CONSTRUCCIÓN DE UNA ARQUITECTURA PARA LA INTEROPERABILIDAD DE LA TECNOLOGÍA BLUETOOTH CON REDES IP CABLEADAS PARA TRANSPORTE DE VOZ

ANEXO B

JUAN PAULO GUZMÁN FLÓREZ
DARYAN FRANCISCO REINOSO ROJAS

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Popayán
2007

Anexo B. Instalación y Manejo paquete de simulación UCBT (*Bluetooth extension for NS2 at the University of Cincinnati*).

En este documento se examinaran aspectos concernientes a la instalación y uso del paquete de simulación UCBT creado por el profesor Qihe Wang de la Universidad de Cincinnati.

1. Instalación

Aunque el paquete UCBT es una extensión de la herramienta NS-2, éste debe ser instalado en conjunto y al mismo tiempo que se instala NS-2. Se deben ejecutar los procedimientos contenidos en la sección 1.1. del Anexo A.

1.1. Descarga del paquete de instalación UCBT

Se deben descargar el archivo necesario que contiene el paquete de simulación UCBT.

```
$ tar -xzvf ns-allinone-2.29.3.tar.gz
$ cd ns-allinone-2.29/ns-2.29
$ wget https://www.ececs.uc.edu/~cdmc/ucbt/src/ucbt-0.9.9.2a.tar.gz
$ tar zxvf ucbt-0.9.9.2a.tar.gz
$ cd ucbt-0.9.9.2a
$./install-bt (con la opción -t o -d se habilita el debug)
```

Nota: El procedimiento de instalación de UCBT se debe realizar sobre una carpeta que contenga una NS-2 **no** instalada previamente.

Una vez instalados los paquetes se debe generar un mensaje de salida que indique que tanto el paquete NS-2 como el UCBT han sido instalados satisfactoriamente. En la Figura B1 se ilustra lo mencionado.

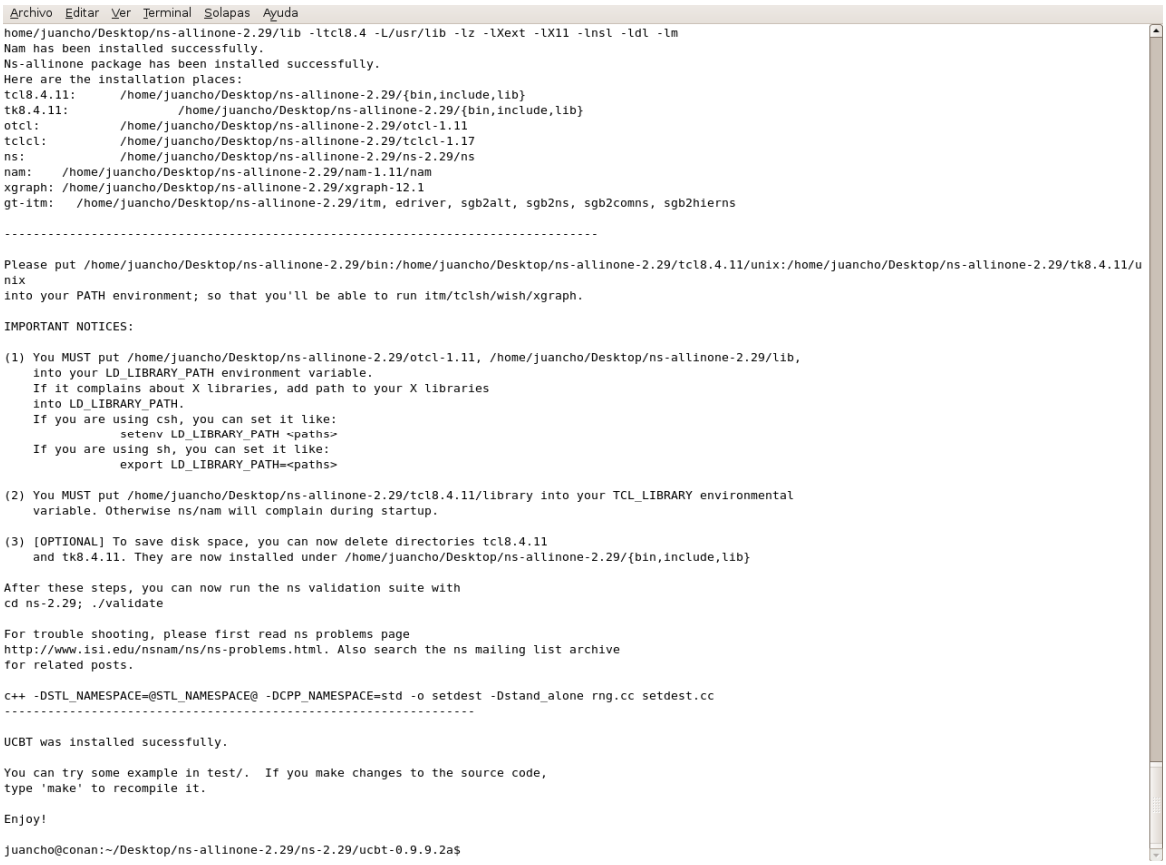

Figura B1. Instalación exitosa de la herramienta de simulación NS-2 y el paquete UCBT

1.2. Ejecución de scripts UCBT

La forma de correr un script es similar a la de NS-2. Sin embargo el archivo de salida debe ser especificado directamente sobre la línea de comandos y no en el *script* como tal.

Para verificar la funcionalidad de la instalación se pueden ejecutar algunos ejemplos incluidos en al paquete así:

$ cd test.
$ ns expample.tcl > example.out

Esto generará un archivo de salida llamado example.out que será de gran utilidad al momento de realizar el análisis de la simulación.

Desafortunadamente el paquete UCBT, no tiene soporte gráfico, es decir, no es posible generar archivos de salida de extensión .nam que puedan ser interpretados por el programa respectivo nam.

1.3. Estructura de los script de simulación UCBT

Para analizar detalladamente la estructura y contenido de un script de simulación UCBT se dará una explicación línea por línea de un ejemplo sencillo de simulación.

En la generación del script se debe utilizar un editor de texto simple tal como el NotePad de Windows o el nano bajo Linux.

[ejemplo.tcl]

```
#Los comentarios deben estar precedidos del signo #

# Se crea la instancia de simulación
set ns [new Simulator]

#Se define el tipo de protocolo de Segundo nivel a usar (BNEP)
$ns node-config -macType Mac/BNEP

#Se define el número de nodos de simulación
set val(nn) 2;

#Se crea un bucle que cree los nodos automáticamente dependiendo del
#número de nodos especificados anteriormente.
for {set i 0} {$i < $val(nn) } {incr i} {
    set node($i) [$ns node $i ]
    $node($i) rt AODV
    $node($i) SchedAlgo DRR
    set lmp [$node($i) set lmp_]
    $node($i) set-statist 7 15 1 adjust-l2cap-hdr

#En este bucle de definen parámetros propios de los nodos tales como
#algoritmos de enrutamiento ad-hoc, algoritmos de temporización, y
#parámetros propios de banda base Bluetooth

#Adicionalmente se definen las ventanas de tiempo para los estados
inquire #y page propios de Bluetooth

    if {$i > 0 } {
      [$node($i) set bb_] set T_w_inquiry_scan_ 36
      [$node($i) set bb_] set T_inquiry_scan_ 4096
      [$node($i) set bb_] set T_w_page_scan_ 4064
      [$node($i) set bb_] set T_page_scan_ 4096
    }
}

#Se inicializan los nodos creados
$ns at 0.0002 "$node(0) on"
$ns at 0.0005 "$node(1) on"

$node(0) trace-all-NULL on
$node(0) trace-all-POLL on
```

```
#En esta sección se especifican todos los flujos de tráfico que se
llevarán #a cabo durante la simulación, así como también los nodos
encargados de #generar y recibir cada uno de ellos

set udp0 [new Agent/UDP]
$ns attach-agent $node(0) $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set packetSize_ 200
$cbr0 set rate_ 64k
$cbr0 set random_ 1

#Particularmente en este ejemplo se define un flujo CBR de 64Kbps con un
#tamaño de paquete de 200bytes que simula un flujo constante de tráfico
RTP #con transportado sobre el protocolo UDP. Dicho flujo de datos es
asociado #al nodo 0.

#También se define el nodo encargado de recibir el flujo CBR. En este
caso #el nodo 0. Luego se conecta la fuente con el destino del flujo
set null [new Agent/Null]
$ns attach-agent $node(1) $null
$ns connect $udp0 $null

#En esta parte, se establece la conexión Bluetooth entre los dos
#dispositivos, de modo que tengan capacidades BNEP, para el perfil PAN.
$ns at 1.01 "$node(0) make-bnep-connection $node(1) DH1 DH1 noqos" ;

#Una vez establecida la conexiÃ³n se inicializa el flujo CBR y luego de
un #tiempo específico se detiene.

$ns at 4 "$cbr0 start"
$ns at 9 "$cbr0 stop"

#Se declara la rutina de finalización donde se ejecuta el comando de
traza #para generar los datos en el archivo de salida.

proc finish {} {
        global node
      $node(0) print-all-stat
        exit 0
}

#Se llama al procedimiento de finalización para que se ejecute la
#simulación

$ns at 12 "finish"
$ns run
```

1.4. Estructura de los archivos de salida UCBT

Ya explicados los componentes y estructura básica de un script de simulación UCBT procedemos a explicar los datos contenidos dentro de los archivos de salida generados durante la simulación donde se pueden encontrar detalles específicos de transmisión de paquete tanto de banda base como de paquetes BNEP.

En general en los archivos de salida de la simulación se encontrará un registro de todos y cada uno de los paquetes que han sido transmitidos y recibidos durante toda la simulación.

Los paquetes pueden ser de dos tipos, paquetes banda base o paquetes BNEP.

El formato de traza de los paquetes banda base se ilustra en detalle en la Figura B2.

Tx/Rx/ Collision/Lost	bdaddr	state	src:slot	dst:slot	pktType	lt_addr
accesscode	frequency	arqno	Seqno	transmission Count	size(in bits)	
timestamp	packetID:frag	# comment				

Figura B2. Formato de traza, paquetes banda base

Los formatos de traza de los paquetes BNEP se ilustran en las Figuras B3 y B4.

Para los paquetes enviados:

bv	bdaddr	ip_src: port_src	- >	ip_dst: port_dst	timestamp	nextHop	hopcount	seqno_ forthisflow	Size (bytes)

Figura B3. Formato de traza, paquetes BNEP enviados

- bv: paquete enviado.
- bdaddr: dirección Bluetooth del dispositivo
- ip_src:port_src: dirección IP del nodo fuente : puerto en el nodo fuente.
- ip_dst:port_dst: dirección IP del nodo destino : puerto en el nodo destino.
- timestamp: marca de tiempo.
- nextHop: siguiente salto.
- hopcount: contador de saltos.
- seqno_forthisflow: número de secuencia para el flujo específico.
- size(in_bytes): tamaño del paquete en bytes.

Para los paquetes recibidos:

b^	bdaddr	ip_src: port_src	- >	ip_dst: port_dst	timestamp	delay	hopcount	seqno_ forthisflow	Size (bytes)

Figura B4. Formato de traza, paquetes BNEP recibidos

- b^ : paquete recibido
- bdaddr: dirección Bluetooth del dispositivo
- ip_src:port_src: dirección IP del nodo fuente : puerto en el nodo fuente.
- ip_dst:port_dst: dirección IP del nodo destino : puerto en el nodo destino.
- timestamp: marca de tiempo.
- delay: retardo.
- hopcount: contador de saltos.
- seqno_forthisflow: número de secuencia para el flujo específico.
- size(in_bytes): tamaño del paquete en bytes.