



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

**"IMPLEMENTACIÓN DE UNA HERRAMIENTA DIDÁCTICA  
PARA LA SIMULACIÓN DE REDES IEEE 802.11B Y 802.11G  
ORIENTADA A PRÁCTICAS EN EL LABORATORIO DE  
TELECOMUNICACIONES"**

**TESIS DE GRADO**

Previo a la obtención del Título de:

**INGENIERO EN ELECTRONICA Y  
TELECOMUNICACIONES**

Presentado por:

Jerry Alfonso Barona Fajardo  
Rosa Fernanda Serpa Román  
Hugo Ernesto Torres Gillade

Guayaquil – Ecuador

2007

## AGRADECIMIENTO

A todas las personas que de una u otra forma hicieron posible que este trabajo se hiciera realidad en especial a nuestro Director de Tesis el Ing. Luis Fernando Vásquez por su invaluable ayuda.

## DEDICATORIA

A Dios

A nuestros padres

A nuestros hermanos

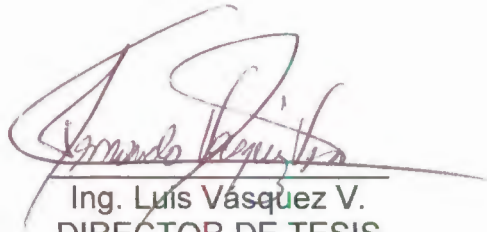
A mi esposo

## TRIBUNAL DE GRADO



---

Ing. Holger Cevallos U.  
SUBDECANO DE LA FIEC  
PRESIDENTE



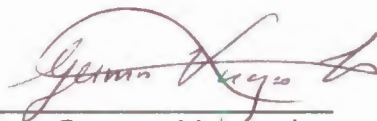
---

Ing. Luis Vásquez V.  
DIRECTOR DE TESIS



---

Ing. Rebeca Estrada P.  
VOCAL PRINCIPAL



---

Ing. German Vargas L.  
VOCAL PRINCIPAL

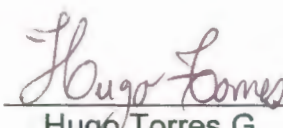


## DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesis de Grado, nos corresponden exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL”.

  
Jerry Barona F.

  
Rosa Serpa R.

  
Hugo Torres G.

## RESUMEN

En el primer capítulo se describe la problemática existente en lo que concierne al análisis y simulación de la tecnología inalámbrica IEEE 802.11b y 802.11g, por lo que no existe una herramienta didáctica para el aprendizaje de esta tecnología en la Facultad, basándonos en los simuladores existentes en ese tipo de redes.

Dentro del ámbito de los simuladores el principal problema es la interpretación de los resultados para lo cual nosotros hemos adaptado un conjunto de herramientas gráficas y textuales. Esto lo hacemos utilizando el simulador y un lenguaje de programación que nos ayudara a una mejor interpretación de los resultados

En el segundo capítulo detallamos los estándares de la tecnología 802.11b y 802.11g, sus variaciones y el tipo de configuraciones que éstos utilizan, así como las situaciones internas de su funcionamiento y aplicaciones.

En el tercer capítulo describimos el funcionamiento de nuestro proyecto así como todas las herramientas que fueron necesarias para su implementación, llegando así a encontrar una solución didáctica al resultado de las simulaciones. El simulador en el cual trabajamos es el Simulador de Redes (Network Simulator versión 2 NS2), y para la simulación usamos la librería 802.11 que viene incluida en el simulador, así como también el programa Animador de Redes (Network Animador NAM) que es una herramienta para interpretar la visualización gráfica de las simulaciones, además hacemos uso del lenguaje de programación AWK que nos permitirá procesar los resultados de las simulaciones y nos mostrará una tabla estadística de los principales parámetros a analizar.

En el cuarto capítulo detallamos los casos más frecuentes dentro de las redes 802.11b y 802.11g con los resultados utilizando nuestra herramienta didáctica, así como sus configuraciones, en las cuales generaremos las guías necesarias tanto para las prácticas como para el tutor. A su vez detallamos todos los equipos necesarios para poder implementar nuestra herramienta didáctica a manera de prácticas dentro de un laboratorio de telecomunicaciones.

# ÍNDICE GENERAL

	Pág.
RESUMEN.....	VI
ÍNDICE GENERAL.....	VII
ÍNDICE DE FIGURAS.....	VIII
GLOSARIO DE TÉRMINOS.....	IX
INTRODUCCIÓN.....	1
CAPÍTULO 1	
1. ANTECEDENTES DEL PROYECTO.....	3
1.1. Identificación de la problemática.....	4
1.1.1. Tecnologías utilizadas.....	6
1.1.2. Limitaciones operacionales.....	6
1.2. Simuladores disponibles.....	7
1.3. Justificación para los cambios.....	8
CAPÍTULO 2	
2. TEORÍA.....	10

2.1. Estándar IEEE 802.11.....	11
2.1.1. Historia .....	11
2.1.2. Arquitectura .....	12
2.1.3. Técnicas de Acceso .....	15
2.1.4. Protocolos.....	21
2.1.5. Formato de la Trama.....	34
2.2. Variaciones de los estándares IEEE 802.11b y del IEEE 802.11g ....	37

### CAPÍTULO 3

3. DESCRIPCION DEL PROYECTO .....	52
3.1. Solución planteada .....	52
3.2. Descripción del Programa NS2, la Librería 802.11 y la Herramienta para el Tratamiento de Datos.....	54
3.3. Funcionamiento General y Diagrama de Bloques.....	58

### CAPÍTULO 4

4. IMPLEMENTACIÓN DEL PROYECTO .....	61
4.1. Simulación de redes 802.11b y 802.11g.....	61
4.1.1. Procedimientos para la creación de una red inalámbrica.....	62
4.1.2. Simulación de modo Ad-hoc.....	71
4.1.3. Simulación de modo infraestructura .....	76
4.2. Resultados de la Simulación.....	80
4.3. Descripción de la guía de las prácticas y del manual del tutor .....	93

4.4. Costos de Implementación..... 96

CONCLUSIONES Y RECOMENDACIONES..... 98

**APÉNDICES**

Apéndice A: manual de NS2

Apéndice B: Código Fuente de las Simulaciones

Apéndice C: Manual De Las Practicas

Apéndice D: Manual Del Tutor

**BIBLIOGRAFÍA**

## ÍNDICES DE FIGURAS

	Pág.
<i>Figura 2.1: 802.11 y Modelo OSI</i> .....	13
<i>Figura 2.2: Modo Infraestructura</i> .....	15
<i>Figura 2.3: Modo Ad-Hoc</i> .....	15
<i>Figura 2.4: Configuración NAV entre dos Estaciones</i> .....	21
<i>Figura 2.5: Trama MSDU</i> .....	23
<i>Figura 2.6: Algoritmo Backoff Exponencial</i> .....	27
<i>Figura 2.7: Formato de la Trama 802.11</i> .....	35
<i>Figura 2.8: Canales de la Banda ISM de 2.4GHz</i> .....	39
<i>Figura 2.9: Formato del PPDU con PLCP largo</i> .....	46
<i>Figura 2.10: Formato del PPPDU con PLCP corto</i> .....	47
<i>Figura 2.11: Formato de la Trama OFDM</i> .....	50
<i>Figura 3.1: Diagrama de Bloques del Proyecto</i> .....	58
<i>Figura 4.1: Simulación de una Red Ad-hoc 802.11b</i> .....	81
<i>Figura 4.2: Instante en el que el Nodo 1 Hace de Puente</i> .....	82



<i>Figura 4.3: Tabla de Resultados de una Red Ad-hoc 802.11b Utilizando el Archivo AWK .....</i>	<i>83</i>
<i>Figura 4.4: Simulación de una Red Ad-hoc 802.11g .....</i>	<i>85</i>
<i>Figura 4.5: Tabla de Resultados de una Red Ad-hoc 802.11g Utilizando el Archivo AWK .....</i>	<i>86</i>
<i>Figura 4.6: Simulación de una Red Infraestructura 802.11b .....</i>	<i>87</i>
<i>Figura 4.7: Tabla de Resultados de una Red Infraestructura 802.11b Utilizando el Archivo AWK .....</i>	<i>88</i>
<i>Figura 4.8: Simulación de una Red Infraestructura 802.11g .....</i>	<i>89</i>
<i>Figura 4.9: Nodo 3 Realizando Transferencia de Paquetes a Través del Foreign Agent .....</i>	<i>90</i>
<i>Figura 4.10: Transmisión del Nodo 0 al Nodo 3 .....</i>	<i>91</i>
<i>Figura 4.11: Tabla de Resultados de la Red Infraestructura 802.11g – Nodo 3 a Nodo 0 .....</i>	<i>92</i>
<i>Figura 4.12: Tabla de Resultados de la Red Infraestructura 802.11g – Nodo 0 a Nodo 3 .....</i>	<i>92</i>



## GLOSARIO DE TÉRMINOS

Este es el glosario de términos que utilizamos en el presente documento, a lado su debido significado:

**ACK:** Acknowledge, reconocimiento

**AODV:** Ad-hoc On-Demand Distance Vector, vector de distancia en demanda Ad-hoc

**AP:** Access Point, punto de acceso

**API:** Application Programming Interface, interfaz de programación de aplicaciones

**BER:** Bit error rate, tasa de error de bit

**BPSK:** Binary Phase Shift keying, desplazamiento de fase binaria

**BS:** Base Station, estación base

**BSS:** Basic Service set, conjunto de servicio básico

**CBR:** Constant Bit Rate, tasa constante de bit

**CCA:** Clear Channel Assessment, evaluación de canal libre

**CCK:** Complementary Code Keying, codificación de código complementaria

**CONSER:** Collaborative Simulation for Education and Research, simulación colaborativa para la educación e investigación

**CRC:** Cyclic Redundancy code, código de redundancia cíclica

**CSMA/CA:** Carrier Sense Multiple Access with Collision Avoidance, acceso múltiple por detección de portadora con evasión de colisión

**CTS:** Clear to send, libre para enviar

**DARPA:** Defense Advanced Research Projects Agency, Agencia de Investigación de Proyectos Avanzados de Defensa

**DIFS:** Distributed coordination function inter frame space, espacio entre trama de la función de coordinación distribuida

**DSDV:** Destination Sequenced Distance Vector, vector de distancia por secuencia de destino

**DSSS:** Direct sequence spread spectrum, espectro ensanchado por secuencia directa

**DTIM:** Delivery Traffic Indication Message, mensaje de indicación de tráfico de entrega

**EIFS:** Extended inter frame space, espacio entre trama extendido

**ERP:** Extended rate PHY, velocidad extendida de capa física

**ESS:** Extended Service Set, conjunto de servicio extendido

**FCC:** Federal Communications Commission, Comisión Federal de Comunicaciones

**FHSS:** Frequency Hopping Spread Spectrum, espectro ensanchado por salto de frecuencia

**FTP:** File Transfer Protocol, protocolo de transferencia de archivo

**GB:** Giga byte

**GHz:** GigaHertz

**GOD:** General Operations Director, director general de operaciones

**Hz:** Hertz

**IBSS:** Independent Service Set, Conjunto de Servicio Independiente

**ICIR:** Institute for Communications and Information Research, Instituto para la Investigación de Comunicaciones e Información

**IEEE:** Institute of Electrical and Electronics Engineers, Instituto de Ingenieros Eléctricos y Electrónicos

**IFQ:** Interface Queue, interface de cola

**IFS:** Inter frame space, espacio entre tramas

**ISM:** Industrial, Scientific and Medical, Industrial, científico y medico

**Kbps:** Kilobits per second, Kilobits por segundo

**KHz:** KiloHertz

**LAN:** Local Area Network, red de área local

**LL:** Link Layer, capa de enlace

**MAC:** Medium Access Control, control de acceso al medio

**MB:** Mega byte

**Mbps:** Megabits per second, Megabits por segundo

**MHz:** MegaHertz

**MPDU:** MAC Sublayer Protocol Data Unit, unidad de datos de protocolo de subcapa MAC

**MSDU:** MAC Service Data Unit, unidad de datos de servicio MAC

**MSps:** MegaSymbols per second, MegaSímbolos por segundos

**NAV:** Network allocation vector, vector de ubicación de red

**NSF:** National Science Foundation, Fundación Nacional de Ciencias

**OFDM:** Orthogonal frequency-division multiplexing, múltiplexación por división de frecuencias ortogonales

**OSI:** Open system interconnection, interconexión de sistemas Abiertos

**PBCC:** Packet Binary Convolutional Code, código convolucional de paquete binario

**PHY:** Physical Layer, capa física

**PIFS:** Point coordination function inter frame space, espacio entre trama de la función de coordinación de punto

**PLCP:** Physical Layer convergence procedure, Procedimiento de convergencia de capa física

**PMD:** Physical Medium dependent, dependencia del medio físico

**PPDU:** PHY Protocol Data Unit, unidad de datos de protocolo de capa física

**PRNG:** Pseudo Random Number Generator, Generador de números pseudo aleatorios

**PSDU:** PLCP Service Data Unit, unidad de datos de servicio PLCP

**QAM:** Quadrature amplitude modulation, modulación de amplitud en cuadratura

**QPSK:** Quaternary Phase Shift keying, desplazamiento de fase cuaternaria

**RAM:** Random Access Memory, memoria de acceso aleatorio

**RTS:** Request to send, petición de envío

**SAMAN:** Simulation Augmented by Measurement and Analysis for Networks, simulación para redes aumentada por medición y análisis

**SFD:** Start frame delimiter, delimitador de inicio de trama

**SIFS:** Short Inter frame space, espacio corto entre tramas

**TCP:** Transmission Control Protocol, protocolo de control de transmisión

**UDP:** User Datagram Protocol, protocolo de datagrama de usuario

**VBR:** Variable Bit Rate, tasa variable de bit

**VCS:** Virtual Carrier Sense, detección de portadora virtual

**VINT:** Virtual Inter Network Test bed, ambiente de prueba inter-red virtual

**VoIP:** Voice over IP, voz sobre IP

**WEP:** Wired Equivalent Privacy, privacidad equivalente al cable

**WiFi:** Wireless Fidelity, fidelidad inalámbrica

**WLAN:** Wireless LAN, Red inalámbrica.



## INTRODUCCIÓN

Las redes inalámbricas son aquellas que utilizan al espacio como medio de transmisión; se valen de ondas electromagnéticas para establecer comunicación entre los dispositivos y así poder realizar transferencia de archivos, conexión a Internet, compartir recursos de red, etc. tal como se lo hace con las redes cableadas.

La tecnología inalámbrica más utilizada en la actualidad es aquella que se encuentra definida en el estándar 802.11, el cual con el paso de los años, ha tenido varios cambios y mejoras, como por ejemplo 802.11a, 802.11b, 802.11g, que trabajan a diferentes velocidades.

El estándar IEEE 802.11 se desarrolla en las 2 capas inferiores del modelo OSI: la capa física y la capa de enlace de datos, ya que lo único en que difiere con respecto a los estándares para las redes cableadas es en el medio utilizado para la comunicación entre los dispositivos, y en la forma en que estos dispositivos acceden a este medio.

El estándar define 2 componentes: Una estación inalámbrica y un punto de acceso (AP). Una estación inalámbrica es básicamente una computadora que tiene incorporada una tarjeta de red inalámbrica. El AP es el dispositivo que nos permite la comunicación entre las estaciones inalámbricas y la red cableada. El estándar define también 2 modos de operación: Infraestructura y Ad-hoc.

El modo Infraestructura consiste en una o varias estaciones inalámbricas conectadas a un punto de acceso, el cual, proveerá a las estaciones de los servicios que nos ofrecen las redes cableadas como compartir impresoras, acceso a bases de datos, Internet, entre otros. A este conjunto: estaciones y punto de acceso, se lo denomina Conjunto de Servicio Básico (BSS). Un conjunto de 2 o más BSS se denominan Conjunto de servicio Extendido (ESS).

El modo Ad-hoc es un conjunto de 2 o más estaciones inalámbricas conectadas directamente una con otra sin la necesidad de un punto de acceso y de esta manera compartir recursos entre ellas. A este conjunto de estaciones se las denomina Conjunto de servicio Básico independiente (IBSS).

# CAPÍTULO 1

## 1. ANTECEDENTES DEL PROYECTO

La investigación es una actividad que engendra desarrollo y apertura hacia nuevos conocimientos. Estamos conscientes de que ahora más que nunca es lo que hace falta en nuestro medio y es lo que estamos dispuestos a hacer desde nuestro campo de acción: el estudio de las telecomunicaciones.

El desenvolvimiento principal de nuestro trabajo ha sido entorno a la investigación y evolución de una alternativa didáctica referente a las comunicaciones inalámbricas y el estándar 802.11 b y 802.11g, dentro de un concreto pero descriptivo marco teórico, estableciendo las prioridades y requerimientos necesarios para brindar un producto altamente competitivo, pedagógico y provechoso, cuya finalidad primordial es la de facilitar la comprensión del funcionamiento de esta tecnología.



Creemos que nuestro proyecto de graduación llegará a convertirse en un apreciable aporte para profesores y educandos, al igual que un paso hacia adelante en el camino del fomento a la investigación.

### **1.1. Identificación de la problemática**

Como estudiantes de la ESPOL, y más específicamente de la Facultad de Ingeniería en Electricidad y Computación, expresamos nuestra gratitud a esta institución por la formación académica que hemos obtenido durante estos años. Concordamos que es justo retribuir de alguna manera el beneficio recibido de un modo en el cual nuestro esfuerzo pueda verse plasmado para futuras promociones.

En el transcurso de este tiempo y, con las vicisitudes comunes de los estudiantes de ingeniería en telecomunicaciones, hemos reparado en ciertas necesidades existentes en nuestros laboratorios. Siendo parte integrante del grupo de investigación de comunicaciones inalámbricas y como egresados hemos advertido que no tenemos a nuestra disposición una herramienta didáctica para ejemplificar las bases teóricas en cuanto a tecnologías inalámbricas, las cuales están basadas en complejas y abstractas nociones teóricas tales como definición de protocolos, capas, modelos de propagación, etc. Esta

carencia se hace aún más patente si se toma en consideración el acelerado auge y popularidad que han alcanzado estas tecnologías. Podemos fácilmente ver a nuestro alrededor como se ha incrementado a gran escala el uso de redes inalámbricas en ambientes domésticos, comerciales y empresariales. Tanto es así que actualmente se les da aplicaciones que hasta antes sólo era imaginables para redes cableadas, como el tráfico de elevados volúmenes de información, voz y video. Continuamente surgen en el mercado nuevas adaptaciones y mejoras a los estándares. En resumen, las comunicaciones inalámbricas se han convertido en un extenso e interesante campo de estudio que merece nuestra atención y dedicación. Es por eso que hemos decidido que la solución de dicha problemática sea hacia donde apunten nuestros objetivos:

- Brindar una herramienta que nos permita comprender y asimilar los principales aspectos del funcionamiento de las redes inalámbricas 801.11b y 802.11g.
- Adecuar esta herramienta a manera de prácticas de laboratorio, para ser utilizada en centros de estudios superiores.
- Incentivar la difusión de la actividad investigativa como herramienta de aprendizaje.

### 1.1.1. Tecnologías utilizadas

Para alcanzar nuestro objetivo, lo primero que debe ser tomado en cuenta es lo que vamos a necesitar para ponerlo en marcha. Nuestro trabajo se basa en el uso de varios programas especializados por lo que son fundamentales ciertos requerimientos, tales como un computador con componentes actualizados (al menos 512M de RAM y 60GB de disco duro) y sistema operativo Linux, o en su defecto se puede hacer uso de la herramienta "máquina virtual" sobre cuya base se puede instalar dicho sistema operativo. La exigencia de este sistema operativo reside en el simulador en el que vamos a trabajar, NS2 versión 2.31. Dentro de este simulador hacemos uso de la librería 802.11, la cual viene incluida como parte del paquete de instalación del NS2.

### 1.1.2. Limitaciones operacionales

En cuanto a nuestro simulador NS2, encontramos algunas limitaciones, siendo casi imposible encontrar un programa perfecto. Y aunque existen motivos muy valiosos por los cuales

hemos elegido el NS2, debemos recalcar también sus desventajas. Entre ellas podemos citar las siguientes:

- No ofrece una interfaz amigable por excelencia.
- Posee un lenguaje de programación un tanto complicado.
- La documentación no es muy profusa.
- El simulador funciona sobre el Sistema Operativo LINUX, el cual no es la plataforma mas comúnmente utilizada.
- La interfaz gráfica del simulador es ligeramente inestable al momento de ejecución de la misma.

## 1.2. Simuladores disponibles

En el ámbito de la utilización del programa para la elaboración de herramientas didácticas podemos hallar varias alternativas, cada una con sus ventajas y desventajas. Uno de los más solicitados sería OPNET, el cual tiene a su haber una interfaz bastante amigable y una gran adaptación a muchas aplicaciones. Pero hay un obstáculo definitivamente insalvable para su utilización: se trata del costo de la adquisición de su licencia.

Otra notable opción podría ser MATLAB, el cual tiene innumerables utilidades en muchos campos, incluso en comunicaciones, pero no presenta el entorno gráfico de otros simuladores.

Definitivamente el simulador seleccionado por nosotros es el NS2, entre nuestras razones se anotan la apertura del programa; la posibilidad de añadir un ambiente gráfico que si nos establecemos dentro de un marco de referencia en el cual el aspecto pedagógico es importante, resulta un complemento apetecible y además, se halla el hecho de que a pesar de que no está copiosamente documentado, existen foros e investigadores alrededor del mundo de quienes se puede obtener información y ejemplos referentes a las múltiples dificultades que se presentan en el desarrollo de nuestro trabajo.

### **1.3. Justificación para los cambios**

Como ya hemos mencionado anteriormente, dentro del simulador, la librería base que estamos utilizando es la 802.11 pero, como es evidente, es imprescindible efectuar ciertos cambios ya que nuestra meta es representar las tecnologías de 802.11b y 802.11g. Por lo tanto, hemos añadido un conjunto de parámetros propios de cada uno de estos estándares.



También hemos creado archivos tipo awk en donde los algoritmos desarrollados son sobre el lenguaje C++, los cuales realizan un tratamiento de datos con los resultados obtenidos del archivo "trace (.tr)" proveniente de un script (el mismo que esta descrito en el capítulo 3 sección 3.2), para generar una tabla cronológica del transcurso de la simulación con datos de la tasa de rendimiento de la transmisión en base a los anchos de banda propuestos en el script.

# CAPÍTULO 2

## 2. TEORÍA

Las redes inalámbricas son aquellas que utilizan al espacio como medio de transmisión; se valen de ondas electromagnéticas para establecer comunicación entre los dispositivos y así poder realizar transferencia de archivos, conexión a Internet, compartir recursos de red, etc., tal como se lo hace con las redes cableadas.

Entre las ventajas de las redes inalámbricas encontramos: permitir la movilidad de los equipos, poder reubicar las estaciones de trabajo sin la necesidad de crear una red cableada. La instalación es muy sencilla, y nos permite abaratar los costos.

Las redes cableadas tienen velocidades de transferencia superiores a las que nos ofrece la tecnología inalámbrica, por lo que no se espera que se la llegue a sustituir, sino más bien que trabajen en conjunto usando el

sistema cableado como parte principal de la red y el inalámbrico como complemento, en lugares en donde se requiera tener movilidad y rápido acceso a la red.

La tecnología inalámbrica más utilizada en la actualidad es aquella que se encuentra definida en el estándar 802.11, el cual con el paso de los años, ha tenido varios cambios y mejoras, como por ejemplo 802.11a, 802.11b, 802.11g, que trabajan a diferentes velocidades. El conjunto de estos estándares es conocido también como WiFi (Fidelidad Inalámbrica).

## **2.1. Estándar IEEE 802.11**

### **2.1.1. Historia**

802.11 surgió como resultado de una decisión tomada en el año 1985 por la FCC (Comisión Federal de Comunicaciones en Estados Unidos), de abrir varias bandas del espectro inalámbrico y ser utilizadas sin necesidad de tener licencia del gobierno.

Estas bandas eran asignadas a equipos tales como los hornos microonda, los cuales utilizan ondas de radio para calentar

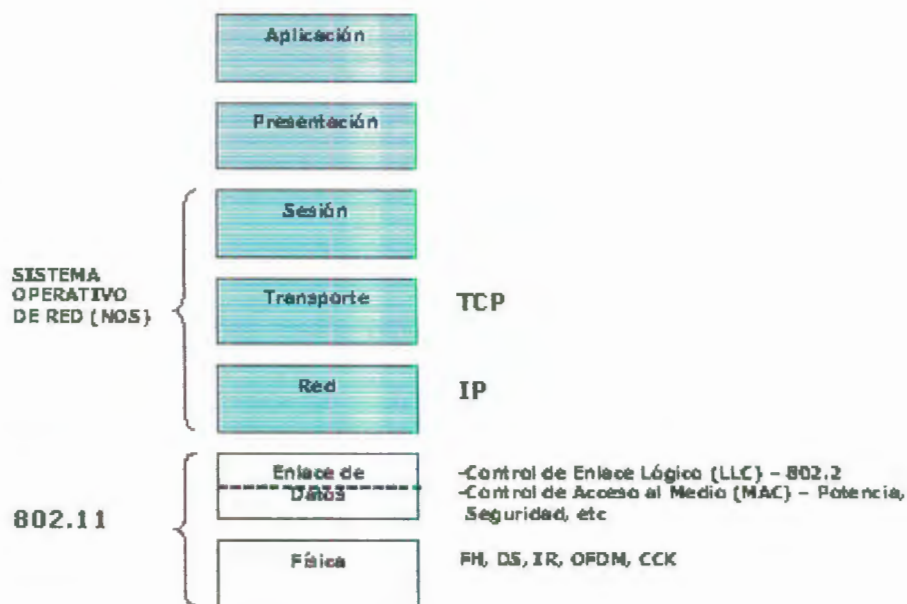


comida. Para operar en estas bandas, los dispositivos requerían el uso de la tecnología denominado “Espectro Ensanchado”. Esta tecnología ensancha la radio señal sobre un amplio rango de frecuencias, haciendo a la señal menos susceptible a la interferencia y difícil de interceptar.

En el año 1990, el IEEE creó un nuevo comité llamado 802.11 para empezar a desarrollar el estándar. En 1991 ya se logró llegar a una velocidad de 1 Mbit/s, lo mínimo para que el comité lo considere una LAN. Finalmente, para el año 1997, el estándar terminó su desarrollo y fue publicado.

### **2.1.2. Arquitectura**

El estándar IEEE 802.11 se desarrolla en las 2 capas inferiores del modelo OSI: la capa física y la capa de enlace de datos, ya que lo único en que difiere con respecto a los estándares para las redes cableadas es el medio utilizado para la comunicación entre los dispositivos, y en la forma en que estos dispositivos acceden a este medio.



*Figura 2.1: 802.11 y Modelo OSI*

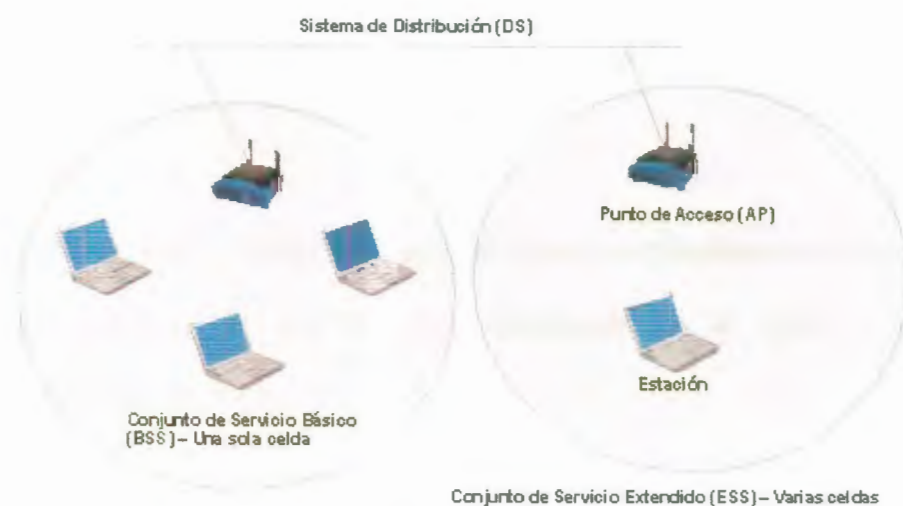
El estándar define 2 componentes: Una estación inalámbrica y un punto de acceso (AP). Una estación inalámbrica es básicamente una computadora que tiene incorporada una tarjeta de red inalámbrica. El AP es el dispositivo que nos permite la comunicación entre las estaciones inalámbricas y la red cableada.

El estándar define también 2 modos de operación: Infraestructura y Ad-hoc.

El modo Infraestructura consiste en una o varias estaciones inalámbricas conectadas a un punto de acceso, el cual, proveerá a las estaciones de los servicios que nos ofrecen las redes cableadas como compartir impresoras, acceso a bases de datos, Internet, entre otros. A este conjunto compuesto de estaciones y punto de acceso se lo denomina Conjunto de Servicio Básico (BSS). Un conjunto de 2 o más BSS se denominan Conjunto de Servicio Extendido (ESS). Los distintos BSS se interconectan por medio de la arquitectura denominada DS (Sistema de Distribución). El punto de acceso (AP) provee el acceso de las estaciones al DS.

El modo Ad-hoc es un conjunto de 2 o más estaciones inalámbricas conectadas directamente una con otra sin la necesidad de un punto de acceso, para que de esta manera se pueda compartir recursos entre ellas. A este conjunto de estaciones se lo denomina Conjunto de Servicio Básico de Independiente (IBSS).

La interconexión entre una red 802.11 y una No 802.11 se realiza mediante un Portal. Un mismo dispositivo puede cumplir las funciones de AP y Portal.



**Figura 2.2: Modo Infraestructura**



**Figura 2.3: Modo Ad-Hoc.**

### 2.1.3. Técnicas de Acceso

Más allá de la funcionalidad del estándar usualmente protagonizado por las capas MAC, la del 802.11 realiza otras funciones que son típicamente relacionadas con los protocolos

de capa superior, tal como fragmentación, retransmisión de paquetes y reconocimientos.

La capa MAC define 2 diferentes tipos de métodos de acceso: La función de coordinación distribuida y la función de coordinación de punto.

### **El método de acceso básico: CSMA/CA**

El mecanismo básico de acceso, llamado función de coordinación distribuida es básicamente un Acceso Múltiple por Detección de Portadora (CSMA), con Evasión de Colisión (CSMA/CA). Los protocolos CSMA son muy conocidos en la industria, de los cuales el más popular es Ethernet.

Un protocolo CSMA funciona de la siguiente manera: Una estación que desea transmitir detecta el medio, si el medio está ocupado (Ej. otra estación está transmitiendo), entonces la estación postergará su transmisión, si el medio se encuentra libre entonces se le permite a la estación transmitir.

Estas clases de protocolos son muy efectivos cuando el medio no está muy cargado ya que permite a las estaciones transmitir con un retardo mínimo, pero siempre hay la posibilidad de que las estaciones transmitan al mismo tiempo (colisión), causado por el hecho de que las estaciones detectaron que el medio estaba libre al mismo tiempo.

Estas situaciones de colisión deben ser identificadas, por lo que la capa MAC puede retransmitir el paquete por sí misma y no por las capas superiores, lo cual causaría un retardo significativo. En el caso de Ethernet, esta colisión es reconocida por estaciones transmisoras las cuales pasan a una fase de retransmisión basada en un algoritmo de “backoff aleatorio exponencial”.

Mientras que estos mecanismos de detección de colisiones son una buena idea en una LAN alámbrica, no pueden ser usados en una inalámbrica, por dos razones principales:

1. Implementar un mecanismo de detección de colisión requeriría la implementación de un radio full-dúplex, capaz



de transmitir y recibir al mismo tiempo. Esto aumentaría demasiado el precio.

2. En un ambiente inalámbrico no podemos asumir que todas las estaciones se escuchan unas con otras (lo cual es la base del esquema de detección de colisión) y no porque la estación que desea transmitir detecta que el medio está libre significa que lo esté en las proximidades del receptor.

Para superar estos problemas, el 802.11 usa un mecanismo de evasión de colisión (CA) junto con un esquema de reconocimiento positivo, como sigue:

Una estación que desea transmitir detecta que el medio está libre, si el medio está ocupado espera. Si el medio está libre por un tiempo especificado (llamado DIFS, Espacio de Entre trama distribuido) entonces la estación puede transmitir, la estación que recibe chequeará el CRC del paquete recibido y enviará un ACK. La recepción de este paquete ACK indicará al transmisor que no ha ocurrido ninguna colisión. Si el emisor no recibe el ACK, entonces retransmitirá el fragmento hasta recibirlo o lo descartará después de un número de retransmisiones dado.

### **Detector de Portadora Virtual (Virtual Carrier Sense)**

Para reducir la probabilidad de que dos estaciones entren en colisión debido a que no se escuchan entre sí, el estándar define el mecanismo de función de coordinación de punto que es el mecanismo de portadora virtual:

Una estación que desea transmitir, primero tendrá que transmitir un paquete de control corto llamado RTS (Petición para enviar), el cual incluye la fuente, destino y la duración de la siguiente transacción (Ej. el paquete y su respectivo ACK), el destino responderá (si el medio está libre) con un paquete CTS (Despejado para enviar), el cual incluirá la misma información de duración.

Todas las estaciones que reciben sea el RTS y/o el CTS, establecerán sus VCS (llamado también NAV), para la duración dada, y usarán esta información junto con el Detector de Portadora Física cuando detecten el medio.

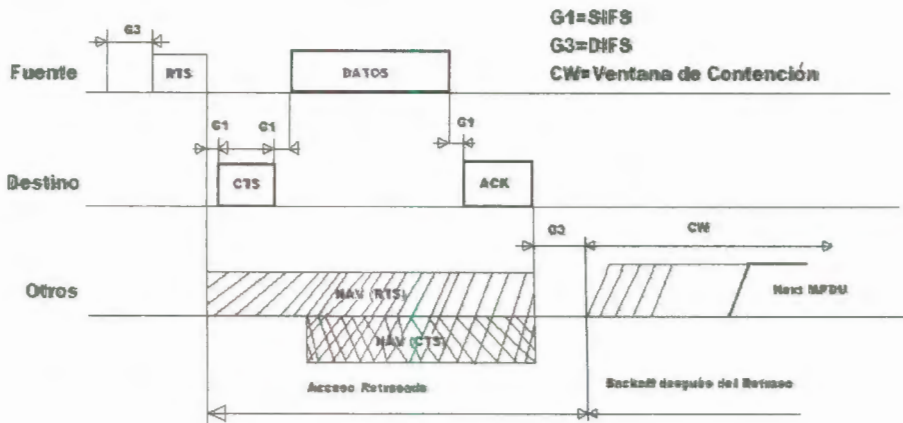
Este mecanismo reduce la probabilidad de una colisión en el área del receptor que está “escondido” desde el transmisor, a la



corta duración de la transmisión RTS, porque la estación escuchará el CTS y “reservará” el medio como ocupado hasta el fin de la transacción. La duración de información en el RTS también protege el área de transmisión de las posibles colisiones durante la transmisión del ACK (debido a las estaciones que están fuera del rango de reconocimiento de la estación).

También debe notarse que debido al hecho de que los RTS y CTS son tramas cortas, se reduce la probabilidad de colisión, ya que estos son reconocidos más rápido que si todo el paquete tuviera que ser transmitido, (esto es verdad si el paquete es significativamente mayor que el RTS, por lo que el estándar permite a los paquetes pequeños ser transmitidos sin el intercambio previo de RTS/CTS, y esto es controlado por cada estación por un parámetro llamado Umbral RTS (RTSThreshold)).

El siguiente diagrama es una transacción entre dos estaciones A y B y la configuración NAV de sus vecinos:



**Figura 2.4:** Configuración NAV entre dos Estaciones

#### 2.1.4. Protocolos

Como se dijo antes, la capa MAC realiza la detección de colisiones al esperar un ACK para cada fragmento (excepto cuando tienen más de un destino, como un multicamino, de los cuales no se espera un ACK).

#### Fragmentación y Reensamblaje

Los protocolos típicos de LAN usan paquetes de cientos de bytes (Ej. el paquete más largo de Ethernet podría ser de hasta 1518 bytes de longitud). Pero en redes inalámbricas hay

algunas razones por las que se prefiere usar paquetes más cortos.

- Porque debido al elevado BER de un radio enlace, la probabilidad de que un paquete se corrompa se incrementa en relación al tamaño de éste.
- En caso de daño del paquete (sea por colisión o por ruido), requerirá menos recursos retransmitir los paquetes más pequeños.
- En un sistema de Salto de Frecuencia, el medio es periódicamente por el salto (en nuestro caso es cada 20 milisegundos), por lo que mientras más pequeño sea el paquete, menor será la probabilidad de que la transmisión sea pospuesta.

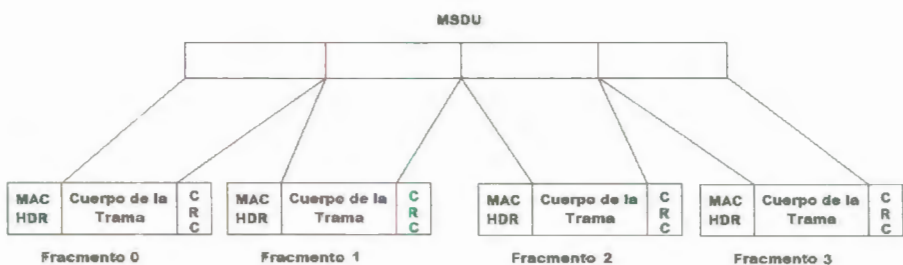
Por otro lado, no tiene sentido introducir un nuevo protocolo de LAN que no pueda tratar con paquetes de 1518 bytes los cuales son usados en Ethernet, por lo que el comité decidió resolver el problema añadiendo un simple mecanismo de fragmentación y reensamblaje en la capa MAC.

El mecanismo es un simple algoritmo Enviar y Esperar (Send-and-Wait), donde no se le permite a la estación transmitir un nuevo fragmento hasta que suceda uno de los siguientes:

1. Que reciba un ACK para dicho fragmento, o
2. Que decida que el fragmento fue retransmitido demasiadas veces y deseche la trama entera.

Debe notarse que el estándar permite a la estación transmitir a diferentes direcciones entre retransmisiones de un fragmento dado, esto es particularmente útil cuando un AP tiene varios paquetes para varios destinos y uno de ellos no responde.

El siguiente diagrama muestra una trama (MSDU) siendo dividido en fragmentos (MPDU).



**Figura 2.5:** Trama MSDU

## Espacios entre tramas

El estándar define 4 tipos de espacio entre tramas, los cuales se usan para diferentes prioridades.

- SIFS - El cual significa Espacio Entre Trama Corto, se usa para separar transmisiones que pertenecen a un dialogo simple que es el espacio entre trama mínimo y siempre hay al menos una estación que quiere transmitir en este tiempo, por tanto tiene prioridad sobre todas las otras estaciones.

Este es un valor fijado por PHY y es calculado de tal manera que la estación transmisora sea capaz de volver al modo de recepción y que sea capaz de decodificar el paquete entrante. En la PHY de 802.11 este valor es fijado a 28 microsegundos.

- PIFS – Espacio entre trama de función de coordinación de punto, es usado por el AP para ganar acceso al medio antes que cualquier otra estación.

Este valor es igual al SIFS más un slot de tiempo, 78 microsegundos.

- DIFS – Espacio entre trama de función de coordinación distribuida, es el espacio entre tramas usado para una

estación que desea empezar una nueva transmisión, el cual es calculado como el PIFS más un slot de tiempo, 128 microsegundos.

- EIFS – Espacio entre trama extendido, el cual es un IFS más largo usado por una estación que ha recibido un paquete que no pudo entender, esto es necesario para prevenir a la estación (quien no pudo entender la información de duración para el Perceptor de Portadora Virtual), de colisionar con un paquete futuro que pertenezca al dialogo actual.

### **Algoritmo Exponencial Backoff.**

Backoff es un método bien conocido para resolver la contención entre diferentes estaciones que desean acceder al medio. El método requiere que cada estación escoja un número aleatorio (n) entre cero y un número dado, y esperar a este número de slots antes de acceder al medio, siempre comprobando si una estación diferente haya accedido al medio antes.

El slot de tiempo es definido de tal manera que una estación siempre sea capaz de determinar si otra estación ha accedido



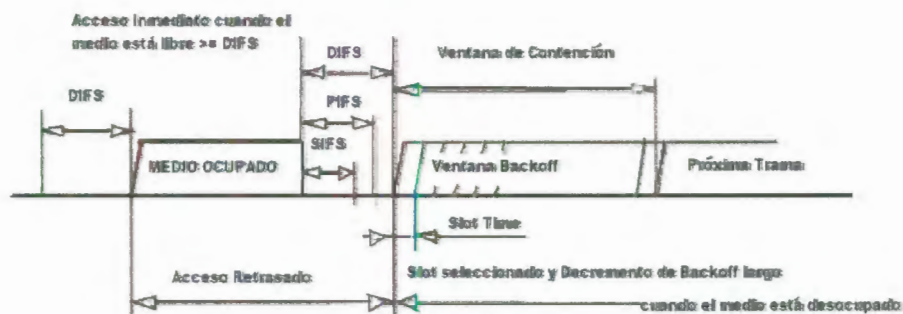
al medio al principio del slot previo. Esto reduce la probabilidad de colisión a la mitad.

Backoff Exponencial significa que cada vez que la estación escoja un slot y colisione, esta incrementará el número máximo para la selección aleatoria exponencialmente.

El estándar 802.11 define que el algoritmo backoff exponencial debe ser ejecutado en los siguientes casos:

- Cuando la estación detecta el medio antes de la primera transmisión de un paquete, y el medio está ocupado.
- Después de cada retransmisión.
- Después de una transmisión exitosa.

El único caso cuando este mecanismo no es usado es cuando la estación decide transmitir un nuevo paquete y el medio ha estado libre por más de un DIFS.



**Figura 2.6:** Algoritmo Backoff Exponencial

### Cómo se une una estación a una celda ya existente (BSS)?

Cuando una estación quiere acceder a una BSS existente (sea después de haberse encendido, de haber salido del modo de dormido, o justo después de haber entrado al área de la BSS), la estación necesita obtener información de sincronización del AP (o de otra estación cuando funciona en modo ad-hoc).

La estación puede obtener esta información por uno de dos medios:

1. Escaneo pasivo: En este caso la estación sólo espera recibir una trama beacon del AP, (la trama beacon es una trama periódica enviado por el AP con la información de sincronización), o

2. Escaneo activo: En este caso la estación trata de encontrar un AP transmitiendo una trama de requisito de prueba, y esperando una respuesta de prueba del AP.

Los dos métodos son válidos, y cualquiera puede ser escogido de acuerdo con la preferencia entre consumo de potencia y rendimiento.

### **El proceso de autenticación**

Una vez que la estación ha encontrado un AP y ha decidido unirse a la BSS, atravesará el proceso de autenticación, el cual es el intercambio de información entre el AP y la estación, donde cada lado prueba el conocimiento de una clave dada.

### **El proceso de asociación**

Cuando la estación es autenticada, entonces empezará el proceso de asociación, el cual es el intercambio de información acerca de las estaciones y las capacidades de la BSS, y permite el DSS (el conjunto de APs que conocen acerca de la posición actual de la estación). Sólo después de que el proceso

de asociación está completo, una estación es capaz de transmitir y recibir tramas de datos.

## Roaming

Roaming es el proceso de moverse de una celda (o BSS) a otra sin perder conectividad. Esta función es similar a la de los teléfonos celulares, con la diferencia que en un sistema LAN el cual es basado en paquetes, la transición de celda a celda puede ser llevada a cabo entre transmisiones de paquetes, al contrario de la telefonía donde la transición puede ocurrir durante una conversación telefónica, esto hace al roaming de LAN un poco más fácil, pero en un sistema de voz, una desconexión temporal puede no afectar la conversación, mientras que en un ambiente basado en paquetes esto reduce significativamente el desempeño debido a que la retransmisión sería llevada a cabo por las capas superiores.

El estándar 802.11 no define como debería ser realizado el roaming, pero define las herramientas básicas para aquello, esto incluye el escaneo activo y pasivo, y un proceso de

reasociación, donde una estación que está pasando de un AP a otro llegará a estar asociado con el segundo.

### **Manteniendo la sincronización**

Las estaciones necesitan mantener la sincronización, esto se necesita para mantener el salto sincronizado y otras funciones como Resguardo de Potencia (power saving). En una infraestructura de BSS esto es realizado por todas las estaciones que actualizan sus relojes de acuerdo al reloj del AP, usando el siguiente mecanismo:

El AP transmite tramas periódicas llamadas Beacon, estas tramas contienen el valor del reloj del AP en el momento de la transmisión (nótese que éste es el momento cuando la transmisión realmente ocurre, y no cuando está es puesta en la cola, ya que el beacon es transmitido usando las reglas de CSMA, la transmisión puede ser retardada significativamente).

Las estaciones que reciben comprueban el valor de su reloj el momento mismo en que se recibe, y lo corrigen para mantener el sincronismo con el reloj del AP, esto previene que el reloj se

pierda lo cual podría causar una pérdida de la sincronización después de un par de horas de operación.

## **Seguridad**

La seguridad es uno de los temas que primero preocupa a las personas que emplean una LAN Inalámbrica. El comité del 802.11 ha dirigido el problema proveyendo lo que se conoce como WEP (Privacidad Equivalente al Cable).

Las principales preocupaciones para los usuarios respecto a los intrusos es que no sean capaces de:

- Acceder a los recursos de red al usar equipos similares de LAN inalámbricas, y
- Ser capaces de interceptar el tráfico de la WLAN (husmeo).

## **Previniendo el acceso a los recursos de red**

Esto se logra por el uso del mecanismo de autenticación donde una estación necesita probar que tiene conocimiento de la actual clave, esto es muy similar a la privacidad de LAN



alámbrica, en el sentido de que un intruso necesita ingresar los parámetros (usando una clave física) para conectar su estación de trabajo a la LAN alámbrica.

### **Husmeo (Eavesdropping)**

El husmeo es prevenido por el uso del algoritmo WEP el cual es un generador de número pseudo aleatorio por una clave secreta compartida. La salida del PRNG es una secuencia de bits pseudo aleatoria igual en longitud al paquete más largo posible el cual es combinado con el paquete entrante o saliente que produce el paquete transmitido en el aire.

El algoritmo WEP es un simple algoritmo basado en el algoritmo RC4 de RSA que tiene las siguientes propiedades:

- Razonablemente fuerte.- Un ataque de fuerza bruta a este algoritmo es difícil debido al hecho de que todas las tramas son enviadas con un vector de inicialización que reinicia el PRNG para cada trama.

- Sincronización propia.- El algoritmo se sincroniza para cada mensaje, esto se necesita para trabajar en un ambiente sin conexión, donde los paquetes pueden perderse (como en cualquier otra LAN).

### **Resguardo de potencia (power saving)**

Las LAN inalámbricas están típicamente relacionadas con aplicaciones móviles y en este tipo de aplicaciones las baterías de poder son un recurso escaso, esta es la razón de por qué el estándar 802.11 dirige directamente el problema del resguardo de potencia y define un mecanismo entero para permitir a las estaciones ir al “modo de sueño” por largos períodos de tiempo sin perder información.

La idea principal detrás del mecanismo de resguardo de potencia es que el AP mantiene un registro actualizado de las estaciones que al momento están trabajando en el modo resguardo de potencia y retiene los paquetes direccionados a estas estaciones hasta que requieran obtener los paquetes mandando una solicitud, o hasta que cambien su modo de operación.

El AP también transmite periódicamente (como parte de sus tramas beacon) información acerca de la cual las estaciones con resguardo de potencia tengan tramas retenidas en el AP, de manera que esas estaciones deban despertarse para recibir una de estas tramas beacon, y si hay una indicación de que hay una trama almacenada en el AP esperando por ser entregada, entonces la estación deberá estar despierta y enviar un mensaje de solicitud al AP para obtener dichas tramas.

Los multicast y broadcast son almacenados por el AP, y transmitidos en un tiempo preconocido (cada DTIM Módulo de Interfase de Diálogo Telefónico), donde todas las estaciones con resguardo de potencia que desean recibir esta clase de tramas deberán estar despiertas.

#### **2.1.5. Formato de la Trama**

Existen 3 tipos principales de tramas:

- Tramas de datos: Son usadas para transmisión de datos.
- Tramas de control: Se usan para controlar el acceso al medio. (Ej. RTS, CTS y ACK)

- **Tramas de administración:** Se transmiten de la misma manera que las tramas de datos para intercambiar información de administración, pero no son reenviadas a las capas superiores.

Cada uno de estos tipos de trama se subdivide en diferentes subtipos, de acuerdo a su función específica.

Todas las tramas de 802.11 están formadas de los siguientes componentes

<b>Preámbulo</b>	<b>Cabecera PLCP</b>	<b>Datos MAC</b>	<b>CRC</b>
------------------	----------------------	------------------	------------

*Figura 2.7: Formato de la Trama 802.11*

### **Preámbulo**

Es dependiente de la capa física e incluye:

- **Sincronismo(Sinc).**- Una secuencia de 80 bits de ceros y unos alternados, la cual es usada por la circuitería de la capa física para seleccionar la antena apropiada (si la

diversidad es apropiada), y para alcanzar la corrección del offset de la frecuencia de estado estable y sincronización con el packet timing recibido.

- Delimitador de Inicio de Trama (SFD).- consiste en el patrón 0000 1100 1011 1101 de 16 dígitos binarios el cual es usado para definir la trama periódica.

### **Cabecera PLCP (PLCP Header)**

La cabecera PLCP es siempre transmitida a 1 Mbit/s y contiene información lógica que será usada por la capa física para decodificar la trama, y consiste de:

- Longitud de palabra PLCP\_PDU.- Representa el número de bytes contenidos en el paquete, esto es muy útil para que la capa física detecte correctamente el final del paquete.
- Campo de señalización PLCP.- Contiene solo información de la tasa, codificada en incrementos de 0.5 Mbps desde 1 Mbit/s hasta 4.5 Mbit/s.
- Campo de chequeo de error de cabecera.- es un campo de Chequeo Redundancia Cíclica de detección de errores de 16 bits.

## Datos MAC

Este campo contiene información necesaria para la capa MAC, como la dirección de la fuente, dirección del destino, dirección de la estación inalámbrica receptora, dirección de la estación inalámbrica transmisora, datos, entre otros.

## Chequeo de Redundancia Cíclica (CRC)

El CRC es un campo de 32 bits que contiene un chequeo de redundancia cíclica de 32 bits (CRC).

## 2.2. Variaciones de los estándares IEEE 802.11b y del IEEE 802.11g

El estándar IEEE 802.11 ha sufrido cambios y mejoras dando como resultado extensiones de este estándar, en donde la principal diferencia es la velocidad de transferencia de datos:

802.11: estándar original publicado en 1997 con velocidades de 1 y 2 Mbit/s. en la banda ISM de 2.4 Ghz.

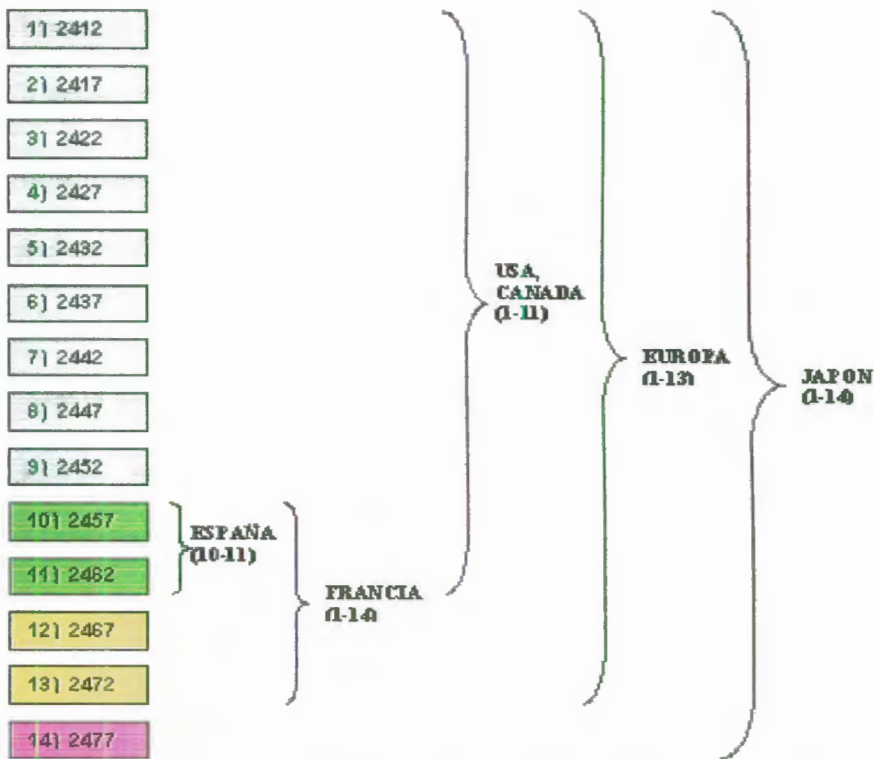


802.11b: publicado en 1999 con velocidades de hasta 11 Mbit/s. trabaja en la banda ISM de 2.4 Ghz.

802.11a: publicado en 1999 con velocidades de hasta 54 Mbit/s utilizando Multiplexación por División de Frecuencia Ortogonal (OFDM). Trabaja en la banda ISM de 5 Ghz, no puede interoperar con el estándar 802.11b.

802.11g: publicado en 2003 con velocidades de hasta 54 Mbit/s utilizando OFDM. Trabaja en la banda ISM de 2,4 Ghz, puede interoperar con el estándar 802.11b.

Existen 3 bandas no licenciadas, en 900 MHz, 2,4 GHz y 5 GHz, que son para el uso industrial, científico y médico (ISM). IEEE 802.11 opera en la banda de 2,4 GHZ al igual que sus posteriores extensiones IEEE 802.11b y IEEE 802,11g.



**Figura 2.8:** Canales de la Banda ISM de 2.4GHz.

En esta figura se muestran los 14 canales en los que se divide la banda ISM de 2,4 GHz. y los canales disponibles que existen en distintos países dependiendo de su regulación del espectro radioeléctrico.

El estándar 802.11 original define tasas de transferencia de 1 Mbps y 2 Mbps, utilizando Espectro Ensanchado por Salto de Frecuencia (FHSS) o Espectro Ensanchado por Secuencia Directa (DSSS), que son mecanismos de señalización completamente diferentes y no pueden interoperar uno con otro.

Usando FHSS la banda de 2,4 GHz es dividida en 75 subcanales de 1 MHz. Tanto el transmisor como el receptor tienen un patrón de salto determinado y el dato es enviado sobre una secuencia de subcanales.

Cada comunicación que existe sobre la red 802.11 se realiza con un patrón de salto diferente, los patrones son diseñados para minimizar la probabilidad de que dos transmisores utilicen el mismo canal simultáneamente.

FHSS permite un diseño de radio simple, pero está limitado a velocidades que no excedan los 2 Mbps. Esta limitación es debido a la regulación de la FCC de restringir el ancho del subcanal a 1 MHz. Esta regulación fuerza a los sistemas FHSS a utilizar toda la banda de 2,4 GHz, realizando muchos saltos, lo que hace que haya un aumento considerable en el overhead.

La técnica de Señalización DSSS divide la banda de 2.4 GHz en 14 canales de 22 MHz, en donde los canales adyacentes se traslapan parcialmente, con 3 de los 14 que no se traslapan. El dato es enviado a través de uno de los canales de 22 Mhz sin que haya salto hacia otro canal. Para compensar el ruido en un canal dado, la técnica del

"código de expansión (chipping)" es utilizada. Cada bit de dato es convertido en una serie de patrones de bits llamados "chips". La redundancia que da cada chip, sumado al ensanchamiento de la señal a través del canal de 22 MHz, nos provee una forma de chequeo y corrección de errores; incluso si parte de la señal está dañada, en muchos casos se la puede recuperar, minimizando la necesidad de retransmisiones.

La mejora que existe en el estándar 802.11b es el soporte de dos nuevas velocidades, 5.5 Mbps y 11 Mbps, utilizando DSSS como técnica de señalización, debido a que FHSS no soporta velocidades superiores a los 2 Mbps. Esto hace que 802.11b pueda interoperar con sistemas a 1 y 2 Mbps utilizando DSSS, pero no podrá hacerlo con sistemas que utilizan FHSS. El estándar 802.11 original especifica 11 bits de código de expansión llamado "Secuencia de Barker", para codificar todo el dato enviado sobre el aire. Cada secuencia representa un solo bit de dato (1 o 0), y es convertido en una forma de onda llamada "símbolo", que puede ser enviado sobre el aire. Estos símbolos son transmitidos a 1 Mbps usando una técnica llamada Codificación por Desplazamiento de Fase Binaria (BPSK). En el caso de 2 Mbps, se utiliza una técnica sofisticada llamada Codificación por Desplazamiento de Cuadratura de Fase

(QPSK); esta duplica la tasa de transferencia en BPSK, mejorando la eficiencia del ancho de banda.

Para incrementar la tasa de transferencia en 802.11b, son empleadas técnicas de codificación avanzadas. En vez de utilizar secuencias de barker de 11 bits, 802.11b especifica Codificación de Código Complementario (CCK), el cual consiste en un conjunto de 64 palabras de 8 bits. Como conjunto, estas palabras código tienen propiedades matemáticas únicas que permiten que sean bien diferenciadas una de la otra por un receptor aún con la presencia de ruido e interferencia multicamino elevadas. La tasa de 5.5 Mbps utiliza CCK para codificar 4 bits por portadora, mientras que la tasa de 11 Mbps codifica 8 bits por portadora. Ambas velocidades utilizan QPSK como técnica de modulación y señal a 1.375 MSps.

Las velocidades de transmisión en la banda de 2.4GHz propuestas por el estándar 802.11 y 802.11b no son suficientes para llegar a alcanzar velocidades parecidas a las de las actuales redes LAN (100Mbps). El problema de alcanzar velocidades mayores es que las técnicas de modulación se vuelven más complejas. Esto influye de forma negativa sobre la relación señal ruido y por lo tanto el alcance de los radio enlaces disminuye.



Para alcanzar velocidades parecidas a las redes LAN se crea el estándar 802.11g, el cual alcanza velocidades de transmisión de hasta 54 Mbps, para lo cual usa como técnica la Modulación por División de Frecuencias Ortogonales (OFDM).

OFDM funciona dividiendo una portadora de datos de alta velocidad en pequeñas subportadoras de baja velocidad que son transmitidas simultáneamente es decir, en paralelo. Cada portadora de alta velocidad es de 20 MHz de ancho y se divide en 52 subcanales, cada uno de 300 Khz. de ancho aproximadamente y utiliza 48 de estos subcanales para datos, mientras que los 4 restantes son usados para corrección de errores.

BPSK es usado para codificar 125 Kbps de datos por subcanal, resultando en una tasa de datos de 6 Mbps. Usando QPSK se duplica la cantidad de datos codificados en 250 Kbps, dando 12 Mbps de tasa de datos usando la modulación 16-QAM se codifica a 4 bits por hertz, llegando así a una tasa de datos de 24 Mbps.

Para llegar a tasas de transferencia de 54 Mbps se utiliza la modulación 64-QAM, dando así de 8 o 10 bits por ciclo, para un total



de 1.125 Mbps por subcanal. Con 48 subcanales utilizados para datos, esto da como resultando una tasa de datos de 54 Mbps.

El estándar 802.11g especifica una extensión de velocidad para la capa física la cual se denomina Tasa Extendida PHY (ERP), y soporta velocidades de 1, 2, 5.5 y 11 Mbit/s utilizando modulación CCK/BARKER para que exista compatibilidad con el estándar 802.11b. OFDM provee tasas de datos de 6, 9, 12, 18, 24, 36, 48 y 54 Mbps. La codificación opcional es la Codificación Convolutiva Binaria en Paquetes (PBCC) que provee tasas de datos de 22 y 33 Mbps.

La capa física de 802.11b está dividida en dos partes: Protocolo de Convergencia de Capa Física (PLCP) y Dependencia del Medio Físico (PMD). El PMD se encarga de la codificación inalámbrica. El PLCP presenta una interfase común con la subcapa MAC para escribir y provee detección de portadora y valoración de canal vacío (clear channel Assessment CCA).

El PLCP tiene dos estructuras, un preámbulo largo y corto, todos los sistemas 802.11b tienen que soportar el preámbulo largo. La opción de preámbulo corto está provista en el estándar para mejorar la

eficiencia en la tasa de rendimiento de una red cuando se transmiten datos especiales como voz, VoIP (voz sobre IP) y video en tiempo real.

El Formato de la trama del PLCP consiste de lo siguiente:

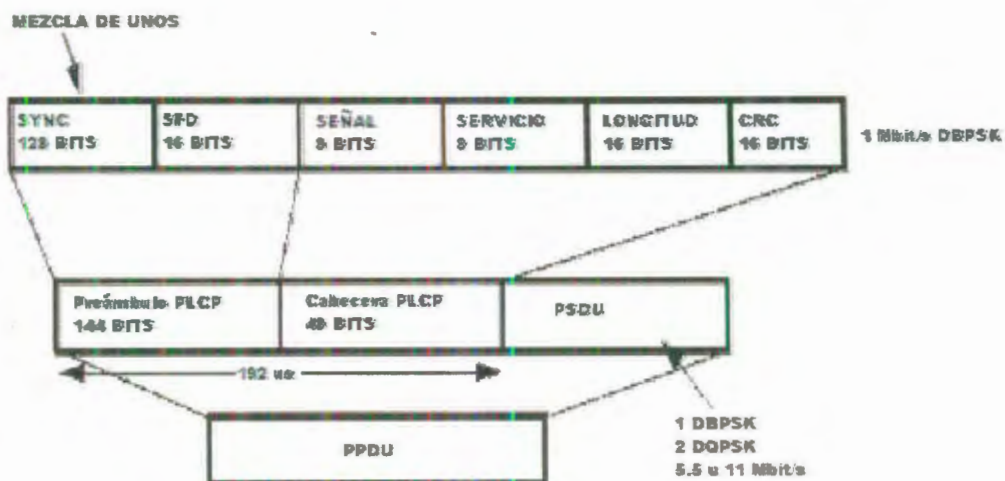
### 1. Preámbulo PLCP

- campo de sincronización el cual consiste de 128 bits para preámbulo largo y 56 bits para preámbulo corto.
- 16 bits para el campo delimitador de inicio de trama (SFD) el cual es usado para marcar el inicio de cualquier trama.

### 2. Cabecera PLCP

- 8 bits para el campo de señal o tasa de datos, que indica la velocidad del dato que será transmitido.
- 8 bits para el campo de servicio, que se reserva para uso futuro.
- 16 bits para el campo de longitud, que indica la longitud de todo el paquete.

- 16 bits para el campo CRC, que se utiliza para la detección de errores.

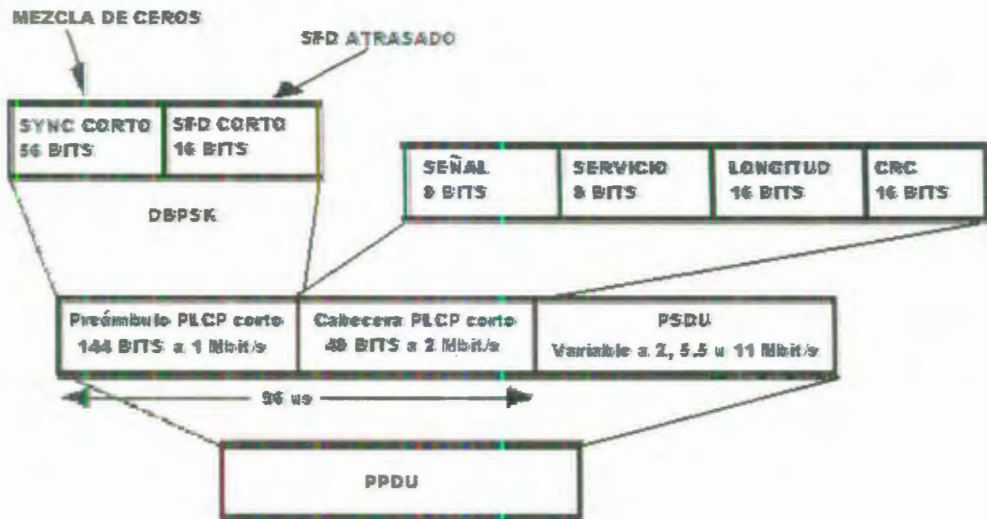


**Figura 2.9:** Formato del PPDU con PLCP largo

En el estándar 802.11b, el preámbulo PLCP largo y la cabecera PLCP largo no difieren del estándar 802.11 original para 1 Mbit/s y 2 Mbit/s, las únicas excepciones son:

- La codificación de la velocidad es en el campo señal.
- EL uso de un bit en el campo de servicio, para resolver una incertidumbre del PSDU en octetos, cuando la longitud es expresada en microsegundos.
- El uso de un bit en el campo de servicio, para indicar si el modo PBCC opcional está en uso.

- El uso de un bit en el campo de servicio, para indicar que la frecuencia de tránsito y el reloj de bit están bloqueados.



**Figura 2.10:** Formato del PPPDU con PLCP corto

En el PLCP largo, el preámbulo y la cabecera se transmiten a 1 Mbit/s utilizando modulación DBPSK. En el PLCP corto, el preámbulo se transmite a 1 Mbit/s con BPSK y la cabecera se transmite a 2 Mbit/s utilizando QPSK.

Un transmisor que usa el PLCP corto solo podrá interoperar con un receptor que tenga la capacidad de recibir PLCP corto. Para interoperar con un receptor que no tenga esta capacidad, entonces se deberá utilizar el PLCP largo.

Como sabemos, todo paquete 802.11 esta formado por un preámbulo, una cabecera y un campo de datos. El estándar 802.11g soporta preámbulo largo y corto.

El estándar 802.11g utiliza las siguientes técnicas de modulación entre las cuales dos son obligatorias y dos son opcionales:

- ERP-DSSS/CCK, es una técnica utilizada para asegurar la compatibilidad con los equipos WiFi existentes en el mercado.
- ERP-OFDM, como ya es sabido el estándar llega hasta velocidades de 54 Mbits/s, para ello esta técnica es requerida puesto que soporta tasa de datos muy alta.
- ERP-PBCC (OPCIONAL), esta técnica es un esquema de modulación de portadora simple que codifica el campo de datos usando PBCC de 256 estados. Esta provee tasa de datos de 22 y 33 Mbits/s.
- DSSS/OFDM (OPCIONAL), esta es una modulación híbrida combinando un preámbulo y una cabecera DSSS con transmisiones de campo de datos OFDM y provee tasa de datos en el campo de datos de 6, 9, 12, 18, 24, 36, 48 y 54Mbits/s. Si esta técnica es usada, las tasas soportadas son las mismas que ERP-OFDM.



La principal diferencia entre las **varias** técnicas de modulación tanto las **obligatorias** como las **opcionales** es en los elementos que conforman la estructura de los paquetes.

A continuación se dará una **descripción** más detallada con respecto a las técnicas de modulación **obligatorias**, que son las más utilizadas.

### **ERP-DSSS/CCK**

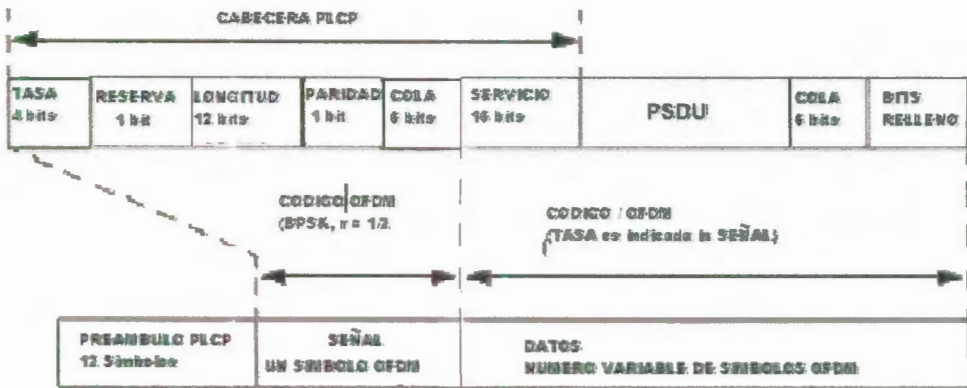
La capa física usa la misma **trama** que 802.11b Velocidad Alta DSSS con algunas excepciones.

- 1) El preámbulo PLCP corto es **obligatorio**.
- 2) El CCA tiene un mecanismo que detecta todos los símbolos **sincrónicos obligatorios**.
- 3) El aseguramiento de la transmisión central de potencia y el reloj de frecuencia de símbolos **referenciados** al mismo oscilador son **obligatorios**.

### **ERP/OFDM**

La trama OFDM es de la siguiente forma:





**Figura 2.11: Formato de la Trama OFDM**

El formato PDU incluye el preámbulo, una cabecera PLCP OFDM, PSDU, cola, y los bits de relleno. La cabecera PLCP contiene los siguientes campos:

Longitud, tasa, un bit reservado, un bit de paridad par y el campo de servicio. En términos de modulación la longitud, la tasa, el bit reservado y el bit de paridad constituyen un simple símbolo OFDM separado, denotado como señal, el cual es transmitido con la más robusta combinación de modulación BPSK y una tasa de codificación ( $R = 1/2$ ).

El campo de servicio de la cabecera PLCP y el PSDU son denotados como datos y son transmitidos con la tasa de datos descritos en el campo tasa y constituye múltiples símbolos OFDM. Los bits cola en el

símbolo señal habilitan la decodificación de los campos tasa y longitud inmediatamente después de la recepción de los bits cola. La tasa y longitud son requeridas para decodificar la parte de datos del paquete.

Además, el mecanismo CCA puede ser argumentado por la predicción de la duración de los paquetes desde los contenidos de los campos tasa y longitud, aún cuando la STA no soporta la tasa de datos. El time slot es de 20  $\mu\text{s}$ , excepto que una opción es 9  $\mu\text{s}$  cuando la BSS consiste sólo de STA ERP y por último el tiempo SIFS es de 10  $\mu\text{s}$ .

# CAPÍTULO 3

## 3. DESCRIPCION DEL PROYECTO

### 3.1. Solución planteada

Una vez identificado el problema, el que es el no disponer de una herramienta didáctica para ejemplificar las bases teóricas en cuanto a las tecnologías inalámbricas, se adoptó como solución, que esta herramienta didáctica debe ser a manera de simulaciones en computadoras, utilizando un programa especializado en tecnologías de redes inalámbricas.

La simulación por computadora, es un programa que nos permite reproducir las características principales de un proceso o fenómeno mediante ecuaciones matemáticas, utilizando las ventajas de procesamiento de un computador.

Las simulaciones de procesos y fenómenos, son modelos o esquemas tomados de la realidad, que nos permiten entender de mejor manera las bases teóricas estudiadas en clase. Pero nunca podrán reemplazar a la implementación real, ya que los resultados obtenidos en las simulaciones son aproximaciones de la realidad, y hay que realizar bastantes pruebas y cambios, para que estos resultados sean lo más cercanos a la realidad.

Sin embargo, el hecho de implementar estos fenómenos o procesos de manera real para la investigación y desarrollo de tecnologías en establecimientos educativos no es óptimo, ya que requiere de elevados presupuestos para la adquisición de equipos y materiales, que al poco tiempo con el acelerado avance tecnológico, terminan siendo discontinuados y muchas veces obsoletos.

Esta es la ventaja principal del uso del programa de simulación como complemento teórico y en nuestro uso particular, que es simular redes inalámbricas IEEE 802.11b y 802.11g, las cuales son tecnologías que se encuentran en pleno auge en lo que a redes de hogar y oficina se refiere.

### **3.2. Descripción del Programa NS2, la Librería 802.11 y la Herramienta para el Tratamiento de Datos**

NS2 es un simulador de eventos discretos, el cual es ampliamente utilizado tanto en investigación como para propósitos docentes.

NS comenzó en el año 1989, como una variación del "Real Network Simulator" desarrollado por S. Keshav, y ha evolucionado con el paso de los años.

En 1995, el desarrollo del NS se ha llevado a cabo por la Agencia de Investigación de Proyectos Avanzados de Defensa (DARPA), a través del proyecto VINT, que es un proyecto de investigación financiado cuyo objetivo es el desarrollo de un simulador de redes y es un trabajo en conjunto realizado por UC Berkley, USC/ISI, LBL, y Xerox PARC.

En la actualidad NS está siendo desarrollado, a través de DARPA con el proyecto SAMAN y a través de la Fundación Nacional de Ciencias (NSF) con CONSER, ambos en colaboración con otros investigadores incluyendo a ICIR.

Un simulador de eventos discretos es aquel que genera números aleatorios de una distribución discreta en especial y con estos se puede simular instancias de tiempo que pueden ser de transmisión o recepción de paquetes de datos, entre otros.

NS2 está basado en dos lenguajes: C++ en el cual está escrito el simulador, y lenguaje orientado a objetos (Otcl) como interfaz de configuración y comandos.

El lenguaje Otcl es el que se utiliza para configurar los escenarios de las simulaciones en la cuales constan la topología de la red y el tráfico de paquetes que posee la misma, y de esta manera poder observar y analizar el comportamiento de una red con tales características.

NS2 ya contiene implementados protocolos tales como TCP y UDP, generadores de tráfico como son FTP, CBR, VBR, Telnet, Web. Maneja también diversos mecanismos de colas que se generan en los ruteadores tales como DropTail, RED, CQD, algoritmo de Dijkstra, etc.



NS2 dispone de una interfaz gráfica para visualizar las simulaciones llamada Animador de Redes (Network Animador NAM), este también dispone de un editor gráfico, que nos va a permitir no tener que usar el código tcl para crear las animaciones. Se puede crear la topología de red y simular varios protocolos y fuentes de tráfico mediante el uso del ratón. Para una mayor comprensión sobre lo expuesto anteriormente consultar el apéndice A.

En NS2, dos protocolos de capa MAC son implementados para escenarios inalámbricos: 802.11 y TDMA. En nuestro caso utilizamos el protocolo MAC802\_11 que ha sido implementado por CMU y utiliza la Función de Coordinación Distribuida (DCF). Este protocolo usa patrones RTS/CTS/datos/ACK para representar todos los paquetes unicast y solamente transmite datos para todos los paquetes broadcast. La implementación usada para ambos paquetes es Detección de Portadora Virtual y Detección de Portadora Física (sección 2.1.3). La clase MAC802\_11 es implementada en `~ns/mac-802_11. {cc, h}`.

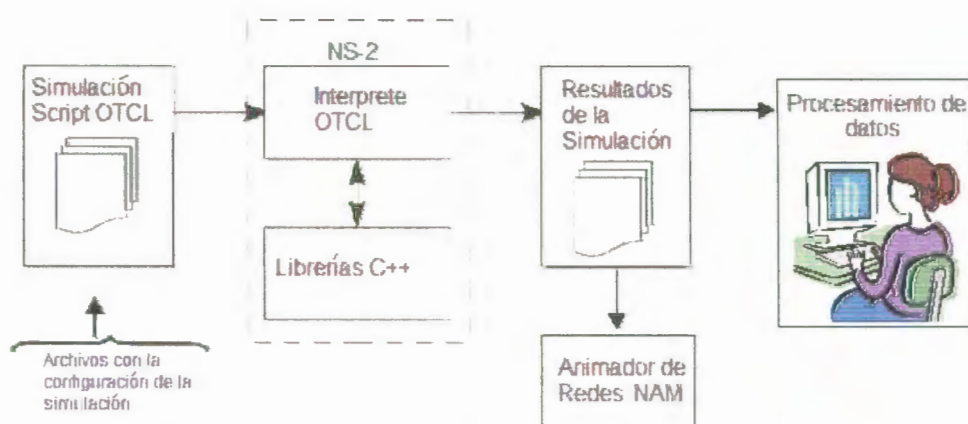
La herramienta creada para el tratamiento de datos es "awk". Este nombre se deriva de los apellidos de los autores: Alfred Aho, Meter Weinberger y Brian Kernighan. Ésta es una utilidad que nos traen los

sistemas operativos basados en UNIX, es básicamente un lenguaje de programación con una sintaxis parecida al C, que está especialmente diseñado para el procesamiento de archivos estructurados y patrones de texto. Este lenguaje nos permite hacer simples operaciones en archivos de datos, tales como el promedio de valores de una columna dada, la suma o multiplicación de término por término, o entre algunas columnas, todas las áreas de datos reformados, etc. Debido a todas estas funciones, awk resulta bastante apropiado para trabajar con archivos que tienen información estructurada en campos.

Viendo todas las virtudes que posee este lenguaje de programación, hemos decidido crearlo como herramienta para procesar los datos de salida del simulador, el cual es el archivo de trama de extensión ".tr", de esta manera podremos obtener información muy importante para el análisis, como es el promedio de la tasa de rendimiento de la simulación, número de paquetes recibidos, tiempo de inicio, tiempo de fin, entre otros. Y con estos valores demostrar que los parámetros que hemos definido para las simulaciones IEEE 802.11b y 802.11g funcionan.

### 3.3. Funcionamiento General y Diagrama de Bloques

La siguiente figura representa el diagrama de bloques del funcionamiento general del proyecto, el cual nos permitirá comprender de una manera sencilla los procesos que son llevados a cabo por el simulador.



**Figura 3.1:** Diagrama de Bloques del Proyecto.

Como observamos en la figura 3.1 comenzamos con el desarrollo del script, que es un archivo escrito en lenguaje Otcl, que tiene diversos componentes internos y es lo que el usuario codifica para simular. En estos componentes se configura la topología de la red, planifica los eventos, carga las funciones necesarias para la simulación, planifica cuando iniciar o terminar el tráfico de un determinado paquete, especifica los archivos de trazas y los tiempos de comienzo de los eventos iniciales de la simulación, estos crearán nuevos eventos a lo

largo de la simulación y así sucesivamente. El simulador siempre ejecuta los eventos en el orden especificado en la lista de eventos, y en esta lista los eventos siempre estarán ordenados por tiempo.

El intérprete Otcl es el que orienta el script configurado hacia las herramientas o librerías implementadas en el NS2 o también llamados archivos objeto de extensión “.oo”. Este intérprete tiene una relación biunívoca con el lenguaje C++ el mismo que permite una gran personalización de la simulación de las rutinas a nivel de paquetes. El intérprete Otcl es el que crea la lista de eventos y genera los archivos de salida.

Dentro de los resultados de la simulación tenemos dos tipos de archivos de salida, el archivo tipo traza “.tr” y el archivo NAM. La manera de obtener los archivos de traza es a través del uso de objetos traza, estos recolectan datos para cada llegada de paquete, salida de paquete o descarte. El formato del archivo de traza en NS2 tiene un formato fijo, en donde cada columna corresponden a diversos parámetros como tipos de evento, instante del tiempo en que se ejecuta el evento, tamaño del paquete, descripción del paquete, entre otros.



El animador de redes más conocido como NAM, es una aplicación externa del NS2 que permite visualizar de forma gráfica lo que sucede durante la simulación. Su funcionamiento es a partir de las definiciones del entorno establecidos en el archivo Otcl, creando un entorno gráfico que representan al sistema. Además, permite ver que es lo que pasa a lo largo del tiempo, pudiendo personalizar la velocidad de representación, retroceder, avanzar o manejar el tiempo a gusto propio. Esto lo consigue a través del archivo de traza definido anteriormente, que especifica los eventos en cada instante de tiempo, de modo que la NAM se encarga de interpretar dicho archivo en formato de salida de NS2 y convertirlo en un entorno visual más agradable para el usuario.

Como ya hemos mencionado el procesamiento de datos se lo realiza a través del archivo awk, que interpreta el archivo traza pero de una manera más sencilla que la NAM, con el fin de obtener una mayor comprensión del usuario. Para nuestro proyecto hemos creado el archivo awk para obtener el valor promedio de la tasa de rendimiento y así demostrar las configuraciones realizadas en el script para una red dada. Este archivo se muestra en forma de una tabla de datos donde se lo ejecuta después de haber ejecutado el script correspondiente.

# CAPÍTULO 4

## 4. IMPLEMENTACIÓN DEL PROYECTO

### 4.1. Simulación de redes 802.11b y 802.11g

En este capítulo desarrollaremos las simulaciones de la tecnología de una forma didáctica para una mejor comprensión del estudiante, es decir, describiremos desde los pasos básicos hasta llegar a la simulación completa y análisis de los casos existentes que se definen en el estándar.

Los procedimientos básicos se explicarán de una manera simple para que el estudiante o cualquier persona puedan familiarizarse rápidamente con el lenguaje NS2, con la finalidad de crear los componentes necesarios para compilar un escenario inalámbrico.



Una vez que se haya expuesto la primera etapa esencial, procederemos con la simulación de los modos de operación, que como sabemos son el modo Ad-hoc y el modo infraestructura.

En la parte final de este capítulo presentaremos el análisis de los resultados de nuestras simulaciones, así como también describiremos la elaboración de las prácticas realizadas tanto para el estudiante como para el tutor y por último detallaremos nuestra propuesta de los costos económicos necesarios para la implementación de nuestro proyecto.

El lenguaje de programación que se encuentra en este capítulo, está escrito utilizando otro tipo de fuente de letra, para que pueda ser fácilmente diferenciado, y evite confusiones.

#### **4.1.1. Procedimientos para la creación de una red inalámbrica**

Para que el estudiante tenga una idea de lo que comprende un ambiente inalámbrico primero debe tener en cuenta qué partes lo integran, por lo tanto mencionaremos y explicaremos secuencialmente como simular dichos componentes de la red.

Para ello, creamos un script que contendrá nuestra programación en lenguaje Otcl.

Una red inalámbrica en principio consta de uno o más nodos móviles, un punto de acceso, el establecimiento de una transferencia de datos entre ellos y en algunos casos la conectividad con una LAN alámbrica. A continuación describiremos como crear dichos componentes.

### **Creación de un nodo móvil**

Para la elaboración del script abrimos el editor de texto de FEDORA e ingresamos los comandos esenciales en lenguaje Otcl para la creación del nodo móvil. Los primeros comandos que se deben escribir están explicados en el apéndice A en la sección comandos.

En la creación del nodo móvil se necesita declarar varios parámetros que definirán las características que poseerá el nodo, tales como la capa de enlace (LL), interfase de colas (IFQ), capa MAC, el canal inalámbrico en donde se transmite y recibe señales desde los nodos, el tipo de antena, el modelo de propagación, el

tipo de protocolo de ruteo, etc. La configuración API consiste en definir estos tipos de componentes.

Primero, nosotros configuramos los nodos antes de crearlos. La configuración API típica se define como sigue:

```
$ns_ node-config -adhocRouting AODV \  
    -llType      LL \  
    -macType     Mac/802_11 \  
    -propType    "Propagation/TwoRayGround" \  
    -ifqType     "Queue/DropTail/PriQueue" \  
    -ifqLen      50 \  
    -phyType     "Phy/WirelessPhy" \  
    -antType     "Antenna/OmniAntenna" \  
    -channelType "Channel/WirelessChannel" \  
    -topoInstance $topo \  
    -agentTrace  ON \  
    -routerTrace ON \  
    -macTrace    ON \  
    -movementTrace OFF
```

Lo próximo es crear el nodo móvil, para ello escribimos la siguiente sentencia:

```
set node_(0) [$ns_ node]  
$node_(0) random-motion 0
```

Sabemos que el lenguaje Otcl es un programa orientado a objetos, por tal razón, al crear este nodo se le da la dirección de la configuración de los parámetros, así se puede manipular la información de los nodos. El *random-motion* permite que las posiciones y movimientos (velocidad y dirección) de los nodos sean aleatorios. Por este motivo lo hemos deshabilitado (0) para reservarnos la opción de asignarles movimientos específicos. Para la asignación de movimiento a los nodos por favor consultar con la guía de NS2 en el apéndice A en la sección comandos.

### **Creación de un Punto de Acceso**

Al crear un punto de acceso hay que tener en claro que éste sólo existe en el modo infraestructura, lo cual quiere decir que en el escenario deben coexistir un ambiente inalámbrico y otro alámbrico. Por esta razón, el “GOD” (Director General de

Operaciones), el cual es una conjetura específica para NS2 que establece en secuencia los eventos concernientes a los nodos inalámbricos, se define de la siguiente manera:

```
create-god expr[$opt(nn) + 1]
```

Donde la suma de un nodo adicional es para crear el AP que aquí, se define como Estación Base (BS). Para crear el nodo fijo previamente se debe definir una configuración jerárquica tal como se muestra:

```
$ns_ node-config -addressType hierarchical
```

```
AddrParams set domain_num_ 2
```

```
lappend cluster_num 1 1
```

```
AddrParams set cluster_num_ $cluster_num
```

```
lappend eilastlevel 1 2
```

```
AddrParams set nodes_num_ $eilastlevel
```

Realizando esta configuración se especifica el número de dominios, clusters y nodos en esta simulación, donde la parte alámbrica es un dominio y la inalámbrica es otro dominio. El cluster es una entidad jerárquica interna del dominio. El número

de clusters y nodos depende del tipo de simulación, en este caso se trata de un cluster para cada dominio. Se debe especificar a cada nodo, el dominio y cluster que le corresponde. En nuestro presente caso hay un nodo en el primer cluster y dos nodos para el segundo como muestra la sentencia *lappend eilastlevel 1 2*. La creación del nodo fijo se realiza de la siguiente forma:

```
set W(0) [$ns_ node 0.0.0]
```

```
$W(0) set X_ 10.0
```

```
$W(0) set Y_ 10.0
```

```
$W(0) set Z_ 0.0
```

Siendo el dominio 0, cluster 0 y nodo 0. La sentencia *\$W(0) set X\_ 10.0* es para dar una posición al nodo en la parte gráfica.

Lo siguiente es la creación de la BS; tenemos que definir los parámetros de sus componentes como ya hemos mencionado. Hay que recordar que la BS debe tener conectividad con la parte alámbrica, por esta razón, la sentencia *\$ns\_ node-config - wiredRouting ON \* tiene mucha importancia para su configuración. Después procedemos a crear la BS tal como se crea un nodo



móvil, ubicándola en su respectiva jerarquía que corresponde como se muestra:

```
set BS(0) [ $ns_node 1.0.0 ]
```

```
$BS(0) random-motion 0
```

```
$BS(0) set X_ 1.0
```

```
$BS(0) set Y_ 1.0
```

```
$BS(0) set Z_ 0.0
```

Luego creamos los nodos móviles que necesitamos para nuestro análisis de la misma forma que en la sección anterior, sólo que si se trata de más de un nodo, se utiliza un lazo *for* (consulte con el apéndice A). Debemos además asignar la jerarquía correspondiente a los nodos, siendo importante desactivar la configuración alámbrica para comenzar la inalámbrica como se muestra:

```
$ns_node-config -wiredRouting OFF
```

```
set temp {1.0.1}
```

```
set node_(0) [ $ns_node [lindex $temp 0] ]
```

```
$node_(0) base-station [AddrParams addr2id [$BS(0) node-addr]]
```

*\$node\_(0) random-motion 0*

## Creación de la red alámbrica

En esta sección realizamos las conexiones alámbricas entre los nodos, claro está, que para ello utilizamos lo que aprendimos en las secciones anteriores. Adicionalmente, escribimos las sentencias para crear los enlaces fijos dúplex entre ellos como se muestra:

```
$ns_ duplex-link $W(1) $BS(0) 100Mb 2ms DropTail
```

```
$ns_ duplex-link-op $W(1) $BS(0) orient right
```

Se necesita de la sentencia *\$ns\_ duplex-link-op \$W(1) \$BS(0) orient right* donde se relacionan los nodos enlazados con su debida orientación para una mejor visualización en la NAM. Para información adicional sobre lo explicado consultar en el apéndice A en la sección comandos.

## Creación de la transferencia de paquetes entre nodos

La transferencia de datos se realiza con la ayuda de los siguientes protocolos de transporte (TCP y UDP), que utilizan a su vez las siguientes aplicaciones y fuentes de tráfico (FTP, Web, Telnet, CBR, VBR, On-Off).

Al crear una conexión TCP, se deben especificar ciertos parámetros, tales como la fuente, el destino y la conexión entre ellos, es por esta razón que a un nodo se le atribuye el tipo agente (fuente) y al otro nodo el tipo sumidero o sink (destino).

```
set tcp [new Agent/TCP]
```

```
set tcpsink [new Agent/TCPSink]
```

```
$ns attach-agent $W(0) $tcp
```

```
$ns attach-agent $node_(0) $tcpsink
```

```
$ns connect $tcp $tcpsink
```

Para generar tráfico sobre TCP se requiere de la aplicación FTP o Telnet. Nosotros escogimos FTP por su amplia utilización en nuestro medio tal como se muestra:

```
set ftp [new Application/FTP]
```

```
$ftp attach-agent $tcp
```

```
$ns_ at 1.0 "$ftp start"
```

```
$ns_ at 23.0 "$ftp stop"
```

#### 4.1.2. Simulación de modo Ad-hoc

Para desarrollar la simulación en modo ad-hoc necesitamos recordar algunos pasos ya aprendidos como por ejemplo, la creación de nodos móviles, y la transferencia de paquetes, puesto que la arquitectura de modo Ad-hoc es puramente inalámbrica y no necesitamos crear un punto de acceso.

Comenzamos creando 3 nodos móviles con un lazo *for* (como se muestra). Para esto, la configuración utilizada es la API, en donde elegimos como protocolo de ruteo al AODV (vector de distancia en demanda Ad-hoc) que por su definición es para redes Ad-hoc solamente. También declaramos nuestra librería que es la *Mac\802\_11*, y todos los parámetros ya expuestos. En el apéndice A hay información adicional sobre el protocolo AODV.

```
for {set i 0} {$i < $opt(nn)} {incr i} {
```

```

set node_($i) [$ns_ node]
$node_($i) random-motion 0
}

```

Luego, creamos la transferencia de datos. Para ello utilizaremos el agente TCP. La razón es que el protocolo TCP tiene propiedades de reconocimiento, es decir que verifica si los paquetes han llegado a su destino, claro que posee su desventaja, que es más consumo de ancho de banda del canal inalámbrico. La transferencia de datos se la realiza entre los nodos 0 a nodo 2 en los respectivos tiempos.

Al crear una red Ad-hoc 802.11b tenemos que añadir información de capa física, para ello utilizaremos los parámetros de una tarjeta inalámbrica 802.11b que se encuentra en el mercado, como por ejemplo la tarjeta de Orinoco con los siguientes parámetros:

```
Antenna/OmniAntenna set Gt_ 3.0
```

```
Antenna/OmniAntenna set Gr_ 3.0
```

```
Phy/WirelessPhy set CPTresh_ 10.0
```

*Phy/WirelessPhy set CStresh\_ 5.011872e-12*

*Phy/WirelessPhy set RXThresh\_ 5.82587e-09*

*Phy/WirelessPhy set bandwidth\_ 11Mb*

*Phy/WirelessPhy set Pt\_ 0.031622777*

*Phy/WirelessPhy set freq\_ 2.437e9*

*Phy/WirelessPhy set L\_ 1.0*

*Mac/802\_11 set CWMin\_ 31*

*Mac/802\_11 set CWMax\_ 1023*

*Mac/802\_11 set SlotTime\_ 0.000020*

*Mac/802\_11 set SIFS\_ 0.000010*

*Mac/802\_11 set PreambleLength\_ 144*

*Mac/802\_11 set ShortPreambleLength\_ 72*

*Mac/802\_11 set PreambleDataRate\_ 2.0e6*

*Mac/802\_11 set PLCPHeaderLength\_ 48*

*Mac/802\_11 set PLCPDataRate\_ 1.0e6*

*Mac/802\_11 set ShortPLCPDataRate\_ 2.0e6*

*Mac/802\_11 set RTSThreshold\_ 3000*

*Mac/802\_11 set ShortRetryLimit\_ 7*

*Mac/802\_11 set LongRetryLimit\_ 4*

*Mac/802\_11 set dataRate\_ 11Mb*



*Mac/802\_11 set basicRate\_ 1Mb*

En el apéndice B se encuentra esta simulación donde se puede verificar la creación del script que hemos descrito con el nombre de Adhoc\_80211b.tcl.

Los parámetros anteriormente mencionados son absolutamente necesarios para poder obtener la simulación de la tecnología 802.11b puesto que hemos manipulado los datos de la MAC así como los datos de potencia y de ancho de banda.

Para realizar la simulación Ad-hoc 802.11g, escribimos el script de la misma forma que hicimos la simulación de 802.11b, con la diferencia de los parámetros que se insertó en dicho script, por ello vamos a mostrar la diferencia de estos parámetros:

*Antenna/OmniAntenna set Gt\_ 3.0*

*Antenna/OmniAntenna set Gr\_ 3.0*

*Phy/WirelessPhy set CPTthresh\_ 10.0*

*Phy/WirelessPhy set CSTthresh\_ 5.011872e-12*

*Phy/WirelessPhy set RXThresh\_ 5.82587e-09*

*Phy/WirelessPhy set bandwidth\_ 54Mb*

*Phy/WirelessPhy set Pt\_ 0.031622777*

*Phy/WirelessPhy set freq\_ 2.437e9*

*Phy/WirelessPhy set L\_ 1.0*

*Mac/802\_11 set CWMin\_ 15*

*Mac/802\_11 set CWMax\_ 1023*

*Mac/802\_11 set SlotTime\_ 0.000009*

*Mac/802\_11 set CCATime\_ 0.000003*

*Mac/802\_11 set RxTxTurnaroundTime\_ 0.000002*

*Mac/802\_11 set MaxPropagationDelay\_ 0.0000005*

*Mac/802\_11 set SIFSTime\_ 0.000010*

*Mac/802\_11 set PreambleLength\_ 72*

*Mac/802\_11 set PreambleDataRate\_ 6.0e6*

*Mac/802\_11 set PLCPHeaderLength\_ 40*

*Mac/802\_11 set PLCPDataRate\_ 6.0e6*

*Mac/802\_11 set ShortPLCPDataRate\_ 2.0e6*

*Mac/802\_11 set RTSThreshold\_ 3000*

*Mac/802\_11 set ShortRetryLimit\_ 7*

*Mac/802\_11 set LongRetryLimit\_ 4*

*Mac/802\_11 set CCAtime\_ 0.000003*

```
Mac/802_11 set dataRate_ 54Mb
```

```
Mac/802_11 set basicRate_ 6Mb
```

Aquí utilizamos la tasa básica de transmisión de datos para OFDM en 6 Mbps y su ancho de banda en 54Mbps como nos indica el estándar, también el preámbulo corto es necesario según la tecnología.

En el apéndice B se encuentra esta simulación donde se puede verificar la creación del script que hemos descrito con el nombre de Adhoc\_80211g.tcl.

#### **4.1.3. Simulación de modo infraestructura**

Para desarrollar la simulación en modo infraestructura, recordamos los procedimientos para crear una red alámbrica, un nodo móvil, y un punto de acceso. Al igual que se hizo en la sección anterior se puede realizar la simulación de los estándares 802.11b y 802.11g añadiendo los parámetros necesarios para dichas simulaciones.

Primero tenemos que desarrollar una red infraestructura 802.11b cuyos procedimientos se encuentran detallados en el ítem 4.1.1:

un nodo móvil, un nodo fijo y la BS, también necesitamos la transferencia de datos, siendo el agente TCP el que usamos por los argumentos ya expuestos anteriormente. Configuramos el nodo móvil con los siguientes componentes: el protocolo de ruteo DSDV (vector de distancia por secuencia de destino), elegimos nuestra librería que es la *Mac\802\_11*, la interfase de colas IFQ, y los demás componentes con los valores ya usados.

Creamos además la configuración con el tipo de direccionamiento jerárquico con dos dominios, un cluster en cada dominio, un nodo en un dominio y dos nodos en el siguiente. Siendo que el primer dominio representa la parte fija y el otro dominio la parte móvil. Utilizamos la creación de la red alámbrica y colocamos la jerarquía correspondiente siendo el nodo fijo la jerarquía 0.0.0, la BS 1.0.0 y el nodo móvil que registraremos a la BS con la jerarquía 1.0.1. De este modo la BS sabe que sólo se puede transmitir al nodo móvil a través de ella.

La comunicación TCP se la realizó de igual forma que en el procedimiento para la creación de la transferencia de paquetes siendo el nodo  $W(0)$  la fuente y el nodo  $node_{-}(0)$  el destino. Para

consultar este script dirigirse al apéndice B con el nombre `Infra_80211b.tcl`.

Para la simulación del estándar 802.11g vamos a realizar una red infraestructura con IP móvil. Éste es un protocolo el cual permite a un nodo móvil cambiarse de la cobertura de un AP a otro manteniendo su misma dirección IP.

Necesitamos incluir los parámetros para el estándar 802.11g, un nodo fijo, la creación de los APs y un nodo móvil, así como también la transferencia de paquetes TCP.

Entonces creamos la jerarquía como se muestra a continuación:

```
$ns_ node-config -addressType hierarchical
```

```
AddrParams set domain_num_ 3
```

```
lappend cluster_num 1 1 1
```

```
AddrParams set cluster_num_ $cluster_num
```

```
lappend eilastlevel 1 2 1
```

```
AddrParams set nodes_num_ $eilastlevel
```

Luego creamos el GOD donde aumentamos 2 nodos adicionales para representar a los APs. Creamos los nodos APs que en nuestro caso los nombraremos como *HA* y *FA*, respectivamente, como se muestra:

```
create-god [expr $opt(nn) + 2]
```

```
set HA [$ns_ node 1.0.0]
```

```
set FA [$ns_ node 2.0.0]
```

```
$HA random-motion 0
```

```
$FA random-motion 0
```

La creación del nodo fijo se realiza como ya se ha explicado, pero para el nodo móvil tenemos que efectuar unos cambios en los cuales registramos el nodo móvil *MH* al nodo AP llamado *HA*, para ello desactivamos la parte alámbrica como se muestra:

```
$ns_ node-config -wiredRouting OFF
```

```
set MH [$ns_ node 1.0.1]
```

```
set node_(0) $MH
```

```
set HAaddress [AddrParams addr2id [$HA node-addr]]
```

```
[$MH set regagent_] set home_agent_ $HAaddress
```

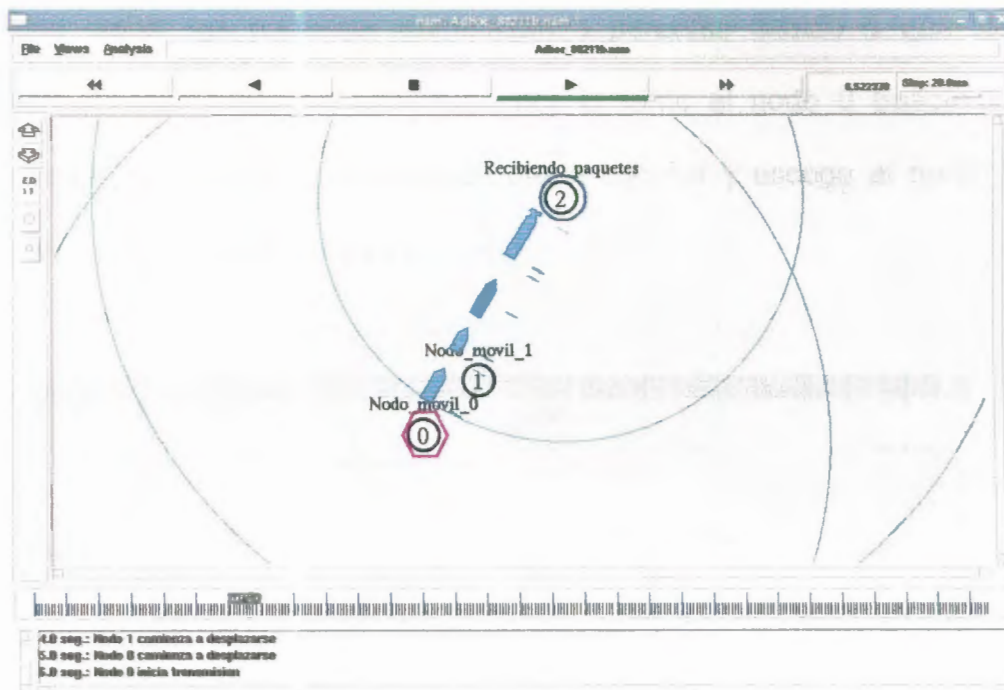


La transferencia de paquetes TCP se la realiza de igual forma que en el estándar anterior siendo el nodo  $W(0)$  la fuente y el nodo  $MH$  el destino. Para consultar este script dirigirse al apéndice B con el nombre `Infra_IP.tcl`.

## 4.2. Resultados de la Simulación

En esta sección, nosotros efectuaremos el análisis de los resultados obtenidos en las simulaciones descritas en la sección anterior, teniendo en cuenta que nuestro objetivo principal es dar a conocer a los estudiantes este tipo de alternativa didáctica para una mejor comprensión de la tecnología inalámbrica utilizada.

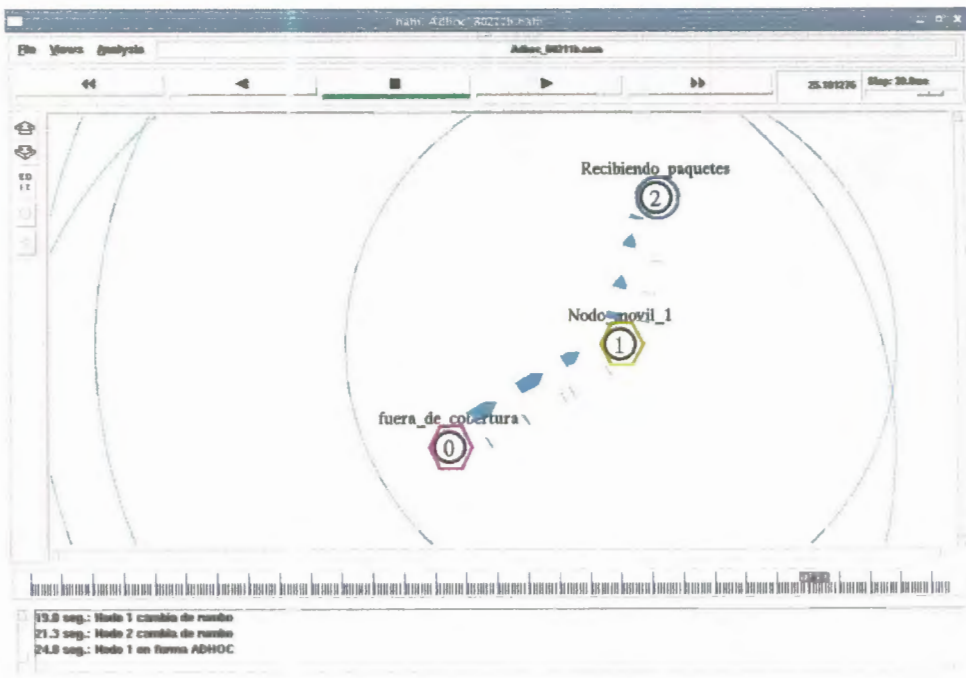
Empezaremos por el análisis del archivo `Adhoc_80211b.tcl`, que como su nombre nos indica se trata de una red Ad-hoc bajo el estándar IEEE 802.11b. Al compilarlo nos muestra los resultados en la NAM de la siguiente forma:



**Figura 4.1:** Simulación de una Red Ad-hoc 802.11b

Como se puede observar, los nodos móviles son representados por los círculos de color negro, numerados de 0 a 2 respectivamente. Al transcurrir el tiempo, observamos que los nodos 2, nodo 1 y nodo 0 comienzan a desplazarse hacia las coordenadas descritas en el script y a transmitir entre ellos paquetes RTS y CTS representados por círculos concéntricos crecientes. En el instante 6 segundos comienza la transmisión de paquetes TCP y ACK entre los nodos 0 y 2 identificados por los rectángulos puntiagudos azules y rectángulos más pequeños también azules, siendo el nodo 0 la fuente y el nodo 2 el destino. Este procedimiento dura hasta el instante 23.2 segundos durante el cual los nodos se movilizan continuamente. Después de

este lapso, los paquetes comienzan a perderse debido a que los nodos están fuera de cobertura, por lo tanto el nodo 0 busca un puente para seguir transmitiendo hasta concluir y escoge al nodo 1 por la cercanía con el nodo 2.



**Figura 4.2:** Instante en el que el Nodo 1 Hace de Puente

Al ejecutar la herramienta awk para que realice el procesamiento del archivo de salida de la simulación, podemos observar los siguientes resultados:

#	Tiempo	Throughput[Mbps]	Eficiencia[%]
6.00768	0.00197081	0.0179164	
7.00881	4.54145	41.2859	
8.00938	4.44932	40.4483	
9.01094	4.43309	40.3008	
10.0115	4.36645	39.695	
11.0126	4.29319	39.029	
12.0138	4.38742	39.8856	
13.0138	4.29777	39.0706	
14.0176	4.25838	38.7126	
15.0192	4.26708	38.7916	
16.0205	4.20975	38.2705	
17.0211	4.20077	38.1888	
18.0243	4.26069	38.7335	
19.0279	4.4475	40.4318	
20.0306	4.26273	38.7521	
21.0307	4.29762	39.0692	
22.0319	4.32814	39.3468	
23.0339	4.32459	39.3144	
24.0355	2.0451	18.5918	
25.0372	2.17487	19.7716	
26.0405	2.17149	19.7408	

ID del Flujo: 0  
 Tipo de Flujo: tcp  
 Nodo Fuente: 0

**Figura 4.3:** Tabla de Resultados de una Red Ad-hoc 802.11b Utilizando el Archivo AWK

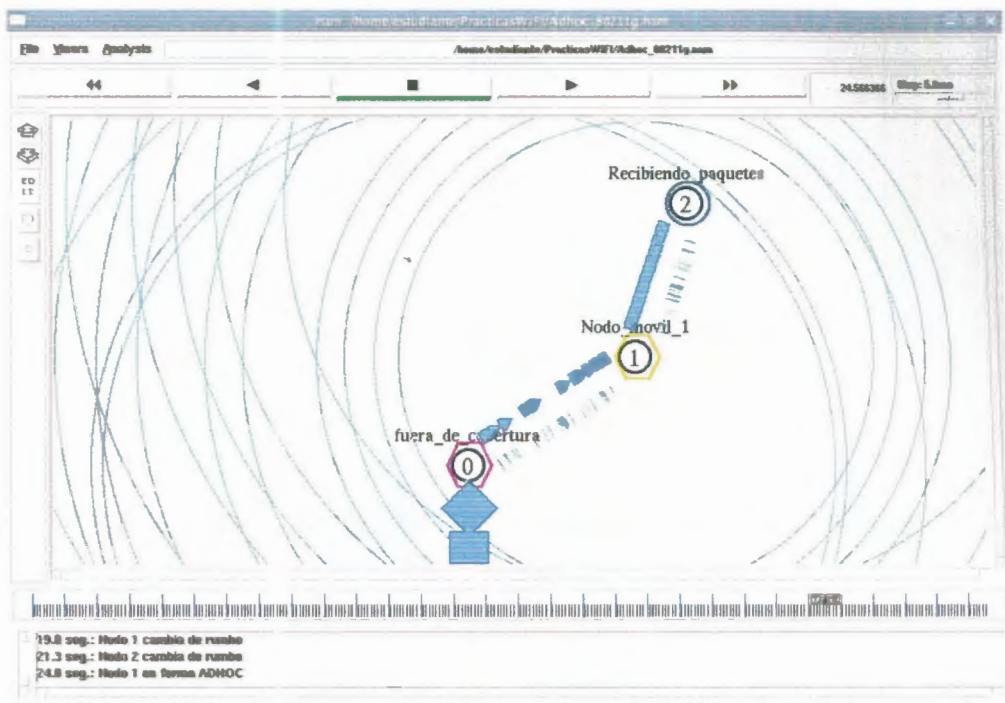
En esta tabla vemos la tasa de rendimiento (throughput) en relación al tiempo transcurrido de la simulación, rindiendo una tasa promedio de 4.3 Mbps hasta el instante 23 segundos a partir del cual disminuye a 2.1 Mbps. Estos valores se atribuyen a diversos factores que inciden en la degradación de la tasa de rendimiento. El estándar 802.11b define una tasa de transmisión de datos de 11Mbps. Debido principalmente al mecanismo de acceso al medio usado en comunicaciones inalámbricas (CSMA/CA) y a otros factores como la distancia entre los móviles, la línea de vista entre ellos y las condiciones de propagación, la tasa de rendimiento disminuye

drásticamente. Aún en condiciones ideales y sin retransmisiones, la máxima tasa de rendimiento que se alcanzaría es de 6.2 Mbps. Esto se debe a que el paquete contiene información de sincronismo y corrección de errores que necesita ser transmitida a una tasa mucho más baja (1 Mbps). Además, los espacios entre trama, el tiempo de ejecución del algoritmo de backoff y la duración de la transmisión del paquete ACK afectan la eficiencia de la tasa de rendimiento. En nuestro caso, ésta se ve aún más disminuida por el intercambio de paquetes RTS y CTS y considerando que hay retransmisiones.

La caída en la tasa de rendimiento al instante 23 segundos tiene su explicación en que, por condiciones de la simulación, el nodo transmisor se aleja de la cobertura del receptor, haciéndose necesaria la intervención de un tercer nodo actuando como puente, lo cual es factible gracias al protocolo de enrutamiento AODV. Al adherirse el tercer nodo a la comunicación, se provoca una reducción del ancho de banda disponible debido a que hay un segundo nodo transmitiendo y recibiendo paquetes lo cual conlleva a la utilización de más retransmisiones, paquetes ACK y doble mecanismo de acceso al medio.



Ahora analizamos el archivo Adhoc\_80211g.tcl, el cual contiene su topología igual a la simulación anterior pero basada en el estándar IEEE 802.11g como se muestra a continuación:



**Figura 4.4:** Simulación de una Red Ad-hoc 802.11g

La diferencia radica en la tasa de transferencia de datos, puesto que el estándar 802.11g utiliza la técnica de modulación OFDM, haciendo que esta tasa sea significativamente mayor (54 Mbps) y por lo tanto envíe mayor cantidad de paquetes.



#	Tiempo	Throughput [Mbps]	Eficiencia [%]
	6.00142	0.00197207	0.00365346
	7.0015	23.465	43.4536
	8.00197	23.4321	43.3929
	9.00238	23.3746	43.2863
	10.0029	22.4372	41.5503
	11.0031	21.0243	38.9339
	12.0035	22.8636	42.34
	13.0039	21.6117	40.0216
	14.0042	21.0699	39.0183
	15.0049	21.1309	39.1313
	16.0054	21.0059	38.8999
	17.0058	20.9369	38.772
	18.0062	20.8995	38.7029
	19.0068	22.9794	42.5545
	20.0072	22.0624	40.8563
	21.0074	21.0829	39.0425
	22.0075	21.6414	40.0766
	23.0077	20.9989	38.8868
	24.0078	11.0107	20.3902
	25.0088	10.9408	20.2608
	26.0097	10.8002	20.0003

ID del flujo: 0  
 Tipo de Flujo: tcp  
 Nodo Fuente: 0

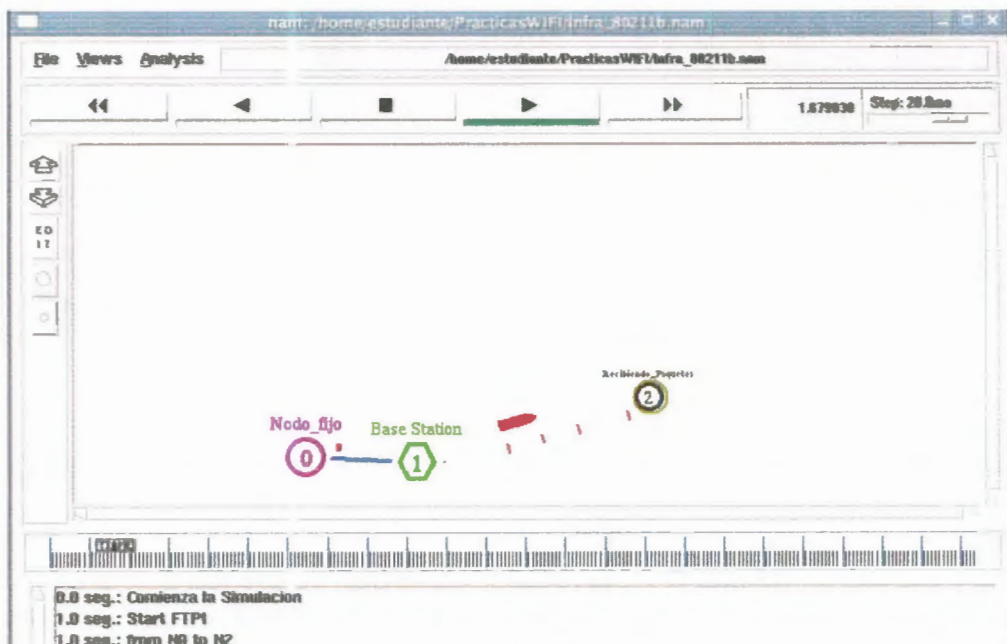
Ln 24, Col 1

**Figura 4.5:** Tabla de Resultados de una Red Ad-hoc 802.11g Utilizando el Archivo AWK

En esta tabla vemos la tasa de rendimiento en relación al tiempo transcurrido de la simulación, rindiendo una tasa promedio de 22.4 Mbps hasta el instante 23 segundos a partir del cual disminuye a 10.9 Mbps debido a que el nodo transmisor se aleja de la cobertura del receptor, haciéndose necesaria la intervención de un tercer nodo actuando como puente. La tasa de rendimiento máxima que se puede alcanzar en condiciones ideales para este estándar es de 27 Mbps sin contar con los RTS/CTS y retransmisiones. Si consideramos estos parámetros, la tasa de rendimiento llega a ser igual a 23.3 Mbps,

entonces el valor obtenido en nuestra simulación está dentro de este rango.

El archivo `Infra_80211b.tcl` es sujeto de análisis en este caso, en donde mostraremos una red infraestructura bajo el estándar IEEE 802.11b con su topología como se muestra a continuación.



**Figura 4.6:** Simulación de una Red Infraestructura 802.11b

Aquí podemos observar una transferencia de paquetes entre un nodo fijo (nodo 0) y un nodo móvil (nodo 2). Esta comunicación se realiza a través de un punto de acceso (nodo 1), que nos permite comunicar la parte alamburada a la parte inalámbrica.

#	Tiempo	Throughput [Mbps]	Eficiencia [%]
1.	0.01101	0.0117111	0.106464
2.	0.01475	4.28189	38.9263
3.	0.01526	4.26023	38.7294
4.	0.01547	4.22601	38.4183
5.	0.01788	4.27574	38.8704
6.	0.01881	4.55419	41.4017
7.	0.02153	4.49882	40.8984
8.	0.02325	4.326	39.3272
9.	0.02519	4.27777	38.8688
10.	0.0264	4.24532	38.5938
11.	0.028	4.46857	40.6234
12.	0.0294	4.46922	40.6293
13.	0.0329	4.53054	41.1867
14.	0.0356	4.23911	38.5373
15.	0.0366	4.18747	38.0679
16.	0.0402	4.30605	39.1459
17.	0.0431	4.32064	39.2786
20.	0.1189	0.0962363	0.874875
21.	0.02519	4.30129	39.1026
22.	0.1234	4.38168	39.8334

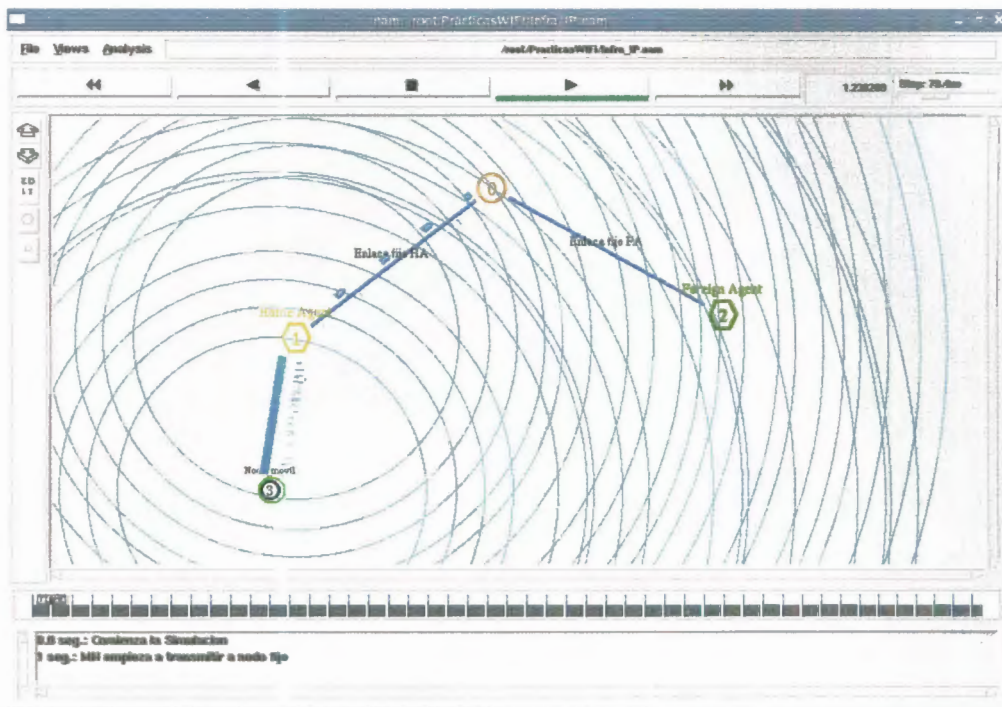
ID del flujo: 0  
 Tipo de Flujo: tcp  
 Nodo Fuente: 0  
 Nodo Destino: 2

Loaded file "home/estudiante/PracticasWiFi/stats5.out" Ln 1, Col 1 1MS

**Figura 4.7:** Tabla de Resultados de una Red Infraestructura 802.11b Utilizando el Archivo AWK

Como estamos trabajando con el estándar 802.11b vemos que la tabla nos muestra valores de tasa de rendimiento dentro de los valores aceptables para una comunicación TCP, pero en algún instante de tiempo la tasa de rendimiento cae a cero debido a que el nodo móvil se aleja lo suficiente como para estar fuera de cobertura del nodo 1 y comienza a perder paquetes hasta el momento que deja de transmitir.

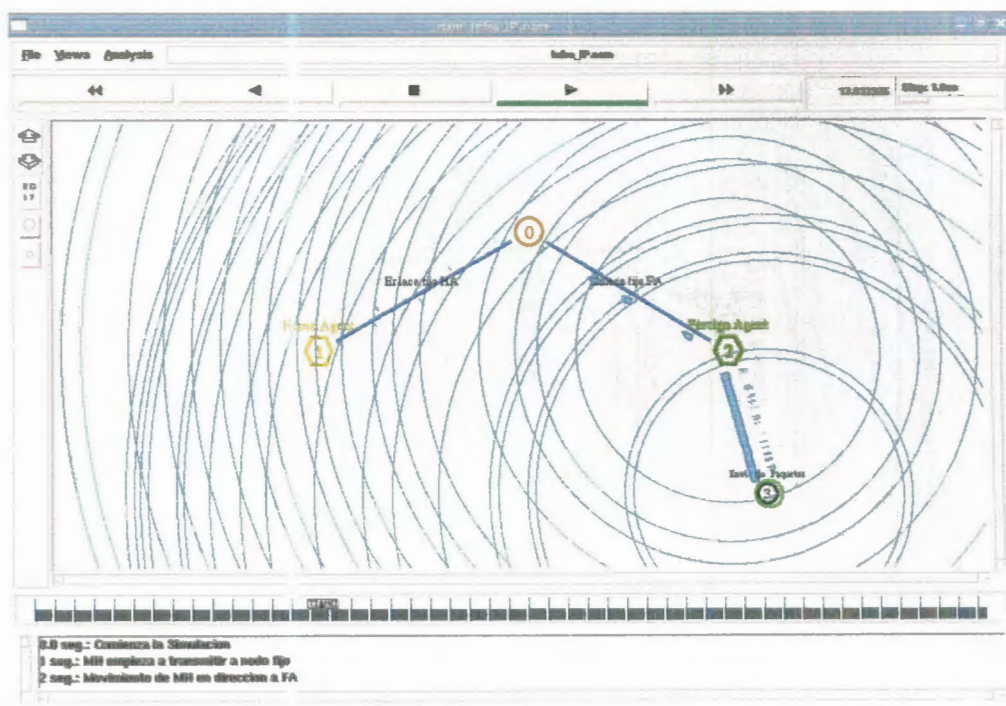
Por último analizaremos el archivo Infra\_IP.tcl que simula una topología de red infraestructura con protocolo IP móvil para el estándar 802.11g como se muestra en la siguiente figura:



**Figura 4.8:** Simulación de una Red Infraestructura 802.11g

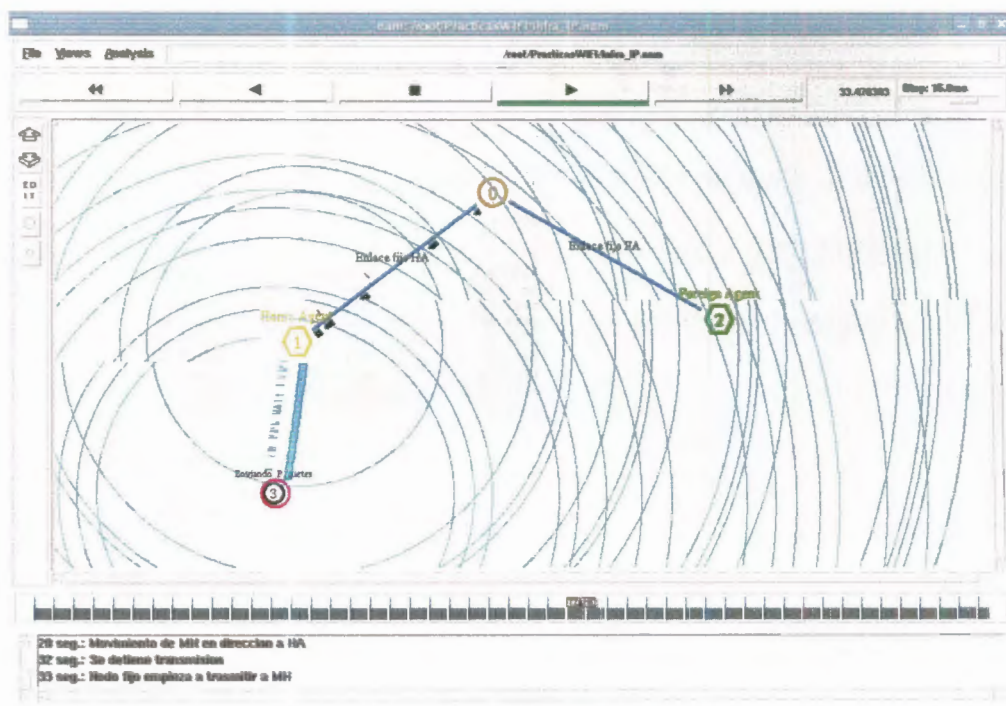
Observamos que nuestra topología consta de un nodo fijo (nodo 0), dos APs el Home Agent (nodo 1) y el Foreign Agent (nodo 2), y de un nodo móvil (nodo 3). El nodo móvil está registrado al Home Agent para que se realice la comunicación a través de él. La transferencia de paquetes se realiza del nodo 3 al nodo 0. Como ya hemos mencionado utilizamos el protocolo Mobile IP que permite al nodo 3 trasladarse de cobertura de AP (de Home Agent a Foreign Agent), manteniendo la misma dirección IP.





**Figura 4.9:** *Nodo 3 Realizando Transferencia de Paquetes a Través del Foreign Agent*

En la figura 4.9 se puede observar que el nodo 3 realiza la transferencia de paquetes con el nodo alámbrico ahora a través del Foreign Agent. Posteriormente en el instante de tiempo 33 segundos, se realiza la misma comunicación pero ahora en dirección opuesta, es decir del nodo 3 al nodo cero (figura 4.10),



**Figura 4.10:** Transmisión del Nodo 0 al Nodo 3

A continuación se muestran los valores de la tasa de rendimiento de esta simulación, para lo cual se tienen que generar dos archivos de salida utilizando las herramientas AWK, esto se debe a que existen dos comunicaciones. Una del nodo 3 al nodo 0, y la otra del nodo 0 al nodo 3.



Tiempo	Throughput [Mbps]	Eficiencia [%]
1.00683	0.0117636	0.0217845
2.00751	21.2504	39.3525
3.00776	21.5433	39.8951
4.0082	21.137	39.1425
5.32926	0	0
15.9299	0	0
16.9299	0.529615	0.980768
17.93	19.321	35.7797
18.9302	21.4487	39.7197
19.9307	20.8644	38.6378
20.9308	22.1981	41.1075
21.9314	21.3576	39.551
28.9561	2.49114	4.61322
29.9561	21.1579	39.1814
30.9564	20.8447	38.6012

ID del flujo: 0  
 Tipo de Flujo: tcp  
 Nodo Fuente: 3  
 Nodo Destino: 0  
 Tamaño del Paquete: 1480  
 Paquetes Recibidos: 19980  
 Tput promedio[Mbps]: 16.4735  
 Retardo promedio[ms]: 0

**Figura 4.11:** Tabla de Resultados de la Red Infraestructura 802.11g – Nodo 3 a Nodo 0

Tiempo	Throughput [Mbps]	Eficiencia [%]
1.00683	0.0117636	0.0217845
2.00751	21.2504	39.3525
3.00776	21.5433	39.8951
4.0082	21.137	39.1425
5.32926	0	0
15.9299	0	0
16.9299	0.529615	0.980768
17.93	19.321	35.7797
18.9302	21.4487	39.7197
19.9307	20.8644	38.6378
20.9308	22.1981	41.1075
21.9314	21.3576	39.551
28.9561	2.49114	4.61322
29.9561	21.1579	39.1814
30.9564	20.8447	38.6012

ID del flujo: 0  
 Tipo de Flujo: tcp  
 Nodo Fuente: 3  
 Nodo Destino: 0  
 Tamaño del Paquete: 1480  
 Paquetes Recibidos: 19980  
 Tput promedio[Mbps]: 16.4735  
 Retardo promedio[ms]: 0

**Figura 4.12:** Tabla de Resultados de la Red Infraestructura 802.11g – Nodo 0 a Nodo 3

Se muestran las tablas de resultados donde los valores de tasa de rendimiento obtenidos concuerdan con los valores de tasa de rendimiento que debe tener la red para un paquete TCP. En los instantes que la tasa de rendimiento se vuelva cero es debido a que el nodo móvil se encuentra fuera de cobertura del Home Agent y por lo tanto no se realiza la transmisión.

#### **4.3. Descripción de la guía de las prácticas y del manual del tutor**

Una vez desarrolladas las simulaciones que exponen el funcionamiento y comportamiento de los estándares de redes inalámbricas anteriormente analizados, se procede a la elaboración de prácticas de laboratorio en donde se implementan estas simulaciones con un enfoque pedagógico. Estas prácticas tienen como meta, que tanto el estudiante como la persona particular involucrada en el ámbito tecnológico, puedan comprender y asimilar de una manera sencilla y detallada estas tecnologías.

Se han desarrollado dos tipos de prácticas, la primera nos explica e involucra en el desarrollo de simulaciones de redes utilizando NS2, esta práctica tiene como objetivo hacer que el estudiante conozca y pueda desarrollar por su propia cuenta, simulaciones de redes. Consta

de dos experimentos, el primero es la creación de una red alámbrica en la cual hacemos que el mismo estudiante cree la simulación desde cero, y se explican los comandos utilizados para el desarrollo de la misma. En el segundo experimento se procede a desarrollar una red inalámbrica, cuya meta es que el estudiante a parte de poder desarrollar la simulación, también conozca los comandos necesarios para la creación de este tipo de redes.

La segunda práctica y objetivo principal de este proyecto, consiste en simulaciones de redes inalámbricas de 802.11b y 802.11g, en donde damos a conocer estas dos tecnologías. Esta práctica consiste en un conjunto de experimentos en donde se pueden observar los 2 modos de operación anteriormente explicados en conjunto con los parámetros necesarios para simular las tecnologías.

El primer conjunto de experimentos comprende la creación y visualización de redes Ad-hoc, en donde se han colocado los parámetros requeridos para poder simular redes 802.11b y 802.11g, y se puedan observar las diferencias entre ellas tanto visual como analíticamente. Para ello hacemos uso de la NAM y del archivo AWK, este último nos permite observar la tasa de rendimiento de cada tecnología.

El segundo conjunto de experimentos contenidos en esta práctica, corresponde a la creación y visualización de redes infraestructura, en donde se hace el mismo análisis anteriormente descrito, pero también se añade la implementación del protocolo IP Móvil, que es utilizado en los modos de operación infraestructura.

En cada experimento se ha procurado plantear preguntas clave en los momentos más interesantes con la finalidad de que el estudiante capte de la mejor manera posible la idea principal de lo que está tratando la práctica y asimismo incentivarlo a la investigación. Dentro de este conjunto de interrogantes, existen aquellas que permiten hacer modificaciones a las simulaciones, que tienen el objeto de crear dinamismo en las prácticas, es por ese motivo que se desarrolló la primera práctica introductoria al NS2 que hará que los estudiantes logren este objetivo. Hay otro conjunto de interrogantes en las que es necesario utilizar el archivo AWK para generar una tabla de resultados e importar estos datos a Excel para generar gráficas bidimensionales que expliquen de mejor manera el comportamiento de la simulación.

Se han desarrollado también manuales de tutor, en donde se encuentran resueltas las interrogantes propuestas en las prácticas



procurando ser lo más detallado posible. De esta manera, la persona a cargo de dirigir la clase tendrá a su disposición una guía para poder orientar al estudiante y lograr que tome en consideración los momentos críticos y detalles de cada simulación.

#### **4.4. Costos de Implementación**

El implementar las prácticas, requiere esencialmente el uso de computadoras, las cuales deben poseer componentes que nos permitan operar NS2 de una manera ágil y con tiempos de compilación reducidos. En nuestro caso particular, nuestra universidad dispone en el laboratorio de telecomunicaciones de los equipos computacionales necesarios para poner en práctica las simulaciones sin ningún problema, y al ser NS2 un software libre, esto hace que los costos de adquisición de licencia sean nulos.

Ahora, considerando la creación de un nuevo laboratorio de telecomunicaciones o un laboratorio ya constituido el cual no posea computadoras, detallamos a continuación una lista de los componentes mínimos necesarios para correr NS2 a tener en consideración al momento de la adquisición de computadoras:

<b>COMPONENTE</b>	<b>COSTO UNITARIO</b>
Procesador Intel Pentium 4 – 2 GHz o procesador equivalente	\$100.00
Tarjeta Madre con Sonido, video y Lan Integrado	\$50.00
Memoria RAM 512MB	\$50.00
Disco Duro de 40GB	\$40.00
Unidad Lectora de CD's	\$30.00
Carcasa, teclado, Mouse, parlantes	\$50.00
Monitor CRT de 15 Pulgadas	\$100.00
Adquisición e instalación de sistema operativo Linux Fedora Core 4	\$0.00
Adquisición e instalación de programa NS2	\$0.00
<b>TOTAL:</b>	<b>\$420.00</b>

**Tabla 1: Costos de Implementación**

Se pretende que el laboratorio tenga un máximo de 10 estudiantes y que sean distribuidos en grupos de dos personas por máquina, por lo que tendríamos un total general de:

$$5 * \$420.00 = \$2100.00$$



## **CONCLUSIONES Y RECOMENDACIONES**

## CONCLUSIONES

1. La investigación profusa y exhaustiva que se realizó sobre los estándares de redes inalámbricas 802.11b y g, sus arquitecturas, protocolos, técnicas de acceso y demás minuciosos detalles, nos ha servido para poder entender el funcionamiento, aplicaciones y beneficios de estas tecnologías, así como las mejoras que se implementan en el estándar g sobre su predecesor. Este extensivo estudio teórico previo efectuado fue el cimiento de nuestro trabajo ya que nos permitió seleccionar los temas y escenarios fundamentales que fueron desarrollados en el transcurso del proyecto.
2. El esfuerzo empleado en el aprendizaje, investigación, documentación e implementación de los scripts en el lenguaje Otcl del NS2, tuvo su recompensa en ver las simulaciones finalizadas con los aspectos que deseábamos cubrir plasmados en ellas, necesarios para la comprensión del funcionamiento de las tecnologías tales como modos de operación y métodos de acceso.
3. El esmero que hemos inyectado para la utilización de la interfaz gráfica NAM nos ha permitido que de manera visual puedan ser observados el funcionamiento, topología, transmisión de datos y comportamiento de las simulaciones creadas, además de apreciar,

corroborar o descartar las hipótesis sobre los eventos que de acuerdo a las configuraciones se suscitan. El hecho de tener implementada la interfaz visual desde el punto de vista del aprendizaje, resulta eficiente y ventajoso.

4. Se desarrolló una herramienta que nos permite procesar la información de salida que nos entrega el simulador (archivos "trace"). De esta manera podemos obtener datos de parámetros importantes que son puntos clave de análisis en toda red de transmisión de datos, como lo es la tasa de rendimiento en relación al tiempo transcurrido. Esta función fue implementada en el lenguaje de programación AWK y su finalidad y utilidad radican en que nos da la ventaja de presentar los datos anteriormente mencionados en forma de tabla de resultados lo cual evidentemente es más provechoso, ya que es posible observar y discutir paso a paso cada acontecimiento que sucede durante la simulación.
5. Las prácticas de laboratorio son el aporte principal del proyecto, utilizan como fundamento las simulaciones desarrolladas en los capítulos anteriores. Contienen objetivos claros, marco teórico, experimentos concretos, interrogantes desafiantes, interacción directa con el lenguaje de programación y con el sistema operativo, todo con

la finalidad de involucrar al estudiante dinámicamente y despertar su interés al aprendizaje de estos estándares.

## RECOMENDACIONES

1. Recomendamos utilizar estas prácticas a estudiantes universitarios o personas particulares que deseen aprender sobre estas tecnologías inalámbricas. Todo el software utilizado en la elaboración de este proyecto es de código abierto y se lo puede conseguir gratuitamente.
2. Los tutores encargados de los laboratorios donde pueden ser implementadas estas practicas, recomendamos que sean personas con sólidos conocimientos de lenguaje Otcl y de NS2 para de esta forma y si fuera necesario, realicen modificaciones a las simulaciones a su conveniencia, para que así aporten al desarrollo de estas prácticas.
3. Recomendamos a toda persona involucrada en el ámbito tecnológico y que particularmente en el estudio y análisis de las redes inalámbricas, que realicen un autoaprendizaje e investigación previo al uso de estas practicas para así obtener el máximo provecho de este trabajo.

# APÉNDICES



# APÉNDICE A

## GUÍA DE NS2

### A.1 CONCEPTO

NS (más conocido como NS-2 por su versión actual) es un simulador de redes de eventos discretos utilizado principalmente en ambientes académicos debido a que está escrito en código abierto y a la abundancia de documentación en línea. Se pueden simular tanto protocolos unicast como multicast y se utiliza intensamente en la investigación de redes de datos. Puede simular una amplia gama de protocolos tanto para redes cableadas o redes wireless, así como mixtas.

### A.2 INSTALACION

NS2 funciona bajo el sistema operativo Linux, a continuación se detallan los pasos para la adquisición e instalación de este software:

- Descargar el archivo comprimido “ns-allinone-2.31.tar.gz” de la siguiente dirección:

[http://sourceforge.net/project/downloading.php?group\\_id=149743&use\\_mirror=superb-west&filename=ns-allinone-2.31.tar.gz&9258915](http://sourceforge.net/project/downloading.php?group_id=149743&use_mirror=superb-west&filename=ns-allinone-2.31.tar.gz&9258915)

- Abrir el archivo descargado y extraer la información contenida en la siguiente dirección:

```
/usr/local/
```

- Dar clic derecho en el escritorio y abrir una ventana de terminal, ejecutar la siguiente sentencia para dirigirte a la dirección donde se encuentra la carpeta extraída:

```
cd /usr/local/ns-allinone-2.31/
```

- Ejecutar la siguiente sentencia que compilará e instalará todos los paquetes contenidos:

### *./install*

- Este proceso demora unos cuantos minutos y al finalizar mostrará los resultados de la instalación, procedemos ahora a crear los enlaces simbólicos del NS2 y de la NAM, para lo cual ejecutamos las siguientes sentencias en la ventana de terminal:

```
ln -s /usr/local/ns-allinone-2.31/ns-2.31/ns /usr/local/bin
```

```
ln -s /usr/local/ns-allinone-2.31/nam-1.13/nam /usr/local/bin
```

- Terminado este proceso ejecutaremos en la ventana de terminal las sentencias **ns** y luego **nam** para comprobar que los enlaces simbólicos han sido creados, de esta manera al ejecutar estas sentencias se abrirán automáticamente los programas.

### **A.3 COMANDOS**

Aquí explicaremos los comandos más utilizados a la hora de desarrollar un script, así como las sentencias obligatorias para una correcta implementación de una simulación. Para una mejor comprensión de este documento nosotros hemos diferenciado los comandos de las variables, siendo los comandos escritos con negrita.

Al realizar el código fuente de un programa específico debemos comentar las sentencias utilizadas para cuando algún programador desea revisar nuestro código fuente comprenda su estructura, para ello en nuestro script al iniciar un comentario en una línea nueva se utiliza el símbolo **#** y para cuando se comenta una sentencia específica, a un lado de esta se digita **;** para comentarla.

Por lo tanto definiremos las siguientes sentencias obligatorias:

```
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.

```
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.

Lo siguiente es crear una Topología Objeto que mantiene huellas del movimiento de los nodos móviles dentro de una limitación topológica.

```
set variable_topografia [new Topography]  
$variable_topografia load_flatgrid x y
```

A continuación crearemos el Objeto GOD:

```
create-god nn
```

GOD (General Operations Director) es el director general de operaciones o también llamado “observador poderoso”, que es usado como registro de información global a cerca de los estados del ambiente, las redes o nodos. Normalmente, el objeto God registra el número total de nodos móviles y el número de saltos requeridos para la búsqueda desde un nodo a otro. La información del próximo salto es cargada dentro del objeto God desde el modelo de movimiento de archivos, antes del comienzo de la simulación, hasta cálculos de sus vuelos durante la corrida del simulador que puede ser el tiempo consumido. Una sola instancia global del objeto GOD es usada durante la simulación. Para la evaluación de la funcionalidad, el objeto GOD es llamado internamente objeto MAC en los nodos móviles. **nn** esta orientado al número de nodos inalámbricos que se desea para la simulación.

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

```

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*.

Ahora definiremos las sentencias más utilizadas para el desarrollo de un escenario inalámbrico, así como los tipos de configuración que existen.

Un nodo móvil consiste de componente de la red tales como la capa de enlace (LL), interfase de colas (IFQ), capa MAC, el canal inalámbrico en donde se transmite y recibe señales desde los nodos, el tipo de antena, el modelo de propagación, el tipo de protocolo de ruteo, etc. La configuración API consiste en definir estos tipos de componentes.

### *\$ns\_ node-config*

esta sentencia me indica que voy a configurar un nodo móvil, la configuración API se define como sigue:

```

$ns_ node-config -addressingType flat o hierarchical o expanded
    -adhocRouting AODV o DSDV o DSR o TORA
    -llType LL
    -macType Mac/802_11
    -propType "Propagation/TwoRayGround"
    -ifqType "Queue/DropTail/PriQueue"
    -ifqLen 50
    -phyType "Phy/WirelessPhy"
    -antType "Antenna/OmniAntenna"
    -channelType "Channel/WirelessChannel"

```

**-topoInstance** *variable\_topografia*  
**-energyModel** *"EnergyModel"*  
**-initialEnergy** *(in Joules)*  
**-rxPower** *(in W)*  
**-txPower** *(in W)*  
**-agentTrace** *ON o OFF*  
**-routerTrace** *ON o OFF*  
**-macTrace** *ON o OFF*  
**-movementTrace** *ON o OFF*

todos los valores por default para estas opciones son NULL excepto el addressingType que es flat. Lo próximo es crear los nodos móviles, para ello escribimos la siguiente sentencia:

```

set node(id_nodo) [$variable_simulador node]
$node_(id_nodo) random-motion 0

```

sabemos que el lenguaje Otcl es un programa orientado a objetos por tal razón al crear este nodo se le da la dirección de la configuración de los parámetros y así se poder manipular la información de los nodos. El random-motion da las posiciones y movimientos (velocidad y dirección) aleatorio de los nodos, por este motivo se ha deshabilitado (0) y nosotros podremos colocar la posición y movimiento de dichos nodos.

La siguiente sentencia indica las coordenadas del nodo en el escenario de simulación:

```

$node(id_nodo) set X_ x.0
$node(id_nodo) set Y_ y.0
$node(id_nodo) set Z_ z.0

```

Los tipos de transferencia de paquetes 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 transmisor y receptor.

***\$variable\_simulador at t\_inicio "\$var\_ftp start"***  
***\$variable\_simulador at t\_final "\$var\_ftp stop"***

estas sentencias realizan el proceso de inicio y final de la transferencia de paquetes FTP

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.

***\$variable\_simulador at t\_inicio "\$var\_cbr start"***  
***\$variable\_simulador at t\_final "\$var\_cbr stop"***

estas sentencias realizan el proceso de inicio y final de la transferencia de paquetes CBR



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*".

## ENLACE FIJO

La sentencia para crear un nodo fijo esta descrito aquí, donde especificamos el ancho de banda, el retardo y el tipo de encolamiento que se desea, tal como vemos a continuación:

```
$variable_simulador duplex-link $nodo_fijo1 $nodo_fijo2 anchoBanda retardo tipoCola
```

el tipo de cola puede escogerse entre DropTail, RED, CBQ, etc. El ancho de banda esta en Mega bits por segundo y el retardo esta en milisegundos.

Para una mejor visualización en la NAM se orienta el enlace creado de la siguiente forma:

```
$variable_simulador duplex-link-op $nodo_fijo1 $nodo_fijo2 orient right-down
```

donde la orientación puede ser *right*, *left*, *up*, *down*, *right-down*, *left-down*, *right-up*, y *left-up*.

## MOVIMIENTO DE LOS NODOS

El movimiento de los nodos se realiza en un ambiente inalámbrico para una mayor aproximación con la realidad, para ello utilizamos la siguiente sentencia:

```
$variable_simulador at t3 "$node(id_nodo) setdest pos_X pos_Y velocidad"
```

donde  $t3$  significa que es tiempo en el cual se realiza el movimiento, y el comando *setdest* hace que comience a moverse el nodo hacia el destino (x, y) a una velocidad dada. Esta configuración es usada para cambiar de dirección y velocidad del movimiento de los nodos móviles.

El procedimiento descrito a continuación define el tamaño del nodo y lo posiciona inicialmente en las coordenadas ya mencionadas.

*\$variable* *\_simulador* *initial\_node\_pos* *\$node(id\_nodo)* tamaño

## LAZOS

El lazo más utilizado es el lazo *for* debido a que nos limita el número de repeticiones de nuestra simulación, su programación es similar al lenguaje C++, como se muestra a continuación:

```
for {set variable 0} {$variable < nn} {incr variable} {  
procedimiento  
}
```

## PROTOCOLOS

Los protocolos de análisis para esta guía son el DSDV y el AODV.

**DSDV (*vector de distancia por secuencia de destino*):** es esencialmente una modificación del algoritmo de encaminamiento Vector Distancia Bellman- Ford, bien conocido por su utilidad en redes fijas, como por ejemplo en el protocolo RIP. En este algoritmo, los nodos vecinos intercambian periódicamente (proactivo) sus tablas de encaminamiento enteras para estimar la distancia a la que se encuentran los demás nodos no vecinos. Las modificaciones introducidas por DSDV proporcionan básicamente la obtención de rutas sin bucles mediante la introducción de números de secuencia para la determinación de las rutas más nuevas. Aunque DSDV sólo proporciona un camino para cada destino, siempre elige el camino más corto basándose en el número de saltos hacia este destino. DSDV utiliza dos tipos de mensajes de actualización, uno más grande (*full-dump*) y otro mucho más pequeño (incremental). Los mensajes incrementales pueden utilizarse para actualizaciones intermedias entre envíos periódicos (*fulldump*) de la tabla entera de encaminamiento. Además se realizan estimaciones de los tiempos de establecimientos de ruta que retrasarán el envío de mensajes incrementales para evitar envíos en cadena de estos mensajes.

**AODV (*vector de distancia en demanda Ad-hoc*):** En el protocolo AODV los nodos mantienen una tabla de encaminamiento para los destinos conocidos (empleando el algoritmo vector distancia). Inicialmente esta tabla estará formada por sus vecinos. Solamente se le añadirán destinos nuevos cuando sea necesario, es decir, cuando un nodo necesita comunicarse con otro que no está en su tabla, inicia un proceso de descubrimiento de ruta (reactivo) hacia el destino concreto. Para ello se emiten mensajes de descubrimiento de ruta RREQ que se van propagando entre todos los nodos de modo similar al DSR. En cambio, aquí los nodos generan una tabla de

encaminamiento inversa para que puedan regresar las contestaciones RREP a las solicitudes de ruta al nodo que la originó. Se recomienda el uso de mensajes HELLO entre vecinos para determinar la conectividad, aunque para reducir el volumen de estos mensajes, sólo debe permitirse su envío a los nodos que estén transmitiendo datos. Debemos destacar además la utilización de las técnicas de "búsqueda secuencial por anillos" y "reparación local del enlace" así como también que es capaz de proporcionar soporte *multicast*.

# APÉNDICE B

## Código Fuente de las Simulaciones

En este apéndice se encuentra el script de cada simulación realizada para el desarrollo de este proyecto de graduación, siendo de mucha importancia su implementación.

### B.1 Archivo Adhoc\_80211b.tcl

```
#####  
# Definir Opciones  
#####  
set opt(chan)      Channel/WirelessChannel  ;# Tipo de canal  
set opt(prop)      Propagation/TwoRayGround ;# Modelo de propagacion  
set opt(netif)     Phy/WirelessPhy         ;# Tipo de interface de red  
set opt(mac)       Mac/802_11              ;# Tipo de MAC  
set opt(ifq)       Queue/DropTail/PriQueue ;# Tipo de cola  
set opt(ll)        LL                       ;# Tipo de capa de enlace  
set opt(ant)       Antenna/OmniAntenna     ;# Modelo de antena  
set opt(ifqlen)    50                       ;# Maximo de paquetes en cola  
set opt(nn)        3                       ;# Numero de nodos moviles  
set opt(rp)        AODV                     ;# Protocolo de enrutamiento  
set opt(x)         90                       ;#X dimension de topologia  
set opt(y)         90                       ;#Y dimension de topologia  
set opt(stop)     30.0                     ;# Tiempo de simulacion  
set opt(packet_size) 1480                   ;# Tamano del paquete  
  
#####  
# Programa Principal  
#####  
#Configuracion para Tarjeta de red Oricono 802.11b -> 22.5 Metros de Rango  
#La altura de la antena transmisora y receptora es de 1.5m.  
#El modelo de propagacion utilizado es el de 2 Rayos.  
  
Antenna/OmniAntenna set Gt_ 3.0  
Antenna/OmniAntenna set Gr_ 3.0  
  
Phy/WirelessPhy set L_ 1.0  
Phy/WirelessPhy set freq_ 2.437e9 ;# Canal 6  
Phy/WirelessPhy set bandwidth_ 11Mb  
Phy/WirelessPhy set Pt_ 0.03162277  
Phy/WirelessPhy set CPTthresh_ 10.0  
Phy/WirelessPhy set CSTthresh_ 5.011872e-12  
Phy/WirelessPhy set RXTthresh_ 5.82587e-09  
  
#Definiendo parametros de estandar IEEE 802.11b
```



```

Mac/802_11 set CWMin_          31
Mac/802_11 set CWMax_          1023
Mac/802_11 set SlotTime_       0.000020 ;# 20us
Mac/802_11 set SIFSTime_       0.000010 ;# 10us
Mac/802_11 set PreambleLength_ 144 ;# 144 bit
Mac/802_11 set ShortPreambleLength_ 72 ;# 72 bit
Mac/802_11 set PreambleDataRate_ 1.0e6 ;# 1Mbps
Mac/802_11 set PLCPHeaderLength_ 48 ;# 48 bits
Mac/802_11 set PLCPDataRate_   1.0e6 ;# 1Mbps
Mac/802_11 set ShortPLCPDataRate_ 2.0e6 ;# 2Mbps
Mac/802_11 set RTSThreshold_    3000 ;# bytes
Mac/802_11 set ShortRetryLimit_ 7 ;# retransmisiones
Mac/802_11 set LongRetryLimit_  4 ;# retransmisiones

Mac/802_11 set dataRate_ 11Mb ;# 802.11 velocidad de transmision de datos
Mac/802_11 set basicRate_ 1Mb ;# 802.11 velocidad basica de transmision

```

```

Agent/TCP set packetSize_ $opt(packet_size)

```

```

# Removiendo cabeceras innecesarias
remove-all-packet-headers ;# removiendo todas excepto las comunes
add-packet-header IP LL Mac ARP TCP ;# cabeceras necesarias

```

```

# Definiendo una instancia del simulador

```

```

#
set ns_ [new Simulator]
set tracefd [open "Adhoc_80211b.tr" w]
set tracenam [open Adhoc_80211b.nam w]
$ns_ use-newtrace

```

```

$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $tracenam $opt(x) $opt(y)
$ns_ color 0 steelblue

```

```

# Definiendo el objeto "Topografia"
set topo [new Topography]

```

```

$topo load_flatgrid $opt(x) $opt(y)

```

```

#
# Creando DIOS
#
create-god $opt(nn)

```

```

#
# 3 nodos seran creados : node(0), node(1) and node(2)
#

```

```

# configuracion del nodo

```



```

$ns_ node-config -adhocRouting $opt(rp) \
    -llType $opt(ll) \
    -macType $opt(mac) \
    -ifqType $opt(ifq) \
    -ifqLen $opt(ifqlen) \
    -antType $opt(ant) \
    -propInstance [new $opt(prop)] \
    -phyType $opt(netif) \
    -channel [new $opt(chan)] \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace ON \
    -movementTrace OFF
#
# Creando un numero especifico de nodos moviles [$opt(nn)] y aderiendolos al canal
#
for {set i 0} {$i < $opt(nn)} {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0
}
# Definiendo la posicion inicial de los nodos
#
$node_(0) set X_ 5.00
$node_(0) set Y_ 4.00
$node_(0) set Z_ 0.00
$node_(1) set Z_ 0.00
$node_(1) set Y_ 15.35
$node_(1) set X_ 37.04
$node_(2) set Z_ 0.00
$node_(2) set Y_ 69.37
$node_(2) set X_ 14.25
#
# Definiendo la velocidad y posicion final de los nodos moviles en un tiempo dado
#
$ns_ at 3.0 "$node_(2) setdest 54.46 76.51 12.37"
$ns_ at 4.0 "$node_(1) setdest 35.82 35.85 14.91"
$ns_ at 5.0 "$node_(0) setdest 75.66 79.49 19.15"
#
$ns_ at 10.2 "$node_(1) setdest 62.78 41.12 17.37"
$ns_ at 11.2 "$node_(0) setdest 36.65 88.54 14.99"
$ns_ at 12.0 "$node_(2) setdest 65.63 71.44 19.38"
#
$ns_ at 18.2 "$node_(0) setdest 33.32 16.92 12.37"
$ns_ at 19.0 "$node_(1) setdest 72.78 41.12 19.32"
$ns_ at 21.3 "$node_(2) setdest 81.65 75.38 14.92"
#
$ns_ at 28.5 "$node_(2) setdest 86.91 16.31 12.58"

```

```
$ns_ at 27.0 "$node_(1) setdest 63.49 75.50 14.25"  
$ns_ at 29.6 "$node_(0) setdest 55.13 29.42 19.83"
```

```
# Definiendo la posicion inicial de los nodos en la nam in NAM
```

```
# 7 define el tamaño del nodo en la NAM, se lo debe ajustar acorde al escenario  
# Esta funcion debe ser llamada despues de ser definida la tipo de movilidad de los nodos
```

```
for {set i 0} {$i < $opt(nn)} {incr i} {
```

```
    $ns_ initial_node_pos $node_($i) 7  
}
```

```
# configurando el flujo de trafico entre los nodos  
# Conexion TCP entre los nodos 0 y 2:
```

```
set tcp_(0) [new Agent/TCP]  
$tcp_(0) set fid_0  
set sink_(0) [new Agent/TCPSink]  
$ns_ attach-agent $node_(0) $tcp_(0)  
$ns_ attach-agent $node_(2) $sink_(0)  
$ns_ connect $tcp_(0) $sink_(0)  
set ftp_(0) [new Application/FTP]  
$ftp_(0) attach-agent $tcp_(0)  
$ns_ at 6.0 "$ftp_(0) start"  
$ns_ at 27.0 "$ftp_(0) stop"
```

```
#  
# Anadiendo anotaciones y definiendo colores y etiquetas a los nodos  
#
```

```
$ns_ at 0.0 "$ns_ trace-annotate \"0.0 seg.: Empieza la simulacion\""  
$ns_ at 0.0 "$node_(0) label Nodo_movil_0"  
$ns_ at 0.0 "$node_(1) label Nodo_movil_1"  
$ns_ at 0.0 "$node_(2) label Nodo_movil_2"  
$ns_ at 0.0 "$node_(2) add-mark m2 navy circle"
```

```
$ns_ at 3.0 "$ns_ trace-annotate \"3.0 seg.: Nodo 2 comienza a desplazarse\""  
$ns_ at 4.0 "$ns_ trace-annotate \"4.0 seg.: Nodo 1 comienza a desplazarse\""  
$ns_ at 5.0 "$ns_ trace-annotate \"5.0 seg.: Nodo 0 comienza a desplazarse\""  
$ns_ at 6.0 "$ns_ trace-annotate \"6.0 seg.: Nodo 0 inicia transmision\""  
$ns_ at 6.0 "$node_(2) label Recibiendo_paquetes"  
$ns_ at 6.0 "$node_(0) add-mark m0 magenta hexagon"
```

```
$ns_ at 10.2 "$ns_ trace-annotate \"10.2 seg.: Nodo 1 cambia de rumbo\""  
$ns_ at 11.2 "$ns_ trace-annotate \"11.2 seg.: Nodo 0 cambia de rumbo\""  
$ns_ at 12.0 "$ns_ trace-annotate \"12.0 seg.: Nodo 2 cambia de rumbo\""
```

```
$ns_ at 18.2 "$ns_ trace-annotate \"18.2 seg.: Nodo 0 cambia de rumbo\""  
$ns_ at 19.0 "$ns_ trace-annotate \"19.0 seg.: Nodo 1 cambia de rumbo\""  
$ns_ at 21.3 "$ns_ trace-annotate \"21.3 seg.: Nodo 2 cambia de rumbo\""
```

```

$ns_ at 21.4 "$node_(0) label Nodo_0_alejandose"
$ns_ at 23.4 "$node_(0) label pierde_paquetes"
$ns_ at 24.0 "$node_(1) add-mark m1 yellow hexagon"
$ns_ at 24.0 "$ns_trace-annotate \"24.0 seg.: Nodo 1 en forma ADHOC\""
$ns_ at 24.4 "$node_(0) label fuera_de_cobertura"
$ns_ at 24.4 "$node_(2) label Recibiendo_paquetes"
$ns_ at 26.0 "$node_(0) delete-mark m0"
$ns_ at 27.0 "$ns_trace-annotate \"27.0 seg.: Nodo 0 detiene transmision\""

```

```

$ns_ at 27.0 "$node_(1) delete-mark m1"

```

```

$ns_ at 28.5 "$ns_trace-annotate \"28.5 seg.: Nodo 2 cambia de rumbo\""
$ns_ at 29.0 "$ns_trace-annotate \"29.0 seg.: Nodo 1 cambia de rumbo\""
$ns_ at 29.6 "$ns_trace-annotate \"29.6 seg.: Nodo 0 cambia de rumbo\""

```

```

#
# Definiendo termino de la simulacion
#
for {set i 0} {$i < $opt(nn)} {incr i} {
    $ns_ at $opt(stop) "$node_($i) reset"
}
$ns_ at $opt(stop).0001 "stop"
$ns_ at $opt(stop).0002 "puts \"NS EXITING...\"; $ns_halt"

```

```

proc stop {} {
    global ns_tracefd tracenam
    $ns_flush-trace
    close $tracefd
    close $tracenam
    exec nam Adhoc_80211b.nam &
    exit 0
}

```

```

puts "Empezando Simulacion..."
$ns_run

```

## B.2 Archivo Adhoc\_80211g.tcl

```

=====
# Definir Opciones
=====
set opt(chan)      Channel/WirelessChannel ;# Tipo de canal
set opt(prop)     Propagation/TwoRayGround ;# Modelo de propagacion
set opt(netif)    Phy/WirelessPhy ;# Tipo de interface de red
set opt(mac)      Mac/802_11 ;# Tipo de MAC
set opt(ifq)      Queue/DropTail/PriQueue ;# Tipo de cola
set opt(ll)       LL ;# Tipo de capa de enlace
set opt(ant)      Antenna/OmniAntenna ;# Modelo de antena
set opt(ifqlen)   50 ;# Maximo de paquetes en cola
set opt(nn)       5 ;# Numero de nodos moviles
set opt(rp)       AODV ;# Protocolo de enrutamiento

```

```
set opt(x)          90          ;#X dimension de topologia
set opt(y)          90          ;#Y dimension de topologia
set opt(stop)      30.0        ;# Tiempo de simulacion
set opt(packet_size) 1480       ;# Tamano del paquete
```

```
#=====
# Programa Principal
#=====
```

```
#Configuracion para Tarjeta de red Oricono 802.11b -> 22.5 Metros de Rango
#La altura de la antena transmisora y receptora es de 1.5m.
#El modelo de propagacion utilizado es el de 2 Rayos.
```

```
Antenna/OmniAntenna set Gt_ 3.0
Antenna/OmniAntenna set Gr_ 3.0
```

```
Phy/WirelessPhy set CPTresh_ 10.0 ;#capture threshold
Phy/WirelessPhy set CSTresh_ 5.011872e-12 ;#Carrier Sensing threshold
Phy/WirelessPhy set RXThresh_ 5.82587e-09 ;#Receiver signal threshold
Phy/WirelessPhy set bandwidth_ 54Mb ;#Ancho de banda del canal
Phy/WirelessPhy set Pt_ 0.031622777 ;#Potencia de la senial de transmision (Watt)
Phy/WirelessPhy set freq_ 2.437e9 ;#Canal 6 (Hz)
Phy/WirelessPhy set L_ 1.0
```

```
#Definiendo parametros de estandar IEEE 802.11g
```

```
Mac/802_11 set CWMin_ 15
Mac/802_11 set CWMax_ 1023
Mac/802_11 set SlotTime_ 0.000009 ;# 9us
Mac/802_11 set SIFSTime_ 0.000016 ;# 10us
Mac/802_11 set RxTxTurnaroundTime_ 0.000002
Mac/802_11 set MaxPropagationDelay_ 0.0000005
Mac/802_11 set PreambleLength_ 72 ;# 72 bit
#Mac/802_11 set ShortPreambleLength_ 72 ;# 72 bit
Mac/802_11 set PreambleDataRate_ 6.0e6 ;# 1Mbps
Mac/802_11 set PLCPHeaderLength_ 40 ;# 48 bits
Mac/802_11 set PLCPDataRate_ 6.0e6 ;# 6Mbps OFDM
Mac/802_11 set ShortPLCPDataRate_ 2.0e6 ;# 2Mbps
Mac/802_11 set RTSThreshold_ 3000 ;# bytes
Mac/802_11 set ShortRetryLimit_ 7 ;# retransmisiones
Mac/802_11 set LongRetryLimit_ 4 ;# retransmisiones
Mac/802_11 set CCATime_ 0.000003 ;# packet to be correctly received in place
;# of the first sensed packet
Mac/802_11 set dataRate_ 54Mb ;# 802.11 velocidad de transmision de datos
Mac/802_11 set basicRate_ 6Mb ;# 802.11 velocidad basica de transmision
```

```
Agent/TCP set packetSize_ $opt(packet_size)
```

```
# Removiendo cabeceras innecesarias
remove-all-packet-headers ;# removiendo todas excepto las comunes
add-packet-header IP LL Mac ARP TCP ;# cabeceras necesarias
```

```
# Definiendo una instancia del simulador
```

```

#
set ns_          [new Simulator]
set tracefd     [open "Adhoc_80211g.tr" w]
set tracenam    [open Adhoc_80211g.nam w]
$ns_ use-newtrace

$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $tracenam $opt(x) $opt(y)
$ns_ color 0 steelblue

# Definiendo el objeto "Topografia"
set topo       [new Topography]

$topo load_flatgrid $opt(x) $opt(y)

#
# Creando DIOS
#
create-god $opt(nn)

#
# 3 nodos seran creados : node(0), node(1) and node(2)
#

# configuracion del nodo

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

#
# Creando un numero especifico de nodos moviles [$opt(nn)] y aderiendolos al canal
#
for {set i 0} {$i < $opt(nn)} {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0
}

# Definiendo la posicion inicial de los nodos
#
$node_(0) set X_ 5.00

```



```
$node_(0) set Y_ 4.00
$node_(0) set Z_ 0.00
$node_(1) set Z_ 0.00
$node_(1) set Y_ 15.35
$node_(1) set X_ 37.04
$node_(2) set Z_ 0.00
$node_(2) set Y_ 69.37
$node_(2) set X_ 14.25
$node_(3) set Z_ 0.00
$node_(3) set Y_ 42.3
$node_(3) set X_ 44.2
$node_(4) set Z_ 0.00
$node_(4) set Y_ 73.2
$node_(4) set X_ 53.3
```

```
#
```

```
# Definiendo la velocidad y posicion final de los nodos moviles en un tiempo dado
```

```
#
```

```
$ns_ at 3.0 "$node_(2) setdest 54.46 76.51 12.37"
$ns_ at 4.0 "$node_(1) setdest 35.82 35.85 14.91"
$ns_ at 5.0 "$node_(0) setdest 75.66 79.49 19.15"
```

```
$ns_ at 10.2 "$node_(3) setdest 62.78 41.12 17.37"
$ns_ at 11.2 "$node_(4) setdest 36.65 88.54 14.99"
$ns_ at 12.0 "$node_(2) setdest 65.63 71.44 19.38"
```

```
$ns_ at 18.2 "$node_(0) setdest 33.32 16.92 12.37"
$ns_ at 19.0 "$node_(1) setdest 72.78 41.12 19.32"
$ns_ at 21.3 "$node_(2) setdest 81.65 75.38 14.92"
```

```
$ns_ at 28.5 "$node_(3) setdest 86.91 16.31 12.58"
$ns_ at 27.0 "$node_(4) setdest 63.49 75.50 14.25"
$ns_ at 29.6 "$node_(0) setdest 55.13 29.42 19.83"
```

```
# Definiendo la posicion inicial de los nodos en la nam in NAM
```

```
# 7 define el tamaño del nodo en la NAM, se lo debe ajustar acorde al escenario
```

```
# Esta funcion debe ser llamada despues de ser definidad la tipo de movilidad de los nodos
```

```
for {set i 0} {$i < $opt(nn)} {incr i} {
```

```
    $ns_ initial_node_pos $node_($i) 7
}
```

```
# configurando el flujo de trafico entre los nodos
```

```
# Conexion TCP entre los nodos 0 y 2:
```

```

set tcp_(0) [new Agent/TCP]
$tcp_(0) set fid_0
set sink_(0) [new Agent/TCPSink]
$ns_ attach-agent $node_(0) $tcp_(0)
$ns_ attach-agent $node_(2) $sink_(0)
$ns_ connect $tcp_(0) $sink_(0)
set ftp_(0) [new Application/FTP]
$ftp_(0) attach-agent $tcp_(0)
$ns_ at 6.0 "$ftp_(0) start"
$ns_ at 27.0 "$ftp_(0) stop"

set tcp_(0) [new Agent/TCP]
$tcp_(0) set fid_0
set sink_(0) [new Agent/TCPSink]
$ns_ attach-agent $node_(0) $tcp_(0)
$ns_ attach-agent $node_(2) $sink_(0)
$ns_ connect $tcp_(0) $sink_(0)
set ftp_(0) [new Application/FTP]
$ftp_(0) attach-agent $tcp_(0)
$ns_ at 6.0 "$ftp_(0) start"
$ns_ at 27.0 "$ftp_(0) stop"

#
# Anadiendo anotaciones y definiendo colores y etiquetas a los nodos
#
$ns_ at 0.0 "$ns_ trace-annotate \"0.0 seg.: Empieza la simulacion\""
$ns_ at 0.0 "$node_(0) label Nodo_movil_0"
$ns_ at 0.0 "$node_(1) label Nodo_movil_1"
$ns_ at 0.0 "$node_(2) label Nodo_movil_2"
$ns_ at 0.0 "$node_(2) add-mark m2 navy circle"

$ns_ at 3.0 "$ns_ trace-annotate \"3.0 seg.: Nodo 2 comienza a desplazarse\""
$ns_ at 4.0 "$ns_ trace-annotate \"4.0 seg.: Nodo 1 comienza a desplazarse\""
$ns_ at 5.0 "$ns_ trace-annotate \"5.0 seg.: Nodo 0 comienza a desplazarse\""
$ns_ at 6.0 "$ns_ trace-annotate \"6.0 seg.: Nodo 0 inicia transmision\""
$ns_ at 6.0 "$node_(2) label Recibiendo_paquetes"
$ns_ at 6.0 "$node_(0) add-mark m0 magenta hexagon"

$ns_ at 10.2 "$ns_ trace-annotate \"10.2 seg.: Nodo 1 cambia de rumbo\""
$ns_ at 11.2 "$ns_ trace-annotate \"11.2 seg.: Nodo 0 cambia de rumbo\""
$ns_ at 12.0 "$ns_ trace-annotate \"12.0 seg.: Nodo 2 cambia de rumbo\""

$ns_ at 18.2 "$ns_ trace-annotate \"18.2 seg.: Nodo 0 cambia de rumbo\""
$ns_ at 19.0 "$ns_ trace-annotate \"19.0 seg.: Nodo 1 cambia de rumbo\""
$ns_ at 21.3 "$ns_ trace-annotate \"21.3 seg.: Nodo 2 cambia de rumbo\""

$ns_ at 21.4 "$node_(0) label Nodo_0_alejandose"
$ns_ at 23.4 "$node_(0) label pierde_paquetes"
$ns_ at 24.0 "$node_(1) add-mark m1 yellow hexagon"
$ns_ at 24.0 "$ns_ trace-annotate \"24.0 seg.: Nodo 1 en forma ADHOC\""
$ns_ at 24.4 "$node_(0) label fuera_de_cobertura"
$ns_ at 24.4 "$node_(2) label Recibiendo_paquetes"

```

```

$ns_ at 26.0 "$node_(0) delete-mark m0"
$ns_ at 27.0 "$ns_ trace-annotate \"27.0 seg.: Nodo 0 detiene transmision\"""

$ns_ at 27.0 "$node_(1) delete-mark m1"

$ns_ at 28.5 "$ns_ trace-annotate \"28.5 seg.: Nodo 2 cambia de rumbo\"""
$ns_ at 29.0 "$ns_ trace-annotate \"29.0 seg.: Nodo 1 cambia de rumbo\"""
$ns_ at 29.6 "$ns_ trace-annotate \"29.6 seg.: Nodo 0 cambia de rumbo\"""

#
# Definiendo termino de la simulacion
#
for {set i 0} {$i < $opt(nn)} {incr i} {
    $ns_ at $opt(stop) "$node_($i) reset"
}
$ns_ at $opt(stop).0001 "stop"
$ns_ at $opt(stop).0002 "puts \"NS EXITING...\"; $ns_ halt"

proc stop {} {
    global ns_ tracefd tracenam
    $ns_ flush-trace
    close $tracefd
    close $tracenam
    exec nam Adhoc_80211g.nam &
    exit 0
}

```

```

puts "Empezando Simulacion..."
$ns_ run

```

### B.3 Archivo Infra\_80211b.tcl

```

#=====
# Definir Opciones
#=====
set opt(chan)      Channel/WirelessChannel    ;# Tipo de canal
set opt(prop)      Propagation/TwoRayGround   ;# Modelo de propagacion
set opt(netif)      Phy/WirelessPhy          ;# Tipo de interface de red
set opt(mac)        Mac/802_11                ;# Tipo de MAC
set opt(ifq)        Queue/DropTail/PriQueue   ;# Tipo de cola
set opt(ll)         LL                        ;# Tipo de capa de enlace
set opt(ant)        Antenna/OmniAntenna       ;# Modelo de antena
set opt(ifqlen)     50                        ;# Maximo de paquetes en cola
set opt(nn)         1                         ;# Numero de nodos moviles
set opt(rp)         DSDV                      ;# Protocolo de enrutamiento
set opt(x)          90                        ;# X dimension de topologia
set opt(y)          90                        ;# Y dimension de topologia
set opt(packet_size) 1480                    ;# Tamano del paquete

#=====
# Programa Principal
#=====

```

*#Pavon and Choi, "Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement"*

*#La altura de la antena transmisora y receptora es de 1.5m.  
#El modelo de propagacion utilizado es el de 2 Rayos.*

*Antenna/OmniAntenna set Gt\_ 3.0  
Antenna/OmniAntenna set Gr\_ 3.0*

*Phy/WirelessPhy set L\_ 1.0  
Phy/WirelessPhy set freq\_ 2.437e9 ;# canal 6  
Phy/WirelessPhy set bandwidth\_ 11Mb  
Phy/WirelessPhy set Pt\_ 0.1  
Phy/WirelessPhy set CPThresh\_ 10.0  
Phy/WirelessPhy set CSThresh\_ 5.011872e-12  
Phy/WirelessPhy set RXThresh\_ 5.82587e-09*

*#Definiendo parametros de estandar IEEE 802.11b*

*Mac/802\_11 set CWMin\_ 31  
Mac/802\_11 set CWMax\_ 1023  
Mac/802\_11 set SlotTime\_ 0.000020 ;# 20us  
Mac/802\_11 set SIFSTime\_ 0.000010 ;# 10us  
Mac/802\_11 set PreambleLength\_ 144 ;# 144 bit  
Mac/802\_11 set ShortPreambleLength\_ 72 ;# 72 bit  
Mac/802\_11 set PreambleDataRate\_ 1.0e6 ;# 1Mbps  
Mac/802\_11 set PLCPHeaderLength\_ 48 ;# 48 bits  
Mac/802\_11 set PLCPDataRate\_ 1.0e6 ;# 1Mbps  
Mac/802\_11 set ShortPLCPDataRate\_ 2.0e6 ;# 2Mbps  
Mac/802\_11 set RTSThreshold\_ 3000 ;# bytes  
Mac/802\_11 set ShortRetryLimit\_ 7 ;# retransmisiones  
Mac/802\_11 set LongRetryLimit\_ 4 ;# retransmisiones*

*Mac/802\_11 set dataRate\_ 11Mb ;# 802.11 velocidad de transmision de datos  
Mac/802\_11 set basicRate\_ 1Mb ;# 802.11 velocidad basica de transmision*

*Agent/TCP set packetSize\_ \$opt(packet\_size)*

*# remove useless headers  
remove-all-packet-headers ;# removes all except common  
add-packet-header IP LL Mac ARP TCP ;# needed headers*

*# Definiendo una instancia del simulador*

*#  
set ns\_ [new Simulator]  
set tracefd [open "Infra\_80211b.tr" w]  
set tracenam [open "Infra\_80211b.nam" w]  
\$ns\_ use-newtrace*

*\$ns\_ trace-all \$tracefd  
\$ns\_ namtrace-all-wireless \$tracenam \$opt(x) \$opt(y)*

*\$ns\_color 0 red*

*set topo [new Topography]  
\$topo load\_flatgrid \$opt(x) \$opt(y)*

*# Definiendo enrutamiento jerarquico  
\$ns\_node-config -addressType hierarchical  
AddrParams set domain\_num\_2  
lappend cluster\_num 1 1  
AddrParams set cluster\_num\_ \$cluster\_num  
lappend eilastlevel 1 2  
AddrParams set nodes\_num\_ \$eilastlevel*

*#GOD necesita saber el numero de todas las interfaces inalambricas  
create-god [expr \$opt(nn) + 1]*

*# creando nodo alambrado  
set W(0) [\$ns\_node 0.0.0]*

*\$W(0) set X\_ 1.0  
\$W(0) set Y\_ 1.0  
\$W(0) set Z\_ 0.0*

*\$W(0) color "magenta"*

*# configuracion de nodos inalambricos*

*\$ns\_node-config -adhocRouting \$opt(rp) \  
-llType \$opt(ll) \  
-macType \$opt(mac) \  
-ifqType \$opt(ifq) \  
-ifqLen \$opt(ifqlen) \  
-antType \$opt(ant) \  
-propInstance [new \$opt(prop)] \  
-phyType \$opt(netif) \  
-channel [new \$opt(chan)] \  
-topoInstance \$topo \  
-wiredRouting ON \  
-agentTrace ON \  
-routerTrace ON \  
-macTrace OFF*

*# Creando la Estacion Base*

*set BS(0) [\$ns\_node 1.0.0]  
\$BS(0) random-motion 0*

*\$BS(0) set X\_ 1.0  
\$BS(0) set Y\_ 1.0*



```
$BS(0) set Z_ 0.0
```

```
$BS(0) shape "hexagon"  
$BS(0) color "green"  
$BS(0) label "Base Station"
```

```
# Creando Nodos Moviles  
$ns_ node-config -wiredRouting OFF
```

```
set temp {1.0.1}  
set node_(0) [ $ns_ node [lindex $temp 0] ]  
  $node_(0) base-station [AddrParams addr2id [$BS(0) node-addr]]  
  $node_(0) random-motion 0
```

```
$node_(0) set X_ 70.5  
$node_(0) set Y_ 20.0  
$node_(0) set Z_ 0.0
```

```
$ns_ at 5.0 "$node_(0) setdest 19.0 70.0 15.0"  
$ns_ at 10.0 "$node_(0) setdest 50.0 15.0 15.0"  
$ns_ at 13.0 "$node_(0) setdest 80.0 55.0 50.0"  
$ns_ at 15.0 "$node_(0) setdest 89.9 89.9.0 15.0"  
$ns_ at 20.0 "$node_(0) setdest 79.0 70.0 50.0"  
$ns_ at 21.5 "$node_(0) setdest 60.0 25.0 50.0"
```

```
#Creando enlace entre la Estacion Base y el nodo alambrado  
$ns_ duplex-link $W(0) $BS(0) 100Mb 2ms DropTail
```

```
$ns_ duplex-link-op $W(0) $BS(0) orient right  
$ns_ duplex-link-op $W(0) $BS(0) color "blue"
```

```
# Configurando una conexion TCP entre el nodo alambrado y un nodo inalámbrico  
set tcp [new Agent/TCP]
```

```
$tcp set fid_ 0  
set sink [new Agent/TCPSink]  
$ns_ attach-agent $W(0) $tcp  
$ns_ attach-agent $node_(0) $sink  
$ns_ connect $tcp $sink  
set ftp [new Application/FTP]  
$ftp attach-agent $tcp  
$ns_ at 1.0 "$ftp start"  
$ns_ at 23.0 "$ftp stop"
```

```
# Definiendo la posicion inicial de los nodos en la nam in NAM
```

```
# 7 define el tamaño del nodo en la NAM, se lo debe ajustar acorde al escenario  
# Esta función debe ser llamada después de ser definida la tipo de movilidad de los nodos
```

```
$ns_ initial_node_pos $node_(0) 7
```

```
#  
# Anadiendo anotaciones y definiendo colores y etiquetas a los nodos  
#
```

```
$ns_ at 0.0 "$ns_ trace-annotate \"0.0 seg.: Comienza la Simulacion\""  
$ns_ at 1.0 "$ns_ trace-annotate \"1.0 seg.: Start FTP1 \""  
$ns_ at 1.0 "$ns_ trace-annotate \"1.0 seg.: from N0 to N2\""  
$ns_ at 5.0 "$ns_ trace-annotate \"5.0 seg.: N2 begins to move\""  
$ns_ at 10.0 "$ns_ trace-annotate \"10.0 seg.: N2 begins to move\""  
$ns_ at 13.0 "$ns_ trace-annotate \"13.0 seg.: N2 begins to move\""  
$ns_ at 15.0 "$ns_ trace-annotate \"15.0 seg.: N2 begins to move\""  
$ns_ at 20.0 "$ns_ trace-annotate \"20.0 seg.: N2 begins to move\""  
$ns_ at 22.0 "$ns_ trace-annotate \"22.0 seg.: N2 begins to move\""  
$ns_ at 22.0 "$ns_ trace-annotate \"22.0 seg.: FTP1 finish transmissions \""  
$ns_ at 1.0 "$node_(0) add-mark m1 gold circle"  
$ns_ at 22.0 "$node_(0) delete-mark m1"  
$ns_ at 0.0 "$node_(0) label Nodo_movil"  
$ns_ at 0.0 "$W(0) label Nodo_fijo"  
$ns_ at 1.0 "$node_(0) label Recibiendo_Paquetes"
```

```
#  
# Definiendo termino de la simulacion  
#
```

```
$ns_ at 24.0 "$node_(0) reset"
```

```
$ns_ at 24.0000 "$BS(0) reset";  
$ns_ at 24.0001 "stop"  
$ns_ at 24.0002 "puts \"NS EXITING...\"; $ns_ halt"
```

```
proc stop {} {  
    global ns_ tracefd tracenam  
    $ns_ flush-trace  
    close $tracefd  
    close $tracenam  
    exec nam Infra_80211b.nam &  
    exit 0  
}
```

```
puts "Empezando Simulacion..."  
$ns_ run
```

## B.4 Archivo Infra\_IP.tcl

```
#####  
# Definir opciones  
#####  
set opt(chan) Channel/WirelessChannel ;# Tipo de canal
```

```

set opt(prop) Propagation/TwoRayGround ;# Modelo de radio-propagacion
set opt(netif) Phy/WirelessPhy ;# Tipo de interfaz de red
set opt(mac) Mac/802_11 ;# Tipo de capa MAC
set opt(ifq) Queue/DropTail/PriQueue ;# Tipo de cola
set opt(ll) LL ;# Tipo de capa de enlace
set opt(ant) Antenna/OmniAntenna ;# modelo de antena
set opt(ifqlen) 50 ;# Maximo de paquetes en cola
set opt(nn) 1 ;# numero de nodos moviles
set opt(adhocRouting) DSDV ;# protocolo de enrutamiento
set opt(x) 190 ;# coordenada x de la topologia
set opt(y) 150 ;# coordenada y de la topologia
set opt(seed) 0.0 ;# seed aleatorio
set num_wired_nodes 1 ;# numero de estaciones alamblicas
set opt(packet_size) 1480 ;# Tamaño del paquete

```

```

#=====

```

```

# Chequear los limites de la topologia y el seed aleatorio
if { $opt(x) == 0 || $opt(y) == 0 } {
    puts "No X-Y boundary values given for wireless topology\n"
}
if { $opt(seed) > 0 } {
    puts "Seeding Random number generator with $opt(seed)\n"
    ns-random $opt(seed)
}

```

```

# Crear instancia de simulador
set ns_ [new Simulator]

```

```

# Configurar enrutamiento jerarquico
$ns_ node-config -addressType hierarchical

```

```

AddrParams set domain_num_3 ;# numero de dominios
lappend cluster_num 1 1 1 ;# numero de cluster en cada dominio
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel 1 2 1 ;# numero de nodos en cada cluster
AddrParams set nodes_num_ $eilastlevel ;# de cada dominio

```

```

#La altura de la antena es 1.5m para el transmisor y receptor.
#El modelo de propagacion es TwoRayGround.

```

```

Antenna/OmniAntenna set Gt_ 3.0
Antenna/OmniAntenna set Gr_ 3.0

```

```

Phy/WirelessPhy set CPThresh_ 10.0 ;#capture threshold
Phy/WirelessPhy set CSThresh_ 5.011872e-12 ;#Carrier Sensing threshold
Phy/WirelessPhy set RXThresh_ 5.82587e-09 ;#Receiver signal threshold
Phy/WirelessPhy set bandwidth_ 54Mb ;#Ancho de banda del canal
Phy/WirelessPhy set Pt_ 0.031622777 ;#Potencia del transmisor (Watt)
Phy/WirelessPhy set freq_ 2.437e9 ;# Frecuencia de canal 6 (Hz)
Phy/WirelessPhy set L_ 1.0

```

```

Mac/802_11 set CWMin_          15
Mac/802_11 set CWMax_          1023
Mac/802_11 set SlotTime_       0.000009 ;# 9us
Mac/802_11 set SIFSTime_       0.000016 ;# 10us
Mac/802_11 set RxTxTurnaroundTime_ 0.000002
Mac/802_11 set MaxPropagationDelay_ 0.0000005
Mac/802_11 set PreambleLength_ 72 ;# 72 bit
Mac/802_11 set PreambleDataRate_ 6.0e6 ;# 1Mbps
Mac/802_11 set PLCPHeaderLength_ 40 ;# 48 bits
Mac/802_11 set PLCPDataRate_ 6.0e6 ;# 6Mbps OFDM
Mac/802_11 set ShortPLCPDataRate_ 2.0e6 ;# 2Mbps
Mac/802_11 set RTSThreshold_ 3000 ;# bytes
Mac/802_11 set ShortRetryLimit_ 7 ;# retransmisiones
Mac/802_11 set LongRetryLimit_ 4 ;# retransmisiones
Mac/802_11 set CCATime_        0.000003 ;# paquete para ser correctamente recibido en
;# lugar del primer paquete detectado
Mac/802_11 set dataRate_ 54Mb ;# Tasa de transmision de datos en 802.11g
Mac/802_11 set basicRate_ 6Mb ;# Tasa de transmision basica en 802.11g

```

```

Agent/TCP set packetSize_ $opt(packet_size)

```

```

# quitar cabeceras inservibles

```

```

remove-all-packet-headers ;# quita todo excepto
add-packet-header IP LL Mac ARP TCP ;# las cabeceras necesarias comunes

```

```

#

```

```

#Crear el archivo trace y el archivo nam

```

```

#

```

```

set tracefd [open Infra_IP.tr w]
set namtrace [open Infra_IP.nam w]
$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $opt(x) $opt(y)
$ns use-newtrace

```

```

# Crear el objeto de topografia

```

```

set topo [new Topography]

```

```

# Definir la topologia

```

```

$topo load_flatgrid $opt(x) $opt(y)

```

```

# Crear God

```

```

# 2 para HA y FA

```

```

create-god [expr $opt(nm) + 2]

```

```

#Crear nodos alambricos

```

```

set temp {0.0.0} ;# Direcciones jerarquicas para el dominio alambrico

```

```

set W(0) [$ns_node [lindex $temp 0]]

```

```

$W(0) color "orange"

```

```

$W(0) set X_ 95.67

```

```

$W(0) set Y_ 100.01

```

```
$W(0) set Z_ 0.00
```

```
$ns_ color 0 steelblue
```

```
# Configurar los nodos ForeignAgent y HomeAgent
```

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

```
# Crear HA y FA
```

```
set HA [$ns_ node 1.0.0]
```

```
set FA [$ns_ node 2.0.0]
```

```
$HA random-motion 0
```

```
$FA random-motion 0
```

```
# Posiciones iniciales para las estaciones base (HA y FA).
```

```
$HA set X_ 2.29
```

```
$HA set Y_ 70.48
```

```
$HA set Z_ 0.00
```

```
$HA shape "hexagon"
```

```
$HA color "gold"
```

```
$HA label "Home Agent";#label del nodo
```

```
$FA set X_ 175.88
```

```
$FA set Y_ 70.75
```

```
$FA set Z_ 0.000
```

```
$FA shape "hexagon"
```

```
$FA color "darkgreen"
```

```
$FA label "Foreign Agent"
```

```
# Crear un nodo movil q se esta movilizandoo entre HA y FA.
```

```
# Notese q la direccion de MH indica que se encuentra en el mismo dominio que HA.
```

```
$ns_ node-config -wiredRouting OFF
```

```
set MH [$ns_ node 1.0.1]
```

```
set node_(0) $MH
```

```
set HAaddress [AddrParams addr2id [$HA node-addr]]
```

```
[$MH set regagent_] set home_agent_ $HAaddress
```



*# Posicion inicial del MH*

*\$MH set Z\_ 0.00*  
*\$MH set Y\_ 20.00*  
*\$MH set X\_ 2.00*

*# Configuracion del movimiento del MH*

*\$ns\_ at 2.0 "\$MH setdest 185.00 20.00 20.00"*  
*\$ns\_ at 20.0 "\$MH setdest 3.00 20.00 20.00"*  
*\$ns\_ at 36.0 "\$MH setdest 185.00 20.00 20.00"*  
*\$ns\_ at 47.0 "\$MH setdest 3.00 20.00 20.00"*

*#crear enlaces entre el nodo alambriico y las estaciones base*

*\$ns\_ duplex-link \$W(0) \$HA 100Mb 2ms DropTail*  
*\$ns\_ duplex-link \$W(0) \$FA 100Mb 2ms DropTail*

*\$ns\_ duplex-link-op \$W(0) \$HA orient left-down*  
*\$ns\_ duplex-link-op \$W(0) \$FA orient right-down*  
*\$ns\_ duplex-link-op \$W(0) \$FA label "Enlace fijo FA";#label*  
*\$ns\_ duplex-link-op \$W(0) \$HA label "Enlace fijo HA"*  
*\$ns\_ duplex-link-op \$W(0) \$FA color "blue"*  
*\$ns\_ duplex-link-op \$W(0) \$HA color "blue"*

*# Configurar las conecciones TCP entre el nodo alambriico y el MH*

*set tcp1 [new Agent/TCP]*  
*\$tcp1 set fid\_ 0*  
*set sink1 [new Agent/TCPSink]*  
*\$ns\_ attach-agent \$MH \$tcp1*  
*\$ns\_ attach-agent \$W(0) \$sink1*  
*\$ns\_ connect \$tcp1 \$sink1*  
*set ftp1 [new Application/FTP]*  
*\$ftp1 attach-agent \$tcp1*  
*\$ns\_ at 1.0 "\$ftp1 start"*  
*\$ns\_ at 31.0 "\$ftp1 stop"*

*set tcp2 [new Agent/TCP]*  
*\$tcp2 set fid\_ 1*  
*set sink2 [new Agent/TCPSink]*  
*\$ns\_ attach-agent \$W(0) \$tcp2*  
*\$ns\_ attach-agent \$MH \$sink2*  
*\$ns\_ connect \$tcp2 \$sink2*  
*set ftp2 [new Application/FTP]*  
*\$ftp2 attach-agent \$tcp2*  
*\$ns\_ at 33.0 "\$ftp2 start"*  
*\$ns\_ at 57.0 "\$ftp2 stop"*

*\$ns\_ color 2 magenta*

*# Define el tamaño del MH en la nam*

```

for {set i 0} {$i < $opt(nn)} {incr i} {

    # 7 define el tama±o del nodo en la nam, debe ajustare al escenario
    #
    #

    $ns_ initial_node_pos $node_($i) 7
}

$ns_ at 0.0 "$ns_ trace-annotate \"0.0 seg.: Comienza la Simulacion\""
$ns_ at 1.0 "$ns_ trace-annotate \"1 seg.: MH empieza a transmitir a nodo fijo\""
$ns_ at 2.0 "$ns_ trace-annotate \"2 seg.: Movimiento de MH en direccion a FA\""
$ns_ at 20.0 "$ns_ trace-annotate \"20 seg.: Movimiento de MH en direccion a HA\""
$ns_ at 32.0 "$ns_ trace-annotate \"32 seg.: Se detiene transmision\""
$ns_ at 33.0 "$ns_ trace-annotate \"33 seg.: Nodo fijo empieza a trasmitir a MH\""
$ns_ at 36.0 "$ns_ trace-annotate \"36 seg.: Movimiento de MH en direccion a FA\""
$ns_ at 47.0 "$ns_ trace-annotate \"47 seg.: Movimiento de MH en direccion a HA\""
$ns_ at 57.0 "$ns_ trace-annotate \"57 seg.: Se detiene transmision \""

$ns_ at 1.0 "$MH add-mark m1 green circle"
$ns_ at 32.0 "$MH delete-mark m1"
$ns_ at 0.0 "$MH label Nodo_movil"
$ns_ at 2.0 "$MH label Enviando_Paquetes"
$ns_ at 33.0 "$MH add-mark m2 red circle"
$ns_ at 57.0 "$MH delete-mark m2"
$ns_ at 34.0 "$MH label Recibiendo_Paquetes"

# Decirle a todos los nodos en que momento termna la simulacion
for {set i 0} {$i < $opt(nn)} {incr i} {
    $ns_ at 58.0 "$node_($i) reset";
}
$ns_ at 59.0 "$HA reset";
$ns_ at 59.0 "$FA reset";

$ns_ at 59.0002 "puts \"Saliendo de NS..\""; $ns_ halt"
$ns_ at 59.0001 "stop"

proc stop {} {
    global ns_ tracefd namtrace
    close $tracefd
    close $namtrace
    puts "Ejecutando nam..."
    exec nam Infra_IP.nam &
    exit 0
}

puts "Empezando simulacion..."
$ns_ run

```

# APÉNDICE C

## Manual de las Prácticas

Este apéndice de manual de las prácticas tienen el propósito de guiar al estudiante al desarrollo de las simulaciones de redes IEEE 802.11b e IEEE 802.11g. Antes de empezar a trabajar con estas prácticas el estudiante debe estar familiarizado con los tutoriales de NS2, descritos en el apéndice A.

### C.1 Practica N.- 1 “Introducción al simulador NS2”

#### Práctica N.- 1

### Introducción al simulador NS2

#### 1. OBJETIVOS

- Orientar a los estudiantes sobre una herramienta didáctica para la simulación de redes de comunicación.
- Introducir a los estudiantes en un entorno de simulación utilizando el NS2.
- Realizar un script para la implementación de una simulación de una red sencilla.

#### 2. INTRODUCCION

NS (Network Simulator) es un simulador de eventos centrado en la investigación sobre redes. NS dispone de simulación para TCP, routing y multicast sobre redes cableadas o inalámbricas (locales y por satélite).

El simulador permite la configuración de gran cantidad de parámetros tales como la topología de la red, la pila de protocolos y parámetros específicos de cada protocolo. También permite evaluar el impacto de diferentes tipos de tráfico de red.

NS actualmente en la versión 2 nos permite realizar simulaciones de múltiples tipos de redes (cableadas, inalámbricas y por satélite). Para ello utiliza un lenguaje de script llamado Tcl que nos permite ir generando el modelo.

También disponemos de una interfaz gráfica llamada NAM que nos permite

visualizar las simulaciones e incluso crear y editar los modelos a simular. NS es una gran herramienta que nos puede ayudar en muchos campos a la hora de realizar pruebas o generar nuevos tipos de redes

### **3. MARCO TEORICO**

NS empieza como una variante del REAL Network Simulator en 1989 y ha evolucionado substancialmente durante los últimos años. En 1995 el desarrollo lo llevaba acabo DARPA a través del proyecto VINT de LBL, Xerox PARC, UCB y USC/ISI. Actualmente el desarrollo de NS lo lleva DARPA junto a SAMAN y otros. NS siempre ha contado con contribuciones de muchos otros desarrolladores, incluyendo código inalámbrico de los proyectos CMU Monach y UCB Daedelus y también de Sun Microsystems.

NS se está utilizando tanto en entornos de investigación como en entornos educativos. NS se utiliza en entornos educativos ya que nos permite simular sencillas redes que nos van a ayudar a comprender los distintos protocolos y observar como se produce el envío de paquetes entre nodos, etc.

En NS2 se utilizan dos lenguajes C++ y Otcl. El núcleo del simulador se encuentra escrito en C++ y se encuentra disponible dicho código fuente para el estudio y modificación de los protocolos. Otcl es utilizado para definir el entorno de simulación, desde los nodos que componen la simulación, hasta las características del tráfico que se intercambia entre dichos nodos. Una vez simulado el entorno se generan archivos de trazas que podremos evaluar en modo texto con lenguajes como Awk, o gráficamente con la herramienta NAM.

El simulador NS, es un simulador extremadamente potente, diseñado por y para investigadores. A pesar de esto, y debido a que es altamente configurable, resulta también indicado para ejecutar simulaciones de redes concretas y como herramienta educativa para el estudio de protocolos.

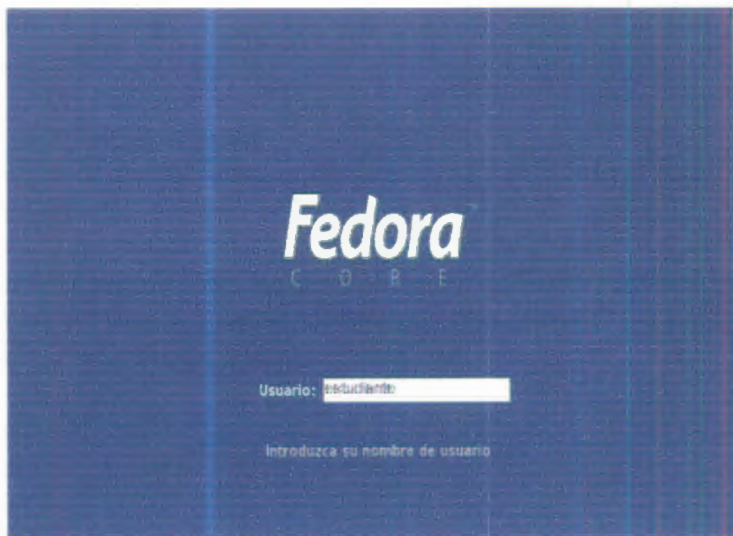
### **4. MATERIALES Y EQUIPOS**

PC con Fedora Core 4 y Network Simulator 2: Network Animador NAM.

### **5. PROCEDIMIENTOS**

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”.

Los pasos a seguir a la hora de utilizar el NS2 son:

- Describir la simulación mediante el lenguaje Otcl. Mediante este lenguaje describimos el entorno de simulación, las características de los nodos que componen dicha simulación, los parámetros de comunicación entre estos nodos, etc. Podemos encontrar un tutorial del lenguaje Otcl en <http://www.isi.edu/nsnam/ns/tutorial/index.html>.
- Una vez descrito el entorno de simulación ejecutamos el simulador:

*ns script.tcl*

- Por último analizamos los resultados que nos ofrece el simulador. El simulador nos ofrece dos archivos de trazas. El primer archivo tiene extensión .nam y es utilizado para la visualización gráfica de la simulación. El otro archivo de resultados tiene extensión .tr y es donde se almacena toda la información correspondiente a los eventos que han sucedido durante la simulación. Para analizarlos podemos usar cualquier herramienta que permita el tratamiento de archivos de texto (p.e. Perl, awk).

### **Paso 1: Descripción de los comandos básicos para la implantación de una simulación**

En este apartado vamos a analizar línea por línea la simulación que crearemos, para ello abrimos un Editor de Texto que se encuentra en *Aplicaciones -> Accesorios->*



*Editor de Texto.* Crearemos una red cableada de 4 host, los cuales en NS2 son llamados nodos.

Escribimos la siguiente sentencia:

```
set ns [new Simulator]
```

Con esta sentencia creamos una variable llamada "ns" que contiene una instancia del objeto Simulator, y así poder crear un entorno para simular. A partir de este objeto creamos el resto de objetos de nuestra simulación.

```
$ns color 0 blue  
$ns color 1 red  
$ns color 2 white
```

Asignamos colores para la visualización gráfica en NAM.

```
set n0 [$ns node]  
set n1 [$ns node]  
set n2 [$ns node]  
set n3 [$ns node]
```

Creamos 4 variables (ns0 a ns3) que representan a cuatro nodos dentro de nuestro entorno de simulación. Podemos observar que estos cuatro objetos node se crean a partir del objeto Simulator.

```
set f [open out.tr w]  
$ns trace-all $f  
set nf [open out.nam w]  
$ns namtrace-all $nf
```

Aquí se declaran los descriptores para la creación de los archivos de salida out.tr (archivo de traza) y out.nam (fichero de animación utilizado por NAM).

```
$ns duplex-link $n0 $n2 5Mb 2ms DropTail  
$ns duplex-link $n1 $n2 5Mb 2ms DropTail  
$ns duplex-link $n2 $n3 1.5Mb 10ms DropTail
```

Definimos los enlaces entre las estaciones. Por cada enlace se indica características tales como el tipo (duplex) el ancho de banda (5 Mb/s o 1.5 Mb/s) el retardo de propagación (2ms o 10 ms) y la política de servicio de las colas (DropTail corresponde a una política FIFO).

```
$ns duplex-link-op $n0 $n2 orient right-up
```

```
$ns duplex-link-op $n1 $n2 orient right-down  
$ns duplex-link-op $n2 $n3 orient right  
$ns duplex-link-op $n2 $n3 queuePos 0.5
```

Comandos utilizados por el visualizador NAM para la orientación de los nodos en nuestro entorno de simulación.

```
set udp0 [new Agent/UDP]  
$ns attach-agent $n0 $udp0  
set cbr0 [new Application/Traffic/CBR]  
$cbr0 attach-agent $udp0
```

Creamos un agente de transporte UDP en el nodo n0. Este agente generará un tráfico CBR, que puede ser de tipo stream multimedia.

```
set udp1 [new Agent/UDP]  
$ns attach-agent $n3 $udp1  
$udp1 set class_ 1  
set cbr1 [new Application/Traffic/CBR]  
$cbr1 attach-agent $udp1
```

Creamos otro agente generador de tráfico UDP en el nodo n3. La clase de servicio de este tráfico será de una clase distinta al creado en el nodo n0.

```
set null0 [new Agent/Null]  
$ns attach-agent $n3 $null0  
set null1 [new Agent/Null]  
$ns attach-agent $n1 $null1
```

La estación n3 tendrá un agente null, que representa un consumidor de tráfico. Este nodo será el receptor del tráfico generado en n0. En el nodo n1 tendremos otro consumidor de tráfico.

```
$ns connect $udp0 $null0  
$ns connect $udp1 $null1
```

Conectamos el agente generador de tráfico udp0 que se encuentra en el nodo n0 al consumidor de tráfico null0 que se localiza en el nodo n3. Además conectamos en nodo n3 con el nodo n1.

```
$ns at 1.0 "$cbr0 start"  
$ns at 1.1 "$cbr1 start"
```

En los instantes 1.0 y 1.1 comienzan los tráficos CBR que hemos definido en los nodos n0 y n3.

```
set tcp [new Agent/TCP]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $n0 $tcp
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 1.2 "$ftp start"
```

Declaramos un agente del tipo TCP entre las estaciones n0 y n3. A continuación se asocia un tráfico FTP (file transfer protocol) sobre dicho agente. En el instante 1.2 se comienza a transferir un paquete de tamaño infinito.

```
$ns at 1.35 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"
```

En el instante 1.35 se para el tráfico FTP.

```
puts "El tamaño del paquetes es [$cbr0 set packetSize _]"
puts "El valor del intervalo es [$cbr0 set interval _]"
```

Mostramos por pantalla el tamaño del paquete CBR y el intervalo entre dos paquetes consecutivos.

```
$ns at 3.0 "finish"
```

La simulación finaliza en el instante 3.0

```
proc finish {} {
    global ns fnf
    $ns_ flush-trace
    close $f
    close $nf
    puts "running nam..."
    exec nam out.nam &
    exit 0
}
```

Procedimiento que cierra los archivos donde almacenamos los resultados. Además lanzamos de forma automática el visualizador NAM para estudiar gráficamente los resultados de nuestra simulación.

*\$ns run*

Indicamos al objeto simulador que ejecute la simulación.

Como se puede observar en este archivo de traza hemos definido todas las características necesarias para llevar a cabo la simulación. Luego guardamos este script con el nombre *ejemplo1.tcl* en el siguiente directorio */Home/Practical*

## **Paso 2: Compilación de un Script**

Luego abrimos una ventana de consola y para ellos nos vamos a *Aplicaciones-> Accesorios-> Terminal* o en su defecto de clic derecho en el escritorio y escogemos *Abrir Terminal* y aparecerá una pantalla de esta forma.



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

```
cd Practical  
ns ejemplo1.tcl
```

## **Paso 3: Herramienta Network Animator, NAM**

La herramienta NAM es una herramienta de animación desarrollada en Tcl/TK que utilizando como entrada las trazas generadas automáticamente por NS permite realizar una animación de la simulación realizada con ns. Para invocar NAM se utiliza la siguiente orden:

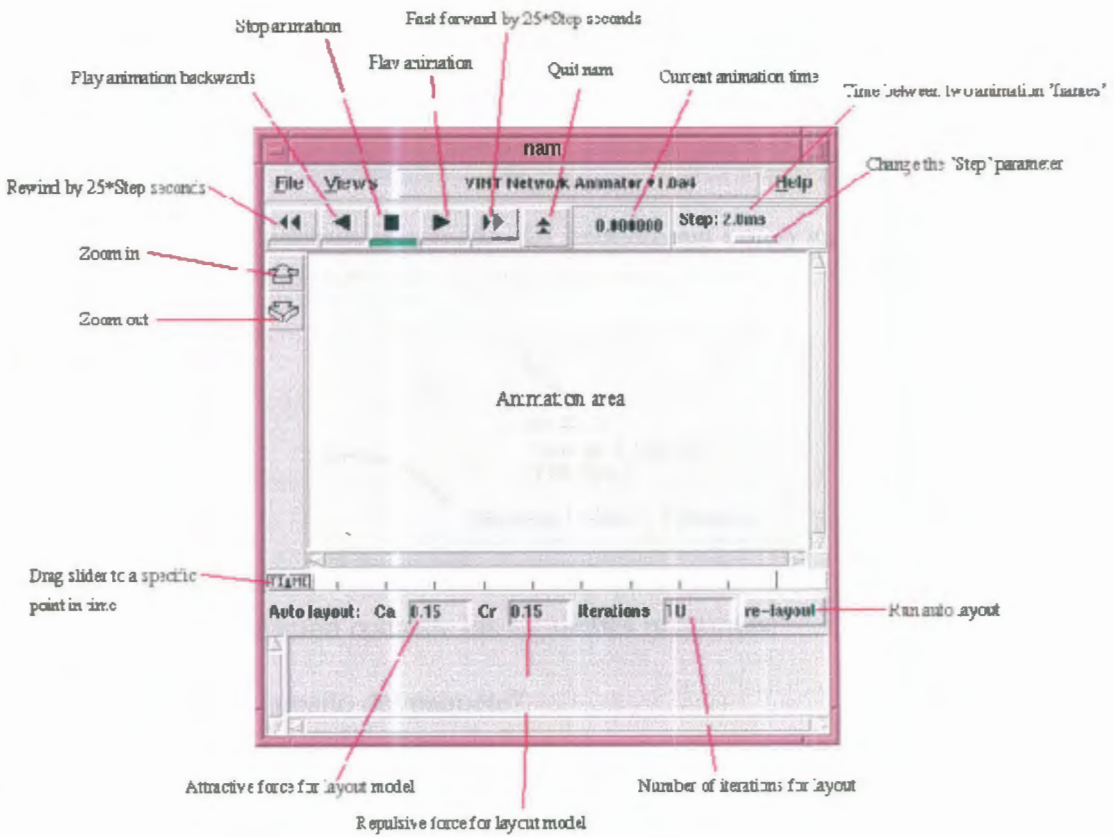
```
nam archivo_traza.nam
```

Cuando se abre un archivo de traza con nam, se crea una ventana con la topología indicada en la simulación. La Figura 1 muestra el aspecto general de la herramienta NAM. Se han remarcado los comandos y zonas más importantes de la herramienta.

Las principales áreas y funciones de la herramienta nam son las siguientes:



- **Animation Area:** En esta zona de la ventana se visualiza el escenario y la animación del mismo.
- **Zoom In y Zoom Out:** Estos dos botones permiten acercar o alejar el escenario y poder abarcar un mayor rango de visión de la simulación.
- **Stop/Play Animation:** Como su nombre indica sirven para iniciar y detener la animación.
- **Current Animation Time:** Indica el tiempo que llevamos de simulación. Su valor llegará hasta el tiempo que le hubiéramos indicado en la simulación.
- **Step:** Este valor nos indica lo rápido que evolucionará la simulación (en milisegundos). Su valor se puede modificar con el deslizador que hay debajo del marcador.
- **Menú:** Agrupa varias opciones como por ejemplo grabar la animación, imprimir el área de animación, ver la energía de los nodos, filtrar el tipo de paquetes a visualizar (Datos, Mac, Routing), etc.



### *Estructura de la NAM*

Para aumentar o disminuir la velocidad de la simulación Ud. puede manipular la escala de tiempo desplazándola de izquierda a derecha según criterio.

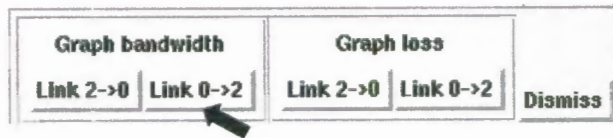




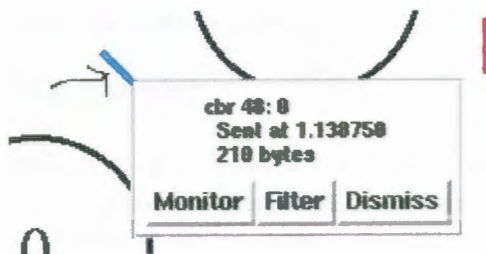
En la parte inferior de la presentación de la NAM se encuentran las leyendas que indican los momentos en que se suscita algún evento en el transcurso del tiempo. Si Ud se perdió algún evento puede dar doble clic sobre la leyenda deseada y la simulación regresará hasta ese instante.



Cuando se este ejecutando la simulación puede dar doble clic justo en el enlace de la comunicación y aparecerá una ventana que me señala una gráfica del ancho de banda. Entonces damos clic en **Graph** y enseguida a se presenta un cuadro de texto que me dice el enlace que deseo graficar y el tipo, de esta manera escoja el que necesita para sus interés.



Se puede observar además el tamaño del paquete así como su tipo dando clic en el paquete deseado como se muestra a continuación:



Al finalizar la simulación conteste las siguientes preguntas:

- ¿Cuál es el tamaño del paquete?
- ¿Cuál es el valor del intervalo?
- ¿Cuántos paquetes son descartados?
- ¿En que instante de tiempo los paquetes son desencolados?
- ¿Qué tipo de paquete se transmite de color rojo y de donde hacia donde se transmite?
- ¿Qué tipo de paquete se transmite de color azul y de donde hacia donde se transmite?

- ¿Qué tipo de paquete se transmite de color blanco y de donde hacia donde se transmite?
- ¿Qué sucede con la transmisión del paquete TCP cuando se inicia su transmisión?

A partir de la línea 26 seguido de lo que vemos a continuación añadimos las siguientes sentencias:

```
$ns duplex-link-op $n1 $n2 orient right-down
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link-op $n2 $n3 queuePos 0.5
```

*Añadimos:*

```
$ns queue-limit $n2 $n3 10
puts "Nuevo tamaño de cola entre 2 y 3 es [[$ns link $n2 $n3] queue] set
limit ]"
```

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

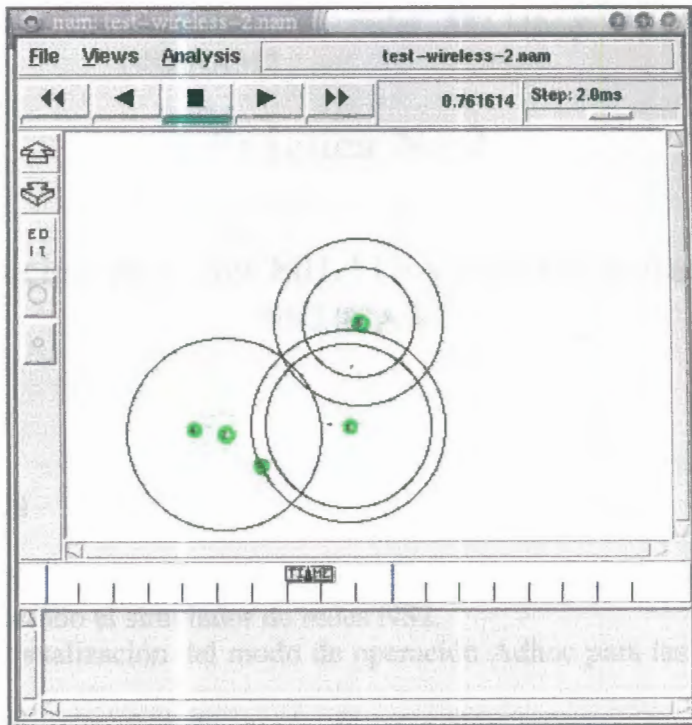
- ¿Qué sucede con la simulación ahora?
- ¿Cuál es el tamaño de la cola?
- ¿Cuántos paquetes fueron descartados?
- ¿En qué nodo se produce la saturación?

#### **Paso 4: Ejemplo de Red Inalámbrica**

Este es un ejemplo de simulación de red inalámbrica con 5 nodos. Los enlaces no son "físicos" y podemos observar como los paquetes viajan de un nodo a otro. También podemos observar los movimientos de los nodos en caso de que estos se trasladasen de un sitio a otro. Para ello nos dirigimos a la ventana del terminal y digitamos lo siguiente

```
nam test-wireless-2.nam
```

También podríamos crear redes mixtas: cableadas e inalámbricas.



Mientras se ejecuta la simulación responda las siguientes preguntas:

- ¿Entre cuales nodos se ve la comunicación de los paquetes?
- ¿Qué tipo de paquete se esta transmitiendo?
- ¿Cuál es el tamaño del paquete?
- ¿Qué tipo de paquete representan las ondas emitidas por los nodos?
- Explique detalladamente el funcionamiento de esta simulación.

## BIBLIOGRAFÍA:

- Pagina de NS, <http://www.isi.edu/nsnam/ns/>
- Manual de NS, <http://www.isi.edu/nsnam/ns/doc/index.html>
- Tutorial de ns de Marc Greis, <http://www.isi.edu/nsnam/ns/tutorial/index.html>
- NS by Example, <http://nile.wpi.edu/NS/>

## **C.2 Práctica N.- 2 “Simulación de redes 802.11b y 802.11g basados en NS2/NAM”**

### **Práctica N.- 2**

#### **Simulación de redes 801.11b y 802.11g basados en NS2/NAM**

##### **1. OBJETIVOS**

- Orientar a los estudiantes al manejo de las simulaciones de redes 802.11b y 802.11g utilizando el simulador de redes NS2.
- Creación y visualización del modo de operación Adhoc para las redes 802.11b y 802.11g
- Creación y visualización del modo de operación Infraestructura para las redes 802.11b y 802.11g
- Verificar la tasa de rendimiento de las simulaciones utilizando la herramienta awk.

##### **2. INTRODUCCIÓN**

En esta práctica se aprenderán los dos modos de operación en que trabajan las redes 802.11b y 802.11g, utilizando un simulador de redes, y a su vez se podrá observar la diferencia entre estos dos estándares de redes inalámbricas utilizando una herramienta para el tratamiento de datos que resultan de estas simulaciones.

El simulador en el cual trabajamos es el Network Simulator versión 2 (NS2). Para la simulación de redes 802.11b y 802.11g usamos la librería de 802.11 que viene incluida en el simulador, y la herramienta para interpretar la visualización gráfica de las simulaciones el Network Animator (NAM), así como la manipulación de los resultados obtenidos para la verificación de su respectivo throughput a través del archivo awk.

##### **3. MARCO TEÓRICO**

802.11 es un estándar de tecnología inalámbrica, el cual define los parámetros necesarios para la creación de redes computacionales que usan como medio de comunicación las ondas electromagnéticas. Este estándar opera en las 2 capas inferiores del modelo OSI: capa física y capa de enlace de datos. Las redes



inalámbricas nos presentan muchas ventajas como son la movilidad de los equipos, la escalabilidad que nos brinda al agregar nuevos equipos a la misma red sin la necesidad de implementar una estructura cableada adicional lo cual hace que los costos de implementación disminuyan considerablemente.

Con el paso de los años el estándar 802.11 ha tenido varias modificaciones, con la finalidad de equiparar lo mejor posible su velocidad de transferencia con las redes cableadas, que en ese entonces era de 1 a 2Mbps. Es así que surgen dos de las extensiones más utilizadas de este estándar que son 802.11b y 802.11g. Ambas con un incremento bastante considerable con respecto al estándar original, llegando a velocidades de 11Mbps y 54Mbps respectivamente.

Existen 3 bandas no licenciadas, en 900 MHz, 2,4 GHz y 5 GHz. Que son para el uso industrial, científico y medico (ISM) respectivamente. IEEE 802.11 opera en la banda de 2,4 GHz al igual que sus posteriores extensiones IEEE 802.11b y IEEE 802.11g. El alcance de las ondas electromagnéticas irradiadas es de 100 metros para entornos cerrados y 400 metros para entornos abiertos.

#### 4. MATERIALES Y EQUIPOS

- PC con Fedora Core 4 y Network Simulator 2: Network Animator NAM, Archivo awk.

#### 5. PROCEDIMIENTOS

En la pantalla de inicio de sesión de Fedora Core 4 proceda a digitar en el campo usuario estudiante, y en el campo contraseña, proceda a digitar estudiante, y al finalizar presione "Enter". Luego abrimos una ventana de consola y para ellos nos vamos a *Aplicaciones-> Accesorios-> Terminal* o en su defecto de clic derecho en el escritorio y escogemos *Abrir Terminal* y aparecerá una pantalla de esta forma.





## **Paso 1: Simulación en modo Ad-hoc**

El estándar define 2 modos de operación: Infraestructura y Ad-hoc. El modo Ad-hoc es un conjunto de 2 o mas estaciones inalámbricas conectadas directamente una con otra sin la necesidad de un punto de acceso y de esta manera compartir recursos entre ellas. A este conjunto de estaciones se las denomina Conjunto de Servicio Básico Independiente (IBSS).

### **Paso 1.1: Redes IEEE 802.11b**

El estándar 802.11 soporta tasas de transferencia de 1 Mbps y 2 Mbps, utilizando Frequency Hopping Spread Spectrum (FHSS) o Direct Sequence Spread Spectrum (DSSS) respectivamente, ambos son mecanismos de señalización totalmente diferentes y no pueden ínter operar uno con otro. FHSS nos permite un diseño de radio simple pero esta limitado a velocidades que no excedan los 2 Mbps. La técnica de Señalización DSSS divide la banda de 2.4 GHz en 14 canales de 22 MHz. El dato es enviado a través uno de los canales de 22 Mhz sin que haya salto hacia otro canal. Para compensar el ruido que hay en el canal se utiliza la técnica de “Chipping”, en la que cada bit de dato es multiplexado con patrón de 11 bits llamados “Secuencia de Barker”, cada una de estas secuencias o símbolos es transmitido a 1Mbps utilizando Binary Phase Shift Keying (BPSK), o a 2 Mbps utilizando una técnica mas sofisticada llamada Quadrature Phase Shift Keying (QPSK) la cual duplica la velocidad de transferencia mejorando el ancho de banda.

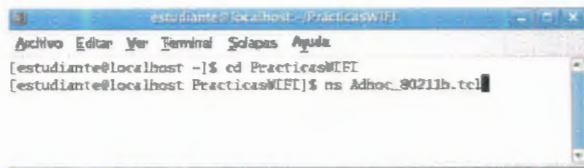
Viendo la necesidad de incrementar la tasa de transferencia, surge el estándar 802.11b, la mejora que ofrece este estándar es el soporte de dos nuevas velocidades, 5.5 Mbps y 11 Mbps. Para incrementar el data rate en 802.11b, son empleadas técnicas de codificación avanzadas. En vez de utilizar secuencias de barker de 11 bits, 802.11b especifica Complementary Code Keying (CCK), el cual consiste en un conjunto de 64 palabras de 8 bits. Como conjunto, estas palabras código tienen propiedades matemáticas únicas que permiten que sean bien diferenciadas una de la otra por un receptor aun en la presencia de ruido e interferencia multicamino elevadas. La taza de 5.5 Mbps utiliza CCK para codificar 4 bits por portadora, Mientras que la taza de 11 Mbps codifica 8 bits por portadora. Ambas velocidades utilizan QPSK como técnica de modulación.

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

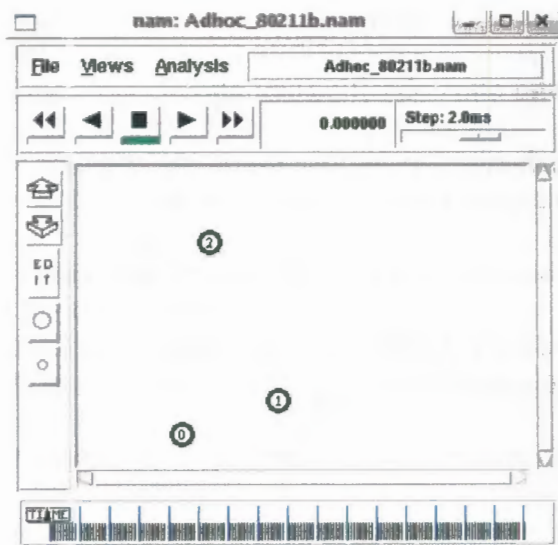
- *cd PracticasWIFI*
- *ns Adhoc\_80211b.tcl*

De esta forma veremos como se compila el programa fuente de nuestra simulación que se encontrará guardado en la carpeta *PracticasWIFI* con el nombre

*Adhoc\_80211b.tcl* y a su vez va creando la interfaz gráfica NAM para su interpretación como se muestra.



Al presionar *play* en la ventana de la NAM se ejecutará la simulación en donde observaremos círculos que son simulados como nodos móviles, así como en la parte superior de esta ventana unos controles muy familiares a los reproductores de video.



Los nodos móviles son numerados de nodo 0 al nodo 2 respectivamente, ubicados en forma aleatoria en un ambiente inalámbrico de 90 x 90m en donde las coordenadas son verificadas dando clic sobre el nodo de interés. Dentro de algún tiempo comienza la transferencia de paquetes entre dos nodos. Tenga en consideración los instantes de tiempo en el que se ejecuta un evento para contestar las preguntas que realizaremos.

Una vez terminada la simulación volvemos a la ventana del terminal escribimos la siguiente sentencia 'awk -f Throughput\_B.awk paquete=1480 fuente=0 destino=2 flujo=0 tic=1.0 Adhoc\_80211b.tr > stats.out', para obtener el valor de la tasa de rendimiento en determinados instantes de tiempo, luego nos dirigimos *Desktop->estudiante->PracticasWIFI* y abrimos el archivo **stats.out** y así verificar que estamos trabajando en un ambiente 802.11b.

```
estudiante@localhost:~/PracticasWIFI
File Edit View Terminal Tabs Help
[estudiante@localhost ~]$ cd PracticasWIFI
[estudiante@localhost PracticasWIFI]$ ns Adhoc_80211b.tcl
num_nodes is set 3
INITIALIZE THE LIST xListHead
Starting Simulation...
SORTING LISTS ...DONE!
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 422.8
[estudiante@localhost PracticasWIFI]$ awk -f Throughput_B.awk paquete=1480 fuente=0 destino=2
flujo=0 tic=1.0 Adhoc_80211b.tr > stats.out
```

Al finalizar esta simulación conteste las siguientes preguntas:

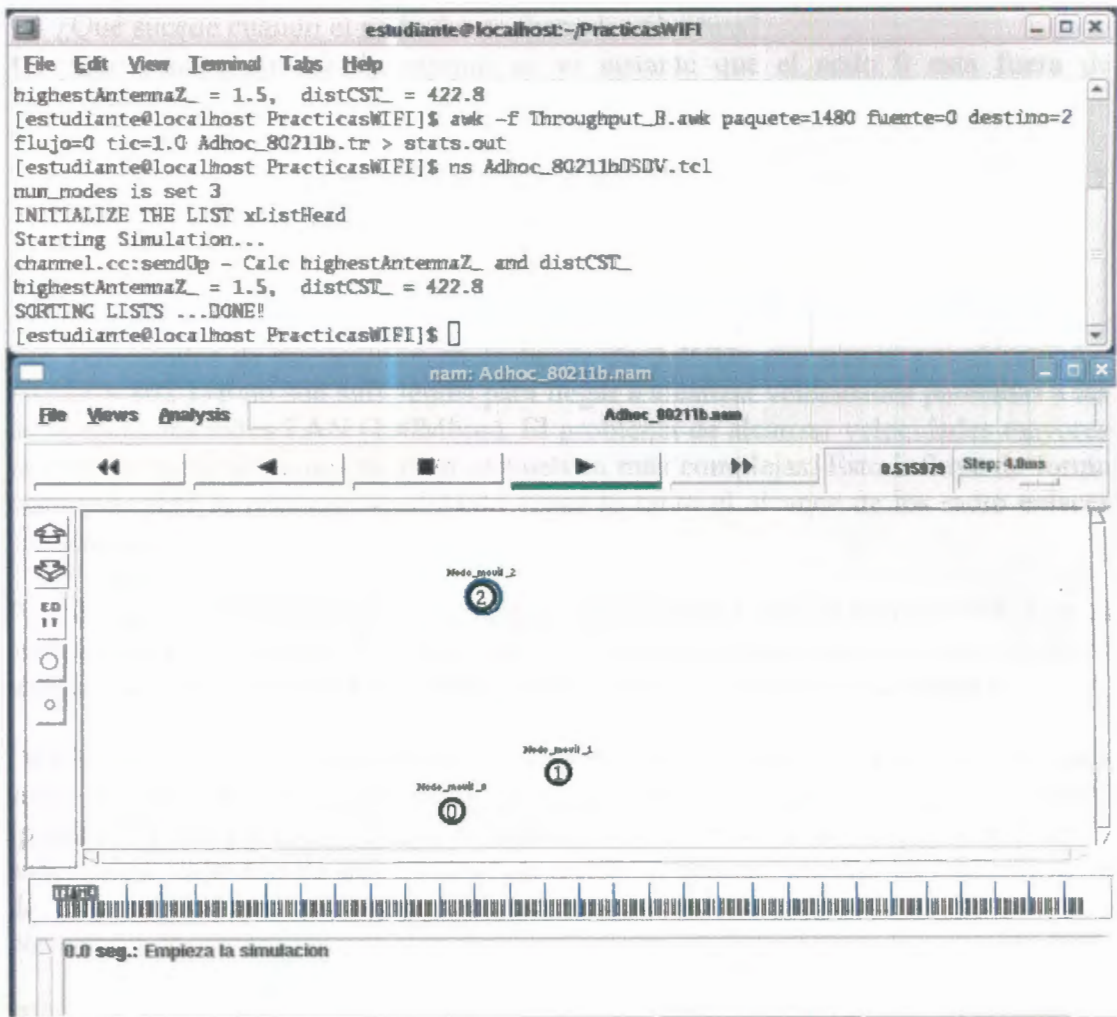
- ¿Entre que nodos se realiza la transferencia de paquetes e indique cual es el nodo fuente y cual el destino?
- ¿Cuál es el tipo de paquete que se esta transmitiendo entre estos nodos y cual es su tamaño?
- Identifique los tipos de paquetes que se encuentran en la simulación.
- ¿Qué sucede cuando el nodo 0 esta fuera de cobertura y en que instante de tiempo se produce este evento?
- ¿Cuál es el throughput promedio de la simulación y que sucede con el throughput cuando el nodo 0 esta fuera de cobertura?
- Investigue que throughput debe obtener las redes 802.11b para el tipo de paquete que se transmite en la simulación y verifique si su Throughput promedio esta dentro de este rango.
- ¿Por qué el throughput resultante no es igual a la tasa de datos de 802.11b?

Abra el script con nombre Adhoc\_80211b.tcl que se encuentra en *Desktop->estudiante->PracticasWIFI* y en la línea 13 cambie el tipo de protocolo de ruteo de AODV por DSDV

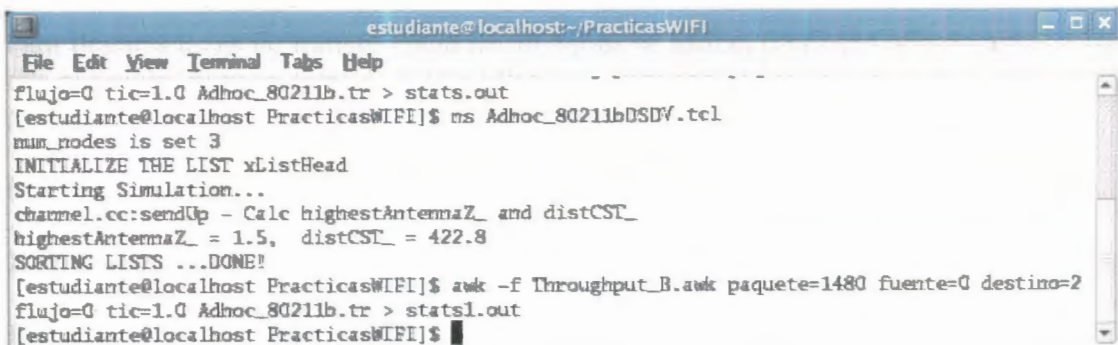
```
set opt(rp)    DSDV                ;#routing protocol
```

Guardamos como Adhoc\_80211bDSDV.tcl, compilamos y observamos en la NAM los cambios que se realizan. Ejecutamos nuevamente el archivo awk con la diferencia que creamos el archivo **stats1.out** así:





'awk -f Throughput\_B.awk paquete=1480 fuente=0 destino=2  
flujo=0 tic=1.0 Adhoc\_80211b.tr > stats1.out'



Al terminar la simulación responda las siguientes preguntas:

- ¿Qué sucede cuando el nodo 0 esta fuera de cobertura?
- ¿Qué sucede con el throughput en el instante que el nodo 0 esta fuera de cobertura?
- Investigue cual es la diferencia entre los protocolos de ruteo AODV y DSDV
- Realice en Excel una grafica comparativa entre los Throughput con AODV y DSDV para una mejor interpretación.

### **Paso 1.2: Redes IEEE 802.11g**

Las velocidades de transmisión en la banda de 2.4GHz propuestas por el estándar 802.11 y 802.11b no son suficientes para llegar a alcanzar velocidades parecidas a las de las actuales redes LAN (100Mbps). El problema de alcanzar velocidades mayores es que las técnicas de modulación se vuelven más complejas. Esto influye de forma negativa sobre la relación señal ruido y por lo tanto el alcance de los radio enlaces disminuye.

Por alcanzar velocidades parecidas a las redes LAN se crea el estándar 802.11g, el cual alcanza velocidades de transmisión de hasta 54 Mbps, para lo cual usa como técnica de modulación OFDM (Orthogonal Frequency Division Modulation).

OFDM funciona dividiendo una portadora de datos de alta velocidad en pequeñas subportadoras de baja velocidad las cuales son transmitidas simultáneamente (paralelo). Cada portadora de alta velocidad es de 20 MHz de ancho y se divide en 52 subcanales, cada uno de de 300 Khz. de ancho aproximadamente. OFDM utiliza 48 de estos subcanales para datos, mientras que los 4 restantes son usados para corrección de errores.

BPSK es usado para codificar 125 Kbps de datos por subcanal, resultando en una taza de datos de 6 Mbps. Usando QPSK se duplica la cantidad de datos codificados en 250 Kbps, dando 12 Mbps de taza de dato. Y usando 16-QAM se codifica a 4 bits por Hertz, llegando así a una taza de datos de 24 Mbps.

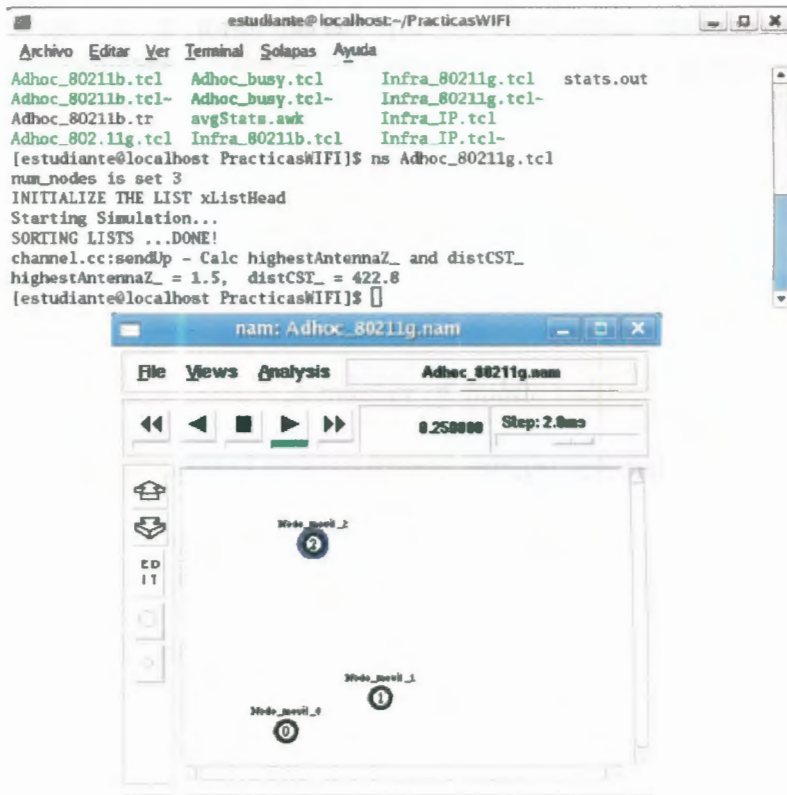
Para llegar a tazas de transferencia de 54 Mbps se utiliza 64-QAM, dando así 8 o 10 bits por ciclo, para un total de 1.125 Mbps por subcanal. Con 48 subcanales utilizados para datos, esto da como resultando una taza de datos de 54 Mbps

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

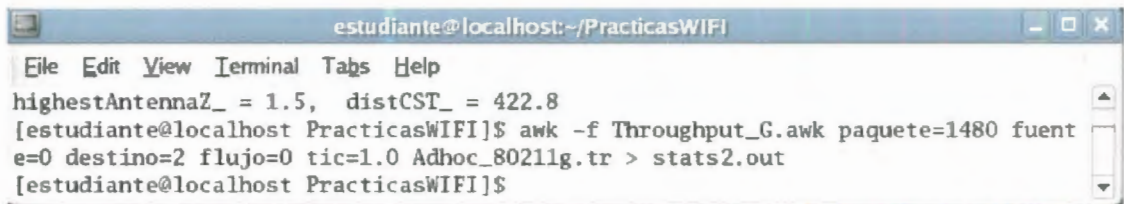
```
➤ ns Adhoc_80211g.tcl
```

Nuestra simulación se encuentra guardado en la carpeta *PracticasWIFI* con el nombre *Adhoc\_80211g.tcl* y a su vez va creando la interfaz gráfica NAM y presionamos *play* como se muestra.





Una vez terminada la simulación volvemos a la ventana del terminal escribimos la siguiente sentencia 'awk -f Throughput\_G.awk paquete= 1480 fuente=0 destino=2 flujo=0 tic=1.0 Adhoc\_80211g.tr > stats2.out', luego vamos al *Desktop->estudiante->PracticasWIFI* y abrimos el archivo **stats2.out**.



Al finalizar esta simulación conteste las siguientes preguntas:

- ¿Cuál es la diferencia básica con la simulación de 802.11b?
- ¿Cuál es el throughput promedio de la simulación y que sucede con el throughput cuando el nodo 0 esta fuera de cobertura?
- Investigue que throughput debe obtener las redes 802.11g para el tipo de paquete que se transmite en la simulación y verifique si su Throughput promedio esta dentro de este rango.

- Indique que tipo de factores harían que el throughput disminuya en una transferencia de paquetes.
- ¿Qué sucedería con el throughput si el paquete fuera UDP y como se visualizaría en la NAM?

Realice los siguientes cambios al script:

Añadiremos más nodos a nuestra simulación para ello buscamos la línea 12 y cambiamos este número por 5

```
set opt(nn) 5 ;# number of mobilenodes
```

En las líneas 67 y 68 cambiamos el archivo trace y nam por los siguientes:

```
set tracefd [open "Adhoc_80211gch.tr" w]
set tracenam [open Adhoc_80211gch.nam w]
```

En la línea 125 colocamos las coordenadas de estos 2 nuevos nodos que creamos de la siguiente manera:

```
$node_(3) set X_ 44.2
$node_(3) set Y_ 42.3
$node_(3) set Z_ 0
$node_(4) set X_ 59.2
$node_(4) set Y_ 73.3
$node_(4) set Z_ 0
```

Luego en la línea 142 cambiamos el nodo que este por el nodo 3 y la siguiente por 4 como se muestra:

```
$ns_ at 10.2 "$node_(3) setdest 62.78 41.12 17.37"
$ns_ at 11.2 "$node_(2) setdest 36.65 88.54 14.99"
$ns_ at 12.0 "$node_(4) setdest 65.63 71.44 19.38"
```

```
$ns_ at 18.2 "$node_(0) setdest 33.32 16.92 12.37"
$ns_ at 19.0 "$node_(3) setdest 72.78 41.12 19.32"
$ns_ at 12.0 "$node_(2) setdest 81.65 75.38 14.92"
```

```
$ns_ at 28.5 "$node_(4) setdest 86.91 16.31 12.58"
$ns_ at 27.0 "$node_(1) setdest 63.49 75.50 14.25"
$ns_ at 29.6 "$node_(0) setdest 55.13 29.42 19.83"
```

Luego creamos una transferencia de paquetes entre los nodos 3 y 4 y añadimos a partir de la línea 185 las siguientes sentencias:

```

set tcp_(1) [new Agent/TCP]
$tcp_(1) set fid_1
set sink_(1) [new Agent/TCPSink]
$ns_ attach-agent $node_(3) $tcp_(1)
$ns_ attach-agent $node_(4) $sink_(1)
$ns_ connect $tcp_(1) $sink_(1)
set ftp_(1) [new Application/FTP]
$ftp_(1) attach-agent $tcp_(1)
$ns_ at 13.0 "$ftp_(1) start"
$ns_ at 22.0 "$ftp_(1) stop"

```

En la línea 246 cambie el archivo nam por el cual creamos como se muestra:

```
exec nam Adhoc_80211gch.nam &
```

Guardamos los cambios como Adhoc\_80211gch, compilamos y observamos los resultados de la NAM. Luego ejecutamos el archivo awk con la diferencia que creamos el archivo **stats3.out** y el archivo **stats4.out** así:

The image shows a terminal window and a NAM (Network Animator) interface. The terminal window displays the following output:

```

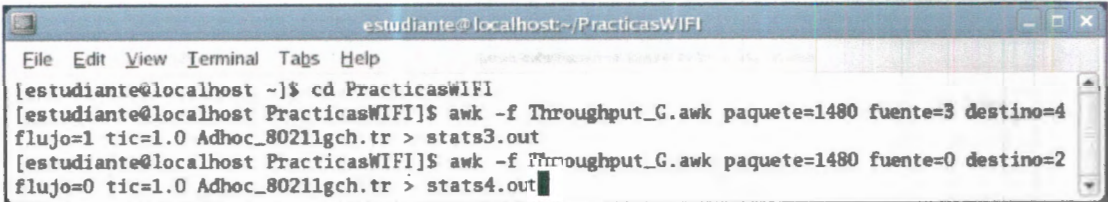
estudiante@localhost:~/PracticasWIFI
File Edit View Terminal Tabs Help
highestAntennaZ_ = 1.5, distCST_ = 422.8
[estudiante@localhost PracticasWIFI]$ ns Adhoc_80211gch.tcl
num_nodes is set 5
INITIALIZE THE LIST xListHead
Starting Simulation...
SORTING LISTS ...DONE!
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 422.8
[estudiante@localhost PracticasWIFI]$

```

The NAM interface shows a network topology with five nodes (1-5) and a progress bar at the bottom indicating '0.0 seg.: Empezar la simulación'.

```
'awk -f Throughput_G.awk paquete=1480 fuente=3 destino=4
flujo=1 tic=1.0 Adhoc_80211gch.tr > stats3.out'
```

```
'awk -f Throughput_G.awk paquete=1480 fuente=0 destino=2
flujo=0 tic=1.0 Adhoc_80211gch.tr > stats4.out'
```



```
estudiante@localhost:~/PracticasWIFI
File Edit View Terminal Tabs Help
[estudiante@localhost ~]$ cd PracticasWIFI
[estudiante@localhost PracticasWIFI]$ awk -f Throughput_G.awk paquete=1480 fuente=3 destino=4
flujo=1 tic=1.0 Adhoc_80211gch.tr > stats3.out
[estudiante@localhost PracticasWIFI]$ awk -f Throughput_G.awk paquete=1480 fuente=0 destino=2
flujo=0 tic=1.0 Adhoc_80211gch.tr > stats4.out
```

Al terminar la simulación conteste lo siguiente:

- ¿Qué sucedería con el throughput si dos o más nodos transmiten simultáneamente?
- ¿En que instante se produce la nueva comunicación entre los nodos creados?
- ¿Qué paso con el throughput del nuevo enlace?
- ¿Qué le ocurre al Throughput del enlace entre los nodos ( y 2)?
- Realice en Excel una grafica comparativa entre los Throughput del archivo stats2 y stats4 para una mejor interpretación.

## Paso 2: Simulación en modo Infraestructura

El modo Infraestructura consiste en una o varias estaciones inalámbricas conectadas a un punto de acceso, el cual proveerá a las estaciones de los servicios que nos ofrecen las redes cableadas como compartir impresoras, acceso a bases de datos, Internet, entre otros. A este conjunto: estaciones, punto de acceso y red cableada se lo denomina Conjunto de Servicio Básico (BSS). Un conjunto de 2 o más BSS se denominan Conjunto de Servicio Extendido (ESS).

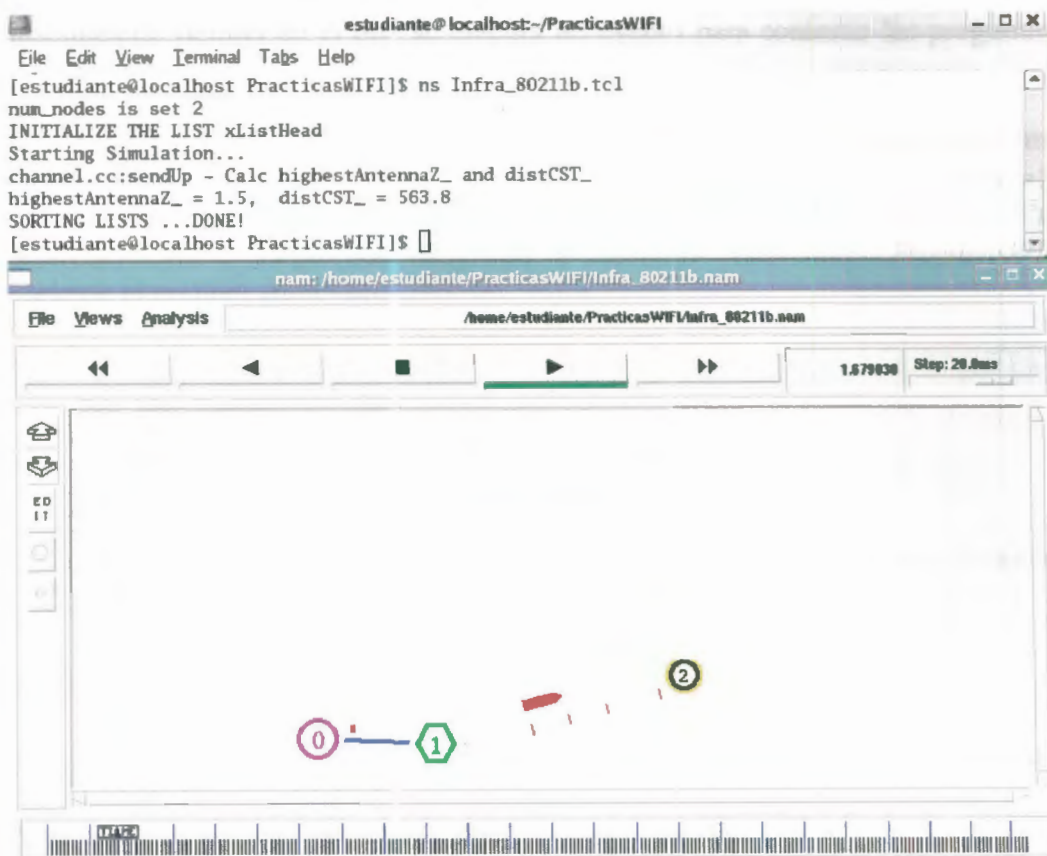
### Paso 2.1: Redes IEEE 802.11b

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

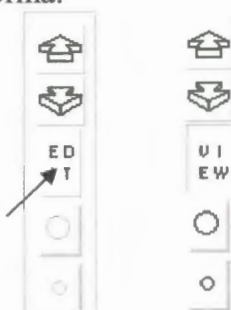
```
> ns Infra_80211b.tcl
```

Nuestra simulación se encuentra guardado en la carpeta *PracticasWIFI* con el nombre *Infra\_80211b.tcl* y a su vez va creando la interfaz gráfica NAM y presionamos *play* como se muestra.





Para una mejor visualización de la simulación editamos la posición de los nodos dentro de la NAM para ello damos clic en la parte lateral izquierda en EDIT y separamos el nodo 0 del nodo 1, volvemos a dar clic en el mismo icono para desactivarlo de la siguiente forma:

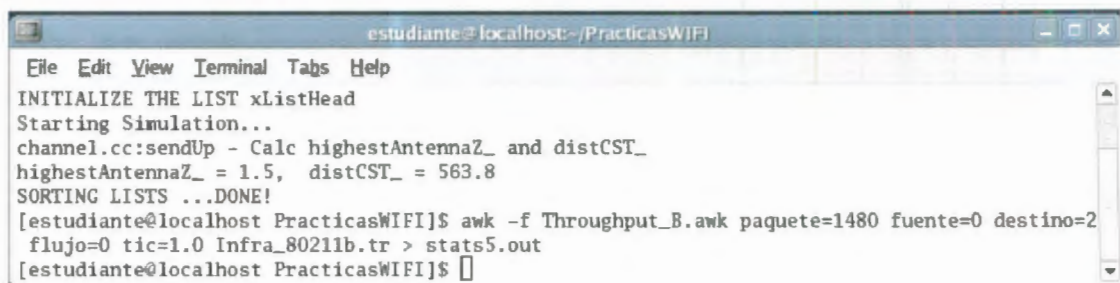


Observamos que la simulación consta de tres nodos los cuales dos son nodos fijos y un nodo es móvil, los nodos son numerados de nodo 0 al nodo 2 respectivamente, ubicados en forma jerárquica en un ambiente inalámbrico de 90 x 90m en donde las coordenadas son verificadas dando clic sobre el nodo de interés. Los nodos fijos están enlazados con una red cableada Ethernet de 100Mbps. Dentro de algún tiempo comienza la transferencia de paquetes entre dos nodos. Tenga en consideración los



instantes de tiempo en el que se ejecuta un evento para contestar las preguntas que realizaremos.

Una vez terminada la simulación volvemos a la ventana del terminal escribimos la siguiente sentencia 'awk -f Throughput\_B.awk paquete=1480 fuente=0 destino=2 flujo=0 tic=1.0 Infra\_80211b.tr > stats5.out', luego nos dirigimos a *Desktop->estudiante->PracticasWIFI* y abrimos el archivo **stats5.out**.



```
estudiante@localhost:~/PracticasWIFI
File Edit View Terminal Tabs Help
INITIALIZE THE LIST xListHead
Starting Simulation...
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 563.8
SORTING LISTS ...DONE!
[estudiante@localhost PracticasWIFI]$ awk -f Throughput_B.awk paquete=1480 fuente=0 destino=2
flujo=0 tic=1.0 Infra_80211b.tr > stats5.out
[estudiante@localhost PracticasWIFI]$
```

Al finalizar esta simulación conteste las siguientes preguntas:

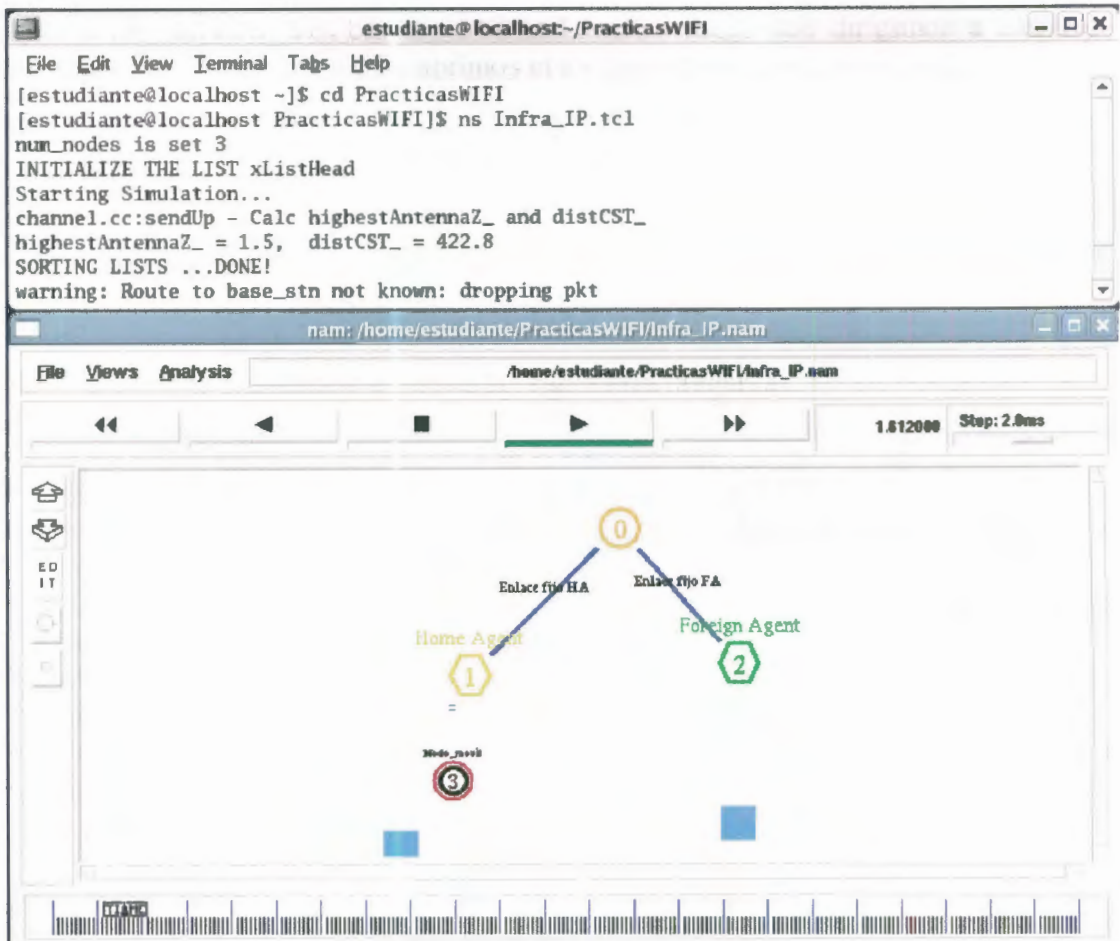
- ¿Entre que nodos se realiza la transferencia de paquetes e indique cual es el nodo fuente y cual el destino?
- ¿Qué función esta ejerciendo el nodo 1 en la simulación?
- ¿Qué sucede cuando el nodo 2 esta fuera de cobertura y en que instante de tiempo se produce este evento?
- ¿Cuál es el throughput promedio de la simulación y que sucede con el throughput cuando el nodo 2 esta fuera de cobertura?

## Paso 2.2: Redes IEEE 802.11g

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

➤ *ns Infra\_IP.tcl*

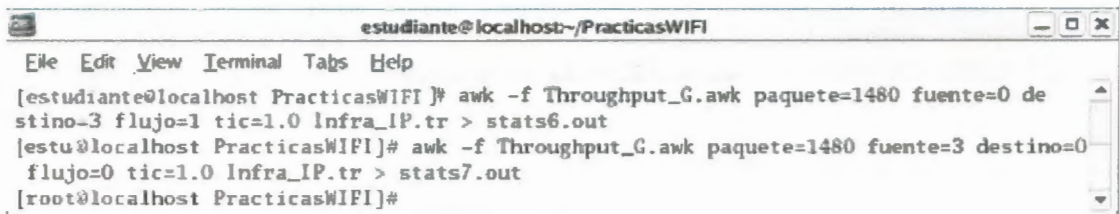
Nuestra simulación se encuentra guardado en la carpeta *PracticasWIFI* con el nombre *Infra\_IP.tcl* y a su vez va creando la interfaz gráfica NAM y presionamos *play* como se muestra.



Podemos ver que la simulación consta de cuatro nodos en los cuales tres nodos son fijos, y un nodo es móvil, ubicados en forma jerárquica en un entorno inalámbrico. Aquí volvemos a editar la posición de los nodos fijos de la misma forma que anteriormente se realizó para una mejor visualización para ello damos clic en *Edit* y movemos los nodos de la siguiente forma: el nodo 0 en las coordenadas (80, 130), el nodo 1 en (15, 70) y el nodo 2 en (180,70), esto datos son aproximados. Esta simulación explica el evento de MOBILE IP que consiste en el intercambio de una estación base hacia otra dependiendo de la cobertura, la densidad de trafico, etc. Tenga en consideración los instantes de tiempo en el que se ejecuta un evento para contestar las preguntas que realizaremos.

Una vez terminada la simulación volvemos a la ventana del terminal escribimos la siguiente sentencia `'awk -f Throughput_G.awk paquete=1480 fuente=0 destino=3 flujo=1 tic=1.0 Infra_IP.tr > stats6.out'`, y la siguiente sentencia para la otra comunicación `'awk -f Throughput_G.awk paquete=1480 fuente=3 destino=0 flujo=0`

tic=1.0 Infra\_IP.tr > stats7.out' luego nos dirigimos a *Desktop->estudiante->PracticasWIFI* y abrimos el archivo **stats6.out** y **stats7.out**.



```
estudiante@localhost:~/PracticasWIFI
File Edit View Terminal Tabs Help
[estudiante@localhost PracticasWIFI]# awk -f Throughput_G.awk paquete=1480 fuente=0 de
stino=3 flujo=1 tic=1.0 Infra_IP.tr > stats6.out
[estu@localhost PracticasWIFI]# awk -f Throughput_G.awk paquete=1480 fuente=3 destino=0
flujo=0 tic=1.0 Infra_IP.tr > stats7.out
[root@localhost PracticasWIFI]#
```

Al finalizar esta simulación conteste las siguientes preguntas:

- ¿Qué representan los nodos 1 y 2 en la simulación?
- ¿En que instante e de tiempo se realiza el traspaso de AP?
- ¿Cuál es el throughput promedio del archivo stats6 y que sucede con el throughput 3 esta fuera de cobertura?
- ¿Cuál es el throughput promedio del archivo stats7 y que sucede con el throughput 3 esta fuera de cobertura?
- ¿A cual estación base se registra el nodo 3 después que esta fuera de cobertura?
- Investigue en que consiste el procedimiento de Mobile IP
- Realice una grafica en Excel que represente el throughput a lo largo de la simulación tanto del archivo stats6 como del stats7.

## BIBLIOGRAFÍA:

- "The *ns* Manual (formerly *ns* Notes and Documentation)1", The VINT Project , UC Berkeley, LBL, USC/ISI, and Xerox PARC, Editor Kevin Fall [kfall@ee.lbl.gov](mailto:kfall@ee.lbl.gov), Editor Kannan Varadhan [kannan@catarina.usc.edu](mailto:kannan@catarina.usc.edu) ,January 8, 2003
- [2]. Wu Xiuchao, "Simulate 802.11b Channel within NS2" SOC, NUS, 2004.
- [3] Proxim. ORINOCO 11b Client PC Card Specification, 2004.
- [4]. Pablo Brenner, "A Technical Tutorial on the IEEE 802.11 Protocol", Breezecom, 1997.
- [5]. Jani Hautakorpi "IEEE 802.11 and wireless simulations in ns2".
- [7] IEEE 802.11WG, Part 11; "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", ANSI/IEEE Std 802.11-1999.



# APÉNDICE D

## Manual del Tutor

### D.1 Practica N.- 1 “Introducción al simulador NS2”

#### Práctica N.- 1

### Introducción al simulador NS2

#### 6. OBJETIVOS

- Orientar a los estudiantes sobre una herramienta didáctica para la simulación de redes de comunicación.
- Introducir a los estudiantes en un entorno de simulación utilizando el NS2.
- Realizar un script para la implementación de una simulación de una red sencilla.

#### 7. INTRODUCCION

NS (Network Simulator) es un simulador de eventos centrado en la investigación sobre redes. NS dispone de simulación para TCP, routing y multicast sobre redes cableadas o inalámbricas (locales y por satélite).

El simulador permite la configuración de gran cantidad de parámetros tales como la topología de la red, la pila de protocolos y parámetros específicos de cada protocolo. También permite evaluar el impacto de diferentes tipos de tráfico de red.

NS actualmente en la versión 2 nos permite realizar simulaciones de múltiples tipos de redes (cableadas, inalámbricas y por satélite). Para ello utiliza un lenguaje de script llamado Tcl que nos permite ir generando el modelo.

También disponemos de una interfaz gráfica llamada NAM que nos permite visualizar las simulaciones e incluso crear y editar los modelos a simular. NS es una gran herramienta que nos puede ayudar en muchos campos a la hora de realizar pruebas o generar nuevos tipos de redes

## **8. MARCO TEORICO**

NS empieza como una variante del REAL Network Simulator en 1989 y ha evolucionado substancialmente durante los últimos años. En 1995 el desarrollo lo llevaba acabo DARPA a través del proyecto VINT de LBL, Xerox PARC, UCB y USC/ISI. Actualmente el desarrollo de NS lo lleva DARPA junto a SAMAN y otros. NS siempre ha contado con contribuciones de muchos otros desarrolladores, incluyendo código inalámbrico de los proyectos CMU Monach y UCB Daedelus y también de Sun Microsystems.

NS se está utilizando tanto en entornos de investigación como en entornos educativos. NS se utiliza en entornos educativos ya que nos permite simular sencillas redes que nos van a ayudar a comprender los distintos protocolos y observar como se produce el envío de paquetes entre nodos, etc.

En NS2 se utilizan dos lenguajes C++ y Otcl. El núcleo del simulador se encuentra escrito en C++ y se encuentra disponible dicho código fuente para el estudio y modificación de los protocolos. Otcl es utilizado para definir el entorno de simulación, desde los nodos que componen la simulación, hasta las características del tráfico que se intercambia entre dichos nodos. Una vez simulado el entorno se generan archivos de trazas que podremos evaluar en modo texto con lenguajes como Awk, o gráficamente con la herramienta NAM.

El simulador NS, es un simulador extremadamente potente, diseñado por y para investigadores. A pesar de esto, y debido a que es altamente configurable, resulta también indicado para ejecutar simulaciones de redes concretas y como herramienta educativa para el estudio de protocolos.

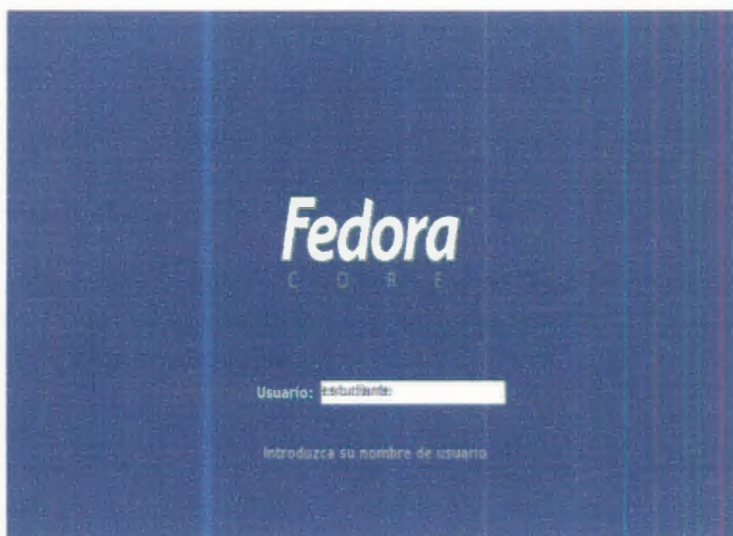
## **9. MATERIALES Y EQUIPOS**

PC con Fedora Core 4 y Network Simulator 2: Network Animador NAM.

## **10. PROCEDIMIENTOS**

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”.

Los pasos a seguir a la hora de utilizar el NS2 son:

- Describir la simulación mediante el lenguaje Otcl. Mediante este lenguaje describimos el entorno de simulación, las características de los nodos que componen dicha simulación, los parámetros de comunicación entre estos nodos, etc. Podemos encontrar un tutorial del lenguaje Otcl en <http://www.isi.edu/nsnam/ns/tutorial/index.html>.
- Una vez descrito el entorno de simulación ejecutamos el simulador:

*ns script.tcl*

- Por último analizamos los resultados que nos ofrece el simulador. El simulador nos ofrece dos archivos de trazas. El primer archivo tiene extensión .nam y es utilizado para la visualización gráfica de la simulación. El otro archivo de resultados tiene extensión .tr y es donde se almacena toda la información correspondiente a los eventos que han sucedido durante la simulación. Para analizarlos podemos usar cualquier herramienta que permita el tratamiento de archivos de texto (p.e. Perl, awk).

### **Paso 1: Descripción de los comandos básicos para la implantación de una simulación**

En este apartado vamos a analizar línea por línea la simulación que crearemos, para ello abrimos un Editor de Texto que se encuentra en *Aplicaciones -> Accesorios->*

*Editor de Texto.* Crearemos una red cableada de 4 host, los cuales en NS2 son llamados nodos.

Escribimos la siguiente sentencia:

```
set ns [new Simulator]
```

Con esta sentencia creamos una variable llamada “ns” que contiene una instancia del objeto Simulator, y así poder crear un entorno para simular. A partir de este objeto creamos el resto de objetos de nuestra simulación.

```
$ns color 0 blue  
$ns color 1 red  
$ns color 2 white
```

Asignamos colores para la visualización gráfica en NAM.

```
set n0 [$ns node]  
set n1 [$ns node]  
set n2 [$ns node]  
set n3 [$ns node]
```

Creamos 4 variables (ns0 a ns3) que representan a cuatro nodos dentro de nuestro entorno de simulación. Podemos observar que estos cuatro objetos node se crean a partir del objeto Simulator.

```
set f [open out.tr w]  
$ns trace-all $f  
set nf [open out.nam w]  
$ns namtrace-all $nf
```

Aquí se declaran los descriptores para la creación de los archivos de salida out.tr (archivo de traza) y out.nam (fichero de animación utilizado por NAM).

```
$ns duplex-link $n0 $n2 5Mb 2ms DropTail  
$ns duplex-link $n1 $n2 5Mb 2ms DropTail  
$ns duplex-link $n2 $n3 1.5Mb 10ms DropTail
```

Definimos los enlaces entre las estaciones. Por cada enlace se indica características tales como el tipo (duplex) el ancho de banda (5 Mb/s o 1.5 Mb/s) el retardo de propagación (2ms o 10 ms) y la política de servicio de las colas (DropTail corresponde a una política FIFO).

```
$ns duplex-link-op $n0 $n2 orient right-up
```

```
$ns duplex-link-op $n1 $n2 orient right-down  
$ns duplex-link-op $n2 $n3 orient right  
$ns duplex-link-op $n2 $n3 queuePos 0.5
```

Comandos utilizados por el visualizador NAM para la orientación de los nodos en nuestro entorno de simulación.

```
set udp0 [new Agent/UDP]  
$ns attach-agent $n0 $udp0  
set cbr0 [new Application/Traffic/CBR]  
$cbr0 attach-agent $udp0
```

Creamos un agente de transporte UDP en el nodo n0. Este agente generará un tráfico CBR, que puede ser de tipo stream multimedia.

```
set udp1 [new Agent/UDP]  
$ns attach-agent $n3 $udp1  
$udp1 set class_ 1  
set cbr1 [new Application/Traffic/CBR]  
$cbr1 attach-agent $udp1
```

Creamos otro agente generador de tráfico UDP en el nodo n3. La clase de servicio de este tráfico será de una clase distinta al creado en el nodo n0.

```
set null0 [new Agent/Null]  
$ns attach-agent $n3 $null0  
set null1 [new Agent/Null]  
$ns attach-agent $n1 $null1
```

La estación n3 tendrá un agente null, que representa un consumidor de tráfico. Este nodo será el receptor del tráfico generado en n0. En el nodo n1 tendremos otro consumidor de tráfico.

```
$ns connect $udp0 $null0  
$ns connect $udp1 $null1
```

Conectamos el agente generador de tráfico udp0 que se encuentra en el nodo n0 al consumidor de tráfico null0 que se localiza en el nodo n3. Además conectamos en nodo n3 con el nodo n1.

```
$ns at 1.0 "$cbr0 start"  
$ns at 1.1 "$cbr1 start"
```

En los instantes 1.0 y 1.1 comienzan los tráficos CBR que hemos definido en los nodos n0 y n3.

```
set tcp [new Agent/TCP]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $n0 $tcp
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 1.2 "$ftp start"
```

Declaramos un agente del tipo TCP entre las estaciones n0 y n3. A continuación se asocia un tráfico FTP (file transfer protocol) sobre dicho agente. En el instante 1.2 se comienza a transferir un paquete de tamaño infinito.

```
$ns at 1.35 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"
```

En el instante 1.35 se para el tráfico FTP.

```
puts "El tamaño del paquetes es [$cbr0 set packetSize_]"
puts "El valor del intervalo es [$cbr0 set interval_]"
```

Mostramos por pantalla el tamaño del paquete CBR y el intervalo entre dos paquetes consecutivos.

```
$ns at 3.0 "finish"
```

La simulación finaliza en el instante 3.0

```
proc finish {} {
    global ns fnf
    $ns_ flush-trace
    close $f
    close $nf
    puts "running nam..."
    exec nam out.nam &
    exit 0
}
```

Procedimiento que cierra los archivos donde almacenamos los resultados. Además lanzamos de forma automática el visualizador NAM para estudiar gráficamente los resultados de nuestra simulación.



*\$ns run*

Indicamos al objeto simulador que ejecute la simulación.

Como se puede observar en este archivo de traza hemos definido todas las características necesarias para llevar a cabo la simulación. Luego guardamos este script con el nombre *ejemplo1.tcl* en el siguiente directorio */Home/Practical*

## **Paso 2: Compilación de un Script**

Luego abrimos una ventana de consola y para ellos nos vamos a *Aplicaciones-> Accesorios-> Terminal* o en su defecto de clic derecho en el escritorio y escogemos *Abrir Terminal* y aparecerá una pantalla de esta forma.



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

```
cd Practical  
ns ejemplo1.tcl
```

## **Paso 3: Herramienta Network Animator, NAM**

La herramienta NAM es una herramienta de animación desarrollada en Tcl/TK que utilizando como entrada las trazas generadas automáticamente por NS permite realizar una animación de la simulación realizada con ns. Para invocar NAM se utiliza la siguiente orden:

```
nam archivo_traza.nam
```

Cuando se abre un archivo de traza con nam, se crea una ventana con la topología indicada en la simulación. La Figura 1 muestra el aspecto general de la herramienta NAM. Se han remarcado los comandos y zonas más importantes de la herramienta.

Las principales áreas y funciones de la herramienta nam son las siguientes:



- **Animation Area:** En esta zona de la ventana se visualiza el escenario y la animación del mismo.
- **Zoom In y Zoom Out:** Estos dos botones permiten acercar o alejar el escenario y poder abarcar un mayor rango de visión de la simulación.
- **Stop/Play Animation:** Como su nombre indica sirven para iniciar y detener la animación.
- **Current Animation Time:** Indica el tiempo que llevamos de simulación. Su valor llegará hasta el tiempo que le hubiéramos indicado en la simulación.
- **Step:** Este valor nos indica lo rápido que evolucionará la simulación (en milisegundos). Su valor se puede modificar con el deslizador que hay debajo del marcador.
- **Menú:** Agrupa varias opciones como por ejemplo grabar la animación, imprimir el área de animación, ver la energía de los nodos, filtrar el tipo de paquetes a visualizar (Datos, Mac, Routing), etc.

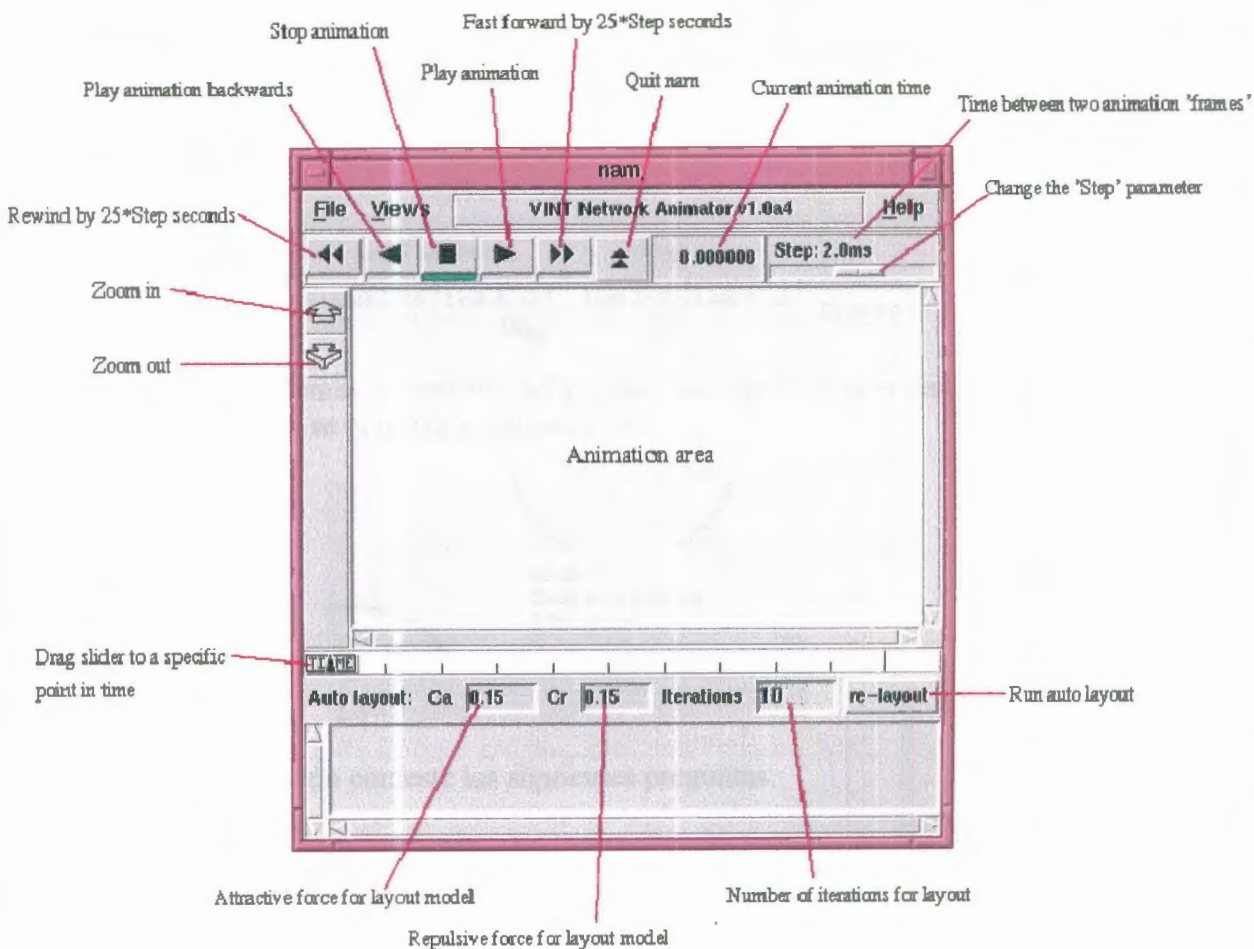
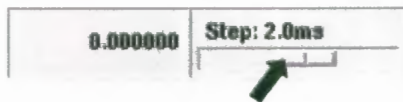


Figura 1. Estructura de la NAM

Para aumentar o disminuir la velocidad de la simulación Ud. puede manipular la escala de tiempo desplazándola de izquierda a derecha según criterio.



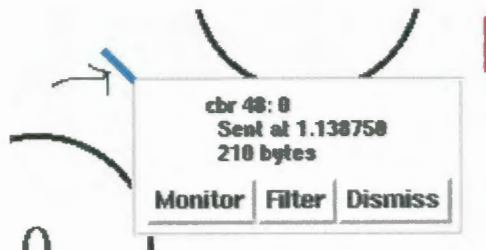
En la parte inferior de la presentación de la NAM se encuentran las leyendas que indican los momentos en que se suscita algún evento en el transcurso del tiempo. Si Ud se perdió algún evento puede dar doble clic sobre la leyenda deseada y la simulación regresará hasta ese instante.



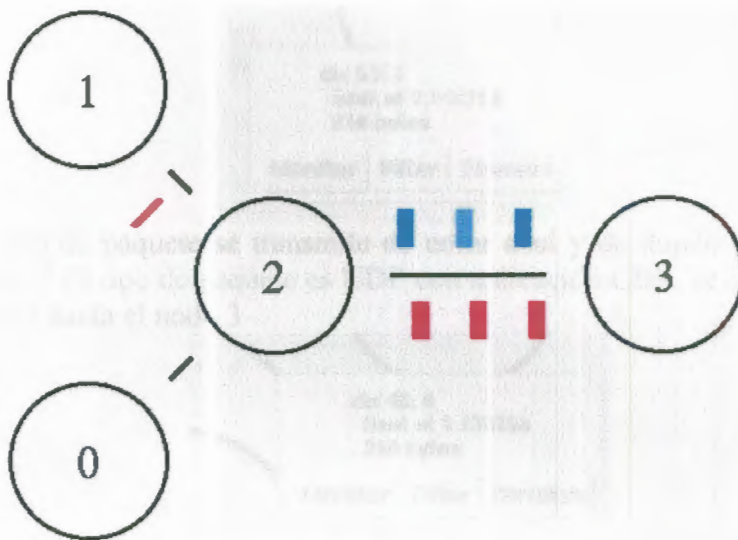
Cuando se este ejecutando la simulación puede dar doble clic justo en el enlace de la comunicación y aparecerá una ventana que me señala una gráfica del ancho de banda. Entonces damos clic en **Graph** y enseguida a se presenta un cuadro de texto que me dice el enlace que deseo graficar y el tipo, de esta manera escoja el que necesita para sus interés.



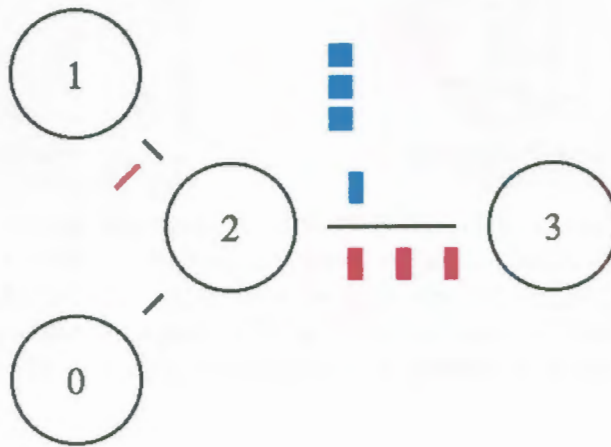
Se puede observar además el tamaño del paquete así como su tipo dando clic en el paquete deseado como se muestra a continuación:



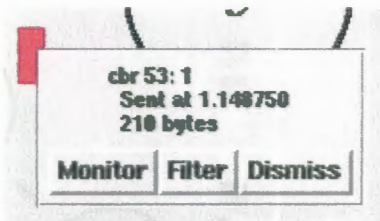
Al finalizar la simulación conteste las siguientes preguntas:



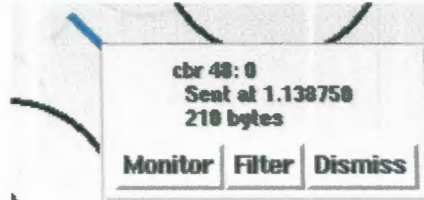
- ¿Cuál es el tamaño del paquete? 210 bytes
- ¿Cuál es el valor del intervalo? 0.003749999999 secs
- ¿Cuántos paquetes son descartados? Ninguno, todos los paquetes son encolados



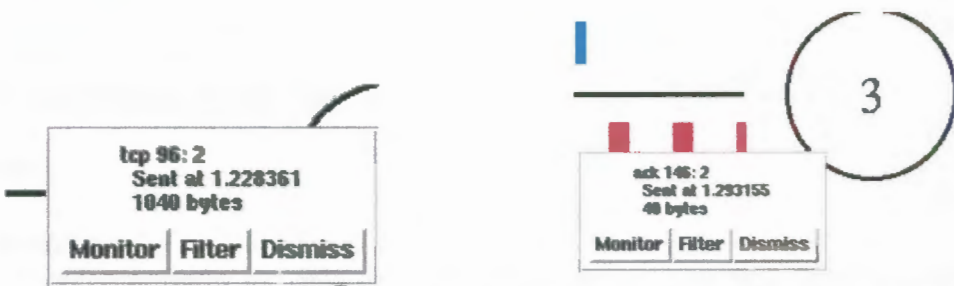
- ¿En que instante de tiempo los paquetes son desencolados? En el instante  $t = 1.43$  secs
- ¿Qué tipo de paquete se transmite de color rojo y de donde hacia donde se transmite? El tipo de paquetes es UDP con aplicación CBR, se transmite desde el nodo 3 hacia el nodo 1



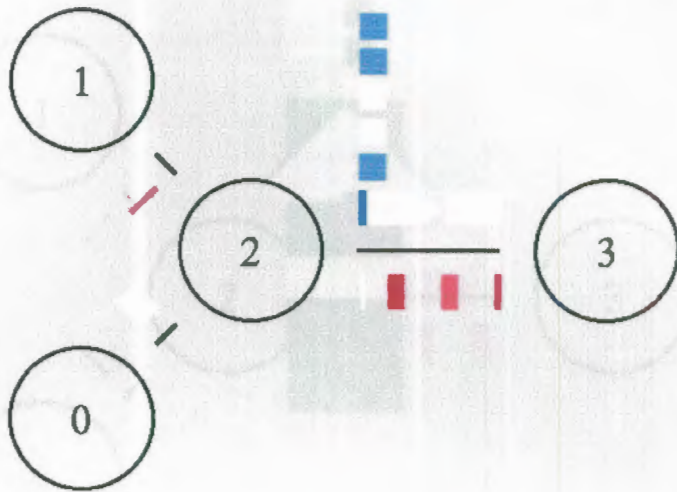
- ¿Qué tipo de paquete se transmite de color azul y de donde hacia donde se transmite? El tipo de paquete es UDP con aplicación CBR, se transmite desde el nodo 0 hacia el nodo 3



- ¿Qué tipo de paquete se transmite de color blanco y de donde hacia donde se transmite? El tipo de paquete es TCP con aplicación FTP, se transmite desde el nodo 0 al nodo 3



- ¿Qué sucede con la transmisión del paquete TCP cuando se inicia su transmisión? El nodo 0 realiza un intercalamiento entre los dos tipo de paquetes enviados pero la mayoría de los paquetes son encolados debido a que se transmite a la vez el paquete TCP y UDP por tanto la cola se llena y hay desencolamiento de paquetes en el instante se termina la transmisión TCP que es  $t = 1.35$  secs



A partir de la línea 26 seguido de lo que vemos a continuación añadimos las siguientes sentencias:

```
$ns duplex-link-op $n1 $n2 orient right-down
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link-op $n2 $n3 queuePos 0.5
```

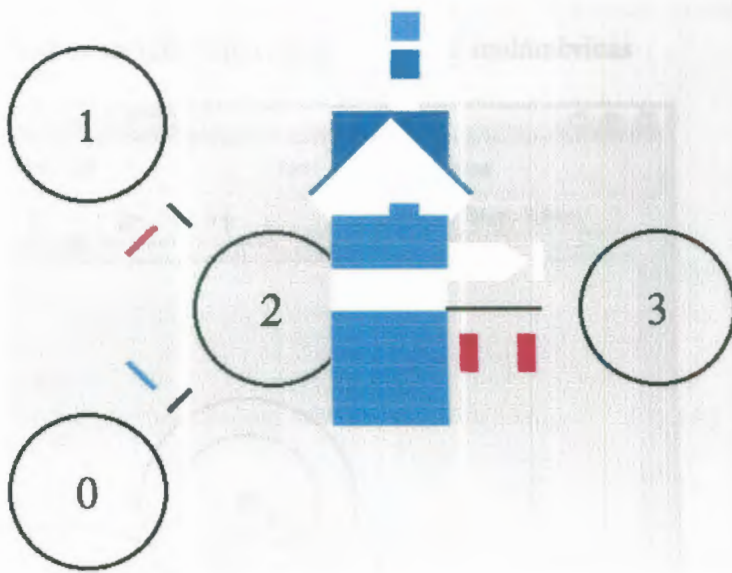
*Añadimos:*

```
$ns queue-limit $n2 $n3 10
puts "Nuevo tamaño de cola entre 2 y 3 es [[[ $ns link $n2 $n3 ] queue ] set limit ]"
```

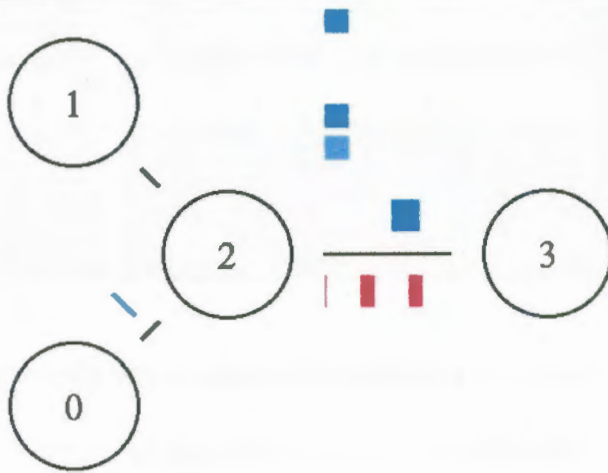
```
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
```

- ¿Qué sucede con la simulación ahora? Debido a que se cambió el tamaño de la cola se descartó paquetes que no pudieron ser encolados





- ¿Cuál es el tamaño de la cola? 10
- ¿Cuántos paquetes fueron descartados? 10
- ¿En qué nodo se produce la saturación? En el nodo 2 puesto que sirve de puente para la transmisión

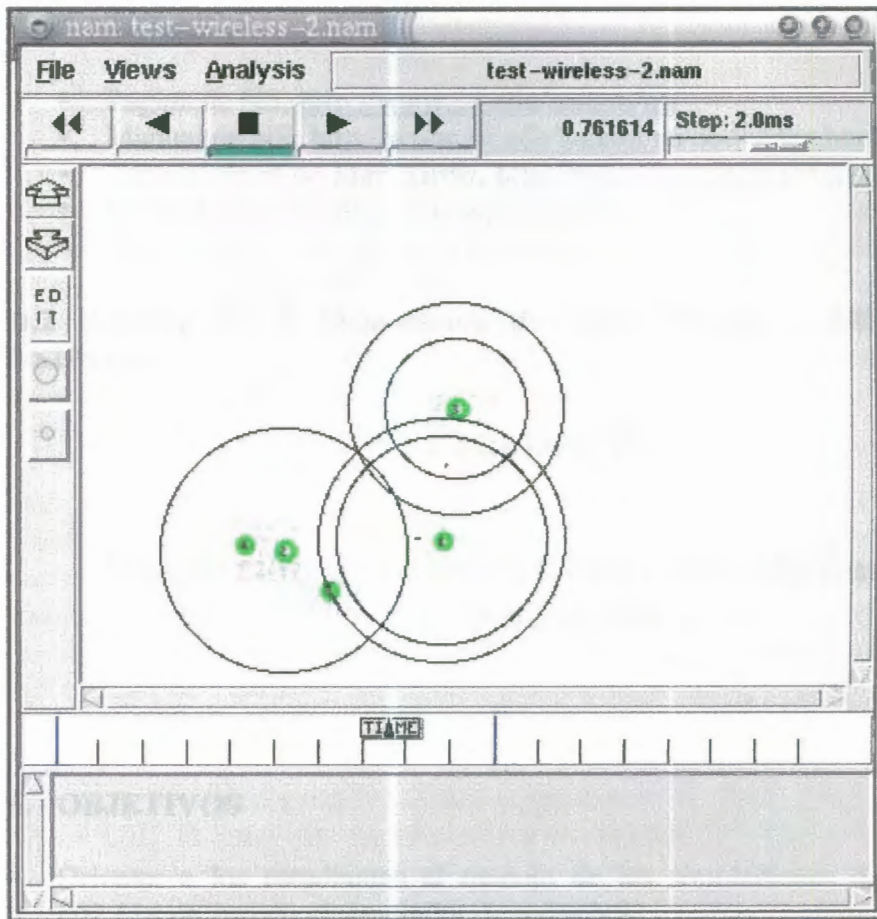


#### Paso 4: Ejemplo de Red Inalámbrica

Este es un ejemplo de simulación de red inalámbrica con 5 nodos. Los enlaces no son "físicos" y podemos observar como los paquetes viajan de un nodo a otro. También podemos observar los movimientos de los nodos en caso de que estos se trasladasen de un sitio a otro. Para ello nos dirigimos a la ventana del terminal y digitamos lo siguiente

*nam test-wireless-2.nam*

También podríamos crear redes mixtas: cableadas e inalámbricas



Mientras se ejecuta la simulación responda las siguientes preguntas:

- ¿Entre cuales nodos se ve la comunicación de los paquetes? Entre los nodos 2 y nodo 3 pero hace puente con el nodo 1
- ¿Qué tipo de paquete se esta transmitiendo? El tipo de paquete es UDP con aplicación CBR
- ¿Cuál es el tamaño del paquete? El tamaño del paquete es de 584 bytes
- ¿Qué tipo de paquete representan las ondas emitidas por los nodos? Representan a los paquetes de capa MAC
- Explique detalladamente el funcionamiento de esta simulación. La transmisión comienza en el instante  $t = 0.157$  secs donde envía las ondas de capa MAC para establecer la comunicación entre los nodos 2 y 3, al ver que se encuentran fuera de cobertura se comunican con el nodo 1 para que

establezcan la comunicación a través de el, este proceso lo realizan todo el tiempo de la simulación.

## **BIBLIOGRAFÍA:**

- Pagina de NS, <http://www.isi.edu/nsnam/ns/>
- Manual de NS, <http://www.isi.edu/nsnam/ns/doc/index.html>
- Tutorial de ns de Marc Greis, <http://www.isi.edu/nsnam/ns/tutorial/index.html>
- NS by Example, <http://nile.wpi.edu/NS/>

## **D.2 Práctica N.- 2 “Simulación de redes 802.11b y 802.11g basados en NS2/NAM”**

### **Práctica N.- 2**

#### **Simulación de redes 801.11b y 802.11g basados en NS2/NAM**

## **6. OBJETIVOS**

- Orientar a los estudiantes al manejo de las simulaciones de redes 802.11b y 802.11g utilizando el simulador de redes NS2.
- Creación y visualización del modo de operación Adhoc para las redes 802.11b y 802.11g
- Creación y visualización del modo de operación Infraestructura para las redes 802.11b y 802.11g
- Verificar la tasa de rendimiento de las simulaciones utilizando la herramienta awk.

## **7. INTRODUCCIÓN**

En esta práctica se aprenderán los dos modos de operación en que trabajan las redes 802.11b y 802.11g, utilizando un simulador de redes, y a su vez se podrá observar la diferencia entre estos dos estándares de redes inalámbricas utilizando una herramienta para el tratamiento de datos que resultan de estas simulaciones.

El simulador en el cual trabajamos es el Network Simulator versión 2 (NS2). Para la simulación de redes 802.11b y 802.11g usamos la librería de 802.11 que viene

incluida en el simulador, y la herramienta para interpretar la visualización gráfica de las simulaciones el Network Animator (NAM), así como la manipulación de los resultados obtenidos para la verificación de su respectivo throughput a través del archivo awk.

## 8. MARCO TEÓRICO

802.11 es un estándar de tecnología inalámbrica, el cual define los parámetros necesarios para la creación de redes computacionales que usan como medio de comunicación las ondas electromagnéticas. Este estándar opera en las 2 capas inferiores del modelo OSI: capa física y capa de enlace de datos. Las redes inalámbricas nos presentan muchas ventajas como son la movilidad de los equipos, la escalabilidad que nos brinda al agregar nuevos equipos a la misma red sin la necesidad de implementar una estructura cableada adicional lo cual hace que los costos de implementación disminuyan considerablemente.

Con el paso de los años el estándar 802.11 ha tenido varias modificaciones, con la finalidad de equiparar lo mejor posible su velocidad de transferencia con las redes cableadas, que en ese entonces era de 1 a 2Mbps. Es así que surgen dos de las extensiones más utilizadas de este estándar que son 802.11b y 802.11g. Ambas con un incremento bastante considerable con respecto al estándar original, llegando a velocidades de 11Mbps y 54Mbps respectivamente.

Existen 3 bandas no licenciadas, en 900 MHz, 2,4 GHz y 5 GHz. Que son para el uso industrial, científico y medico (ISM) respectivamente. IEEE 802.11 opera en la banda de 2,4 GHz al igual que sus posteriores extensiones IEEE 802.11b y IEEE 802.11g. El alcance de las ondas electromagnéticas irradiadas es de 100 metros para entornos cerrados y 400 metros para entornos abiertos.

## 9. MATERIALES Y EQUIPOS

- PC con Fedora Core 4 y Network Simulator 2: Network Animator NAM, Archivo awk.

## 10. PROCEDIMIENTOS

En la pantalla de inicio de sesión de Fedora Core 4 proceda a digitar en el campo usuario estudiante, y en el campo contraseña, proceda a digitar estudiante, y al finalizar presione "Enter". Luego abrimos una ventana de consola y para ellos nos vamos a *Aplicaciones-> Accesorios-> Terminal* o en su defecto de clic derecho en el escritorio y escogemos *Abrir Terminal* y aparecerá una pantalla de esta forma.





## **Paso 1: Simulación en modo Ad-hoc**

El estándar define 2 modos de operación: Infraestructura y Ad-hoc. El modo Ad-hoc es un conjunto de 2 o mas estaciones inalámbricas conectadas directamente una con otra sin la necesidad de un punto de acceso y de esta manera compartir recursos entre ellas. A este conjunto de estaciones se las denomina Conjunto de Servicio Básico Independiente (IBSS).

### **Paso 1.1: Redes IEEE 802.11b**

El estándar 802.11 soporta tasas de transferencia de 1 Mbps y 2 Mbps, utilizando Frecuency Hopping Spread Spectrum (FHSS) o Direct Sequence Spread Spectrum (DSSS) respectivamente, ambos son mecanismos de señalización totalmente diferentes y no pueden ínter operar uno con otro. FHSS nos permite un diseño de radio simple pero esta limitado a velocidades que no excedan los 2 Mbps. La técnica de Señalización DSSS divide la banda de 2.4 GHz en 14 canales de 22 MHz. El dato es enviado a través uno de los canales de 22 Mhz sin que haya salto hacia otro canal. Para compensar el ruido que hay en el canal se utiliza la técnica de "Chipping", en la que cada bit de dato es multiplexado con patrón de 11 bits llamados "Secuencia de Barker", cada una de estas secuencias o símbolos es transmitido a 1Mbps utilizando Binary Phase Shift Keying (BPSK), o a 2 Mbps utilizando una técnica mas sofisticada llamada Quadrature Phase Shift Keying (QPSK) la cual duplica la velocidad de transferencia mejorando el ancho de banda.

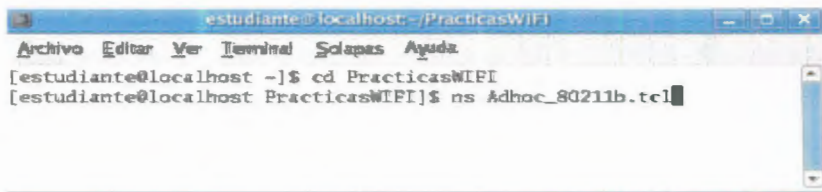
Viendo la necesidad de incrementar la tasa de transferencia, surge el estándar 802.11b, la mejora que ofrece este estándar es el soporte de dos nuevas velocidades, 5.5 Mbps y 11 Mbps. Para incrementar el data rate en 802.11b, son empleadas técnicas de codificación avanzadas. En vez de utilizar secuencias de barker de 11 bits, 802.11b especifica Complementary Code Keying (CCK), el cual consiste en un conjunto de 64 palabras de 8 bits. Como conjunto, estas palabras código tienen propiedades matemáticas únicas que permiten que sean bien diferenciadas una de la otra por un receptor aun en la presencia de ruido e interferencia multicamino elevadas. La taza de 5.5 Mbps utiliza CCK para codificar 4 bits por portadora, Mientras que la taza de 11 Mbps codifica 8 bits por portadora. Ambas velocidades utilizan QPSK como técnica de modulación.



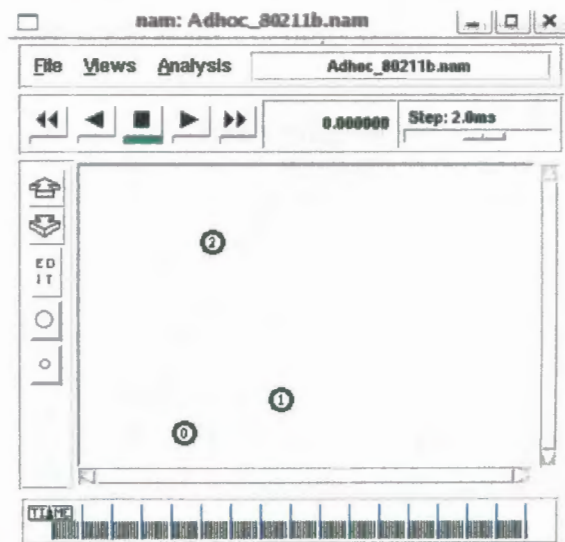
Una vez que tengamos la pantalla del terminal, ejecutaremos las siguientes sentencias:

- `cd PracticasWIFI`
- `ns Adhoc_80211b.tcl`

De esta forma veremos como se compila el programa fuente de nuestra simulación que se encontrará guardado en la carpeta *PracticasWIFI* con el nombre *Adhoc\_80211b.tcl* y a su vez va creando la interfaz gráfica NAM para su interpretación como se muestra.



Al presionar *play* en la ventana de la NAM se ejecutará la simulación en donde observaremos círculos que son simulados como nodos móviles, así como en la parte superior de esta ventana unos controles muy familiares a los reproductores de video.



Los nodos móviles son numerados de nodo 0 al nodo 2 respectivamente, ubicados en forma aleatoria en un ambiente inalámbrico de 90 x 90m en donde las coordenadas son verificadas dando clic sobre el nodo de interés. Dentro de algún tiempo comienza la transferencia de paquetes entre dos nodos. Tenga en consideración los instantes de tiempo en el que se ejecuta un evento para contestar las preguntas que realizaremos.

Una vez terminada la simulación volvemos a la ventana del terminal escribimos la siguiente sentencia 'awk -f Throughput\_B.awk paquete=1480 fuente=0 destino=2 flujo=0 tic=1.0 Adhoc\_80211b.tr > stats.out', para obtener el valor de la tasa de rendimiento en determinados instantes de tiempo, luego nos dirigimos *Desktop->estudiante->PracticasWIFI* y abrimos el archivo **stats.out** y así verificar que estamos trabajando en un ambiente 802.11b.

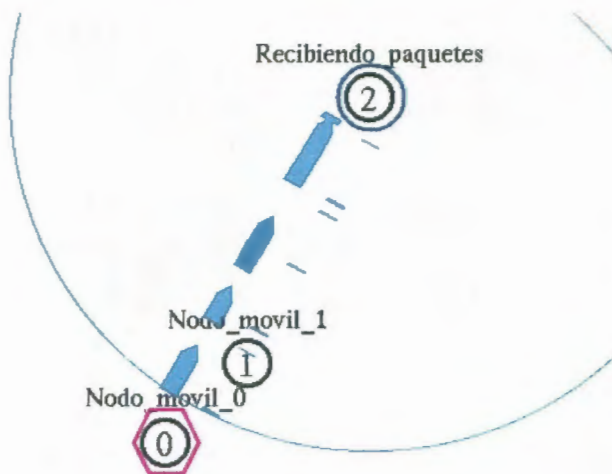
```

estudiante@localhost:~/PracticasWIFI
File Edit View Terminal Tabs Help
[estudiante@localhost ~]$ cd PracticasWIFI
[estudiante@localhost PracticasWIFI]$ ns Adhoc_80211b.tcl
num_nodes is set 3
INITIALIZE THE LIST xListHead
Starting Simulation...
SORTING LISTS ...DONE!
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 422.8
[estudiante@localhost PracticasWIFI]$ awk -f Throughput_B.awk paquete=1480 fuente=0 destino=2
flujo=0 tic=1.0 Adhoc_80211b.tr > stats.out

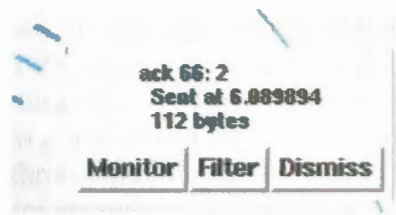
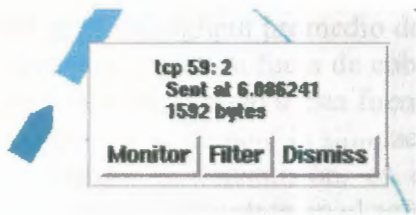
```

Al finalizar esta simulación conteste las siguientes preguntas:

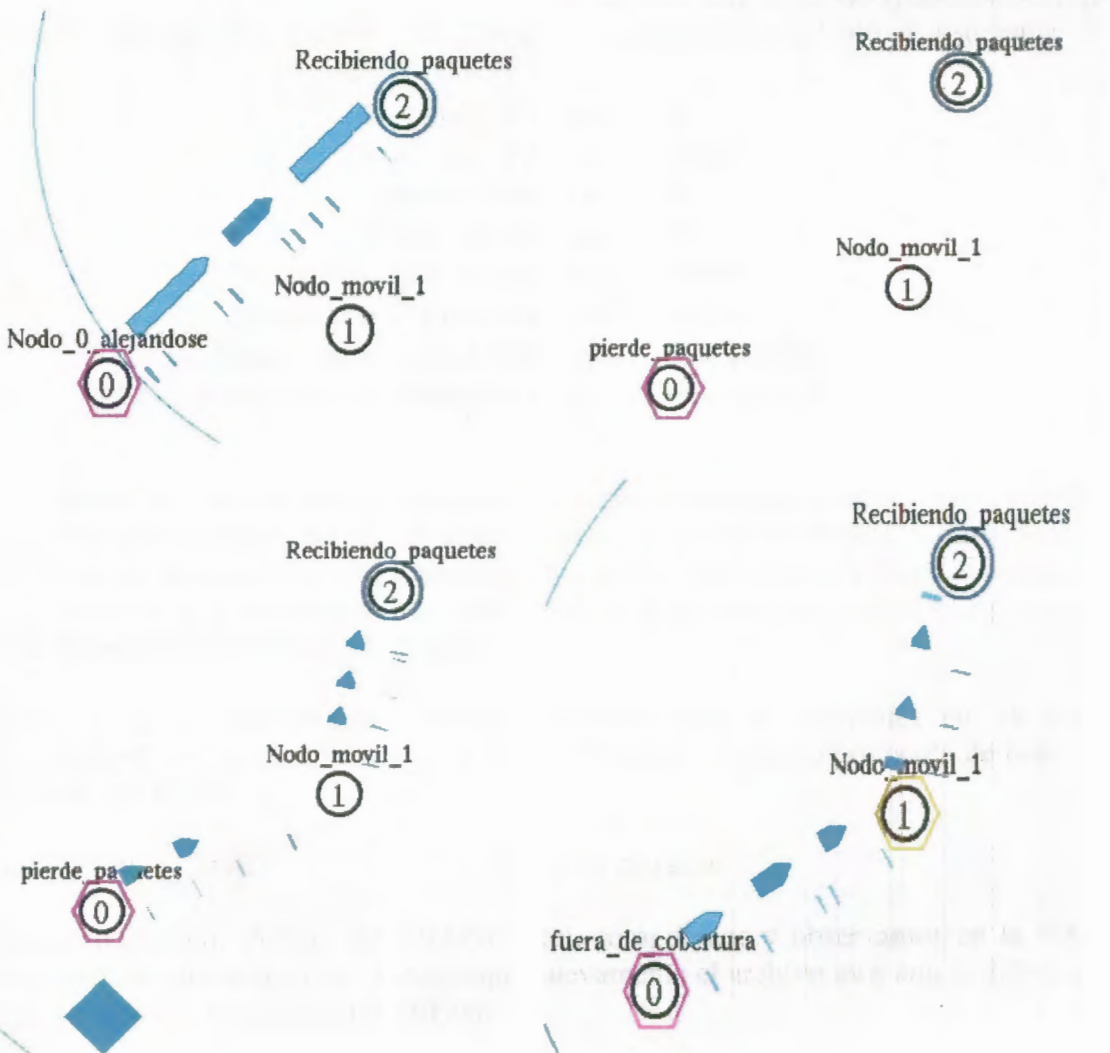
- ¿Entre que nodos se realiza la transferencia de paquetes e indique cual es el nodo fuente y cual el destino? La transferencia de paquetes se realiza entre los nodos 0 y 2 siendo el nodo 0 la fuente y el nodo 2 el destino.



- ¿Cuál es el tipo de paquete que se está transmitiendo entre estos nodos y cuál es su tamaño? El tipo de paquete es TCP con un tamaño de 1592 bytes y los paquetes ACK con un tamaño de 112 bytes, estos últimos son los paquetes de reconocimiento del paquete TCP.



- Identifique los tipos de paquetes que se encuentran en la simulación. Se encuentran los paquetes TCP, los paquetes ACK y las ondas que emiten los nodos son los RTS y los CTS respectivamente.
- ¿Qué sucede cuando el nodo 0 esta fuera de cobertura y en que instante de tiempo se produce este evento? El nodo 0 comienza a alejarse hasta el momento que esta fuera de cobertura en ese instante que es  $t = 23.4$  secs deja de transmitir y pierde paquetes, el nodo 0 sigue transmitiendo RTS por lo tanto se conecta con el nodo mas cercano al nodo 0 que en este caso es el nodo 1, continuando con la transmisión que ya se había realizado hasta terminarla.





- ¿Cuál es el throughput promedio de la simulación y que sucede con el throughput cuando el nodo 0 esta fuera de cobertura? El throughput promedio es de 3.81045 Mbps y cuando el nodo 0 esta fuera de cobertura este se reduce a la mitad durante el tiempo que se demore la simulación debido a que se introduce un nuevo enlace para realizar la transmisión, es decir el throughput depende del numero de enlaces que se encuentran en el entorno y es inversamente proporcional a este.
- Investigue que throughput debe obtener las redes 802.11b para el tipo de paquete que se transmite en la simulación y verifique si su Throughput promedio esta dentro de este rango. El throughput para un enlace optimo y para un paquete TCP es de 5.6 Mbps dando una eficiencia del 57.07% en la red, esto se debe a que el throughput depende ademas del tamaño del paquete y como se transmite a diferentes velocidades (Ej. el header se transmite de 1 a 2 Mbps mientras que el dato a 11Mbps) al realizar el calculo total se realiza un promedio en la transmisión dando como resultado estos valores. Para estos valores obtenidos no se tomo en cuenta las retransmisiones ni los RTS/CTS por tanto el valor del throughput en una transmisión real tiene que ser menor siendo igual a 4.74Mbps por tanto el valor promedio de throughput encontrado esta dentro de este rango.

```

ID del flujo: 0
Tipo de Flujo: tcp
Nodo Fuente: 0
Nodo Destino: 2
Tamaño del Paquete: 1480
Paquetes Recibidos: 6958
Tput promedio[Mbps]: 3.81045
Retardo promedio[ms]: -2.50519

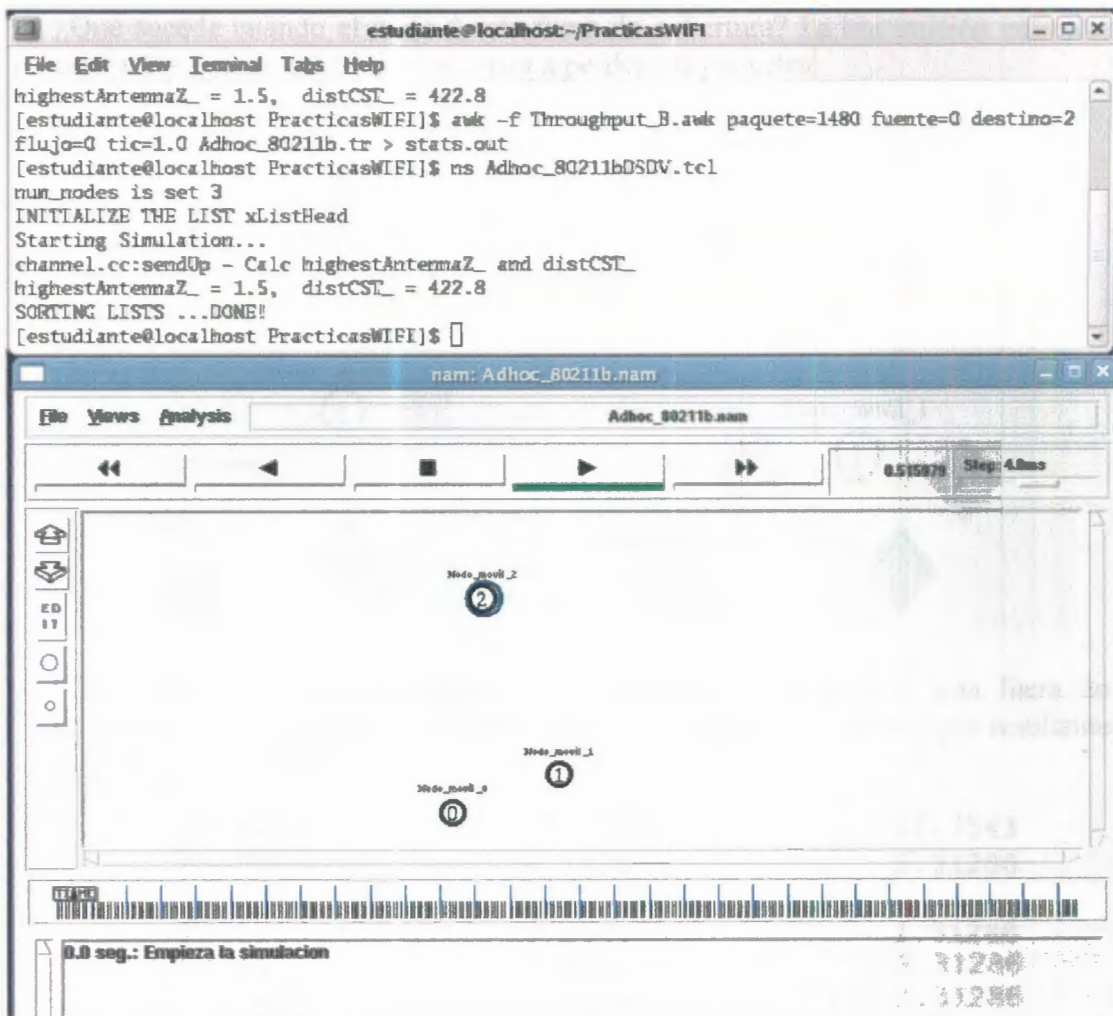
```

- ¿Por qué el throughput resultante no es igual a la tasa de datos de 802.11b? Porque los bits de datos no pueden ser transmitidos por si solos, estos necesitan estar acompañados de bits de sincronismo, correccion de errores, y otros bits que requiere la capa MAC, los cuales están siendo transmitidos a tasas de velocidad inferiores a la que esta siendo transmitido el dato, esto hace que la tasa promedio de toda la transmisión disminuya.

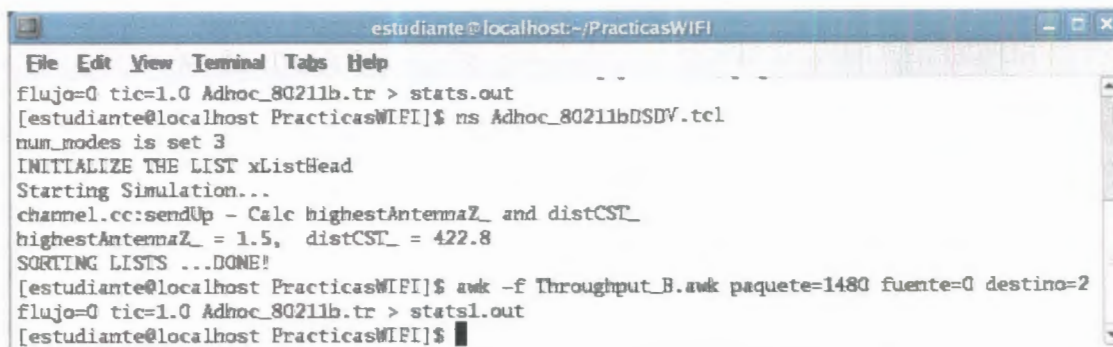
Abra el script con nombre Adhoc\_80211b.tcl que se encuentra en *Desktop->estudiante->PracticasWIFI* y en la línea 13 cambie el tipo de protocolo de ruteo de AODV por DSDV

```
set opt(rp) DSDV ;#routing protocol
```

Guardamos como Adhoc\_80211bDSDV.tcl, compilamos y observamos en la NAM los cambios que se realizan. Ejecutamos nuevamente el archivo awk con la diferencia que creamos el archivo **stats1.out** así:



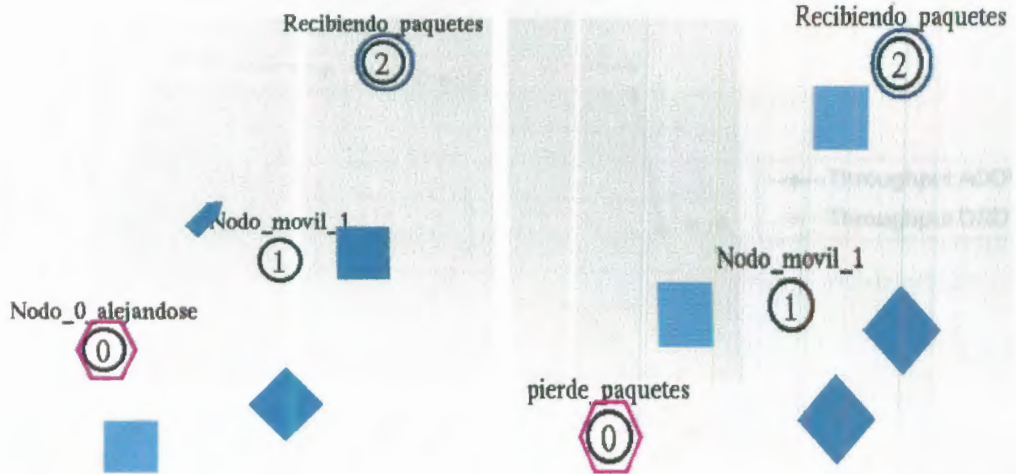
'awk -f Throughput\_B.awk paquete=1480 fuente=0 destino=2  
flujo=0 tic=1.0 Adhoc\_80211b.tr > stats1.out'



Al terminar la simulación responde las siguientes preguntas:



- ¿Qué sucede cuando el nodo 0 esta fuera de cobertura? La transmisión entre los nodos 0 y 2 se detiene y se comienza a perder los paquetes.



- ¿Qué sucede con el throughput en el instante que el nodo 0 esta fuera de cobertura? Como el nodo 0 dejo de transmitir paquetes, el throughput resultante en este tiempo es 0.

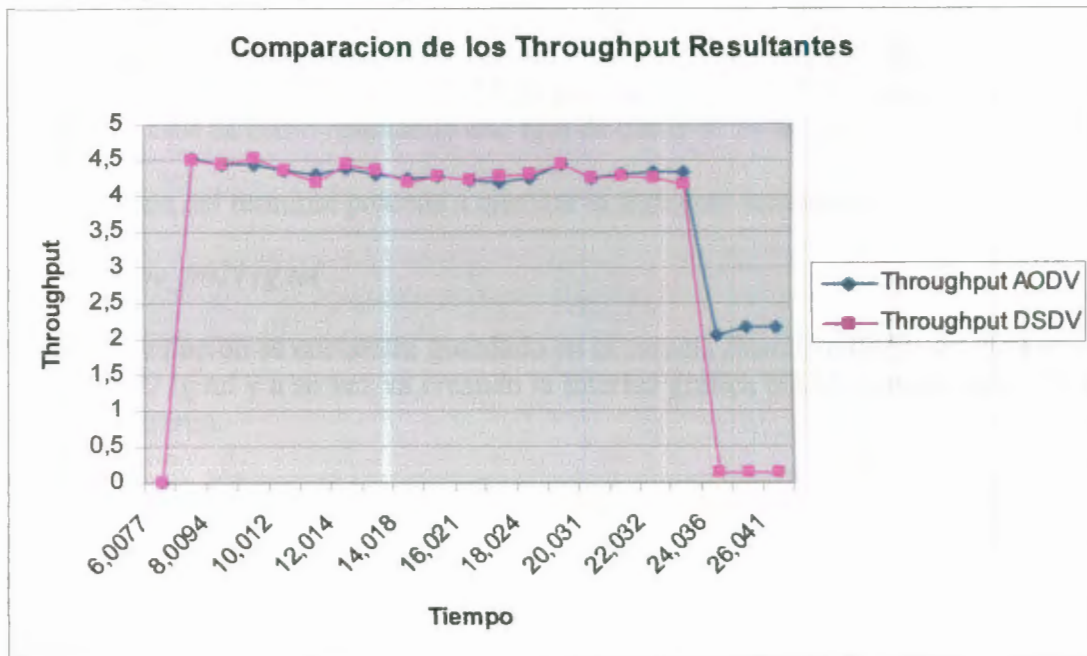
23.0321	4.15297	37.7543
24.0245	0.144415	1.31286
25.026	0.144415	1.31286
26.0285	0.144415	1.31286
27.0321	0.144415	1.31286
29.427	0.144415	1.31286

```

ID del flujo: 0
Tipo de Flujo: tcp
Nodo Fuente: 0
Nodo Destino: 2
Tamaño del Paquete: 1480
Paquetes Recibidos: 6302
Tput promedio[Mbps]: 3.87642
Retardo promedio[ms]: 0
  
```

- Investigue cual es la diferencia entre los protocolos de ruteo AODV y DSDV La diferencia que existe entre estos protocolos es que AODV permite un enrutamiento hop to hop o salto a salto, esta cualidad nos permite encaminar paquetes a destinos sin cobertura directa a través de otros nodos intermedios que se encuentran en la red.

- Realice en Excel una grafica comparativa entre los Throughput con AODV y DSDV para una mejor interpretación.



## Paso 1.2: Redes IEEE 802.11g

Las velocidades de transmisión en la banda de 2.4GHz propuestas por el estándar 802.11 y 802.11b no son suficientes para llegar a alcanzar velocidades parecidas a las de las actuales redes LAN (100Mbps). El problema de alcanzar velocidades mayores es que las técnicas de modulación se vuelven más complejas. Esto influye de forma negativa sobre la relación señal ruido y por lo tanto el alcance de los radio enlaces disminuye.

Por alcanzar velocidades parecidas a las redes LAN se crea el estándar 802.11g, el cual alcanza velocidades de transmisión de hasta 54 Mbps, para lo cual usa como técnica de modulación OFDM (Orthogonal Frequency Division Modulation).

OFDM funciona dividiendo una portadora de datos de alta velocidad en pequeñas subportadoras de baja velocidad las cuales son transmitidas simultáneamente (paralelo). Cada portadora de alta velocidad es de 20 MHz de ancho y se divide en 52 subcanales, cada uno de de 300 Khz. de ancho aproximadamente. OFDM utiliza 48 de estos subcanales para datos, mientras que los 4 restantes son usados para corrección de errores.

BPSK es usado para codificar 125 Kbps de datos por subcanal, resultando en una taza de datos de 6 Mbps. Usando QPSK se duplica la cantidad de datos codificados en 250

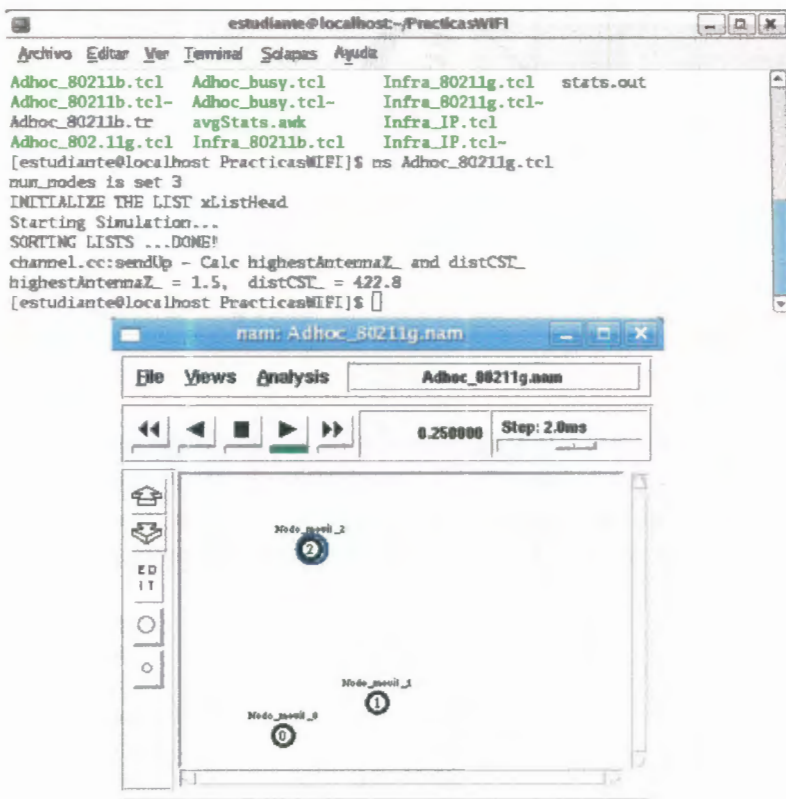
Kbps, dando 12 Mbps de taza de dato. Y usando 16-QAM se codifica a 4 bits por Hertz, llegando así a una taza de datos de 24 Mbps.

Para llegar a tazas de transferencia de 54 Mbps se utiliza 64-QAM, dando así 8 o 10 bits por ciclo, para un total de 1.125 Mbps por subcanal. Con 48 subcanales utilizados para datos, esto da como resultando una taza de datos de 54 Mbps

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

➤ `ns Adhoc_80211g.tcl`

Nuestra simulación se encuentra guardado en la carpeta *PracticasWIFI* con el nombre *Adhoc\_80211g.tcl* y a su vez va creando la interfaz gráfica NAM y presionamos *play* como se muestra.



Una vez terminada la simulación volvemos a la ventana del terminal escribimos la siguiente sentencia '`awk -f Throughput_G.awk paquete= 1480 fuente=0 destino=2 flujo=0 tic=1.0 Adhoc_80211g.tr > stats2.out`', luego vamos al *Desktop->estudiante->PracticasWIFI* y abrimos el archivo **stats2.out**.

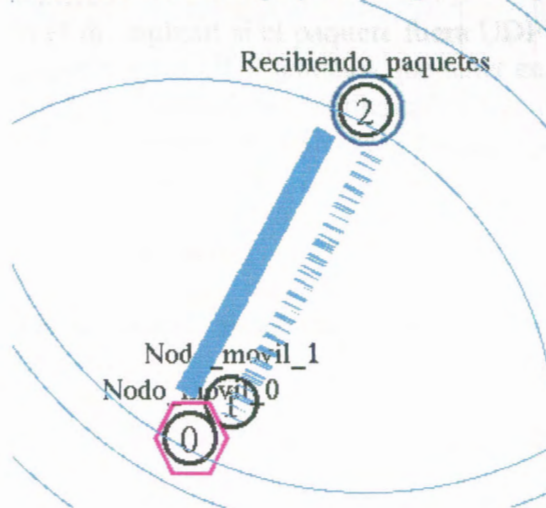
```

estudiante@localhost:~/PracticasWIFI
File Edit View Terminal Tabs Help
highestAntennaZ_ = 1.5, distCST_ = 422.8
[estudiante@localhost PracticasWIFI]$ awk -f Throughput_G.awk paquete=1480 fuente=0 destino=2 flujo=0 tic=1.0 Adhoc_80211g.tr > stats2.out
[estudiante@localhost PracticasWIFI]$

```

Al finalizar esta simulación conteste las siguientes preguntas:

- ¿Cuál es la diferencia básica con la simulación de 802.11b? La diferencia básica consiste en que los paquetes se transmite mas rápidamente y con mayor volumen puesto que 802.11b transmite a 11Mbps y 802.11g a 54Mbps



- ¿Cuál es el throughput promedio de la simulación y que sucede con el throughput cuando el nodo 0 esta fuera de cobertura? El throughput promedio es 19.2748 debido a que cuando el nodo 0 esta fuera de cobertura realiza la transferencia a través del nodo 1 y reduce a la mitad el throughput en ese tiempo.

21.0074	21.0829	39.04:25
22.0075	21.6414	40.07:66
23.0077	20.9989	38.88:68
24.0078	11.0107	20.39:02
25.0088	10.9408	20.26:08
26.0097	10.8002	20.00:03

```

ID del flujo: 0
Tipo de Flujo: tcp
Nodo Fuente: 0
Nodo Destino: 2
Tamaño del Paquete: 1480
Paquetes Recibidos: 35131
Tput promedio[Mbps]: 19.2748
Retardo promedio[ms]: -0.496406

```



- Investigue que throughput debe obtener las redes 802.11g para el tipo de paquete que se transmite en la simulación y verifique si su Throughput promedio esta dentro de este rango. Para valores óptimos de transmisión y sin retransmisiones el valor del throughput es de 27 Mbps con una eficiencia del 57% pero para valores reales con retransmisiones y RTS/CTS el throughput es de 24 Mbps para un paquete TCP por ende nuestro throughput está dentro de este rango
- Indique que tipo de factores harían que el throughput disminuya en una transferencia de paquetes. Como ya hemos mencionado el numero de enlaces en un entorno inalámbrico es de mucha influencia para el throughput, ademas esta la distancia entre los nodos, el medio en el cual se transmite los paquetes y el tamaño en si del paquete.
- ¿Qué sucedería con el throughput si el paquete fuera UDP y como se visualizaría en la NAM? Si el paquete fuera UDP tenemos que tener en cuenta que no existiría los ACK por tanto el throughput aumentaría debido a que no abrían retransmisiones, y la diferencia en la NAM se mostraría igual con la diferencia de que no habrían los paquetes ACK.

Realice los siguientes cambios al script:

Añadiremos más nodos a nuestra simulación para ello buscamos la línea 12 y cambiamos este número por 5

```
set opt(nn) 5 ;# number of mobilenodes
```

En las líneas 67 y 68 cambiamos el archivo trace y nam por los siguientes:

```
set tracefd [open "Adhoc_80211gch.tr" w]
set tracenam [open Adhoc_80211gch.nam w]
```

En la línea 125 colocamos las coordenadas de estos 2 nuevos nodos que creamos de la siguiente manera:

```
$node_(3) set X_ 44.2
$node_(3) set Y_ 42.3
$node_(3) set Z_ 0
$node_(4) set X_ 59.2
$node_(4) set Y_ 73.3
$node_(4) set Z_ 0
```

Luego en la línea 142 cambiamos el número de nodo que este por los siguientes cambios:

```
$ns_ at 10.2 "$node_(3) setdest 62.78 41.12 17.37"
```



```
$ns_ at 11.2 "$node_(2) setdest 36.65 88.54 14.99"  
$ns_ at 12.0 "$node_(4) setdest 65.63 71.44 19.38"
```

```
$ns_ at 18.2 "$node_(0) setdest 33.32 16.92 12.37"  
$ns_ at 19.0 "$node_(3) setdest 72.78 41.12 19.32"  
$ns_ at 12.0 "$node_(2) setdest 81.65 75.38 14.92"
```

```
$ns_ at 28.5 "$node_(4) setdest 86.91 16.31 12.58"  
$ns_ at 27.0 "$node_(1) setdest 63.49 75.50 14.25"  
$ns_ at 29.6 "$node_(0) setdest 55.13 29.42 19.83"
```

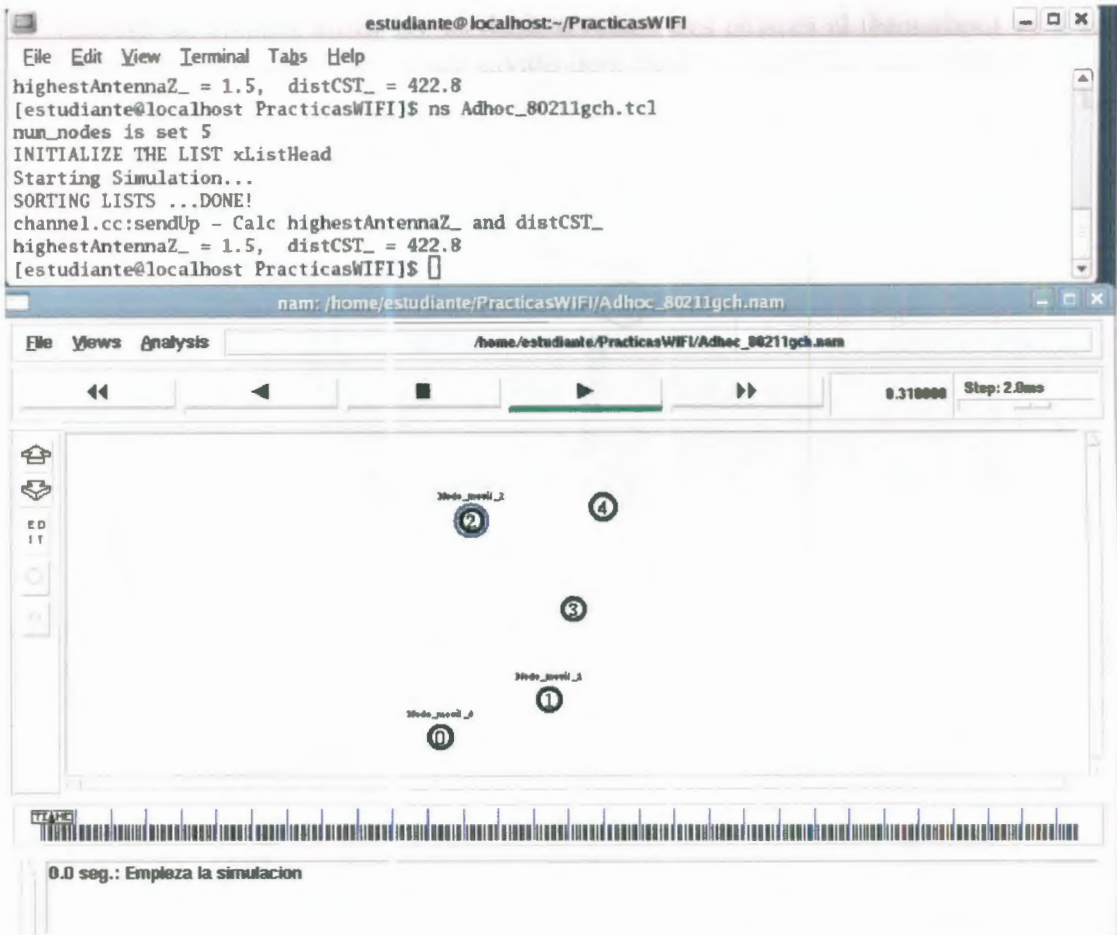
Luego creamos una transferencia de paquetes entre los nodos 3 y 4 y añadimos a partir de la línea 185 las siguientes sentencias:

```
set tcp_(1) [new Agent/TCP]  
$tcp_(1) set fid_1  
set sink_(1) [new Agent/TCPSink]  
$ns_ attach-agent $node_(3) $tcp_(1)  
$ns_ attach-agent $node_(4) $sink_(1)  
$ns_ connect $tcp_(1) $sink_(1)  
set ftp_(1) [new Application/FTP]  
$ftp_(1) attach-agent $tcp_(1)  
$ns_ at 13.0 "$ftp_(1) start"  
$ns_ at 22.0 "$ftp_(1) stop"
```

En la línea 246 cambie el archivo nam por el cual creamos como se muestra:

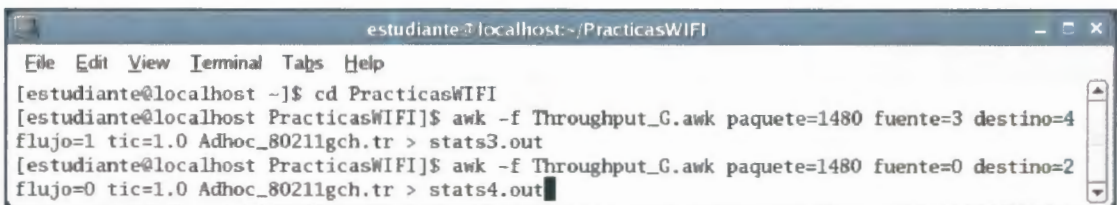
```
exec nam Adhoc_80211gch.nam &
```

Guardamos los cambios como Adhoc\_80211gch, compilamos y observamos los resultados de la NAM. Luego ejecutamos el archivo awk con la diferencia que creamos el archivo **stats3.out** y el archivo **stats4.out** así:



```
'awk -f Throughput_G.awk paquete=1480 fuente=3 destino=4
flujo=1 tic=1.0 Adhoc_80211gch.tr > stats3.out'
```

```
'awk -f Throughput_G.awk paquete=1480 fuente=0 destino=2
flujo=0 tic=1.0 Adhoc_80211gch.tr > stats4.out'
```

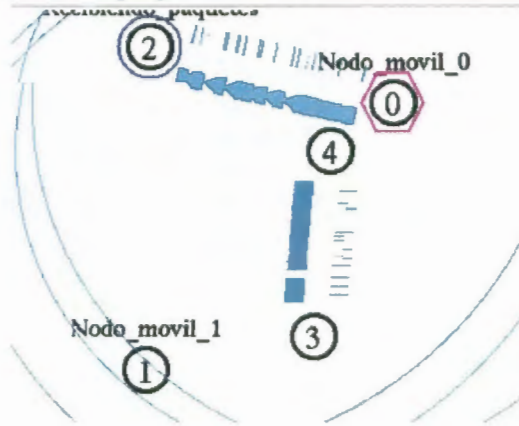


Al terminar la simulación conteste lo siguiente:

- ¿Qué sucedería con el throughput si dos o más nodos transmiten simultáneamente? Como vemos en la simulación al añadir una nueva comunicación entre nodos el throughput disminuye inversamente proporcional al

numero de enlaces añadidos, es decir si tengo tres enlaces el throughput va a ser igual al throughput de un enlace dividido para tres.

- ¿En que instante se produce la nueva comunicación entre los nodos creados? La nueva comunicación de los paquetes se realiza en el instante  $t = 13.0$  secs.



- ¿Qué sucede con el throughput del nuevo enlace? El throughput del nuevo enlace cayó a la mitad puesto que el throughput promedio de un enlace es de 20 Mbps, como ya hemos dicho dividimos para dos debido al enlace nuevo y nuestro nuevo throughput es de 10 Mbps aproximadamente.

Tiempo	Throughput[Mbps]	Eficiencia[%]
1	0	0
12.0052	0	0
13.0052	0.000910403	0.00168593
14.0054	10.1691	18.8317
15.0056	9.58833	17.7562
16.0057	10.679	19.7759
17.0059	10.9255	20.2324
18.0061	9.76592	18.085
19.0066	10.0118	18.5403
20.0073	11.2877	20.9031
21.0098	9.88552	18.3065
22.0122	10.7012	19.8171

```

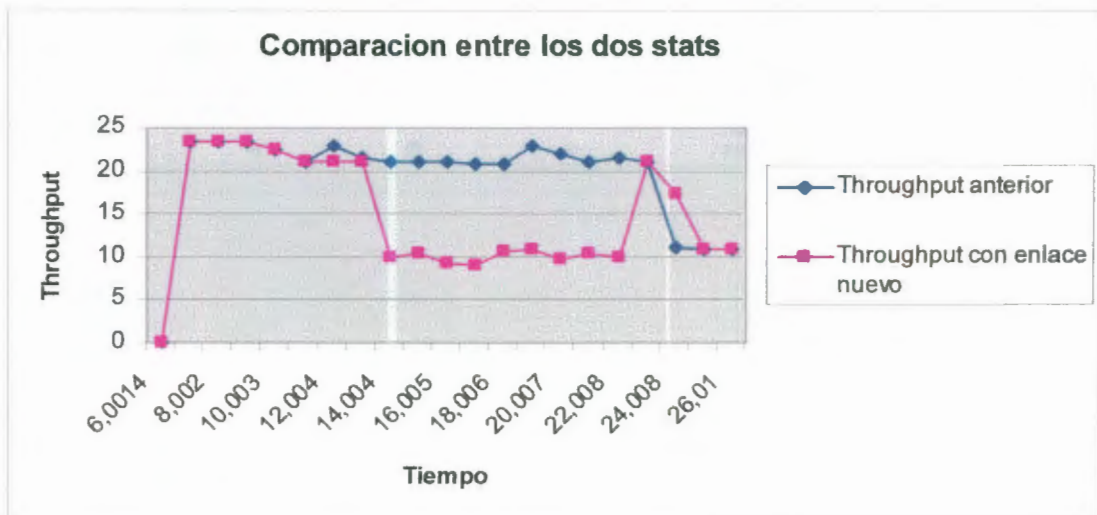
ID del flujo: 0
Tipo de Flujo: tcp
Nodo Fuente: 3
Nodo Destino: 4
Tamaño del Paquete: 1480
Paquetes Recibidos: 7866
Tput promedio[Mbps]: 9.30149
Retardo promedio[ms]: 0
    
```

- ¿Qué le ocurre al Throughput del enlace entre los nodos 0 y 2? El throughput se reduce a la mitad como era de esperarse.

12.0028	20.9993	38.8875
13.0031	21.1395	39.1473
14.0047	9.90629	18.345
15.0047	10.4662	19.3818
16.01	9.24574	17.1217
17.0106	9.1351	16.9169
18.0138	10.5391	19.5169
19.0138	10.9874	20.3471
20.0143	9.72747	18.0138
21.0168	10.5239	19.4887
22.0186	9.84457	18.2307
23.0187	21.0737	39.0254
24.1325	17.4657	32.3439
25.1353	10.8268	20.0497
26.1363	10.952	20.2816

ID del flujo: 0  
 Tipo de Flujo: tcp  
 Nodo Fuente: 0  
 Nodo Destino: 2  
 Tamaño del Paquete: 1480  
 Paquetes Recibidos: 26882  
 Tput promedio[Mbps]: 14.5974  
 Retardo promedio[ms]: -0.934822

- Realice en Excel una grafica comparativa entre los Throughput del archivo stats2 y stats4 para una mejor interpretación.



## Paso 2: Simulación en modo Infraestructura

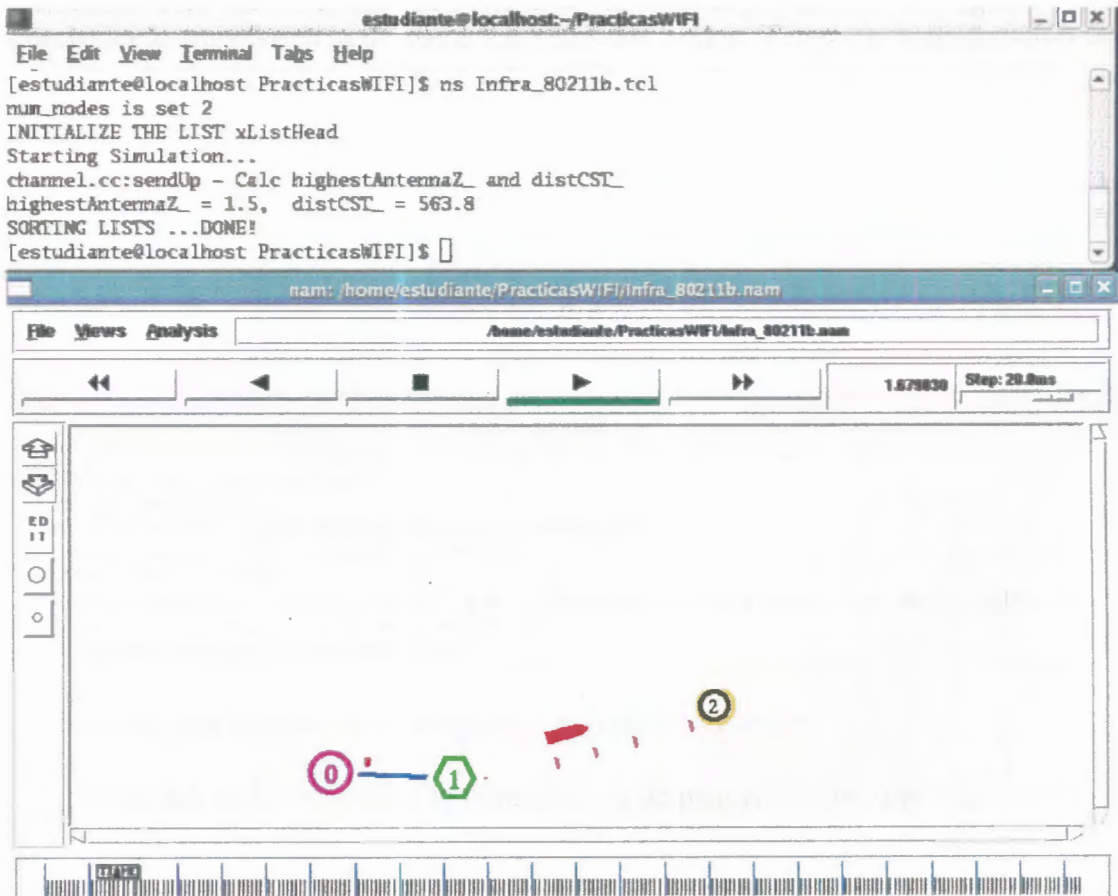
El modo Infraestructura consiste en una o varias estaciones inalámbricas conectadas a un punto de acceso, el cual proveerá a las estaciones de los servicios que nos ofrecen las redes cableadas como compartir impresoras, acceso a bases de datos, Internet, entre otros. A este conjunto: estaciones, punto de acceso y red cableada se lo denomina Conjunto de Servicio Básico (BSS). Un conjunto de 2 o más BSS se denominan Conjunto de Servicio Extendido (ESS).

### Paso 2.1: Redes IEEE 802.11b

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

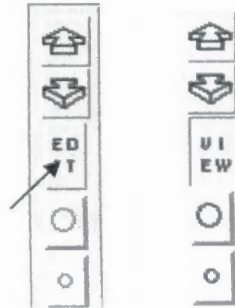
➤ `ns Infra_80211b.tcl`

Nuestra simulación se encuentra guardado en la carpeta *PracticasWIFI* con el nombre *Infra\_80211b.tcl* y a su vez va creando la interfaz gráfica NAM y presionamos *play* como se muestra.



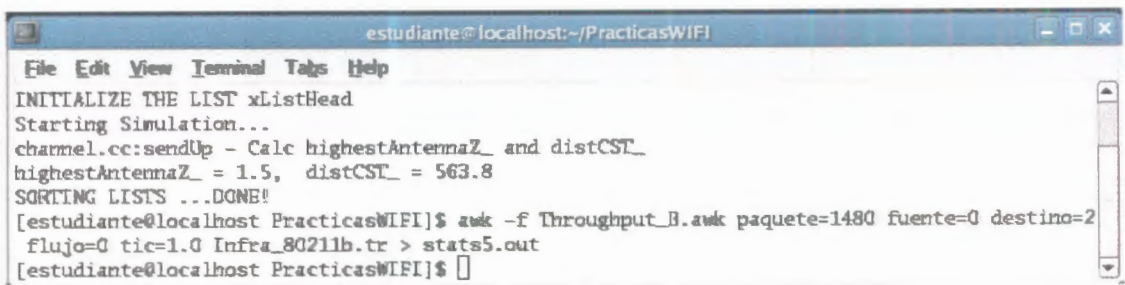


Para una mejor visualización de la simulación editamos la posición de los nodos dentro de la NAM para ello damos clic en la parte lateral izquierda en EDIT y separamos el nodo 0 del nodo 1, volvemos a dar clic en el mismo icono para desactivarlo de la siguiente forma:



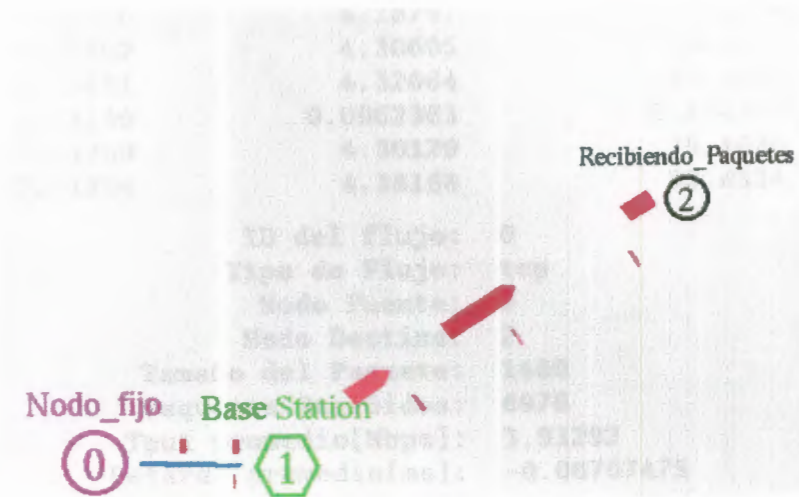
Observamos que la simulación consta de tres nodos el los cuales dos son nodos fijos y un nodo es móvil, los nodos son numerados de nodo 0 al nodo 2 respectivamente, ubicados en forma jerárquica en un ambiente inalámbrico de 90 x 90m en donde las coordenadas son verificadas dando clic sobre el nodo de interés. Los nodos fijos están enlazados con una red cableada Ethernet de 100Mbps. Dentro de algún tiempo comienza la transferencia de paquetes entre dos nodos. Tenga en consideración los instantes de tiempo en el que se ejecuta un evento para contestar las preguntas que realizaremos.

Una vez terminada la simulación volvemos a la ventana del terminal escribimos la siguiente sentencia 'awk -f Throughput\_B.awk paquete=1480 fuente=0 destino=2 flujo=0 tic=1.0 Infra\_80211b.tr > stats5.out', luego nos dirigimos a Desktop->estudiante->Practicawifi y abrimos el archivo stats5.out.



Al finalizar esta simulación conteste las siguientes preguntas:

- ¿Entre que nodos se realiza la transferencia de paquetes e indique cual es el nodo fuente y cual el destino? La comunicación se realiza entre los nodos 0 y 2, siendo el nodo 0 el nodo fuente y el nodo 2 el destino.



- ¿Qué función está ejerciendo el nodo 1 en la simulación? El nodo 1 cumple la función de una Estación Base o también llamado Punto de Acceso.



- ¿Qué sucede cuando el nodo 2 está fuera de cobertura y en qué instante de tiempo se produce este evento? En el instante  $t = 17.16$  secs el nodo 2 está fuera de cobertura en ese instante deja de transmitir y comienza a descartar paquetes.

Recibiendo Paquetes:



- ¿Cuál es el throughput promedio de la simulación y qué sucede con el throughput cuando el nodo 2 está fuera de cobertura? El Throughput promedio de la simulación es de 3.91292 Mbps y cuando el nodo 2 está fuera de cobertura este se hace 0 como se muestra.

15.0300	4.18747	38.0679
16.0402	4.30605	39.1459
17.0431	4.32064	39.2786
20.1189	0.0962363	0.874875
21.1209	4.30129	39.1026
22.1234	4.38168	39.8334

```

ID del flujo: 0
Tipo de Flujo: tcp
Nodo Fuente: 0
Nodo Destino: 2
Tamaño del Paquete: 1480
Paquetes Recibidos: 6976
Tput promedio[Mbps]: 3.91292
Retardo promedio[ms]: -0.00707475

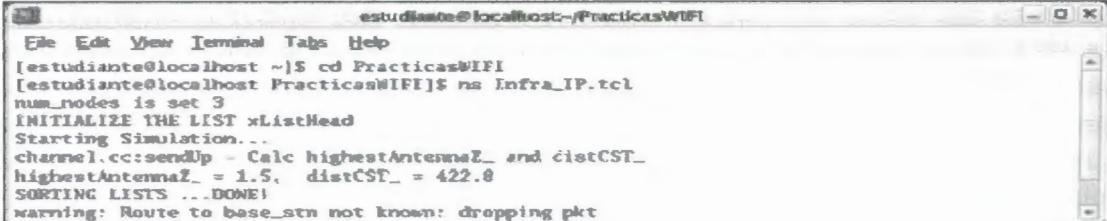
```

## Paso 2.2: Redes IEEE 802.11g

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

➤ *ns Infra\_IP.tcl*

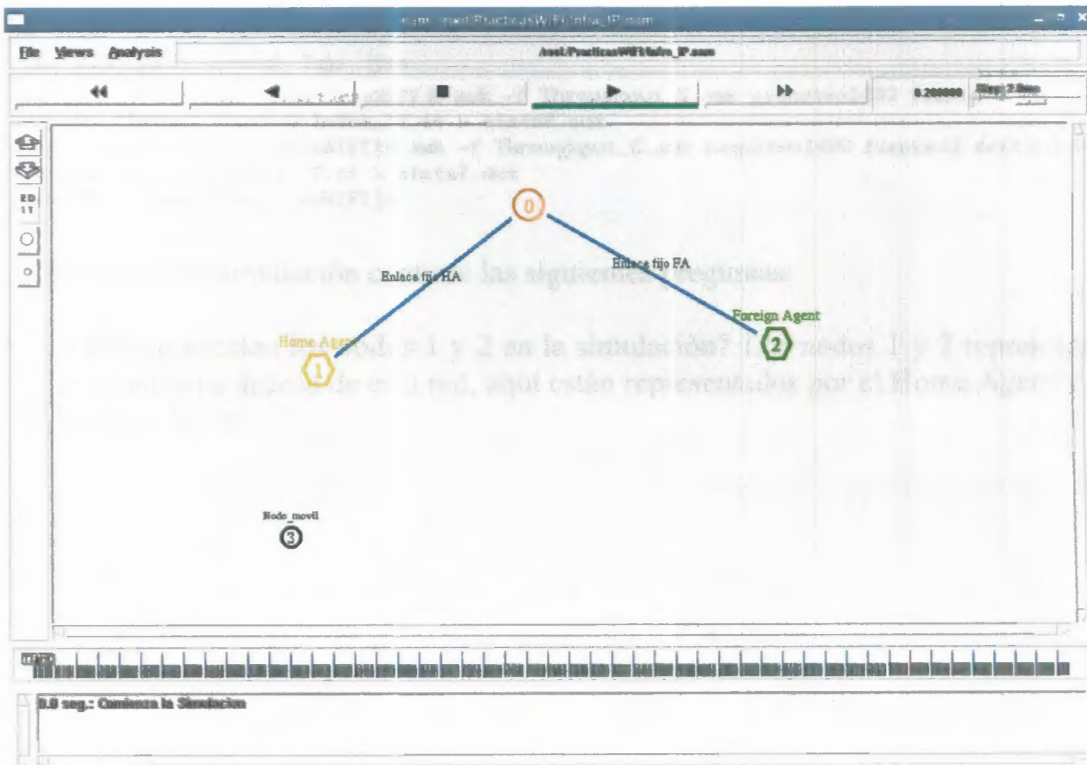
Nuestra simulación se encuentra guardado en la carpeta *PracticasWIFI* con el nombre *Infra\_IP.tcl* y a su vez va creando la interfaz gráfica NAM y presionamos *play* como se muestra.



```

estudiante@localhost:~/PracticasWIFI
File Edit View Terminal Tabs Help
[estudiante@localhost ~]$ cd PracticasWIFI
[estudiante@localhost PracticasWIFI]$ ns Infra_IP.tcl
num_nodes is set 3
INITIALIZE THE LIST xListHead
Starting Simulation...
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 422.0
SORTING LISTS ...DONE!
warning: Route to base_stn not known: dropping pkt

```



Podemos ver que la simulación consta de cuatro nodos en los cuales tres nodos son fijos, y un nodo es móvil, ubicados en forma jerárquica en un entorno inalámbrico. Aquí volvemos a editar la posición de los nodos fijos de la misma forma que anteriormente se realizó para una mejor visualización para ello damos clic en *Edit* y movemos los nodos de la siguiente forma: el nodo 0 en las coordenadas (80, 130), el nodo 1 en (15, 70) y el nodo 2 en (180,70), esto datos son aproximados. Esta simulación explica el evento de MOBILE IP que consiste en el intercambio de una estación base hacia otra dependiendo de la cobertura, la densidad de trafico, etc. Tenga en consideración los instantes de tiempo en el que se ejecuta un evento para contestar las preguntas que realizaremos.

Una vez terminada la simulación volvemos a la ventana del terminal escribimos la siguiente sentencia `'awk -f Throughput_G.awk paquete=1480 fuente=0 destino=3 flujo=1 tic=1.0 Infra_IP.tr > stats6.out'`, y la siguiente sentencia para la otra comunicación `'awk -f Throughput_G.awk paquete=1480 fuente=3 destino=0 flujo=0 tic=1.0 Infra_IP.tr > stats7.out'` luego nos dirigimos a *Desktop->estudiante->PracticasWIFI* y abrimos el archivo **stats6.out** y **stats7.out**.



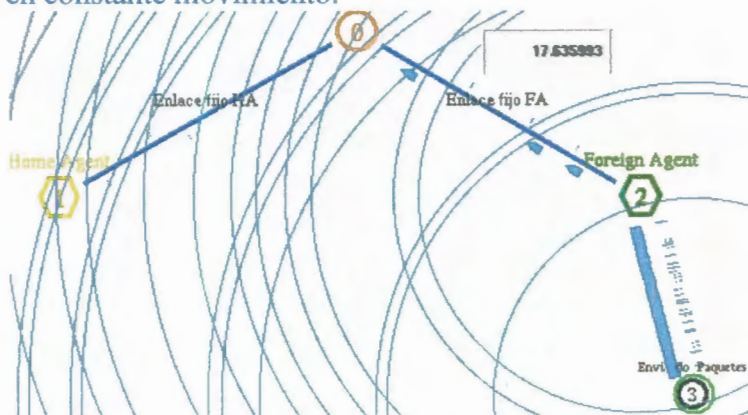
```
estudiante@localhost:~/Practicas/WIFI
File Edit View Terminal Tabs Help
[estudiante@localhost PracticasWIFI]# awk -f Throughput_G.awk paquete=1480 fuente=0 de
stino=3 flujo=1 tic=1.0 Infra_IP.tr > stats6.out
[estudiante@localhost PracticasWIFI]# awk -f Throughput_G.awk paquete=1480 fuente=3 destino=0
flujo=0 tic=1.0 Infra_IP.tr > stats7.out
[root@localhost PracticasWIFI]#
```

Al finalizar esta simulación conteste las siguientes preguntas:

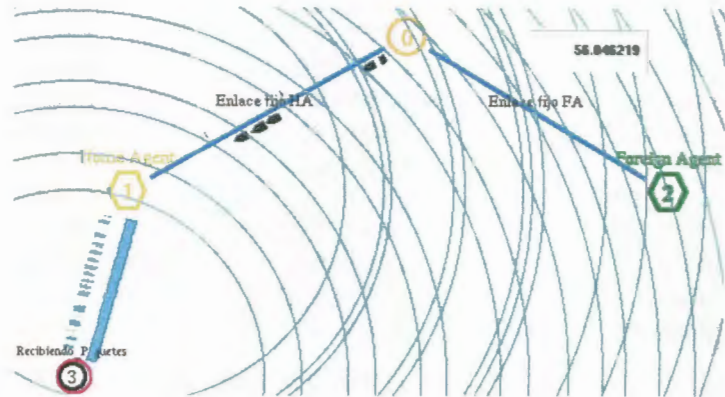
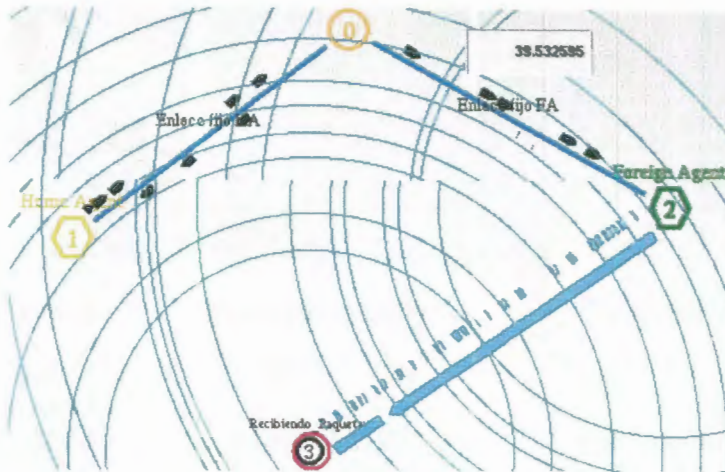
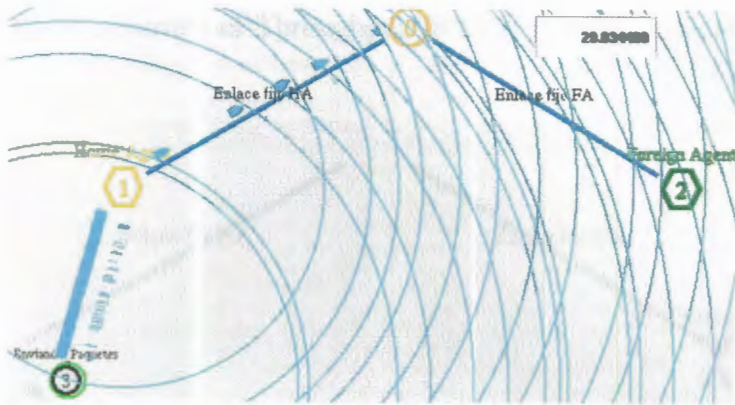
- ¿Qué representan los nodos 1 y 2 en la simulación? Los nodos 1 y 2 representan los puntos de acceso de esta red, aquí están representados por el Home Agent y el Foreign Agent.



- ¿En que instantes de tiempo se realiza el traspaso de AP? En el instante  $t = 17.63$  secs,  $t = 29.03$  secs, el  $t = 39.53$  secs, y  $t = 56.04$  secs, debido a que el nodo se encuentra en constante movimiento.

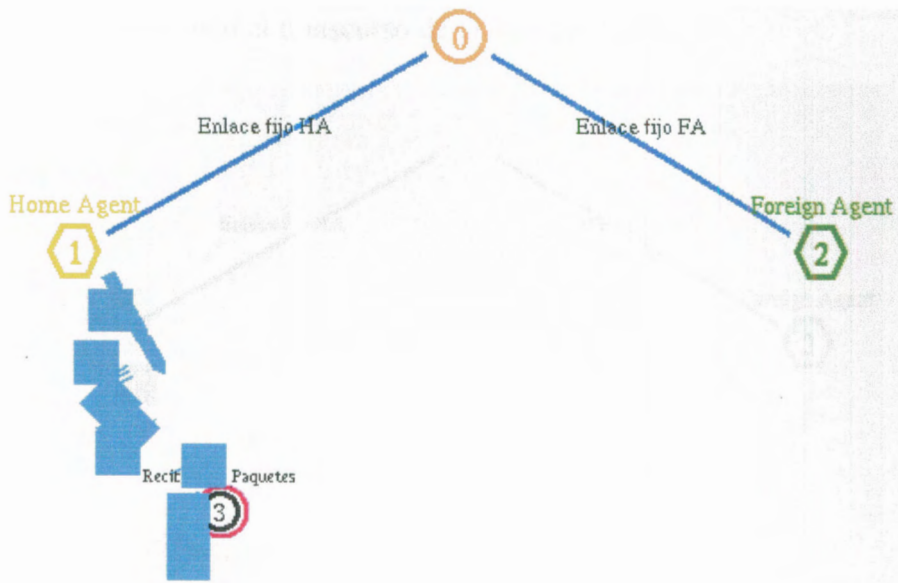






- ¿Cuál es el throughput promedio del archivo stats6 y que sucede con el throughput 3 esta fuera de cobertura? Cuando el nodo 3 esta fuera de cobertura se pierden paquetes pero después de un instante el nodo 3 se mueve nuevamente hasta una distancia que tiene cobertura y reestablece la comunicación entre los

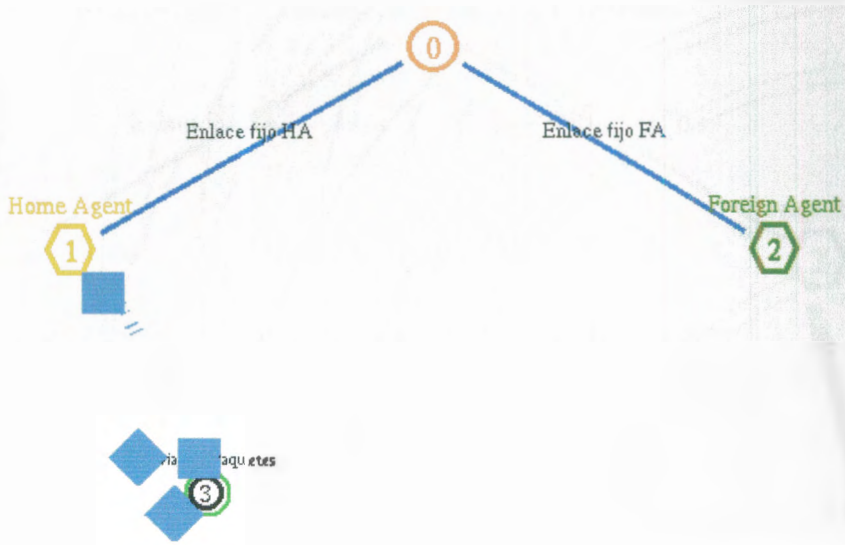
nodos. Como ya sabemos el Throughput se hace 0 en el instante que pierde la conexión.



Tiempo	Throughput [Mbps]	Eficiencia [%]
1.00034	0	0
32.0068	0	0
33.0068	0.000358718	0.000664292
34.0072	21.2314	39.3174
35.0075	21.0804	39.0377
36.0077	21.1421	39.152
37.0082	21.3852	39.6023
38.0083	21.4048	39.6385
44.4855	0.877408	1.62483
45.4856	18.9067	35.0124
46.4865	20.311	37.613
47.4866	21.76	40.2964
48.4868	20.7996	38.5177
49.4868	20.7552	38.4356
55.9567	0.812521	1.50467
56.9567	20.7318	38.3922

ID del flujo: 1  
 Tipo de Flujo: tcp  
 Nodo Fuente: 0  
 Nodo Destino: 3  
 Tamaño del Paquete: 1480  
 Paquetes Recibidos: 20408  
 Tput promedio[Mbps]: 16.5142  
 Retardo promedio[ms]: -0.000501504

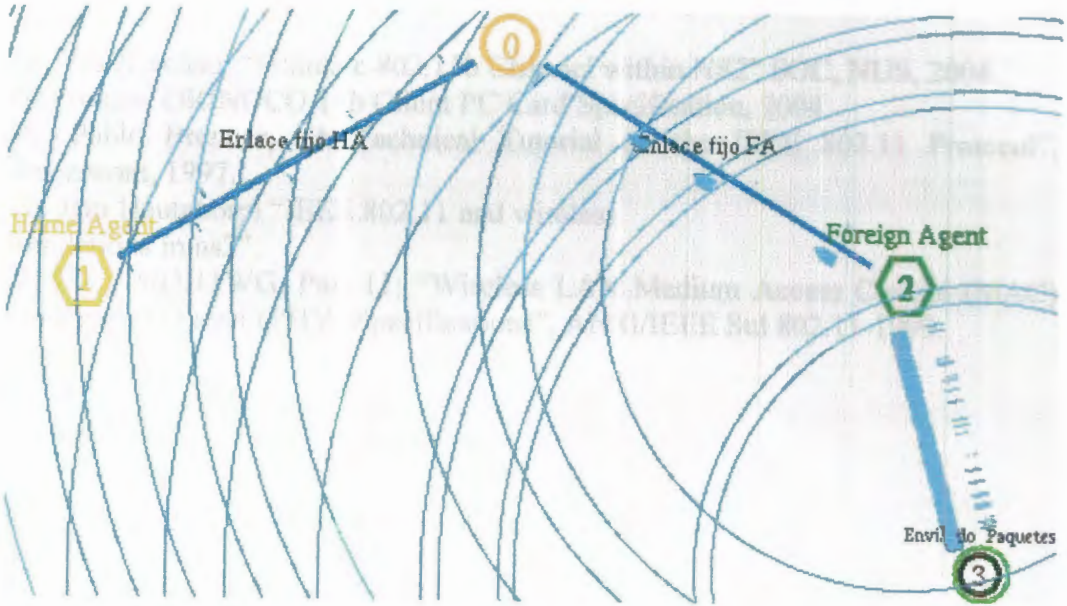
- ¿Cuál es el throughput promedio del archivo stats7 y que sucede con el throughput 3 esta fuera de cobertura? El throughput promedio decae cuando el nodo 3 esta fuera de cobertura es porque se hace cero y el cálculo del throughput se realiza durante todo el transcurso de la comunicación.



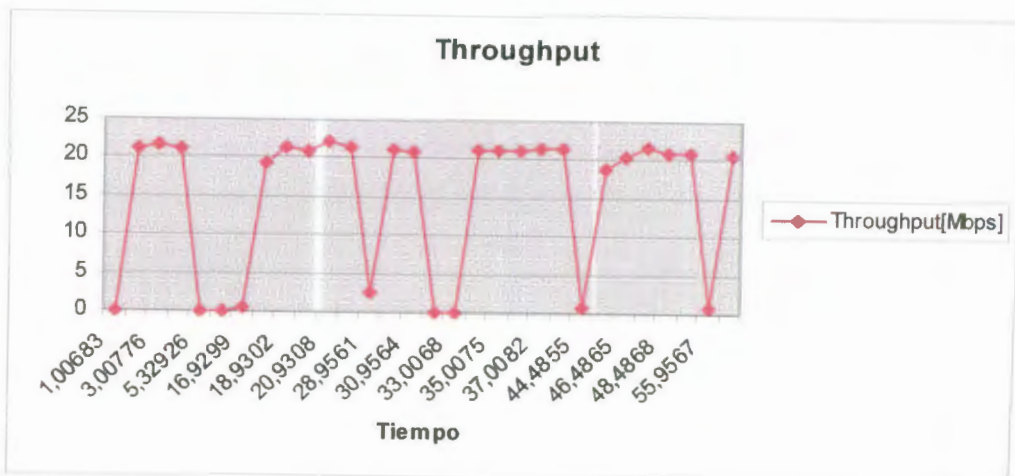
Tiempo	Throughput[Mbps]	Eficiencia[%]
1.00683	0.0117636	0.0217845
2.00751	21.2504	39.3525
3.00776	21.5433	39.8951
4.0082	21.137	39.1425
5.32926	0	0
15.9299	0	0
16.9299	0.529615	0.980768
17.93	19.321	35.7797
18.9302	21.4487	39.7197
19.9307	20.8644	38.6378
20.9308	22.1981	41.1075
21.9314	21.3576	39.551
28.9561	2.49114	4.61322
29.9561	21.1579	39.1814
30.9564	20.8447	38.6012

ID del flujo: 0  
 Tipo de Flujo: tcp  
 Nodo Fuente: 3  
 Nodo Destino: 0  
 Tamaño del Paquete: 1480  
 Paquetes Recibidos: 19980  
 Tput promedio[Mbps]: 16.4735  
 Retardo promedio[ms]: 0

- ¿A cual estación base se registra el nodo 3 después que esta fuera de cobertura? El nodo 3 se registra al nodo Foreign Agent cuando establece nuevamente la comunicación entre ellos puesto que el FA esta dentro de la cobertura que ahora necesita el nodo 3 para la transferencia de paquetes.



- Investigue en que consiste el procedimiento de Mobile IP Mobile IP es un protocolo diseñado para permitir a los usuarios de dispositivos móviles, cambiarse de de una red a otra manteniendo la misma dirección IP. este protocolo es muy utilizado en redes wireless ya que permite a un nodo cambiar de punto de acceso manteniendo el transporte y conexión de alto nivel mientras se mueve.
- Realice una grafica en Excel que represente el throughput a lo largo de la simulación tanto del archivo stats6 como del stats7





## BIBLIOGRAFÍA:

- "The *ns* Manual (formerly *ns* Notes and Documentation)1", The VINT Project , UC Berkeley, LBL, USC/ISI, and Xerox PARC, Editor Kevin Fall [kfall@ee.lbl.gov](mailto:kfall@ee.lbl.gov), Editor Kannan Varadhan [kannan@catarina.usc.edu](mailto:kannan@catarina.usc.edu) ,January 8, 2003
- [2]. Wu Xiuchao, "Simulate 802.11b Channel within NS2" SOC, NUS, 2004.
- [3] Proxim. ORINOCO 11b Client PC Card Specification, 2004.
- [4]. Pablo Brenner, "A Technical Tutorial on the IEEE 802.11 Protocol", Breezecom, 1997.
- [5]. Jani Hautakorpi "IEEE 802.11 and wireless simulations in ns2".
- [7] IEEE 802.11WG, Part 11; "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", ANSI/IEEE Std 802.11-1999.



## BIBLIOGRAFÍA

1. El estándar IEEE 802.11 Wireless LAN por Francisco López Ortiz – 2002 <http://greco.dit.upm.es/~david/TAR/trabajos2002/08-802.11-Francisco-Lopez-Ortiz-res.pdf>.
2. A Technical Tutorial on the IEEE 802.11 Protocol – By Pablo Brenner – 18 de Julio de 1996.
3. El Lenguaje de Programacion AWK/GAWK – Jesús Alberto Vidal Cortes – Madrid, Febrero de 2002.
4. White Paper IEEE 802.11g – Broadcom – 07/02/03.
5. Simulate 802.11b Channel within NS2 – Wu Xiuchao, SOC, NUS - [wuxiucha@comp.nus.edu.sg](mailto:wuxiucha@comp.nus.edu.sg).
6. Aironet Wireless LAN Fundamentals – Cisco Systems – 2003.
7. The ns Manual (formerly ns Notes and Documentation) – The VINT Project – Junio 29 del 2007.
8. IEEE 802.11g Explained – Jim Zyren – Diciembre 6 del 2001.
9. IEEE Standards Board, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” IEEE Std 802.11-1997, Nov 1997.