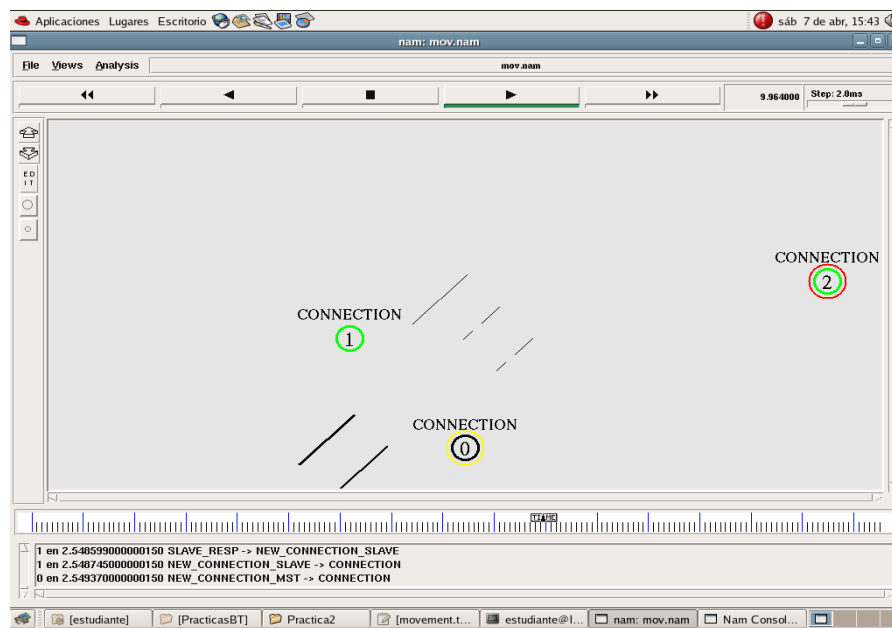
Una vez que la piconet ha sido establecida, los nodos 1 y 2 empezarán a moverse alrededor del nodo 0 o maestro, de tal manera que el nodo 2 termine fuera del rango de cobertura de la piconet y el nodo 1 se mantenga

en el rango de cobertura de la misma. Los comandos que se ejecutarán son los siguientes:

\$ns at 4.7 "\$node(1) setdest 6 10 1"

\$ns at 4.8 "\$node(2) setdest 22 8 5.5"

Estos comandos permiten el movimiento de los nodos 1 y 2. Al finalizar esta acción el nodo 2 se encontrará fuera del rango de cobertura y perderá toda transmisión pendiente con el nodo 0 como podemos apreciar en la siguiente figura:



Al finalizar la simulación conteste las siguientes preguntas:

- ¿En que momento el nodo 2 deja de recibir paquetes del nodo 0?
- Indique las posiciones iniciales y finales de los nodos 1 y 2

5.2 Simular dos piconets conectadas entre ellas por un nodo PMP.

Los nodos PMPs, o también conocidos como nodos participantes en múltiples piconets, son aquellos que establecieron conexión con más de una piconet a la vez. Las razones por las cuáles estos nodos llegaron a participar en varias piconets puede ser: por haber formado parte de un procedimiento de cambio de roles, por haber aceptado aplicaciones de dispositivos que están formando piconets diferentes, o por petición de una topología compleja de red Bluetooth, como una scatternet, haciendo que este dispositivo sirva de nodo puente entre piconets por su situación geográfica.

Una vez que tengamos la pantalla del terminal, ejecutaremos las siguientes sentencias:

- cd PracticasBT
- cd Practica2
- ns pmp.tcl >> pmp.out

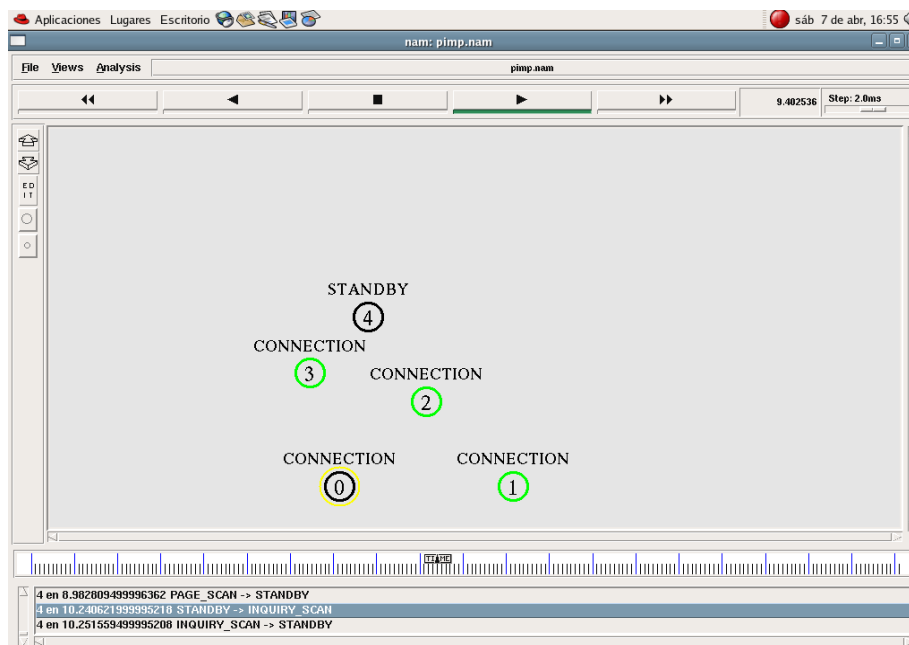
Al presionar play en esta simulación tendremos la formación una piconet inicial entre 0, 1, 2 y 3. Los comandos a utilizar son los siguientes:

```
$ns at 1.2 "$node(0) make-bnep-connection $node(1) DH5 DH3 none"
```

```
$ns at 3.4 "$node(0) make-bnep-connection $node(2) DH5 DH3 none"
```

```
$ns at 5.6 "$node(0) make-bnep-connection $node(3) DH5 DH3 none"
```

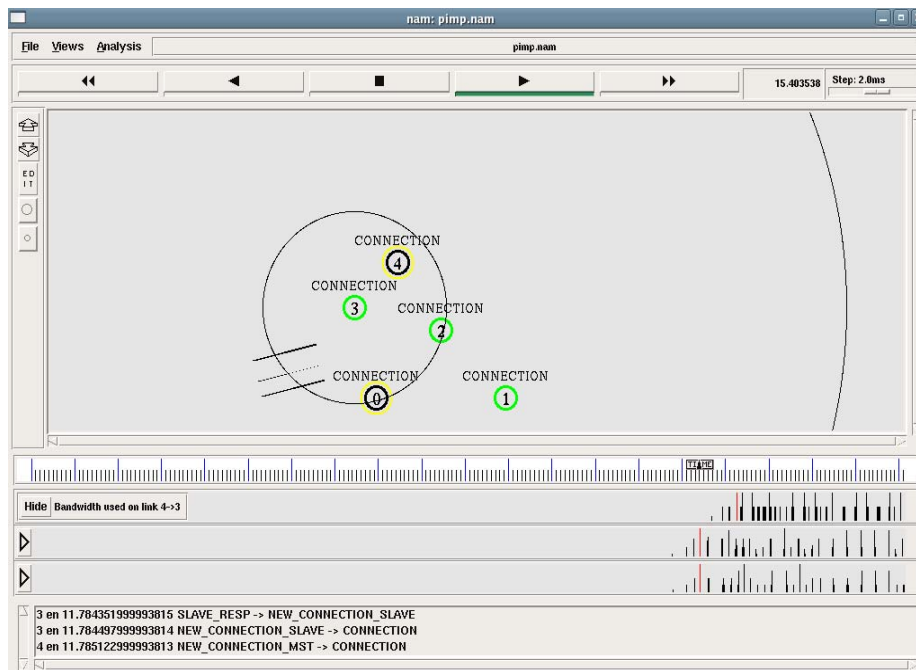
```
$ns at 14.5 "$ftp0 start"
```



Para conectar el nodo 4 con la piconet vamos a utilizar el nodo 3 como puente con el siguiente comando:

```
$ns at 10.5 "$node(4) make-br $node(3) DH5 DH3 none"
```

Con esta sentencia creamos una nueva piconet entre el nodo 3 y el nodo 4 que a su vez se encuentra conectada con la piconet anterior gracias al nodo 3 que actúa como bridge o puente entre ambas piconets. En la siguiente figura se muestra una transferencia de archivos entre las dos piconets:



Al finalizar la simulación conteste las siguientes preguntas:

- ¿Qué tipo de funciones realiza el nodo PMP?
- ¿En qué momento se establece la piconet entre los nodos 3 y 4?
- ¿Qué rol desempeña el nodo PMP en esta simulación?

5.3 Simular el intercambio de roles entre dos dispositivos conectados en una piconet.

Este procedimiento es utilizado únicamente por dos dispositivos que estén conectados en una piconet, es decir, ambos dispositivos intercambiarán roles, dando como resultado que el dispositivo maestro se convierta en el esclavo y que el esclavo se convierta en maestro. Cuando ocurre este procedimiento se forma nuevamente la piconet, en otras palabras, el esquema de saltos entre canales se redefine según el nuevo dispositivo maestro.

Si existiera el caso en el que uno de los dispositivos que va a realizar el cambio de roles tuviera alguna conexión con otro dispositivo, este enlace no se pierde y el dispositivo quedaría trabajando como un nodo participante en múltiples piconets.

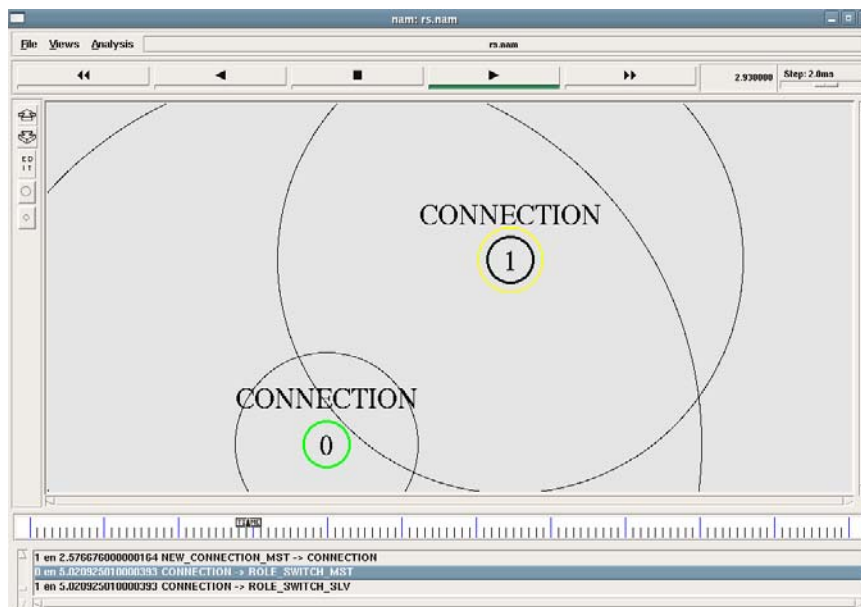
Una vez que tengamos la pantalla del terminal, ejecutaremos las siguientes sentencias:

- cd PracticasBT

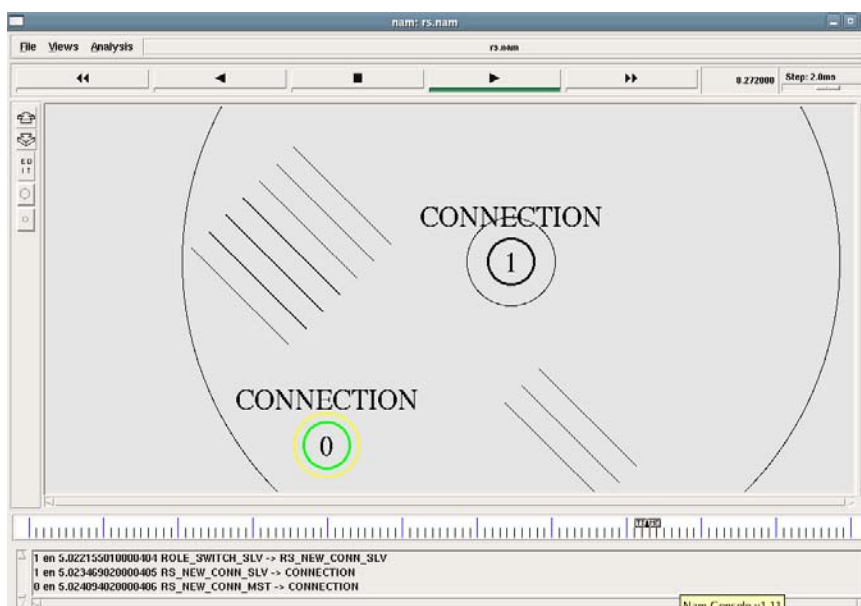
- cd Practica2
- ns rs.tcl >> rs.out

Al ejecutar esta simulación tendremos una piconet entre los nodos 1 y 0, siendo maestro y esclavo respectivamente, utilizando el siguiente comando:

\$ns at 0.1 "\$node(1) make-bnep-connection \$node(0) DH1 DH1"



Una vez establecida la piconet procedemos al intercambio de roles entre el nodo 0 y el nodo 1, así también se establecerá una transferencia TCP entre ambos nodos teniendo ahora como nuevo maestro al nodo 0 y como nuevo esclavo al nodo 1 como lo observamos en la siguiente figura:



Al finalizar la simulación conteste las siguientes preguntas:

- Verificar los canales de frecuencia que se utilizan en la piconet antes de que exista el intercambio de roles.
- Verificar todos los estados y paquetes intercambiados en el momento del intercambio de roles.
- Verificar los canales de frecuencia que usa la nueva piconet. ¿Existe alguna diferencia?
- Verifique el CLK que utiliza la nueva piconet.

5.4 Simular una red Bluetooth avanzada (Scatternet).

Las scatternets básicamente resultan de la interconexión de 2 o más piconets. Es una topología de red bastante compleja, ya que se pueden solapar ciertos canales e incurrir a niveles de interferencias considerables entre las piconets que se interconecten, así como problemas en la transmisión de datos, ancho de banda y sincronización de los dispositivos que participan en la scatternet. Además, los nodos que enlazan estas piconets, conocidos como nodos puentes, deben participar en ambas piconets según el rol que tengan en cada una.

Una vez que tengamos la pantalla del terminal, ejecutaremos las siguientes sentencias:

- `cd PracticasBT`
- `cd Practica2`
- `ns scatternet.tcl >> scatternet.out`

Al ejecutar esta simulación estableceremos tres piconets entre los nodos 0, 1 y 2 la primera, 3, 4 y 5 la segunda y 6, 7 y 8 la tercera con los siguientes comandos:

```
$ns at 1.2 "$node(0) make-bnep-connection $node(1) DH5 DH3 none"
```

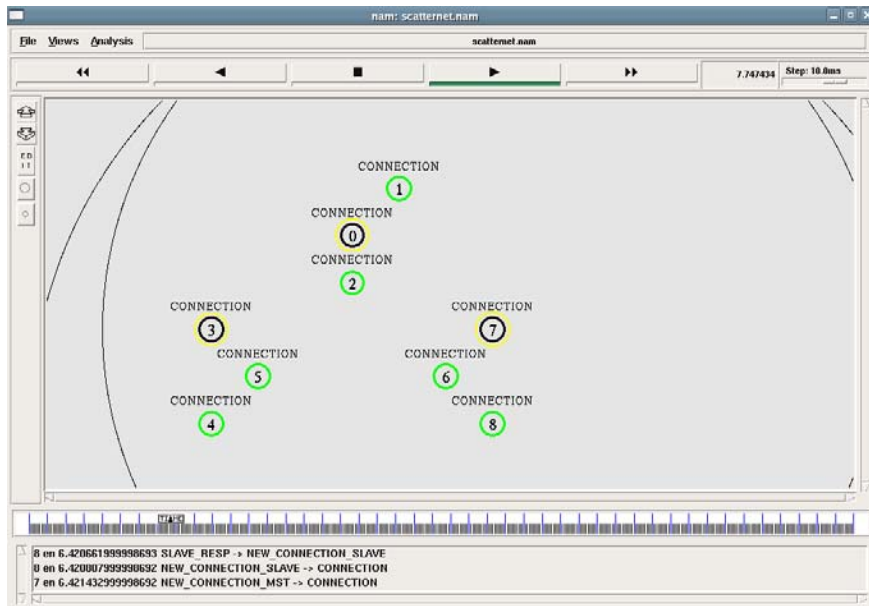
```
$ns at 1.4 "$node(0) make-bnep-connection $node(2) DH5 DH3 none"
```

```
$ns at 2.2 "$node(3) make-bnep-connection $node(4) DH5 DH3 none"
```

```
$ns at 2.4 "$node(3) make-bnep-connection $node(5) DH5 DH3 none"
```

```
$ns at 3.2 "$node(7) make-bnep-connection $node(6) DH5 DH3 none"
```

```
$ns at 3.4 "$node(7) make-bnep-connection $node(8) DH5 DH3 none"
```

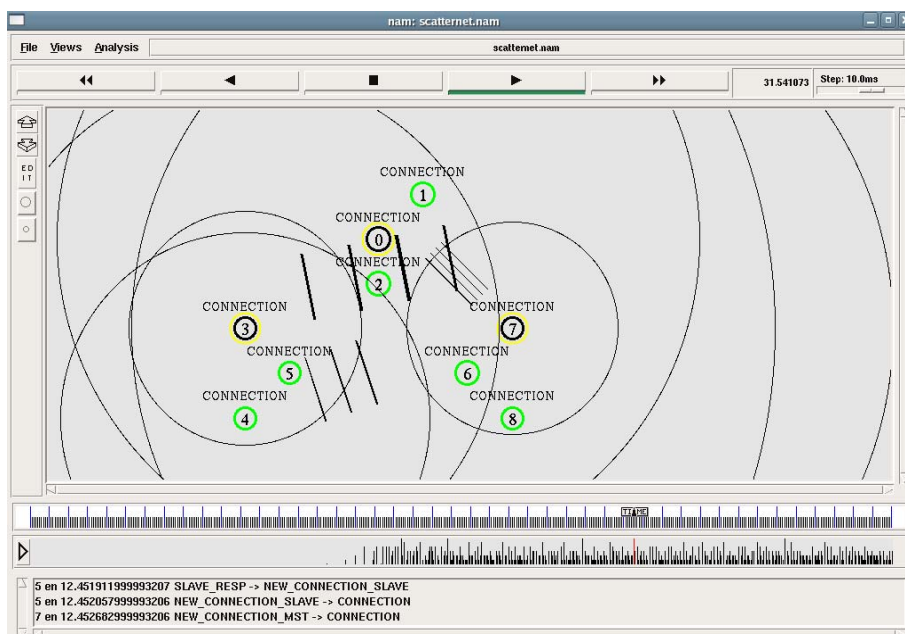
Luego procederemos a interconectar las tres piconets mediante dos nodos puente con el siguiente comando:

```
$ns at 10.0 "$node(3) make-br $node(2) DH5 DH3 none"
$ns at 11.0 "$node(7) make-br $node(5) DH5 DH3 none"
```

Finalizaremos con una transferencia de datos entre los nodos 0 y 7 para verificar que las piconets se encuentren interconectadas por medio del siguiente comando:

```
$ns at 15.0 "$ftp0 start"
```

En la siguiente figura vemos la transferencia de datos entre ambos nodos:



BIBLIOGRAFIA

The VINT Project, The ns Manual, UC Berkeley, LBL, USC/ISI, and Xerox PARC,.Kevin Fall (kfall@ee.lbl.gov), Kannan Varadhan kannan@catarina.usc.edu), Editores; Enero 8, 2003.

BluetoothSIG, Specifications Documents, <http://www.bluetooth.com/Bluetooth/Learn/Technology/Specifications/> revisada el 4 de Enero 2006.

D. Agrawal, Q. Wang, UCBT Bluetooth Extension for NS2 at the University of Cincinnati, <http://www.ececs.uc.edu/~cdmc/ucbt/ucbt.html> revisada el 15 de Enero de 2006.

APÉNDICE C.

Guías resueltas de las prácticas del laboratorio para los tutores.

C.1 Guía de Práctica No. 1 “Simulación de procedimientos básicos para la conexión de dispositivos Bluetooth basados en NS2/UCBT/NAM”

Guía de Práctica No.1

Simulación de procedimientos básicos para la conexión de dispositivos Bluetooth basados en NS2/UCBT/NAM

1. OBJETIVOS

- Orientar a los estudiantes al manejo de las simulaciones de redes Bluetooth utilizando el simulador de redes NS2.
- Conocer y visualizar los procedimientos que están incluidos en la formación de redes Bluetooth.
- Crear una topología de red básica Bluetooth (Piconet).
- Verificar el número máximo de enlaces activos dentro de una piconet.

2. INTRODUCCION

Dentro de esta práctica se aprenderán varios procedimientos básicos para establecer una comunicación vía Bluetooth, dentro de los cuales se mencionan: el Inquiry o Descubrimiento, el Paging o Emparejamiento, y el intercambio de paquetes necesarios para poder establecer una comunicación entre dispositivos.

UCBT es una librería compatible con el simulador de redes NS2. El objetivo de esta librería es utilizar los nodos creados en NS2 y adaptarlos para que funcionen dentro de la especificación Bluetooth, en lo que se refiere a conexión, transferencia de archivos, rango de cobertura, movilidad, entre otros.

3. MARCO TEORICO

Bluetooth es una norma que define un standard global de comunicación inalámbrica, que posibilita la transmisión de voz y datos entre diferentes equipos mediante un enlace por radiofrecuencia. Los dispositivos Bluetooth operan en una frecuencia de radio que se encuentra entre los 2.4 y 2.48GHz, con la posibilidad de realizar hasta 1.600 saltos por segundo entre las 79 frecuencias soportadas en intervalos de 1MHz. Cada unidad incluye una radio, un controlador de enlaces de banda base y el software para la administración de los enlaces y flujo de datos.

Los usuarios tienen la opción de dos potencias de señal: un nivel de baja potencia para distancias de hasta 10 metros, y un nivel de alta potencia de hasta 100 metros de distancia para los puntos de acceso. Los dispositivos Bluetooth pueden conectarse simultáneamente hasta siete aparatos más, sin incluir el dispositivo maestro. La velocidad máxima de transferencia de datos es de aproximadamente 720 Kbps por canal.

4. MATERIALES Y EQUIPOS

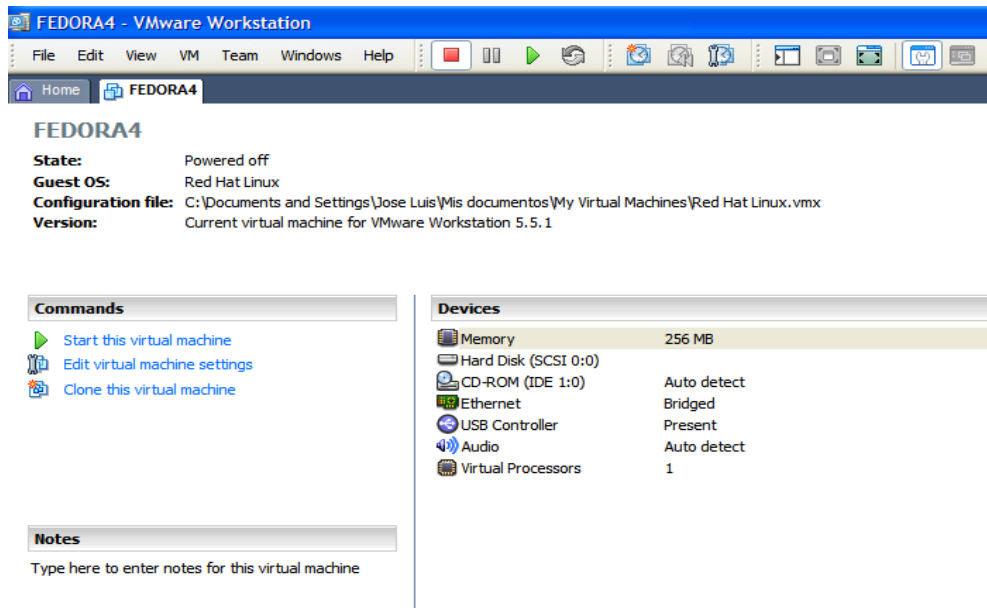
- PC con Fedora Core 4 y Network Simulator 2: Network Animator, librería UCBT modificada.

5. PROCEDIMIENTOS

Abriremos el emulador de sistemas operativos “VMware Workstation”, para esto haga doble click en el icono **VMware Workstation 5.5.1** en el escritorio o vaya a la siguiente dirección:

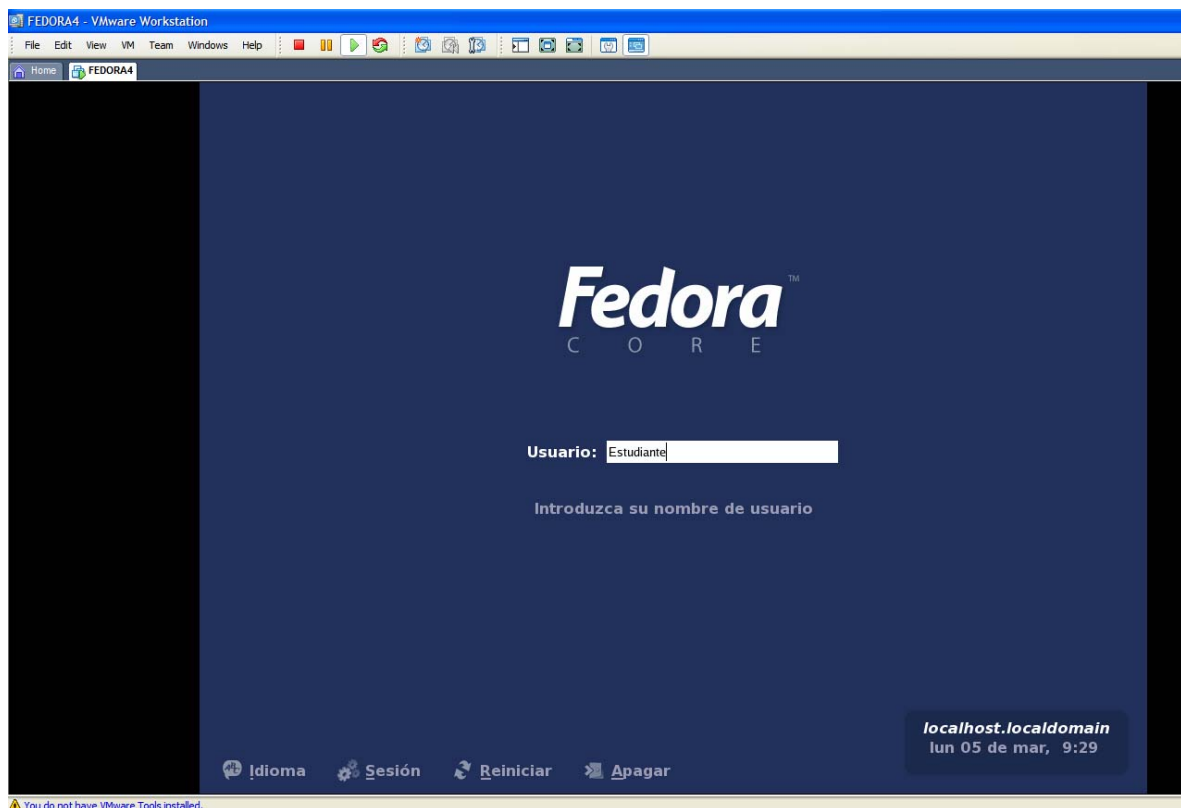
Inicio->Programas->VMware->VMware Workstation 5.5.1

aparecerá la siguiente ventana:



desde aquí haga click en “*Start this virtual machine*” para abrir el sistema operativo “Fedora Core 4”.

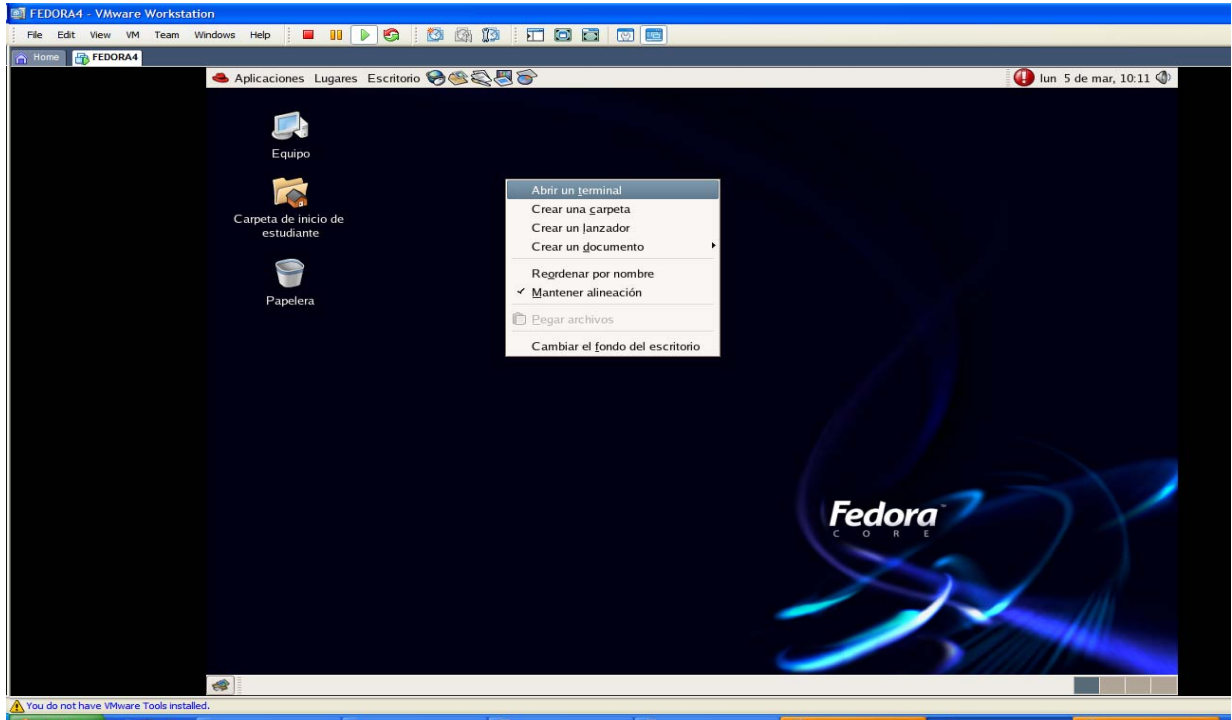
En la pantalla de inicio de sesión de Fedora Core 4 proceda a digitar en el campo usuario lo siguiente: Estudiante, como se muestra a continuación:



Luego, en el campo contraseña, proceda a digitar lo siguiente: Estudiante, y al finalizar presione “*Enter*”.

5.1 Simular un procedimiento de Inquiry y observar los cambios de estados de los nodos.

Dentro del escritorio abrimos una ventana de terminal, esto lo hacemos dando click derecho sobre el escritorio

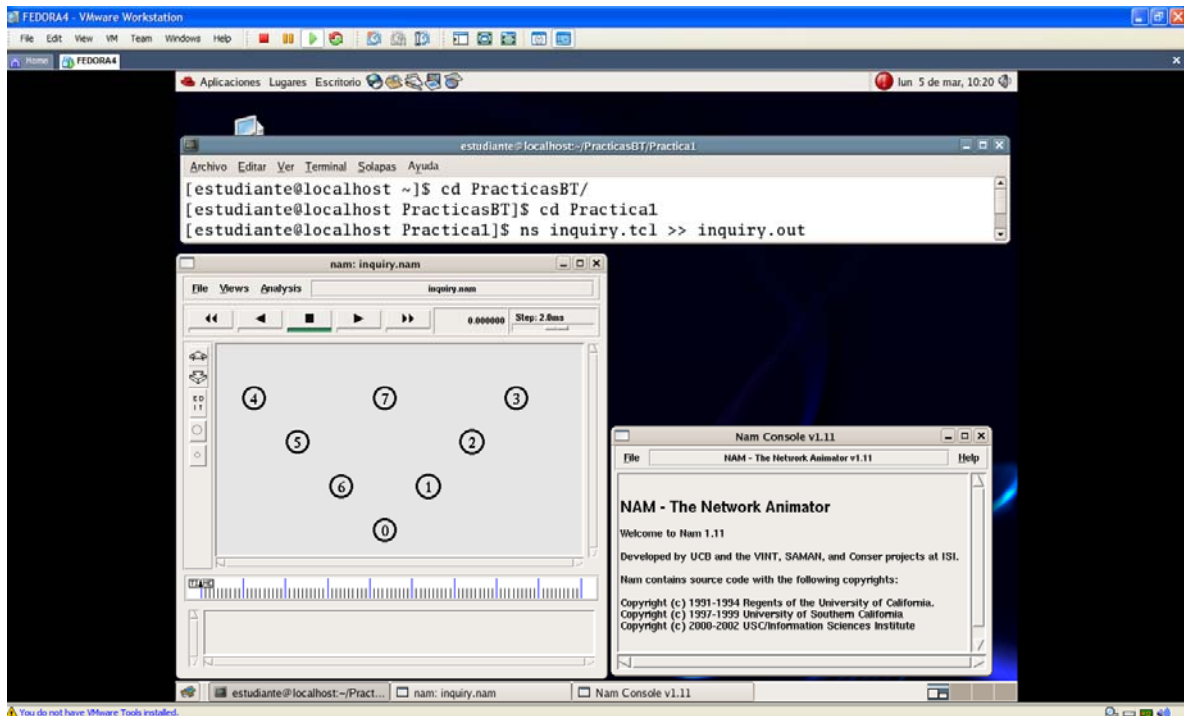


Una vez que tengamos la pantalla del terminal, ejecutaremos las siguientes sentencias:

```
cd PracticasBT
```

```
cd Practical
```

```
ns inquiry.tcl >> inquiry.out
```



Al presionar *play* en la presente simulación se ejecutarán los estados de descubrimiento de dispositivos en los nodos 0 y 7, durante un tiempo determinado. El comando que se utilizará es el siguiente:

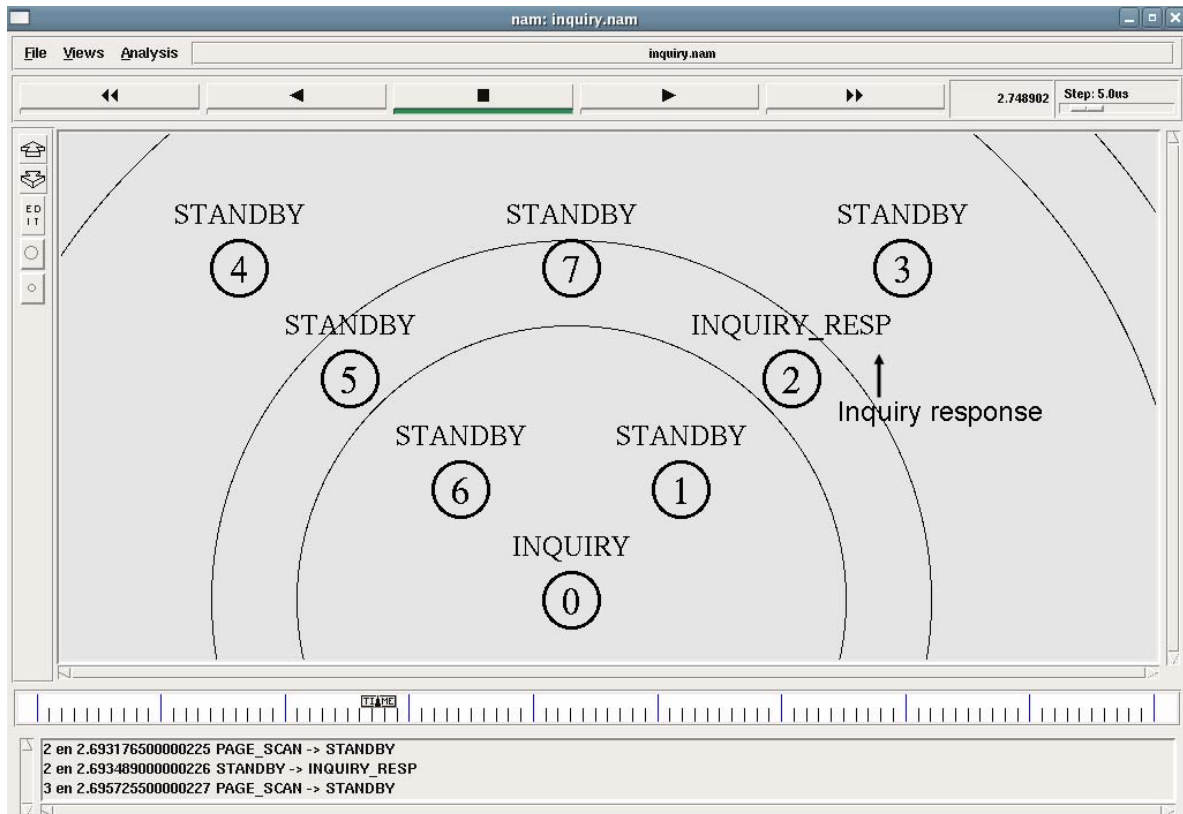
\$ns at 0.1 "\$node(0) inquiry 5 5"

Lo que implica que el nodo 0 permanecerá en un estado de inquiry durante 5x1.28seg. a menos que reciba 5 inquiry responses de dispositivos diferentes. Para el nodo 7 se ejecutará la siguiente sentencia:

\$ns at 6.0 "\$node(7) inquiry 2 5"

Se recomienda modificar los parámetros de respuestas de inquiry y el tiempo que toma el realizar el procedimiento para generar diferentes resultados en las simulaciones. Además se pueden escoger diferentes nodos para que realicen las peticiones de inquiry.

En la siguiente figura se puede apreciar el momento en el cual el nodo 2 envía un inquiry response hacia el nodo 0.



Al finalizar esta simulación conteste las siguientes preguntas:

¿Cuáles son los 5 nodos que envían “inquiry responses” al nodo 0 y en que tiempo ocurre cada respuesta?

Los nodos que envían “inquiry responses” al nodo 0 son: el nodo 1 en 0.2536s, el nodo 2 en 0.3861, el nodo 3 en 2.8670, el nodo 4 en 3.3073 y el nodo 5 en 3.8845

¿Cuánto tiempo se toma el nodo 0 durante el procedimiento de inquiry?

El nodo 0 empieza el procedimiento de inquiry en el tiempo 0.1001s y termina en el tiempo 3.9841s

¿Cuáles son los nodos que envían “inquiry responses” al nodo 7?

Los nodos que envían “inquiry responses” al nodo 7 son: el nodo 2 en el tiempo 8.3043s y el nodo 4 en el tiempo 8.3068s

¿Cuanto tiempo se toma el nodo 7 durante el proceso de Inquiry?

El nodo 7 empieza el procedimiento de inquiry en el tiempo 6.0003s y termina en el tiempo 8.5606s, es decir el nodo 7 se toma $2 * 1.28 = 2.56$ segundos.

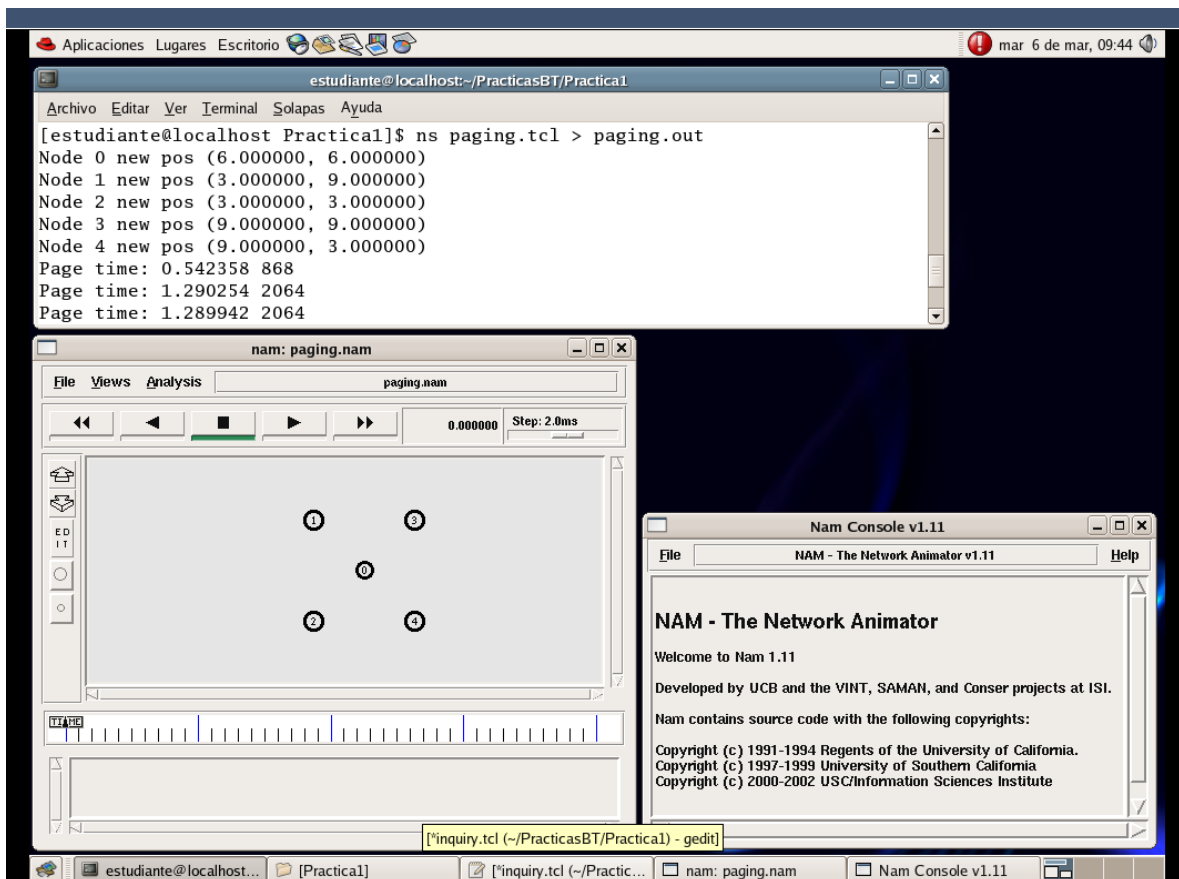
*Cabe recalcar al estudiante que para el nodo 7 se puede utilizar la fórmula $T_{Inq} = n * 1.28s$ debido a que no recibe al menos cinco respuestas durante este tiempo.*

5.2 Simular un procedimiento de Paging y observar los roles con los que quedan los nodos

Una vez que conocemos los nodos disponibles para conectarse, se realiza el procedimiento de conexión o Paging, donde los nodos intercambiarán información con respecto a su posición y su frecuencia de reloj CLK para proceder a sincronizarse a un esquema dominante. El nodo que se impone en este esquema se denominará “maestro” y los que lo seguirán se llamarán “esclavos”.

En la ventana del terminal proceda a ejecutar la siguiente sentencia:

➤ `ns paging.tcl >> paging.out`



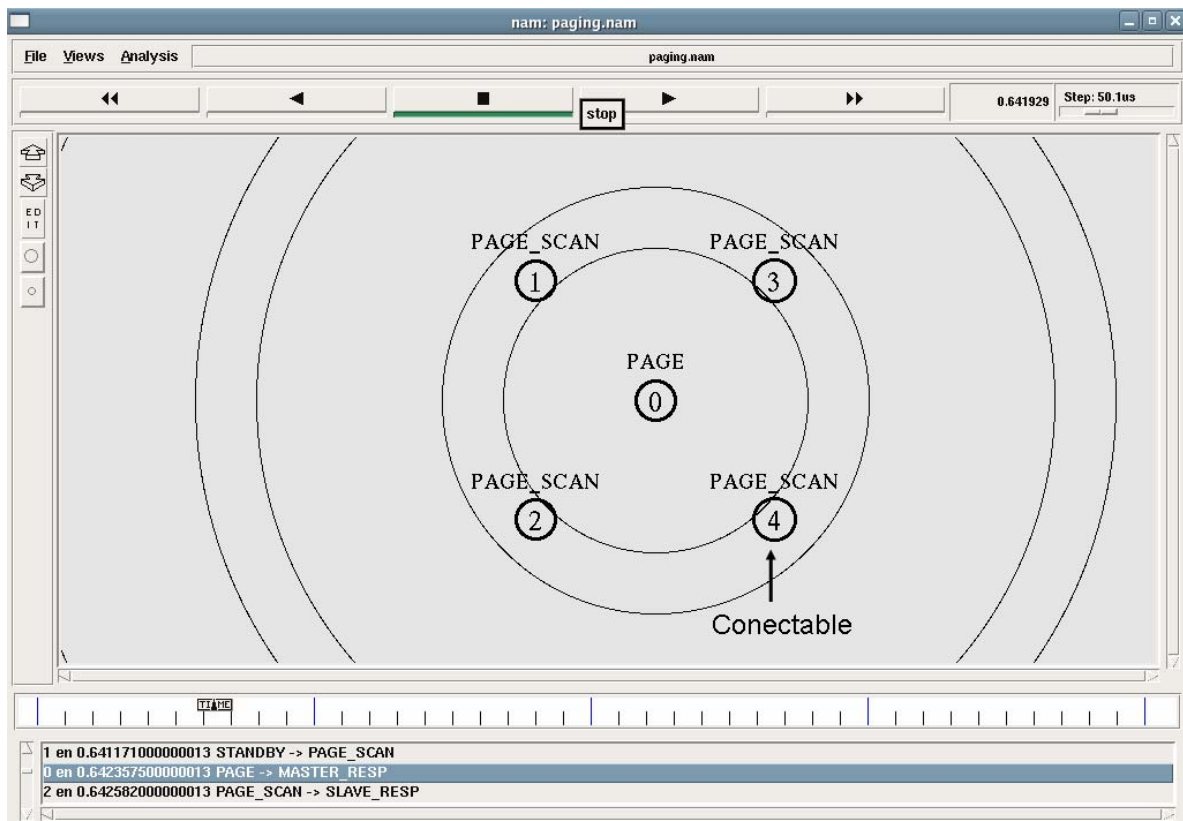
Al presionar el botón *play* en la siguiente simulación, podemos observar que se realizara el procedimiento de conexión de dispositivos, paging, dando como resultado un dispositivo maestro y varios dispositivos esclavos. Esto se lo consigue con el siguiente comando:

`$ns at 0.1 "$node(0) make-bnep-connection $node(1)"`

Este comando permite establecer una conexión hasta la capa BNEP dentro de la pila de protocolos Bluetooth entre el nodo 0 y el nodo 1. Ambos dispositivos entrarán en el procedimiento de paging y de page scan respectivamente, realizando intercambios de paquetes necesarios para la conexión de ambos dispositivos. La sintaxis de este comando especifica que el nodo cero logrará el rol de maestro y el nodo 1 logrará el rol de esclavo respectivamente.

Se recomienda modificar el orden de los nodos en el comando “make-bnep-connection” para obtener diferentes respuestas con lo que respecta a que nodos lograrán el rol de maestro y esclavo. Así como diferenciar el tipo de paquetes que intercambiarán para este nuevo caso.

En la figura se puede apreciar a los dispositivos en el momento en el que se envían respuestas para los dispositivos



Al finalizar esta simulación conteste las siguientes preguntas:

- **¿Qué tipo de paquetes envía y recibe el nodo 0?**
El nodo 0 envía paquetes de tipo ID, POLL y FH mientras que recibe paquetes de tipo ID y NULL.
- **¿Qué tipo de paquetes envían y reciben los otros nodos?**

Los nodos 1, 2, 3 y 4 envían paquetes de tipo ID y NULL mientras que reciben paquetes de tipo ID, FH y POLL.

- **¿Qué nodos logran el rol de maestro?**
Solamente el nodo 0 logra el rol de maestro.
- **¿Qué nodos logran el rol de esclavo?**
Los nodos 1, 2, 3 y 4 logran el rol de esclavo.
- **¿Por cuáles estados deben de pasar los nodos hasta llegar al estado de conexión? Haga una tabla con cada nodo.**

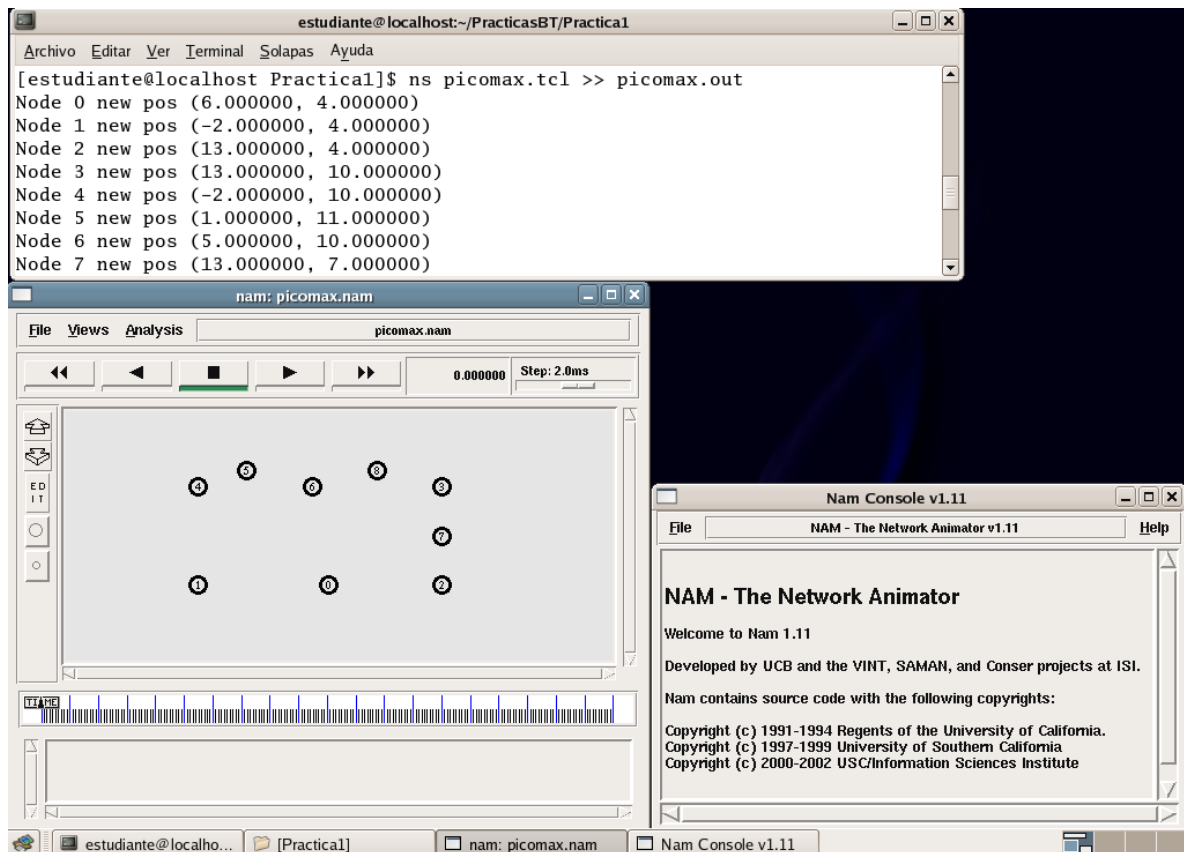
Nodo 0	Nodo 1	Nodo 2	Nodo 3	Nodo 4
<i>Inquiry scan</i>	<i>Inquiry Scan</i>	<i>Inquiry Scan</i>	<i>Inquiry Scan</i>	<i>Inquiry Scan</i>
<i>Standby</i>	<i>Standby</i>	<i>Standby</i>	<i>Standby</i>	<i>Standby</i>
<i>Page</i>	<i>Page Scan</i>	<i>Page Scan</i>	<i>Page Scan</i>	<i>Page Scan</i>
<i>Master response</i>	<i>Slave Response</i>	<i>Slave Response</i>	<i>Slave Response</i>	<i>Slave Response</i>
<i>New Connetion master</i>	<i>New Slave connection</i>	<i>New Slave connection</i>	<i>New Slave connection</i>	<i>New Slave connection</i>
<i>Connection</i>	<i>Connection</i>	<i>Connection</i>	<i>Connection</i>	<i>Connection</i>

5.3 Simular una piconet con la cantidad máxima de nodos conectados y con una transferencia de archivos bajo una aplicación TCP.

Una vez concluido el procedimiento de paging, los dispositivos tomaron roles específicos dentro de una conexión común llamada piconet. Una piconet consta de un único dispositivo maestro al cual se conectan varios dispositivos esclavos. El número máximo de dispositivo esclavos “activos” dentro de una piconet es de 7. Una vez conectados, los nodos pueden intercambiar archivos según aplicaciones que se demanden entre dispositivos. Estas aplicaciones pueden ser basadas en el protocolo TCP o UDP según la aplicación.

En la ventana del terminal proceda a ejecutar la siguiente sentencia:

➤ ns picomax.tcl >> picomax.out



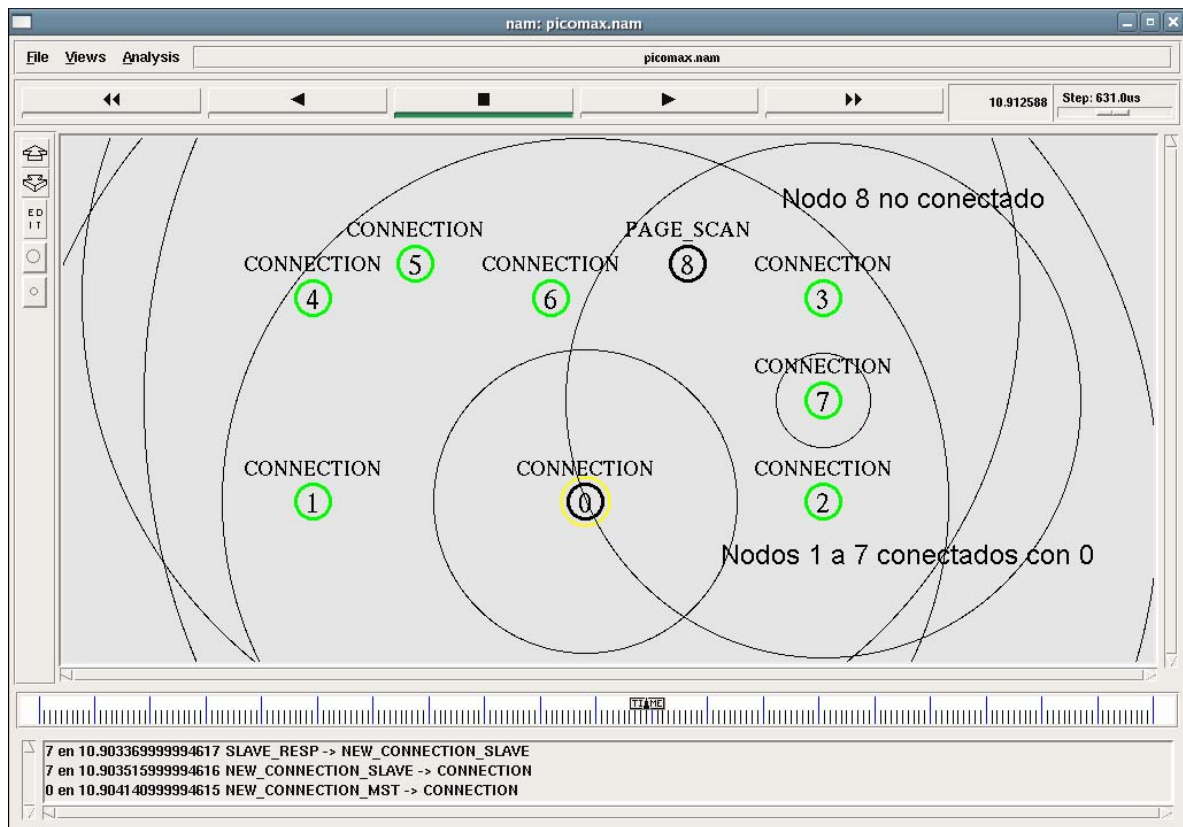
Al presionar el botón *play* en la presente simulación, podemos observar la formación de una piconet con el número máximo de nodos esclavos activos (7). Además habrá una transferencia de tipo TCP entre los nodos 1 y 2 que se la consigue con las siguientes líneas de código:

```
$ns at 0.1 "$node(0) make-bnep-connection $node(1) DH3 DH3 noqos $ifq"
$ns at 0.2 "$node(0) make-bnep-connection $node(2)"
$ns at 12 "$ftp0 start"
```

Estas líneas de código permiten establecer una aplicación de tipo FTP entre los nodos 1 y 2 mediante una transferencia de archivos de tipo TCP. Antes de empezar el traspaso de datos, se debe realizar una conexión hasta la capa BNEP entre el nodo maestro 0 y los nodos esclavos 1 y 2, respectivamente.

Se recomienda comprobar la conectividad del nodo ocho agregándolo a la piconet en lugar de un nodo que ya forma parte de la red. Para esto, modifique el comando "make-bnep-connection" con un nodo que no participe en la transferencia de datos de la piconet.

En la siguiente figura podemos apreciar la formación de una piconet:



Al finalizar esta simulación conteste las siguientes preguntas:

- Explique la función del nodo con marca amarilla y de los nodos con marca verde.**

El nodo con marca amarilla es considerado el maestro de la piconet y su función consiste en sincronizar y establecer conexiones con los nodos que lo demanden. Los nodos con marca verde son considerados los esclavos de la piconet y su función consiste en intercambiar con el maestro información de su estado tales como su posición y conectividad.
- Explique que sucede con el nodo 8.**

El nodo 8 no puede conectarse a la piconet debido a que ya existen 7 nodos esclavos activos conectados.
- En la transferencia de paquetes entre los nodos 1 y 2, explique por qué el tráfico pasa por el nodo 0.**

Debido a que los nodos 1 y 2 son esclavos, el tráfico de paquetes debe pasar por el nodo maestro que los conecta, en este caso, el nodo 0.

BIBLIOGRAFIA

1. The VINT Project, The ns Manual, UC Berkeley, LBL, USC/ISI, and Xerox PARC,.Kevin Fall (kfall@ee.lbl.gov), Kannan Varadhan kannan@catarina.usc.edu), Editores; Enero 8, 2003.
2. BluetoothSIG, Specifications Documents, <http://www.bluetooth.com/Bluetooth/Learn/Technology/Specifications/> revisada el 4 de Enero 2006.
3. D. Agrawal, Q. Wang, UCBT Bluetooth Extension for NS2 at the University of Cincinnati, <http://www.ececs.uc.edu/~cdmc/ucbt/ucbt.html> revisada el 15 de Enero de 2006.

C.2 Guía de Práctica No. 2 “Transferencia de paquetes en diferentes topologías de redes Bluetooth basados en NS2/UCBT/NAM”

Guía de Práctica No.2

Transferencia de paquetes en diferentes topologías de redes Bluetooth basados en NS2/UCBT/NAM

1. OBJETIVOS

- Examinar escenarios móviles de redes Bluetooth y verificar sus rangos de cobertura.
- Conocer el funcionamiento de los nodos PMPs y su importancia en las redes Bluetooth.
- Mostrar el intercambio de roles de dos dispositivos conectados a una piconet.
- Formar una topología de red Bluetooth avanzada (Scatternet).

2. INTRODUCCIÓN

En esta práctica se aprenderán escenarios avanzados de redes Bluetooth, donde tenemos topologías con gran número de nodos y varias aplicaciones simultáneas, así como el intercambio de archivos entre nodos bajo aplicaciones FTP y CBR y procedimientos de intercambio de roles entre dispositivos conectados.

En la librería UCBT se ha especificado como distancia máxima entre dispositivos Bluetooth cerca de 10 metros a la redonda, es decir la cobertura máxima de cada nodo será esa distancia. El objetivo es utilizar los nodos con las potencias mínimas, teniendo en consideración que por el hecho de ser dispositivos móviles, estos requieren un sistema de ahorro de energía.

3. MARCO TEORICO

Un dispositivo Bluetooth puede atender más de una conexión a la vez, y esto lo logra porque cuando se establece una piconet se usan ciertos canales en un esquema de salto de frecuencia, por lo que cada piconet tendrá un esquema propio. Estos dispositivos que participan en más de una piconet a la vez se los conoce como nodos PMP (Participante en Múltiples Piconets), estos nodos suelen desempeñar tareas importantes cuando de interconectar piconets se trata, es decir, forman un enlace o puente para la interconexión entre piconets. A lo que se conoce con el nombre de scatternet.

Muchas veces dentro del establecimiento de una piconet o incluso de una scatternet, los dispositivos requieren hacer un cambio de roles para poder mejorar su topología y establecer una conexión con la menor cantidad de problemas. A este procedimiento se lo conoce con el nombre de Role-Switch.

Un nodo puente puede tener roles de maestro o de esclavo, según como se realice la interconexión. Debido a esto y otros factores, las tasas de datos en este tipo de topología pueden disminuir según sea el nivel de interferencia o la cantidad de dispositivos y aplicaciones simultáneas que se atiendan. Lo bueno de este tipo de topología es que de alguna manera puede ampliar la cobertura entre dispositivos que se encuentren muy lejanos a los 10 metros.

4. MATERIALES Y EQUIPOS

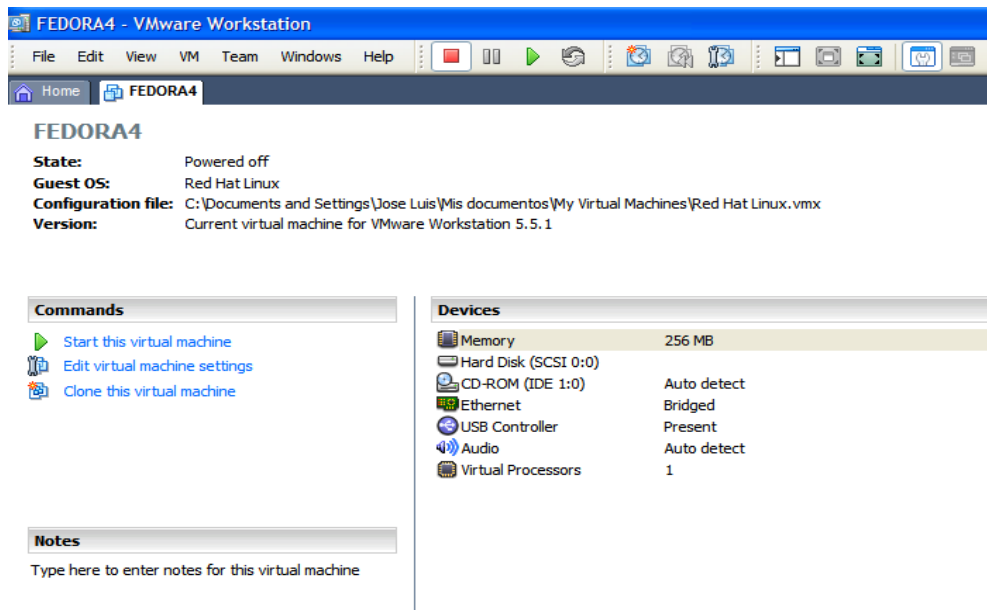
- PC con Fedora Core 4 y Network Simulator 2: Network Animator, librería UCBT modificada.

5. PROCEDIMIENTOS

Abriremos el emulador de sistemas operativos “VMware Workstation”, para esto haga doble click en el icono **VMware Workstation 5.5.1** en el escritorio o vaya a la siguiente dirección:

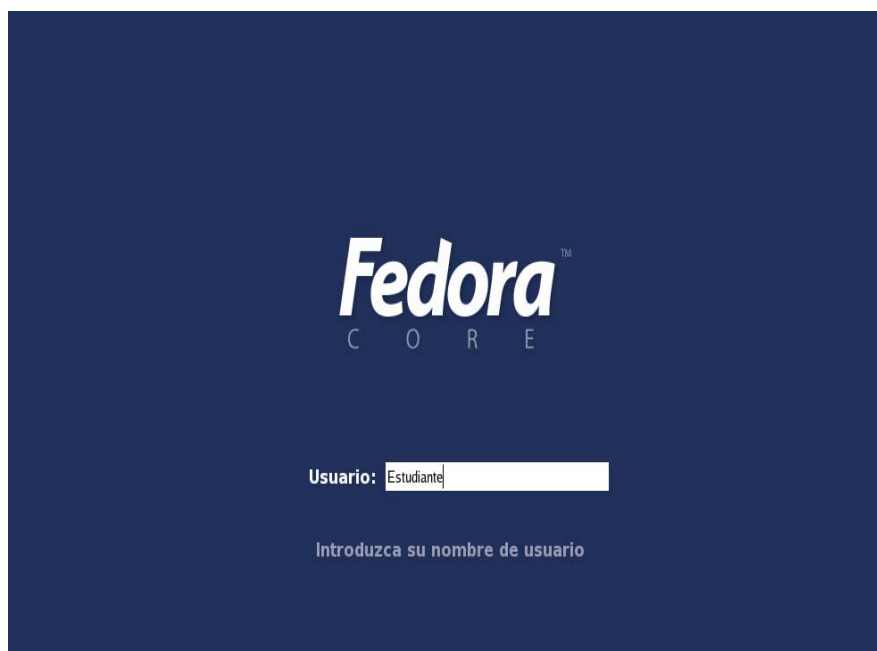
Inicio->Programas->VMware->VMware Workstation 5.5.1

aparecerá la siguiente ventana:

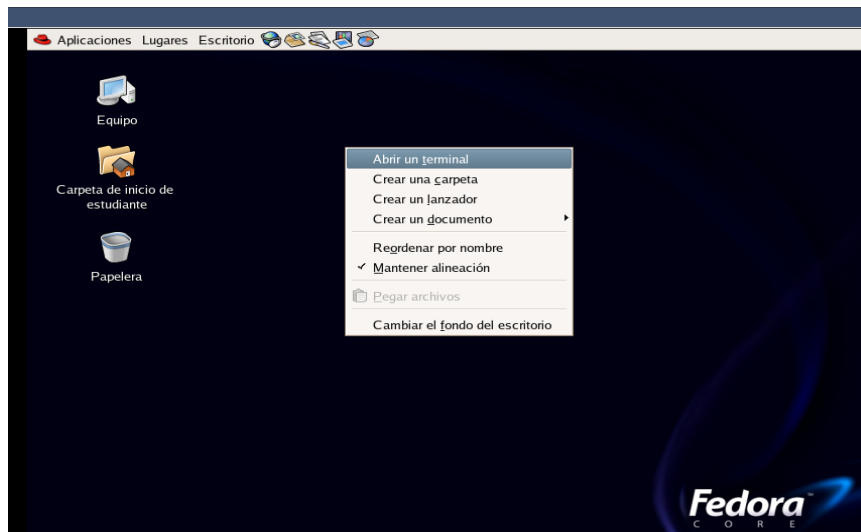


desde aquí haga click en “*Start this virtual machine*” para abrir el sistema operativo “Fedora Core 4”.

En la pantalla de inicio de sesión de Fedora Core 4 proceda a digitar en el campo usuario lo siguiente: Estudiante, como se muestra a continuación:



Luego, en el campo contraseña, proceda a digitar lo siguiente: Estudiante, y al finalizar presione “Enter”. Dentro del escritorio abrimos una ventana de terminal dando click derecho sobre el escritorio.

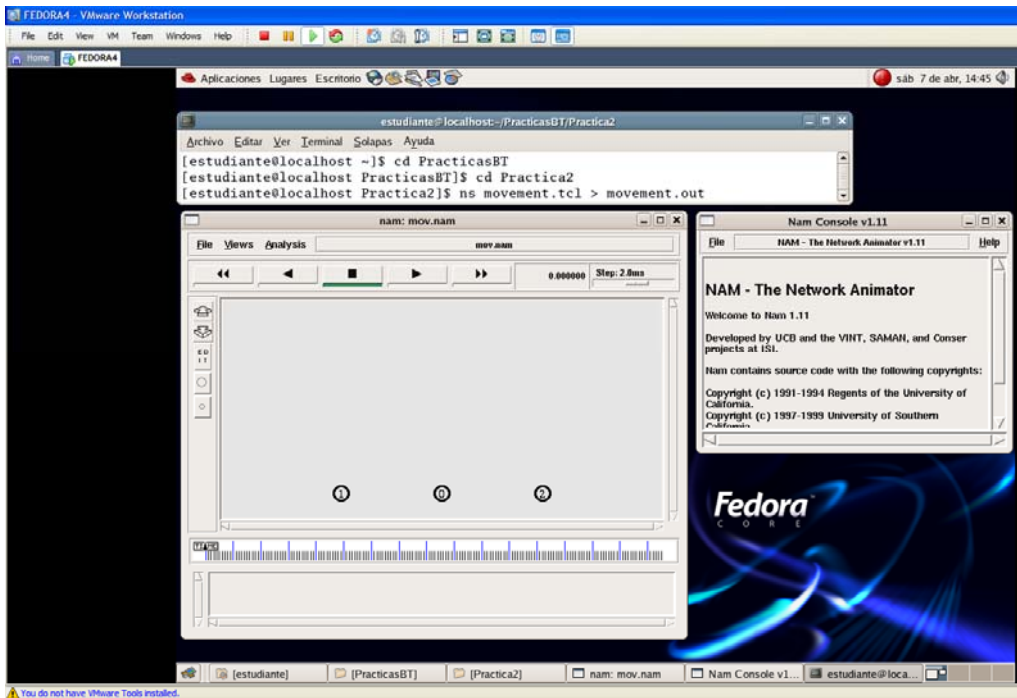


5.1 Simular una piconet con nodos móviles y analizar los rangos de cobertura.

En todo momento, los dispositivos conectados en una piconet deben estar dentro de un espacio físico en el cual puedan comunicarse, es decir, logren un intercambio efectivo de paquetes de información y control. A este espacio físico se lo denomina área de cobertura. Por definición, todos los nodos tienen un área de cobertura de 10 metros para potencias bajas, debido a esto, cuando los dispositivos han establecido una piconet significa que entre el dispositivo maestro y los dispositivos esclavos existen como máximo una distancia de 10 metros.

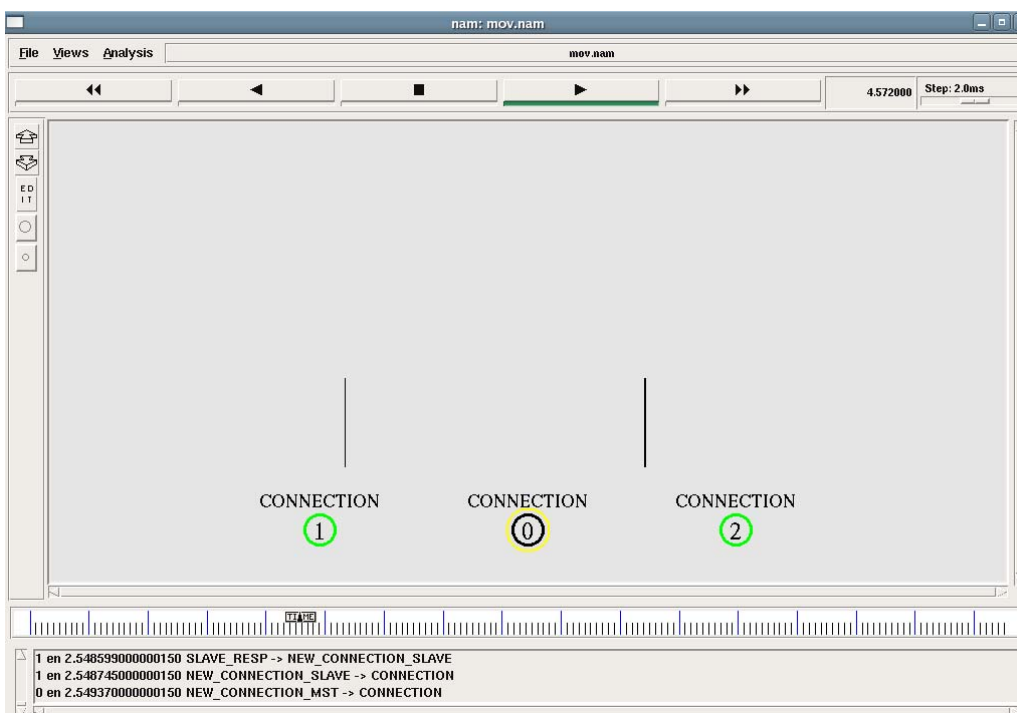
Una vez que tengamos la pantalla del terminal, ejecutaremos las siguientes sentencias:

- `cd PracticasBT`
- `cd Practica2`
- `ns movement.tcl >> movement.out`



Al presionar *play* en la presente simulación se ejecutará una piconet entre los nodos 0, 1 y 2 durante un tiempo determinado. Los comandos que se utilizarán son los siguientes:

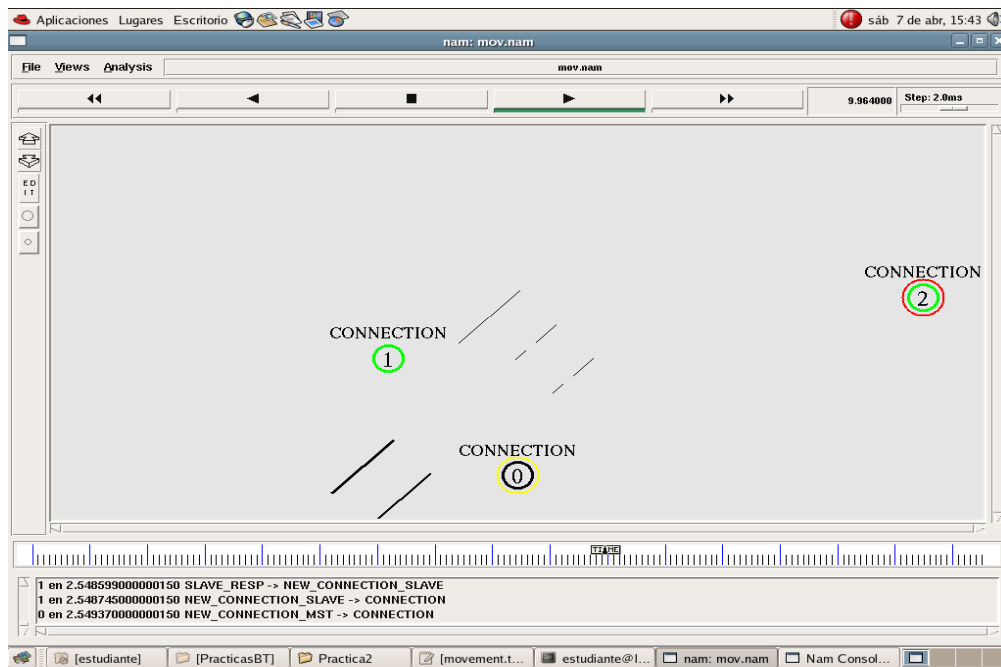
\$ns at 1.0 "\$node(0) make-bnep-connection \$node(2) DH5 DH3 noqos \$ifq1"
\$ns at 1.1 "\$node(0) make-bnep-connection \$node(1) DH5 DH3 noqos \$ifq"
\$ns at 4.0 "\$ftp0 start"
\$ns at 4.1 "\$ftp1 start"



Una vez que la piconet ha sido establecida, los nodos 1 y 2 empezarán a moverse alrededor del nodo 0 o maestro, de tal manera que el nodo 2 termine fuera del rango de cobertura de la piconet y el nodo 1 se mantenga en el rango de cobertura de la misma. Los comandos que se ejecutarán son los siguientes:

```
$ns at 4.7 "$node(1) setdest 6 10 1"
$ns at 4.8 "$node(2) setdest 22 8 5.5"
```

Estos comandos permiten el movimiento de los nodos 1 y 2. Al finalizar esta acción el nodo 2 se encontrará fuera del rango de cobertura y perderá toda transmisión pendiente con el nodo 0 como podemos apreciar en la siguiente figura:



Se recomienda modificar las posiciones finales de los nodos 1 y 2 para verificar si al mantenerse en el área de cobertura la transmisión de información entre estos nodos ocurre sin ningún problema.

Al finalizar la simulación conteste las siguientes preguntas:

- **¿En que momento el nodo 2 deja de recibir paquetes del nodo 0?**
5.845 segundos
- **Indique las posiciones iniciales y finales de los nodos 1 y 2.**
Nodo 1: 1,1 Nodo 2: 15,1 Nodo 1: 6,10 Nodo 2: 22,8

5.2 Simular dos piconets conectadas entre ellas por un nodo PMP.

Los nodos PMPs, o también conocidos como nodos participantes en múltiples piconets, son aquellos que establecieron conexión con más de una piconet a la vez. Las razones por las cuáles estos nodos llegaron a participar en varias piconets puede ser: por haber formado parte de un procedimiento de cambio de roles, por haber aceptado aplicaciones de dispositivos que están formando piconets diferentes, o por petición de una topología compleja de red Bluetooth, como una scatternet, haciendo que este dispositivo sirva de nodo puente entre piconets por su situación geográfica.

Una vez que tengamos la pantalla del terminal, ejecutaremos las siguientes sentencias:

- cd PracticasBT
- cd Practica2
- ns pmp.tcl >> pmp.out

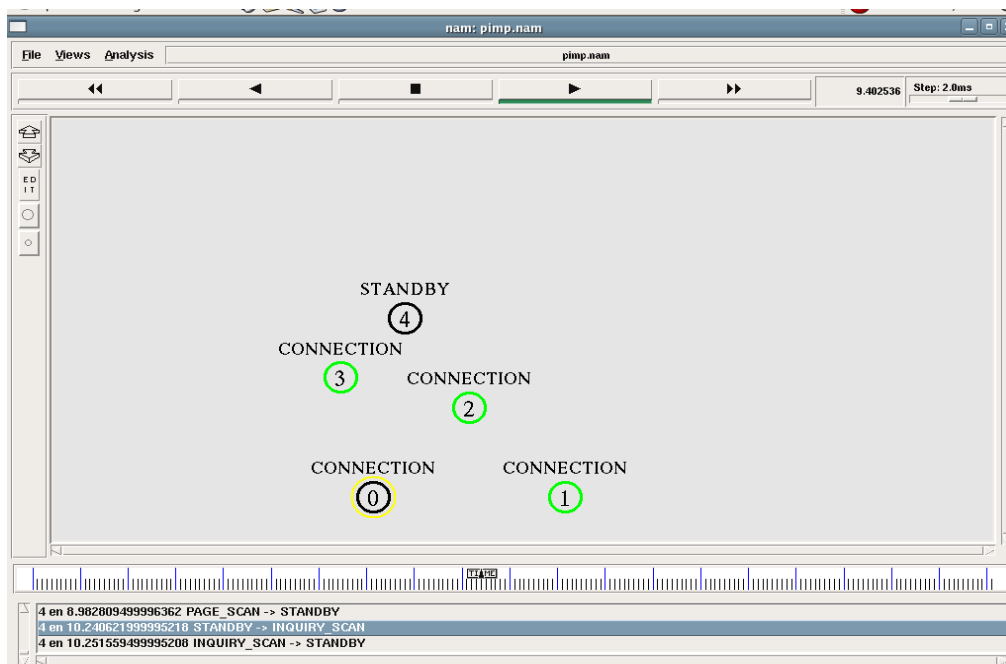
Al presionar play en esta simulación tendremos la formación una piconet inicial entre 0, 1, 2 y 3. Los comandos a utilizar son los siguientes:

```
$ns at 1.2 "$node(0) make-bnep-connection $node(1) DH5 DH3 none"
```

```
$ns at 3.4 "$node(0) make-bnep-connection $node(2) DH5 DH3 none"
```

```
$ns at 5.6 "$node(0) make-bnep-connection $node(3) DH5 DH3 none"
```

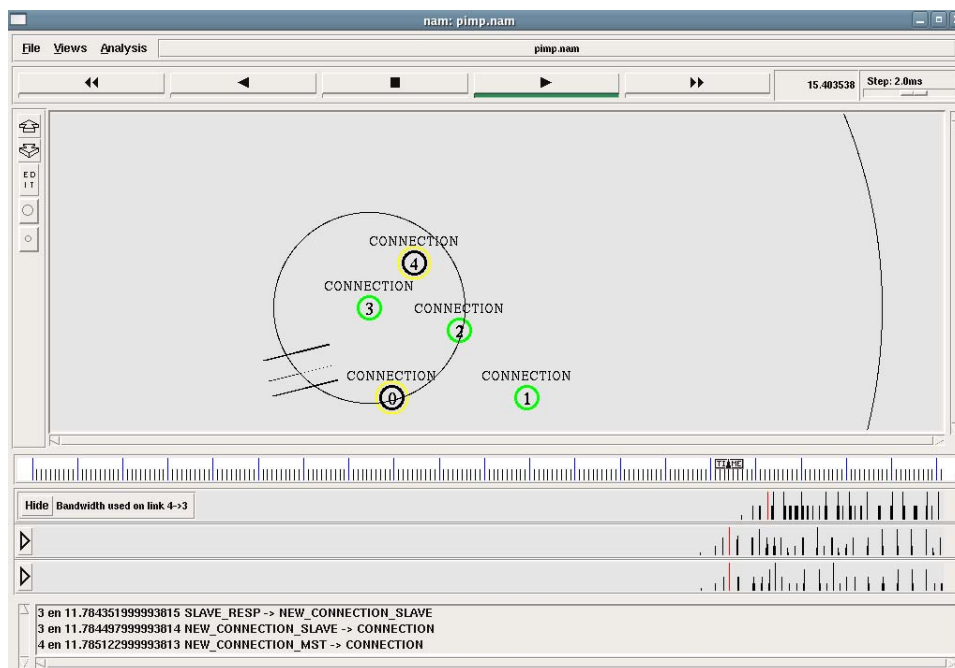
```
$ns at 14.5 "$ftp0 start"
```



Para conectar el nodo 4 con la piconet vamos a utilizar el nodo 3 como puente con el siguiente comando:

\$ns at 10.5 "\$node(4) make-br \$node(3) DH5 DH3 none"

Con esta sentencia creamos una nueva piconet entre el nodo 3 y el nodo 4 que a su vez se encuentra conectada con la piconet anterior gracias al nodo 3 que actúa como bridge o puente entre ambas piconets. En la siguiente figura se muestra una transferencia de archivos entre las dos piconets:



Se recomienda verificar la función del nodo puente escogiendo a otro nodo para que forme la nueva piconet con el nodo cuatro. Recuerde no escoger los nodos que participen en la transferencia de información entre las dos piconets.

Al finalizar la simulación conteste las siguientes preguntas:

- **¿Qué tipo de funciones realiza el nodo PMP?**
Realiza la función de “puente” entre piconets que quieran establecer comunicación entre ellas. Puede adoptar las siguientes combinaciones: maestro-esclavo y esclavo-esclavo.
- **¿En qué momento se establece la piconet entre los nodos 3 y 4?**
11,79 segundos de la simulación
- **¿Qué rol desempeña el nodo PMP en esta simulación?**
Permite la comunicación entre las dos piconets y participa como esclavo en ambas.

5.3 Simular el intercambio de roles entre dos dispositivos conectados en una piconet.

Este procedimiento es utilizado únicamente por dos dispositivos que estén conectados en una piconet, es decir, ambos dispositivos intercambiarán roles, dando como resultado que el dispositivo maestro se convierta en el esclavo y que el esclavo se convierta en maestro. Cuando ocurre este procedimiento se forma nuevamente la piconet, en otras palabras, el esquema de saltos entre canales se redefine según el nuevo dispositivo maestro.

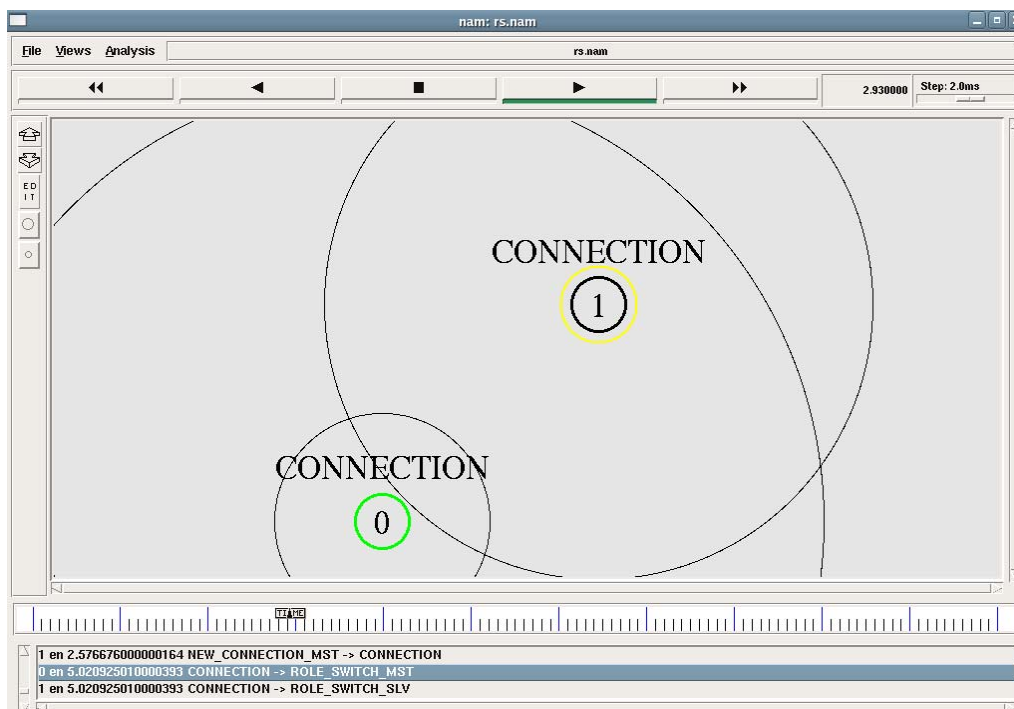
Si existiera el caso en el que uno de los dispositivos que va a realizar el cambio de roles tuviera alguna conexión con otro dispositivo, este enlace no se pierde y el dispositivo quedaría trabajando como un nodo participante en múltiples piconets.

Una vez que tengamos la pantalla del terminal, ejecutaremos las siguientes sentencias:

- cd PracticasBT
- cd Practica2
- ns rs.tcl >> rs.out

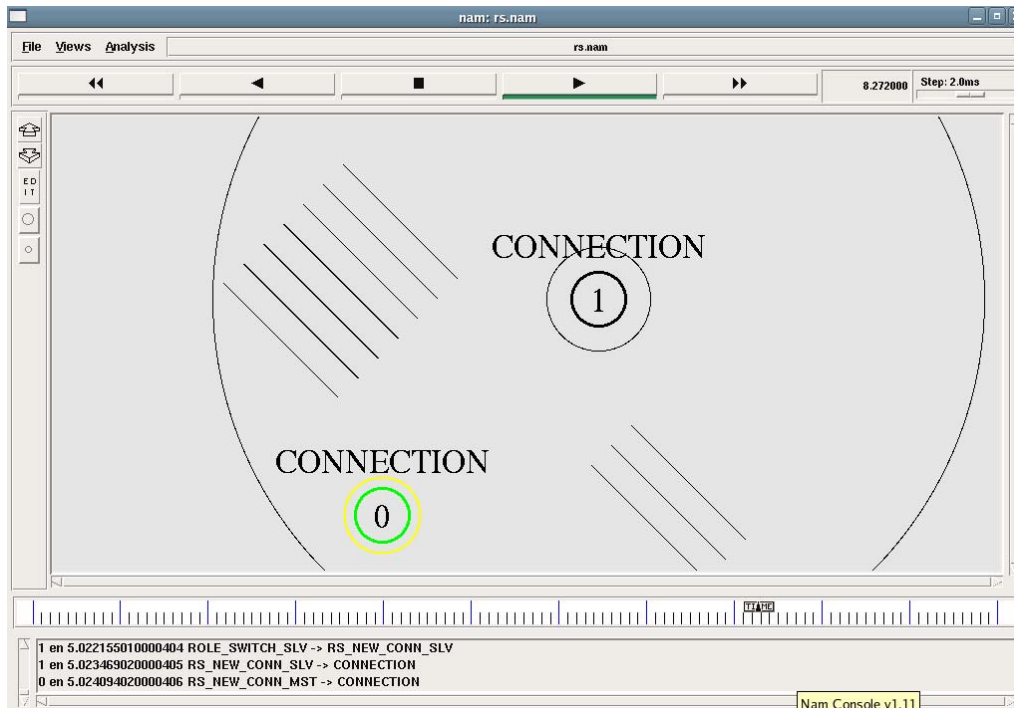
Al ejecutar esta simulación tendremos una piconet entre los nodos 1 y 0, siendo maestro y esclavo respectivamente, utilizando el siguiente comando:

\$ns at 0.1 "\$node(1) make-bnep-connection \$node(0) DH1 DH1"



Una vez establecida la piconet procedemos al intercambio de roles entre el nodo 0 y el nodo 1, así también se establecerá una transferencia TCP entre

ambos nodos teniendo ahora como nuevo maestro al nodo 0 y como nuevo esclavo al nodo 1 como lo observamos en la siguiente figura:



Al finalizar la simulación conteste las siguientes preguntas:

- Verificar los canales de frecuencia que se utilizan en la piconet antes de que exista el intercambio de roles.
- Verificar todos los estados y paquetes intercambiados en el momento del intercambio de roles.
- Verificar los canales de frecuencia que usa la nueva piconet. ¿Existe alguna diferencia?
- Verifique el CLK que utiliza la nueva piconet.

5.4 Simular una red Bluetooth avanzada (Scatternet).

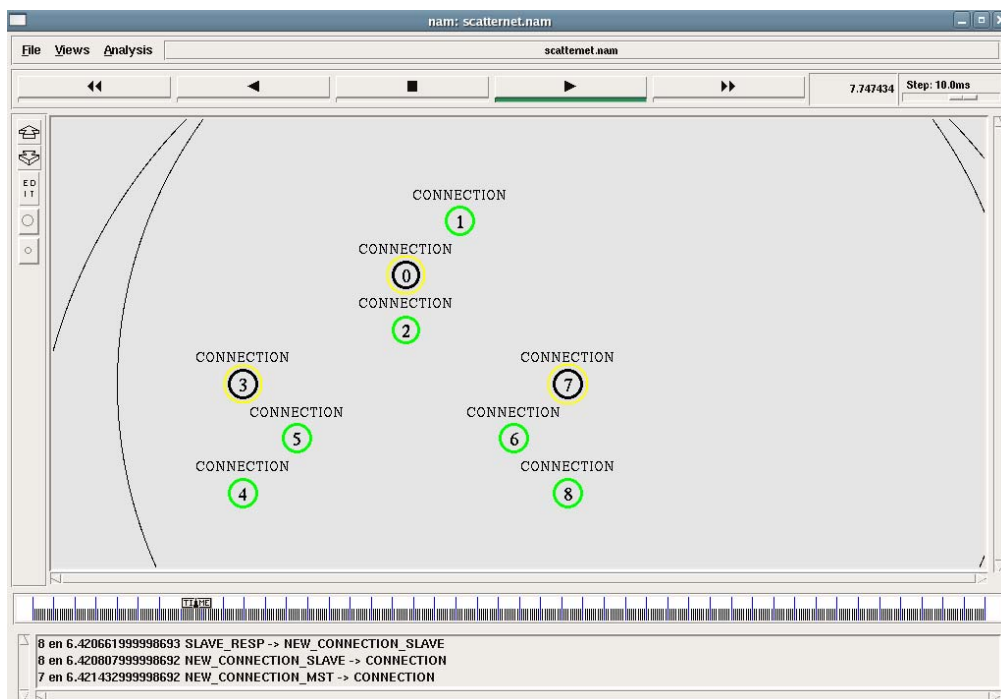
Las scatternets básicamente resultan de la interconexión de 2 o más piconets. Es una topología de red bastante compleja, ya que se pueden solapar ciertos canales e incurrir a niveles de interferencias considerables entre las piconets que se interconecten, así como problemas en la transmisión de datos, ancho de banda y sincronización de los dispositivos que participan en la scatternet. Además, los nodos que enlazan estas piconets, conocidos como nodos puentes, deben participar en ambas piconets según el rol que tengan en cada una.

Una vez que tengamos la pantalla del terminal, ejecutaremos las siguientes sentencias:

- cd PracticasBT
- cd Practica2
- ns scatternet.tcl >> scatternet.out

Al ejecutar esta simulación estableceremos tres piconets entre los nodos 0, 1 y 2 la primera, 3, 4 y 5 la segunda y 6, 7 y 8 la tercera con los siguientes comandos:

```
$ns at 1.2 "$node(0) make-bnep-connection $node(1) DH5 DH3 none"
$ns at 1.4 "$node(0) make-bnep-connection $node(2) DH5 DH3 none"
$ns at 2.2 "$node(3) make-bnep-connection $node(4) DH5 DH3 none"
$ns at 2.4 "$node(3) make-bnep-connection $node(5) DH5 DH3 none"
$ns at 3.2 "$node(7) make-bnep-connection $node(6) DH5 DH3 none"
$ns at 3.4 "$node(7) make-bnep-connection $node(8) DH5 DH3 none"
```



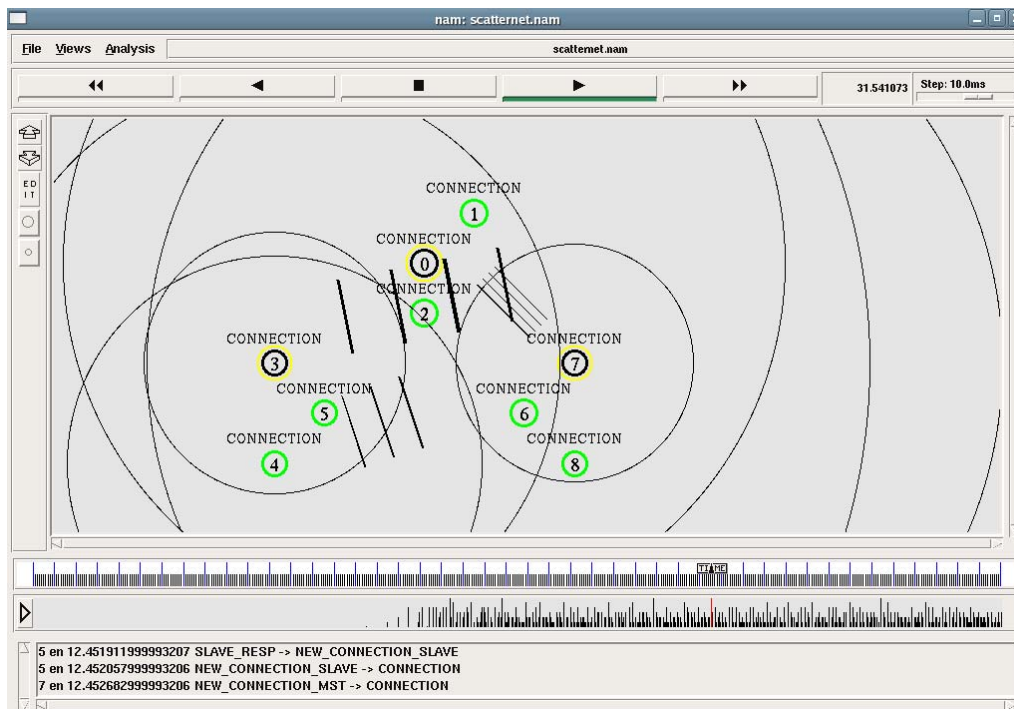
Luego procederemos a interconectar las tres piconets mediante dos nodos puente con el siguiente comando:

```
$ns at 10.0 "$node(3) make-br $node(2) DH5 DH3 none"
$ns at 11.0 "$node(7) make-br $node(5) DH5 DH3 none"
```

Finalizaremos con una transferencia de datos entre los nodos 0 y 7 para verificar que las piconets se encuentren interconectadas por medio del siguiente comando:

```
$ns at 15.0 "$ftp0 start"
```


En la siguiente figura vemos la transferencia de datos entre ambos nodos:



Se recomienda revisar el archivo tcl de esta simulación para modificar los nodos que participan en la transferencia de paquetes entre las tres piconets. Además verifique si los nodos puente siguen participando de este nuevo escenario.

BIBLIOGRAFIA

1. The VINT Project, The ns Manual, UC Berkeley, LBL, USC/ISI, and Xerox PARC,.Kevin Fall (kfall@ee.lbl.gov), Kannan Varadhan kannan@catarina.usc.edu), Editores; Enero 8, 2003.
2. BluetoothSIG, Specifications Documents, <http://www.bluetooth.com/Bluetooth/Learn/Technology/Specifications/> revisada el 4 de Enero 2006.
3. D. Agrawal, Q. Wang, UCBT Bluetooth Extension for NS2 at the University of Cincinnati, <http://www.ececs.uc.edu/~cdmc/ucbt/ucbt.html> revisada el 15 de Enero de 2006.

APÉNDICE D.

Perfiles definidos hasta el momento en la especificación Bluetooth.

Entre los perfiles definidos hasta el momento tenemos:

- **A2DP:** Describe la manera en la cual audio de calidad estéreo es transmitido desde una fuente multimedia hacia un receptor.
- **AVCTP:** Describe los mecanismos de transporte necesarios para intercambiar mensajes en lo que respecta al control de dispositivos de audio y video.
- **AVDTP:** Describe los procedimientos de negociación, establecimiento y transmisión de audio y video.
- **AVRCP:** Diseñado para proveer una interfaz estándar en lo que se refiere al control de TVs, equipos Hi - Fi, u otros por medio de un control remoto capaz de controlar todo el equipo de audio y video al cual el usuario tiene acceso.
- **BIP:** Define como un dispositivo capaz de manejar imágenes puede ser controlado de forma remota.
- **BBP:** Permite a dispositivos enviar texto, e - mails, vCards o imágenes a impresoras listas para trabajar.
- **BNEP:** Permite a los paquetes IP ser transportados dentro de la carga útil de los paquetes L2CAP. BNEP es usado por el perfil PAN.
- **CIP:** Define cómo la señalización ISDN puede ser transmitida a través de una conexión inalámbrica Bluetooth.
- **CTP:** Define como un teléfono inalámbrico puede ser implementado en un enlace inalámbrico Bluetooth.
- **DUN:** Provee un estándar para acceder al Internet y otros servicios dial - up por medio de tecnología Bluetooth.
- **ESDP:** Define como los dispositivos universales “plug and play” se conectan a través de la tecnología inalámbrica Bluetooth.
- **FAX:** Define como un dispositivo FAX gateway puede ser usado por un terminal.
- **FTP:** Define como las carpetas y archivos en un dispositivo servidor pueden ser examinadas desde un dispositivo cliente.

- **GAP:** Provee la base para todos los otros perfiles y define medios consistentes para establecer un enlace de banda base entre dispositivos Bluetooth conectables. GAP asegura un alto grado de interoperabilidad entre aplicaciones y dispositivos.
- **GAVDP:** Provee las bases para el funcionamiento de los perfiles A2DP y VDP.
- **GOEP:** Usado para transmitir un objeto de un dispositivo a otro. El objeto puede ser una imagen, un documento, una tarjeta de negocios, etc.
- **HFP:** Describe como un dispositivo gateway puede ser usado para realizar y recibir llamadas desde un dispositivo “manos libres”.
- **HCRP:** Permite a dispositivos como impresoras y scanners conectarse con dispositivos como laptops y PCs de escritorio sin la necesidad de un cable físico.
- **HSP:** Describe como un auricular bluetooth se debería comunicar con una PC o con otro dispositivo Bluetooth como un teléfono móvil.
- **HID:** Define los protocolos, procedimientos y características que serán usados por dispositivos HID tales como teclados, apuntadores, controles de juegos y de monitoreo.
- **ICP:** Define como dos teléfonos móviles Bluetooth en la misma red se puedan comunicar directamente sin usar la red telefónica pública.
- **OBEX:** Protocolo de intercambio de datos diseñado por IrDA que tiene como fin abstraer y normalizar la forma en que se comunican los pequeños dispositivos electrónicos que van surgiendo con necesidad de comunicarse con otros.
- **OPP:** Define los roles de empuje del servidor y del cliente.
- **PAN:** Describe cómo dos o más dispositivos Bluetooth pueden formar una red Ad -Hoc y como el mismo mecanismo puede ser usado para acceder a una red remota a través de un punto de acceso.
- **RFCOMM:** Proporciona emulación de puertos seriales a través del protocolo L2CAP. Este protocolo se basa en el estándar de la ETSI denominado TS 07.10. RFCOMM es un protocolo de transporte sencillo, con soporte para hasta 9 puertos seriales RS - 232. El protocolo RFCOMM permite hasta 60 conexiones simultáneas entre dos dispositivos Bluetooth.
- **SDP:** Permite a las aplicaciones cliente descubrir la existencia de diversos servicios proporcionados por uno o varios servidores de aplicaciones, junto con los atributos y propiedades de los servicios que se ofrecen. Estos atributos de servicio incluyen el tipo o clase de servicio ofrecido y el mecanismo o la información necesaria para utilizar dichos servicios.
- **SDAP:** Describe como una aplicación debería utilizar el protocolo SDP para descubrir servicios en un dispositivo remoto.

- **SAP:** Permite a dispositivos como teléfonos de automóviles con transceivers GSM conectarse a una tarjeta SIM de un teléfono Bluetooth.
- **SPP:** Permite que dispositivos Bluetooth realicen simulación de RS232. El escenario cubierto por este perfil trata con aplicaciones comerciales que utilizan Bluetooth como un sustituto del cable, utilizando una capa de abstracción que representa un puerto serie virtual.
- **SYNC:** Usado en conjunto con GOEP para habilitar la sincronización del calendario y la información de direcciones entre dos dispositivos Bluetooth.
- **TCS - Binario o TCP:** Define como un dispositivo Bluetooth puede ser usado como teléfono inalámbrico.
- **VDP:** Define como un dispositivo Bluetooth transmite video a través de tecnología inalámbrica Bluetooth.
- **WAP:** Define como el protocolo de aplicación inalámbrica puede funcionar a través de un enlace de tecnología inalámbrica Bluetooth.