

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

**PROYECTO DE GRADO PARA LA OBTENCIÓN DEL TÍTULO
DE INGENIERÍA**

**“ANÁLISIS DEL DESEMPEÑO DE UNA RED CON TECNOLOGÍA
WI-FI PARA LARGAS DISTANCIAS EN LA REGIÓN COSTA DEL
ECUADOR.”**

REALIZADO POR:

**ISABEL E. CANO D.
FERNANDO D. ALMEIDA G.**

**SANGOLQUÍ – ECUADOR
2012**

CERTIFICACIÓN

Se certifica que la Srta. Isabel Estefania Cano Dávila y el Sr. Fernando Darío Almeida García, desarrollaron y finalizaron el proyecto en su totalidad bajo nuestra dirección.

Ing. Román Lara
DIRECTOR

Dr. Gonzalo Olmedo
CODIRECTOR

RESUMEN

En el siguiente trabajo se documenta el estudio del desempeño de una red WiLD (*Wireless Fidelity for Long Distance*) en la región costa del país. Con el estudio propuesto se desea realizar una propuesta para que las redes inalámbricas que se manejan bajo el estándar 802.11 puedan brindar una opción de bajo costo y que tenga los mismos beneficios pero con mayor alcance.

La meta fue probar la tecnología a una distancia mínima de 50 Km, para lo cual se realizaron pruebas en dos enlaces en sectores con características topográficas diferentes, los lugares son: el primero entre el cerro Salinas y el cerro Cabuyas con 75,5 Km de distancia entre ellos y el segundo entre el cerro Salinas y el cerro Ánimas con 66,7 Km de distancia entre ellos.

Para el análisis del comportamiento de la tecnología WiLD en clima cálido se trabajó en la planificación, modelamiento, simulación en *ns-2* y Radio Mobile, implementación y prueba piloto de las redes, para así comparar los diferentes requerimientos en cada zona tomando en cuenta un enlace sobre mar y otro sobre tierra.

Se realizaron pruebas de inyección de tráfico mediante el *software*, D-ITG (*Distributed Internet Traffic Generator*) que permite analizar el tráfico y la Calidad de Servicio (QoS) que posee la red a través de gráficas de *throughput*, *jitter*, *delay* y pérdida de paquetes.

DEDICATORIA

Quiero dedicar este trabajo a mi familia que ha estado siempre conmigo apoyándome en todo y ha soportado mis mal genios. Espero con este nuevo avance en mi vida, retribuir en algo todos los esfuerzos que han hecho por mí.

También quiero dedicar este trabajo a mi hermana Nicole y a mi prima Nathy, las quiero con todo mi corazón, espero que les sirva de ejemplo para su vida, y sé que muy pronto las veré en la misma situación.

Isabel Cano

DEDICATORIA

Dedico este proyecto de tesis a Dios, a mi familia y en especial a mis padres. A Dios porque ha estado conmigo a cada paso que doy, cuidándome y dándome fortaleza para continuar, a mis padres, quienes a lo largo de mi vida han velado por mi bienestar y educación siendo un pilar fundamental en la realización de este proyecto y en la persona que soy ahora. Depositando todo su apoyo y entera confianza en cada reto que se me presentaba. También dedico este proyecto a las personas que tienen un gran significado en mi vida, que sin ellos, este proyecto no hubiese podido ser. Los amo con mi vida.

Fernando Almeida

AGRADECIMIENTO

Principalmente a Dios que me ha dado tantas cosas que agradecer y tan pocas que pedir, y porque a pesar de que me he equivocado tantas veces, sigue a mi lado, me perdona y me muestra el camino correcto.

A mi ángel en el cielo que me ha enseñado el amor más grande que se puede sentir, y me mantiene luchando día a día. Y a mí ángel en la tierra que me recuerda el milagro de la vida. A mi compañero de tesis.

Y un agradecimiento especial al comando conjunto que nos apoyó y brindó todas las facilidades para la ejecución del proyecto.

AGRADECIMIENTO

Agradezco a Dios por haberme guiado en todo el transcurso de mi vida, a cada uno de los que son parte de mi familia, a mi padres, a mis abuelos, a mis hermanos y a todos mis tíos que siempre me han dado la fuerza y el apoyo incondicional que me han ayudado y llevado hasta donde estoy ahora. También agradezco a mi compañera de tesis, a mis ingenieros que son y fueron la base para que todo esto sea posible.

Fernando Almeida

PROLOGO

Debido a que hoy en día, las comunicaciones son esenciales en la vida diaria, es importante que toda la población tenga acceso a diversos medios para interactuar con el mundo. Uno de los más comunes e importantes actualmente es la Internet, que avanza a pasos agigantados, pero no es utilizada por toda la población, sino que es notable la brecha digital que existe entre las zonas mayormente pobladas y urbanas y las zonas rurales y menos pobladas. Una de las principales razones para este problema, es que a las grandes empresas privadas no les resulta conveniente implementar infraestructura para tecnologías de comunicación, en sitios de baja densidad poblacional.

En el Ecuador, se ha propuesto eliminar al máximo esta desigualdad digital, por lo que se trabaja en mecanismos para llegar a la población menos favorecida; A este documento le corresponde el análisis de la tecnología en la región costa del país.

Durante el desarrollo de este trabajo, se puede observar investigaciones, mecanismos, experiencias, análisis y recomendaciones, que pretenden continuar los estudios de la tecnología no estandarizada WiLD (*Wireless Fidelity for Long Distance*), que es la modificación del estándar IEEE 802.11 para largas distancias, en el orden de las decenas de Km. Óptima para terminar con el problema de costos de infraestructura para redes actuales como fibra óptica, satélites, etc.

INDICE DE CONTENIDOS

Capítulo I PRESENTACIÓN

1.1 INTRODUCCIÓN.....	21
1.2 MOTIVACIÓN.....	22
1.3 ALCANCE	23
1.4 OBJETIVOS.....	24
1.4.1 General.....	24
1.4.2 Específicos	24
1.5 ESTRUCTURA.....	25

Capítulo II MARCO TEÓRICO

2.1 ORIGEN Y EVOLUCIÓN.....	27
2.1.1 Estándares de la IEEE.....	28
2.1.2 Extensiones de Estándares Inalámbricos.	29
2.2 ESTRUCTURA Y DESCRIPCIÓN DE LAS CAPAS DEL ESTÁNDAR IEEE 802.11	32
2.2.1 Capa Física	32
• FHSS: Frequency Hopping Spread Spectrum.....	33
• <i>DSSS: DirectSequence Spread Spectrum</i>	34
• <i>OFDM: Orthogonal Frequency Division Multiplexing</i>	36
2.2.2 Capa de Enlace	37
• Subcapa MAC.....	38
• Subcapa LLC.....	42
2.3 AJUSTE DE PARÁMETROS PARA WILD.....	43
2.3.1 Límites impuestos por la capa física	43
2.3.2 Límites impuestos por la capa MAC.....	43
2.3.2 Cambiando el ACK y el <i>Slottime</i>	44
2.4 ESTADO DEL ARTE	45

Capítulo III MATERIALES Y MÉTODOS

3.1 MATERIALES.....	48
3.1.1 Selección de la tarjeta de red.....	48
Soekris Engineering.....	51
ADI Engineering.....	51
3.1.2 Tarjeta ALIX.....	52
• Características de la tarjeta Alix.....	52
• Partes de la tarjeta Alix.....	54
3.1.3 Tarjetas de radio inalámbricas	55
• SR2. Super Range 2.....	56
• SR5. Super Range 5.....	57
• XR2. XtremeRange2.....	58
• XR5. XtremeRange5.....	58
• R52H.....	59
• EMP-8602+S.....	59
3.1.4 Sistema Operativo Linux Voyage.....	60
3.1.5 Herramientas de simulación	61
• Herramientas de simulación para el perfil del terreno.....	62
• Radio Mobile.....	62
• Sirenet.....	63
3.1.5 Definición de la herramienta de simulación para el perfil del enlace.....	63
• Herramientas de simulación para el modelamiento de la WiLD.....	64
• <i>ns-2</i> (Network Simulator 2).....	64
• <i>ns-3</i> (Network Simulator 3).....	66
• NCTUns.....	68
3.1.6 Definición del software para el modelamiento la red WiLD.....	69
• <i>Trace Graph</i>	71
• <i>Trace Converter</i>	72
3.1.7 Herramientas para inyección de tráfico.....	73
• iPERF y jPERF.....	73
• <i>D-ITG</i>	74
3.1.8 Selección de la herramienta para inyección de tráfico.....	76

3.2 MÉTODOS.....	77
3.2.1 Configuración De Equipos	77
• Formateo de la unidad Compact Flash.....	77
• Instalación del sistema operativo Voyage Linux en la Compact Flash.....	80
• Configuración de interfaces en la tarjeta Alix.....	86
• Cambio de parámetros en la capa MAC para larga distancia en la tarjeta Alix	92

Capítulo IV DISEÑO E IMPLEMENTACIÓN DE LA RED

4.1 PLANIFICACIÓN DE LA RED	94
4.1.1 Estudio de los lugares donde se implementaran los enlaces	94
• Enlace Salinas – Ánimas.....	94
• Enlace Salinas – Cabuyas	95
4.1.2 Diseño de la red.....	96
• Enlace Salinas – Animas.....	96
• Enlace Salinas – Cabuyas	99
4.1.3 Modelo de propagación de Longley – Rice (ITS Irregular Terrain Model)	100
4.1.4 Cálculo de Propagación	104
4.1.5 Potencia de transmisión de las tarjetas	105
4.1.6 Longitud y calidad del cable coaxial.....	105
4.1.7 Distancia entre antenas.....	105
4.1.8 Nivel de Recepción	106
• Enlace Salinas – Animas.....	107
• Enlace Salinas – Cabuyas	107
4.1.9 Sensibilidad	108
4.1.10 Alineación de las antenas	109
• Enlace Salinas – Animas.....	110
• Enlace Salinas – Cabuyas	112
4.1.11 Radio de la primera zona de Fresnel	114
• Enlace Salinas – Animas.....	115
• Enlace Salinas – Cabuyas	115
4.1.12 Margen Sobre Obstáculo	115
• Enlace Salinas – Animas.....	116

• Enlace Salinas – Cabuyas	117
4.1.13 Configuración de interfaces para WiLD	118
4.1.14 Simulación con el Software <i>ns-2</i>	122
• Modelo de Propagación <i>Two-Ray Ground Reflection</i>	126
Enlace Salinas – Animas.....	128
Enlace Salinas – Cabuyas	128
• Modelo de Propagación <i>Shadowing</i>	129
Enlace Salinas – Animas.....	131
Enlace Salinas – Cabuyas	132
4.1.15 Simulación con <i>ns-2</i> Mejorado (TeNS).....	133
4.1.16 Trace Converter.....	138
4.1.17 Trace Graph.....	139
4.1.18 Simulación del Enlace Salinas –Ánimas.....	140
• Simulación del enlace con el Protocolo TCP.....	141
a) <i>Throughput</i> de la Red	142
b) <i>Throughput</i> de la Red en Función al Número de Nodos.....	143
c) <i>Throughput</i> Real de la Red en Función al Número de Nodos	143
d) <i>Throughput</i> Normalizado de la Red en Función al Número de Nodos	144
e) <i>Delay End-To-End</i>	146
f) <i>Delivery Ratio</i> de la Red.....	147
• Simulación del enlace con el Protocolo UDP	148
a) <i>Throughput</i> de la Red	148
b) <i>Throughput</i> de la Red en Función al Número de Nodos.....	148
c) <i>Throughput</i> Real de la Red en Función al Número de Nodos	149
d) <i>Throughput</i> Normalizado de la Red en Función al Número de Nodos	149
e) Análisis del <i>Delay End-To-End</i>	151
f) Análisis de <i>Delivery Ratio</i> de la Red	152
4.1.19 Simulación del Enlace Salinas –Cabuyas	152
• Simulación del enlace con el Protocolo TCP	154
a) Análisis del <i>Throughput</i> de la Red.....	154
b) <i>Throughput</i> de la Red en Función al Número de Nodos.....	154
c) <i>Throughput</i> Real de la Red en Función al Número de Nodos	155
d) <i>Throughput</i> Normalizado de la Red en Función al Número de Nodos	155
e) Análisis del <i>Delay End-To-End</i>	157

f) Análisis de <i>Delivery Ratio</i> de la Red	158
• Simulación del enlace con el Protocolo UDP.....	158
a) Análisis del <i>Throughput</i> de la Red.....	159
b) <i>Throughput</i> de la Red en Función al Número de Nodos.....	159
c) <i>Throughput</i> Real de la Red en Función al Número de Nodos	160
d) <i>Throughput</i> Normalizado de la Red en Función al Número de Nodos.	160
e) Análisis del <i>Delay End-To-End</i>	162
f) Análisis de <i>Delivery Ratio</i> de la Red	162
4.2 IMPLEMENTACIÓN DE LA RED.....	163
4.2.1 Equipos y materiales para los enlaces.....	164
• <i>Power station 2-EXT</i> Ubiquiti.....	165
4.2.1 Enlace Salinas – Animas.....	166
• Alineación de las antenas	168
4.2.2 Enlace Salinas – Cabuyas	169
• Alineación de las antenas	170
4.2.3 Inyección de tráfico.	170
• Enlace Salinas – Animas.....	175
• Enlace Salinas - Cabuyas	176

Capítulo V ANÁLISIS DE RESULTADOS

5.1 ANÁLISIS DE LOS RESULTADOS OBTENIDOS EN EL SIMULADOR <i>ns-2</i>	178
5.1.1 Análisis de la Cantidad de Paquetes	178
5.1.2 Desempeño del <i>Throughput</i> en relación a la distancia.	179
5.2 ANÁLISIS DE LAS MEDICIONES OBTENIDAS DURANTE LA IMPLEMENTACIÓN DE LOS ENLACES.....	182
5.2.1 Delay.....	182
• Enlace Salinas - Animas.	183
• Enlace Salinas – Cabuyas	184
5.2.2 Jitter	185
• Enlace Salinas - Animas.	186
• Enlace Salinas – Cabuyas	187
5.2.3 Paquetes Perdidos	188

• Enlace Salinas - Animas.	188
• Enlace Salinas – Cabuyas	190
5.2.4 Throughput	191
• Enlace Salinas - Animas.	192
• Enlace Salinas – Cabuyas	193
5.3 ANÁLISIS COMPARTIVO: SIMULACIÓN VS MEDICIONES	195

Capítulo VI CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES	198
6.2 RECOMENDACIONES	201

INDICE DE FIGURAS

CAPITULO II

Figura 2.1 Capas definidas por 802.11	32
Figura 2.2 Capa física.....	33
Figura 2.3 FHSS.....	34
Figura 2.4 DSSS.....	35
Figura 2.5 Secuencia de código Barker.....	35
Figura 2.6 OFDM.....	36
Figura 2.7 Subcapas de la capa MAC.	37
Figura 2.8 Información organizada en tramas.....	37
Figura 2.9 Envío de ACK.....	38
Figura 2.10 Proceso del acceso CSMA/CA.	40

CAPITULO III

Figura 3.1 Vista superior e inferior, <i>Router Board 333</i>	49
Figura 3.2 <i>Pronghorn SBC 250</i>	50
Figura 3.3 Tarjeta Net4826-30	50
Figura 3.4 Parte superior de la placa 3D3.....	54
Figura 3.5 Parte inferior de la placa 3D3.....	55
Figura 3.6 SR2.....	56
Figura 3.7 SR5.....	57
Figura 3.8 XR2.....	58
Figura 3.9 XR5.....	58
Figura.3.10 R52H.	59
Figura 3.11 EMP-8602+S.	60
Figura 3.12 Arquitectura <i>ns-2</i>	65
Figura 3.13 Esquema de simulación <i>ns-2</i>	66
Figura 3.14 Arquitectura <i>ns-3</i>	67
Figura 3.15 Arquitectura NCTUns.	69
Figura. 3.16 Conversión del archivo .tr.....	73
Figura. 3.17 Arquitectura D-ITG.....	75
Figura. 3.18 Tarjeta compact flash y puerto SCSI-IDC 50 pines.....	77
Figura 3.19 Resultado del comando <i>dmesg</i>	78
Figura 3.20 Desmonte de la Compact Flash.....	78
Figura 3.21 Formateo de la Compact Flash como <i>ext2</i>	79
Figura 3.22. Desactivación de la revisión del disco.....	79
Figura 3.23 Partición con Linux correcta.	80
Figura 3.24 Ingreso a la carpeta <i>Vogaye</i> y creación de directorios.	81
Figura 3.25 Menú de instalación Linux <i>Voyage 0.6.5</i>	81
Figura 3.26. Selección de la tarjeta <i>Alix</i>	82
Figura 3.27. Selección del disco Compact Flash.	83
Figura 3.28 Directorio donde se montará el Sistema Operativo.	83
Figura 3.29 Selección del sistema de arranque.	84

Figura 3.30 Elección y configuración del puerto serial.....	84
Figura 3.31 Detalles de configuración.....	85
Figura 3.32 Sistema Operativo Linux añadido.....	86
Figura 3.33 PuTTY	87
Figura 3.34 Acceso correcto a la Compact Flash.	87
Figura 3.35 Configuración del puerto eth0 Alix master.	88
Figura 3.36 Configuración del puerto eth0 Alix esclavo.....	88
Figura 3.37 Configuración del puerto ath0 Alix master.	89
Figura 3.38 Configuración del puerto ath1 Alix esclavo.....	89
Figura 3.39 Lista de potencias Master.....	90
Figura 3.40 Lista de potencias Esclavo.....	91
Figura 3.41 Lista de potencias Esclavo.....	92
Figura 3.42 Lista de potencias Esclavo.....	92
Figura 3.43 Configuración de parámetros para larga distancia.....	93

CAPITULO IV

Figura 4.1. Vista satelital Salinas – Animas.....	95
Figura 4.2. Vista satelital Salinas - Cabuyas.	96
Figura 4.3. Esquema del enlace Salinas - Animas.....	97
Figura 4.4. Enlace Salinas - Ánimas en Radio Mobile.	98
Figura 4.5. Perfil de radio Salinas - Ánimas.....	98
Figura 4.6 Esquema del enlace Salinas Cabuyas.	99
Figura 4.7 Enlace Salinas - Cabuyas en Radio Mobile.....	99
Figura 4.8 Perfil de radio Salinas - Cabuyas.....	100
Figura 4.9 Modelo de dos rayos sobre tierra.....	103
Figura 4.10 Parámetros para enlace sobre mar según Longley - Rice.....	104
Figura 4.11 Alturas enlace Salinas – Ánimas.....	110
Figura 4.12 Alturas enlace Salinas – Cabuyas.....	112
Figura 4.13 Datos del obstáculo más grande para el enlace Salinas- Animas.....	116
Figura 4.14 Datos del obstáculo más grande para el enlace Salinas- Cabuyas.....	117
Figura 4.15. <i>ACKTimeout</i>	118
Figura 4.16 Compilación del archivo <i>threshold.cc</i>	124
Figura 4.17 Ingreso de datos y obtención del valor de <i>Threshold</i>	127
Figura 4.18 <i>Threshold</i> Salinas – Animas <i>TwoRayGround</i>	128
Figura 4.19 <i>Threshold</i> Salinas – Cabuyas <i>TwoRayGround</i>	129
Figura 4.20 <i>Threshold</i> por defecto Con <i>Shadowing</i>	131
Figura 4.21 <i>Threshold</i> Salinas – Animas con <i>Shadowing</i>	131
Figura 4.22 <i>Threshold</i> Salinas – Cabuyas con <i>Shadowing</i>	132
Figura 4.23 Conversión del archivo <i>simple1.tr</i>	139
Figura 4.24 Creación del archivo <i>simple1_tg.tr</i>	139
Figura 4.25 Simulación NAM 802.11b, 2- nodos (C. Salinas – C. Ánimas).....	141
Figura 4.26 Throughput de bits recibidos TCP en Salinas - Animas.....	145
Figura 4.27 Throughput de bits reenviados TCP en Salinas - Animas.....	145
Figura 4.28 Throughput de bits caídos TCP en Salinas - Animas.....	146
Figura 4.29. Valores del Delay End-To-End TCP Salinas – Animas.....	146

Figura 4.30. Histograma de frecuencia del Delay End-To-End TCP Salinas - Animas.	147
Figura 4.31 Throughput de bits recibidos UDP en Salinas - Animas.....	150
Figura 4.32 Throughput de bits retransmitidos UDP en Salinas – Animas.....	150
Figura 4.33 Throughput de bits caídos UDP en Salinas - Animas.....	151
Figura 4.34 Valores del Delay End-To-End UDP Salinas - Animas.....	151
Figura 4.35 Valores del Delay End-To-End UDP Salinas - Animas.....	152
Figura 4.36 Simulación NAM 802.11b, 2- nodos (C. Salinas – C. Cabuyas).	153
Figura 4.37 Throughput de bits recibidos TCP en Salinas - Cabuyas.....	156
Figura 4.38 Throughput de bits reenviados TCP en Salinas - Cabuyas.	156
Figura 4.39 Throughput de bits caídos TCP en Salinas - Cabuyas.....	157
Figura 4.40. Valores del Delay End-To-End TCP Salinas – Cabuyas.....	157
Figura 4.41 Histograma de frecuencia del Delay End-To-End TCP Salinas - Cabuyas.	158
Figura 4.42 Throughput de bits recibidos UDP en Salinas - Cabuyas.	161
Figura 4.43 Throughput de bits reenviados UDP en Salinas - Cabuyas.....	161
Figura 4.44 Throughput de bits caídos UDP en Salinas - Cabuyas.....	161
Figura 4.45 Valores del Delay End-To-End UDP Salinas - Cabuyas.	162
Figura 4.46 Histograma de frecuencia del Delay End-To-End UDP Salinas - Cabuyas.....	162
Figura 4.47 Cerro Salinas y torre utilizada.....	163
Figura 4.48 Caseta, Alimentación y comunicación.....	164
Figura 4.49 Power Station 2 EXT.	165
Figura 4.50 Cerro Ánimas y torre utilizada.....	166
Figura 4.51 Caja para exteriores de la placa Alix.....	167
Figura 4.52 Caja para exteriores de la placa Alix.....	167
Figura 4.53 Manejo del GPS sobre la antena.....	168
Figura 4.54 Power Station 2EXT, Cerro Animas.....	169
Figura 4.55 Cerro Cabuyas.....	169
Figura 4.56 Power Station 2EXT, cerro Cabuyas,	170
Figura 4.57 D-ITG, Definición de flujo en emisor.	171
Figura 4.58 D-ITG, Configuración de emisor.	172
Figura 4.59 D-ITG, Configuración de <i>Analyzer</i> en emisor.....	173
Figura 4.60. D-ITG, Configuración del equipo receptor.	174
Figura 4.61 D-ITG, Configuración Settings en receptor.....	174
Figura 4.62 Resultados de D-ITG con Protocolo TCP, Salinas - Animas.....	175
Figura 4.63 Resultados de D-ITG con Protocolo UDP, Salinas - Animas.....	176
Figura 4.64 Resultados de D-ITG con Protocolo TCP, Salinas - Cabuyas.....	176
Figura 4.65 Resultados de D-ITG con Protocolo UDP, Salinas - Cabuyas.	177

CAPITULO V

Figura 5.1 Análisis del Throughput vs Distancia con el Protocolo TCP.....	180
Figura 5.2 Análisis del Throughput vs Distancia con el Protocolo UDP.	181
Figura 5.3. Comparación de valores simulados entre TCP y UDP.	181
Figura 5.4 Delay en GNUPlot con TCP Salinas – Animas.	183
Figura 5.5 Delay en GNUPlot con UDP Salinas – Animas.....	184

Figura 5.6 Delay en GNUPlot con TCP Salinas – Cabuyas.....	185
Figura 5.7 Delay en GNUPlot con UDP Salinas – Cabuyas.....	185
Figura 5.8 Jitter en GNUPlot con TCP Salinas – Animas.	186
Figura 5.9 Jitter en GNUPlot con UDP Salinas – Animas.....	187
Figura 5.10 Jitter en GNUPlot con TCP Salinas – Cabuyas.	187
Figura 5.11 Jitter en GNUPlot con UDP Salinas – Cabuyas.	188
Figura 5.12 Paquetes Perdidos en D-ITG con TCP Salinas – Animas.....	189
Figura 5.13 Paquetes Perdidos en D-ITG con UDP Salinas – Animas.	189
Figura 5.14 Paquetes Perdidos en D-ITG con TCP Salinas – Cabuyas.	190
Figura 5.15 Paquetes Perdidos en D-ITG con UDP Salinas – Cabuyas.....	190
Figura 5.16 Throughput en <i>GNUPlot</i> con TCP Salinas - Animas.....	192
Figura 5.17 Throughput en GNUPlot con UDP Salinas - Animas.	193
Figura 5.18 Throughput en GNUPlot con TCP Salinas - Cabuyas.	194
Figura 5.18 Throughput en GNUPlot con UDP Salinas - Cabuyas.	194
Figura 5.20. Valores de throughput, simulados y medidos con el protocolo TCP.	195
Figura 5.21. Comparación de valores simulados y reales con el protocolo UDP.	196

INDICE DE TABLAS

Tabla 2.1. Otros estándares 802.11	31
Tabla 3.1 Comparación de placas para redes inalámbricas.	51
Tabla 3.2 Clasificación de tarjetas ALIX	52
Tabla 3.3 Tarjetas de red inalámbricas.....	56
Tabla 4.1 Coordenadas Salinas – Animas.....	95
Tabla 4.2 Coordenadas Salinas – Cabuyas.....	96
Tabla 4.3 Valores de Conductividad y Permitividad.	102
Tabla 4.4 Cálculo de propagación para los enlaces.....	106
Tabla 4.5 Parámetros de enlace Salinas - Animas.....	107
Tabla 4.6 Parámetros de enlace Salinas - Cabuyas.	108
Tabla 4.7 Valores de sensibilidad vs velocidad.....	108
Tabla 4.8 Verificación de viabilidad en los enlaces.	109
Tabla 4.9 Datos reales en cada nodo.	109
Tabla 4.10 Calculo de Variación de latitud y longitud para el primer enlace.	111
Tabla 4.11 Calculo de Variación de latitud y longitud para el segundo enlace.	113
Tabla 4.12 Datos resumidos el obstáculo para el enlace Salinas- Animas,	117
Tabla 4.13 Datos resumidos el obstáculo para el enlace Salinas- Cabuyas.....	118
Tabla 4.14 Valores teóricos de <i>ACKTimeout</i>	120
Tabla 4.15 Comparación de valores de <i>ACKTimeout</i> en la interfaz ath0.	121
Tabla 4.16 Comparación de valores de <i>ACKTimeout</i> en la interfaz ath1.	121
Tabla 4.17 Parámetros modificables para el umbral.	124
Tabla 4.18 Valores por defecto de <i>ns-2</i>	125
Tabla 4.19 Valores por defecto de <i>ns-2</i> para Salinas - Animas.....	126
Tabla 4.20 Valores por defecto de <i>ns-2</i> para Salinas - Animas.....	126
Tabla 4.21. Valores típicos para Exponente de Pérdidas de Propagación.....	130
Tabla 4.22 Valores típicos de la Desviación Estándar σ_{dB}	130
Tabla 4.23 Parámetros de simulación.....	137
Tabla 4.24 Parámetros Seteados en <i>ns-2</i> Mejorado para Salinas - Animas.....	140
Tabla 4.25 Datos TCP Salinas - Animas.	142
Tabla 4.26 Datos UDP Salinas - Animas.....	148
Tabla 4.27 Parámetros Seteados en <i>ns-2</i> Mejorado para Salinas - Cabuyas.	153
Tabla 4.28 Datos TCP Salinas - Cabuyas.....	154
Tabla 4.29 Datos UDP Salinas - Cabuyas	159
Tabla 4.30. Parámetros para el flujo del emisor.....	172
Tabla 4.31 Parámetros de configuración del emisor.	173
Tabla 4.32. Parámetros definidos en equipo receptor.....	175
Tabla 5.1 Valores de Throughput con el Protocolo TCP.....	179
Tabla 5.2 Valores de Throughput con el Protocolo UDP.	180

GLOSARIO

Wi- Fi.- *Wireless Fidelity*, Fidelidad Inalámbrica

ANCHO DE BANDA.- es la cantidad de información o de datos que se puede enviar a través de una conexión de red en un período de tiempo dado.

Punto de acceso (AP/PA).- Dispositivo que ejerce básicamente funciones de puente entre una red Ethernet cableada y una red con Wi-Fi .

WPA.- Acceso protegido Wi-Fi

IEEE.- Organización de profesionales de la informática con base en Estados Unidos.

CSMA.- Técnica de Acceso múltiple con detección de portadora.

AMD.- *Advanced Micro Devices*, Micro Dispositivos Avanzados

D-ITG.- Generador de Tráfico de Internet Distribuido

IP.- Protocolo de Internet

MAC.- Control de Acceso al medio

NAM.- Animador de red

TCP.- Protocolo de Control de Transmisión

UDP.- Protocolo de Datagrama de Usuario

CAPÍTULO I

PRESENTACIÓN

1.1 INTRODUCCIÓN

Actualmente, las redes de conexión inalámbrica se encuentran en desarrollo constante, y existe una diversidad de maneras de acceso a Internet. Lamentablemente no toda la población del Ecuador tiene acceso a este servicio. Con el fin de realizar un mejoramiento en el desarrollo de las redes inalámbricas que se manejan bajo el estándar IEEE 802.11 y brindar una opción que abarate costos y tenga los mismos beneficios que las tecnologías similares pero con mayor alcance. El siguiente proyecto de graduación estudia la tecnología no estandarizada WiLD (*Wireless Fidelity for Long Distance*) en la región costera del Ecuador.

De acuerdo a proyectos desarrollados anteriormente sobre esta tecnología, es posible llegar a tener un alcance promedio de 50 Km, también se conoce que el récord de distancia logrado en un enlace entre dos puntos es de 382 Km [1].

El mecanismo de la subcapa de acceso al medio del estándar IEEE 802.11, tiene como fundamento un acuse de recibo o ACK (*ACKnowledgement*) por cada paquete enviado en un determinado tiempo (*ACK-timeout*); estos tiempos previstos para la recepción del ACK son insuficientes para distancias de varias decenas de kilómetros, para adaptarlo a la nueva tecnología, se debe recurrir a la modificación de este tiempo y del *Slot Time* (tiempo entre tramas) y así conseguir una distancia

mayor que la establecida por el estándar. Este proceso se lo realiza mediante *software*.

Un trabajo previo de WiLD en la región sierra [2], realizado como proyecto de graduación en la Escuela Politécnica del Ejército, se utilizó el *software* de simulación *ns-2*, en su versión 2.34, con el cual se llegó a determinar un modelo del sistema en una distancia máxima de 30 Km.

1.2 MOTIVACIÓN

En los últimos años se ha podido observar un gran avance tecnológico en nuestro país, pero así mismo se observa que este avance no es equitativo. El estado mediante la Secretaría Nacional de Planificación y Desarrollo, *Semplades* y, a través del Plan Nacional del buen vivir 2009-2013 [3], diagnostica que “El acceso limitado a la información, restringe las posibilidades de formación y aprendizaje continuo y dificulta el desarrollo personal y colectivo de las capacidades”, siendo las zonas rurales las más afectadas, apenas un 7,2% de población rural usa internet y, solo una vez por semana [3].

Es por eso que el plan tiene como objetivo disminuir esta brecha digital, y proporcionar servicios como Internet en las zonas relativamente menos pobladas, donde, las soluciones tradicionales como cable o fibra óptica se encuentran fuera de alcance, pues los proveedores de servicios no están dispuestos a pagar por el despliegue de redes cableadas que requieren de costos muy elevados frente a la baja demanda de sus servicios en estas regiones.

El plan propone como estrategia la conectividad mediante las telecomunicaciones, razón que impulsa a la búsqueda de una nueva alternativa de interconexión, óptima y a bajo costo; y así, llegar a regiones donde la geografía y condiciones de terreno permitan el uso solo ciertas tecnologías, como la propuesta en este trabajo.

Desde el nacimiento de Wi-Fi se ha venido trabajando en el estudio del desarrollo de la tecnología pues a inicios ofrecía coberturas máximas de 1Km, por lo que se han realizado grandes avances con el fin de aumentar su alcance, debido a esto las aplicaciones domésticas, comerciales e industriales han crecido notablemente en todo el país. Actualmente se han realizado experimentos en distintos países para enlazar distintos puntos geográficamente distantes que no cuentan con opción a conectividad.

Las modificaciones realizadas al estándar IEEE 802.11 han permitido que esta tecnología aumente su cobertura, y, mediante el cambio de parámetros de configuración en la capa de enlace de datos MAC, es posible llegar con un enlace Wi-Fi de larga distancia a regiones donde las tecnologías cableadas resultarían costosas.

1.3 ALCANCE

Se realizará el estudio del desempeño de la red Wi-Fi con una cobertura mínima de 50 Km., Se realizarán dos enlaces en dos sectores con características topográficas diferentes en la región costa, para observar el comportamiento en cada uno.

El análisis se lo realizará mediante pruebas piloto, es decir se implementarán los enlaces y se desmontarán una vez concluido el trabajo.

Investigando la necesidad de obtener un bajo costo en la inversión del proyecto, se utilizarán, los equipos proporcionados por el Centro de Investigaciones de redes AD-HOC (CIRAD) como son las tarjetas Alix que serán aplicadas como *routers* inalámbricos, además de los equipos estandarizados y no estandarizados disponibles.

Se realizará el estudio necesario para determinar una arquitectura de red adecuada con el fin de obtener un mejor desempeño en la red inalámbrica WiLD. Mediante el software de simulación de redes *ns-2* se determinará como deberá trabajar idealmente la red.

Para el correcto análisis del comportamiento de la tecnología WiLD en clima cálido se trabajará en la planificación, modelamiento, simulación (mediante *radio mobile*), implementación y prueba piloto de la red, a una distancia mínima de 50 Km por cada enlace, para así comparar los diferentes requerimientos en cada zona. Un enlace sobre mar y otro sobre tierra.

Se realizará pruebas de inyección de tráfico mediante el *software*, de distribución gratuita, *D-ITG (Distributed Internet Traffic Generator)* que permite analizar el tráfico y la Calidad de Servicio QoS que posee la red a través de gráficas de *throughput, jitter, delay* y *pACKetloss*.

1.4 OBJETIVOS

1.4.1 General

Analizar el desempeño de una Red con Tecnología no estandarizada, WiLD en la Región Costa del Ecuador.

1.4.2 Específicos

- ✓ Determinar la factibilidad de los puntos para el enlace.
- ✓ Modificar los parámetros necesarios de la capa MAC del estándar IEEE 802.11 para lograr un enlace a larga distancia.
- ✓ Modelar el sistema de comunicaciones inalámbrico con características específicas de la región costa y en zonas topográficamente diferentes.

- ✓ Simular los enlaces en el software destinado *ns-2*.
- ✓ Implementar los enlaces punto a punto en la región costa.
- ✓ Analizar el desempeño de una red inalámbrica de conformidad con el estándar IEEE 802.11, a larga distancia.

1.5 ESTRUCTURA

El capítulo II de este proyecto presenta los fundamentos teóricos del estándar IEEE 802.11, características y propiedades de la Capa Física y la Capa MAC y las adecuaciones necesarias para un enlace de larga distancia transformando la tecnología estandarizada Wi-Fi en la tecnología no estandarizada WiLD, se describen las modificaciones en los tiempos de propagación como son; el ACK *timeout*, *clear to send* y *slot time* para enlaces de larga distancia.

En el Capítulo III se estudian los materiales disponibles y métodos utilizados para realizar los enlaces, el funcionamiento y configuración de la Tarjeta *Alix* para lograr un alcance de larga distancia, además el estudio de diferentes herramientas de simulación de redes; *ns-2*, *ns-3* y *NCTUns*, e inyección de tráfico, obteniendo así el más adecuado para el trabajo.

El Capítulo IV muestra la planificación e implementación de las redes, la descripción de parámetros para ingresar en las herramientas de simulación, las modificaciones en *ns-2* para alcanzar distancias de hasta 80 Km, el estudio del modelo de propagación adecuado, y la simulación propiamente de los 2 enlaces. Por otro lado se describe también el diseño de la red, los requisitos, para la implementación de las redes, y el proceso completo de implementación.

En el capítulo V se encuentra el análisis de resultados; las mediciones obtenidas durante la implementación de la red, y la comparación con los resultados de la simulación realizada en el capítulo IV.

Finalmente en el capítulo VI se presentan las conclusiones obtenidas una vez finalizado el trabajo y las recomendaciones para trabajos futuros.

CAPÍTULO II

FUNDAMENTO TEÓRICO

La especificación IEEE 802.11 [4], es un estándar internacional creado por el Instituto de Ingenieros Eléctricos y Electrónicos (*Institute of Electrical and Electronics Engineers – IEEE*), que define las características de una red de área local inalámbrica (WLAN). Wi-Fi (*Wireless Fidelity* o, Fidelidad inalámbrica), es el nombre de la certificación otorgada por la *Wi-Fi Alliance*¹, anteriormente *WECA* (*Wireless Ethernet Compatibility Alliance*), grupo que garantiza la compatibilidad entre dispositivos que utilizan el estándar IEEE 802.11. Por el uso indebido de los términos y por razones de marketing el nombre del estándar se confunde con el nombre de la certificación.

2.1 ORIGEN Y EVOLUCIÓN

La IEEE creó el estándar mencionado, como variante del estándar IEEE 802.3 para las redes de área local, variando la forma de transmisión de cableado a ondas de radio.

La tecnología inalámbrica está conviviendo en la vida cotidiana desde principios de los 90, aunque de manera desordenada, debido a que cada fabricante desarrollaba sus propios modelos, generando por ende dificultades a los otros.

¹ Organización comercial que adopta, prueba y certifica que los equipos cumplen los estándares 802.11 relacionados a redes inalámbricas de área local <http://www.wi-fi.org/>,

En abril de 2000 se establece la primera norma: IEEE 802.11b, que utiliza la banda de los 2,4 GHz y que alcanza una velocidad de 11 Mbps. Tras esta especificación llegó IEEE 802.11a, que como se explica en [5], generó algunos problemas entre Estados Unidos y Europa por la banda que se utilizaba (5 GHz). Mientras que en Estados Unidos esta banda estaba libre, en Europa estaba reservada para fines militares, situación que paralizó un tanto esta tecnología inalámbrica, sobre todo teniendo en cuenta que la mayoría de los fabricantes de dispositivos norteamericanos en su mayor parte, tardaron en reaccionar ante la imposibilidad de vender sus productos en el viejo continente.

Tras muchos debates se aprobó una nueva especificación, 802.11g, que al igual que la “b” utilizaba la banda de los 2,4 GHz pero multiplicaba la velocidad hasta los 54 Mbps.

En el momento en que tres especificaciones diferentes conviven en el mercado, se observa que son incompatibles, por lo que se tuvo que crear equipos capaces de interactuar con todas estas, saltando “en caliente” de unas a otras.

Cuando se da este caso, la banda de los 5 GHz, anteriormente reservada para usos militares, se habilitó para usos civiles, lo que fue un gran adelanto, ya que es ese entonces, no existían otras tecnologías inalámbricas, como *Bluetooth*, *Wireless USB* o *ZigBee* que utilizan la misma frecuencia.

2.1.1 Estándares de la IEEE

- 802.11

Creado en 1997, fue el primero entre sus estándares, soportaba una velocidad máxima de conexión de 2 Mbps, demasiado lento para la mayoría de

las aplicaciones, este estándar dejó de utilizarse hace más de una década y no es compatible con los dispositivos actuales [6].

- 802.11 a

Creado en 1999, esta versión funciona en la frecuencia de los 5 GHz esperando encontrar menos interferencia con dispositivos como teléfonos inalámbricos que usan la frecuencia de 2.4 GHz. El alcance es bastante limitado ya que los objetos bloquean fácilmente la frecuencia de los 5 GHz. Cuenta con una tasa de transmisión de hasta 54 Mbps [6].

- 802.11b

También fue creado en 1999, pero usando la frecuencia 2.4 GHz, sin necesidad de licencia, este estándar fue el causante de que la popularidad de Wi-Fi se incrementara; Cuenta con una tasa de transmisión de hasta 11 Mbps [6].

- 802.11g

Se creó en el 2003 usando la banda de 2.4 GHz pero con una velocidad máxima de 54 Mbps, este estándar fue adoptado ampliamente e incluso sigue siendo utilizado hasta la fecha ya que la velocidad sigue siendo adecuada para la mayoría de aplicaciones, sin olvidar que es más barata. Es compatible con 802.11b, y mejora la tasa de transmisión a 54 Mbps [6].

2.1.2 Extensiones de Estándares Inalámbricos.

- 802.11e

Su objetivo es proporcionar soporte de QoS (Calidad de Servicio) para aplicaciones de redes LAN. Se aplica a los estándares físicos a, b y g de 802.11,

debe proporcionar claves de servicio con niveles gestionados de QoS para aplicaciones de datos, voz y video.

- 802.11i

Se refiere al objetivo más frecuente del estándar 802.11, la seguridad. Se aplica a los estándares físicos a, b y g de 802.11, proporcionando una alternativa a la Privacidad Equivalente Cableada (WEP) con nuevos métodos de encriptación y procedimientos de autenticación. IEEE 802.11x constituye una parte clave de 802.11i.

- 802.11d

Constituye un complemento al nivel de control de Acceso al Medio (MAC) en 802.11 para proporcionar el uso, a escala mundial, de las redes WLAN del estándar 802.11. Permite a los puntos de acceso comunicar información sobre los canales de radio admisibles con niveles de potencia aceptables para los dispositivos de los usuarios.

- 802.11f

Su objetivo es lograr la interoperabilidad de Puntos de Acceso (*AP*) dentro de una red *WLAN* mutiproveedor. El estándar define el registro de *AP* dentro de una red y el intercambio de información entre dichos puntos de acceso, cuando un usuario se traslada desde un punto de a otro.

- 802.11h

Creado para cumplir los reglamentos europeos para redes *WLAN* a 5 GHz., que requieren que los productos tengan el control de la potencia de transmisión (*TPC*) y selección de frecuencia dinámica (*DFS*). El control *TPC* limita la potencia

transmitida al mínimo necesario para alcanzar al usuario más lejano. *DFS* selecciona el canal de radio en el punto de acceso para reducir al mínimo la interferencia con otros sistemas en particular el radar.

Se incluye otros estándares menos conocidos, ya sea por su reciente creación o por tener objetivos de menor importancia tales son:

Tabla 2.1. Otros estándares 802.11

Estándar	Funcionalidad Principal
802.11k: <i>Radio Resource Measurement</i>	Mediciones y registros de rendimiento
802.11n: <i>High Throughput</i>	Alta velocidad de transmisión
802.11p: <i>Wireless Access for the Vehicular Environment</i>	Wi-Fi en vehículos
802.11r: <i>Fast Roaming</i>	Transiciones entre puntos de acceso
802.11s: <i>ESS Mesh Networking</i>	Redes Mesh 802.11
802.11u: <i>Interworking with External Networks</i>	Interoperabilidad con otras redes
802.11T: <i>Wireless Performance Prediction</i>	Métodos de prueba y recomendaciones métricas
802.11v: <i>Wireless Network Management</i>	Administrador de redes inalámbricas
802.11w: <i>Protected Management Frames</i>	Tramas de gestión protegidas
802.11y: 3650–3700 MHz	Opera en los Estados Unidos

2.2 ESTRUCTURA Y DESCRIPCIÓN DE LAS CAPAS DEL ESTÁNDAR IEEE 802.11

El estándar IEEE 802.11 define los dos niveles más bajos del modelo OSI para un enlace inalámbrico utilizando ondas electromagnéticas, En la figura 2.1 se presentan dichas capas, estas son: La capa física (PHY), que ofrece 3 tipos de codificación de la información; y la capa de enlace de datos, constituida de dos subcapas:

- ✓ El control de enlace lógico (*Logical Link Control, o LLC*);
- ✓ El control de acceso al medio (*Media Access Control, o MAC*).

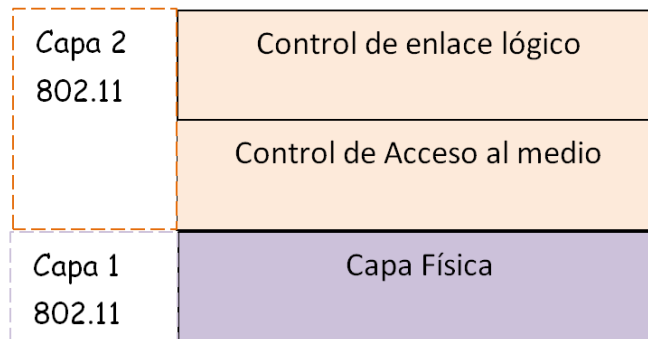


Figura 2.1 Capas definidas por 802.11

2.2.1 Capa Física

La capa física se encarga de transformar un paquete de información binaria (*Frame*) en una sucesión de impulsos adecuados al medio físico utilizado en la transmisión [7]. Como se ve en la figura 2.2, cada segmento de datos proporcionado en la capa 2, al pasar por la capa 1, se transforman en bits. Cuando actúa en modo recepción el trabajo es inverso; se encarga de transformar estos impulsos en paquetes de datos binarios que serán entregados a la capa de enlace.

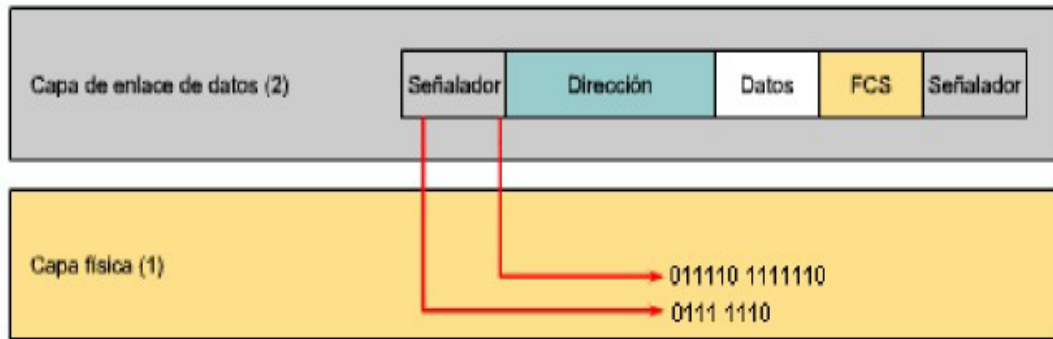


Figura 2.2 Capa física.

IEEE 802.11 define diferentes velocidades de comunicación, lo cual se consigue empleando distintos tipos de modulación y codificación.

Los estándares de transmisión RF son: FHSS, DSSS y OFDM. Estas arquitecturas se definen para operar en la banda de frecuencia de 2,4 GHz, ocupando típicamente los 83 MHz de banda desde los 2,400 GHz hasta 2,483 GHz.

- **FHSS: *Frequency Hopping Spread Spectrum***

Consiste en transmitir una parte de la información en una determinada frecuencia durante un intervalo de tiempo. Pasado este tiempo se cambia la frecuencia de emisión, sin interrumpir la transmisión. De esta manera cada tramo de información se va transmitiendo en una frecuencia distinta durante un intervalo muy corto de tiempo. Como se indica en la figura 2.3, donde cada porción del mensaje se envía en diferente frecuencia.

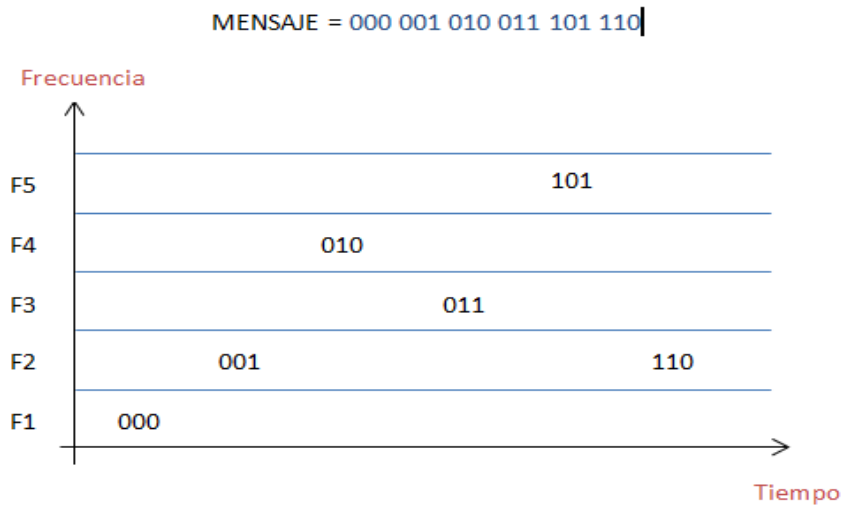


Figura 2.3 FHSS.

El orden en los saltos en frecuencia se determina según una secuencia aleatoria almacenada en tablas, que tanto el emisor y el receptor deben conocer. Si se mantiene la sincronización en los saltos de frecuencias se consigue mantener a nivel lógico, un solo canal de comunicación a pesar que físicamente se cambie de canal.

FHSS es utilizado por sistemas tales como *bluetooth* (enlaces de 10 metros normalmente). No es conveniente para poder transmitir a largas distancias ya que los saltos en frecuencia generan mucha dispersión a la señal lo que hace que se aumente la tasa de bits errados y se aumente el retardo de la transmisión y sea difícil poder recuperar el mensaje.

- ***DSSS: Direct Sequence Spread Spectrum.***

El espectro ensanchado por secuencia directa es una técnica de codificación que utiliza un código de pseudoruido para "modular" digitalmente una portadora, de tal forma que aumente el ancho de banda de la transmisión y reduzca la densidad de potencia espectral (es decir, el nivel de potencia en cualquier frecuencia dada). La

señal resultante tiene un espectro muy parecido al del ruido, como se observa en la figura 2.4, de tal forma que a todos los receptores les parecerá un ruido menor al cual va dirigida la señal.

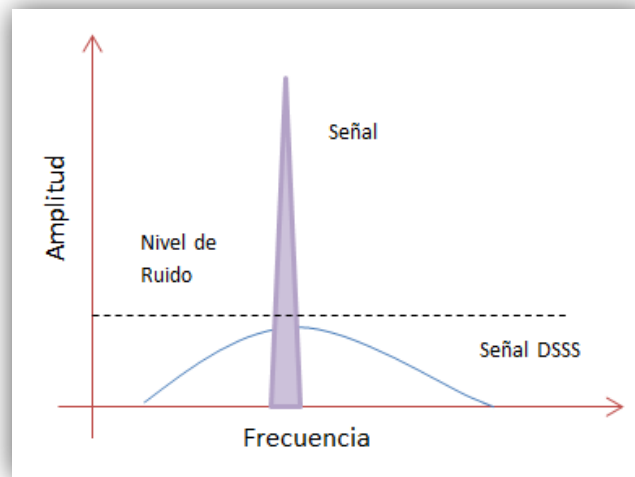


Figura 2.4 DSSS.

Las señales de datos, ya sean un uno o un cero, se ensancha a un código de 11 bits que corresponde al código *Barker* [8]. Cada uno de los bits tiene un tiempo, que se puede observar en la figura 2.5. En el espectro aumenta el ancho de Banda pero se reduce la potencia de la señal.

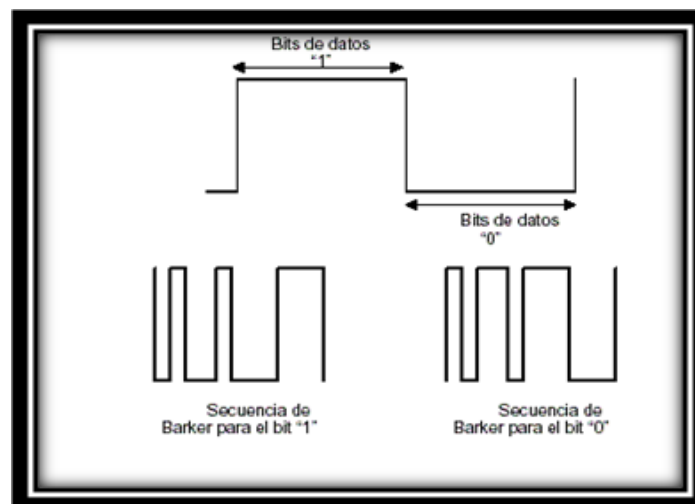


Figura 2.5 Secuencia de código Barker.

A los distintos tipos de modulación y codificación corresponden distintos *BER* (*bit error rate*) para una misma relación señal/ruido. Básicamente, cuanto mayor es la velocidad, más potencia hay que recibir para tener una tasa de error de bits suficientemente baja. La diferencia en la sensibilidad de recepción entre 1 y 11Mbps, aunque depende de equipos, suele ser de más de 10 dB, lo cual equivale a cuadruplicar con 1Mbps el alcance que se tiene con 11Mbps.

- **OFDM: Orthogonal Frequency Division Multiplexing**

La multiplexación por división de frecuencias ortogonales. Es una combinación de la multiplexación y modulación. En un canal de banda ancha OFDM se divide en un canal estrecho. Estos canales de banda estrecha son modulados por los datos y después se multiplexan de nuevo, para crear portadora OFDM, como se indica en la figura 2.6. Donde se encuentran varias subportadoras o bandas estrechas transmitidas simultáneamente.

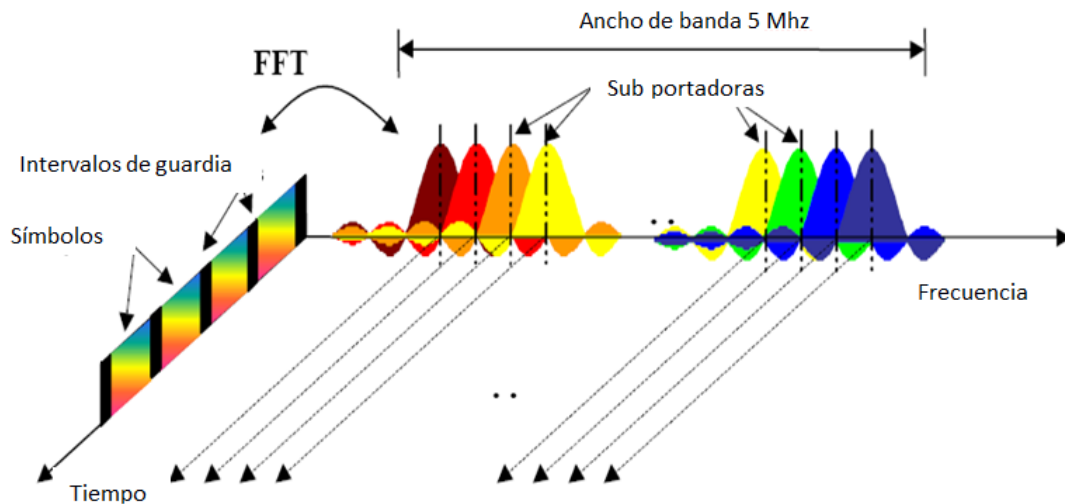


Figura 2.6 OFDM.

2.2.2 Capa de Enlace

La capa de enlace de datos es responsable de la transferencia fiable de información a través de un circuito de transmisión de datos. El nivel de enlace es el segundo nivel del modelo OSI. Recibe peticiones del nivel de red y utiliza los servicios del nivel físico. El objetivo del nivel de enlace es conseguir que la información fluya, libre de errores, entre dos dispositivos conectados directamente (servicio orientado a conexión).

Se divide en dos subcapas que indica la figura 2.7; llamadas la capa de acceso al medio MAC propiamente; y la capa de control lógico LLC.

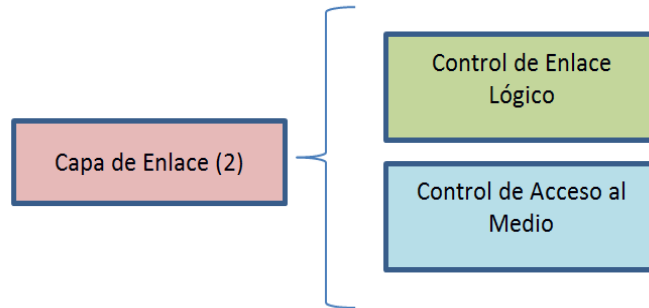


Figura 2.7 Subcapas de la capa MAC.

En la capa de enlace, los datos se organizan en unidades llamadas tramas. Cada trama tiene patrones para diferenciar segmentos distintos por ejemplo en la figura 2.8 se observa el patrón de inicio y fin de trama que se diferencian de segmentos como los destinados a dirección, datos, etc.

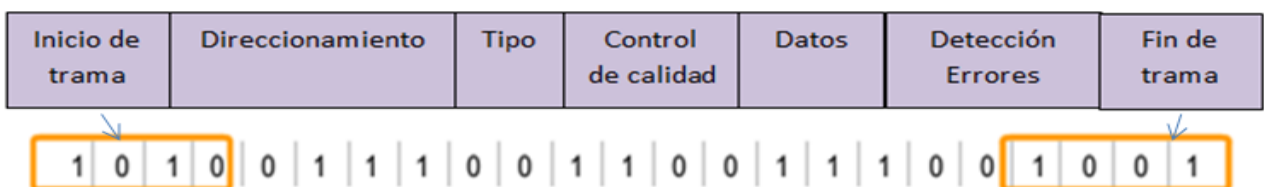


Figura 2.8 Información organizada en tramas.

- **Subcapa MAC.**

La subcapa MAC se encarga, como bien su nombre lo dice, del control de acceso que se tiene al medio, es decir, la forma en que se accederá; que es el principal problema en redes donde muchos ordenadores comparten el mismo medio y pueden hablar entre todos [10].

El método de acceso múltiple en IEEE 802.11 es la Función de Distribución Coordinada (*Distributed Coordination Function, DCF*) que trabaja con el método de Acceso Múltiple por Censado de Portadora con Detección de Colisiones, (*Carrier Sense Multiple Access / Collision Avoidance, CSMA/CA*).

El estándar 802.11 incorpora un ACK para todo *frame* transmitido, es un *ACKnowledgement* o acuse de recibo, es un mensaje que se envía para confirmar que un *frame* o un conjunto de *frames* han llegado. Proceso que se muestra en la figura 2.9. Si el terminal de destino tiene capacidad para detectar errores, el significado de ACK es "ha llegado y además ha llegado correctamente".

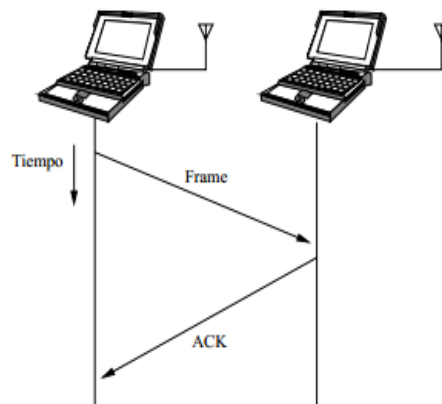


Figura 2.9 Envío de ACK.

El mecanismo de control de acceso al medio tiene que vencer algunos retos específicos de la naturaleza inalámbrica de las redes Wi-Fi, donde se destaca la alta probabilidad de que las redes contengan nodos que puedan colisionar sin si quiera escucharse, que es llamado el fenómeno de nodo oculto y para el que se propone el mecanismo de RTS/CTS².

El protocolo CSMA/CA deriva de la familia de la familia de acceso múltiple CSMA, que se caracterizan porque resuelven el acceso de muchas estaciones a un medio común mediante el censado previo del canal en busca de una portadora reconocible; esto se hace para evitar transmitir si hay una transmisión en curso procedente de otra estación. La particularidad de CSMA/CA es que se adapta a canales de radio al incorporar mecanismos para evitar al máximo las colisiones incluso cuando existen nodos ocultos [11].

Aunque el nivel MAC no es ranurado, se define una duración de espacio, que es un *Slottime*, cuyo valor se fija en el nivel físico. Una ranura es cada una de las fracciones resultantes de particionar el tiempo en ranuras de igual tamaño con inicio y fin bien definidos. Para el caso de DSSS se fija el estándar un valor de 20us y un opcional de 9us en 802.11g que se puede usar si existe la garantía de que todos los equipos involucrados lo soportan.

El resto de tiempos definidos en la MAC fijaran su función en la duración del slot del nivel físico subyacente. En el estándar se denomina IFS (*inter frame space*) a los intervalos entre tramas y, se definen cuatro de ellos para proporcionar niveles de prioridad para el acceso al medio inalámbrico en función del tipo de trama y del modo de coordinación DCF y PCF.

DCF (*Distributed Coordinación Function*), permite compartir el medio de forma automática a través de CSMA/CA; espera un tiempo aleatorio mientras está ocupado

² Normalmente, el envío de *frames* RTS / CTS no se produce a menos que el tamaño del paquete supera el umbral definido. Si el tamaño del paquete a transmitir es mayor que el umbral, el protocolo de enlace RTS / CTS se dispara. De lo contrario, la trama de datos se envía inmediatamente.

y tras enviar una trama a un destinatario concreto, debe recibir un ACK, si este no se recibe en el tiempo determinado se debe retransmitir la misma trama. Este modo de coordinación no hace diferencias entre los puntos de acceso y las estaciones.

PCF (*Point Coordination Function*), proporciona una transferencia de tramas libre de contienda y usa un punto coordinador para determinar que estación transmite y en qué tiempo.

El proceso del protocolo CSMA/CA se describe en la figura 2.10; antes de empezar la contienda, todos los participantes esperan un tiempo DIFS, si el canal está libre se envían los datos y se espera un tiempo SIFS para el envío de la trama ACK, se espera nuevamente un tiempo DIFS después de esta trama y una vez recibido el ACK el participante que envió los datos realiza un proceso de *BACK off* para que otras estaciones puedan disputar el acceso al medio, durante el tiempo la ventana de contención.

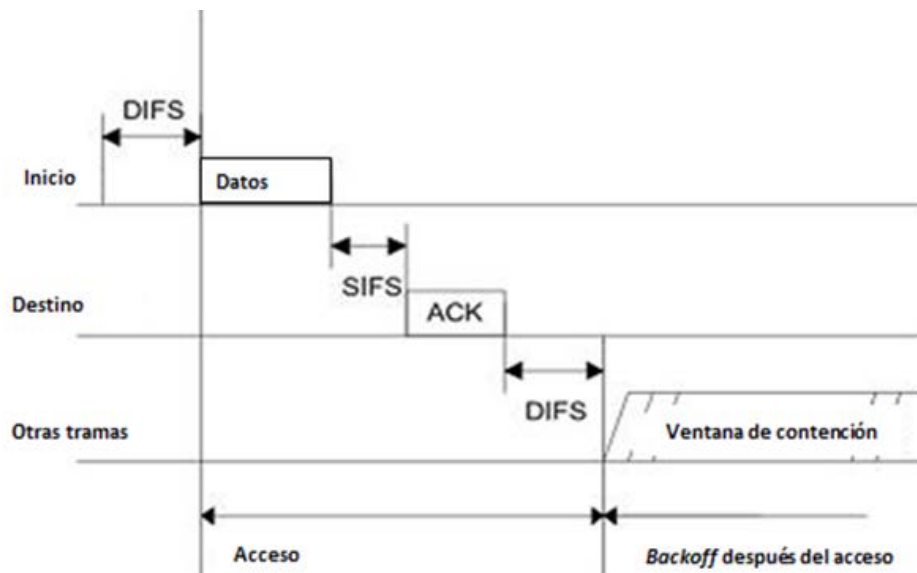


Figura 2.10 Proceso del acceso CSMA/CA.

A continuación se describe cada uno de los tiempos empleados durante la contienda:

SIFS.- (*Short Interface Sopase*), es el mínimo espacio entre tramas. Usado para separar transmisiones pertenecientes a la misma conversación, como una trama y su ACK. Su valor se fija de tal forma que a la estación transmisora le sea posible conmutar al modo de recepción.

PIFS.- (*Point Coordinación IFS*), lo utiliza el AP y solo bajo PCF para ganar el acceso al medio al comienzo del periodo de la contienda para comunicar a cuál de ellas le corresponde la transmisión.

DIFS.- (*Distributed IFS*), utilizado bajo DFS por las estaciones que desean transmitir tramas de datos y control. Su valor se fija como $1 \text{ SIFS} + 1 \text{ Slottime}$. Una estación podrá transmitir tras una trama correctamente recibida (aunque no haya sido dirigida a esta) una vez transmitido el DIFS, si detecta que el medio está libre y cuando su ventana de contención haya expirado.

EIFS.- (*Extended IFS*), es el IFS más largo si una estación recibe una trama que no es capaz de comprender, espera este tiempo antes de intentar transmitir.

Bajo el estándar 802.11b, la IEEE fija las constantes como:

$$\text{Slottime} = 20 \mu s \quad (2.1)$$

$$\text{SIFS} = 10 \mu s \quad (2.2)$$

$$\text{PIFS} = \text{SIFS} + \text{Slottime} = 30 \mu s \quad (2.3)$$

$$\text{DIFS} = \text{SIFS} + 2(\text{Slottime}) = 50 \mu s \quad (2.4)$$

En CSMA/CA se define una ventana de contienda rasurada y se asume que la ranura contiene al tiempo de vulnerabilidad de forma que dos estaciones solo

pueden colisionar si están en la misma ranura. Esto supone que las distancias entre estaciones están limitadas por la duración máxima que puede tener el tiempo de propagación de ida y vuelta, siempre inferior a una ranura, lo que explica el rango en el orden de cientos de metros de los equipos Wi-Fi del mercado.

- **Subcapa LLC**

Define la forma en que los datos son transferidos sobre el medio físico, proporcionando. Es la más alta de las dos subcapas de enlace de datos definidas por el IEEE y la responsable del control de enlace lógico. La subcapa LLC maneja el control de errores, control del flujo, entramado y direccionamiento de la subcapa MAC. Ofrece dos tipos de servicio que son:

- ✓ Servicio orientado a la conexión

Es un servicio que establece una conexión entre las estaciones del enlace, y que garantiza la entrega de las unidades de datos que fluyen a través de dicha conexión. El servicio de conexión le garantiza al receptor la entrega en secuencia de las unidades de datos y la protección contra pérdidas y duplicados. Con ese fin dispone de los mecanismos necesarios para controlar el flujo y corregir los errores.

- ✓ Servicio no orientado a conexión

No establece una conexión previa entre las estaciones, por lo que cada trama intercambiada es independiente de todas las demás (de las enviadas antes y después). Cada trama es individualmente autónoma y autosuficiente ante el receptor. Es un servicio que tiene utilidad cuando el establecimiento de una conexión implica retrasos que son inaceptables para el funcionamiento del sistema. El servicio de enlace sin conexión puede ser con o sin confirmación.

2.3 AJUSTE DE PARÁMETROS PARA WILD

2.3.1 Límites impuestos por la capa física

El estándar IEEE 802.11 permite que la capa física limite el alcance de las comunicaciones Wi-Fi en términos de distancia, además del balance del enlace; los límites físicos de la distancia alcanzable dependerán de:

- ✓ La máxima potencia a la que se pueda transmitir (PIRE)
- ✓ Las pérdidas de propagación
- ✓ La sensibilidad del receptor.
- ✓ La mínima relación señal a ruido que se acepte como suficiente.

2.3.2 Límites impuestos por la capa MAC

Se tienen tres tipos de limitaciones que son: el temporizador de espera de los ACK, la definición de los tiempos relacionados con el tamaño de la ranura, y el cálculo del Vector de Asignación de Redes NAV³ (*Network Allocation Vector*) para la detección de la portadora virtual.

El *ACK-timeout*, el *Slottime* y el DIFS dependen en cierto modo de la propagación sobre el aire. La IEEE define el tiempo de propagación en $1 \mu\text{s} \pm(10\% \text{ SIFS.})$

ACK-timeout.- Si el ACK es menor que el tiempo de propagación, es decir el tiempo que toma desde el último paquete enviado más 1 SIFS + más el inicio del ACK del mismo paquete para regresar al lugar de envío, la MAC que transmite, asumirá que se perdió el paquete, y será innecesario retransmitir el mismo. Si el paquete es retransmitido, terminara por colapsar con el ACK que está de regreso y termina por reducir el *throughput*.

³ Mecanismo de sensado de portadora virtual, donde un valor determinado tiene que llegar a cero en la cuenta regresiva, para que se vuelva a censar el canal.

Si se da un valor muy largo de ACK, el transmisor espera un tiempo innecesariamente largo antes de retransmitir alguna trama perdida y esto también reduce el *throughput* del enlace. El enlace seguirá funcionando pero el *throughput* decaerá mientras la distancia aumente.

Se puede decir lo mismo acerca del CTS aunque el RTS/CTS es solo para reducir colisiones producidas por nodos ocultos, y pueden ser desactivados en la mayoría de casos.

Slottime.- Una estación solo puede transmitir al inicio de un *Slottime*, entonces este tiempo no debe terminar antes de que la señal alcance su destino. Esto nos da una distancia máxima de aproximadamente 6000 metros⁴. Asumiendo que el ACK fue aumentado correctamente, esta sería la máxima distancia entre cualquier estación en un ambiente punto a multipunto (asumiendo que no hay estaciones ocultas).

2.3.2 Cambiando el ACK y el Slottime

Por cada 300 metros que se aumenten en distancia, se añade 1us al *Slottime* de un dispositivo y 2 us de ACK y *CTK-timeout*. Cada estación que se conecte al mismo canal deberá tener las mismas constantes de tiempo [12]. Entonces se tiene que:

$$ACKt = T(máx) + S + t \quad (2.5)$$

$$s = R + T(máx) \quad (2.6)$$

$$D = S + 2s \quad (2.7)$$

4 ACK Timeouts y sus efectos en enlaces de larga distancia. http://www.air-stream.org/ACK_Timeouts

Dónde:

$ACKt = ACK\ Timeout$

$T =$ Tiempo de propagación

$S =$ SIFS

$t =$ Tiempo para transmitir 14 bytes en un ACK *frame* [14x8 / bitrate]

$s = Slottime = +$ Tiempo de propagación (máx.)

$R =$ Retrasos de las capas MAC y PHY

$D =$ DIFS

2.4 ESTADO DEL ARTE

Hasta el momento, la comunidad científica ha dedicado muchos esfuerzos de investigación para determinar las condiciones de usabilidad y de enlaces y redes IEEE 802.11 en que las distancias entre estaciones estén en el orden de los kilómetros o decenas de kilómetros. Aunque no sea han realizado esfuerzos para estandarizar esta tecnología, la razón es que la tecnología Wi-Fi simplemente no fue diseñada para eso y otras tecnologías resultan teóricamente más apropiadas para esos rangos de aplicación propios de las redes metropolitanas y rurales extensas.

No obstante, en esto la realidad contradice en cierto modo a la lógica científica y técnica; cada vez más se ven instalaciones de enlaces con equipamiento Wi-Fi para comunicar equipos que distan varios kilómetros entre sí.

En los congresos y foros nacionales [13], e internacionales especializados en Tecnologías de Información y Comunicación para zonas en desarrollo, en los últimos años se ha ido consolidando el discurso de que Wi-Fi podría ser una solución estratégica para extender la conectividad en zonas aisladas o deficientemente

comunicadas. A pesar de las otras tecnologías, no hay duda que Wi-Fi seguirá teniendo su lugar, por diversas razones, tecnológicas y sobre todo económicas.

Se han realizado ya, varios trabajos de investigación para el desarrollo de esta tecnología; desde sus inicios en [14], donde se empieza a trabajar con enlaces de larga distancia bajo el estándar IEEE 802.11b; y, se concluye que un enlace de estas características, deberá ser planeado para situaciones totalmente predecibles, y que cualquier variación de cualquier clase, afectará drásticamente a la red.

En la India, un nuevo protocolo para las redes IEEE 802.11 de larga distancia, es diseñado por *Raman* y *Chebrolu* en [15], donde se planifica un protocolo llamado *2P MAC*, que trabajaría como reemplazo de CSMA/CA, y alcanzaría la máxima eficiencia usando un solo canal, y si no se requiere de sincronización. Este protocolo fue diseñado con hardware propio para él, lo que es un nuevo avance en la tecnología no estandarizada WiLD, pero aún no lo suficiente para lograr que la tecnología se adapte al medio.

El Doctor Javier Simo en [16] ha propuesto bases como el establecimiento de un modelo analítico que justifica el comportamiento de enlaces con distancias entre 6 y 100 Km modelando el estándar 802.11 y analizando la optimización del estándar 802.11b.

La Ingeniera, Patricia Ludeña en [17], analiza el estándar 802.11n, propone su aplicabilidad en entornos rurales específicamente de América Latina que combinados con 802.1n, conlleva ciertas ventajas como mayor ancho de banda y resistencia a los multitrayectos para estos entornos.

Los ingenieros Ricardo Águila y Jimmy Sánchez en [18], brindan el estudio para los estándares 802.11e y 802.11n donde se indica que en 802.11e cada tipo de tráfico debe recibir un tratamiento diferenciado por parte de la red para garantizar la calidad de servicio. Y al igual que el caso anterior mediante el estándar 802.11n se

puede alcanzar mayor velocidad, cabe recalcar que en este trabajo solo se alcanzó la distancia máxima de 15 Km.

En la escuela politécnica del ejército, las ingenieras Viviana Tamayo y Evelin Barrionuevo en [2] realizan el estudio de la tecnología WiLD para la región sierra del Ecuador, en el estándar 802.11b mediante 2 saltos, alcanzando 60 Km.

El récord de un enlace punto a punto lo tiene en Venezuela el profesor Pietrosémoli que en [1] ha probado enlaces Wi-Fi punto a punto y en el 2006 consigue un enlace de 382 Km entre cadenas montañosas de Venezuela, empleando equipos Wi-Fi no convencionales con antenas parabólicas.

En el país se realizó en el 2007 una propuesta de un enlace a manera de corredor fluvial sobre el río Napo en la zona amazónica fronteriza entre Ecuador y Perú, de casi 350 Km en 13 saltos como se explica en [16].

CAPÍTULO III

MATERIALES Y METODOS

Se han revisado diferentes fuentes de información y mediante una investigación sistemática, se llegó a determinar los materiales óptimos para la realización de este trabajo de graduación. EL presente capitulo detalla los materiales y los métodos empleados durante el proceso de diseño, simulación e implementación de la red.

3.1 MATERIALES

La elección de tarjetas Wi-Fi se basa en parámetros tales como la potencia de transmisión, sensibilidad de recepción, temperatura y humedad soportada en operación, así como chipset incorporado. Esta última condición es importante, ya que es necesario disponer de soporte para un S.O. GNU/Linux para todos los modos (Master, Managed, Ad-hoc,Monitor) para poder construir puntos de acceso.

3.1.1 Selección de la tarjeta de red.

Se realiza una descripción de las tarjetas que existen para trabajar con Wi-Fi en distancias mayores a las comunes en el estándar. Estas son:

- **Router Board.-** Es una de las placas que se basa en el procesador *FreescaleMPC8321* por lo que no tiene FPU (*Floating Point Unit*- Unidad de punto flotante) [19]. Posee la gran ventaja de manejar hasta 4 y 6 interfaces inalámbricas

miniPCI (por medio de tarjetas adicionales) pero con el inconveniente que a más interfaces menos potencia administrada a cada uno [19].

Además viene con un sistema operativo propietario instalado llamado *RouterOS*, esto es un inconveniente por que no se podrá adicionar o cambiar aplicaciones particulares; pero existe la posibilidad de instalar otro sistema operativo no propietario no sólo en su memoria integrada sino que al poseer una ranura para *CF* (compact Flash), se puede instalar otro sistema operativo sin perder el sistema propietario. En la figura 3.2 se observa su vista superior e inferior.

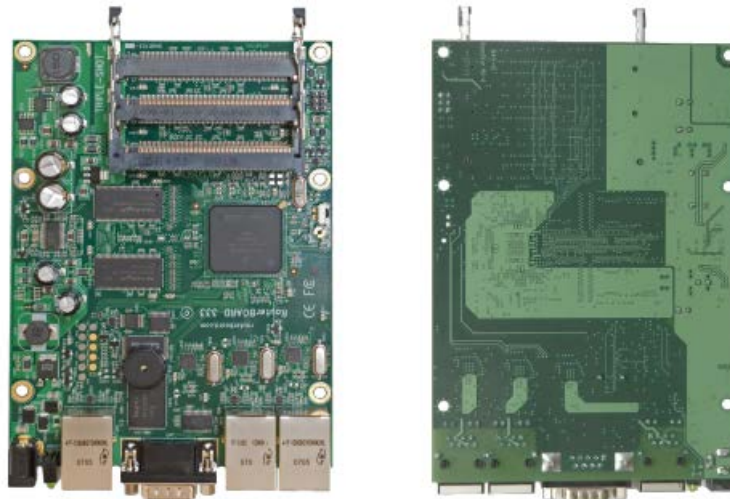


Figura 3.1 Vista superior e inferior, *Router Board 333*

- **Tarjeta Alix.-** Se basa en el procesador *AMD Geode*, es un x86 y posee *FPU*. No viene con ningún sistema operativo, posee ranura para *CF* para instalar uno. Posee dos ranuras *miniPCI* que con modificaciones soportará interfaces inalámbricas de alta potencia. Al ser un sistema x86 se tiene la gran facilidad de trabajar con las distintas aplicaciones de redes conocidas de alto nivel [19].

- ***Pronghorn SBC (Single Board Computer)-250-*** Es una tarjeta que se basa en un procesador Intel *IXP425*, 533 Mhz, por lo que no posee *FPU*. Posee dos ranuras *miniPCI* que soportan interfaces inalámbricas de alta potencia [20]. No trae un sistema operativo pero se le puede instalar el que ofrece el mismo fabricante u

otro porque posee una ranura para *CF*. Es una placa que suministra más corriente a las interfaces *miniPCI*. La figura 3.2 muestra la tarjeta *Pronghorn SBC 250*.

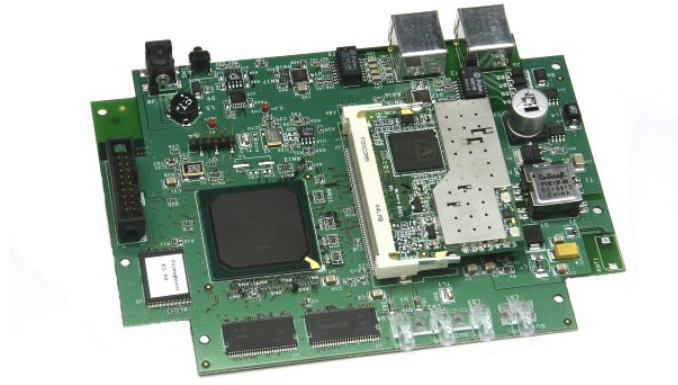


Figura 3.2 *Pronghorn SBC 250*

- **Net4826-30.-** Es una tarjeta basada en un procesador de 233 o 266Mhz, tiene el puerto Ethernet 10/100 Mbit, memoria *SDRam* de 256 Mbytes como memoria principal y una compact flash soldada. Esta placa es optimizada para ser usada como *router* inalámbrico, pero tiene la flexibilidad de varios usos, y cuenta con un rango extendido de temperatura [21].



Figura 3.3 Tarjeta *Net4826-30*

En la tabla 3.1 se resumen las características principales de las algunas de las tarjetas que existen para redes inalámbricas a larga distancia, y de estas la tarjeta Alix cuenta con la posibilidad de instalar el sistema operativo que se desee ya que no viene con uno. Es por eso que se decidió adquirir estas tarjetas en el Centro de Investigaciones de La ESPE, y desarrollar el proyecto ya se requiere ver las diferentes opciones de sistemas para verificar el mejor para modificar los parámetros necesarios.

Tabla 3.1 Comparación de placas para redes inalámbricas.

Equipo	Fabricante	Procesador	Tipo de almacena miento	Puertos Ethernet	Alimenta- ción	Sistema operativo
Router Board 333/600	MikroTik	<i>FreescaleMP</i> C8321	64Mb integrada y ranura CF.	3	12 a 28V DC fuente de 25W	MikroTik Router OS
Alix	PCEngines	AMD Geode, serie, LX	1 ranura CF.	2	7 a 20V, fuente de 15W	Alix Linux, Voyage Linux, Ruoter OS.
Net 4826- 30	Soekris Engineering	AMD Geode serie, SC	256 Mb de Compact Flash integrada.	1	6 a 28V DC fuente de 15W.	Voyage Linux
Pronghor n SBC	ADI Engineering	Intel XScale serie, IXP	16Mb soldado y 1 ranura CF.	2	5VDC	AD Linux Distribution StarOS, Linux OpenWRT, Ikarus OS

3.1.2 Tarjeta ALIX

La empresa *PC-Engines* creó la Tarjeta *ALIX*, con el objetivo de sustituir a *WARP* que es un sistema embebido de bajo costo utilizado para la construcción de *routers*, *firewalls*, balanceadores de carga, etc.

Entre los beneficios se tiene su bajo costo y dimensiones así como también la posibilidad de portar el sistema operativo Linux mediante una compact flash, y principalmente la disminución en el consumo de energía del router. Una tarjeta *Alix* consume alrededor de 5W y 12V DC.

Se utiliza esta tarjeta ya que permite ejecutar la modificación del estándar a nivel de la capa de enlace mediante programación, no requiere refrigeración y tiene una muy buena estructura para exteriores.

- **Características de la tarjeta Alix**

Existen distintos modelos de tarjetas, a continuación en la tabla 3.2, se muestra un resumen de ellas, con algunas características.

Tabla 3.2 Clasificación de tarjetas ALIX

Modelo	CPU	DRAM	LAN	MiniPCI	PCI	USB	Otro	Bios	Tamaño(cm)
Alix 1D	LX800	256MB	1	1	1	2	VGA, Audio, PS/2, LPT, GPIO	Award	(MiniTX)
Alix 2D0	LX700	128MB	2	2	0	0		TinyBIOS	15.2 x 15.2
Alix 2D1	LX700	128MB	3	1	0	0		TinyBIOS	15.2 x 15.2
Alix 2D2	LX800	256MB	2	2	0	2		TinyBIOS	15.2 x 15.2
Alix 2D3	LX800	256MB	3	1	0	2		TinyBIOS	15.2 x 15.2

Alix 2D13	LX800	256MB	3	1	0	2	Batería, I2C, COM2, Cabecera USB interna	TinyBIOS	15.2 x 15.2
Alix 3D1	LX700	128MB	1	2	0	0		TinyBIOS	10 x 16
Alix 3D2	LX800	256MB	1	2	0	2		TinyBIOS	10 x 16
Alix 3D3	LX800	256MB	1	2	0	2		TinyBIOS	10 x 16
Alix 6E1	LX800	256MB	2	1	0	2	Mini PCI Express	TinyBIOS	10 x 16

Para el presente trabajo se utilizó el modelo 3D3 que es una de las últimas versiones y su característica física principal es la optimización de espacio ya que ocupa con funciones útiles las dos caras de la placa. Sus características específicas son:

- ✓ Procesador AMD Geode LX, frecuencia de 500 MHz. LX800
- ✓ 256 kb de memoria cache (64K data + 64K instruction + 128K L2)
- ✓ 1 canal Ethernet de 10/100 Mbit/s.
- ✓ 2 puertos miniPCI para tarjetas 802.11 y otras expansiones.
- ✓ 256 Kb de memoria DDR SDRAM, 64 bits para un mayor ancho de banda en la memoria.
- ✓ 512 kb para el BIOS.
- ✓ 1 puerto para Compact Flash para operaciones del usuario y uso del sistema.
- ✓ Conectividad: Canal Ethernet (10 / 100 Mbits/seg),), ranuras miniPCI para tarjetas inalámbricas y otro tipo de expansiones.
- ✓ Entrada/Salida:
 - 1 puerto serial (DB9 macho).

- 2 puertos USB 2.0,
 - 1 puerto VGA,
 - Salida de audio / entrada de micrófono.
 - ✓ Alimentación: Una de 7 a 20V DC (máximo recomendado 18V) a través de un JACK. Y una pasiva a través de *POE (Power Over Ethernet)*. Para 18V se sugiere 15W.
 - ✓ Cabecera para bus LPC, usado para recuperar la memoria flash o la expansión de Entrada/Salida.
 - ✓ Rango de temperatura de 0° a 50°C.
 - ✓ Dimensiones: 100 x 160mm
- **Partes de la tarjeta Alix**

La placa Alix 3D3 consta de una parte superior y otra parte posterior donde se encuentran los siguientes elementos como se muestran en las Figuras 3.4 y 3.5.

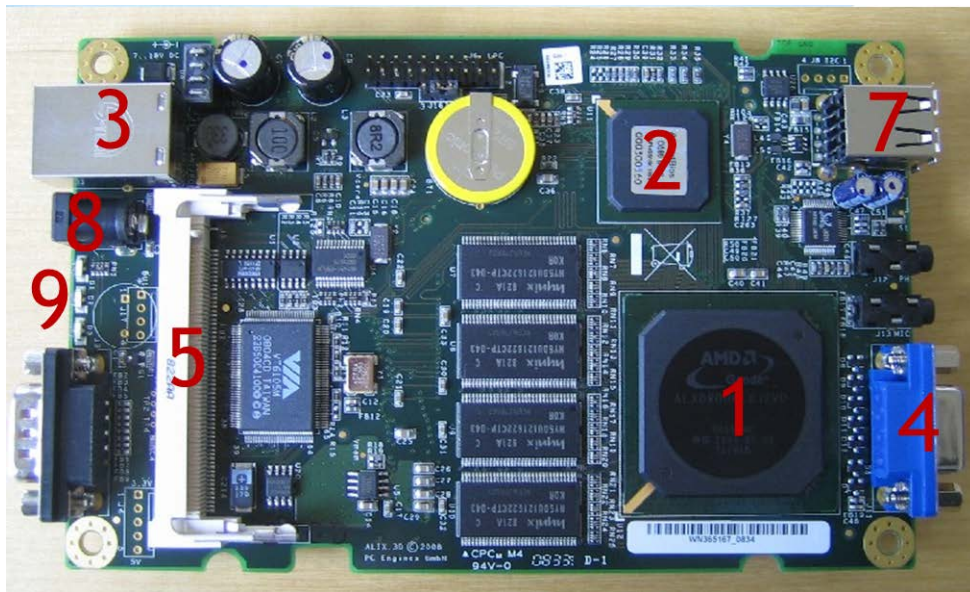


Figura 3.4 Parte superior de la placa 3D3.

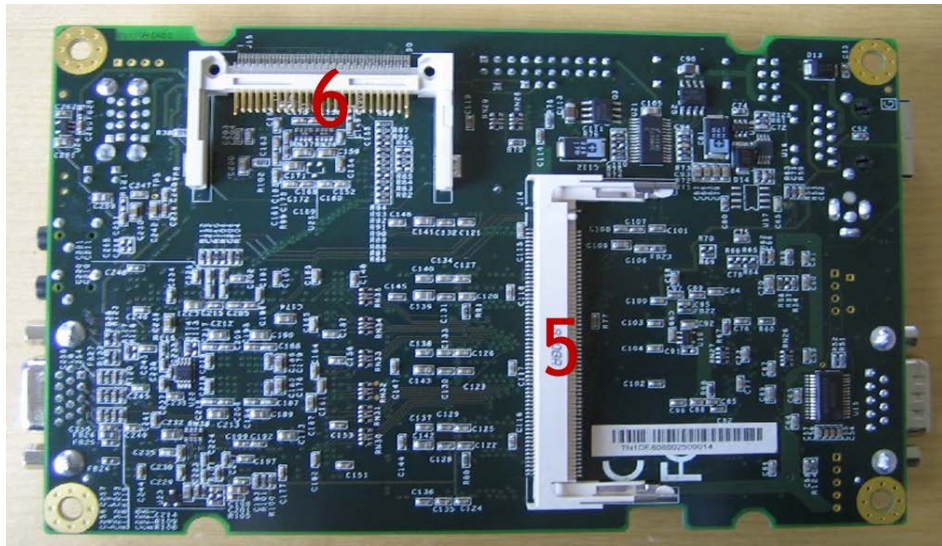


Figura 3.5 Parte inferior de la placa 3D3.

Las partes más sobresalientes de la tarjeta descritas en las figuras 3.4 y 3.5 son:

1. Procesador AMD Geode LX
2. Flash Award Bios que es el sistema básico de entrada salida.
3. Puertos Ethernet
4. Puerto Serial DB9
5. Ranuras PCI para tarjetas inalámbricas.
6. Ranura tarjeta Compact Flash.
7. Puertos USB 2.0
8. JACK DC.
9. Leds indicadores. Estos pueden ser programados para dar avisos según se requiera.

3.1.3 Tarjetas de radio inalámbricas

Las tarjetas de red inalámbricas, también llamadas tarjetas de radio, juegan un papel fundamental dentro del trabajo ya que son las que determinan la óptima

potencia para la transmisión de la señal y la sensibilidad en la recepción. En la tabla 3.3 se presenta un resumen de algunas de las más utilizadas con sus características.

Tabla 3.3 Tarjetas de red inalámbricas.

Fabricante	Interface	Estándar	Potencia Max en a (dbm)	Potencia Max en g (dbm)	Sensibilidad (dbm)
Ubiquiti Networks	SR2	b	-	26 a 1-24 Mbps	6Mbps / -94 12Mbps / -91
	SR5	a	26 a 6-24 Mbps	-	
	XR2	b	-	28 a 1-24 Mbps	
	XR5	a	28 a 6-24 Mbps	-	
Mikrotik	R52H	a/b/g	26 a 6Mbps	26 a 6Mbps	6Mbps / -90
EnGenius Technologies	EMP-8602+S	a/b/g	27 a 6-24 Mbps	22 a 6-24 Mbps	6Mbps / -90 12Mbps / -92

A continuación en las figuras de la 3.6 a la 3.11 se muestran las tarjetas *wireless* que nos permiten alcanzar largas distancias y con las que puede trabajar la tarjeta Alix.

- SR2. Super Range 2

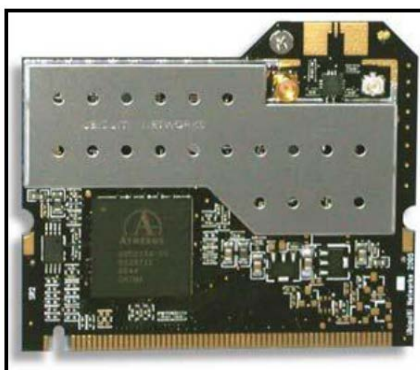


Figura 3.6 SR2.

- SR5. Super Range 5

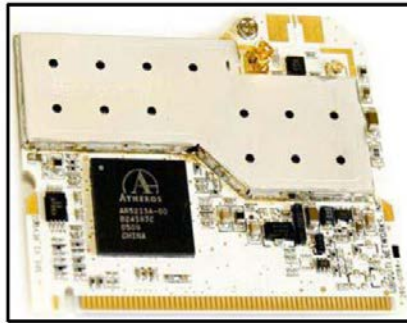


Figura 3.7 SR5.

Información técnica proporcionada para La serie SR-X:

- ✓ Chipset: Atheros 4ta Generación AR5213
- ✓ Interface: 32 bit mini PCI tipo IIIA
- ✓ Voltaje de operación: 3.3 VDC
- ✓ Puertos de antena: u.fl (principal) , MMCX (secundario)
- ✓ Rango de temperatura: -40C a + 80C
- ✓ Seguridad: WPA, WPA2, AES-CCM, encriptación TKIP, 802.1X, WEP 64/128/152 bits.
- ✓ Tasas de trasmisión: 6, 9, 12, 24, 36, 48, 54Mbps,
- ✓ Soporte del canal: Tx 5 /10 /20 / 40 MHz
- ✓ Conformidad RoHS⁵: si

⁵ Restricción de ciertas Sustancias Peligrosas en aparatos eléctricos y electrónicos, (RoHS del inglés "*Restriction of Hazardous Substances*"), fue adoptada en febrero de 2003 por la Unión Europea. Restringe el uso de seis materiales peligrosos que son. Plomo, Mercurio, Cadmio, Cromo VI ,PBB,PBDE

- XR2. XtremeRange2

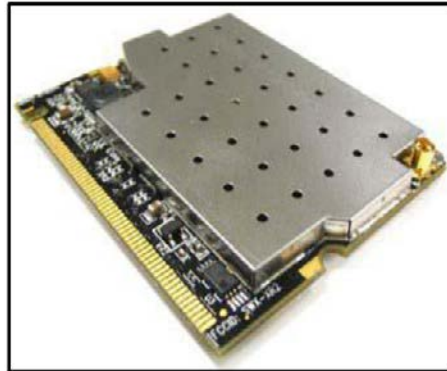


Figura 3.8 XR2.

- XR5. XtremeRange5

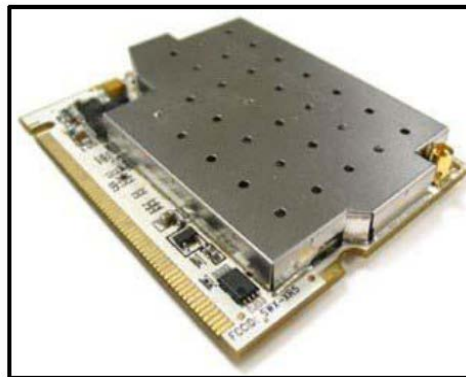


Figura 3.9 XR5.

Información técnica proporcionada para La serie XR-X:

- ✓ Chipset: Atheros 4ta Generación AR5414
- ✓ Interface: 32 bit mini PCI tipo IIIA
- ✓ Voltaje de operación: 3.3 VDC
- ✓ Puertos de antena: MMCX
- ✓ Rango de temperatura: -45C a + 90C y extendido a +95C

- ✓ Seguridad: WPA, WPA2, AES-CCM, encriptación TKIP, 802.1X, WEP 64/128/152 bits.
- ✓ Tasas de transmisión: 6, 9, 12, 24, 36, 48, 54Mbps,
- ✓ Soporte del canal: Tx 5 /10 /20 / 40 MHz
- ✓ Conformidad RoHS¹: si
- ✓ Potencia de salida: 600 mW

- R52H



Figura.3.10 R52H.

Información técnica:

- ✓ Tarjeta *miniPCI* para aplicaciones de alta velocidad en multibanda.
- ✓ 350mW de potencia de salida.
- ✓ Frecuencias de trabajo: 2.192-2.539 y 4.920-6.100GHz.
- ✓ Modo turbo para transferencias más rápidas.
- ✓ Es optimizada para trabajar con *Routers* y protocolos de la misma marca para alcanzar distancias mayores.

- EMP-8602+S



Figura 3.11 EMP-8602+S.

Información técnica:

- ✓ Tarjeta de radio MiniPCI 802.11a+b+g
- ✓ Chipset Atheros - AR5414.
- ✓ Dos conectores de antena, para el estandar a y b.
- ✓ Hasta 600mw output in 802.11b.
- ✓ Sensibilidad:
 - 802.11a: -88dBm en 6Mbps, -70dBm en 54Mbps
 - 802.11g: -90 dBm en 6Mbps, -74 dBm en 54Mbps
 - 802.11b: -95 dBm en 1Mbps -90 dBm en 11Mbps

Debido a sus características y factibilidad de trabajo junto con las tarjetas Alix, las más opcionadas son las provenientes del fabricante Ubiquiti, y de entre estas se trabaja con la serie XR2, por su mayor alcance, sensibilidad y trabajo dedicado con cada estándar.

3.1.4 Sistema Operativo Linux Voyage

Linux Voyage es una distribución de Debian basado en GNU, adaptado específicamente para las placas de PC *Engines* ALIX/WRAP o las de Soekris 45xx/48xx.

Para su instalación se necesitan 128 Mb pero se puede aumentar la capacidad dependiendo de la aplicación; dicha instalación se la realiza dentro de la tarjeta *Compact Flash*.

Hasta la versión 2.6 se encuentra con soporte para el driver *madwifi* y, a partir de la 2.7 en adelante, viene con soporte para el driver *hostap* [22]. Como se muestra, el driver *hostap* es útil sólo para enlaces de hasta 30 Km, lo que para el presente trabajo le es inútil, por lo que se utilizó el driver *madwifi* que a pesar de venir en las versiones anteriores del sistema operativo, es el indicado para los enlaces realizados.

No contiene todas las aplicaciones para implementar redes de datos; contiene sólo lo básico; como los controladores para el manejo de las interfaces Ethernet y para el manejo de las interfaces inalámbricas en base a *chipset Atheros*. Es adecuado para ejecutarlo con características como firewall, *access point* inalámbrico, *gateway* de VoIP, etc.

Para instalar el sistema operativo Linux Voyage, es necesario formatear la tarjeta compact flash primero, para luego instalar el programa dentro de ésta.

3.1.5 Herramientas de simulación

Para poder realizar el trabajo, se utilizaron dos simuladores que se complementan para el resultado final:

- ✓ Radio Mobile.- Con él, se cubre el perfil del terreno y factibilidad del enlace.
- ✓ Ns-2.- Es usado para la probar la transmisión de datos mediante tecnología WiLD. Este es elegido después de una comparación entre simuladores similares.

- **Herramientas de simulación para el perfil del terreno.**

A continuación se describen las principales herramientas utilizadas para simular perfiles de terreno.

- **Radio Mobile**

El programa Radio Mobile es una herramienta para analizar y planificar el funcionamiento de un sistema de radiocomunicaciones fijo o móvil. Este software utiliza mapas con datos digitales de elevación del terreno, junto con los datos de las estaciones de Radiocomunicación y algunos algoritmos, que desarrollan modelos de propagación radio, para obtener los niveles de señal en distintos puntos bien de un trayecto (junto con el perfil del trayecto entre emisor y receptor), utilizable para el cálculo y diseño de Radioenlaces o bien la cobertura sobre una zona determinada para el análisis y la planificación de comunicaciones móviles y fijos en entornos rurales [23].

Este software implementa el modelo de *Longley-Rice* con muy buenas prestaciones para la ejecución de nuestro proyecto, el modelo de predicción troposférica para transmisión radio sobre terreno irregular en enlaces de largo-medio alcance. Además de tener múltiples utilidades de apoyo al diseño y simulación de los enlaces y las redes de telecomunicaciones. Los parámetros a introducir para realizar las simulaciones permiten reflejar de forma fiel los equipos reales que se piensa utilizar en la instalación para la que estarían destinados.

Se utilizó este software para la evaluación de los enlaces y el perfil geográfico de las zonas de trabajo. La obtención de estos mapas puede realizarse directamente desde una opción del software que permite descargarlos de Internet. Hay tres tipos de mapas disponibles: los SRTM, los GTOPO30 y los DTED.

Permite trabajar con frecuencias entre los 20MHz y 40GHz y longitudes de trayecto de entre 1 y 2000 Km.

- **Sirenet**

SIRENET (*Simulation of Radio Electric NETWORKS*) es una aplicación informática que facilita la planificación y gestión de redes radio, es una herramienta de planificación y gestión radioeléctrica que permite el manejo de recursos espectrales de manera rápida y eficiente.

Esta herramienta incorpora las más modernas tecnologías para gestionar de forma instantánea toda la información geográfica necesaria, trabaja sobre Windows desde la versión 95 en adelante. Entre las características principales se encuentran.

La presentación y arrastre de los mapas muy rápido; conversión automática de cualquier sistema de proyección a coordenadas geográficas y viceversa; importación de ficheros cartográficos: cotas, demografía, tráfico, planimetría; facilidades de la base de datos: copiar, duplicar, guardar, guardar como, modificar, borrar, exportar; base de datos multimedia: iconos vinculados a imágenes y vídeo, algoritmos de predicción: UIT-R 370, UIT-R 526, Deygout, Línea de Vista, Okumura-Hata, COST-231 urbano [24].

- ✓ Superposición de resultados sobre mapas.
- ✓ Generación de informes con los resultados.
- ✓ Dibujo: inserción de elementos gráficos que pueden ser salvados a fichero y recuperados.
- ✓ Modo multiusuario: permite compartir recursos (mapas, resultados) por varios usuarios.

3.1.5 Definición de la herramienta de simulación para el perfil del enlace.

Para simular el perfil de terreno de ambas redes, se optó por Radio Mobile, los dos simuladores presentados son aceptables para la simulación; una de las ventajas de Radio Mobile es que no necesita licencia para su uso al contrario de Sirenet. Otra de las ventajas para este caso es que Radio Mobile cuenta con el modelo de propagación de Longley Rice.

También se utilizó este software ya que su distribución es libre y es óptimo para el cálculo de radio enlaces de larga distancia en terreno irregular. Para ello utiliza perfiles geográficos combinados con la información de los equipos que se quiere simular como potencia, sensibilidad del receptor, características de las antenas, pérdidas, etc.

- **Herramientas de simulación para el modelamiento de la WiLD.**

A continuación se describen las principales herramientas utilizadas en la simulación de redes.

- ***ns-2* (Network Simulator 2)**

ns-2 es un es un simulador de redes manejado por eventos discretos. *ns-2* presta un apoyo sustancial para la simulación de *TCP*, el enrutamiento y los protocolos de *multicast* a través de redes y conexiones inalámbricas (locales y por satélite). *ns* también implementa protocolos *multicasting* y algunos de los protocolos de la capa *MAC* para simular *LANs* [25].

ns-2 se ejecuta en Linux, *FreeBSD*, Solaris, Mac OS X y Windows con *Cygwin*, y esta licenciado bajo la versión 2 de la Licencia Pública General de GNU. El hecho de ser un programa de código abierto hace que no sea un producto acabado y q este en contante proceso de desarrollo.

ns-2 fue construido en C++ y proporciona una interfaz de simulación a través de OTcl, un lenguaje orientado a objetos.

Arquitectura

Un usuario general puede trabajar en la esquina inferior izquierda, diseñando y corriendo aplicaciones en Tcl usando el simulador de objetos en la librería. El planificador de eventos y la mayor parte de las componentes de red están implementadas en C++ y están disponibles para OTcl a través de un vinculador (OTcl linkage). El conjunto de todos estos componentes como lo muestra la figura 3.12, forman *ns-2*.



Figura 3.12 Arquitectura *ns-2*.

El usuario describe una topología de red escribiendo scripts OTcl, luego, el núcleo de *ns-2* simula la topología con los parámetros especificados. *ns* es un intérprete de scripts del lenguaje Tcl orientado a Objetos, el cual tiene un planificador de eventos de simulación, librerías de objetos, componentes de red y librerías de módulos de instalación de red (implementación). Como se indica en la figura 3.13

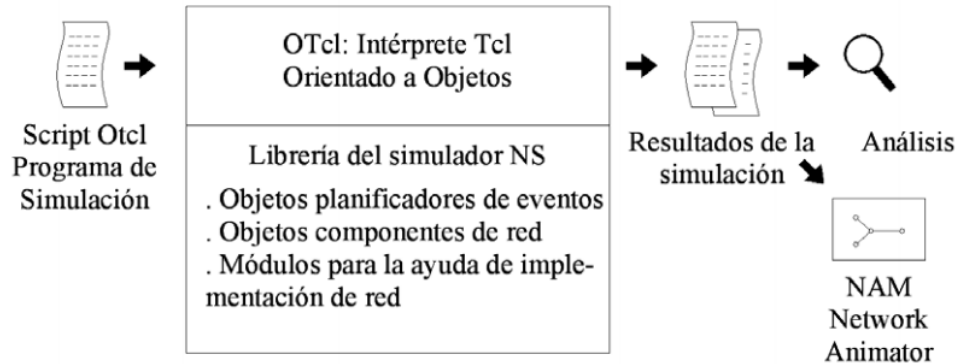


Figura 3.13 Esquema de simulación *ns-2*.

Para establecer y ejecutar una simulación de red, el usuario debe escribir un script Otcl que inicialice un planificador de eventos, establezca la topología de la red usando las funciones de las librerías mencionadas, y diga a las fuentes de tráfico cuando empezar y cuando dejar de transmitir paquetes a través del planificador de eventos.

Un aspecto importante en *ns-2* es la obtención de resultados de las simulaciones. Cuando una simulación termina, *ns-2* produce uno o más archivos de texto de salida que contienen datos detallados de la simulación, siempre y cuando en el script OTcl se lo haya especificado. Los datos pueden ser usados para analizar la simulación o para desplegar una simulación gráfica con la herramienta NAM.

- ***ns-3 (Network Simulator 3)***

ns-3 es un simulador de redes con eventos discretos pensado principalmente para usos educativos. Software libre bajo la licencia de GNU GPLv2⁶ y de código abierto, escrito en C++ como *ns-2*, pero no es una extensión de este, y no soporta sus API's (*Application Programming Interface*), lo que no es compatible con *ns-2* [26].

⁶ GPL. *General public license* de GNU versión 2

Alineado con las necesidades de las redes modernas, las simulaciones de red en *ns-3* pueden ser implementadas en C++ puro, y opcionalmente partes de la simulación se puede realizar utilizando el lenguaje de programación *Python*. El usuario puede transmitir y recibir paquetes generados en *ns-3* en dispositivos reales. *ns-3* provee soporte para la simulación de TCP, *routing*, y protocolos *multicasts* sobre redes cableadas y *wireless*.

La infraestructura de *ns-3* fomenta el desarrollo de modelos de simulación lo suficientemente realísticos lo que permite que sea usado como un simulador en tiempo real. Soporta redes basadas y no basadas en IP, aunque en su mayoría involucra modelos para Wi-Fi, *WIMAX* o *LTE* en capas 1 y 2 y una gran variedad de protocolos de ruteo estáticos y dinámicos como *OLDR* y *AODV*.

Arquitectura

La organización del simulador se describe en la figura 3.14, se trabaja de abajo hacia arriba y los módulos tienen dependencia con los que se encuentran por debajo suyo.

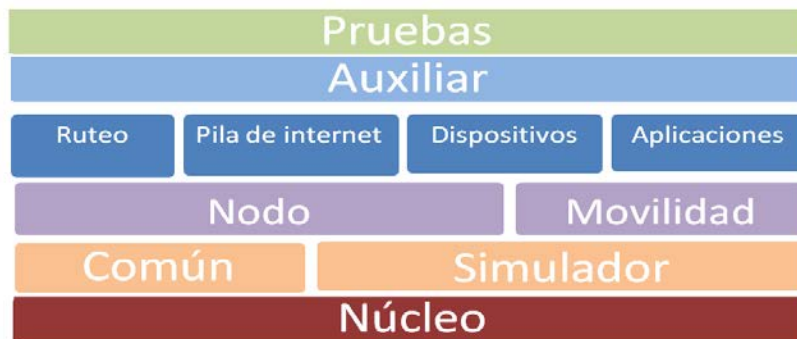


Figura 3.14 Arquitectura *ns-3*.

- ✓ Núcleo.- Contiene los punteros inteligentes y los atributos dinámicos del sistema. Se ocupa del registro y seguimiento de variables randómicas.

- ✓ Común.- Es un registro que contiene paquetes, etiquetas, cabeceras y la escritura de archivos en Ascii.
- ✓ Simulador.- Contiene los eventos, planificaciones de tiempo y aritmética.
- ✓ Nodo.- Se encuentra la clase, tipos de direcciones de los dispositivos de red (IPV4, MAC, etc.), colas y puertos de paquetes.
- ✓ Movilidad.- alberga los modelos de movilidad como estático o dinámico.
- ✓ Auxiliar.- Envolturas de alto nivel para todos los demás detalles, dirigido a las secuencias de comandos.

- **NCTUns**

Es un simulador y emulador⁷ de redes alámbricas, inalámbricas, ópticas celulares satelitales y sistemas de telecomunicaciones avanzados, sus siglas vienen de la universidad donde fue creado (*National Chiago Tung University, netwok simulator*). Es un software ejecutado sobre la plataforma de Linux. Durante su simulación, modifica el núcleo de Linux lo que hace más precisos a los resultados [27].

Para describir la topología, parámetros y configuración, *NCTUns* utiliza una sintaxis sencilla desde la interfaz gráfica totalmente integrada, es decir ya no es un simulador generado por código.

Arquitectura

⁷ A diferencia de un simulador que solo trata de reproducir el comportamiento de un programa, un emulador modela el dispositivo de manera que funcione como si estuviese siendo usado en el aparato original

La figura 3.15 representa la arquitectura distribuida de *NCTUns*. La máquina de simulación se compone de las modificaciones del kernel, los módulos de protocolo, la maquina propiamente y el coordinador.

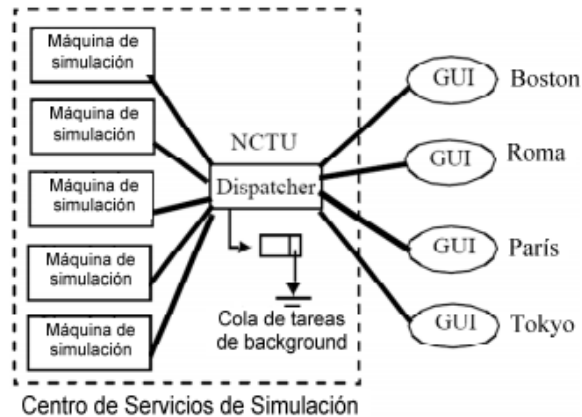


Figura 3.15 Arquitectura NCTUns.

Soporta la emulación de largas redes en diferentes máquinas, las máquinas de simulación pueden estar a largas distancias de donde se ejecutan los programas GUI.

3.1.6 Definición del software para el modelamiento la red WiLD.

Comparando las características de cada simulador de red, se observa que la mayoría no permite la modificación de la capa Mac a tan alto nivel como el requerido por la tecnología propuesta.

El simulador más conveniente para este trabajo es *ns-2* ya que permite establecer las modificaciones necesarias mediante código y exportar los datos a graficadores externos. Su instalación completa se detalla en el anexo 1.

A pesar de que nos brinda un muy buen alcance de varias decenas de kilómetros, no es suficiente para la tecnología en curso con distancias de hasta 80

Km, por lo que se investigó un parche para el programa llamado TeNS el cual permite optimizar y lograr la cobertura total de la red.

The Enhanced Network Simulator (TeNS) es una extensión del Network Simulator 2. TeNS está diseñado para abordar las deficiencias de *ns-2* en el modelado de protocolo IEEE 802.11 de la capa MAC, que en el *ns* original está muy simplificada [28].

Aparte de una aplicación más realista de esto, también incorpora características adicionales como soporte de múltiples interfaces de los nodos móviles, un protocolo de enrutamiento estático para los escenarios móviles, y también cuenta con la inclusión de simples antenas direccionales. La utilidad se puede ver en la simulación de enlaces de larga distancia.

- ✓ Las características más sobresalientes de esta mejora son:
- ✓ Soporte para interfaces múltiples
- ✓ Ruteo estático para nodos inalámbricos
- ✓ Interferencia co-canal añadida
- ✓ Soporte para tasas adaptativas para 802.11
- ✓ Esquema de modulación bpsk añadido.
- ✓ Soporte para antena direccional.
- ✓ Números de canales configurables
- ✓ Entrada para ARP
- ✓ Se añaden protocolos de enlace punto a punto
- ✓ Se añaden algunos parámetros de la capa MAC como RTS, *Threshold*,

La instalación de este parche se detalla en el anexo 2 y el trabajo con la mejora del *Threshold*, se detalla en el capítulo 4 durante la simulación de la red.

Para optimizar el trabajo dentro de *ns-2* son necesarios los siguientes programas que serán útiles al momento de graficar los resultados con mayor precisión.

- ***Trace Graph***

Trace Graph es una herramienta libre, para graficar y especialmente analizar las trazas generadas (*.tr) por el simulador de redes *ns-2*. Para el funcionamiento de *Trace graph* es necesario tener instalado en el sistema, ciertas librerías de *Matlab* y un convertidor de archivos “.tr “ a para que sean reconocidos por *tracegraph*, el cual es llamado *Trace Converter* [29].

La gran problemática que ha existido en el ámbito de analizar los resultados generados por las simulaciones, es que los graficadores utilizados (como *Xgraph*), no soportan en su estado natural los distintos tipos de trazas existentes, sin existir tampoco ningún tipo de convertidor de formato de archivos automático.

Es un analizador de archivos de traza que puede ser instalado en los Sistemas Windows, Linux y Unix y requiere del programa *Matlab* 6.0 o superior. Las trazas generadas por *ns-2* proporcionan toda la información sobre la simulación, pero su formato complejo hace difícil la extracción de resultados de forma que el usuario pueda entender, por esto se utilizan programas capaces de obtener los principales parámetros de rendimiento de red como: *throughput*, *jitter*, retardos, etc., y mostrarlos de manera gráfica. Su instalación y forma de trabajo se describe en el anexo 5.

Trace Graph soporta los siguientes formatos de archivos de Traza:

- ✓ Wired.
- ✓ Satélite.
- ✓ Wireless (old and new trace).
- ✓ New trace.
- ✓ wired-wireless

- Características:

Las características más importantes de *Tracegraph 2.02* son las siguientes:

- ✓ Posibilidad de exportar la información a cientos de tipos de gráficos 2D.
- ✓ Posibilidad de exportar la información a 12 tipos de gráficos 3D.
- ✓ Algunos de los gráficos y análisis que se pueden realizar son:
 - Retrasos (delays).
 - Variaciones (jitters).
 - Tiempos de procesamiento.
 - Round Trip Times (RTT).
 - Throughput (productividad).
 - Estadísticas.
 - Etc.
- ✓ Es posible obtener información y estadísticas de la red entera y relaciones entre nodos (links).
- ✓ Los resultados pueden ser guardados como archivos de textos, gráficos, imágenes JPG y TIFF.
- ✓ Toda traza que posea 2 o 3 columnas de información puede ser graficada.
- ✓ Información en los ejes x, y, z, como mínimos, máximos, desviaciones, etc.
- ✓ Todo script al ser procesado es analizado automáticamente.

- ***Trace Converter***

Trace Converter es una herramienta desarrollada por *Jaroslav Malek* para facilitar el análisis de los archivos de traza generados por el simulador *ns-2*, aplicación generada con Free Pascal; Permite la conversión de archivos de traza `archivo.tr` en archivos de traza `archivo_tg.tr` para su utilización dentro de un entorno gráfico denominado *Trace Graph* [30].

La conversión del archivo de traza al formato de *Trace Graph* por parte de *ns-2* ocupa más del 90% del tiempo total de carga del archivo, si se usa el *Trace Converter* la conversión al formato solo tarda el 10% del tiempo total. Por lo tanto, la conversión al formato *Trace Graph* es casi instantánea.

La manera más simple de ejecutar la conversión de los archivos de traza es mediante la utilización de comandos a través de la consola. A través del comando *batch* permite la conversión de archivos de traza de redes inalámbricas. Una vez que se realiza el cambio de formato el nuevo archivo adquiere la misma extensión pero con una variación en el nombre.

archivo_tg.tr

Con el archivo ya convertido se puede trabajar dentro de la interfaz gráfica de *Trace Graph*. En la figura 3.16 se muestra como se ejecuta el comando para la conversión de un archivo test .tr al formato requerido por *Trace Graph*.

```
C:\Users\sony vaio\Desktop\tesis\trconverter\pr1>trconvert test.tr -batch
Creating trace files list...
File 1/1
Trying to recognize format of test.tr trace file...
10000 lines have been processed to recognize trace format.
old wireless format has been detected
Converting test.tr trace file...
test.tr has been converted to test_tg.tr
```

Figura. 3.16 Conversión del archivo .tr.

3.1.7 Herramientas para inyección de tráfico.

- **iPERF y jPERF**

Son herramientas de software que permiten la generación y monitoreo de tráfico mediante el establecimiento de conexiones lógicas entre 2 PCs bajo el esquema cliente servidor. Permiten la creación de tráfico Tcp y Udp [31].

Ambos programa trabajan con Windows y GNU/Linux.

La diferencia entre estos dos programas es que Iperf se ejecuta mediante línea de comandos, y Jperf se ejecuta sobre Java.

- **D-ITG**

Distributed Internet Traffic Generator (D-ITG), es una plataforma de código abierto capaz de producir tráfico con paquetes de tamaño determinado, replicando procesos estocásticos apropiados para el tamaño de paquetes (PS - *PACKet Size*) y para el tiempo interno de salida entre paquetes (IDT - *Inter Departure Time between pACKets*) [32]. La instalación completa de este software se detalla en el anexo 3.

Soporta protocolos IPv4 e IPv6, tráfico UDP y TCP, que pueden ser generados en las capas de red, transporte y aplicación. Trabaja sobre Linux, OSX y Windows.

El programa puede calcular el retardo de ida (OWD - *One Way Delay*) y el de ida y vuelta (RTT - *Round Trip Time*); entrega también la evaluación de paquetes perdidos y la medida del *throughput* por lo que está concebido para ser usada como una herramienta distribuida de medición de rendimiento.

Es posible configurar una capsula con las variables aleatorias para cada generación de tráfico, esta opción brinda la posibilidad de repetir exactamente el patrón de tráfico, al usar la misma capsula durante diferentes repeticiones.

- **Arquitectura**

La plataforma D-ITG sigue el modelo cliente-servidor y muestra una arquitectura distribuida multicomponente. La figura 3.17 muestra la relación entre los principales bloques del programa que son ITGSend, ITGRecv, ITGLog, e ITGDec, la

comunicación entre el transmisor y el receptor se realiza usando un canal de señalización separado de los datos bajo un protocolo de configuración. (*Traffic Specification Protocol - TSP*).

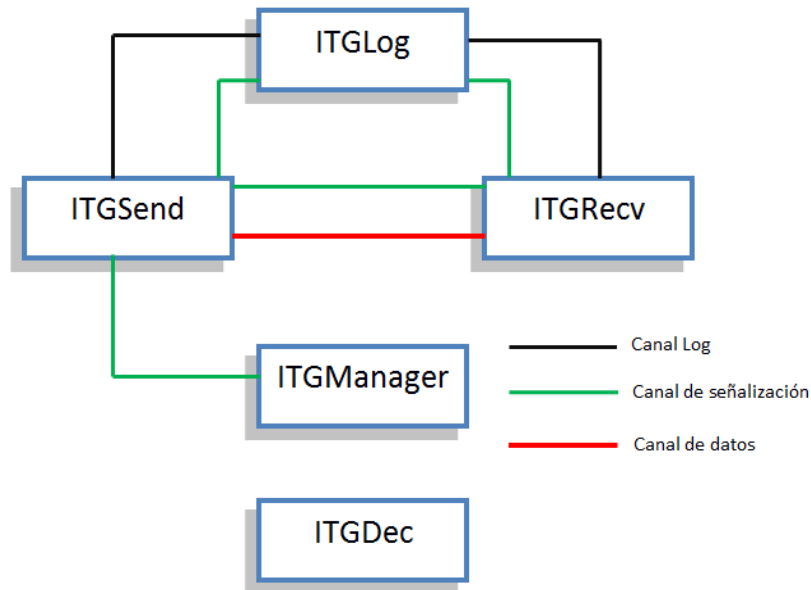


Figura. 3.17 Arquitectura D-ITG.

ITGSend.- Es el componente destinado a la transmisión, opera en 3 modos diferentes:

- Modo de flujo único.** Un solo canal es responsable por la generación de un solo flujo y del manejo de la señal a través del protocolo TSP.
- Modo de flujo múltiple.** Se genera un conjunto de flujos, que opera como una aplicación multiproceso, un canal implementa el protocolo TSP y guía el proceso mientras que el otro genera el flujo de tráfico.
- Modo daemon.** El componente de transmisión es controlado remotamente por el ITGManager. Para recoger estadísticas, durante el proceso ITGSend registra la información acerca de los flujos generados.

ITGRecv.- Trabaja siempre como una herramienta daemon⁸, escuchando permanentemente si hay nuevas conexiones TSP. Cuando una nueva conexión requiere entrar, éste genera un nuevo segmento por cada flujo, responsable por el manejo de la comunicación con el transmisor.

ITGLog.- Es un servidor de registro, corre en un distinto alojamiento que el ITGSend y el ITGRecv, se encarga de recibir y almacenar la múltiple información del transmisor y receptor. Para esta actividad de registro se usa un protocolo de señalización que permite la entrada y salida al servidor. La información puede ser enviada mediante un canal confiable (TCP) o por un canal no tan confiable (UDP).

ITGDec.- Es una utilidad que permite decodificar toda la información generada en el “log” de ITGSend y en el “log” de ITGRecv para analizar los resultados.

D-ITG permite almacenar información en el transmisor y en el receptor, y es posible recuperar información desde donde se generó, adicionalmente permite al transmisor y al receptor conectarse con un servidor remoto mediante una operación de *Log* o registro, lo que es útil cuando el transmisor o el receptor tienen limitada capacidad de almacenamiento como en el caso de PDAs, o celulares.

Para graficar los datos obtenidos se utiliza el programa GNUPlot. Cuya instalación se detalla en el anexo 4.

3.1.8 Selección de la herramienta para inyección de tráfico.

Para realizar la inyección de tráfico se optó por el programa D-ITG, Se escogió esta herramienta para trabajar debido a su versatilidad ya que es desarrollado mediante una plataforma basada en Java y posee un ambiente gráfico que simplifica su uso, y le da la ventaja de analizar todos los parámetros, caracterizar el tráfico inyectado

⁸ En Unix y otros S.O. multitareas, un daemon es un tipo especial de proceso informático no interactivo, es decir, que se ejecuta en segundo plano y de forma continua, en lugar de ser controlado directamente por el usuario.

estadísticamente, y medir los parámetros de Calidad de Servicio (QoS) mencionados de una manera amigable con el usuario. A diferencia de otros inyectores de tráfico similares, como el Iperf, Jperf, Rude/Crude [33], éstos últimos solo pueden generar tráfico Udp.

3.2 MÉTODOS

3.2.1 Configuración De Equipos

- **Formateo de la unidad Compact Flash**

El proceso se realiza en el puerto SCSI-IDC de 50 pines, (lector de tarjetas CF) mostrado en la figura 3.18, directamente en un ordenador, es decir no se debe realizar la instalación sobre la placa Alix.



Figura. 3.18 Tarjeta compact flash y puerto SCSI-IDC 50 pines.

Luego se abre un terminal de Ubuntu, para este paso nos ubicamos en Aplicaciones > Accesorios > Terminal. Una vez abierto el terminal se procede a escribir:

```
sudo su + ENTER,
```

para posteriormente digitar la contraseña de administrador, esto permite cambiar de usuario a root el cuál brinda los permisos necesarios para no tener problemas al momento del formateo de la memoria. Si es que no se ha entrado como *root* directamente al momento de iniciar el sistema.

Se digitar el comando:

`dmesg + ENTER,`

Éste nos muestra los numerosos mensajes que genera el núcleo, así como los dispositivos de hardware que el kernel detecta, dónde están ubicados y si pueden ser configurados o no. En la figura 3.19 se muestra el ingreso de la compact flash, ubicada en el puerto “*sdb*” y de 3.77GB de espacio disponible.

```
50.016757] CE: hpet5 increasing min_delta_ns to 15000 nsec
50.016763] CE: hpet5 increasing min_delta_ns to 22500 nsec
50.485645] sd 6:0:0:0: [sdb] 7372512 512-byte logical blocks: (3.77 GB/3.
```

Figura 3.19 Resultado del comando `dmesg`.

Los dispositivos pueden tomar los nombres de “*sd x* ”, donde “*x*” son letras del abecedario como: a, b, c, d, e, f, g, h, j, k, l,...,z. y se enlistaran de acuerdo a la cantidad de dispositivos conectados.

Hay que señalar que la letra “a” está asociada al disco donde se encuentra instalado Linux, por lo tanto se debe tener mucho cuidado en el

`umount/dev/sdb1 + ENTER.`

Como lo indica la figura 3.20

```
root@isabela-desktop:~# umount /dev/sdb1
```

Figura 3.20 Desmonte de la Compact Flash.

Una vez desmontada la Compact Flash se digita:

```
mkfs.ext2 /dev/sdb1,
```

para realizar un formato a la unidad sdb1, como se ilustra en la figura 3.21. Se utilizó el sistema de ficheros ext2 ya que admite particiones de disco de hasta 4TB y ficheros de hasta 2GB de tamaño, además proporciona nombres de ficheros largos, de hasta 255 caracteres.

```
root@isabela-desktop:~# mkfs.ext2 /dev/sdb1
mke2fs 1.41.11 (14-Mar-2010)
Etiqueta del sistema de ficheros=
Tipo de SO: Linux
Tamaño del bloque=4096 (bitácora=2)
Tamaño del fragmento=4096 (bitácora=2)
Stride=0 blocks, Stripe width=0 blocks
230144 nodos-i, 920574 bloques
46028 bloques (5.00%) reservados para el superusuario
Primer bloque de datos=0
Número máximo de bloques del sistema de ficheros=943718400
29 bloque de grupos
32768 bloques por grupo, 32768 fragmentos por grupo
7936 nodos-i por grupo
Respaldo del superbloque guardado en los bloques:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Escribiendo las tablas de nodos-i: hecho
Escribiendo superbloques y la información contable del sistema de ficheros:
hecho
```

Figura 3.21 Formateo de la Compact Flash como ext2.

El siguiente comando que se utiliza es:

```
tune2fs -c 0 /dev/sdb1,
```

Para ajustar las características de una partición ext2/3/4, y deshabilitar la revisión del disco cada cierto número de montajes o días, como lo muestra la figura 3.22; se configura la respuestas del kernel frente a un error.

```
root@isabela-desktop:~# tune2fs -c 0 /dev/sdb1
tune2fs 1.41.11 (14-Mar-2010)
Se pone la cuenta de montajes máxima a -1
```

Figura 3.22. Desactivación de la revisión del disco.

Si se desea comprobar la partición, se usa el comando

```
fdisk -l,
```

En el que debe constar el sistema operativo Linux en unidad /dev/sdb1. La figura 3.23 indica el correcto particionamiento para el nuevo sistema.

```
Disco /dev/sdb: 2096 MB, 2096898048 bytes
65 cabezas, 62 sectores/pista, 1016 cilindros
Unidades = cilindros de 4030 * 512 = 2063360 bytes
Tamaño de sector (lógico / físico): 512 bytes / 512 bytes
Tamaño E/S (mínimo/óptimo): 512 bytes / 512 bytes
Identificador de disco: 0x27fd8f0a

Disposit. Inicio    Comienzo      Fin          Bloques  Id  Sistema
/dev/sdb1  *              1             1016        2047239+ 83  Linux
```

Figura 3.23 Partición con Linux correcta.

- **Instalación del sistema operativo Voyage Linux en la Compact Flash**

Como se mencionó, la versión utilizada es la 6.5, la última que permite trabajar con el driver *Madwifi*. Su instalación se realiza dentro de la unidad CompactFlash y para ello se realizan los siguientes pasos:

1.- Se descarga el Sistema Operativo Linux Voyage 0.6.5 de la página oficial⁹ en un archivo comprimido:

2.- Se guarda el archivo en un lugar accesible para el administrador, en nuestro caso fue en el escritorio.

3.- Dentro de un terminal se dirige a la ubicación del archivo y se procede a descomprimirlo de la siguiente forma:

⁹ <http://www.voyage.hk/download/voyage/voyage-0.6.5.tar.bz2>


```
tar jxvfp voyage-0.6.5.tar.bz2,
```

4.-Se dirige al directorio donde se encuentra el archivo extraído y se crea los directorios “mnt y cf”, con el comando *mkdir*, como se ilustra en la Figura 3.24.

```
root@isabela-desktop:/home/isabela/Escritorio# cd voyage-one-0.6.5
root@isabela-desktop:/home/isabela/Escritorio/voyage-one-0.6.5# mkdir /mnt/cf
```

Figura 3.24 Ingreso a la carpeta Voyage y creación de directorios.

NOTA: este paso no es necesario para versiones de Voyage 0.7.x ya que al descomprimirlo crea los directorios mencionados anteriormente por default.

5.- Para actualizar el sistema operativo voyage y arrancar el script de instalación, se ejecuta el siguiente comando:

```
./usr/local/sbin/voyage.update
```

Inmediatamente se muestra el menú para la instalación como se muestra en la figura 3.25.

```
root@isabela-desktop:/home/isabela/Escritorio/voyage-one-0.6.5# ./usr/local/s
bin/voyage.update
What would you like to do?
 1 - Create new Voyage Linux disk
 2 - Update existing Voyage configuration
 3 - Exit
(default=1 [Create new Voyage Linux disk]):
```

Figura 3.25 Menú de instalación Linux Voyage 0.6.5.

6.- Se elegir la opción “1”, para crear un nuevo disco Linux Voyage, posteriormente se pregunta dónde se encuentra en archivo Voyage, como se

encuentra en este caso en el Escritorio no existe ningún problema ya que esa es la dirección por default.

7.- En el siguiente menú, se opta por la opción “2” porque se elegirá dónde se va a instalar el sistema operativo, y luego digitar “5” para seleccionar la tarjeta Alix con la cual se va a trabajar, como muestra la figura3.26.

```
What would you like to do?
 1 - Specify Distribution Directory
 2 - Select Target Profile
 3 - Select Target Disk
 4 - Select Target Bootstrap Loader
 5 - Configure Target Console
 6 - Partition and Create Filesystem
    (default=2 [Select Target Profile]):

Please select Voyage profile:
 1 - 4501
 2 - 4511/4521
 3 - 4801
 4 - 5501
 5 - ALIX
 6 - Generic PC
 7 - Notebook (pcmcia)
 8 - WRAP
    (default=5 [ALIX]): 
```

Figura 3.26. Selección de la tarjeta Alix.

8.- En el siguiente menú se selecciona el número 3, “Select Target Disk”, que permite escoger el disco donde se instalará, se muestran todos los bloques de memoria disponibles; Se debe escribir donde se encuentra el dispositivo Compact Flash, ya que no está dado por defecto, así:

/dev/sdb

Como existe la posibilidad de que haya más dispositivos conectados en el sdb, el sistema pregunta si debe usar la partición 1 por defecto, como se muestra en la figura 3,27.

```

1 - Specify Distribution Directory
2 - Select Target Profile
3 - Select Target Disk
4 - Select Target Bootstrap Loader
5 - Configure Target Console
6 - Partition and Create Filesystem
   (default=3 [Select Target Disk]):

Partitions information
major minor #blocks name
-----
8         0  244140625 sda
8         1    102400 sda1
8         2  117656576 sda2
8         3  111467002 sda3
8         4   14908320 sda4
8        16   3686256 sdb
8        17   3682296 sdb1

Which device accesses the target disk [/dev/sdb]? /dev/sdb
Which partition should I use on /dev/sdb for the Voyage system [1]?

```

Figura 3.27. Selección del disco Compact Flash.

9.- Una vez escogida la partición para la instalación de Linux Voyage, se muestra la información del bloque elegido para verificar errores, y se debe aceptar el uso del directorio “/mnt/cf” que fueron creados anteriormente, como muestra la figura 3.28.

```

Which partition should I use on /dev/sdb for the Voyage system [1]?
Device information for /dev/sdb1
device    fs_type label      mount point    UUID
-----
/dev/sdb1 ext2          /media/9f70c152-e8b7-4569-8efd-3580102da2a8 9f70c152-e8b7-4569-8efd-3580102da2a8

Where can I mount the target disk [/mnt/cf]?

```

Figura 3.28 Directorio donde se montará el Sistema Operativo.

10.- En este menú se debe escoger el número 4 donde especificaremos el “*Target Bootstrap Loader*” que es el que maneja el arranque del dispositivo, existen 2

de estos gestores, son *Grub* y *Lilo*¹⁰. En este caso elegiremos lilo ya que es el estándar óptimo. El proceso se indica en la figura 3.29.

```
What would you like to do?
 1 - Specify Distribution Directory
 2 - Select Target Profile
 3 - Select Target Disk
 4 - Select Target Bootstrap Loader
 5 - Configure Target Console
 6 - Partition and Create Filesystem
   (default=4 [Select Target Bootstrap Loader]):

Which loader do you want (grub or lilo) [lilo]? lilo

What would you like to do?
 1 - Specify Distribution Directory
 2 - Select Target Profile
 3 - Select Target Disk
 4 - Select Target Bootstrap Loader
 5 - Configure Target Console
 6 - Partition and Create Filesystem
   (default=5 [Configure Target Console]):
```

Figura 3.29 Selección del sistema de arranque.

11.- Ahora se debe configurar la consola que elegimos, mediante la opción “5” como muestra también la figura 3.30.

```
Select terminal type:
 1 - Serial Terminal
 2 - Console Interface
   (default=1 [Serial Terminal]):

Please choose speed:
 1 - 2400
 2 - 4800
 3 - 9600
 4 - 19200
 5 - 38400
 6 - 57600
 7 - 115200
   (default=5 [38400]):
```

Figura 3.30 Elección y configuración del puerto serial.

¹⁰ Grub (GRand Unified Bootloader) y Lilo (Linux LOader) son gestores de arranque que soportan a sistemas abiertos; lilo viene como estándar en todas las distribuciones de Linux y tiene métodos de grabación de memoria un poco riesgosos ya que se deben sobrescribir manualmente, y Grub fue creado con mayor efectividad de tiempo y procesos.

12.- Se determina el tipo de terminal en este caso es serial que se indica por default, y la velocidad de trabajo de 38400. Este proceso también se muestra en la figura 3.28.

13.- Como siguiente paso verificamos las características que hemos configurado previamente, para que con éstas se permita copiar la distribución al medio seleccionado. Para esto se requiere ingresar la letra “y” como respuesta al ítem “6” de la instalación, como muestra la figura 3.31.

```
6 - Partition and Create Filesystem
  (default=7 [Copy Distribution to Target]):

Configuration details:
-----
Distribution directory: /home/isabela/Escritorio/voyage-one-0.7.5
Disk/Flash Device:    /dev/sdb
Installation Partition: /dev/sdb1
Create Partition and FS: yes

Will be mounted on:  /mnt/cf

Target system profile: ALIX
Target console:       serial
Target baud rate:     38400

Bootstrap installer:  lilo

OK to continue (y/n)? 
```

Figura 3.31 Detalles de configuración.

14.- Le tomará un tiempo al sistema instalar el nuevo Linux Voyage, si todo esta correcto solo se mostrara una advertencia solucionada. Al finalizar se selecciona la opción 8 para salir del menú de instalación como en la figura 3.32.

```
Running lilo ....
Warning: LBA32 addressing assumed
Added Linux *
Skipping /vmlinuz.old
One warning was issued.
copyfiles.sh script completed
What would you like to do?
 1 - Specify Distribution Directory
 2 - Select Target Profile
 3 - Select Target Disk
 4 - Select Target Bootstrap Loader
 5 - Configure Target Console
 6 - Partition and Create Filesystem
   (default=8 [Exit]):
```

Figura 3.32 Sistema Operativo Linux añadido.

- **Configuración de interfaces en la tarjeta Alix**

Para modificar los parámetros necesarios para la red, Se trabaja ingresando a la compact flash mediante SSH¹¹, una vez insertada en la tarjeta Alix y a través de un puerto Ethernet. Los pasos a seguir son:

1.- Se ingresa desde un terminal en Linux, y se accede a la tarjeta mediante su IP, así:

```
ssh 12.0.0.5
```

Contraseña: voyage

Si es que no se conoce la dirección IP, se ingresa al directorio media/ROOTS_FS.

Si se va a trabajar desde Windows, se necesita un programa llamado PuTTY, de instalación sencilla y donde se ingresa la dirección IP de la tarjeta y el modo de conexión SSH, como se muestra en la figura 3.33.

¹¹ Secure SHell, en español: intérprete de órdenes segura) es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red.

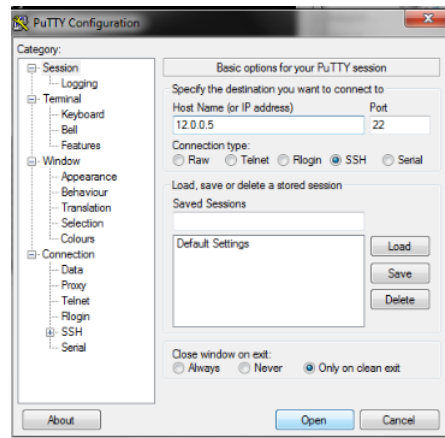


Figura 3.33 PuTTY

El usuario para este proceso será “root” y la clave “voyage”. Una vez que hemos ingresado por cualesquier manera, se nos mostrara la ventana que muestra la figura 3.34.

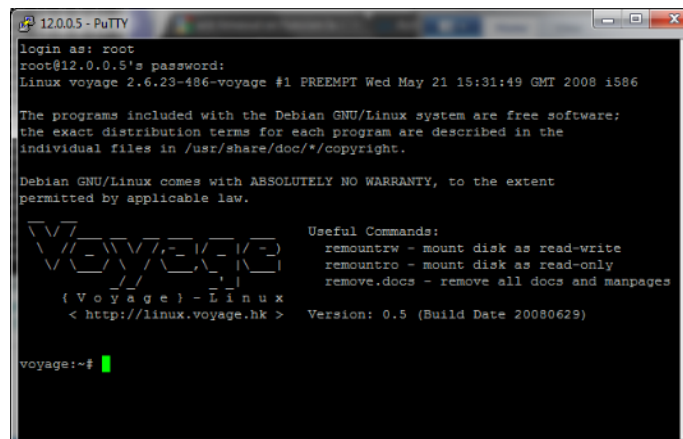


Figura 3.34 Acceso correcto a la Compact Flash.

2.- Una vez dentro de la tarjeta Compact Flash, se digita el comando “*remountrw*” para que el disco funcione en modo lectura y escritura, y luego se ingresa al archivo *interfaces* mediante el editor de textos “VI¹²”, así.

¹² Se lo puede hacer con cualquier editor de texto, pero VI permite el manejo de información en forma remota y sobre la marcha, es decir con la Compact Flash insertada en la tarjeta Alix.

vi etc/network/interfaces

3.- Dentro del archivo se modifica la información de master y esclavo. La interfaz *eth0* es la red cableada con la se comunica el tarjeta Alix a un ordenador a través de un puerto Ethernet. La configuración para las 2 tarjetas se indica en las figuras 3.35 y 3.36.

```
voyage:~# remountrw
voyage:~# vi /etc/network/interfaces
# Used by ifup(8) and ifdown(8). See the interfaces(5) manpage or
# /usr/share/doc/ifupdown/examples for more information.
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 12.0.0.5
    network 12.0.0.0
    netmask 255.255.255.0
    broadcast 12.0.0.255
    #up nat.sh ath0 eth0 "12.0.0.0/24"
```

Figura 3.35 Configuración del puerto eth0 Alix master.

```
auto eth0
iface eth0 inet static
    address 13.0.0.6
    network 13.0.0.0
    netmask 255.255.255.0
    broadcast 13.0.0.255
```

Figura 3.36 Configuración del puerto eth0 Alix esclavo.

4.- Las interfaces “wlan0” y “wlan1” se utilizan para trabajar con el driver de hostapd. Y las interfaces “ath0” y “ath1” son para madwifi. Estas son las interfaces que se encargan de la parte inalámbrica entre las tarjetas en los dos puntos del enlace. Para este trabajo se usan las interfaces “ath”, la configuración para ambas tarjetas se indica en las figuras 3.37 y 3.38.


```
auto ath0
iface ath0 inet static
    address 10.0.0.5
    netmask 255.255.255.0
    #gateway 10.0.0.1
    broadcast 10.0.0.255
    wlanconfig ath create wlandev wifil wlanmode ap
    madwifi-base wifil
    wireless-mode Master
    # wireless-mode Adhoc
    # wireless-mode Man
    up iwpriv ath0 mode 2
    up iwconfig ath0 mode master
    up iwconfig ath0 essid VOYAGE
    up iwconfig ath0 txpower 16
    up iwconfig ath0 enc off
    up iwconfig ath0 rate 5MB
    up nat.sh ath0 eth0 "10.0.0.0/24"
    up route add -net 13.0.0.0 netmask 255.255.255.0 gw 10.0.0.6
```

Figura 3.37 Configuración del puerto ath0 Alix master.

```
auto ath1
iface ath1 inet static
    address 10.0.0.6
    netmask 255.255.255.0
    broadcast 10.0.0.255
    wlanconfig ath1 create wlandev wifil wlanmode sta
    madwifi-base wifil
    # wireless-mode master
    # wireless-mode Adhoc
    wireless-mode Managed
    up iwpriv ath1 mode 2
    up iwconfig ath1 mode managed
    up iwconfig ath1 essid VOYAGE
    up iwconfig ath1 txpower auto
    up iwconfig ath1 enc off
    up iwconfig ath1 rate 11Mb
    up iwconfig ath1 channel 1
    up nat.sh ath1 eth0 "10.0.0.0/24"
    up route add -net 12.0.0.0 netmask 255.255.255.0 gw 10.0.0.5
```

Figura 3.38 Configuración del puerto ath1 Alix esclavo.

La interfaz ath0 se ubica en el puerto número 5 en la parte superior de la tarjeta Alix. Y la ath1 en la parte inferior. (Véase Figura 3.1 y 3.2).

Algunos comandos para la edición en “VI” son:

- ✓ **i**: habilita la escritura
- ✓ **x**: borra el carácter que este bajo esta.

- ✓ **dd:** borra la línea completa
- ✓ **wq:** guarda los cambios realizados

Después de cada comando se debe accionar la tecla de escape. Se dan algunos lugares para obtener tutoriales y más información¹³ acerca de este editor.

- **Cambio de parámetros de la capa física en la Tarjeta Alix**

Dentro del sistema operativo Voyage, se permite examinar características del enlace como potencias y canales. Para la tarjeta que cumple las funciones de *master*, las potencias obtenidas mediante el comando “iwlist txpower” son las mostradas en la figura 3.37 y para el esclavo son las mostradas en la figura 3.38.

```
voyage:~# iwlist txpower
lo      no transmit-power information.
eth0    no transmit-power information.
wifi0   no transmit-power information.
wifi1   no transmit-power information.
ath1    8 available transmit-powers :
        0 dBm      (1 mW)
        7 dBm      (5 mW)
        9 dBm      (7 mW)
        11 dBm     (12 mW)
        13 dBm     (19 mW)
        15 dBm     (31 mW)
        17 dBm     (50 mW)
        19 dBm     (79 mW)
Current Tx-Power:19 dBm      (79 mW)
```

Fig 3.39 Lista de potencias Esclavo.

¹³ <http://www.dc.fi.udc.es/~afyanez/info-vi/index.html> , <http://www.unirioja.es/cu/enriquez/docencia/Quimica/vi.pdf>

```
voyage:~# iwlist txpower
lo      no transmit-power information.
eth0    no transmit-power information.
wifi0   no transmit-power information.
wifil   no transmit-power information.
ath0    8 available transmit-powers :
        0 dBm      (1 mW)
        4 dBm      (2 mW)
        6 dBm      (3 mW)
        8 dBm      (6 mW)
        10 dBm     (10 mW)
        12 dBm     (15 mW)
        14 dBm     (25 mW)
        16 dBm     (39 mW)
        Current Tx-Power=16 dBm      (39 mW)
```

Figura 3.40 Lista de potencias Master.

En el archivo interfaces se puede configurar la potencia máxima de salida, pero basándose en la lista de potencias disponibles, estas varían según la tarjeta de radio con la que se trabaje.

De manera similar, mediante el comando “*iwlist ath frequency*” se puede observar los canales disponibles para trabajar, se ingresa la interfaz donde se requiera obtener los datos. En el archivo interfaces se configura un mismo canal para las dos tarjetas, y si no se hace se elegirá por defecto el más libre.

La lista de canales para master y esclavo se muestran en las figuras 3.41 Y 3.42 respectivamente.

```

voyage:~# iwlist ath1 frequency
ath1    46 channels in total; available frequencies :
Channel 01 : 2.412 GHz
Channel 02 : 2.417 GHz
Channel 03 : 2.422 GHz
Channel 04 : 2.427 GHz
Channel 05 : 2.432 GHz
Channel 06 : 2.437 GHz
Channel 07 : 2.442 GHz
Channel 08 : 2.447 GHz
Channel 09 : 2.452 GHz
Channel 10 : 2.457 GHz
Channel 11 : 2.462 GHz
Channel 36 : 5.18 GHz
Channel 40 : 5.2 GHz
Channel 42 : 5.21 GHz
Channel 44 : 5.22 GHz
Channel 48 : 5.24 GHz
Channel 50 : 5.25 GHz
Channel 52 : 5.26 GHz
Channel 56 : 5.28 GHz
Channel 58 : 5.29 GHz
Channel 60 : 5.3 GHz
Channel 64 : 5.32 GHz
Channel 149 : 5.745 GHz
Channel 152 : 5.76 GHz
Channel 153 : 5.765 GHz
Channel 157 : 5.785 GHz
Channel 160 : 5.8 GHz
Channel 161 : 5.805 GHz
Channel 165 : 5.825 GHz
Current Frequency=2.452 GHz (Channel 9)

```

Figura 3.41 Lista de canales Esclavo.

```

voyage:~# iwlist frequency
lo      no frequency information.

eth0    no frequency information.

wifio   no frequency information.

wifil   no frequency information.

ath0    23 channels in total; available frequencies :
Channel 01 : 2.412 GHz
Channel 02 : 2.417 GHz
Channel 03 : 2.422 GHz
Channel 04 : 2.427 GHz
Channel 05 : 2.432 GHz
Channel 06 : 2.437 GHz
Channel 07 : 2.442 GHz
Channel 08 : 2.447 GHz
Channel 09 : 2.452 GHz
Channel 10 : 2.457 GHz
Channel 11 : 2.462 GHz
Current Frequency=2.412 GHz (Channel 1)

```

Figura 3.42 Lista de canales Master.

- **Cambio de parámetros en la capa MAC para larga distancia en la tarjeta Alix**

El proceso de configuración culmina con el ajuste de parámetros para trabajar a largas distancias, los parámetros a cambiar son, como se explicó en el capítulo 2, el *ACK-time-out*, CTS (Clear to Send) y el intervalo de tiempo definido Slot-time.

El driver de Madwifi hace que este proceso sea nada complicado, ya que solo ingresando la distancia del enlace, modifica los parámetros indicados.

Dentro del sistema operativo Linux Voyage se ingresa la distancia en metros, requerida para el enlace mediante el comando:

```
athctrl ath0 -d 67000
```

Donde:

athctrl: Comando necesario para modificar los parámetros de la capa MAC

ath0: Interfaz con la que se va a trabajar.

-d: Comando que hace referencia a la distancia del enlace.

67000: Valor en metros de la distancia entre emisor y receptor, este valor cambia según los requerimientos de la red.

Ahora verificamos los valores obtenidos para el *ACK-timeout*, *CTS-timeout*, y *slottime*. Mediante el comando "cat", y accediendo al directorio de cada parámetro. Este proceso se muestra en la figura 3.43.

```
voyage:~# athctrl ath0 -d 67000
Setting distance on interface wifi0 to 67000 meters
voyage:~# cat /proc/sys/dev/wifi0/acktimeout
469
voyage:~# cat /proc/sys/dev/wifi0/ctstimeout
469
voyage:~# cat /proc/sys/dev/wifi0/slottime
233
voyage:~#
```

Figura 3.43 Configuración de parámetros para larga distancia.

Este proceso se lo realiza tanto en el master como en el esclavo si es un enlace punto a punto, o en tantos saltos como tenga el enlace.

Los ficheros virtuales (a los que se acceden mediante el comando *cat*) de cada parámetro guardan los valores que éstos tienen a cada instante. El valor que toman el *ACK-timeout* y *CTS-timeout* son iguales, ya que los dos son un tiempo de respuesta, el primero indica que se recibió el paquete y el segundo que no se está efectuando otro proceso de transmisión.

CAPÍTULO IV

PLANIFICACIÓN E IMPLEMENTACIÓN DE LA RED

4.1 PLANIFICACIÓN DE LA RED

4.1.1 Estudio de los lugares donde se implementarán los enlaces

Los enlaces serán implementados de manera que se pueda observar características diferentes dentro de la misma región costa, debido a la diversidad climática propia de la región. Por lo que se decidió montar un enlace sobre tierra y uno sobre mar.

Las estaciones pertenecen al Comando Conjunto Ecuatoriano, se pudo aprovechar su ubicación. Cumpliendo con los requisitos de distancia con mínimo 50 Km, se eligieron los puntos mencionados en el capítulo 1, que son entre los cerros Salinas, Ánimas y Cabuyas.

- **Enlace Salinas – Ánimas**

El primer enlace fue sobre tierra, se tomó los puntos del cerro Salinas ubicado dentro de la Base Naval Salinas en la provincia de Santa Elena y el cerro Ánimas en la provincia del Guayas a 312 m sobre el nivel del mar, la distancia entre estos dos puntos es de 66,7 Km.

En la estación Salinas se trabaja en el cerro que lleva su mismo nombre, éste cuenta con una altura de apenas 92 m, suficiente para enlazarlo con los otros cerros.

La facilidad que se tiene al trabajar en este ambiente es la línea de vista, ya que no existe ningún elemento que obstaculice el enlace punto a punto. La figura 4.1 muestra la vista satelital del enlace entre el cerro Salinas y el cerro Ánimas y la distancia entre ellos. La tabla 4.1 contiene las coordenadas de estos 2 puntos.

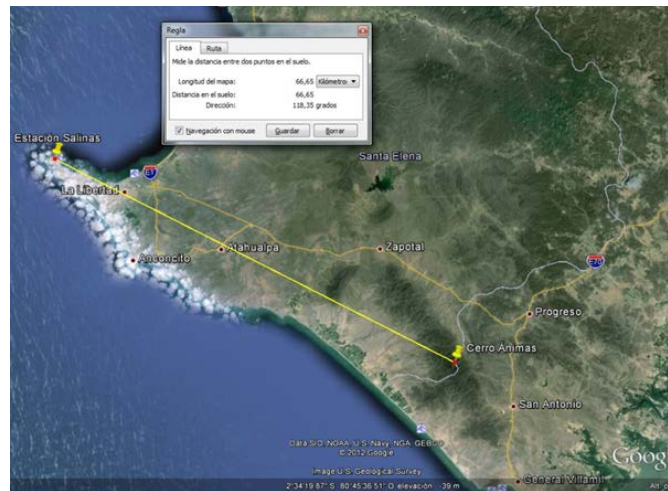


Figura 4.1. Vista satelital Salinas – Animas.

Tabla 4.1 Coordenadas Salinas – Animas.

	Latitud	Longitud	Altura
C. Salinas	2° 11' 10.18'' S	80 ° 59' 34'' W	92 m
C. Ánimas	2° 28' 12.5 '' S	80 ° 27' 52.8'' W	312 m

- **Enlace Salinas – Cabuyas**

Para el segundo enlace que fue sobre mar, se tomaron los puntos de la base naval en salinas provincia de Santa Elena y el cerro Cabuyas en la provincia de Manabí a 322 m sobre el nivel del mar, la distancia entre estos dos puntos es de 75 Km aproximadamente.

La figura 4.2 muestra la vista satelital del enlace entre el cerro Salinas y el cerro Cabuyas y la distancia entre ellos. La tabla 4.2 contiene las coordenadas de estos cerros.

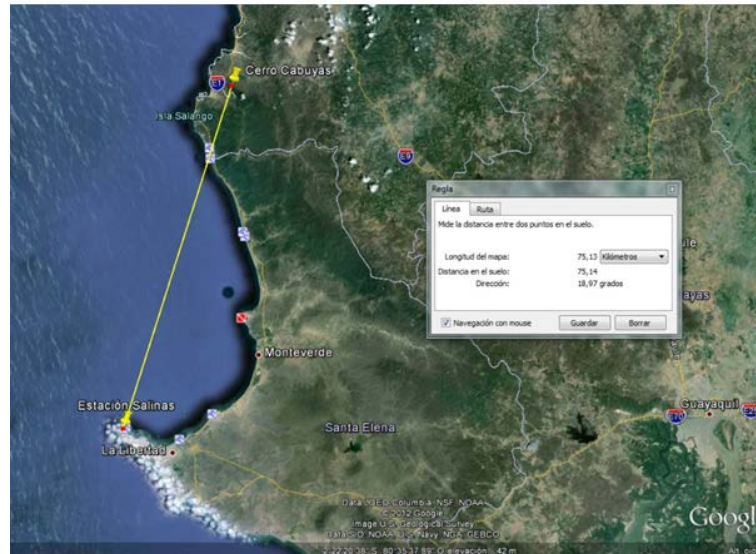


Figura 4.2. Vista satelital Salinas - Cabuyas.

Tabla 4.2 Coordenadas Salinas – Cabuyas.

	Latitud	Longitud	Altura
C. Salinas	2° 11' 10.18" S	80 ° 59' 34" W	92 m
C. Cabuyas	1° 32' 38" S	80 ° 46' 19" W	322 m

4.1.2 Diseño de la red.

- Enlace Salinas – Animas

Como primer paso se realizó un esquema de la red a implementar, como se muestra en la figura 4.3. Se cuenta con un dispositivo “Power station” de largo alcance fabricado por Ubiquiti, este nos ayuda en la alineación de las antenas ya que

cuenta con *leds* indicadores y con un software que nos permite verificar el nivel de la señal recibida. Una vez obtenido el enlace, se reemplaza este equipo por las tarjetas Alix 3D3 configuradas previamente como master y esclavo respectivamente.

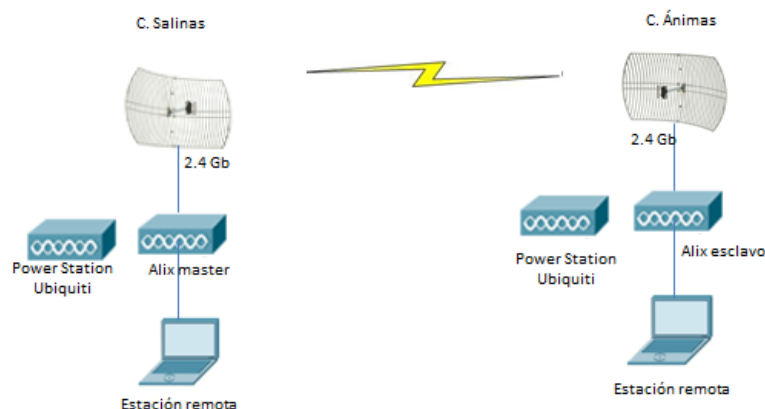


Figura 4.3. Esquema del enlace Salinas - Animas.

Para el ruteo es necesario separar en 3 subredes (como muestra la configuración de tarjetas en el capítulo 3), una que ira desde la estación remota a la tarjeta Alix, la segunda que saldrá inalámbricamente entre antenas y la tercera entre la tarjeta y la segunda estación remota, estas estaciones pueden ser cualquier dispositivo con capacidad para SSH en caso de que se requiera modificar la configuración de las interfaces; o en su defecto serán los terminales a los que se les brinde algún servicio proporcionado por la red.

A continuación se realizó la simulación del radio enlace en el software destinado Radio Mobile, la figura 4.4 muestra la factibilidad del enlace en el simulador.



Figura 4.4. Enlace Salinas - Ánimas en Radio Mobile.

Para el enlace se ingresan datos como la altura de las antenas, la potencia de transmisión, la ganancia de las antenas, frecuencia de operación de la red. Como se indica en la figura 4.5, mediante la interfaz gráfica de radio Mobile.

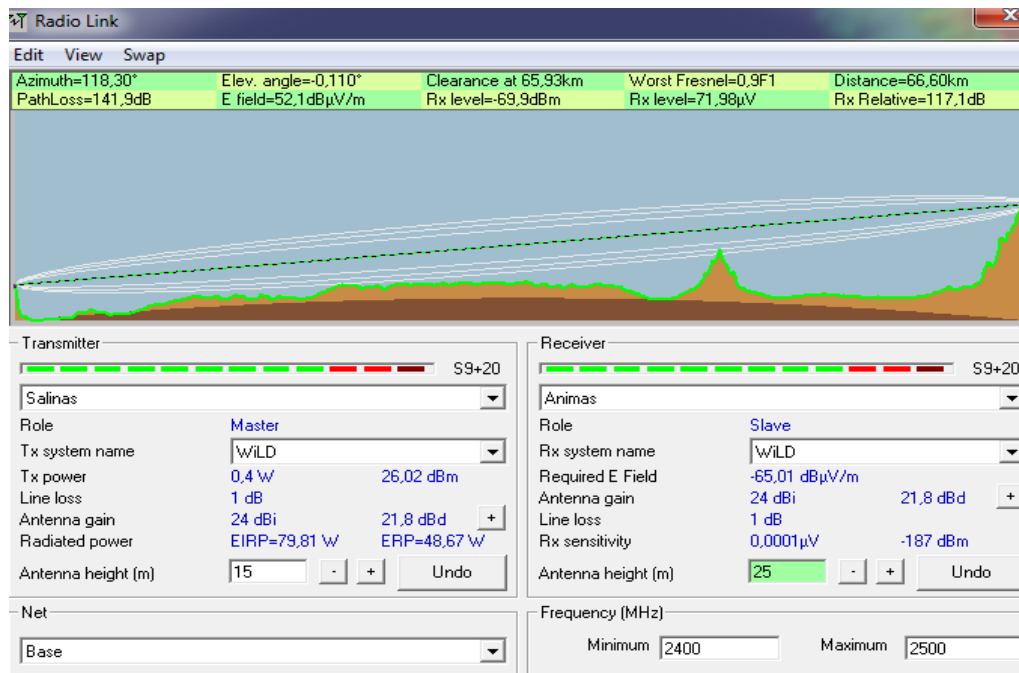


Figura 4.5. Perfil de radio Salinas - Ánimas.

- **Enlace Salinas – Cabuyas**

De manera similar se realiza el esquema de diseño para el enlace entre Salinas y Cabuyas, mostrado en la figura 4.6

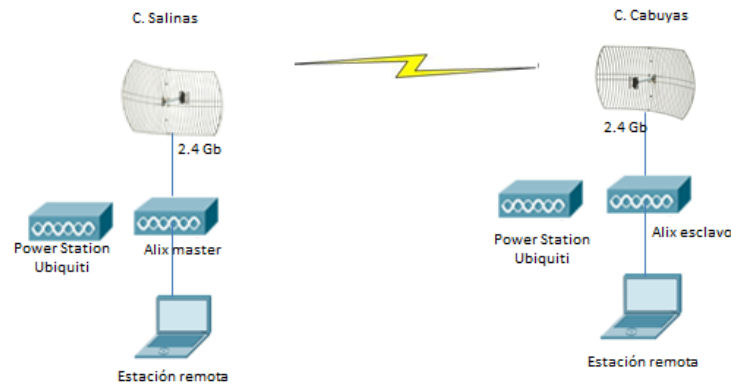


Figura 4.6 Esquema del enlace Salinas Cabuyas.

De igual manera se realiza el proceso de simulación en Radio Mobile, lo que varía son datos de altura del cerro, y el clima, es decir la propagación que es en este caso sobre mar. Para esto el simulador cuenta con modelos de propagación que interpretan muy bien este tipo de climas, se eligió al modelo de *Longley Rice*. Como se explicará más adelante. La figura 4.6, muestra la factibilidad del enlace número 2 y la figura 4.7 muestra los parámetros del enlace de radio.

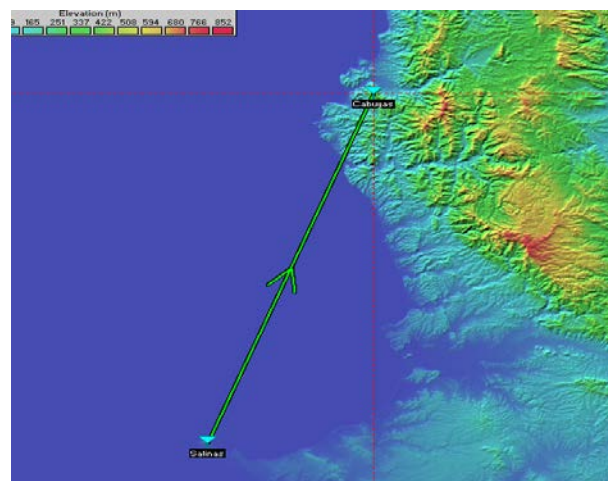


Figura 4.7 Enlace Salinas - Cabuyas en Radio Mobile.

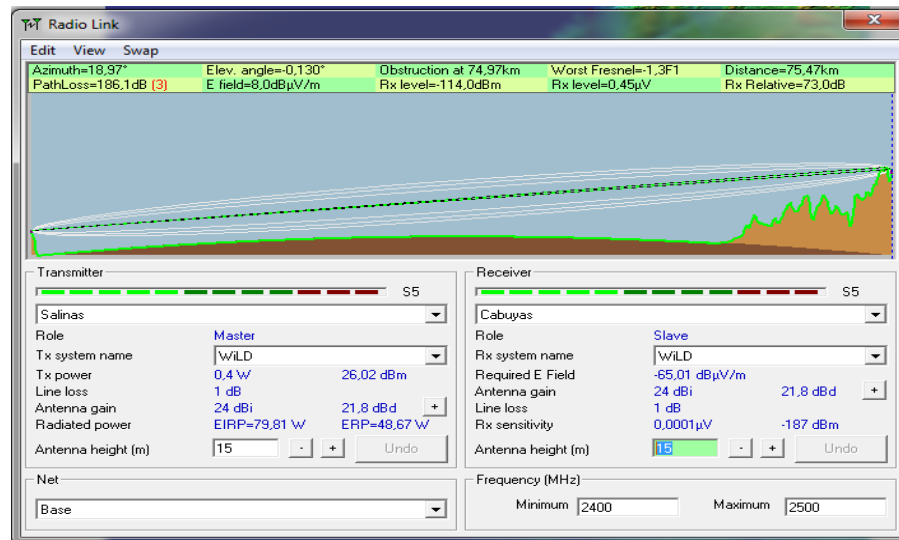


Figura 4.8 Perfil de radio Salinas - Cabuyas.

4.1.3 Modelo de propagación de Longley – Rice (*ITS Irregular Terrain Model*)

El modelo *Longley-Rice* predice la posible propagación a media y larga distancia sobre terreno irregular. Fue diseñado para frecuencias entre los 20 MHz y 20 GHz, para longitud desde trayecto de entre 1 y 2000 Km [34]. Es un modelo estadístico pero toma en cuenta muchos más parámetros para el cálculo de las pérdidas como:

- ✓ Altura media del terreno (ondulación)
- ✓ Refracción de la troposfera
- ✓ Perfiles del terreno
- ✓ Conductividad y permisividad del suelo

Para el cálculo de la propagación, el modelo *Longley-Rice* tiene los siguientes parámetros comunes al de otros modelos de propagación:

- ✓ Frecuencia: el rango de frecuencias nominales para el modelo varía entre 20 MHz y 20 GHz

- ✓ ERP (*Effective Radiated Power*): potencia efectiva de radiación, se introducen en las unidades que se fije en la opción de configuración del sistema (mW, W, kW, dBm, dBW, dBk).
- ✓ Altura de la antena: Altura a la que se sitúa la antena, medida en metros, (sobre el nivel del mar), para transmitir y recibir.

Los parámetros específicos para el modelo de Longley-rice que por su naturaleza se requieren, son:

Polarización: debe especificarse si se trabaja con polarización horizontal o vertical. El modelo de Longley-Rice asume que ambas antenas tienen la misma polarización, vertical y horizontal.

Refractividad: la refractividad de la atmósfera determina la cantidad de “bending” o curvatura que sufrirán las ondas radio. En otros modelos, el parámetro de refractividad puede introducirse como la curvatura efectiva de la tierra, típicamente $4/3$ (1.333).

Para el modelo Longley-Rice, hay tres formas de especificar la refractividad. Se puede introducir el valor de refractividad de superficie directamente, típicamente en el rango de 250 a 400 Unidades de n (correspondiente a valores de curvatura de la tierra de 1,232 a 1,767). Una curvatura efectiva de la tierra de $4/3$ ($K = 1,333$) corresponde a una refractividad de superficie de valor aproximadamente 301 Unidades de n .

Se dice que la onda está en condiciones de $K = 4/3$, que es el valor para una atmósfera estándar, y que de acuerdo a valores experimentales se encontró que éste era el valor medio. De manera que el factor K multiplicado por el radio terrestre da el radio ficticio de la Tierra.

Permitividad y Conductividad: la permitividad relativa o constante dieléctrica del medio (ϵ), tiene unos valores típicos al igual que la conductividad que es medida en Siemens por metro, estos se muestran en la tabla 4.3.

Tabla 4.3 Valores de Conductividad y Permitividad.

	PERMITIVIDAD	CONDUCTIVIDAD
Tierra media	15	0.005
Tierra pobre	4	0.001
Tierra rica	25	0.020
Agua fresca	81	0.010
Agua de mar	81	5.000

Clima: Hay 7 modelos de clima caracterizados en el modelo:

- ✓ Ecuatorial (Congo);
- ✓ Continental Sub-tropical (Sudan y Centro América);
- ✓ Marítimo Sub-tropical (Oeste de las costas de África);
- ✓ Desierto (Sahara);
- ✓ Continental templado;
- ✓ Marítimo Templado, sobre la tierra (Reino Unido y continentes de la costa Oeste);
- ✓ Marítimo Templado, sobre el mar

Variabilidad: el modelo de *Longley-Rice* define cuatro modos de variabilidad. El modo seleccionado determina el significado de la fiabilidad de los valores usados en el modelo. El modo de variabilidad puede ser considerado como la especificación para determinar la fiabilidad de los cálculos. Los modelos de variabilidad definidos son:

- ✓ Modo de mensaje simple.

- ✓ Modo individual.
- ✓ Modo Móvil.
- ✓ Modo *broadcast*.

Los tipos de variabilidad descritos en el modelo Longley-Rice son el tiempo, la posición, y la variabilidad de situación. Estas tres dimensiones de variabilidad, fueron desarrolladas para considerar y clasificar variaciones en los niveles de señal medidos (mediana) La variabilidad de corto plazo del tipo asociado con la propagación de multitrayecto no es cubierta por el modelo.

La pérdida media de propagación se obtiene usando la información sobre la geometría del terreno entre el receptor y el transmisor y los parámetros mencionados antes. Para predecir la potencia de la señal se usa el modelo de 2 rayos. Los parámetros se muestran en la figura 4.9.

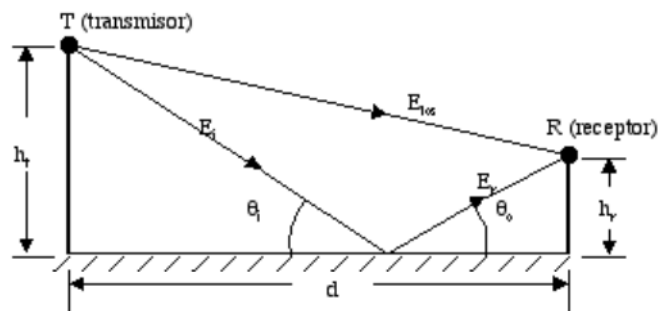


Figura 4.9 Modelo de dos rayos sobre tierra.

- ✓ Altura de la antena Transmisora, h_T y Receptora, h_R .
- ✓ Campo eléctrico de la componente de onda directa, E_{LOS} .
- ✓ Campo eléctrico de la componente de onda reflejada, E_p .
- ✓ Campo eléctrico de la onda incidente sobre la superficie terrestre, E_i .
- ✓ Distancia plana entre la antena Transmisora y la antena Receptora, d .

Para estos enlaces a larga distancia no se puede considerar el modelo de tierra plana, el modelo de 2 rayos también puede trabajar con ello. Y para el enlace sobre mar se tiene en la figura 4.10 los parámetros para tomar en cuenta [34].

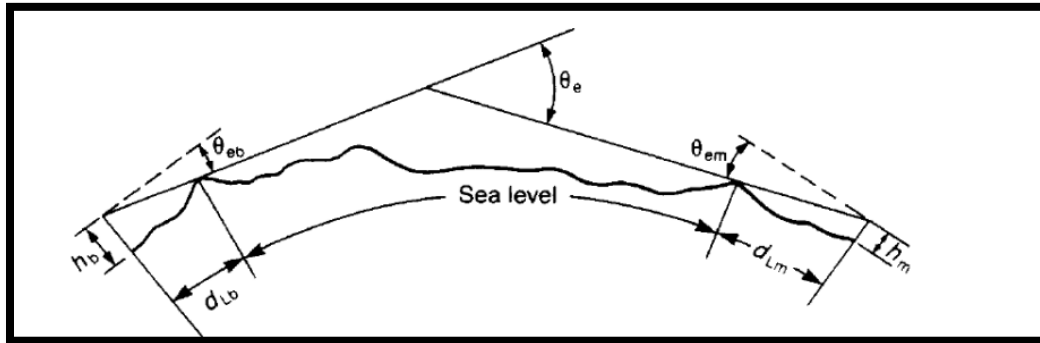


Figura 4.10 Parámetros para enlace sobre mar según Longley - Rice.

- ✓ Altura efectiva de la antena sobre el nivel del mar.
- ✓ Distancia de las antenas sobre el horizonte d_{LB} y d_{LM} .
- ✓ Ángulos de elevación, θ_{cb} y θ_{cm} .
- ✓ Distancia angular para un campo horizontal, θ_c .
- ✓ Parámetro de terreno irregular, Δh .

4.1.4 Cálculo de Propagación

Para realizar la implementación y montaje de los enlaces primero se debe realizar los cálculos necesarios para tener un idea clara de cómo se va a comportar nuestra red una vez implementada y poder brindar una solución óptima a problemas reales.

Para la obtención de los cálculos se debe tomar en cuenta ciertos factores de gran importancia como la potencia de transmisión de las tarjetas, la distancia entre antenas y la longitud y calidad del cable coaxial.

4.1.5 Potencia de transmisión de las tarjetas

Mientras mayor sea la potencia de transmisión, mayor será el alcance del enlace pero no se debe dejar de lado que al incrementar la potencia se corre el riesgo de violar las normas vigentes en el país.

La potencia de transmisión de las tarjetas que se utilizó fue de 19 dBm como se mostró anteriormente en la lista de potencias de la interfaz *ath0*.

4.1.6 Longitud y calidad del cable coaxial

La utilización de cable coaxial es uno de los factores importantes en la instalación de las antenas, debido a que deberá ir conectado desde la antena hasta la ubicación de las tarjetas ALIX. Se debe tomar en cuenta que cuanto más largo sea el cable coaxial, mayor será la pérdida de señal, debido que por cada 1 metro de cable existe 0.1 dB de pérdida.

4.1.7 Distancia entre antenas

La distancia entre antenas es un factor vital en la implementación de una red inalámbrica, ya que al tener una mayor distancia entre transmisor y receptor, mayor también será la pérdida de la señal que se transmite.

La distancia máxima es la del enlace Salinas – Cabuyas con 77.5 Km, mientras que el enlace Salinas – Ánimas es de 66.7 Km.

Una vez conocidas las distancias de los enlaces inalámbricos se realizará el análisis para calcular la pérdida por propagación en cada uno utilizando la ecuación 4.1, como se muestra a continuación.

$$Pp = 40 + 20 \times \text{Log}(d) \quad (4.1)$$

Dónde:

$Pp = \text{Pérdidas por propagación (dB)}$

$d = \text{distancia entre antenas (m)}$

Los resultados de propagación obtenidos para los dos enlaces se indican en la tabla 4.4.

Tabla 4.4 Cálculo de propagación para los enlaces.

Enlace Salinas – Animas	Enlace Salinas – Cabuyas
$d = 66.7\text{Km} = 66700\text{ m}$ (4.2)	$d = 75.5\text{ Km} = 75500\text{ m}$ (4.6)
$Pp = 40 + 20 \times \text{Log}(d)$ (4.3)	$Pp = 40 + 20 \times \text{Log}(d)$ (4.7)
$Pp = 40 + 20 \times \text{Log}(66700)$ (4.4)	$Pp = 40 + 20 \times \text{Log}(75500)$ (4.8)
$Pp = 136.48\text{ dB}$ (4.5)	$Pp = 137.55\text{ dB}$ (4.9)

4.1.8 Nivel de Recepción

Para calcular el nivel de recepción se utiliza la ecuación 4.2, que utiliza los parámetros necesarios para establecer el enlace inalámbrico.

$$Nrs = Pt_a - Pco_a - Pca_a + Ga_a - P_p + Ga_b - Pca_b - Pco_b \quad (4.10)$$

Dónde:

$Nrs = \text{Nivel de la potencia de recepción de la señal}$

$Pt_a = \text{Potencia de Transmisión}$

$Pco_a = \text{Pérdida de Conectores de Antena Transmisora}$

P_{ca_a} = Pérdida de Cable de Antena Transmisora

G_{a_a} = Ganancia de Antena Transmisora

P_p = Pérdida de Propagación

G_{a_b} = Ganancia de Antena Receptora

P_{ca_b} = Pérdida de Conectores de Antena Receptora

P_{co_b} = Pérdida de Cable de Antena Receptora

- **Enlace Salinas – Animas**

Los parámetros que obtuvieron para el primer enlace se detallan en la tabla 4.5, y con estos se obtiene el nivel de recepción en este enlace.

Tabla 4.5 Parámetros de enlace Salinas - Animas.

PARÁMETROS	VALOR
P_{t_a}	79,43 mW
P_{co_a}	1 dB
P_{ca_a}	0,1 dB
G_{a_a}	24 dBi
P_p	136,48 dB
G_{a_b}	24 dBi
P_{ca_b}	0,1 dB
P_{co_b}	1 dB

$$N_{rs} = P_{t_a} - P_{co_a} - P_{ca_a} + G_{a_a} - P_p + G_{a_b} - P_{ca_b} - P_{co_b} \quad (4.11)$$

$$N_{rs} = 19dBm - 1dBm - 0.1dBm + 24dBi - 136.48dB + 24dBi - 0.1dBm - 1dBm \quad (4.12)$$

$$N_{rs} = -71.68dBm \quad (4.13)$$

- **Enlace Salinas – Cabuyas**

Para el segundo enlace los parámetros se detallan en la tabla 4.6, y de igual manera se obtiene el nivel de recepción para éste.

Tabla 4.6 Parámetros de enlace Salinas - Cabuyas.

PARÁMETROS	VALOR
Pt_a	79,43 mW
Pco_a	1 dB
Pca_a	0,1 dB
Ga_a	24 dBi
P_p	137,55 dB
Ga_b	24 dBi
Pca_b	0,1 dB
Pco_b	1 dB

$$Nrs = Pt_a - Pco_a - Pca_a + Ga_a - P_p + Ga_b - Pca_b - Pco_b \quad (4.14)$$

$$Nrs = 19dBm - 1dBm - 0.1dBm + 24dBi - 137.55 dB + 24dBi - 0.1dBm - 1dBm \quad (4.15)$$

$$Nrs = -72.75dBm \quad (4.16)$$

4.1.9 Sensibilidad

Para que el enlace sea óptimo, el valor del Nivel de Recepción de la Señal (Nrs) debe ser mayor que la sensibilidad más el margen que tienen los dispositivos, que suele ser igual al 10% de la sensibilidad. Se debe tomar en cuenta que la sensibilidad en los equipos cambia cuando se modifica la velocidad de transmisión.

En la Tabla 4.7, se indican los valores promedios de sensibilidad con su respectiva velocidad de transferencia para las tarjetas Atheros UBIQUITI de largas distancias, que se utilizaron para desplegar los enlaces.

Tabla 4.7 Valores de sensibilidad vs velocidad.

VELOCIDAD	SENSIBILIDAD
1 Mbps	-97 dBm
2 Mbps	-96 dBm
5.5 Mbps	-95 dBm
6 Mbps	-94 dBm
9 Mbps	-93 dBm
11 Mbps	-92 dBm

Para verificar si los enlaces son viables se remplazan los valores de Nivel de Recepción, Sensibilidad y Margen de los equipos en la siguiente ecuación:

$$Nrs > Sensibilidad + Margen \quad (4.17)$$

Tabla 4.8 Verificación de viabilidad en los enlaces.

Enlace Salinas – Animas	Enlace Salinas – Cabuyas
$-71,68dBm > -95dBm + 9,5dBm$	$-72,75dBm > -95dBm + 9,5dBm$
$-71,68dBm > -85,5dBm$	$-72,75dBm > -85,5dBm$

Como se puede observar en la tabla 4.8, en ambos enlaces se cumple la condición para obtener un enlace viable. También se puede notar que al aumentar la distancia de enlace, esta condición tiende a no cumplirse.

4.1.10 Alineación de las antenas

La tabla 4.9 muestra las coordenadas de los puntos de estudio tratando de que sean lo más exactas posibles con los datos recolectados en la visita al lugar, y también indica la altura total a la que se ubica la antena, es decir, la altura del cerro más la altura de la torre.

Tabla 4.9 Datos reales en cada nodo.

UBICACIÓN	LATITUD	LONGITUD	ALTURA = CERRO + TORRES
C. Salinas	2°11'10.18" S	80°59'34" W	92m + 15m = 107m
C. Ánimas	2°28'12.5" S	80°27'52.8" w	312m + 25m = 337
C. Cabuyas	1°32'38" S	80°46'19" W	322m + 10m = 332m

- **Enlace Salinas – Animas**

Para el cálculo de la diferencia de distancia entre las antenas se utiliza la siguiente expresión:

$$\Delta H = H_2 - H_1 \quad (4.18)$$

Dónde:

$$H_1 = \text{Altura del Cerro Salinas} + \text{Altura de la Torre}$$

$$H_2 = \text{Altura del Cerro Ánimas} + \text{Altura de la Torre}$$

D = Distancia entre antenas , parámetros Indicados en la figura 4.11.

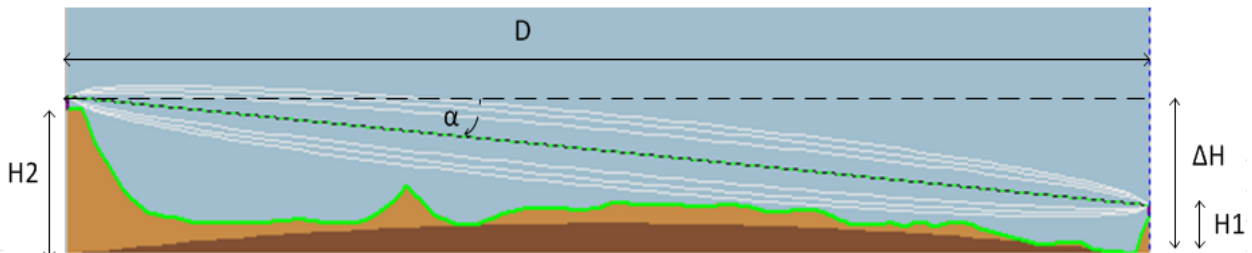


Figura 4.11 Alturas enlace Salinas – Ánimas.

Se reemplaza los valores correspondientes a la Tabla 4.6 en la expresión anterior para calcular la diferencia entre las alturas de las antenas, como sigue:

$$\Delta H = 337(m) - 107(m) \quad (4.19)$$

$$\Delta H = 230(m) \quad (4.20)$$

Utilizando el teorema de Pitágoras y basándose en la Figura xxx, se obtiene el ángulo de elevación.

$$\text{Tan}(\alpha) = \frac{\Delta H}{D} \quad (4.21)$$

$$\alpha = \text{Tan}^{-1} \left(\frac{230(m)}{66700(m)} \right) \quad (4.22)$$

$$\alpha = 0.1975^\circ \quad (4.23)$$

El Azimut se define como el ángulo horizontal que se forma entre la dirección de viaje de una onda electromagnética radiada por una antena y el plano vertical.

Para su medición se hace referencia al Norte como cero grados, y se continúa girando en sentido de las agujas del reloj, hacia el Este corresponden 90° de Azimut, hacia el Sur se tendría 180° de Azimut, hacia el Oeste con 270° , hasta llegar al punto inicial de los 360° que coinciden con los 0° del Norte.

Para determinar el Azimut es necesario determinar el ángulo de apuntamiento, para lo cual se requiere primero la variación de latitud y longitud, que se indica en la tabla 4.10.

Tabla 4.10 Calculo de Variación de latitud y longitud para el primer enlace.

Variación de latitud.	Variación de longitud.
$\Delta_{latitud} = lat_{salinas} - lat_{ánimas} \quad (4.24)$	$\Delta_{longitud} = lon_{salinas} - lon_{ánimas} \quad (4.28)$
$\Delta_{latitud} = 2^\circ 11' 18'' - 2^\circ 28' 12,5'' \quad (4.25)$	$\Delta_{longitud} = 80^\circ 59' 34'' - 80^\circ 27' 52,8'' \quad (4.29)$
$\Delta_{latitud} = -0^\circ 16' 54,8'' \quad (4.26)$	$\Delta_{longitud} = 0^\circ 31' 41,2'' \quad (4.30)$
$\Delta_{latitud} = 0,2818^\circ \quad (4.27)$	$\Delta_{longitud} = 0,5281^\circ \quad (4.31)$

Con estos datos, se calcula el ángulo de apuntamiento:

$$\theta = \tan^{-1} \left(\frac{\Delta_{latitud}}{\Delta_{longitud}} \right) \quad (4.32)$$

$$\theta = \tan^{-1} \left(\frac{0,2818^\circ}{0,5281^\circ} \right) \quad (4.33)$$

$$\theta = -28.09^\circ \quad (4.34)$$

Y, Una vez realizados los cálculos anteriores se procede a obtener el Azimut para cada antena.

El ángulo Azimut Salinas.

$$Az_{Salinas} = 90^\circ + |\theta| \quad (4.35)$$

$$Az_{Salinas} = 90^\circ + 28.09^\circ \quad (4.36)$$

$$Az_{Salinas} = 118.09^\circ \quad (4.37)$$

El ángulo Azimut Ánimas.

$$Az_{\acute{A}nimas} = 270^\circ + |\theta| \quad (4.38)$$

$$Az_{\acute{A}nimas} = 270^\circ + 28.09^\circ \quad (4.39)$$

$$Az_{\acute{A}nimas} = 298.09^\circ \quad (4.40)$$

- **Enlace Salinas – Cabuyas**

Para el cálculo de la diferencia de distancia entre las antenas se utiliza la siguiente expresión:

$$\Delta H = H_2 - H_1 \quad (4.41)$$

Dónde:

$H_1 = \text{Altura del Cerro Salinas} + \text{Altura de la Torre}$

$H_2 = \text{Altura del Cerro Cabuyas} + \text{Altura de la Torre}$

$D = \text{Distancia entre antenas}$, los parámetros se indican en la figura 4.12

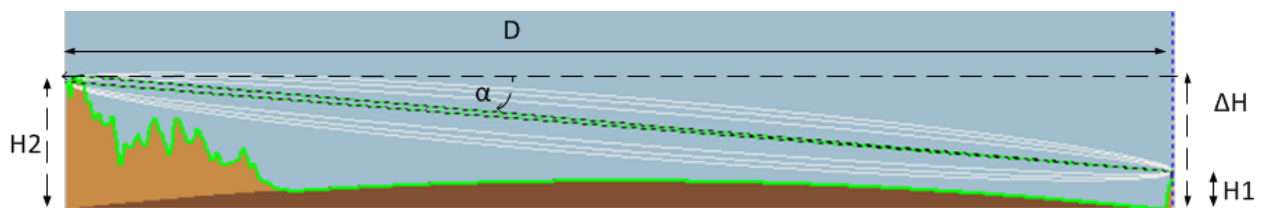


Figura 4.12 Alturas enlace Salinas – Cabuyas.

Igual que en caso anterior, se reemplaza los valores correspondientes a la Tabla 4.6 en la expresión anterior para calcular la diferencia entre las alturas de las antenas, como se muestra:

$$\Delta H = 332(m) - 107(m) \quad (4.41)$$

$$\Delta H = 225(m) \quad (4.42)$$

Utilizando el teorema de Pitágoras y basándose en la Figura 4.12, se obtiene el ángulo de elevación.

$$\text{Tan}(\alpha) = \frac{\Delta H}{D} \quad (4.43)$$

$$\alpha = \text{Tan}^{-1}\left(\frac{225(m)}{66700(m)}\right) \quad (4.44)$$

$$\alpha = 0.19327^\circ \quad (4.45)$$

Se determina a continuación el Azimut para el segundo enlace de manera similar a la realizada en el enlace anterior. Estos valores se detallan en la tabla 4.11.

Tabla 4.11 Cálculo de Variación de latitud y longitud para el segundo enlace.

Variación de latitud.	Variación de longitud.
$\Delta_{latitud} = lat_{Salinas} - lat_{Cabuyas} \quad (4.46)$	$\Delta_{longitud} = lon_{Salinas} - lon_{Cabuyas} \quad (4.50)$
$\Delta_{latitud} = 2^\circ 11' 18'' - 1^\circ 32' 38'' \quad (4.47)$	$\Delta_{longitud} = 80^\circ 59' 34'' - 80^\circ 46' 19'' \quad (4.51)$
$\Delta_{latitud} = 0^\circ 38' 40'' \quad (4.48)$	$\Delta_{longitud} = 0^\circ 31' 15'' \quad (4.52)$
$\Delta_{latitud} = 0,644^\circ \quad (4.49)$	$\Delta_{longitud} = 0,2208^\circ \quad (4.53)$

Se calcula el ángulo de apuntamiento:

$$\theta = \tan^{-1}\left(\frac{\Delta_{latitud}}{\Delta_{longitud}}\right) \quad (4.54)$$

$$\theta = \tan^{-1}\left(\frac{0.644}{0.2208}\right) \quad (4.55)$$

$$\theta = 71.13^\circ \quad (4.56)$$

Con estos valores, nuevamente se procede a obtener el Azimut para cada antena.

El ángulo Azimut Salinas.

$$Az_{Salinas} = 90^\circ + |\theta| \quad (4.57)$$

$$Az_{Salinas} = 90^\circ + 71.13^\circ \quad (4.58)$$

$$Az_{Salinas} = 161.13^\circ \quad (4.59)$$

El ángulo Azimut Cabuyas.

$$Az_{Cabuyas} = 270^\circ + |\theta| \quad (4.60)$$

$$Az_{Cabuyas} = 270^\circ + 71.13^\circ \quad (4.61)$$

$$Az_{Cabuyas} = 341.13^\circ \quad (4.62)$$

4.1.11 Radio de la primera zona de Fresnel

La ecuación que se utiliza para calcular este factor es la siguiente:

$$F1 = \sqrt{\lambda \frac{d1xd2}{d}} \quad (4.63)$$

Se simplifica la fórmula anterior y se tiene como ecuación resultante la siguiente:

$$F1 = 548 \sqrt{\frac{d1xd2}{f*d}} \quad (4.63)$$

Dónde:

$F1 = \text{Radio de la primera zona de Fresnel (m)}$

$d1 =$ Distancia desde la menor altura al obstáculo (Km)

$d2 =$ Distancia desde la menor altural obstáculo (Km)

$d =$ Distancia total del enlace (Km)

$f =$ Frecuencia (MHz)

- **Enlace Salinas – Animas**

Para el cálculo del radio del primer elipsoide de Fresnel en el primer enlace se usa la ecuación 4.63 reemplazando los valores requeridos.

$$F1 = 548 \sqrt{\frac{(66,7-20,82)20,82}{2400 \times 66,7}} \quad (4.64)$$

$$F1 = 4,33 \text{ m} \quad (4.65)$$

En la práctica, es suficiente mantener despejado sólo el 60% de la primera zona de Fresnel. Es decir en este enlace el primer elipsoide está completamente despejado.

- **Enlace Salinas – Cabuyas**

Usando la misma ecuación 4.63 se obtiene el primer elipsoide de Fresnel para el segundo enlace.

$$F1 = 548 \sqrt{\frac{(77,5-7,55)7,55}{2400 \times 77,5}} \quad (4.66)$$

$$F1 = 29,2 \text{ m} \quad (4.67)$$

Se observa que para este enlace el primer elipsoide también está despejado completamente.

4.1.12 Margen Sobre Obstáculo

Otro parámetro para determinar un buen enlace y la correcta altura de las antenas, es el margen sobre el obstáculo.

La ecuación que define este parámetro es la que se muestra a continuación [35]:

$$c = h_1 + \frac{(d-d_1)(h_2-h_1)}{d} - h_{obs} \quad (4.68)$$

Dónde:

c = Margen sobre obstáculo (m)

d_1 = Distancia desde la mayor altura al obstáculo (Km)

d = Distancia total del enlace (Km)

h_s = Altura del obstáculo sobre el nivel del mar (m)

h_1 = Menor altura de las estaciones (m)

h_2 = Mayor altura de las estaciones (m)

- **Enlace Salinas – Animas**

La figura 4.13 muestra la ubicación y detalles del obstáculo más grande entre estos dos cerros, y la tabla 4.12 muestra los datos resumidos de las distancias de los 2 nodos y del obstáculo entre ellos.

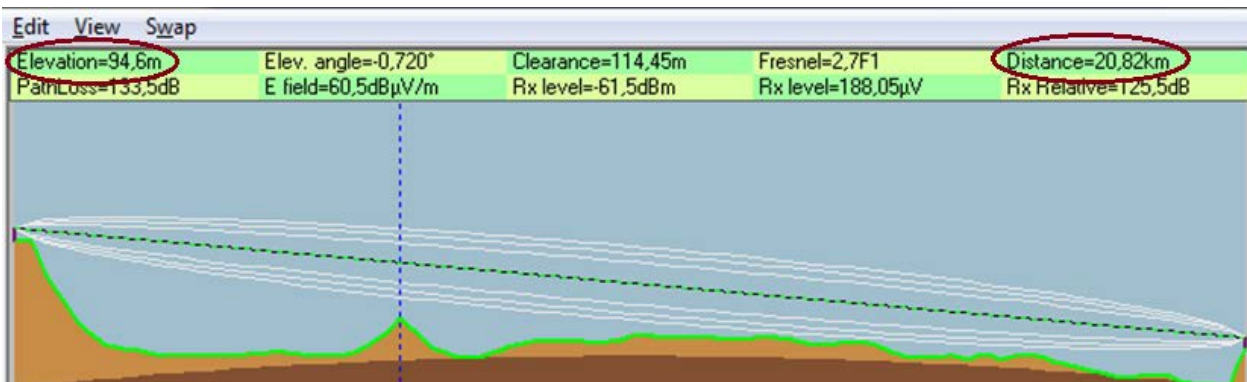


Figura 4.13 Datos del obstáculo más grande para el enlace Salinas- Animas.

Tabla 4.12 Datos resumidos el obstáculo para el enlace Salinas- Animas,

UBICACIÓN	DISTANCIA (Km)	ALTURA (m)
C. ÁNIMAS	0	337
OBSTÁCULO	20,82	94,6
C. SALINAS	66,7	107

Se reemplazan los datos de la tabla 4.12 en la ecuación 4.68.

$$c = 107 + \frac{(66,7-20,82)(337-107)}{66,7} - 94,6 \quad (4.69)$$

$$c = 170,6 \text{ m} \quad (4.70)$$

$$c > 60\% * F1 \quad (4.71)$$

$$170,6 > 0,6 * 42,33 \quad (4.72)$$

$$170,6 > 25,39 \quad (4.73)$$

Al cumplirse la condición anterior se determina que la altura de las antenas es correcta.

- **Enlace Salinas – Cabuyas**

La figura 4.14 muestra la ubicación y detalles del obstáculo más grande entre estos dos cerros, y la tabla 4.13 muestra los datos resumidos de las distancias de los 2 nodos y del obstáculo entre ellos.

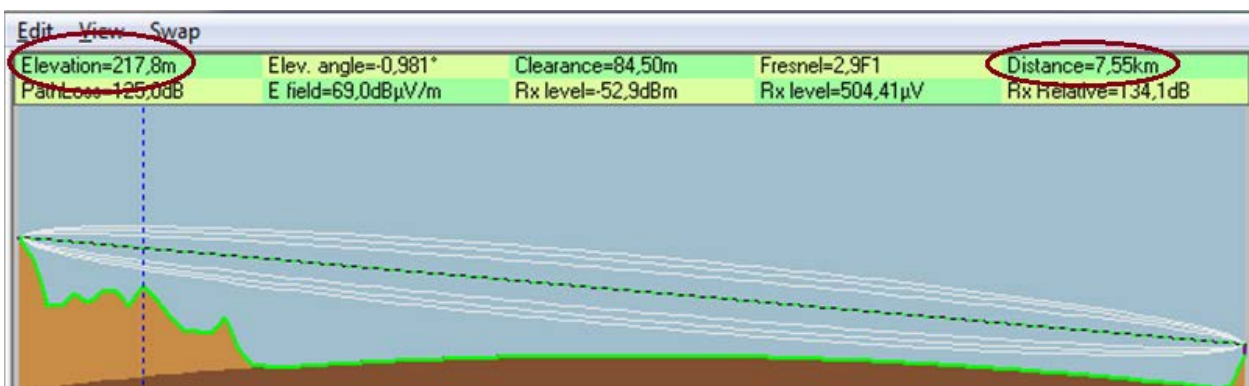


Figura 4.14 Datos del obstáculo más grande para el enlace Salinas- Cabuyas.

Tabla 4.13 Datos resumidos el obstáculo para el enlace Salinas- Cabuyas.

UBICACIÓN	DISTANCIA (Km)	ALTURA (m)
C. ÁNIMAS	0	332
OBSTÁCULO	7,55	217,8
C. SALINAS	77,5	107

De igual manera se reemplaza los datos de la tabla 4.13 en la ecuación 4.68.

$$c = 107 + \frac{(77,5-7,55)(332-107)}{77,5} - 217,8 \quad (4.74)$$

$$c = 92,28 \text{ m} \quad (4.75)$$

$$c > 60\% (F1) \quad (4.76)$$

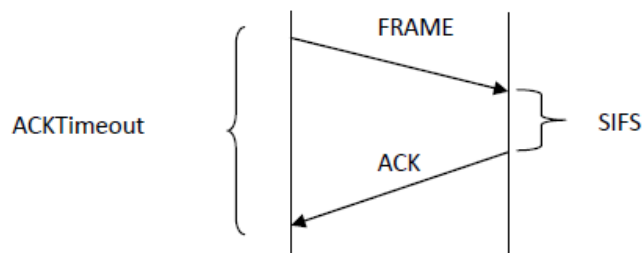
$$92,28 > 0,6 (29,2) \quad (4.77)$$

$$92,28 > 17,52 \quad (4.78)$$

Como resultado se obtiene que se cumple la condición, por lo que para este enlace las alturas de las antenas también son correctas.

4.1.13 Configuración de interfaces para WiLD

Como se menciona en los capítulos 2 y 3 el parámetro necesario para el obtener el enlace a larga distancia es el *ACKTimeout* que es el tiempo que se espera por un ACK tras el envío de una trama de datos como se indica en la figura 4.11. Transcurrido ese tiempo, se retransmite o se abandona.

Figura 4.15. *ACKTimeout*.

El parámetro SIFS es el tiempo que separa la recepción del paquete de la transmisión de su ACK en el receptor. Su valor es de 10 μ s para 802.11b/g y de 16 μ s para 802.11a.

Las tarjetas de *chipset Atheros* suelen contar con un *ACKTimeout* por defecto de 48 μ s definido por el driver *Madwifi*, deficiente para distancias mayores a los 3 Km. En las pruebas realizadas en campo se ha comprobado cómo se mejora sensiblemente las prestaciones del enlace al modificar estos parámetros.

Para calcular este valor se emplea la ecuación 4.79 [35]:

$$ACKTimeout = ACKTimeDefault(\mu s) - SlotTime(\mu s) + 2\delta \quad (4.79)$$

Reemplazando el valor del *ACKTimeout* por defecto en la ecuación 4.7, se obtiene.

$$ACKTimeout = 48(\mu s) - SlotTime(\mu s) + 2\delta \quad (4.80)$$

Dónde:

$SlotTime = 9\mu s$; Para 802.11b

$\delta = \text{tiempo de proagación} = \frac{d}{c}$

$d = \text{distancia (Km)}$

$c = \text{velocidad de la luz} = 300.000.000 \frac{m}{s}$

Entonces se tiene:

$$ACKTimeout = 39(\mu s) + 2\frac{d}{c} \quad (4.81)$$

Se procede a calcular el valor del *ACKTimeout* variando la distancia y utilizando la ecuación 4.81, como se muestra en la Tabla 4.14.

Tabla 4.14 Valores teóricos de *ACKTimeout*.

Distancia (Km)	<i>ACKTimeout</i> Calculado (μ s)
1	45,6
3	59
5	72,3
10	105,6
20	172,3
30	239
40	305,6
50	372,3
60	439
67	485,6
70	505,6
78	559
80	572,3

Al comparar el valor del *ACKTimeout* calculado con el leído se notó que dichos valores son bastante aproximados en cada interfaz.

Con los valores leídos se procede a calcular el tiempo que utiliza realmente cada interfaz en el *ACKTimeout*.

Para la interfaz **ath0** se tiene un *ACKTimeout* por defecto de 30 μ s. Este valor se sustituye en la ecuación 4.84 y variando la distancia encontramos el valor del *ACKtimeout* exacto para esta interfaz. Los resultados se muestran en la tabla 4.15

$$ACKTimeout = ACKTimeDefault(\mu s) - SlotTime(\mu s) + 2\delta \quad (4.82)$$

$$ACKTimeout = 30(\mu s) - 9(\mu s) + 2 \frac{d}{3(10^8)} \quad (4.83)$$

$$ACKTimeout = 21(\mu s) + 2 \frac{d}{3(10^8)} \quad (4.84)$$

Tabla 4.15 Comparación de valores de ACKTimeout en la interfaz ath0.

Distancia (Km)	ACKTimeout leído (μ s)	CtsTimeout leído (μ s)	SlotTime leído (μ s)	ACKTimeout real (μ s)	Error
1	29	29	13	27,6	4,82
3	41	41	19	41	0
5	55	55	26	54,3	1,27
10	89	89	43	87,6	1,57
20	155	155	76	154,3	0,45
30	221	221	109	221	0
40	289	289	143	287,6	0,48
50	355	355	176	354,3	0,19
60	421	421	209	421	0
67	469	469	233	467,6	0,29
70	489	489	243	487,6	0,28
78	541	541	270	541	0
80	555	555	276	554,3	0,12

Para la interfaz **ath1** se tiene un *ACKTimeout* por defecto de 39 μ s. Este valor se sustituye en la ecuación 4.84 y variando la distancia encontramos el valor del *ACKtimeout* exacto para esta interfaz. Los resultados se ven en la tabla 4.16

Tabla 4.16 Comparación de valores de ACKTimeout en la interfaz ath1.

Distancia (Km)	ACKTimeout leído (μ s)	CtsTimeout leído (μ s)	SlotTime leído (μ s)	ACKTimeout real (μ s)	Error
1	36	36	26	36,6	1,66
3	50	50	40	50	0
5	63	63	53	63,3	0,47
10	96	96	86	96,6	0,62
20	163	163	153	163,3	0,18
30	230	230	220	230	0
40	297	297	287	296,6	0,13
50	364	364	354	363,3	0,19
60	431	431	421	430	0,23
67	478	478	468	476,6	0,29
70	498	498	488	496,6	0,28
78	551	551	541	550	0,18
80	565	565	555	563,3	0,30

Se puede notar ahora que la diferencia entre valores leídos y calculados de *ACK-timeout* es mínima para las dos interfaces.

Los valores de *ACK-timeout* y *CTS-timeout* para cada distancia son iguales, esto se debe a que los dos forman parte de un tiempo de respuesta, difiriendo que el ACK es un acuse de recibo que indica que el destinatario recibió el paquete y el CTS es una trama que envía también el receptor para indicar que no se está efectuando otro proceso de transmisión y que el canal está libre para poder comunicar a los dos nodos.

4.1.14 Simulación con el Software *ns-2*

ns-2 presenta 3 Modelos de Propagación, cada uno está diseñado para cierto tipo de ambiente ya sea para medios con problemas de refracción, urbanos o de climatización diferente, a su vez cada modelo de propagación utiliza diferentes parámetros que se acoplan más al tipo de enlace que se desea desplegar. Los modelos de propagación que existen en *ns-2* son:

- ✓ Two Ray Ground Reflection
- ✓ Shadowing
- ✓ Free Space

A continuación se va a detallar los cálculos respectivos para los dos primeros modelos de propagación, ya que están relacionados con el perfil de los enlaces realizados, y la comparación de los mismos con los datos que brinda la herramienta que integra *ns-2* llamada THRESHOLD.

Threshold.cc es un programa que sirve para calcular los umbrales de recepción de la señal, utilizando la relación entre varios parámetros físicos relacionados con las transmisiones en *ns-2*, como la distancia, altura de las antenas,

modelo de propagación, etc. Esto se debe a que ciertos parámetros como la altura de las antenas no se configura directamente desde el script de simulación.

Para lograr utilizar la herramienta Threshold que en sí es un ejecutable es necesario seguir los pasos siguientes:

Se ingresa por terminal al directorio donde se encuentra el archivo `threshold.cc`, para evitar problemas de permisos es recomendable entrar al directorio como `root`, el archivo generalmente se ubica en `~/ns-allinone-2.1b9a/ns-2.1b9a/indep-utils/propagation`. Luego se procede a abrir el archivo con el editor de texto.

Una vez que se ha abierto el archivo `threshold.cc` se elimina la línea:

```
#include <iostream.h>
```

Y en su lugar colocamos las siguientes 3 líneas:

```
#include <iostream>
#include <cstring>
using namespace std;
```

Esto se realiza para evitar problemas con la librería `iostream.h` que no se encuentra actualizada para las nuevas versiones de Ubuntu y el compilador `g++` (V4.3.3).

Se compila `threshold.cc` como se muestra en la figura 4.15 para obtener el archivo ejecutable el cuál se utiliza para calcular el valor de los umbrales de recepción de la señal.

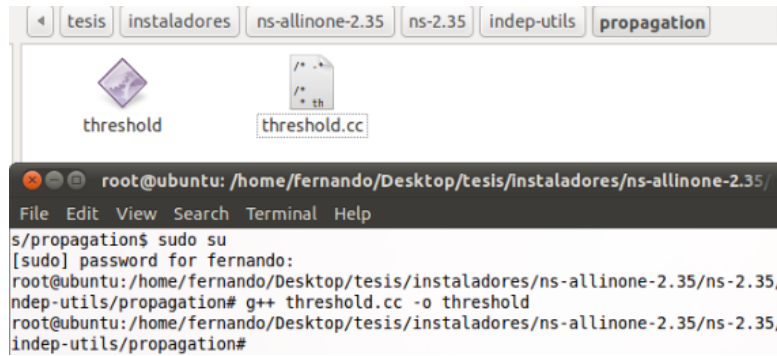


Figura 4.16 Compilación del archivo threshold.cc

Para obtener el valor del threshold se ingresa por terminal los parámetros del enlace como indica en la siguiente línea:

```
$ threshold -m TwoRayGround 66700 -L 1 -Pt 0.2818 -Gr 24 -Gt 24 -ht 1.5 -hr 1.5
```

La tabla 4.17. Ilustra los valores que se pueden modificar para obtener el valor del umbral de recepción de poder, que es el encargado de discriminar si el paquete llega con la potencia suficiente.

Tabla 4.17 Parámetros modificables para el umbral.

COMANDO	DESCRIPCIÓN
-m	Modelo de Propagación
-pl	Exponente de Pérdidas de Propagación (dB)
-fr	Frecuencia (Hz)
-L	Pérdida por Propagación (dB)
-d0	Distancia de referencia (m)
-std	Desviación de Shadowing (dB)
-Gt	Ganancia de antena transmisora (dBi)
-Gr	Ganancia de antena receptora (dBi)
-Pt	Potencia de Transmisión (W)
-ht	Altura de antena transmisora (m)
-hr	Altura de antena receptora (m)

Una vez que se obtiene el valor del umbral de recepción de poder, es necesario colocar este valor en el script de simulación para tener valores más reales y acordes los requerimientos del enlace a desplegar, ya que este valor viene definido por defecto en *ns-2* y es de $3.652e-10$.

Esto se puede verificar abriendo el archivo *ns-default.tcl* que se encuentra en el directorio *~ns/tcl/lib/*.

Se puede ver en la siguiente línea, que se encuentra *seteado* con el valor que se mencionó anteriormente.

Phy/WirelessPhy set RXThresh_ $3.652e-10$

En la Tabla 4.18 se muestra los valores por defecto de *ns-2*; en las tablas 4.19 y 4.20 se indica los valores para el enlace Salinas – Ánimas y del enlace Salinas – Cabuyas respectivamente para calcular el valor de umbral de recepción de poder para colocarlo en el script de simulación, para esto se utiliza las ecuaciones propias de cada tipo de modelo de propagación.

Tabla 4.18 Valores por defecto de *ns-2*.

PARÁMETRO	VALOR
D	250 m
λ	0,125 m
L	1 Db
G_{RX}	1 dBi
G_{TX}	1 dBi
P_{TX}	0,2818 W
h_{RX}	1,5 m
h_{TX}	1,5 m
d_0	1 m
β	2 dB

Tabla 4.19 Valores por defecto de ns-2 para Salinas - Animas.

PARÁMETRO	VALOR
d	66700 m
λ	0,125 m
L	136,48 dB
G_{RX}	24 dBi
G_{TX}	24 dBi
P_{TX}	0,07943 W
h_{RX}	10 m
h_{TX}	15 m
d_0	1 m
β	2 dB

Tabla 4.20 Valores por defecto de ns-2 para Salinas - Animas.

PARÁMETRO	VALOR
d	77500 m
λ	0,125 m
L	137,55 dB
G_{RX}	24 dBi
G_{TX}	24 dBi
P_{TX}	0,07943 W
h_{RX}	25 m
h_{TX}	15 m
d_0	1 m
β	2 dB

- **Modelo de Propagación *Two-Ray Ground Reflection***

La ecuación para el modelo de propagación es la siguiente [35]:

$$P_{RX}(d) = \frac{(P_{TX})(G_{TX})(G_{RX})(h_{TX}^2)(h_{RX}^2)}{(d^4) L} \quad (4.85)$$

Dónde:

d = Distancia total del enlace (m)

h_{RX} = Altura de la antena de recepción (m)

h_{TX} = Altura de la antena de transmisión (m)

$L =$ Factor de pérdidas por propagación (dB)

$G_{RX} =$ Ganancia de la antena de recepción (dBi)

$G_{TX} =$ Ganancia de la antena de transmisión (dBi)

$P_{TX} =$ Potencia de transmisión (W)

$P_{RX}(d) =$ Potencia recibida en función de la distancia (W)

Valores por defecto ns-2

Usando la ecuación 4.85 y remplazándolos con los valores de la tabla 4.18 se calcula el valor umbral por defecto del simulador.

$$P_{RX}(d) = \frac{(0.2818)(1)(1)(1.5)^2(1.5)^2}{(250)^4(1)} \quad (4.86)$$

$$P_{RX}(d) = 3,65262 \times 10^{-10} \text{ W} \quad (4.87)$$

En la figura 4.16 se indica cómo deben ser ingresados éstos parámetros en el simulador. Y como éste nos entrega el valor umbral.

```
root@ubuntu: /home/fernando/Desktop/2.30/ns-allinone-2.30/ns-2.30/indep-utils/propa
gation# ./threshold -m TwoRayGround -Pt 0.2818 -hr 1.5 -ht 1.5 -Gr 1 -Gt 1 250
distance = 250
propagation model: TwoRayGround

Selected parameters:
transmit power: 0.2818
frequency: 9.14e+08
transmit antenna gain: 1
receive antenna gain: 1
system loss: 1
transmit antenna height: 1.5
receive antenna height: 1.5

Receiving threshold RXThresh is: 3.65213e-10
```

Figura 4.17 Ingreso de datos y obtención del valor de Threshold.

Enlace Salinas – Animas

Usando la ecuación 4.85 y remplazando con los valores de la tabla 4.19 se calcula el valor umbral para el primer enlace. La figura 4.19 muestra el proceso en el simulador.

$$P_{RX}(d) = \frac{(0,07943)(24)(24)(15)^2(25)^2}{(66700)^4(136,48)} \quad (4.88)$$

$$P_{RX}(d) = 2,38175 \times 10^{-15} W \quad (4.89)$$

```

root@ubuntu:/home/fernando/Desktop/2.30/ns-allinone-2.30/ns-2.30/indep-utils/propa
gation# ./threshold -m TwoRayGround -Pt 0.07943 -hr 15 -ht 25 -Gr 24 -Gt 24 -L 1
36.48 66700
distance = 66700
propagation model: TwoRayGround

Selected parameters:
transmit power: 0.07943
frequency: 9.14e+08
transmit antenna gain: 24
receive antenna gain: 24
system loss: 136.48
transmit antenna height: 25
receive antenna height: 15

Receiving threshold RXThresh_ is: 2.38176e-15

```

Figura 4.18 Threshold Salinas – Animas TwoRayGround.

Enlace Salinas – Cabuyas

Usando la ecuación 4.85 y remplazado con los valores de la tabla 4.20 se calcula el valor umbral para el primer enlace. La figura 4.18 muestra el proceso en el simulador.

$$P_{RX}(d) = \frac{(0,07943)(24)(24)(15)^2(10)^2}{(77500)^4(137,55)} \quad (4.90)$$

$$P_{RX}(d) = 2,07454 \times 10^{-16} W \quad (4.91)$$


```

root@ubuntu:/home/fernando/Desktop/2.30/ns-allinone-2.30/ns-2.30/indep-utils/propa
gation# ./threshold -m TwoRayGround -Pt 0.07943 -hr 15 -ht 10 -Gr 24 -Gt 24 -L 1
37.55 77500
distance = 77500
propagation model: TwoRayGround

Selected parameters:
transmit power: 0.07943
frequency: 9.14e+08
transmit antenna gain: 24
receive antenna gain: 24
system loss: 137.55
transmit antenna height: 10
receive antenna height: 15

Receiving threshold RXThresh is: 2.07454e-16

```

Figura 4.19 Threshold Salinas – Cabuyas TwoRayGround.

- **Modelo de Propagación *Shadowing***

La ecuación para el modelo de propagación es la siguiente [35]:

$$\left(\frac{P_{RX}(d)}{P_{RX}(d_0)}\right) = -10 \times \beta \times \log\left(\frac{d}{d_0}\right) + X_{dB} \quad (4.92)$$

Dónde:

β = Constante definida como Path Loss Exponent (dB)

d = Distancia total del enlace (m)

d_0 = Distancia de referencia (m)

X_{dB} = Variable randómica con media igual a cero y una desviación estándar σ_{dB}

$\left(\frac{P_{RX}(d)}{P_{RX}(d_0)}\right)$ = Potencia recibida en función de la distancia (dB)

Las tablas 4.21 y 4.22 listan los valores típicos de este modelo de propagación para perdidas de propagación y para desviación estándar respectivamente.

Tabla 4.21. Valores típicos para Exponente de Pérdidas de Propagación.

AMBIENTES EXTERIORES	β
Espacio Libre	2
Área urbana con sombra	2,7 a 5,0
DENTRO DEL EDIFICIO	β
Línea de vista	1,6 a 1,8
Obstrucción	4,0 a 6,0

Tabla 4.22 Valores típicos de la Desviación Estándar σ_{dB} .

AMBIENTE	$\sigma_{dB}(dB)$
Ambientes exteriores	4,0 a 12
Oficina con alta ocupación	7
Oficina con baja ocupación	9,6
Industrial con Línea de vista	3,0 a 6,0
Industrial con obstáculos	6,8

Valores por defecto $ns-2$

De manera similar se utiliza la ecuación 4.93 y se reemplaza con los valores de las tablas 4.17, 4.20 y 4.21 para calcular el valor umbral por defecto del simulador. La figura 4.22 indica el valor obtenido en el simulador.

$$\left(\frac{P_{RX}(d)}{P_{RX}(d_0)}\right) = -10 \times 2 \times \log\left(\frac{250}{1}\right) + X_{dB} \quad (4.93)$$

$$\left(\frac{P_{RX}(d)}{P_{RX}(d_0)}\right) = -47,95 + X_{dB} (dB) \quad (4.94)$$

```

root@ubuntu:/home/fernando/Desktop/2.30/ns-allinone-2.30/ns-2.30/indep-utils/propa
gation# ./threshold -m Shadowing -Pt 0.2818 -hr 1 -ht 1 -Gr 1 -Gt 1 -L 1 -fr 2.4
e9 250
distance = 250
propagation model: Shadowing

Selected parameters:
transmit power: 0.2818
frequency: 2.4e+09
transmit antenna gain: 1
receive antenna gain: 1
system loss: 1
path loss exp.: 2
shadowing deviation: 4
close-in reference distance: 1
receiving rate: 0.95

Receiving threshold RXThresh_is: 9.80663e-11

```

Figura 4.20 Threshold por defecto Con Shadowing.

Enlace Salinas – Animas

Se utiliza la ecuación 4.93 y se reemplaza con los valores de las tablas 4.18, 4.20 y 4.21 para calcular el valor umbral por defecto del simulador. La figura 4.21 indica el valor obtenido en el simulador.

$$\left(\frac{P_{RX}(d)}{P_{RX}(d_0)}\right) = -10 \times 2 \times \log\left(\frac{66700}{1}\right) + X_{dB} \quad (4.95)$$

$$\left(\frac{P_{RX}(d)}{P_{RX}(d_0)}\right) = -96,48 + X_{dB} \text{ (dB)} \quad (4.96)$$

```

root@ubuntu:/home/fernando/Desktop/2.30/ns-allinone-2.30/ns-2.30/indep-utils/propa
gation# ./threshold -m Shadowing -Pt 0.07943 -hr 10 -ht 15 -Gr 24 -Gt 24 -L 136.
48 -fr 2.4e9 66700
distance = 66700
propagation model: Shadowing

Selected parameters:
transmit power: 0.07943
frequency: 2.4e+09
transmit antenna gain: 24
receive antenna gain: 24
system loss: 136.48
path loss exp.: 2
shadowing deviation: 4
close-in reference distance: 1
receiving rate: 0.95

Receiving threshold RXThresh_is: 1.63887e-15

```

Figura 4.21 Threshold Salinas – Animas con Shadowing.

Enlace Salinas – Cabuyas

Se utiliza la ecuación 4.83 y se reemplaza con los valores de las tablas 4.19, 4.20 y 4.21 para calcular el valor umbral por defecto del simulador. La figura 4.24 indica el valor obtenido en el simulador.

$$\left(\frac{P_{RX}(d)}{P_{RX}(d_0)}\right) = -10 \times 2 \times \log\left(\frac{77500}{1}\right) + X_{dB} \quad (4.97)$$

$$\left(\frac{P_{RX}(d)}{P_{RX}(d_0)}\right) = -97.78 + X_{dB} \text{ (dB)} \quad (4.98)$$

```

root@ubuntu:/home/fernando/Desktop/2.30/ns-allinone-2.30/ns-2.30/indep-utils/propa
gation# ./threshold -m Shadowing -Pt 0.07943 -hr 25 -ht 15 -Gr 24 -Gt 24 -L 137.
55 -fr 2.4e9 77500
distance = 77500
propagation model: Shadowing

Selected parameters:
transmit power: 0.07943
frequency: 2.4e+09
transmit antenna gain: 24
receive antenna gain: 24
system loss: 137.55
path loss exp.: 2
shadowing deviation: 4
close-in reference distance: 1
receiving rate: 0.95

Receiving threshold RXThresh_is: 1.20449e-15

```

Figura 4.22 Threshold Salinas – Cabuyas con Shadowing.

No se realizan los cálculos completos para el modelo de propagación de Shadowing debido a que en su ecuación incluye una variable randómica que posee una media igual a cero y una desviación estándar, cuyos valores típicos se muestran en la Tabla 4.22.

Por lo tanto, la propagación Shadowing amplía su concepto y permite considerar un canal real debido a la utilización de un modelo probabilístico y la inclusión de la distribución Gaussiana.

Se nota claramente que los valores obtenidos del archivo ejecutable Threshold y los teóricos son prácticamente iguales, por lo tanto se puede utilizar cualquiera de estos valores para ubicarlo en el script de simulación.

4.1.15 Simulación con *ns-2* Mejorado (TeNS)

Para la simulación en largas distancias el *ns-2* en sí es insuficiente, debido a que este simulador fija una constante llamada *DSSS Propagation Delay* en 2 μ s, que corresponde al tiempo de propagación en un sólo sentido. El alcance máximo por lo tanto es de 600 m, aunque al tratarse de un programa de fuente abierta se puede modificar esa constante y recompilar, adaptándolo a cualquier distancia.

No se puede variar esa constante en tiempo de ejecución a través del script de simulación.

Para un enlace a larga distancia es necesario el parche TeNS (The Enhanced Network Simulator) que trata de abordar las deficiencias de *ns-2* en el modelamiento de protocolo IEEE 802.11 de la capa MAC, también incorpora características adicionales como soporte de múltiples interfaces de los nodos móviles, un protocolo de enrutamiento estático para los escenarios móviles, y también cuenta con la inclusión de simples antenas direccionales. Como se menciona en el capítulo 2.

A continuación se va a detallar las partes más importantes del script para enlaces de larga distancia que se utilizó, es importante notar que la sintaxis de TeNS es parecida a la del *ns-2* básico pero no es igual, en este script se puede ver la inclusión de antenas direccionales a diferencia del script básico de *ns-2* que sólo permite antenas omnidireccionales, además implementa la interferencia co-canal que se debe tener en cuenta durante la configuración de los canales para las interfaces.

WLSTATIC es el nuevo protocolo de enrutamiento que incluye el simulador *ns-2* una vez que se ha complementado con el parche TeNS, se utilizó este protocolo ya que a diferencia otros presentes en el *ns-2*, WLSTATIC hace referencia a una red

en un ambiente inalámbrico y es generado con ciertos parámetros configurados por el administrador como el número de saltos, el siguiente salto, el nodo de destino y la interfaz usada. Además las rutas estáticas tienen como característica que no requieren la misma cantidad de procesamiento y sobrecarga que requieren los protocolos de enrutamiento dinámico.

```
#####  
#This is a sample Tcl script file for using with TeNS  
#####  
  
#Seteo del tipo de canal  
  
set val(chan)      Channel/Channel_802_11  
  
# Modelo de propagación  
  
set val(prop)      Propagation/Shadowing  
  
# Tipo de interfaz de la red que se va a utilizar  
  
set val(netif)     Phy/WirelessPhy/Wireless_802_11_Phy  
  
# Con TeNS ahora si se puede utilizar una antena direccional  
  
set val(ant)       Antenna/DirAntenna  
  
#Número de nodos en la red.  
  
set val(nn)        2  
  
#Protocolo de enrutamiento WLSTATIC, también se utiliza AODV  
  
set val(rp)        WLSTATIC  
  
#Tipo de modulación addition  
  
set opt(mod)       Modulation/BPSK  
  
#####  
  
# Se inicializan las variables globales y se crea el objeto simulador  
  
set ns_            [new Simulator]
```

```

# Abertura del archivo de texto para un análisis posterior
set tracefd [open simple1.tr w]

$ns_ trace-all $tracefd

# Abertura del trace de NAM y se define un área de 10000*10000.

set namtrace [open sim12.nam w]

$ns_ namtrace-all-wireless $namtrace 10000 10000

#####

# Se define la velocidad de transmisión, la tasa básica a la que trabajará el enlace y
el umbral de recepción de poder.

Phy/WirelessPhy set RXThresh_ 1.6388e-15

Mac/802_11 set dataRate_      5.5mb

Mac/802_11 set basicRate_     5.5mb

#####

# Se crea un agente UDP, luego se adjuta un generador de tráfico CBR al agente
UDP. Por CBR se entiende como la tasa constante de bits. El paquete se fija a 1500
bytes y un paquete será enviado cada 0.005 segundos (200 paquetes por segundo).

proc create_cbr_connection { from to startTime interval pACKetSize } {

    global ns_
    set udp0 [new Agent/UDP]
    set src [new Application/Traffic/CBR]
    $udp0 set rate_ 5.5mb
    $udp0 set pACKetSize_ 1500
    $src set pACKetSize_ 0.005
    $src set interval_ $interval

    set sink [new Agent/Null]

    $ns_ attach-agent $from $udp0
    $src attach-agent $udp0
    $ns_ attach-agent $to $sink

    $ns_ connect $udp0 $sink

```

```

    $ns_ at $startTime "$src start"
    return $udp0
}

```

```
#####
```

Creación de un nodo la asociación a una interfaz, el nodo va a estar ubicado en coordenadas x, y ,z, de acuerdo a la distancia que se desee *setear* a los nodos. Se añade al nodo la potencia de transmisión (W) y el canal en el que se va a trabajar para evitar la interferencia de co- canal en cada interfaz.

```
$ns_ node-config -numif 1
```

```
set node_(0) [create_node 10 66710 0]
```

```
[$node_(0) set netif_(0)] set channel_number_ 1
```

```
[$node_(0) set netif_(0)] set Pt_ 0.07943
```

Además adjunta el tipo de antena que se va a utilizar, en este caso se trabaja con Tipo 1 que hace referencia a la antena Hyper Link hg2424g que trabaja a 2.4 Gz y que es la que se usó en la implementación real.

```
set a [new Antenna/DirAntenna]
```

```
$a setType 1
```

La antena también trabaja como direccional, que impone una simulación más real, se puede configurar el ángulo inicial de la antena como se muestra.

```
$a setAngle 90
```

```
[$node_(0) set netif_(0)] dir-antenna $a
```

```
#####
```

Se indica a los nodos que la simulación ha terminado

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

```
    $ns_ at 110.0 "$node_($i) reset";
```

```
}
```

```
#####
```

Configuración de los temporizadores e inicio de la simulación.


```

$ns_ at 110.0 "stop"

$ns_ at 110.01 "puts \"NS EXITING...\" ; $ns_ halt"

proc stop {} {
    global ns_ tracefd

    $ns_ flush-trace

    close $tracefd

    #close $par
}

puts "Starting Simulation..."

$ns_ run
#####

```

Los tipos de antenas que hace referencia el *ns-2* mejorado son los siguientes:

- ✓ Tipo 1 = HG2424G
- ✓ Tipo 2 = HG2414P
- ✓ Tipo 3 = HG2418P
- ✓ Tipo 4 = HG2417P-090
- ✓ Tipo 5 = HG2417P-120

En las simulaciones realizadas en *ns-2* las capas de transporte y de aplicación fueron distribuidas de la siguiente manera, se utilizó CBR sobre UDP y FTP sobre TCP como se muestra en la tabla 4.23:

Tabla 4.23 Parámetros de simulación.

CAPA DE TRANSPORTE	CAPA DE APLICACIÓN
TCP	FTP
UDP	CBR

4.1.16 *Trace Converter*

Trace Converter es una herramienta la cual simplifica la sintaxis de las trazas generadas por *ns-2* para hacerlas entendibles dentro de un entorno gráfico denominado *Trace Graph*, el cuál va a ser detallado posteriormente.

Cabe mencionar que *Trace Graph* trabaja en conjunto con *ns-2*, y a su vez *Trace Graph* trabaja en conjunto con *Trace Converter*.

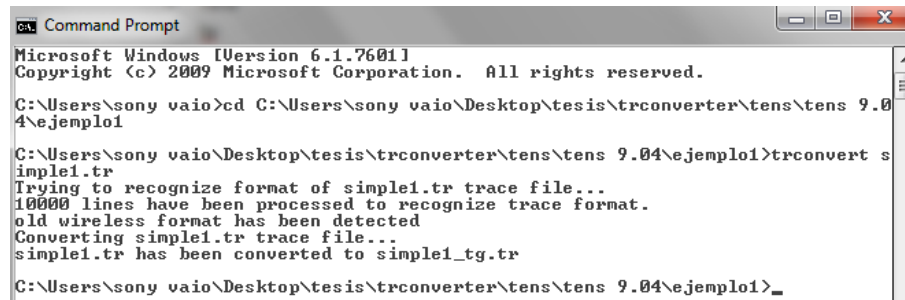
Por otro lado, como *ns-2* genera distintos tipos de trazas, entre ellas tenemos:

- ✓ batch
- ✓ old
- ✓ new
- ✓ lp
- ✓ ip FN
- ✓ hex
- ✓ mixed12
- ✓ mixed13

Para realizar la conversión se debe ubicar en el directorio donde se encuentre el archivo generado por el *ns-2* mejorado y copiar el archivo ejecutable *trconverter.exe*.

Luego se procede a ingresar al directorio donde se encuentran los dos archivos anteriores, esto se realiza mediante *cmd*.

Se procede a convertir el archivo de traza que tiene por nombre *simple1.tr*, se utiliza el comando *-batch*, que es uno de los tipos de trazas permitidas en *ns-2*. En la Figura 4.23 se muestra la línea de comando que se utilizó para la conversión del archivo *simple1.tr*



```

Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\sony vaio>cd C:\Users\sony vaio\Desktop\tesis\trconverter\tens\tens 9.04\ejemplo1
C:\Users\sony vaio\Desktop\tesis\trconverter\tens\tens 9.04\ejemplo1>trconvert simple1.tr
Trying to recognize format of simple1.tr trace file...
10000 lines have been processed to recognize trace format.
old wireless format has been detected
Converting simple1.tr trace file...
simple1.tr has been converted to simple1_tg.tr
C:\Users\sony vaio\Desktop\tesis\trconverter\tens\tens 9.04\ejemplo1>

```

Figura 4.23 Conversión del archivo simple1.tr

Una vez que se realiza el cambio de formato, el nuevo archivo adquiere la misma extensión pero con una adición en su nombre original, en este caso el archivo resultante tiene el nombre simple1_tg.tr como se muestra en la Figura 4.26.

Name	Date modified	Type	Size
ejel	7/4/2012 11:44 AM	File folder	
ejemplo1	5/4/2012 3:25 PM	TCL File	6 KB
simple1	7/4/2012 10:55 AM	TR File	10,207 KB
simple1_tg	7/8/2012 9:49 AM	TR File	5,416 KB
trconvert	10/8/2007 6:29 PM	Application	252 KB

Figura 4.24 Creación del archivo simple1_tg.tr

Con el archivo ya convertido se puede trabajar dentro de la interfaz gráfica de *Trace Graph*.

4.1.17 Trace Graph

La gran problemática que ha existido en el ámbito de analizar los resultados generados por las simulaciones, es que los graficadores utilizados como Xgraph, no soportan en su estado natural los distintos tipos de trazas que genera *ns-2*, sin existir tampoco ningún tipo de convertidor de formato de archivos automático. Por esta razón en este proyecto se hace uso del graficador *Trace Graph* y del conversor de trazas *Trace Converter*.

Además *Trace Graph* ofrece una gran variedad de análisis en parámetros como Throughput, paquetes perdidos, paquetes enviados, recibidos, delay, jitter, etc.

4.1.18 Simulación del Enlace Salinas –Ánimas

Con todas las premisas vistas anteriormente, se procede a simular los enlaces planificados, en el *ns-2* con el parche TeNS. Para el primer enlace entre Salinas y Ánimas en la tabla 4.24 se detallan todos los parámetros necesarios para incluir en el script del TeNS, el código completo del programa para este enlace se detalla en el anexo 7.

Tabla 4.24 Parámetros Seteados en *ns-2* Mejorado para Salinas - Animas.

PARÁMETRO	VALOR
Modelo de Propagación	Shadowing
Tipo de Canal	Wireless Channel
Network Interface	Wireless 802.11
MAC	MAC 802.11
Tipo de Antena	Direccional
Número de Nodos	2
Máximo Número de Paquetes	50
Protocolo de Enrutamiento	WLSTATIC
Capa de Transporte	TCP/UDP
Capa de Aplicación	CBR
Tasa de transmisión	5,5 Mbps
Tasa Básica	5,5 Mbps
Umbral de Captura	10 dB
Umbral Sensado de Portador	2,38176e-15 W
Umbral de Recepción	2,38176e-15 W
Potencia de Transmisión	0,07943 W
Frecuencia de Transmisión	2,4 GHz
Pérdidas por Propagación	136,48 dB
Ganancia de la Antena de Transmisión	24 dBi
Ganancia de la Antena de Recepción	24 dBi
Distancia entre Nodos	66700 m
Tamaño del Paquete	1500 Bytes
Tiempo de Simulación	110 s
Área	10000 x 10000 m

La topología de la red inalámbrica se la presenta en la Figura 4.25, donde se emplea la herramienta NAM (*Network Animator*), que es el entorno gráfico para *ns-2*, y permite ver y crear simulaciones mediante un entorno sencillo y visual, este enlace se realiza desde el nodo maestro hacia el nodo esclavo que se encuentra en la parte inferior, las líneas entrecortadas que los unen simboliza la transmisión de paquetes, y las circunferencias que rodea el nodo, representa la confirmación de paquetes recibidos. En ambos enlaces se realiza el envío de paquetes mediante los protocolos TCP y UDP

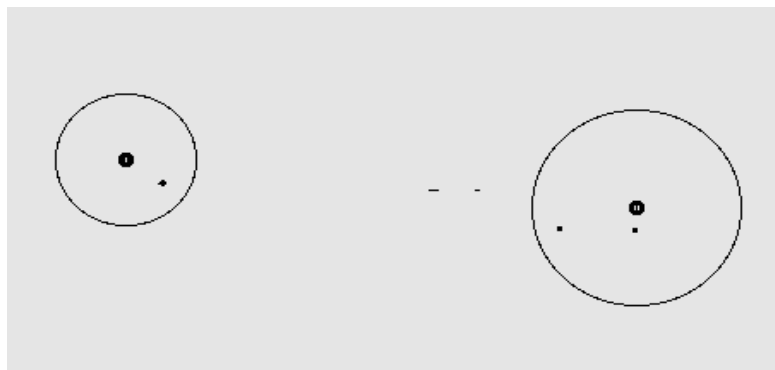


Figura 4.25 Simulación NAM 802.11b, 2- nodos (C. Salinas – C. Ánimas).

- **Simulación del enlace con el Protocolo TCP.**

La tabla 4.25 muestran los resultados que entrega el simulador bajo las condiciones establecidas para este enlace. El análisis de estos resultados se describe más adelante en el capítulo 5.

Tabla 4.25 Datos TCP Salinas - Animas.

PARÁMETRO	VALOR
Tiempo de Simulación	142,1981886 s
Número de Nodos	2
Número de Nodos Receptores	1
Número de Nodos Transmisores	1
Número de Paquetes Generados	46316
Número de Paquetes Enviados	33304
Número de Paquetes Caídos	4517
Número de Paquetes Perdidos	0
Tamaño Máximo del Paquete	1572 Bytes
Tamaño Mínimo del Paquete	28 Bytes
Promedio de Tamaño del Paquete	1061,9393 Bytes
Número de Bytes Enviados	50620640
Número de Bytes Caídos	11957392
Número de Bytes Perdidos	7826263
Retardo Mínimo	0,09012 s
Retardo Máximo	2,946578163 s
Retardo Promedio	2,62113345 s

a) *Throughput* de la Red

El *Throughput* de la red se obtiene aplicando la Ecuación 4.99 [35]:

$$\textit{Throughput de la Red} = \frac{B_{TX}}{t_{Tx}} \left[\frac{\textit{Bytes}}{\textit{s}} \right] \quad (4.99)$$

Dónde:

B_{TX} = Número de Bytes enviados

t_{Tx} = Tiempo total de simulación

Reemplazando los valores obtenidos en la tabla 4.25, se tiene:

$$\textit{Throughput de la Red} = \frac{50620640}{142,1} \quad (4.100)$$

$$\textit{Throughput de la Red} = 356232,51 \left[\frac{\textit{Bytes}}{\textit{s}} \right] \quad (4.101)$$

b) *Throughput* de la Red en Función al Número de Nodos

El *Throughput* de la red en función al número de nodos se obtiene empleando la Ecuación 102 [35]:

$$\eta = \frac{8 \times N_e}{T} \left[\frac{\text{bits}}{\text{s}} \right] \quad (4.102)$$

Dónde:

N_e = Número de Bytes enviados

T = Tiempo total de simulación

Reemplazando en la ecuación 4.102 los valores obtenidos en la tabla 4.25 se tiene:

$$\eta = \frac{8 (50620640)}{24,944} \quad (4.103)$$

$$\eta = 2849860,099 \left[\frac{\text{bits}}{\text{s}} \right] \quad (4.104)$$

$$\eta = 2,84 \text{ [Mbps]} \quad (4.105)$$

c) *Throughput* Real de la Red en Función al Número de Nodos

Para determinar el *Throughput* real de la red en función al número de nodos se utiliza la Ecuación 4.106 [35]:

$$\eta' = \frac{8 (N_e) (1-r)}{t} \left[\frac{\text{bits}}{\text{s}} \right] \quad (4.106)$$

Dónde:

$r = \frac{P_p}{P_e}$ = Relación entre Paquetes Perdidos y Paquetes Enviados

N_e = Número de Bytes enviados

t = Tiempo Total de Simulación

Reemplazando los valores de la tabla 4.25 en la ecuación 4.106 se tiene:

$$r = \frac{0}{33304} \quad (4.107)$$

$$r = 0 \quad (4.108)$$

$$\eta' = \frac{8 (50620640) (1-0)}{142,1} \left[\frac{\text{bits}}{\text{s}} \right] \quad (4.109)$$

$$\eta' = 2849860,099 \left[\frac{\text{bits}}{\text{s}} \right] \quad (4.110)$$

$$\eta' = 2,84 \text{ [Mbps]} \quad (4.111)$$

d) *Throughput* Normalizado de la Red en Función al Número de Nodos

El *Throughput* Normalizado de la red en función al número de nodos se define en base a la Ecuación 4.112 [35].

$$\eta_{\eta} = \frac{\Delta Th}{RBR} \quad (4.112)$$

Dónde:

$\Delta Th =$ Variación del *throughput* real de la red

$RBR =$ Round Bitrate (Tasa de transmisión)

Reemplazando los valores de la tabla 4.25 en la ecuación 4.112 se tiene:

$$\eta_{\eta} = \frac{2.84 \times 10^6}{5.5 \times 10^6} \quad (4.113)$$

$$\eta_{\eta} = 0,5163 \quad (4.114)$$

El *Throughput* normalizado de la red con el protocolo TCP alcanza un valor de 2.84 [Mbps], que en relación a la velocidad de transmisión teórica de 5.5 [Mbps], representa el 51,63 % de la capacidad máxima de la red. Existen también periodos de tiempo en los cuales el rendimiento de la red decrece notablemente.

A continuación, se muestran en las Figuras 4.26, 4.27 y 4.28 los resultados y análisis del Throughput obtenidas de la herramienta Trace Graph:

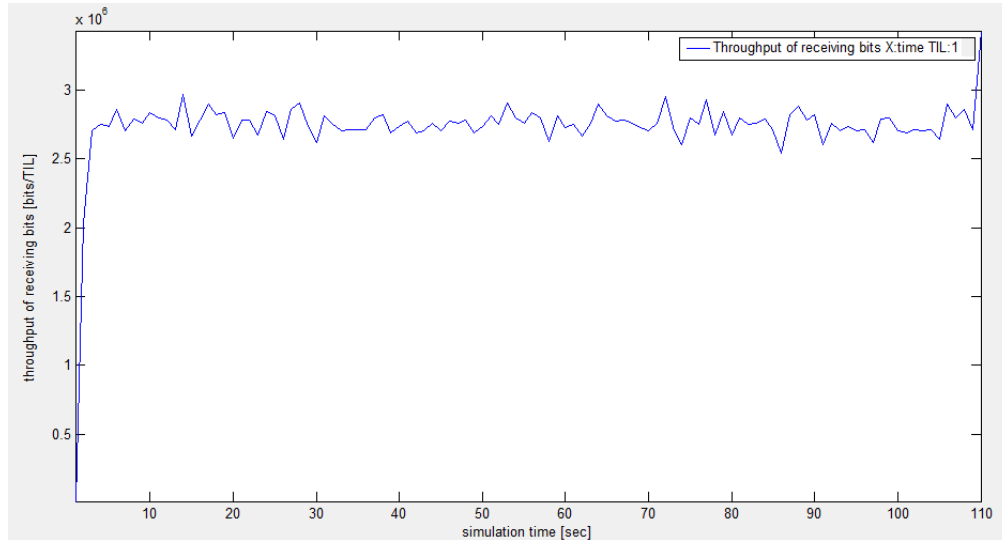


Figura 4.26 Throughput de bits recibidos TCP en Salinas - Animas.

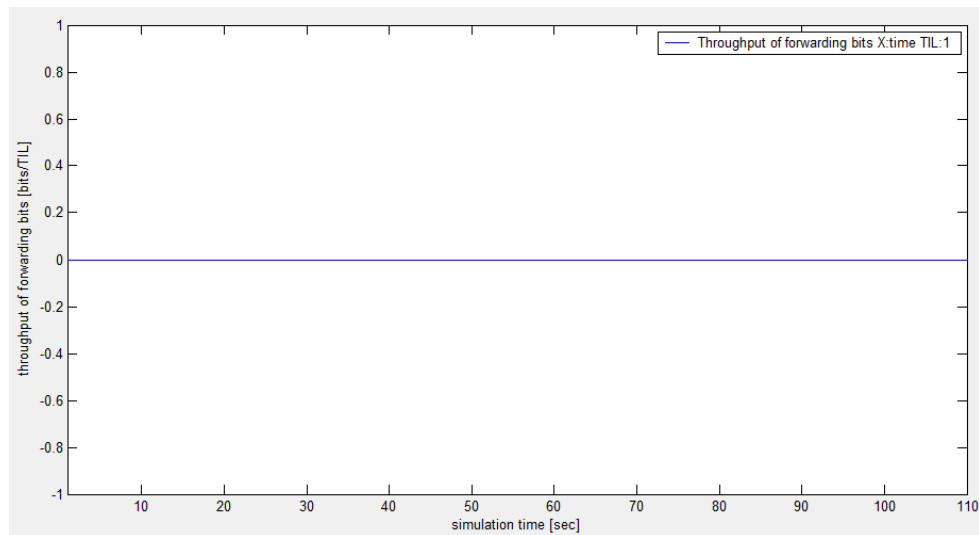


Figura 4.27 Throughput de bits reenviados TCP en Salinas - Animas.

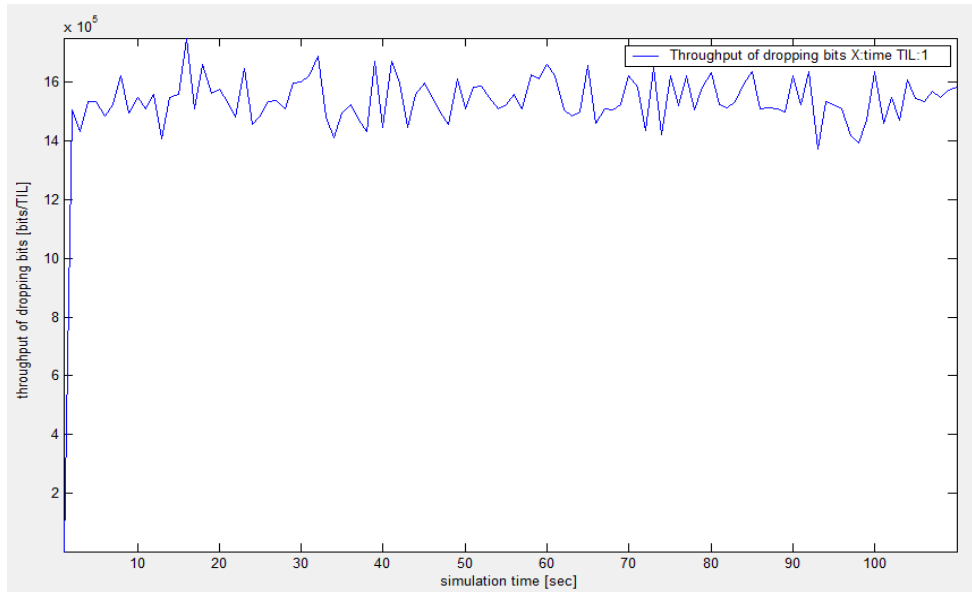


Figura 4.28 Throughput de bits caídos TCP en Salinas - Animas.

e) *Delay End-To-End*

El retardo *Delay End-To-End* es el tiempo medio que le toma a un paquete de datos para llegar a su destino. También incluye el retraso causado por el proceso de ruta descubrimiento y la cola en la transmisión de paquetes de datos.

Para el análisis del parámetro *Delay End-To-End* se toman directamente los valores resultantes del *Network Information de Trace Graph*, por lo que no es necesaria la realización de cálculos adicionales.

Además, en las Figuras 4.29 y 4.30 se muestran los valores de retardo del *Delay End-To-End* y el histograma de la distribución de frecuencia del *Delay End-To-End* respectivamente.

Simulation End2End delays in	
Minimal delay	0.09012 (0,1,12)
Maximal delay	2.946578163 (0,1,7881)
Average	2.62113345

Figura 4.29. Valores del Delay End-To-End TCP Salinas – Animas.

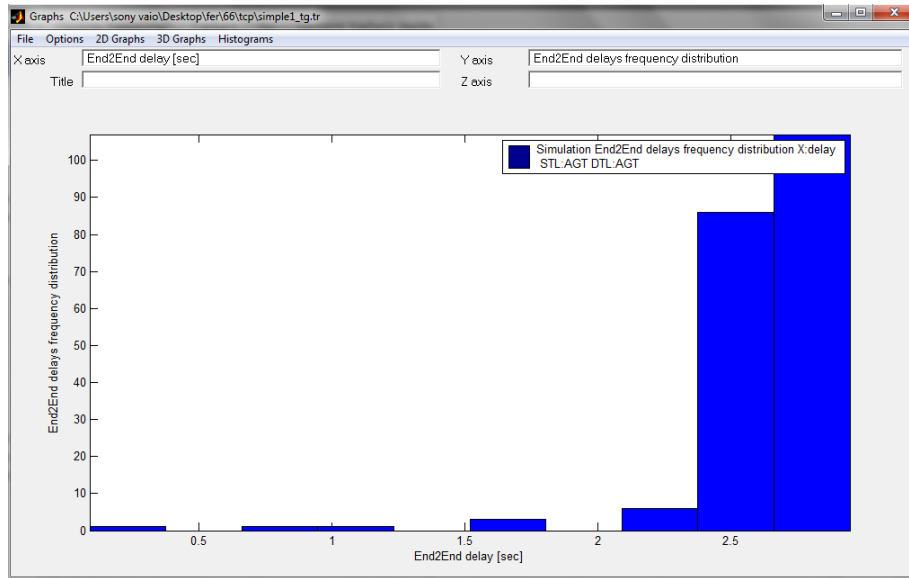


Figura 4.30. Histograma de frecuencia del Delay End-To-End TCP Salinas - Animas.

f) *Delivery Ratio* de la Red

El análisis del *Delivery Ratio* o relación de entrega permite determinar el nivel de congestamiento de la red, por lo que este parámetro tiene gran importancia en la definición del desempeño de los paquetes. Para su cálculo se aplica la Ecuación 4.115 [35].

$$Delivery\ Ratio(\%) = \frac{Número\ de\ Paquetes\ Recibidos}{Número\ de\ Paquetes\ Enviados} 100\% \quad (4.115)$$

Dónde:

de Paquetes Recibidos = *# de Paquetes Enviados* – *# de Paquetes Perdidos*

Al reemplazar en la ecuación 4.15 los datos de la tabla 4.25 se obtiene:

$$Delivery\ Ratio(\%) = \frac{33304}{33304} 100\% \quad (4.116)$$

$$Delivery\ Ratio(\%) = 100\% \quad (4.117)$$

- **Simulación del enlace con el Protocolo UDP**

La tabla 4.26 indican los resultados que entrega el simulador bajo las condiciones establecidas con tráfico UDP. El análisis de estos resultados se describe más adelante en el capítulo 5.

Tabla 4.26 Datos UDP Salinas - Animas.

PARÁMETRO	VALOR
Tiempo de Simulación	146,3959667 s
Número de Nodos	2
Número de Nodos Receptores	1
Número de Nodos Transmisores	1
Número de Paquetes Generados	36618
Número de Paquetes Enviados	23878
Número de Paquetes Caídos	0
Número de Paquetes Perdidos	7444
Tamaño Máximo del Paquete	1572 Bytes
Tamaño Mínimo del Paquete	28 Bytes
Promedio de Tamaño del Paquete	1281,7808 Bytes
Número de Bytes Enviados	8698604
Número de Bytes Caídos	0
Retardo Mínimo	0,460967576 s
Retardo Máximo	2,969754707 s
Retardo Promedio	2,611104796 s

a) *Throughput* de la Red

Reemplazando en la ecuación 4.99 los valores obtenidos en la tabla 4.26, se tiene:

$$Throughput\ de\ la\ Red = \frac{36233840}{146,3} \quad (4.118)$$

$$Throughput\ de\ la\ Red = 247668.079 \left[\frac{Bytes}{s} \right] \quad (4.119)$$

b) *Throughput* de la Red en Función al Número de Nodos

Reemplazando en la ecuación 4.102 los valores obtenidos en la tabla 4.26 se tiene:

$$\eta = \frac{8 \cdot 36233840}{146.3} \quad (4.120)$$

$$\eta = 1981344.63 \left[\frac{\text{bits}}{\text{s}} \right] \quad (4.121)$$

$$\eta = 1.98 \text{ [Mbps]} \quad (4.122)$$

c) *Throughput* Real de la Red en Función al Número de Nodos

Reemplazando en la ecuación 4.106 los valores obtenidos en la tabla 4.26 se tiene:

$$r = \frac{7444}{23878} \quad (4.123)$$

$$r = 0.311 \quad (4.124)$$

$$\eta' = \frac{8 \cdot 36233840 \cdot \left(1 - \frac{7444}{23878}\right)}{146.3} \left[\frac{\text{bits}}{\text{s}} \right] \quad (4.125)$$

$$\eta' = 1363657.66 \left[\frac{\text{bits}}{\text{s}} \right] \quad (4.126)$$

$$\eta' = 1.36 \text{ [Mbps]} \quad (4.127)$$

d) *Throughput* Normalizado de la Red en Función al Número de Nodos

Reemplazando en la ecuación 4.112 los valores obtenidos en la tabla 4.26 se tiene:

$$\eta_{\eta} = \frac{1.36 \cdot 10^6}{5.5 \cdot 10^6} \quad (4.128)$$

$$\eta_{\eta} = 0.2472 \quad (4.129)$$

El *Throughput* de la red en función al número de nodos con el protocolo UDP alcanza un valor de 1,36 [Mbps] que en relación a la velocidad de transmisión teórica de 5.5 [Mbps], representa el 24,72% de la capacidad máxima de la red. Y al

compararla con la velocidad alcanzada con el protocolo TCP se muestra una caída del 26,91%.

En las Figuras 4.31, 4.32 y 4.33 se muestra los resultados y análisis del Throughput obtenidas de la herramienta Trace Graph:

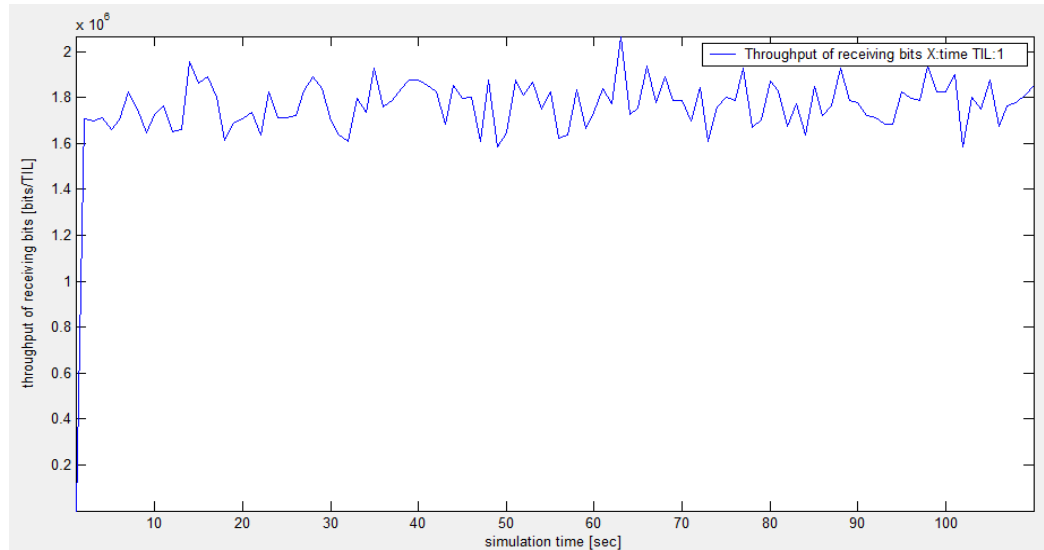


Figura 4.31 Throughput de bits recibidos UDP en Salinas - Animas.

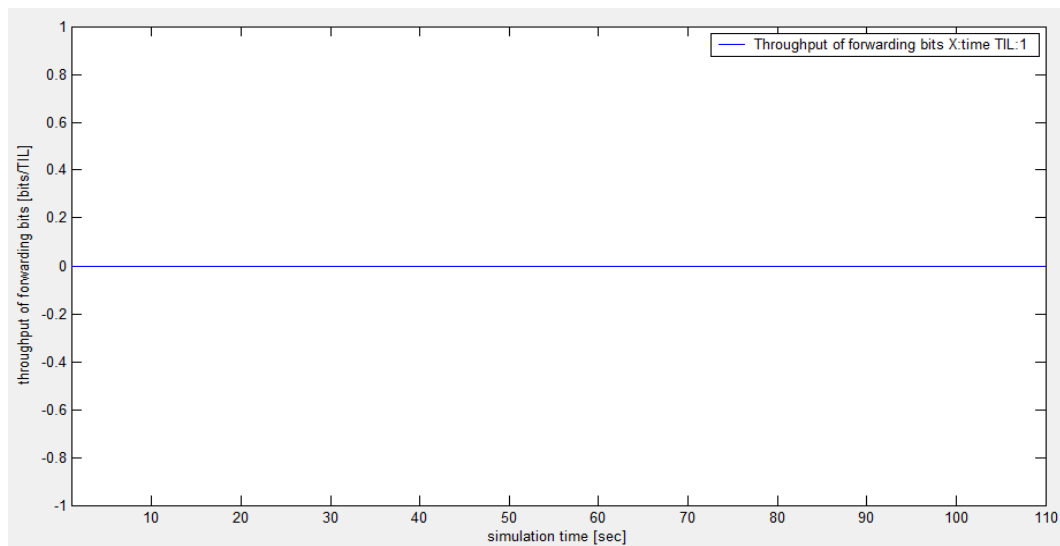


Figura 4.32 Throughput de bits retransmitidos UDP en Salinas – Animas.

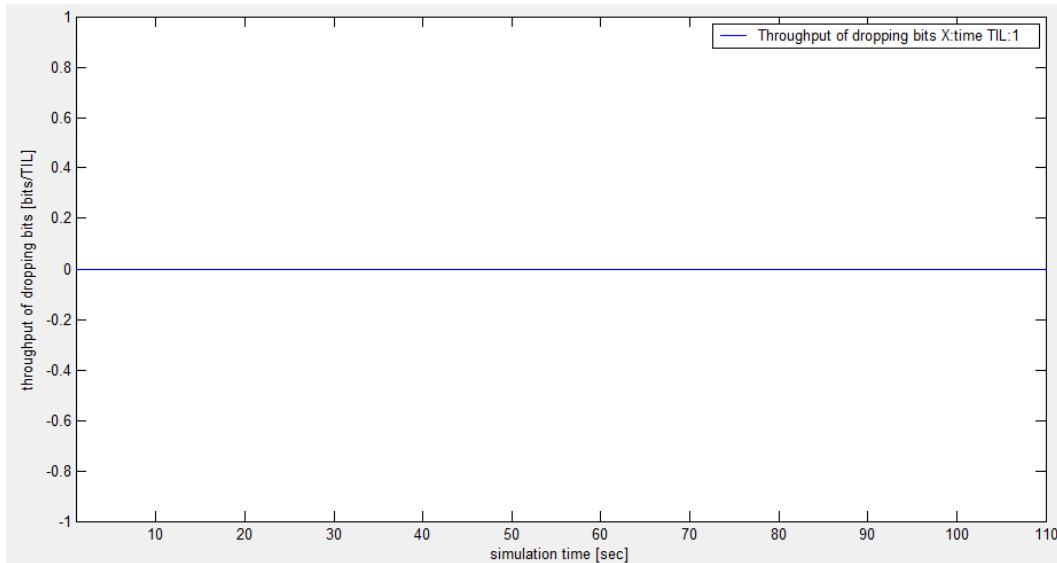


Figura 4.33 Throughput de bits caídos UDP en Salinas - Animas.

e) Análisis del *Delay End-To-End*

En las Figuras 4.34 y 4.35 se muestran los valores de retardo del *Delay End-To-End* y el histograma de la distribución de frecuencia del *Delay End-To-End* correspondientemente.

Simulation End2End delays in	
Minimal delay	0.460967576 (0,1,18)
Maximal delay	2.969754707 (0,1,1882)
Average	2.611104796

1Figura 4.34 Valores del Delay End-To-End UDP Salinas - Animas.

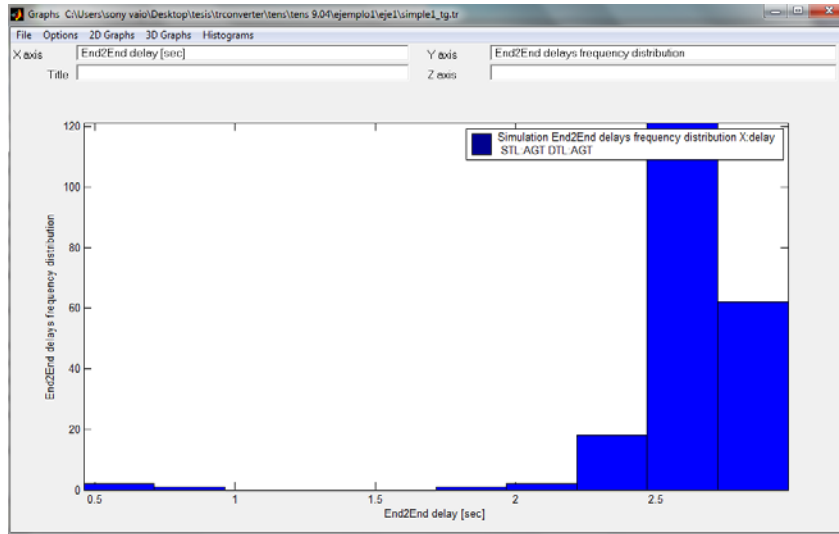


Figura 4.35 Valores del Delay End-To-End UDP Salinas - Animas.

Figura 5.11 Histograma de frecuencia del Delay End-To-End UDP Salinas – Animas

f) Análisis de *Delivery Ratio* de la Red

Al reemplazar en la ecuación 5.5 los valores obtenidos en la tabla 4.25 se tiene:

$$Delivery Ratio(\%) = \frac{16434}{23878} 100\% \quad (4.130)$$

$$Delivery Ratio(\%) = 68,8\% \quad (4.131)$$

4.1.19 Simulación del Enlace Salinas –Cabuyas

La topología de la red inalámbrica con la herramienta NAM (*Network Animator*) se la presenta en la Figura 4.36. Este enlace se realiza desde el nodo maestro, el de Salinas en la parte inferior hacia el nodo esclavo Cabuyas en la parte superior, las líneas entrecortadas que los unen simboliza la transmisión de paquetes, y las circunferencias que rodea el nodo, representa la confirmación de paquetes recibidos.

La tabla 4.27 detalla los parámetros necesarios para incluir en el script del TeNS, el código completo del programa para este enlace se detalla en el anexo 8.

Tabla 4.27 Parámetros Seteados en ns-2 Mejorado para Salinas - Cabuyas.

PARÁMETRO	VALOR
Modelo de Propagación	TwoRayGround
Tipo de Canal	Wireless Channel
Network Interface	Wireless 802.11
MAC	MAC 802.11
Tipo de Antena	Direccional
Número de Nodos	2
Máximo Número de Paquetes	50
Protocolo de Enrutamiento	WLSTATIC
Capa de Transporte	TCP/UDP
Capa de Aplicación	CBR
Tasa de transmisión	5,5 Mbps
Tasa Básica	5,5 Mbps
Umbral de Captura	10 dB
Umbral Sensado de Portador	2,07454e-15 W
Umbral de Recepción	2,07454e-15 W
Potencia de Transmisión	0,07943 W
Frecuencia de Transmisión	2,4 GHz
Pérdidas por Propagación	137,55 dB
Ganancia de la Antena de Transmisión	24 dBi
Ganancia de la Antena de Recepción	24 dBi
Distancia entre Nodos	77500 m
Tamaño del Paquete	1500 Bytes
Tiempo de Simulación	110 s
Área	10000 x 10000 m

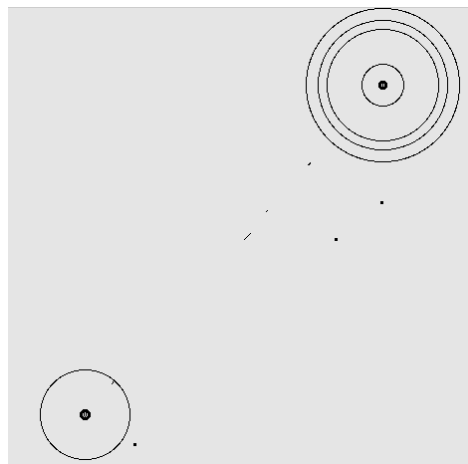


Figura 4.36 Simulación NAM 802.11b, 2- nodos (C. Salinas – C. Cabuyas).

- **Simulación del enlace con el Protocolo TCP**

La tabla 4.28 muestran los resultados que entrega el simulador bajo las condiciones establecidas para este enlace.

Tabla 4.28 Datos TCP Salinas - Cabuyas

PARÁMETRO	VALOR
Tiempo de Simulación	186,3 s
Número de Nodos	2
Número de Nodos Receptores	1
Número de Nodos Transmisores	1
Número de Paquetes Generados	32814
Número de Paquetes Enviados	23775
Número de Paquetes Caídos	4436
Número de Paquetes Perdidos	0
Tamaño Máximo del Paquete	1572 Bytes
Tamaño Mínimo del Paquete	28 Bytes
Promedio de Tamaño del Paquete	12,860.125 Bytes
Número de Bytes Enviados	36100992
Número de Bytes Retransmitidos	11693108
Retardo Mínimo	0,106773431 s
Retardo Máximo	3,321.535.434 s
Retardo Promedio	2,994.291.375 s

a) Análisis del *Throughput* de la Red

Reemplazando en la ecuación 4.99 los valores de la Tabla 4.28, se tiene:

$$Throughput\ de\ la\ Red = \frac{36100992}{186.3} \quad (4.132)$$

$$Throughput\ de\ la\ Red = 193778.808 \left[\frac{Bytes}{s} \right] \quad (4.133)$$

b) *Throughput* de la Red en Función al Número de Nodos

Reemplazando en la ecuación 4.102 los valores obtenidos en la tabla 4.28 se tiene:

$$\eta = \frac{8 \cdot 36100992}{186.3} \quad (4.134)$$

$$\eta = 1550230.46 \left[\frac{\text{bits}}{\text{s}} \right] \quad (4.135)$$

$$\eta = 1.55 [\text{Mbps}] \quad (4.136)$$

c) *Throughput* Real de la Red en Función al Número de Nodos

Al reemplazar en la ecuación 4.106 los valores obtenidos en la tabla 4.28 se tiene:

$$r = \frac{0}{23775} \quad (4.137)$$

$$r = 0 \quad (4.138)$$

$$\eta' = \frac{8 \cdot 36100992 \cdot (1-0)}{186.3} \left[\frac{\text{bits}}{\text{s}} \right] \quad (4.139)$$

$$\eta' = 1550230.46 \left[\frac{\text{bits}}{\text{s}} \right] \quad (4.140)$$

$$\eta' = 1.55 [\text{Mbps}] \quad (4.141)$$

d) *Throughput* Normalizado de la Red en Función al Número de Nodos

Al reemplazar en la ecuación 4.112 los valores obtenidos en la tabla 4.28 se tiene:

$$\eta_{\eta} = \frac{1.55 \cdot 10^6}{5.5 \cdot 10^6} \quad (4.142)$$

$$\eta_{\eta} = 0.2818 \quad (4.143)$$

El *Throughput* de la red en función al número de nodos con el protocolo TCP alcanza un valor de 1.55 [Mbps], que en relación a la velocidad de transmisión teórica de 5.5 [Mbps], representa el 28,18 % de la capacidad máxima de la red.

Las Figuras 4,37, 4.38 y 4.39 muestran los resultados y análisis del Throughput obtenidas de la herramienta Trace Graph:

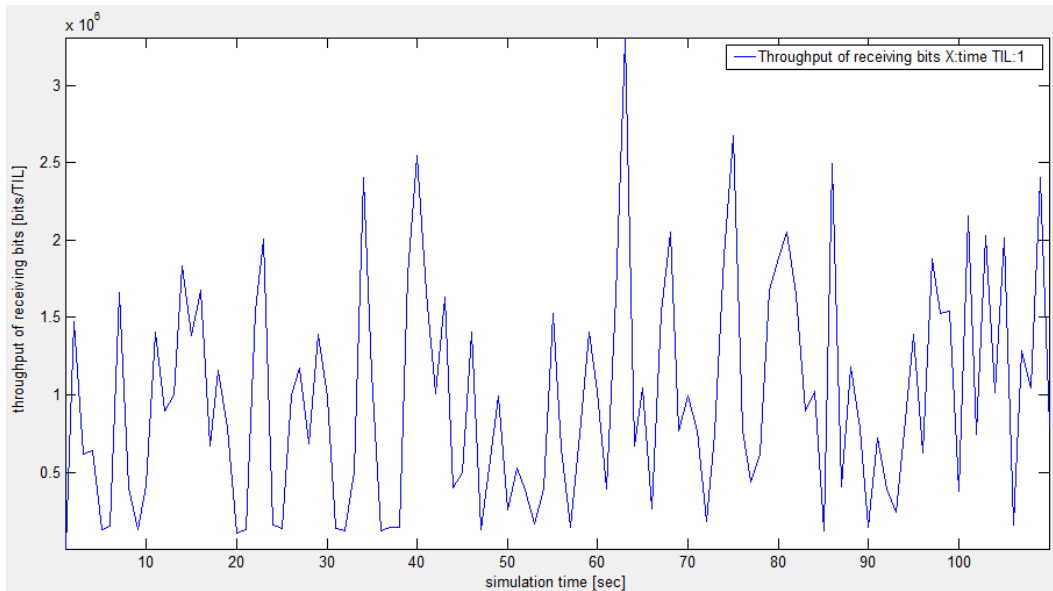


Figura 4.37 Throughput de bits recibidos TCP en Salinas - Cabuyas.

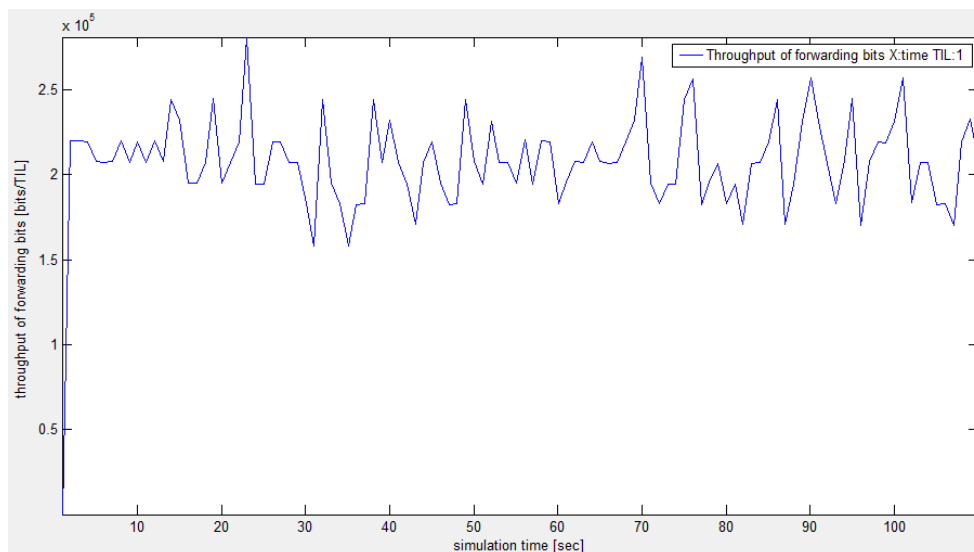
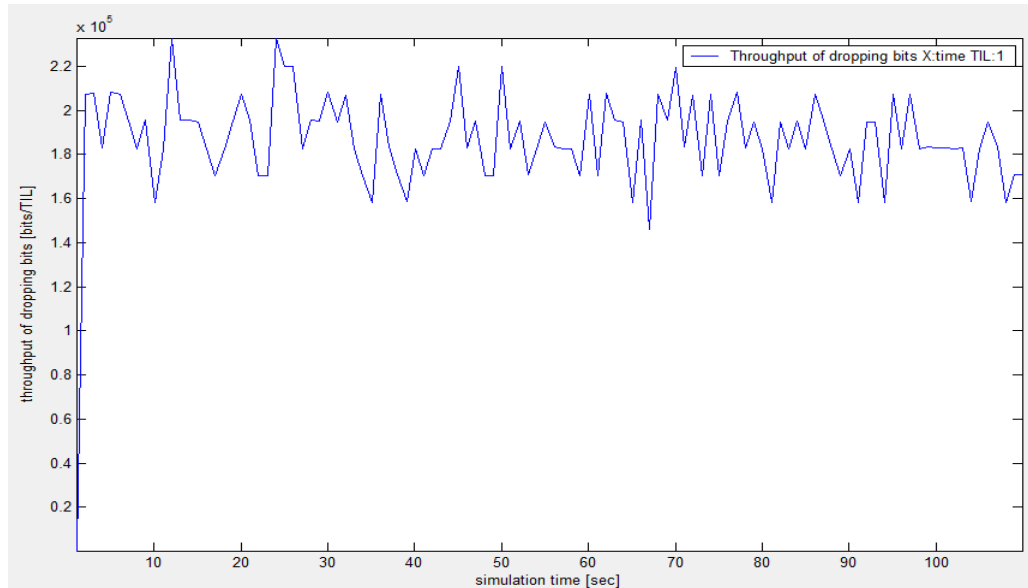


Figura 4.38 Throughput de bits reenviados TCP en Salinas - Cabuyas.



2Figura 4.39 Throughput de bits caídos TCP en Salinas - Cabuyas.

e) Análisis del *Delay End-To-End*

Al igual que en caso del enlace anterior en las Figuras 4.40 y 4.41 se muestran los valores de retardo del *Delay End-To-End* y el histograma de la distribución de su frecuencia respectivamente.

Simulation End2End delays in	
Minimal delay	0.106773431 (0,1,12)
Maximal delay	3.321535434 (0,1,18392)
Average	2.994291375

Figura 4.40. Valores del Delay End-To-End TCP Salinas – Cabuyas.

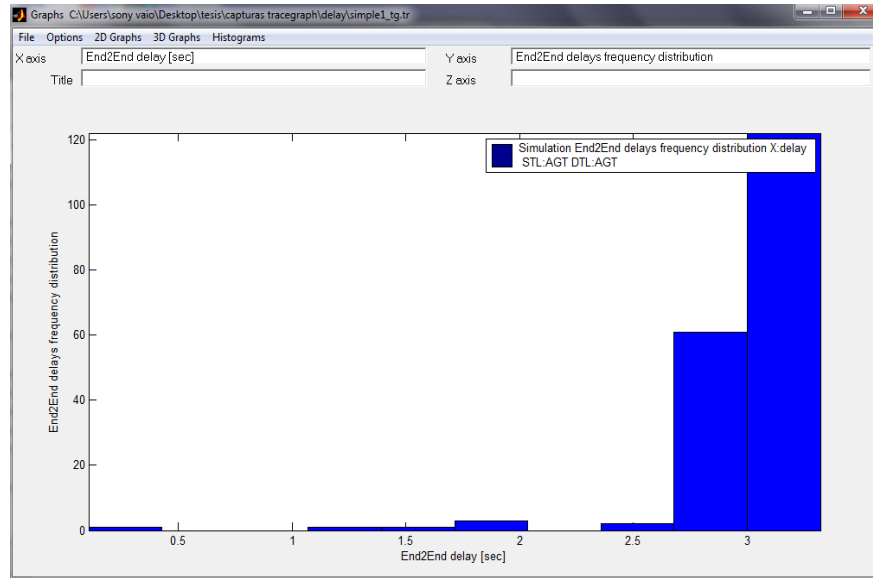


Figura 4.41 Histograma de frecuencia del Delay End-To-End TCP Salinas - Cabuyas.

f) Análisis de *Delivery Ratio* de la Red

Al reemplazar en la ecuación 4.115 los valores obtenidos en la tabla 4.28 se tiene:

$$Delivery Ratio(\%) = \frac{23775}{23775} 100\% \quad (4.144)$$

$$Delivery Ratio(\%) = 100\% \quad (4.145)$$

- **Simulación del enlace con el Protocolo UDP**

La tabla 4.29 indica los resultados que entrega el simulador bajo las condiciones establecidas con tráfico UDP. El análisis de estos resultados se describe más adelante en el capítulo 5.

Tabla 4.29 Datos UDP Salinas - Cabuyas

PARÁMETRO	VALOR
Tiempo de Simulación	98,28 s
Número de Nodos	2
Número de Nodos Receptores	1
Número de Nodos Transmisores	1
Número de Paquetes Generados	4136
Número de Paquetes Enviados	3332
Número de Paquetes Caídos	0
Número de Paquetes Perdidos	1317
Tamaño Máximo del Paquete	1572 Bytes
Tamaño Mínimo del Paquete	28 Bytes
Promedio de Tamaño del Paquete	1061,9821 Bytes
Número de Bytes Enviados	5065104
Número de Bytes Caídos	0
Retardo Mínimo	0,486382796 s
Retardo Máximo	3,413608829 s
Retardo Promedio	3,003035818 s

a) Análisis del *Throughput* de la Red

Al reemplazar en la ecuación 4.99 los valores obtenidos en la tabla 4.29 se tiene:

$$\textit{Throughput de la Red} = \frac{5065104}{98.2} \quad (4.146)$$

$$\textit{Throughput de la Red} = 51579.47 \left[\frac{\textit{Bytes}}{\textit{s}} \right] \quad (4.147)$$

b) *Throughput* de la Red en Función al Número de Nodos

Al reemplazar en la ecuación 4.102 los valores obtenidos en la tabla 4.29 se tiene:

$$\eta = \frac{8 (5065104)}{98,2} \quad (4.148)$$

$$\eta = 412635,76 \left[\frac{\textit{bits}}{\textit{s}} \right] \quad (4.149)$$

$$\eta = 412,63 [Kbps] \quad (4.150)$$

c) *Throughput* Real de la Red en Función al Número de Nodos

Al reemplazar en la ecuación 4.106 los valores de la tabla 4.29 se tiene:

$$r = \frac{1317}{3332} \quad (4.151)$$

$$r = 0.3952 \quad (4.152)$$

$$\eta' = \frac{8 (5065104) (1 - \frac{1317}{3332})}{98.2} \left[\frac{bits}{s} \right] \quad (4.153)$$

$$\eta' = 249538,134 \left[\frac{bits}{s} \right] \quad (4.154)$$

$$\eta' = 249,53 [kbps] \quad (4.155)$$

d) *Throughput* Normalizado de la Red en Función al Número de Nodos.

Al reemplazar en la ecuación 4.112 los valores de la tabla 4.29 se tiene:

$$\eta_{\eta} = \frac{249,53 \times 10^3}{5,5 \times 10^6} \quad (4.156)$$

$$\eta_{\eta} = 0,0453 \quad (4.157)$$

El *Throughput* de la red en función al número de nodos con el protocolo UDP alcanza un valor de 243,53 [Kbps], que en relación a la velocidad de transmisión teórica de 5.5 [Mbps], representa el 4,53 % de la capacidad máxima de la red.

Y al compararla con la velocidad alcanzada con el protocolo TCP se muestra una caída del 23,65 %.

En las Figuras 5.19, 5.20 y 5.21 los resultados y análisis del *Throughput* obtenidas de la herramienta *Trace Graph*:

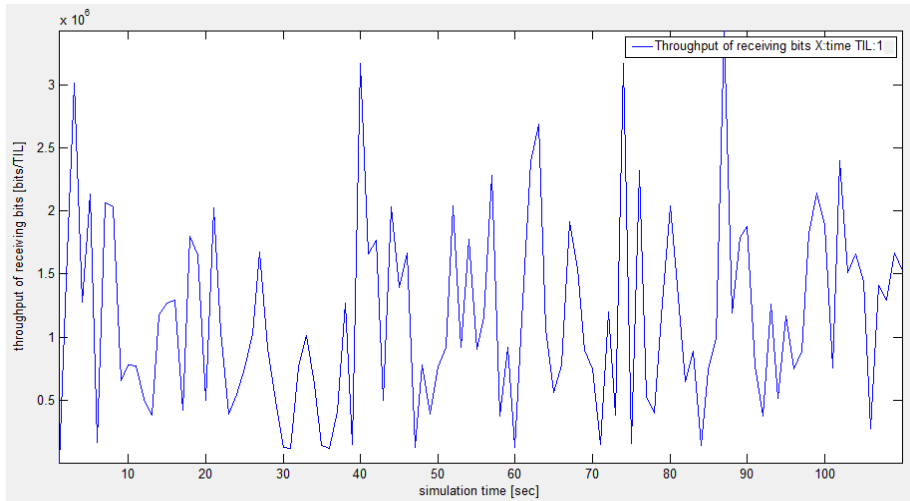


Figura 4.42 Throughput de bits recibidos UDP en Salinas - Cabuyas.

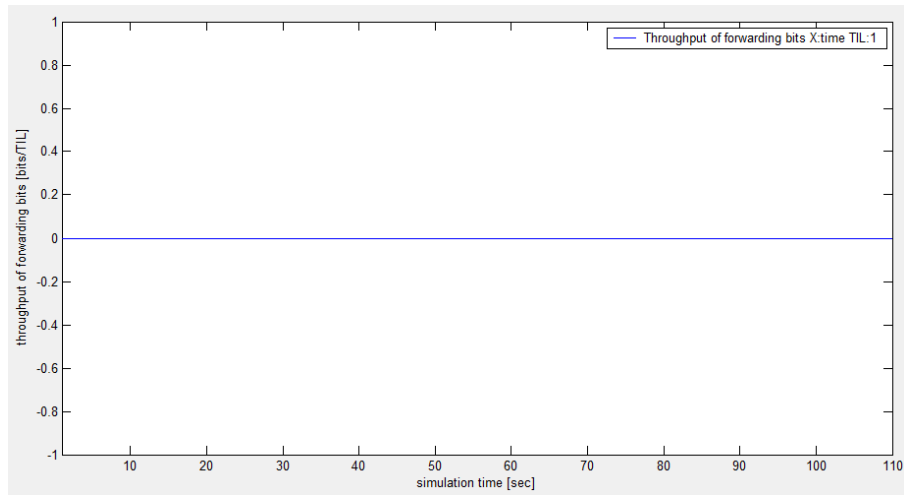


Figura 4.43 Throughput de bits reenviados UDP en Salinas - Cabuyas.

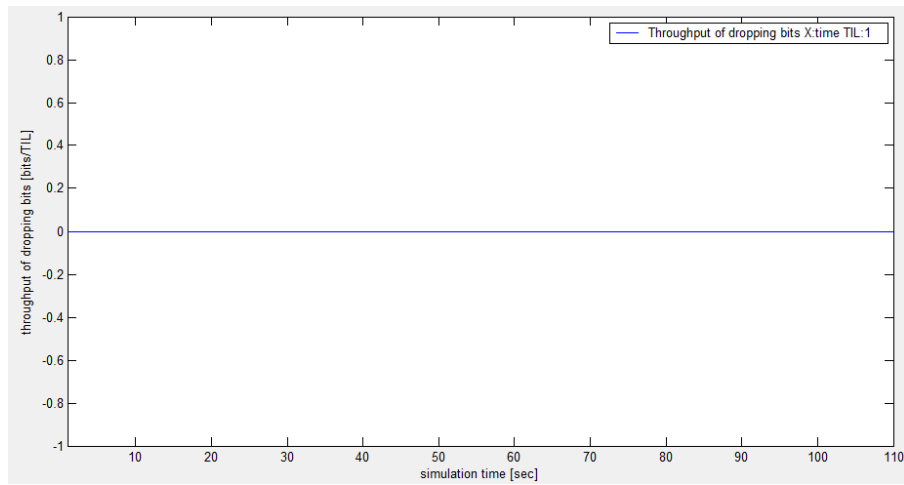


Figura 4.44 Throughput de bits caídos UDP en Salinas - Cabuyas.

e) Análisis del *Delay End-To-End*

La figura 4.45 muestra los valores de retardo del *Delay End-To-End* y en figura 4.46 se encuentra el histograma de la distribución de frecuencia del *Delay End-To-End* correspondientemente.

Simulation End2End delays in	
Minimal delay	0.486382796 (0,1,17)
Maximal delay	3.413608829 (0,1,18788)
Average	3.003035818

Figura 4.45 Valores del Delay End-To-End UDP Salinas - Cabuyas.

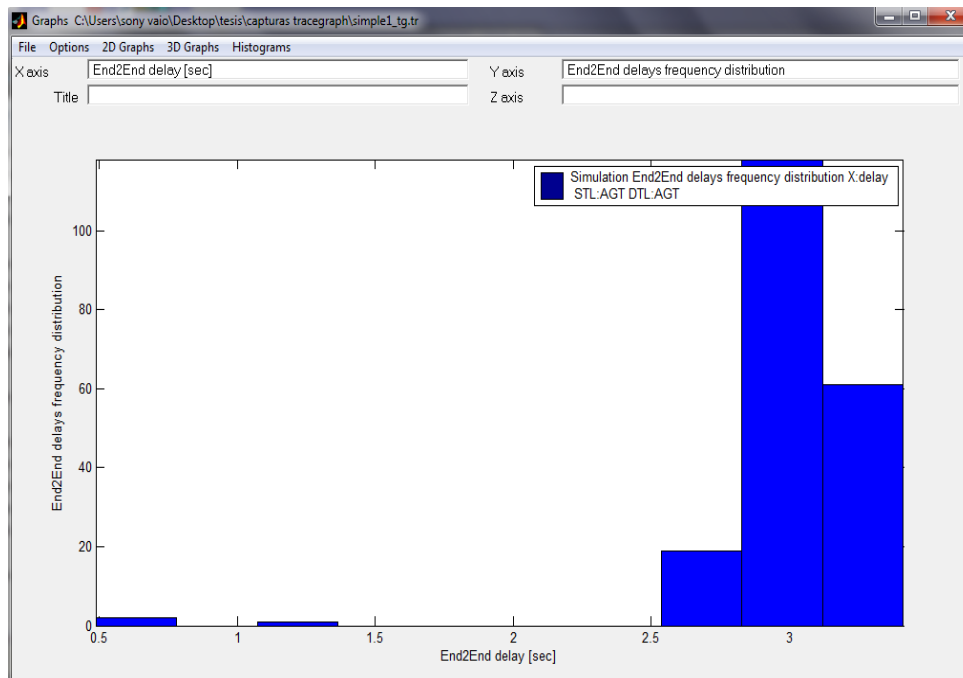


Figura 4.46 Histograma de frecuencia del Delay End-To-End UDP Salinas - Cabuyas.

f) Análisis de *Delivery Ratio* de la Red

Al reemplazar en la ecuación 4.115 los valores de la tabla 4.29 se tiene:

$$Delivery\ Ratio(\%) = \frac{2015}{3332} 100\% \quad (4.158)$$

$$Delivery\ Ratio(\%) = 60,4\% \quad (4.159)$$

4.2 IMPLEMENTACIÓN DE LA RED

Como se manifestó anteriormente para el enlace en la región costa, la mayor ventaja que se tiene es la línea de vista. Ambos enlaces tienen un punto en común que es el cerro salinas, que se muestra en la figura 4.47 y en la parte derecha se indica la torre que se utilizó para los dos enlaces.

El cerro se encuentra ubicado dentro de la base naval en salinas, y es uno de los puntos más extremos del lugar, se encuentra junto a la chocolatera, que es el punto más sobresaliente del perfil costanero de América del Sur, sobre el océano pacífico.

Los 3 puntos cuentan con comunicación fija y alimentación adecuada, dentro de cada torre lo que facilita la implementación de estos.



Figura 4.47 Cerro Salinas y torre utilizada.

El acceso a este cerro esta en excelentes condiciones, la cumbre se encuentra a 5 minutos del nivel del mar, y la torre de la estación salinas que se utilizó es de 17 m. las antenas fueron ubicadas a 15 m.

La figura 4.48 muestra el interior de la caseta de la torre, a la izquierda se observa la alimentación proporcionada y a la derecha la comunicación existente entre los puntos.



Figura 4.48 Caseta, Alimentación y comunicación.

4.2.1 Equipos y materiales para los enlaces.

Los equipos que se requieren para los enlaces son:

- ✓ Radios para comunicación.
- ✓ Computadoras portátiles con todos los programas necesarios.
- ✓ Cables Ethernet directos para interiores.
- ✓ Cables Ethernet directos para exteriores.
- ✓ Antenas direccionales tipo grilla HG2424 (Especificación en el anexo).
- ✓ Placas Alix.
- ✓ *Power station* Ubiquiti.
- ✓ *Pigtails*.
- ✓ *GPS*
- ✓ Abrazaderas.

- ✓ Caja de herramientas.
- ✓ Cinta auto fundente.
- ✓ Equipo de seguridad para subir a las torres.

- **Power station 2-EXT Ubiquiti.**

Es un equipo de arquitectura 802.11 versátil, basada en la plataforma inalámbrica externa, mostrado en la figura 4.49, que puede ser usado como AP, puente, estación, cliente u otras aplicaciones. Incluye un diseño de antena y radio que permite llegar a más de 50 Km e introducir un gran incremento de Throughput trabajando sobre TCP/IP [36].



Figura 4.49 Power Station 2 EXT.

Cuenta con 4 *leds* indicadores de recepción y un software que permite su configuración y la visualización en el ordenador de la calidad de apuntamiento con la antena en el otro extremo.

Si bien es cierto que este equipo trabaja a distancias menores a las que se implementan en estos dos enlaces, se puede aprovechar las ventajas de éste. Al hacerlo trabajar conjuntamente con la antena grilla de mayor ganancia, se obtiene un

indicador gráfico de apuntamiento entre las antenas ya que con la placa Alix no se tiene ésta opción lo que dificulta el alineamiento.

4.2.1 Enlace Salinas – Animas

El cerro Ánimas se encuentra al sur oeste de la provincia del Guayas en el pueblo de Olmedo, y cuenta con un excelente acceso por carretera desde la entrada hasta la parte más alta, en cuya recorrido se emplea 15 minutos. La figura 4.50 muestra la cumbre de este cerro y la torre utilizada fue la que se muestra en la parte derecha de la figura, pertenece al comando conjunto, con una altura de 25 m.



Figura 4.50 Cerro Ánimas y torre utilizada.

Las placas Alix cuentan con una caja especial para exteriores que es necesaria para transportar las placas y evitar su deterioro, como se puede observar en la figura 4.51; se debe tener cuidado de que al cerrar la caja, esta quede totalmente hermética e incluso sellarla por fuera si así lo requiere la aplicación. También se debe cuidar que el *pigtail* quede bien sujeto al momento de guardar la tarjeta en la caja.



Figura 4.51 Caja para exteriores de la placa Alix.

La antena es este cerro fue colocada en la parte más alta de la torre a 25 m., solo por la facilidad existía para colocar la antena y porque era más conveniente el espacio de trabajo a esa altura. Como se ve en la figura 4.52.



Figura 4.52 Caja para exteriores de la placa Alix.

Para el ascenso a la torre es necesario el equipo de seguridad a pesar de que cuenta con una muy buena y muy segura estructura.

- **Alineación de las antenas**

Para este procedimiento resultó de mucha ayuda el hecho de que estos dos puntos (Salinas y Ánimas) ya están unidos por parte del ejército con diferentes tecnologías, lo que brinda una antena de referencia en cada punto, para colocar la antena grilla.

El primer paso para la alineación de las antenas es el uso del *GPS*, donde se miran datos de latitud, longitud y ángulo de elevación para dirigir lo mejor posible la antena, como se ve en la figura 4.53. Durante este juego de movimientos, una deberá moverse mientras la del otro extremo se mantiene estática, y luego se cambian los papeles.



Figura 4.53 Manejo del GPS sobre la antena.

Una vez ubicadas las antenas con las dos referencias anteriores, es momento de usar el equipo *Power Station 2EXT*. Se conecta a la antena y se busca gráficamente la respuesta de alineación como muestra la figura 4.54, mientras más alto sea el nivel, mejor será la recepción.



Figura 4.54 Power Station 2EXT, Cerro Animas.

4.2.2 Enlace Salinas – Cabuyas

En el cerro Cabuyas se encuentra al sur este de la provincia de Manabí en Jipijapa, este cuenta con apenas 2 torres, el acceso a la cima no es muy bueno, el recorrido demora 20 minutos, por un camino de tierra muy accidentado y donde no entra cualquier vehículo. La figura 4.55 muestra este cerro y la torre que se usó con una altura de 45 m, perteneciente también al comando conjunto.



Figura 4.55 Cerro Cabuyas.

- **Alineación de las antenas**

Se empleó el mismo método utilizado para el enlace anterior ya que estos puntos (Salinas y Cabuyas) también se encuentran unidos por el ejército, para comunicación y monitoreo de embarcaciones y aviones.

En este enlace el nivel de recepción obtenido fue poco, como se aprecia en la figura 4.56, esto se debe a que el rango de acción para el equipo fabricado por *Ubiquiti* estaba al 175 % del límite aceptable, es decir casi 80Km; otro factor es que el enlace se ubica sobre mar y este equipo no cuenta con especificaciones al respecto. Al final cumplió el objetivo que es mostrar la correcta alineación de las antenas.



Figura 4.56 Power Station 2EXT, cerro Cabuyas,

4.2.3 Inyección de tráfico.

La inyección de tráfico fue realizada con la ayuda del programa D-ITG (*Distributed Internet Traffic Generator*) que es una plataforma de código abierto para la generación de tráfico para paquetes, como se detalló en el capítulo 3.

Configuración de D-ITG

Para realizar las diversas pruebas es necesario configurar los equipos que intervendrán en la comunicación correctamente, cada vez que se haga una prueba se debe copiar el archivo *.log* resultante y sus análisis obtenidos mediante la herramienta *Analyzer*, en una carpeta distinta a la original ya que la información se sobre escribe.

- **Emisor**

Para la configuración en el equipo maestro primeramente se debe definir el flujo de datos como se puede observar en la Figura 4.57, con los parámetros resumidos en la Tabla 4.30. El tiempo por defecto es de 10 s, para un mejor tratamiento de datos este tiempo deberá ser de 30 s.

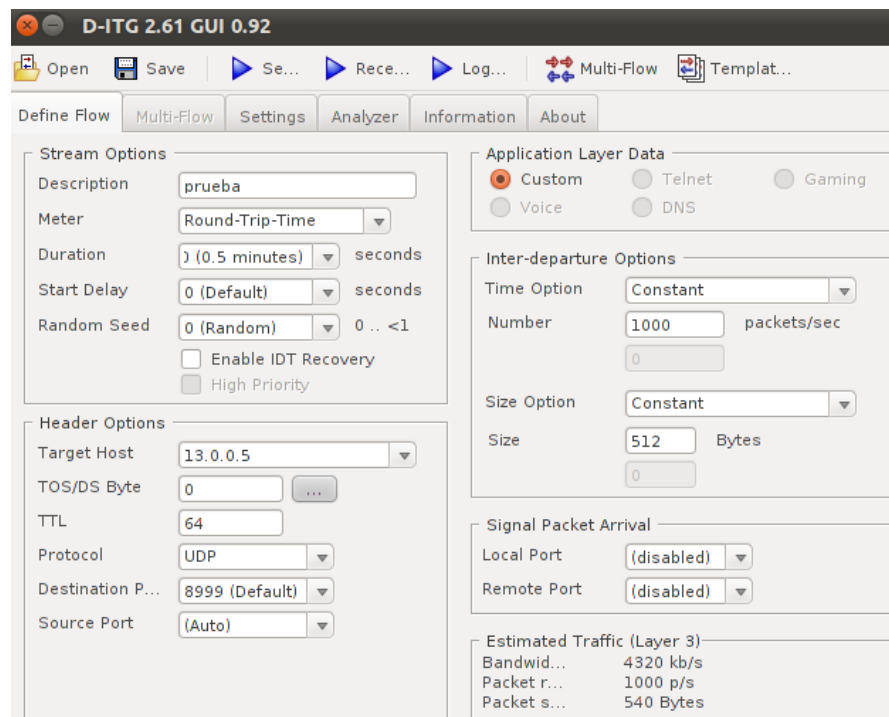


Figura 4.57 D-ITG, Definición de flujo en emisor.

Tabla 4.30. Parámetros para el flujo del emisor.

PARÁMETRO	VALOR
Dirección de Destino	13.0.0.5
TTL	64
Protocolo	UDP/TCP
Número de Paquete por Segundo	1000
Tamaño (bytes)	512

La configuración del emisor se indica en la Figura 4.58; en la misma se especifica la dirección binaria y del log en donde se encuentran los archivos, el fichero de registro a enviar se denomina *itgsend.log* y se renombra el archivo de salida que se alojará en el receptor. En la Tabla 4.31 se indican las opciones que deben ser activadas en receptor y emisor.

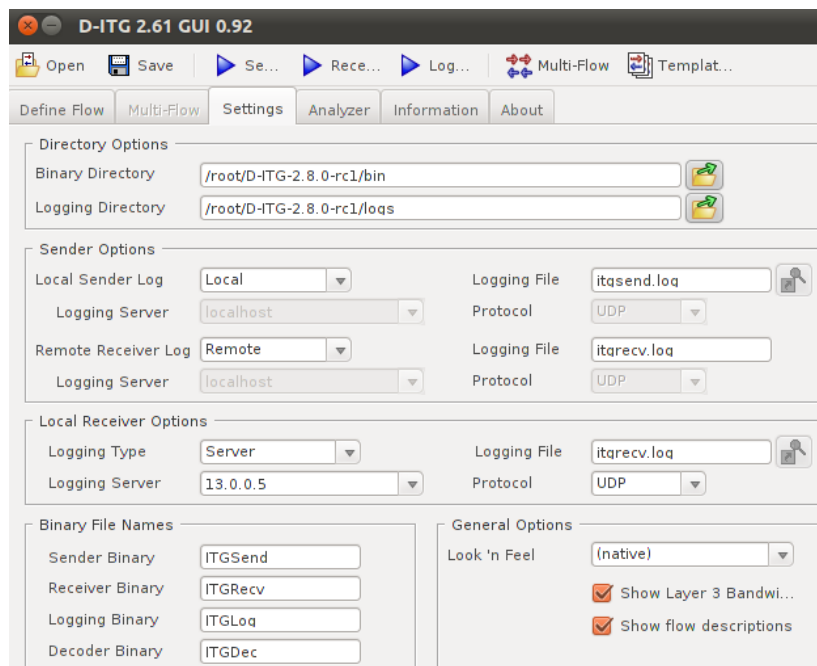
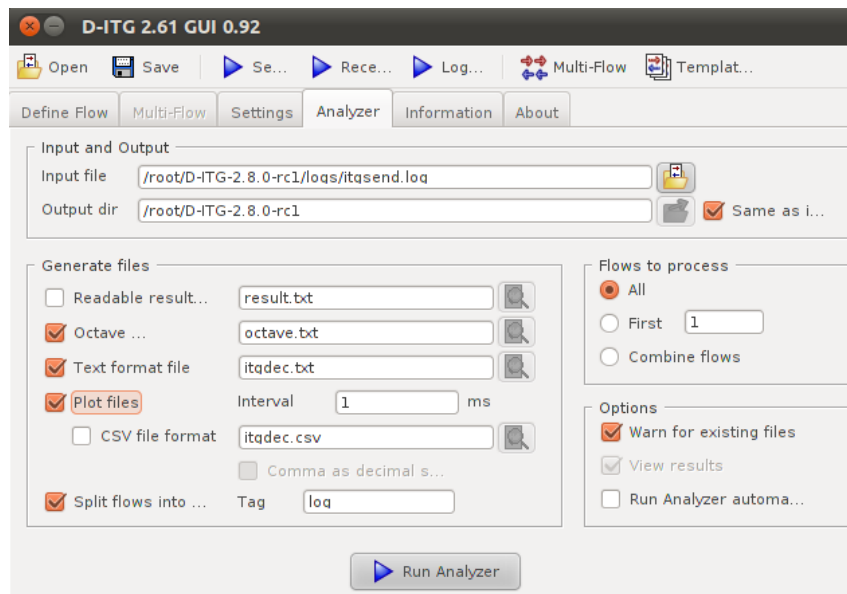
**Figura 4.58 D-ITG, Configuración de emisor.**

Tabla 4.31 Parámetros de configuración del emisor.

PARÁMETRO	TIPO
Registro del Emisor	Local
Registro del Receptor	Remoto
Receptor Local	Servidor
Receptor Local	Dirección de destino

En la pestaña *Analyzer*, como se muestra en la Figura 4.59, se determina las direcciones del archivo de entrada y de salida, además se señalan los ficheros que se desean generar, en este caso se obtuvieron archivos *.txt* que resumen los resultados de la simulación, y los archivos *.dat* que posteriormente servirán para obtener las gráficas de *delay*, *jitter*, *pACKet loss* y *throughput* con la herramienta *ITGplot*. Esta configuración se la realiza únicamente en el equipo receptor.

Figura 4.59 D-ITG, Configuración de *Analyzer* en emisor.

- **Receptor**

La configuración en el receptor se asemeja a la del inyector en el equipo emisor, con la única diferencia que el host objetivo es local, como se muestra en la Figura 4.60.

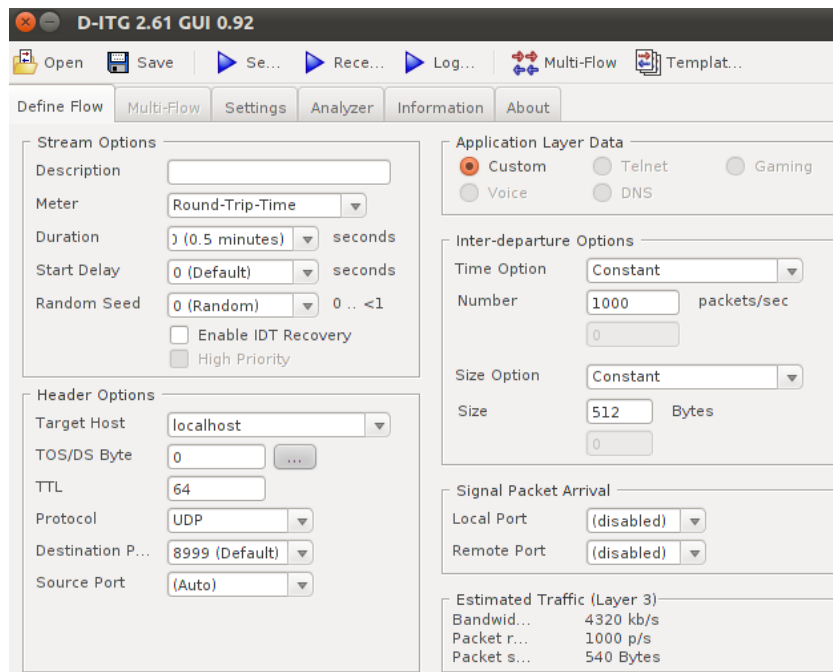


Figura 4.60. D-ITG, Configuración del equipo receptor.

En la pestaña de Settings se definen las direcciones del archivo log y el archivo binario, además se configuran las opciones del emisor y receptor como la Figura 4.61; las opciones que deben ser seleccionadas se indican en la Tabla 4.32.

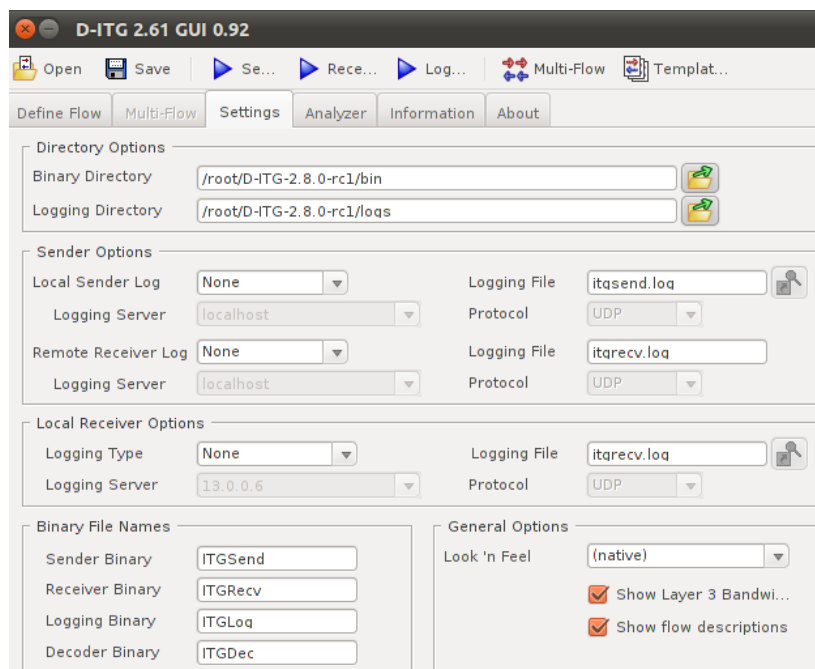


Figura 4.61 D-ITG, Configuración Settings en receptor.

Tabla 4.32. Parámetros definidos en equipo receptor.

PARÁMETRO	TIPO
Registro del Emisor	<i>None</i>
Registro del Receptor	<i>None</i>
Receptor Local	<i>None</i>
Receptor Local	<i>None</i>

Los resultados que arroja el programa D-ITG, se muestran a continuación, donde las figuras 4.62 y 4.63 muestran los resultados para el enlace Salinas – Animas, con protocolos TCP y UDP respectivamente. Y las figuras 4.64 y 4.65 muestran lo propio pero para en enlace entre Salinas y Cabuyas. Estos datos se analizan en el capítulo 5, que corresponde al análisis de resultados.

- **Enlace Salinas – Animas**

```
ITGDec version 2.8.0-rc1 (r459)
Compile-time options:
-----
Flow number: 1
From 12.0.0.6:37721
To 13.0.0.5:8999
-----
Total time = 32.505847 s
Total packets = 10003
Minimum delay = 4.542809 s
Maximum delay = 12.687090 s
Average delay = 7.252392 s
Average jitter = 0.006118 s
Delay standard deviation = 2.085612 s
Bytes received = 5121536
Average bitrate = 1260.459018 Kbit/s
Average packet rate = 307.729253 pkt/s
Packets dropped = 0 (0.00 %)
Average loss-burst size = 0.000000 pkt
-----
```

Figura 4.62 Resultados de D-ITG con Protocolo TCP, Salinas - Animas.

```

ITGDec version 2.8.0-rc1 (r459)
Compile-time options:
-----
Flow number: 1
From 12.0.0.6:45379
To    13.0.0.5:8999
-----
Total time           =      30.965081 s
Total packets        =           7703
Minimum delay        =       4.487995 s
Maximum delay        =       5.954893 s
Average delay        =       5.629160 s
Average jitter       =       0.006808 s
Delay standard deviation =    0.185472 s
Bytes received       =      3943936
Average bitrate      =    1018.937687 Kbit/s
Average packet rate  =     248.764084 pkt/s
Packets dropped      =         22289 (74.32 %)
Average loss-burst size =    36.599343 pkt
-----

```

Figura 4.63 Resultados de D-ITG con Protocolo UDP, Salinas - Animas.

- **Enlace Salinas - Cabuyas**

```

ITGDec version 2.8.0-rc1 (r459)
Compile-time options:
-----
Flow number: 1
From 12.0.0.6:37721
To    13.0.0.5:8999
-----
Total time           =      31.886934 s
Total packets        =           9123
Minimum delay        =       4.567024 s
Maximum delay        =      13.457801 s
Average delay        =       9.660173 s
Average jitter       =       0.007877 s
Delay standard deviation =    2.189754 s
Bytes received       =      3365345
Average bitrate      =     527.127847 Kbit/s
Average packet rate  =     307.729253 pkt/s
Packets dropped      =              0 (0.00 %)
Average loss-burst size =    0.000000 pkt
-----

```

Figura 4.64 Resultados de D-ITG con Protocolo TCP, Salinas - Cabuyas.


```
ITGDec version 2.8.0-rc1 (r459)
Compile-time options:
-----
Flow number: 1
From 12.0.0.6:45379
To 13.0.0.5:8999
-----
Total time = 31.114875 s
Total packets = 6248
Minimum delay = 5.967448 s
Maximum delay = 15.334879 s
Average delay = 11.215363 s
Average jitter = 0.007015 s
Delay standard deviation = 0.225863 s
Bytes received = 2363617
Average bitrate = 271.554712 Kbit/s
Average packet rate = 115.896453 pkt/s
Packets dropped = 39756 (93.67%)
Average loss-burst size = 52.778363 pkt
-----
```

Figura 4.65 Resultados de D-ITG con Protocolo UDP, Salinas - Cabuyas.

CAPÍTULO V

ANÁLISIS DE RESULTADOS

5.1 ANÁLISIS DE LOS RESULTADOS OBTENIDOS EN EL SIMULADOR *ns-2*.

5.1.1 Análisis de la Cantidad de Paquetes

Es importante realizar el análisis de la cantidad de paquetes que llegan al otro lado del enlace, ya que cada red se comporta de manera diferente y es preciso conocer si los parámetros se han establecido correctamente. En las tablas 4.25 y 4.28 correspondientes al envío de datos con tráfico TCP para el enlace Salinas – Ánimas y Salinas – Cabuyas respectivamente, se muestran la cantidad de paquetes enviados, recibidos, perdidos y caídos correspondientes a este tráfico. Aquí se puede observar primero que la cantidad de paquetes enviados es un poco menos a la de los paquetes generados, debido al tiempo que se toma en generar el paquete para en el envío; también se puede ver que no existen paquetes perdidos ya que es un protocolo que en lugar de desechar el paquete, lo retransmite hasta que llegue a su destino, por lo que podemos observar un pequeño porcentaje de paquetes retransmitidos.

Para el análisis de paquetes en el envío de tráfico UDP, se observa que no existen paquetes caídos o retransmitidos ya que UDP es un protocolo en tiempo real, es decir no es necesario que lleguen todos los paquetes enviados mientras la

información se mantenga legible, por lo que en la tabla 4.26 y 4,29 correspondientes al envío de datos con tráfico UDP para los enlaces Salinas – Ánimas y Salinas – Cabuyas respectivamente se puede observar un porcentaje de paquetes perdidos, y en 0 la cantidad de paquetes caídos.

5.1.2 Desempeño del *Throughput* en relación a la distancia.

Se realizó también el cálculo del *Throughput*, en intervalos 10 Km desde el punto de partida hasta el valor del mayor enlace a 77.5 Km. Para observar el desempeño de este parámetro mientras se incrementa la distancia. Los resultados se muestran a continuación. La tabla 5.1 presenta los datos del valor del *Throughput* para tráfico TCP.

Tabla 5.1 Valores de Throughput con el Protocolo TCP.

Distancia (m)	<i>Throughput</i> (Mbps)
10000	5,500
20000	5,382
30000	4,938
40000	4,378
50000	3,705
60000	3,142
66700	2,789
70000	2,293
77500	1,364

La Figura 5.1 muestra la gráfica de desempeño del *Throughput* según la distancia. Se observa que el *Throughput* decae notablemente mientras crece la distancia, pero su menor valor es de 1.36 Mbps lo cual es un valor aceptable para la transmisión de datos en regiones de baja densidad de usuarios.

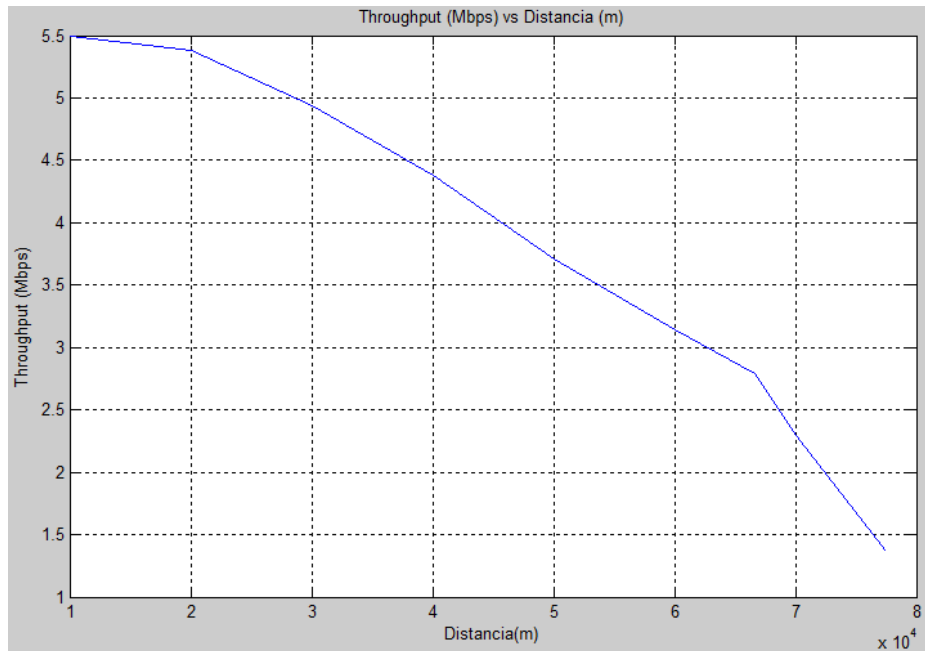


Figura 5.1 Análisis del Throughput vs Distancia con el Protocolo TCP.

De manera similar se realiza el proceso para el tráfico UDP. La tabla 5.2 presenta los datos del valor del *Throughput* para tráfico este protocolo.

Tabla 5.2 Valores de Throughput con el Protocolo UDP.

Distancia (m)	Throughput (Mbps)
10000	5,500
20000	5,144
30000	4,752
40000	4,155
50000	3,346
60000	2,421
66700	1,844
70000	1,209
77500	0,377

La Figura 5.2 muestra la gráfica de desempeño del *Throughput* según la distancia. Y de igual manera que el caso anterior, el menor valor de *Throughput* en este caso 1.21 Mbps es un valor aceptable como para la transmisión de datos.

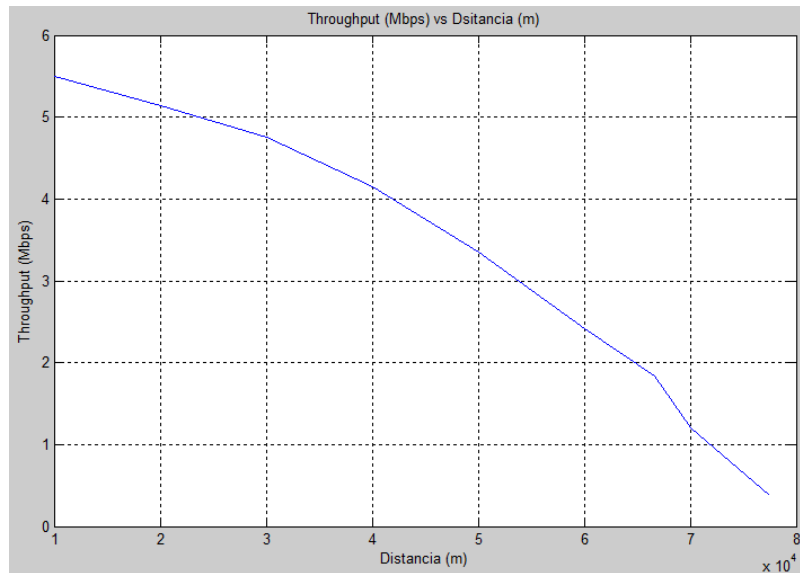


Figura 5.2 Análisis del Throughput vs Distancia con el Protocolo UDP.

La figura 5.3 muestra la comparación entre los protocolos TCP y UDP obtenidos en la simulación. Se observa claramente que el protocolo UDP tiene una menor tasa de transmisión, se debe a que TCP al permitir la retransmisión de paquetes, necesita más ancho de banda para la información de control que incorpora.

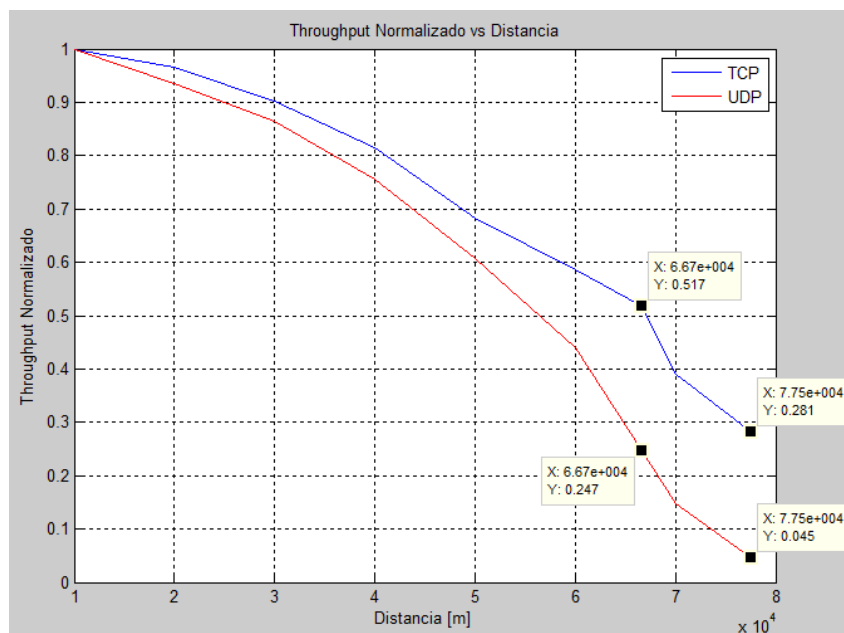


Figura 5.3. Comparación de valores simulados entre TCP y UDP.

5.2 ANÁLISIS DE LAS MEDICIONES OBTENIDAS DURANTE LA IMPLEMENTACIÓN DE LOS ENLACES.

El análisis de estos parámetros tiene como fin, revisar y analizar la calidad de servicio (QoS) de las redes implementadas. La calidad de servicio es el rendimiento de extremo a extremo de los servicios de red tal como lo percibe el usuario final. Los parámetros de QoS son: el retardo, la variación del retardo, la pérdida de paquete y *throughput*. Una red debe garantizar un cierto nivel de calidad de servicio para un nivel de tráfico que sigue un conjunto especificado de parámetros.

Durante cada inyección de tráfico, D-ITG guarda los ficheros con información sobre los resultados, llamados ficheros *logs*. Estos ficheros se almacenan en un servidor de *logs* que en estos experimentos fue el computador receptor del tráfico. La información recogida en los ficheros se procesa mediante comandos de D-ITG para obtener unos ficheros *.dat* que contienen la información sobre el *throughput*, la latencia, los paquetes perdidos y el *jitter*. Estos ficheros pueden ser fácilmente procesados con cualquier herramienta de análisis de datos para generar las gráficas de resultados. Se realizaron las pruebas en los dos enlaces, en cada uno de ellos se utilizó los protocolos UDP y TCP, su análisis se va a detallar posteriormente.

5.2.1 Delay

El *Delay* o tiempo de latencia, mide la cantidad de tiempo que tarda un paquete en viajar desde el origen hasta el destino. Juntos, la latencia y el ancho de banda definen la velocidad de una red. El tiempo de latencia se ve influenciado por tres cosas: la distancia por la que los datos deben viajar, el tamaño de los paquetes y el número de redes que existen entre los terminales.

- **Enlace Salinas - Animas.**

Las Figura 5.4 y 5.5 muestran el comportamiento de la latencia frente al tiempo, para un tamaño de paquete de 512 bytes, el retardo existente en la red fue obtenido después de realizar las pruebas con el inyector D-ITG durante un tiempo de 30 segundos. La inyección del tráfico corresponde al flujo de paquetes que se envían desde la dirección 12.0.0.6 que corresponde al transmisor, a la dirección destino 13.0.0.5, el flujo agregado entregó como resultados retardos promedios de 7.25 y 5.62 segundos para TCP y UDP correspondientemente.

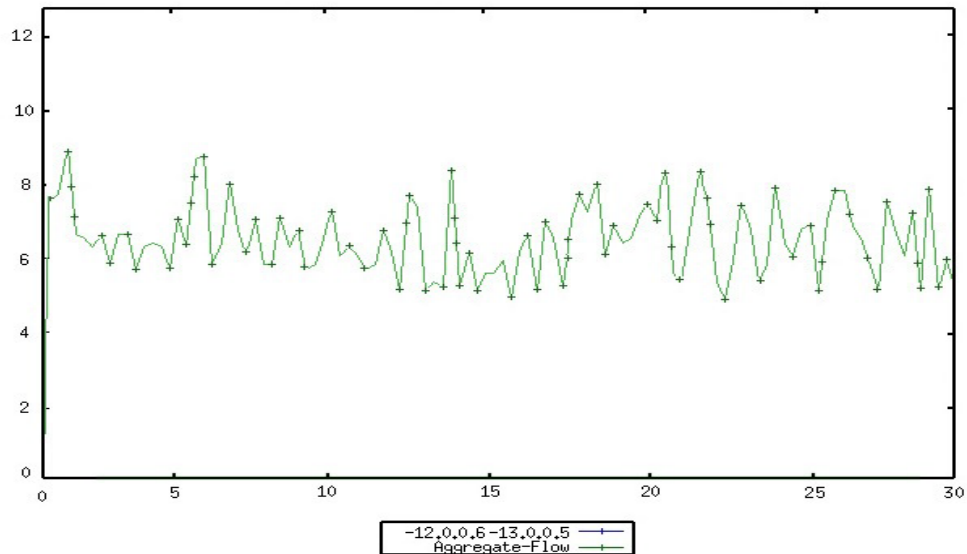


Figura 5.4 Delay con TCP Salinas – Animas.

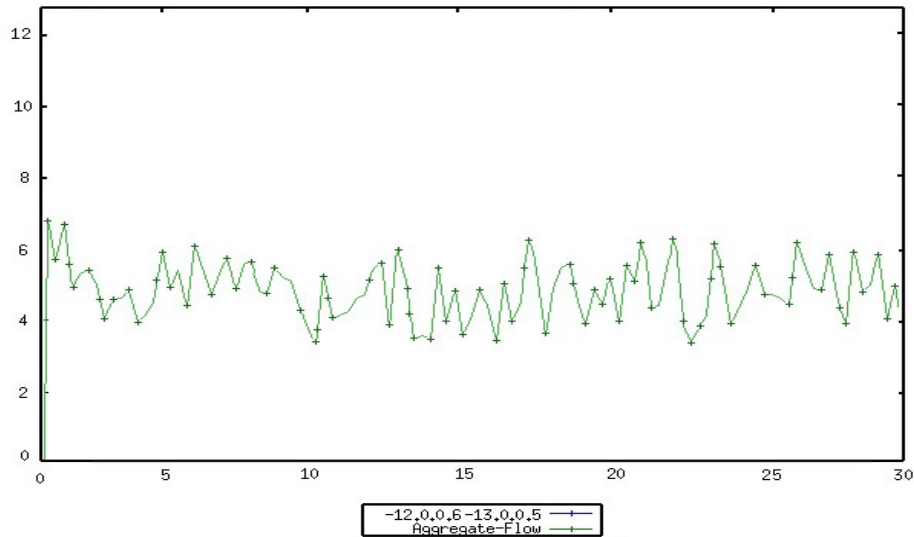


Figura 5.5 Delay con UDP Salinas – Animas.

- **Enlace Salinas – Cabuyas**

En las Figuras 5.6 y 5.7 muestran el retraso en los paquetes de acuerdo a la variación del tiempo. La inyección del tráfico corresponde al flujo de paquetes que se envían desde la dirección 12.0.0.6 del transmisor a la dirección destino 13.0.0.5, el flujo agregado entregó como resultados retardos promedios de 9.66 y 11.21 segundos para TCP y UDP correspondientemente, teniendo unos picos que sobrepasan los 13 s.

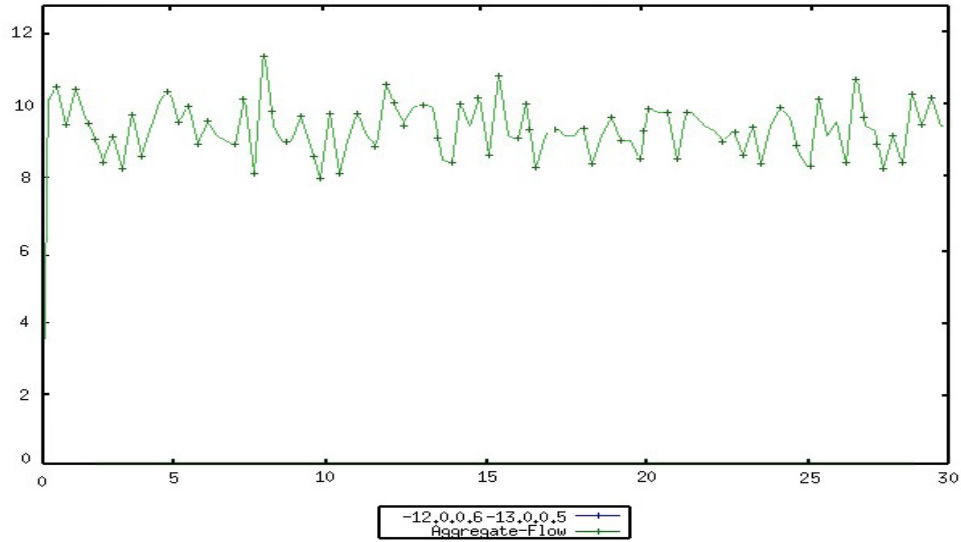
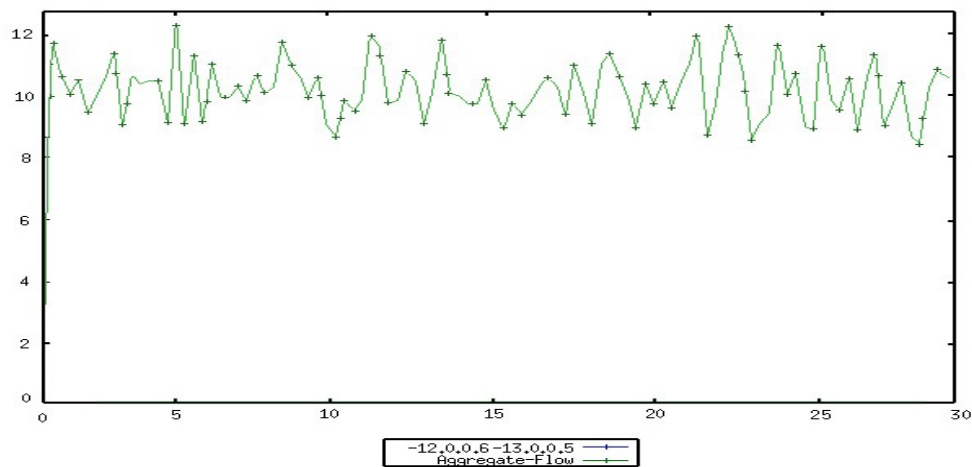


Figura 5.6 Delay con TCP Salinas – Cabuyas.



3Figura 5.7 Delay con UDP Salinas – Cabuyas.

5.2.2 Jitter

El *Jitter* es la variación en cuanto a la cantidad de latencia entre paquetes de datos que se reciben, es decir el tiempo con el que llega cada paquete al receptor, es exclusivamente de uso digital aunque sus efectos son muy parecidos al de la naturaleza análoga, cuando el *jitter* es de valores bajos, su comportamiento es despreciable pero, este parámetro puede afectar mucho a la calidad de servicio

cuando tiene valores altos en relación a la red con la que se trabaje, ya que afecta a la calidad de flujo que puede llegar a ser muy variable, esto se da por el retardo en el que un paquete varía impredeciblemente con su posición en las colas de los ruteadores y debido también a los tiempos de espera de conexión, congestión del tráfico de datos y la interferencia a lo largo del camino entre el transmisor.

- **Enlace Salinas - Animas.**

En las Figuras 5.8 y 5.9 se puede observar que existe una diferencia de amplitud al momento de la transmisión de datos, con un promedio mínimo de jitter de 0,0061 y 0,0068 segundos para TCP y UDP correspondientemente, esto se produce debido a que en los periodos en el cual se envía la información existe congestionamiento en la red, por causa de pérdida de sincronización o por las diferentes rutas seguidas por los paquetes para llegar al destino.

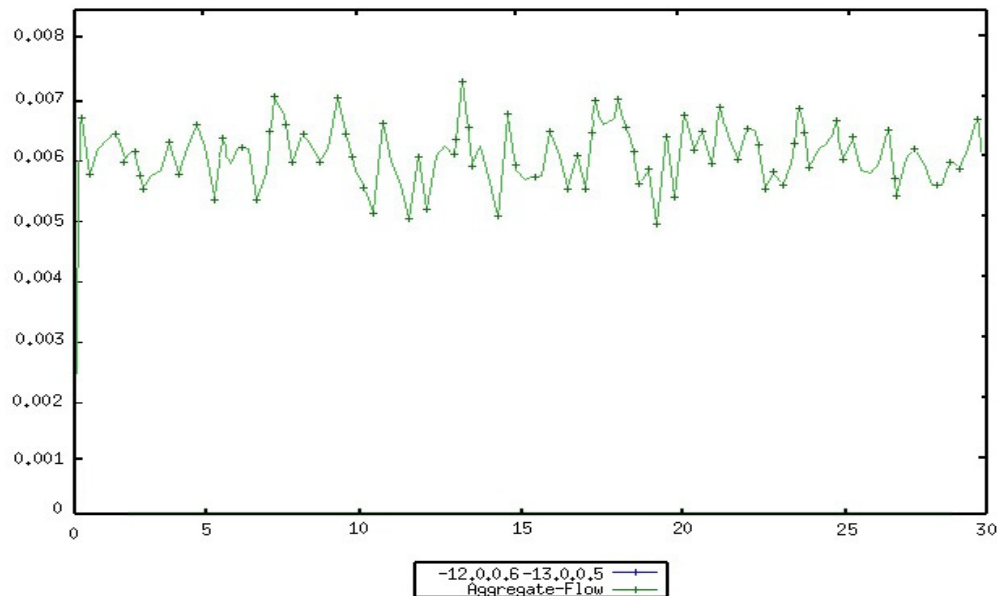


Figura 5.8 Jitter con TCP Salinas – Animas.

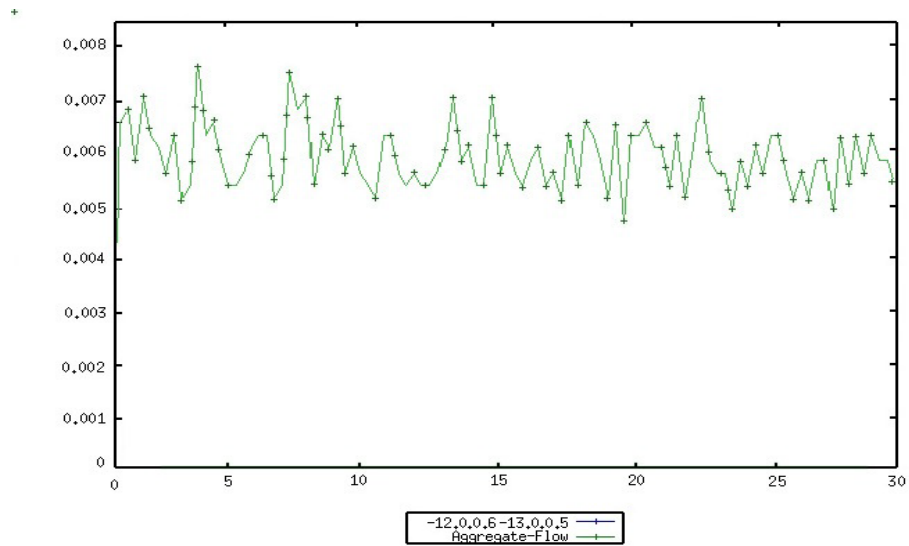
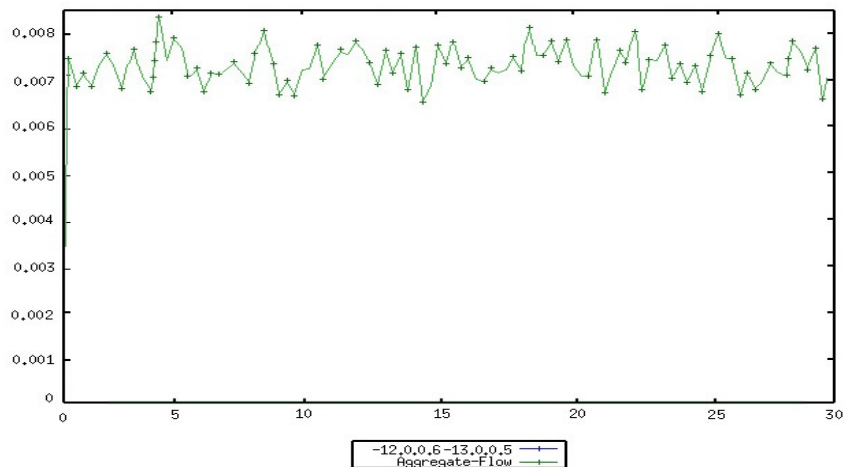


Figura 5.9 Jitter con UDP Salinas – Animas.

- **Enlace Salinas – Cabuyas**

La Figuras 5.10 y 5.11, presentan un promedio mínimo *jitter* de 0,0078 y 0,007 segundos para TCP y UDP correspondientemente. Se comprueba con estos datos que el valor de *jitter* incrementa conforme incrementa el flujo enviado, ya que el tráfico se vuelve más robusto y se comporta de manera irregular.



Figuras. 5.10 Jitter con TCP Salinas – Cabuyas.

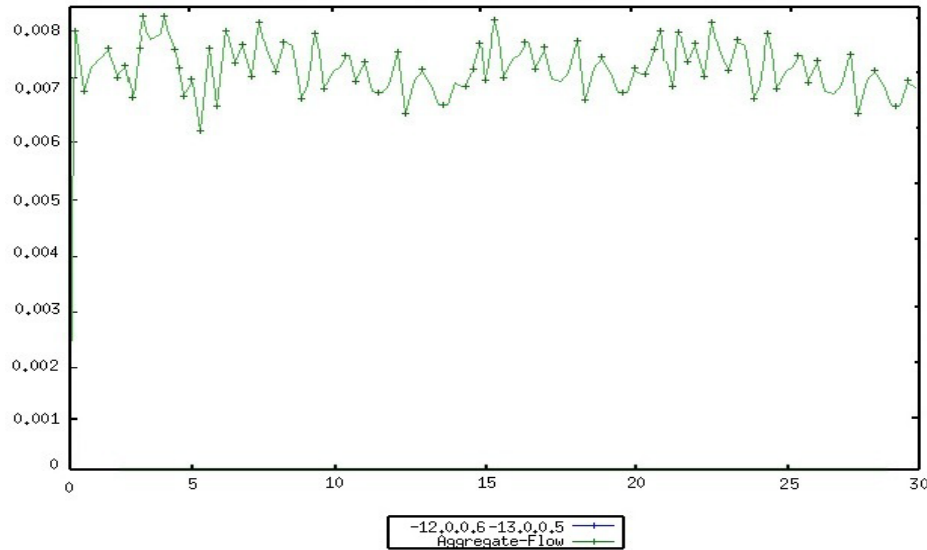


Figura 5.11 Jitter con UDP Salinas – Cabuyas.

5.2.3 Paquetes Perdidos

La pérdida de paquetes es uno de los problemas más comunes de las redes Wi-Fi y vienen después de que el tiempo de espera para la recepción del paquete se haya agotado, como estas redes son mayormente compartidas, existe gran cantidad de colas y congestión de paquetes, interferencia adyacente y competencia de acceso al canal, cuando se encuentren varios usuarios conectados, la pérdida de paquetes dependerá también del protocolo de transmisión, el protocolo TCP no admite pérdida de paquetes sino que retransmite un mismo paquete hasta que llegue a su destino. El protocolo UDP por el contrario permite que si un paquete no llegó, el que viene detrás lo pueda intentar, descartando completamente al primero.

- **Enlace Salinas - Animas.**

Las Figuras 5.12 y 5.13 se muestran los paquetes perdidos en la transmisión de la red implementada durante los 30 segundos que duró, se observa que a lo largo de la transmisión con el protocolo TCP no existieron paquetes perdidos lo que concuerda con las razones mencionadas anteriormente, y mantiene un valor del 0%,

TCP realiza retransmisiones cuando existen paquetes caídos pero esto afecta a su eficiencia ya que el tamaño de los paquetes es mayor en TCP debido a la estructura de su trama. Mientras que en UDP el porcentaje de paquetes perdidos se incrementa de forma directamente proporcional a la velocidad de transmisión de envío, con un valor total de paquetes perdidos de 22289, la pérdida considerable de paquetes se debe a la gran distancia del enlace y fallos en el hardware y software.

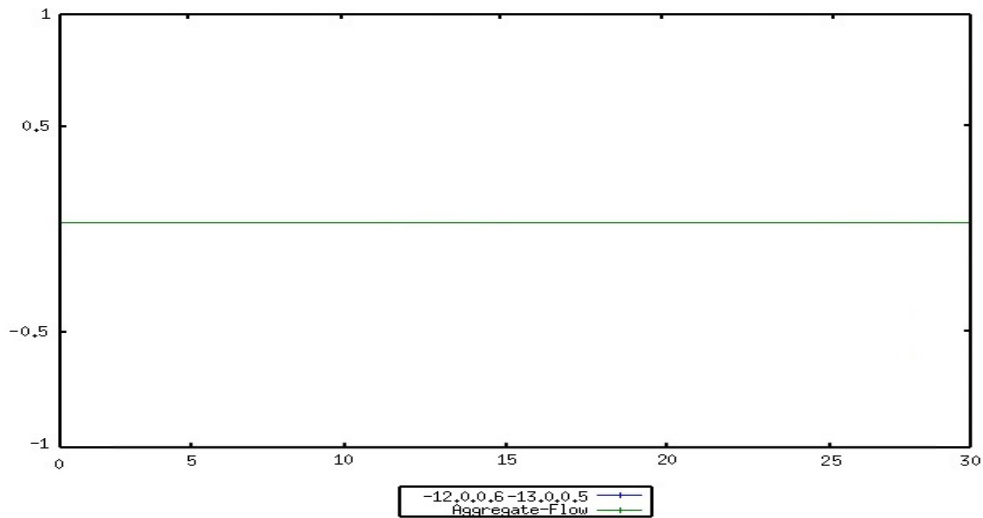


Figura 5.12 Paquetes Perdidos en D-ITG con TCP Salinas – Animas.

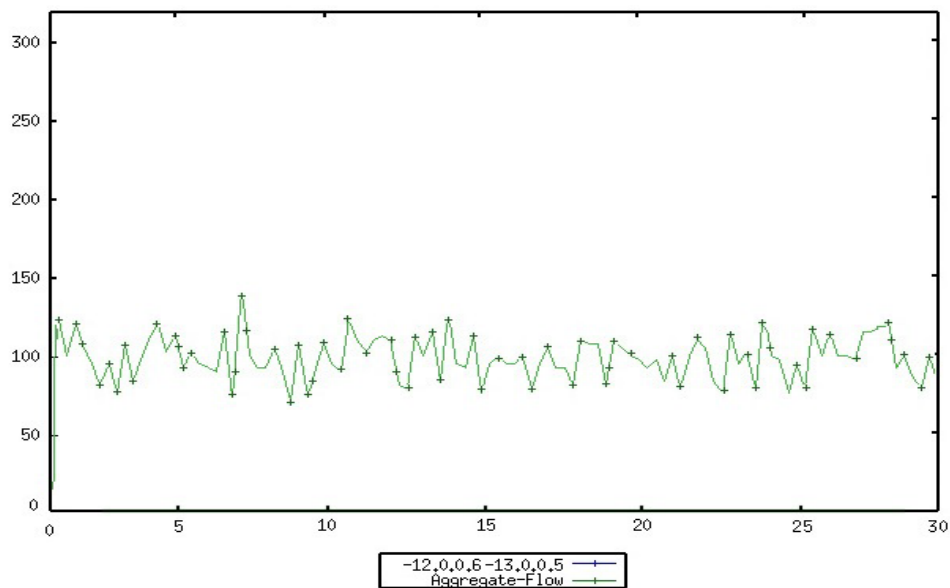


Figura 5.13 Paquetes Perdidos en D-ITG con UDP Salinas – Animas.

- **Enlace Salinas – Cabuyas**

Para este enlace, las Figuras 5.14 y 5.15 muestran los paquetes perdidos en la transmisión. Durante este periodo se observa con el protocolo TCP no existen paquetes perdidos y se mantiene en un valor del 0%, al igual que en enlace anterior. Por otro lado en UDP el porcentaje de paquetes perdidos se incrementa de forma directamente proporcional a la velocidad de transmisión de envío, con un valor total de paquetes perdidos de 39756.

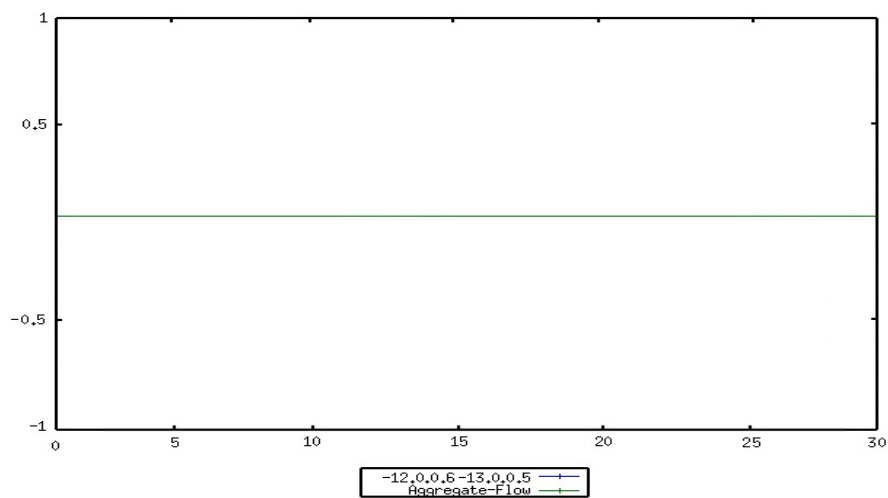


Figura 5.14 Paquetes Perdidos en D-ITG con TCP Salinas – Cabuyas.

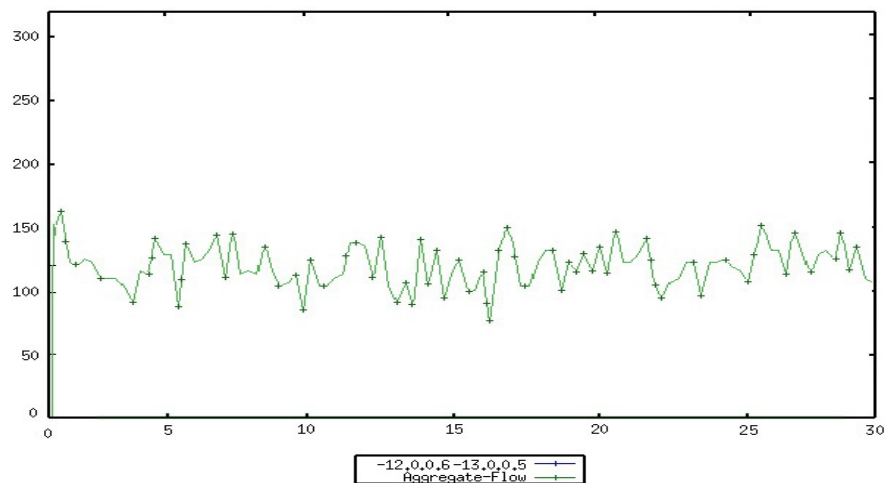


Figura 5.15 Paquetes Perdidos en D-ITG con UDP Salinas – Cabuyas.

5.2.4 *Throughput*

A pesar de que a veces es definido como el tiempo durante el cual se analiza la calidad de servicio, el *throughput*, es el rendimiento de la red, que está dado por la cantidad de bits que pueden ser transmitidos en un determinado tiempo.

Se puede decir que es el volumen real de datos que atraviesa la red en cualquier sentido. Está ligado con la velocidad de trasmisión, el ancho de banda del canal, y el número de usuarios que comparten la red, si es que existen menos usuarios, el canal estará libre y se tendrá mayor espacio para la transmisión. Se debe diferenciar al *throughput* del ancho de banda que es el termino con el cual se lo suele confundir fácilmente, si se tiene un ancho de banda para varios canales de transmisión, cada canal tendrá su propio *throughput*.

En redes inalámbricas y más específicamente para redes Wi-Fi, los factores que hace que el *throughput* sea inferior al ancho de banda máximo son, en su mayoría por el medio físico:

- ✓ Usar un medio compartido: como 802.11b, En general dos no pueden transmitir a la vez, y los mecanismos para acceder al medio sin pegarnos con las otras estaciones puede hacer que baje el rendimiento.
- ✓ Los protocolos que intervengan: en el caso de TCP hay que esperar paquetes ACK de vuelta al enviar, con lo que se ve afectado el rendimiento, sobre todo si se pierden esos mensajes de vuelta porque enviamos muy rápido.
- ✓ También depende de la relación entre los tiempos dedicados a la transmisión y control de tráfico. Detallados en el capítulo 2. Mientras estos sean más pequeños, el *throughput* será menor.

- **Enlace Salinas - Animas.**

Para cada enlace (TCP y UDP) se desea observar el desempeño de la red con una tasa de transmisión de datos de 5.5 Mbps, y un flujo de datos con 512 bytes. Por las causas mencionadas anteriormente, en las redes inalámbricas IEEE 802.11, la presencia de interferencias produce considerablemente la baja de las tasas de transmisión de datos. El resultado de esa acción es que se disminuye la capacidad de la red.

El resultado se observa en que disminuye la capacidad de la red mientras se incrementa la distancia, como se muestra en las Figuras 5.16 y 5.17. Se obtiene un valor promedio de 1260.459018 Kbit/s para el protocolo TCP y de 1018.937687 Kbit/s para el protocolo UDP.

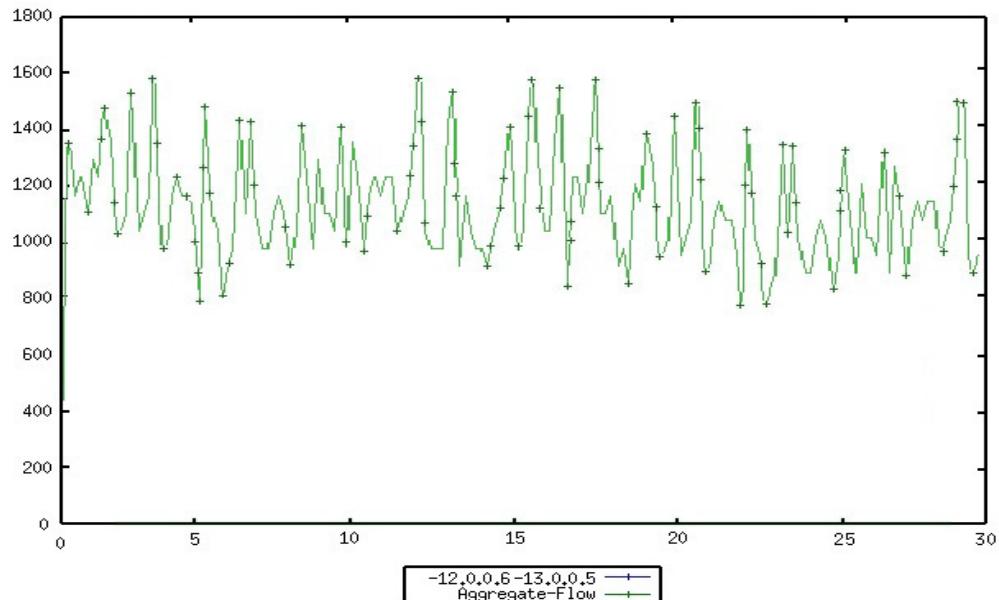
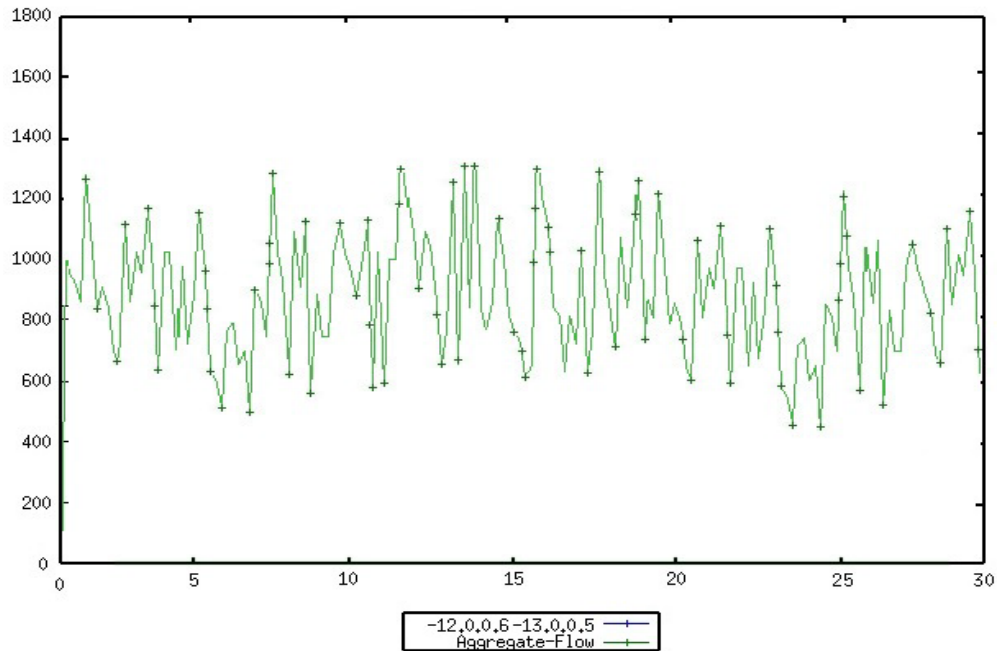


Figura 5.16 Throughput con TCP Salinas - Animas.



4Figura 5.17 Throughput con UDP Salinas - Animas.

Comparando con el valor obtenido de la simulación de la red en *ns-2* mejorado (TeNS), en el capítulo 4, se encuentra, que el resultado de la simulación con el protocolo TCP fue de 2.789 Mbps, y el valor contenido con por el software D-ITG para este mismo protocolo es de 1.260 Mbps.

Igualmente en el análisis del protocolo UDP, los resultados fueron, de 1.844 Mbps, para la simulación, y 1.018 Mbps con el inyector de tráfico.

- **Enlace Salinas – Cabuyas**

De manera análoga al primer enlace, las figuras 5.18 y 5.19 muestran los valores de *throughput* para TCO y UDP obtenidos del el enlace sobre mar, obteniendo un valor medio de 527.127847 Kbit/s para el protocolo TCP y 271.554712 Kbit/s para el protocolo UDP.

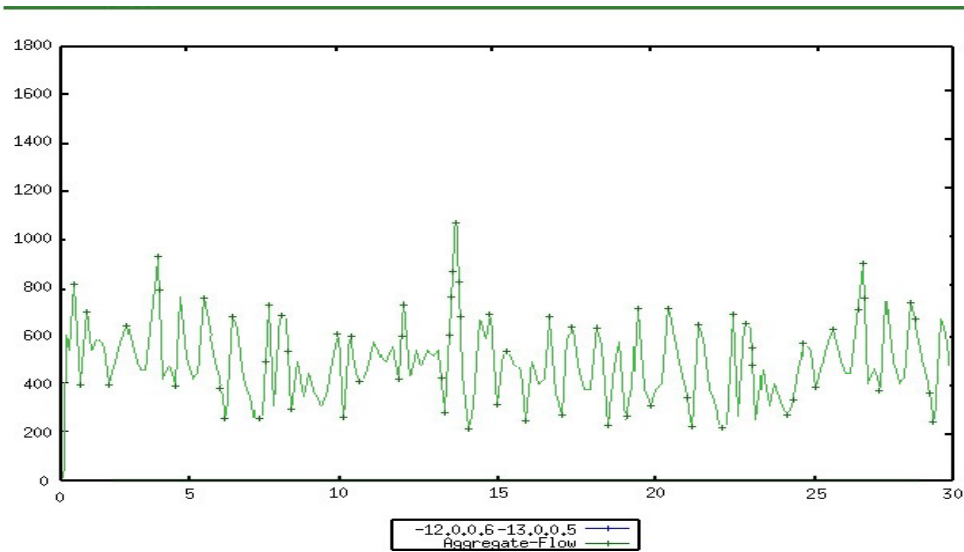


Figura 5.18 Throughput con TCP Salinas - Cabuyas.

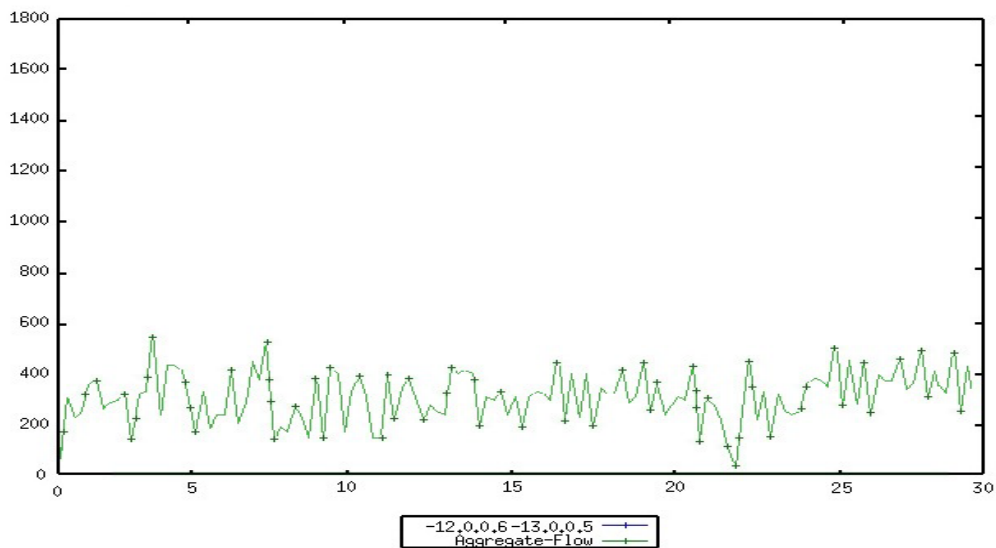


Figura 5.18 Throughput con UDP Salinas - Cabuyas.

Comparando con el valor obtenido de la simulación de la red en *ns-2* mejorado (TeNS), en el capítulo 4, se encuentra, que el resultado de la simulación con el protocolo TCP fue de 1.364 Mbps, y el valor contenido con por el software D-ITG para este mismo protocolo es de 527.127 Kbps.

Igualmente en el análisis del protocolo UDP, los resultados fueron, de 377.533 Kbps, para la simulación, y 271.554 Kbps con el inyector de tráfico.

5.3 ANÁLISIS COMPARATIVO: SIMULACIÓN VS MEDICIONES

La figura 5.20 muestra las gráficas obtenidas con los valores, simulados, calculados y medidos de los dos enlaces con tráfico TCP.

Se puede observar que para el enlace Salinas – Ánimas correspondiente a los puntos ubicados a 6,6 Km, mediante el simulador se esperaba obtener un Throughput de 2,84 Mbps, pero en los resultados medidos, se obtuvo 1,26 Mbps. Esto se debe a razones antes mencionadas como los equipos que intervinieron en los enlaces, pudiendo optimizar éstos enlaces al cambiar de equipos.

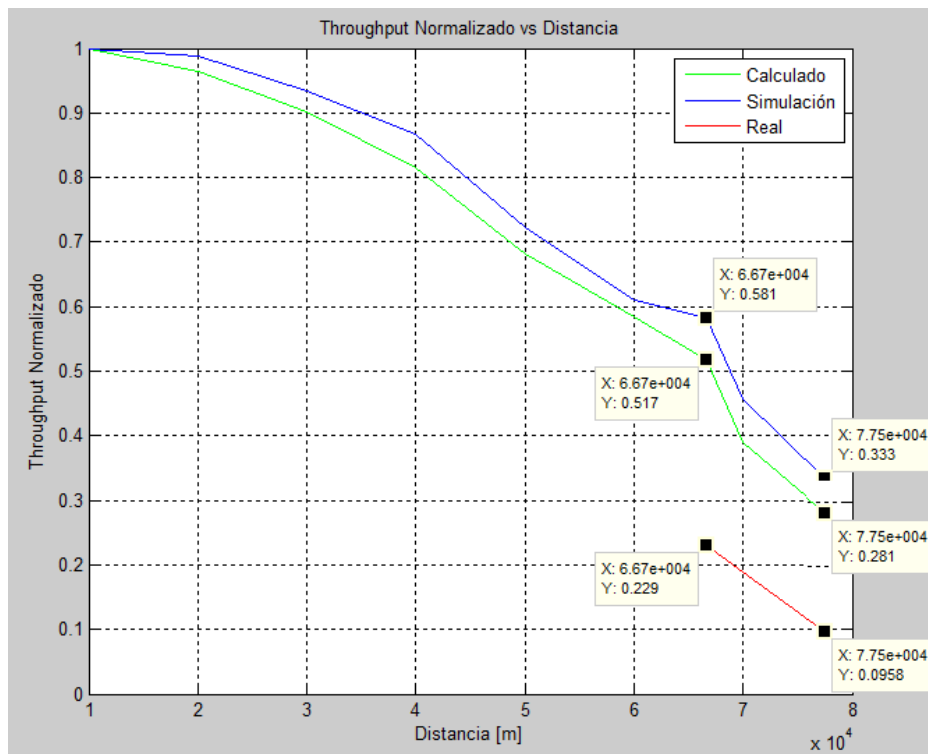


Figura 5.20. Valores de throughput, simulados, calculados y medidos con el protocolo TCP.

Para el enlace Salinas – Cabuyas que corresponde a los puntos ubicados en 77,6 Km, se observa que el resultado esperado por el simulador es de 1.55 Mbps, y el medido fue de 0,527 Mbps; el valor arrojado por el simulador ya es un poco bajo para aplicaciones que requieran de un gran ancho de banda, el valor medido es mucho más bajo lo que podría perjudicar el envío de ciertos datos, este error se debe principalmente a la distancia del enlace.

La figura 5.21 muestra las gráficas obtenidas tanto en el simulador como en la medición de los dos enlaces con tráfico UDP.

Para el enlace Salinas – Ánimas correspondiente a los puntos ubicados a 6,6 Km, mediante el simulador se esperaba obtener un *throughput* de 1.36 Mbps, en los resultados medidos, se obtuvo 1,01 Mbps. Aquí se observa una variación menor entre el simulador y la medición que para el caso de TCP, esto se debe a que UDP aprovecha mejor el ancho de bando al no necesitar tanta información de control.

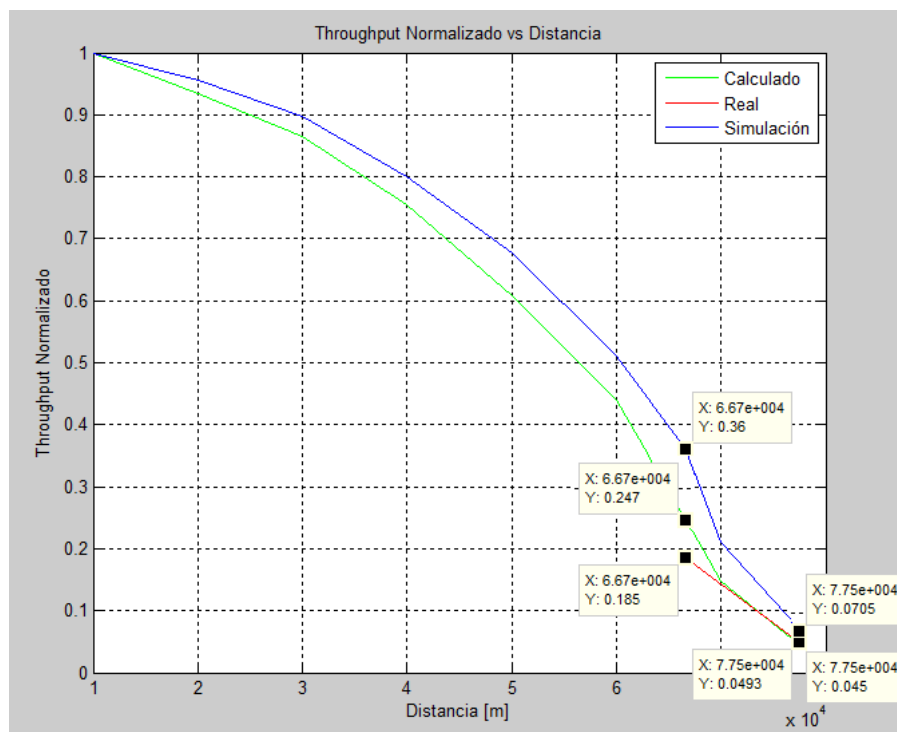


Figura 5.21. Comparación de valores simulados, calculados y medidos con el protocolo UDP.

Para el enlace Salinas – Cabuyas que corresponde a los puntos ubicados en 77,6 Km, se observa que el resultado esperado por el simulador es de 0.24 Mbps, y el medido fue de 0,271 Mbps. En este caso los valores están bastante próximos uno del otro, la distancia es el principal factor en el bajo ancho de banda, y las adecuaciones realizadas en el simulador fueron decayendo a esta distancia.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

Se analizaron los parámetros presentes en la capa MAC del estándar IEEE 802.11 y se modificaron los tiempos del *ACK-timeout*, *CTS-timeout* y *Slottime*, con los cambios realizados en dichos tiempos se consiguió desplegar enlaces de largo alcance, para conseguirlo se determinó que se debe aumentar el *ACK-timeout*, para que el paquete no pueda perderse debido al tiempo de espera del receptor.

Se realizó un estudio referente a las velocidades de transmisión y se concluyó que mientras mayor sea la velocidad a utilizarse en los enlaces, la potencia de recepción también debe incrementarse para mantener una tasa baja de *bits* erróneos, para lograr los enlaces de largo alcance, se utilizó una tasa 5.5 Mbps en la implementación de las redes WiLD.

Mediante una observación del proyecto previamente realizado con implementaciones WiLD en el Ecuador, se realizaron los cambios respectivos en factores como la climatización, medio de propagación, distancias y accesos, para así determinar los puntos de implementación óptimos y poder analizar el comportamiento y desempeño de una red WiLD en un ambiente diferente , un enlace sobre mar y el otro sobre tierra.

Los puntos de implementación se encuentran en la Región Costa del Ecuador y tuvieron lugar en la Base Naval de Salinas, Cerro Ánimas y Cerro Cabuyas, los cuales cuentan con línea de vista notable y adecuada para cada enlace, fácil acceso a la infraestructura de telecomunicaciones y al suministro de alimentación eléctrica, cuentan también con comunicación dedicada entre ellos.

Se seleccionó el software de simulación más adecuado que logre predecir el comportamiento de los enlaces a larga distancia y brindar un análisis más detallado de los mismo, el software utilizado fue el *ns-2*, pero con la adición de un parche llamado *TeNS*, necesario para simulaciones de largo alcance en donde hace referencia a prestaciones reales utilizadas en redes WILD, como la inclusión de antenas directivas, interferencia co-canal y protocolos de enrutamiento adecuados para redes inalámbricas, este software también presente facilidades en el estudio del desempeño de la red, con una gran gama de herramientas que permiten un análisis detallado de parámetros como *throughput*, *delay*, *jitter*, paquetes perdidos, paquetes enviados, entre otros.

Se determinó que el Sistema Operativo más eficaz con el que se va a trabajar es Linux Voyage 0.6.5 ya que pruebas y estudios anteriores demuestran un buen rendimiento y desempeño de este software en enlaces referidos a WILD, además la tarjetas utilizadas en este proyecto son compatibles y trabajan sin ningún problema con la inclusión de este Sistema Operativo. Se debe notar que el Sistema Operativo Linux Voyage trabaja con los driver *Madwifi* y *Hostapd*, a partir de esto se concluyó también que el uso de el driver *Madwifi* es el adecuado y necesario ya que permite modificar los valores de *ACKtimeout*, *ctstimeout* y *Slottime* presentes en la capa MAC del estándar IEEE 802.11. Mientras que el driver *Hostapd* no permite el cambio en los valores mencionados de la capa MAC y los enlaces aceptables pueden llegar hasta 30 Km. A partir de la distribución Linux Voyage 0.7.0 ya no se incluye el driver *Madwifi*, es por eso que se utilizó la versión 0.6.5 de este Sistema Operativo.

Se realizó una comparación y estudio de los equipos estandarizados y no estandarizados, necesarios y disponibles en el Centro de Investigación y Desarrollo de la ESPE, entre los equipos se destaca las radios inalámbricas *Ubiquiti Networks*, utilizadas para implementaciones de largas distancias, su estudio se basó en parámetros como la sensibilidad de recepción, potencia de transmisión, rangos de frecuencia y en la temperatura que pueden soportar estas radios en operación, así como también el manejo detallado de las mismas para lograr un óptimo funcionamiento de la red, por esto se determinó que las radios inalámbricas *Ubiquiti Networks* son la mejor opción para desplegar nuestra redes WiLD.

De acuerdo con los valores simulados y los reales se concluyó que el simulador brinda valores que varían de los reales, para el enlace Salinas – Ánimas el valor simulado fue de 2,84 Mbps y el medido fue 1,26 Mbps para el tráfico TCP, para el tráfico UDP el valor simulado fue de 1,36 Mbps y el medido fue de 1,01 Mbps. Para el enlace Salinas – Cabuyas la diferencia de valores es menor, con tráfico TCP el valor simulado fue de 1,55 Mbps y el medido fue de 0,527, con el tráfico UDP el valor simulado fue de 0,24 Mbps y el medido fue de 0,27 Mbps, por tanto se debe tomar en cuenta que los valores reales también presentan complicaciones, en mayor parte debido a pérdidas por propagación, pérdidas de cables, pérdidas conectores, y la distancia.

6.2 RECOMENDACIONES.

El uso de la tecnología Wi-Fi es una opción factible si se desea realizar enlaces a larga distancia, siempre y cuando no se presentes problemas de interferencia, es por eso que se recomienda cambiar el uso de rango de frecuencias de la banda de 2.4 GHz a la banda de 5 GHz que presenta menos problemas de interferencia pero muestra un índice mayor respecto a las pérdidas por absorción debido a la lluvia.

Se recomienda un estudio más detallado del software de simulación propuesto *ns-2* y de otros posibles para redes inalámbricas a larga distancia. Para seguir la comparación de cuál es el más óptimo para estas redes y tener más apoyo para en un futuro lograr la estandarización de la tecnología WiLD

Es de gran importancia definir el tipo de sistema operativo así como la versión a utilizar, ya que la gran mayoría de sistemas no brinda las prestaciones necesarias para la realización de un enlace a larga distancia.

Se recomienda realizar los estudios pertinentes en el uso de la tecnología Wi-Fi para aumentar el *throughput* en este tipo de enlaces, ya que la capacidad de red es un factor importante para brindar un sistema eficiente.

Es recomendable realizar una comparación entre las mediciones obtenidas y trabajos realizados sobre este tipo de redes, para así tener una referencia en aspectos técnicos y así poder determinar si la red desplegada es adecuada y cumple con ciertos parámetros de calidad.

ANEXO 1

INSTALACIÓN DEL SOFTWARE *ns-2* ALLINONE 2.35

Como antesala para la *ns-2*, se debe instalar las siguientes dependencias y librerías:

- ✓ buildessential
- ✓ autoconf
- ✓ automake
- ✓ libxmu-dev

Para instalarlas se abre un terminal y se ejecuta el comando:

```
sudo apt-get install build-essential autoconf automake libxmu-dev
```

```
root@ubuntu:~# sudo apt-get install build-essential autoconf automake libxmu-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
autoconf is already the newest version.
automake is already the newest version.
build-essential is already the newest version.
libxmu-dev is already the newest version.
You might want to run 'apt-get -f install' to correct these:
The following packages have unmet dependencies:
 libstdc++6-4.4-dev-armel-cross : Depends: g++-4.4-arm-linux-gnueabi (= 4.4.5-15
 ubuntu1.40) but it is not going to be installed
E: Unmet dependencies. Try 'apt-get -f install' with no packages (or specify a s
olution).
```

Dependencias descargadas: Paquetes y librerías

Ahora se descarga el programa *ns-2* allinone donde ya viene empaquetado todo lo necesario para su funcionamiento.

Se accede al directorio *src* y se ejecuta el comando:

```
cd /usr/src
```

```
wget http://ufpr.dl.sourceforge.net/project/nsnam/allinone/ns-allinone-2.35/ns-
allinone-2.35.tar.gz
```

Se descomprime el paquete ns-allinone-2.35.tar.gz ejecutando el comando:

```
tar xvfz ns-allinone-2.35.tar.gz
```

Luego se ingresa en la carpeta descomprimida con el comando:

```
cd ns-allinone-2.35
```

Para la instalación se ejecuta el comando “ ./install “ y se espera a que compile el programa, este proceso tardara un poco.

```
Ns-allinone package has been installed successfully.
Here are the installation places:
tcl8.5.10:      /home/fernando/Desktop/tesis/instaladores/ns-allinone-2.35/{bin,include,lib}
tk8.5.10:      /home/fernando/Desktop/tesis/instaladores/ns-allinone-2.35/{bin,include,lib}
otcl:          /home/fernando/Desktop/tesis/instaladores/ns-allinone-2.35/otcl-1.14
tclcl:         /home/fernando/Desktop/tesis/instaladores/ns-allinone-2.35/tclcl-1.20
ns:            /home/fernando/Desktop/tesis/instaladores/ns-allinone-2.35/ns-2.35/ns
nam:           /home/fernando/Desktop/tesis/instaladores/ns-allinone-2.35/nam-1.15/nam
xgraph:        /home/fernando/Desktop/tesis/instaladores/ns-allinone-2.35/xgraph-12.2
gt-itm:        /home/fernando/Desktop/tesis/instaladores/ns-allinone-2.35/itm, edriver, sgb2alt, sgb2ns, sgb2comns, sgb2hierns
```

Compilación ns-2

Configurar las variables de entorno

Para añadir las variables de entorno que solicita el programa es necesario abrir el archivo bashrc .Se ejecuta el comando:

```
gedit ~/.bashrc
```

Con este comando se abre un archivo de texto, en el cual se agregan las siguientes líneas al final:

```
# LD_LIBRARY_PATH
OTCL_LIB=/usr/src/ns-allinone-2.35/otcl-1.14
NS2_LIB=/usr/src/ns-allinone-2.35/lib
X11_LIB=/usr/X11R6/lib
USR_LOCAL_LIB=/usr/local/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$X11_LIB:$USR_LOCAL_LIB
# TCL_LIBRARY
TCL_LIB=/usr/src/ns-allinone-2.35/tcl8.5.10/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB

# PATH
XGRAPH=/usr/src/ns-allinone-2.35/bin:/usr/src/ns-allinone-
2.35/tcl8.5.10/unix:/usr/src/ns-allinone-2.35/tk8.5.10/unix
NS=/usr/src/ns-allinone-2.35/ns-2.35/
NAM=/usr/src/ns-allinone-2.35/nam-1.15/
export PATH=$PATH:$XGRAPH:$NS:$NA
```

Configuración de variables de entorno

Finalmente se debe recargar el archivo bashrc ejecutando el comando:

```
source ~/.bashrc
```

Como paso final se reinicia el ordenador y se prueba la instalación ejecutando en un terminal el comando “ns”. Si la respuesta que se obtiene es el siguiente prompt “%” quiere decir que la instalación está correcta, para salir del prompt % ejecutamos “exit”.

```
fernando@ubuntu:~$ sudo su
root@ubuntu:/home/fernando# ns
%
```

Ejecución del comando ns.

Validación del programa.

Se accede al directorio ns-35

```
cd /usr/src/ns-allinone-2.35/ns-2.35/ y,
```

se aplica el comando:

```
sudo ./validate
```

Es importante ejecutar el comando validate con sudo para evitar tener problemas con los permisos.

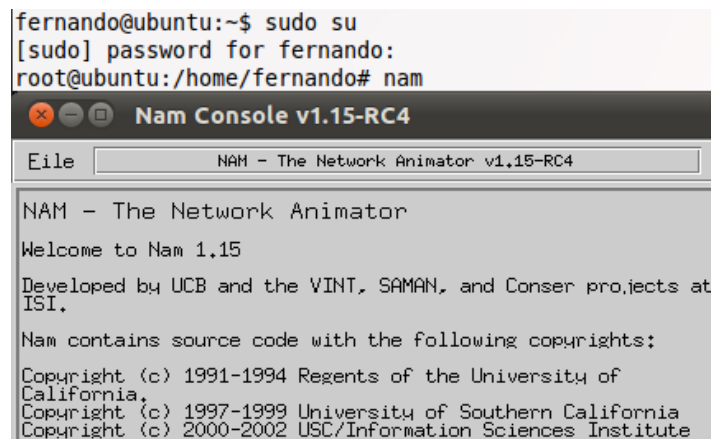
Por último necesitamos descargar la interfaz gráfica “*nam*”, para eso escribimos en la línea de comando:

Sudo apt-get install nam

```
root@ubuntu:/home/fernando/Desktop/tesis/instaladores/ns-allinone-2.35# cd
root@ubuntu:~# sudo apt-get install nam
Reading package lists... Done
Building dependency tree
Reading state information... Done
nam is already the newest version.
You might want to run 'apt-get -f install' to correct these:
The following packages have unmet dependencies:
 libstdc++6-4.4-dev-armel-cross : Depends: g++-4.4-arm-linux-gnueabi (= 4.4.5
 ubuntu1cross1.40) but it is not going to be installed
E: Unmet dependencies. Try 'apt-get -f install' with no packages (or specify
olution).
```

Instalación nam.

Para verificar la instalación de la interfaz gráfica se digita nam, si está instalado correctamente aparece la ventana de consola de Nam.



```
fernando@ubuntu:~$ sudo su
[sudo] password for fernando:
root@ubuntu:/home/fernando# nam
```

Nam Console v1.15-RC4

File NAM - The Network Animator v1.15-RC4

NAM - The Network Animator
Welcome to Nam 1.15
Developed by UCB and the VINT, SAMAN, and Conser projects at
ISI.
Nam contains source code with the following copyrights:
Copyright (c) 1991-1994 Regents of the University of
California,
Copyright (c) 1997-1999 University of Southern California
Copyright (c) 2000-2002 USC/Information Sciences Institute

Consola Nam.

ANEXO 2

INSTALACIÓN GCC-3.3 y G++-3.3 (TeNS)

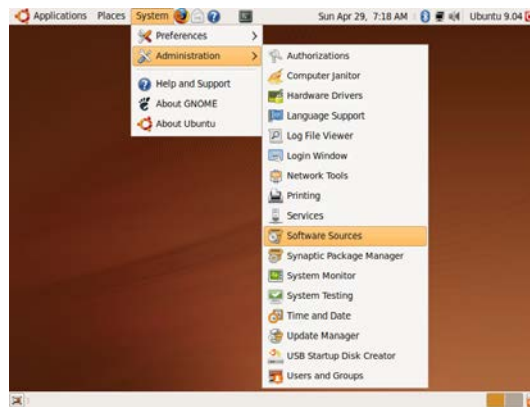
Para la instalación del gcc-3.3 y g++-3.3 se deben seguir los siguientes pasos:

Se remueve la versión actual del gcc instalada por defecto escribiendo en el terminal ***sudo apt-get remove gcc-4.3*** como se muestra en la Figura:

```
cirad@ubuntu:~$ sudo apt-get remove gcc-4.3
```

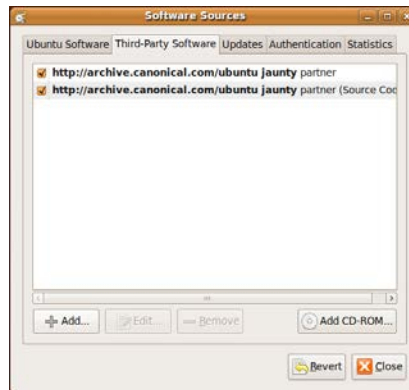
Remover la versión actual del gcc

Dentro del menú *System – Administration* seleccionar la opción *Software Sources* así:



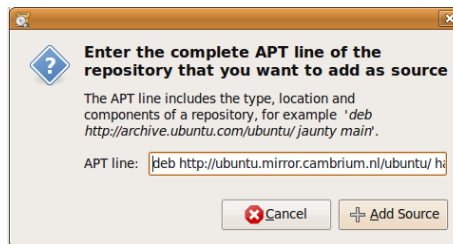
Acceso a System – Administration – Software Sources

En la ventana *Software Sources* se selecciona la pestaña *Third-Party Software* como se muestra en la Figura, la cual permite seleccionar las fuentes desde las cuales se van a descargar las aplicaciones que se deseen instalar. Para agregar una nueva fuente seleccionar la opción *Add*:



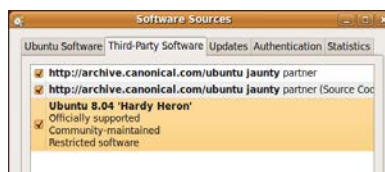
Ventana Software Sources

En la siguiente ventana que aparece se coloca *deb http://ubuntu.mirror.cambrium.nl/ubuntu/ hardy main universe multiverse* lo cual permitirá agregar la fuente que contiene el repositorio gcc-3.3 y seleccionamos la opción *Add Source*:



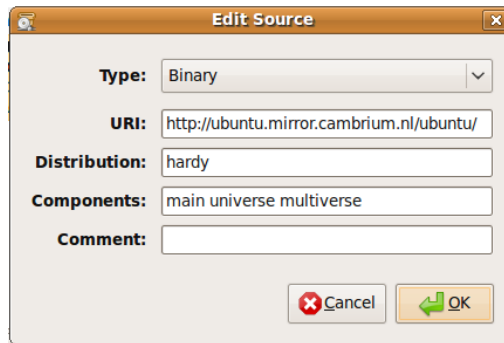
Agregar una fuente de descarga

Si el proceso de agregar una fuente se realizó correctamente en la ventana **Software Sources** se desplegará el nombre de la fuente y la información de la misma como se muestra en la Figura:



Fuente Agregada correctamente

Si se desea verificar la información de la fuente que ha sido agregada seleccionar la fuente agregada y la opción **Edit**, con lo cual se abrirá una ventana donde se puede modificar el URL, la distribución y los componentes como se indica la figura:



Edición de una fuente

Una vez finalizado el proceso para agregar la fuente en el terminal escribir `sudo apt-get update` para actualizar los repositorios que contiene la fuente que se agregó:

```
cirad@ubuntu:~$ sudo apt-get update
```

Actualización de repositorios que contiene la fuente agregada

Mientras se realiza el proceso de actualización correctamente nos mostrará información:

```
cirad@ubuntu:~$ sudo apt-get update
Get:1 http://ubuntu.mirror.cambrium.nl hardy Release.gpg [1898]
Ign http://ubuntu.mirror.cambrium.nl hardy/main Translation-en_US
Ign http://ubuntu.mirror.cambrium.nl hardy/universe Translation-en_US
Ign http://archive.ubuntu.com jaunty Release.gpg
Ign http://archive.ubuntu.com jaunty/main Translation-en_US
Ign http://archive.ubuntu.com jaunty/restricted Translation-en_US
Hit http://archive.canonical.com jaunty Release.gpg
Ign http://archive.canonical.com jaunty/partner Translation-en_US
Ign http://ubuntu.mirror.cambrium.nl hardy/multiverse Translation-en_US
Get:2 http://ubuntu.mirror.cambrium.nl hardy Release [65.9kB]
Ign http://archive.ubuntu.com jaunty/universe Translation-en_US
Ign http://archive.ubuntu.com jaunty/multiverse Translation-en_US
Ign http://archive.ubuntu.com jaunty-updates Release.gpg
Ign http://archive.ubuntu.com jaunty-updates/main Translation-en_US
Ign http://archive.ubuntu.com jaunty-updates/restricted Translation-en_US
Ign http://archive.ubuntu.com jaunty-updates/universe Translation-en_US
Ign http://archive.ubuntu.com jaunty-updates/multiverse Translation-en_US
Ign http://archive.ubuntu.com jaunty-security Release.gpg
Hit http://archive.canonical.com jaunty Release
Ign http://archive.ubuntu.com jaunty-security/main Translation-en_US
Ign http://archive.ubuntu.com jaunty-security/restricted Translation-en_US
Ign http://archive.ubuntu.com jaunty-security/universe Translation-en_US
Hit http://archive.canonical.com jaunty/partner Packages
```

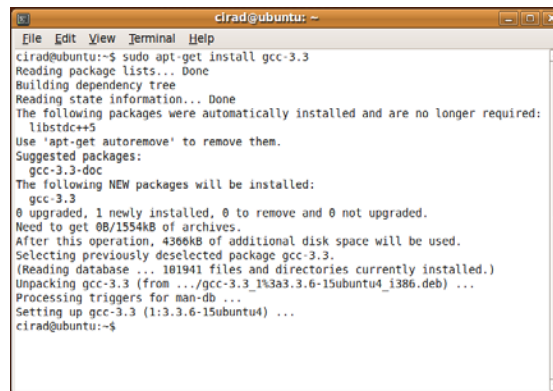
Información que se muestra durante la actualización

Una vez finalizado el proceso escribir ***sudo apt-get install gcc-3.3*** que permite la instalación de la versión deseada del gcc en este caso la 3.3:

```
cirad@ubuntu:~$ sudo apt-get install gcc-3.3
```

Instalación del gcc-3.3.

Mientras se realiza el proceso de instalación correctamente nos mostrará la siguiente información:



```
cirad@ubuntu: ~
File Edit View Terminal Help
cirad@ubuntu:~$ sudo apt-get install gcc-3.3
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
libstdc++5
Use 'apt-get autoremove' to remove them.
Suggested packages:
gcc-3.3-doc
The following NEW packages will be installed:
gcc-3.3
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0B/1554kB of archives.
After this operation, 4306kB of additional disk space will be used.
Selecting previously deselected package gcc-3.3.
(Reading database ... 101941 files and directories currently installed.)
Unpacking gcc-3.3 (from .../gcc-3.3_1:3a3.3.6-15ubuntu4_i386.deb) ...
Processing triggers for man-db ...
Setting up gcc-3.3 (1:3.3.6-15ubuntu4) ...
cirad@ubuntu:~$
```

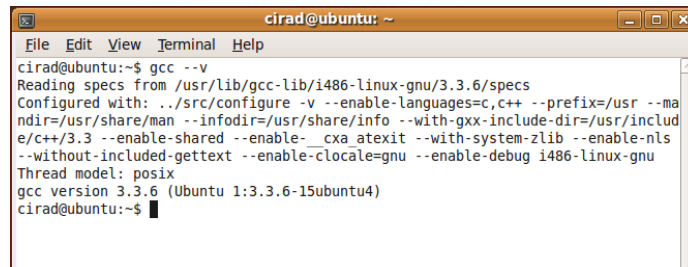
Proceso de instalación gcc.

En el caso que se desee mantener más de una versión del gcc, se debe dar prioridades a cada una, representando con un valor más alto al de mayor prioridad. Para este caso se requiere que el gcc-3.3 tenga una mayor prioridad por lo cual escribir en el terminal ***sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-3.3 100*** y luego para dar una menor prioridad a la otra versión instalada por defecto ***sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.3 50***:

```
cirad@ubuntu:~$ sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-3.3 100
cirad@ubuntu:~$ sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.3 50
```

Asignación de prioridades del gcc.

Para verificar la versión que tiene mayor prioridad o si el proceso de instalación se realizó correctamente escribir en el terminal `gcc -v`, así:

A terminal window titled 'cirad@ubuntu: ~' with a menu bar (File, Edit, View, Terminal, Help). The terminal output shows the command 'gcc --v' and its output: 'Reading specs from /usr/lib/gcc-lib/i486-linux-gnu/3.3.6/specs', 'Configured with: ../src/configure -v --enable-languages=c,c++ --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-gxx-include-dir=/usr/include/c++/3.3 --enable-shared --enable-__cxa_atexit --with-system-zlib --enable-nls --without-included-gettext --enable-clocale=gnu --enable-debug i486-linux-gnu', 'Thread model: posix', and 'gcc version 3.3.6 (Ubuntu 1:3.3.6-15ubuntu4)'. The prompt returns to 'cirad@ubuntu:~\$'.

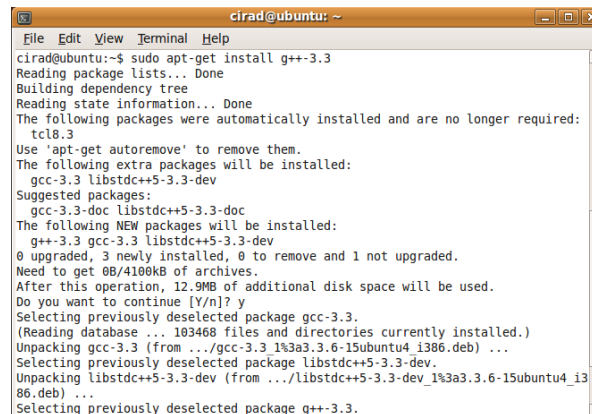
Verificación de la versión del gcc con mayor prioridad.

De igual manera se realiza la instalación del g++-3.3 escribiendo en el terminal sudo apt-get install g++-3.3:

```
cirad@ubuntu:~$ sudo apt-get install g++-3.3
```

Instalación del g++-3.3

Mientras se realiza el proceso de instalación correctamente nos mostrará la siguiente información:

A terminal window titled 'cirad@ubuntu: ~' with a menu bar (File, Edit, View, Terminal, Help). The terminal output shows the command 'sudo apt-get install g++-3.3' and its output: 'Reading package lists... Done', 'Building dependency tree', 'Reading state information... Done', 'The following packages were automatically installed and are no longer required: tcl8.3', 'Use \'apt-get autoremove\' to remove them.', 'The following extra packages will be installed: gcc-3.3 libstdc++5-3.3-dev', 'Suggested packages: gcc-3.3-doc libstdc++5-3.3-doc', 'The following NEW packages will be installed: g++-3.3 gcc-3.3 libstdc++5-3.3-dev', '0 upgraded, 3 newly installed, 0 to remove and 1 not upgraded.', 'Need to get 68/4100KB of archives.', 'After this operation, 12.9MB of additional disk space will be used.', 'Do you want to continue [Y/n]? y', 'Selecting previously deselected package gcc-3.3.', '(Reading database ... 183468 files and directories currently installed.)', 'Unpacking gcc-3.3 (from .../gcc-3.3_1%3a3.3.6-15ubuntu4_i386.deb) ...', 'Selecting previously deselected package libstdc++5-3.3-dev.', 'Unpacking libstdc++5-3.3-dev (from .../libstdc++5-3.3-dev_1%3a3.3.6-15ubuntu4_i386.deb) ...', 'Selecting previously deselected package g++-3.3.'

Proceso de instalación g++-3.3.

ANEXO 3

INSTALACIÓN DEL SOFTWARE D-ITG

Previo al proceso de instalación del simulador de tráfico DITG se deben instalar los repositorios necesarios para su correcto funcionamiento, se lo puede realizar en Ubuntu desde líneas de comando en el terminal o mediante paquetes synaptic. Los repositorios necesarios son:

Sun-java6-jre, g++, octave3.0

Ahora se procede a descargar los paquetes propios del simulador y la interfaz gráfica:

D-ITG-2.8.0-Beta e itggui-0911

Se los puede descargar desde la siguiente página web:

<http://www.grid.unina.it/software/ITG/>

Con todos los paquetes listos, se crea una carpeta dentro de root con cualquier nombre de preferencia DITG, en el cual se descomprime lo antes descargados con la línea de comando:

```
tar -zxvf NOMBRE.tgz
```

como se muestra en la figura.

```
root@ubuntu:/root/DITG# ls
D-ITG-2.8.0-rc1.tgz
root@ubuntu:/root/DITG# tar -zxvf D-ITG-2.8.0-rc1.tgz
```

Descompresión D-ITG 2.8.0.

A través del terminal se ingresa a la carpeta creada:

```
cd /root/DITG/src
```

Y se ejecuta el instalador mediante el comando “*make*”, indicado en la figura. Se compilan las herramientas, cuyos ejecutables son alojados en la carpeta *bin*, desde la cual se puede generar flujos, siguiendo la sintaxis explicada perfectamente en el manual que viene con la herramienta.

```
root@ubuntu:/root/DITG/D-ITG-2.8.0-rc1/src# make
g++ -DVERSION=\"2.8.0-rc1\" -DREVISION=\"457:458\" -DLINUX_OS -Wall -Wno-deprecated -fPIC -c -o common/thread.o common/thread.cpp
g++ -DVERSION=\"2.8.0-rc1\" -DREVISION=\"457:458\" -DLINUX_OS -Wall -Wno-deprecated -fPIC -c -o common/ITG.o common/ITG.cpp
g++ -DVERSION=\"2.8.0-rc1\" -DREVISION=\"457:458\" -DLINUX_OS -Wall -Wno-deprecated -fPIC -c -o common/timestamp.o common/timestamp.cpp
g++ -DVERSION=\"2.8.0-rc1\" -DREVISION=\"457:458\" -DLINUX_OS -Wall -Wno-deprecated -fPIC -c -o common/serial.o common/serial.cpp
```

Ejecución del comando “*make*”.

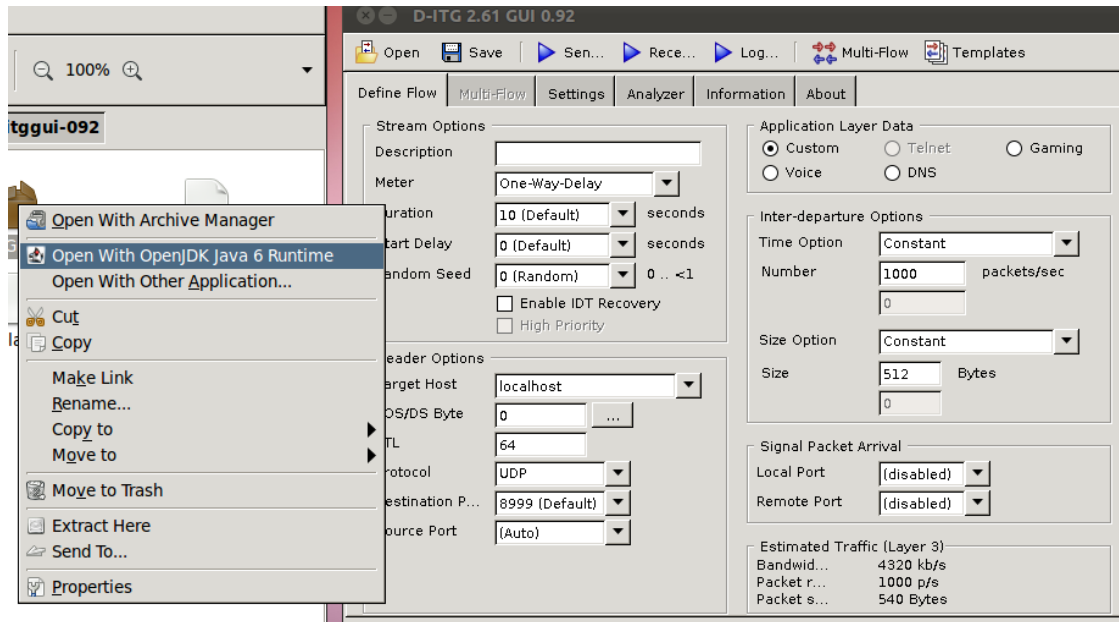
Se crea un directorio llamado *logs*, donde se guardaran los logs, cuando se haga uso del inyector, se la puede crear gráficamente, o mediante el comando:

```
mkdir /D-ITG-2.8.0-Beta/logs
```

Para acceder a la interfaz gráfica se ubica la carpeta donde se descomprimió *itgui-0911*, luego se escribe en la línea de comandos:

```
java -jar ITGGUI.jar
```

También se puede acceder haciendo click derecho sobre *ITGGUI.jar* mediante la opción abrir con Sun java 6 Runtime, desplegándose la ventana que se muestra a continuación:



Ingreso a la Interfaz Gráfica.

Se debe tomar en cuenta que al intentar graficar los archivos creados por el simulador, hay que acceder a la carpeta donde se encuentra el archivo ITGplot y escribir en la línea de comandos (`chmod 777`), esto nos dará los permisos necesarios para poder crear nuestras gráficas.

```

CHANGES.TXT  KNOWNISSUES.TXT  README.TXT      TODO.txt
ITGGUI.jar    LICENSE          templates.lst
root@ubuntu:~/Desktop/tesis/instaladores/itggui-092# chmod 777 ITGGUI.jar
root@ubuntu:~/Desktop/tesis/instaladores/itggui-092# ls
CHANGES.TXT  KNOWNISSUES.TXT  README.TXT      TODO.txt
ITGGUI.jar    LICENSE          templates.lst

```

Ejecución del comando “`chmod 777`”

ANEXO 4

INSTALACIÓN Y EJECUCIÓN DE *GNUplot*

Sobre un terminal de Ubuntu se escribe: `apt-get install gnuplot`, como se muestra en la figura.

```
root@ubuntu:~# apt-get install gnuplot
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Instalación Gnuplot

La descarga es de aproximadamente 10.5 Mb y la instalación ocupará casi 30 Mb de espacio. Si se desea descargar el código fuente o se tiene algún error en la instalación visitar la página oficial:

<http://www.gnuplot.info/documentation>

Para la ejecución se digita en el terminal “gnuplot+ENTER”, luego se muestra la versión, sistema y créditos.

```
root@ubuntu:~# gnuplot

G N U P L O T
Version 4.4 patchlevel 2
last modified Wed Sep 22 12:10:34 PDT 2010
System: Linux 2.6.38-13-generic

Copyright (C) 1986-1993, 1998, 2004, 2007-2010
Thomas Williams, Colin Kelley and many others

gnuplot home:      http://www.gnuplot.info
faq, bugs, etc:   type "help seeking-assistance"
immediate help:   type "help"
plot window:      hit 'h'
```

Características Gnuplot.

Para acceder a la ayuda de Gnuplot se usa el comando `gnuplot> help plot`

Para graficar nuestros archivos `.dat` se digita como se muestra a continuación:

```
gnuplot> plot "recibidos.dat"
```

Donde `recibidos.dat`, se refiere al archivo donde están los datos que deseamos graficar, si ingresamos solo el nombre con la extensión que tiene, buscare el archivo dentro de la carpeta donde este actualmente ejecutándose Gnuplot, si queremos especificar otra ruta, ingresamos la ruta completa del archivo.

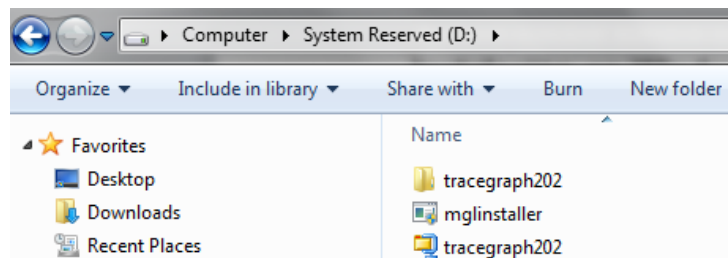
ANEXO 5

INSTALACION DE TRACE GRAPH

Su instalación se la puede realizar bajo Linux, como en Windows. Por otro lado, es posible que en la instalación, de Linux sobretodo, existan problemas de no reconocimiento de librerías Matlab, las cuales deberán ser instaladas y direccionadas en la Path de Linux antes de comenzar a utilizar Trace Graph, es por eso que se realizó la instalación en Windows la misma que se muestra a continuación.

Instalación en Windows.

- 1) Descargar de la página www.tracegraph.com/download.html el tracegraph , el mglinstaller (instalador de librerías matlab) y el traceconverter. No se debe olvidar seleccionar la opción Windows.
- 2) Descomprimir el archivo tracegraph202.zip en algún directorio (D:).



Descompresión de tracegraph202.zip

- 3) Copiar mglinstaller.exe dentro de la carpeta tracegraph202 la cual fue previamente comprimida y ejecutar mglinstaller.exe.

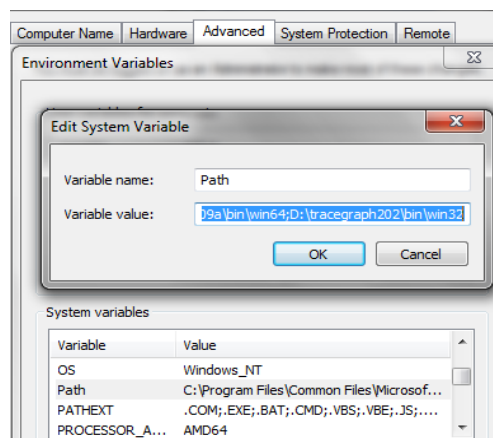

```

D:\tracegraph202\mglinstaller.exe
MWUnZipSFX 5.41 of 16 April 2000, by Info-ZIP. Modified by The MathWorks, Inc.
Send bug reports to support@mathworks.com.
  inflating: installer.exe
  inflating: mglarchive.exe

```

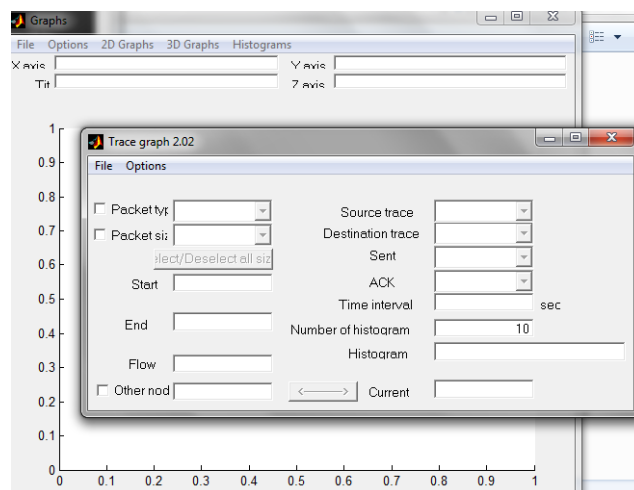
Ejecución de mglinstaller.exe.

4) Agregar al Path de Windows (Botón derecho del Mouse en Mi PC/ propiedades/opciones avanzadas/ variables de entorno y seleccione Path en variables del sistema) la dirección: \lugar donde descomprimió tracegraph202\bin\win32.



Ubicación del path de Trace Graph en Windows.

5) Ejecutar trgrap.exe que de encuentra en la carpeta tracegraph202



Ventanas de Trace Graph.

ANEXO 6

HOJAS TÉCNICAS PARA LA ANTENA HG2424G

2.4 GHz 24 dBi High Performance Reflector Grid Wireless LAN Antenna

Model: HG2424G

Applications and Features

- Applications:**
- 2.4 GHz ISM Band
 - IEEE 802.11b and 802.11g Wireless LAN
 - WFI Systems
 - Long-range Directional Applications
 - Point to Point Systems
 - Point to Multi-point Systems
 - Wireless Bridges
 - Backhaul Applications
- Features:**
- Superior performance
 - Cast aluminum construction
 - UV stable light gray powder coat finish
 - All weather operation
 - 8° beam-width
 - 12 inch coax lead
 - Easy to assemble



Description

Superior Performance

The HyperGain® HG2424G High-Performance Reflector Grid Wi-Fi Antenna provides 24 dBi gain with an 8 degree beam-width for long-range highly directional applications. Applications include point to point systems, point to multi-point and wireless bridges in the 2.4GHz ISM band as well as IEEE 802.11b and 802.11g wireless LAN systems. It can be installed for either vertical or horizontal polarization.

Specifications

Electrical Specifications

Frequency	2400-2500 MHz
Gain	24 dBi
-3 dBi Beam Width	8 degrees
Cross Polarization Rejection	26 dBi
Front to Back Ratio	24 dB
Sidelobe	-20dB Max
Impedance	50 Ohm
Max. Input Power	50 Watts
VSWR	< 1.5:1 avg.

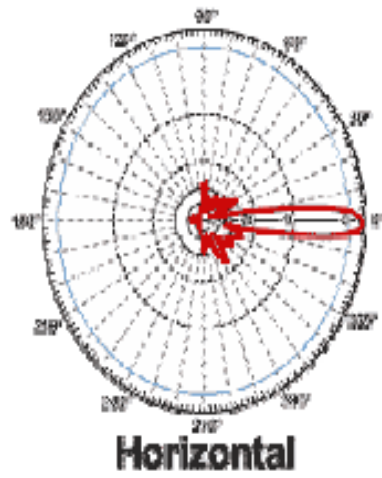
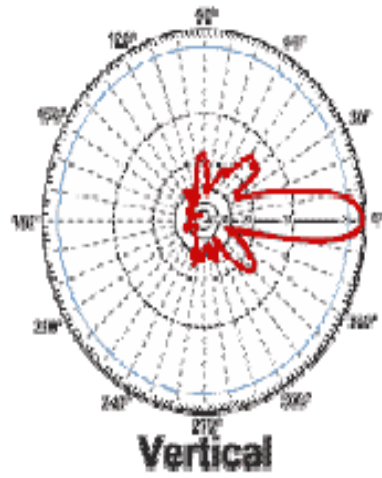
Mechanical Specifications

Weight	4.8 lbs. (2.18 kg)
Grid Dimensions	39.5 in (100 cm) x 23.5 in (60 cm)
Mounting	2 in. (50.8 mm) diameter mast max.
Elevation Angle	0 to +10 degrees
Operating Temperature	-40° C to to 85° C (-40° F to 185° F)

Wind Loading Data

Wind Speed (MPH)	Loading (2.1 sq. ft.)
40	16.4 lb.
60	36.3 lb.
80	63.8 lb.
100	97.0 lb.
120	147.0 lb.
140	199.5 lb.

RF Antenna Gain Patterns



ANEXO 7

CÓDIGO EN *ns-2* PARA EL ENLACE SALINAS – ÁNIMAS

```
#####
#                               ns-2 (TeNS)
#####

set val(chan)      Channel/Channel_802_11    ;# Tipo de canal
set val(prop)      Propagation/Shadowing     ;# Modelo de propagación
set val(netif)     Phy/WirelessPhy/Wireless_802_11_Phy ;# Tipo de interfaz de red
set val(mac)       Mac/802_11                ;# Tipo de MAC
set val(ifq)       Queue/DropTail/PriQueue   ;# Tipo de interfaz de cola
set val(ll)        LL                        ;# Capa de enlace
set val(ant)       Antenna/DirAntenna       ;# Modelo de antena
set val(ifqlen)    50                        ;# max pACKet in ifq
set val(nn)        2                         ;# Número de nodos

#También se puede utilizar el protocolo AODV
set val(rp)        WLSTATIC                  ;# Protocolo de enrutamiento

set val(ni)        1                        ;# Número de interfaces

#Nueva adicción para el tipo de modulación
set opt(mod)       Modulation/BPSK

# =====
# Programa principal
# =====

#
# Inialización de variables globales
#

set ns_            [new Simulator]
set tracefd       [open simple1.tr w]
#set par          [open param.tr w]
$ns_ trace-all $tracefd

#$ns_ use-newtrace

# *** Inialización del Animador de Red ***
set namtrace      [open sim12.nam w]
$ns_ namtrace-all-wireless $namtrace 10000 10000
```

```
# Seteo de la topografía del objeto
set topo [new Topography]
$topo load_flatgrid 10000 10000
```

```
# Crea un número específico de nodos móviles[$val(nn)] y los adjunta al canal
# Son creados los nodos : node(0) y node(1)
```

```
# Configuración de nodo
```

```
  $ns_ node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -channel [new $val(chan)] \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace ON \
    -numif $val(ni) \
```

```
create-god $val(nn)
```

```
proc create_node { x y z } {
```

```
  global ns_
  Phy/WirelessPhy set RXThresh_ 1.6388e-15
  Phy/WirelessPhy set CStresh_ 1.6388e-15
  Phy/WirelessPhy set CPhresh_ 10.0
  Phy/WirelessPhy set L_ 1.0
  Phy/WirelessPhy set freq_ 2.4e+9
  Phy/WirelessPhy set bandwidth_ 5.5mb
  Mac/802_11 set CWMin_ 31
  Mac/802_11 set CWMax_ 1023
  Mac/802_11 MAC_RTSThreshold 2000
  Mac/802_11 MAC_FragmentationThreshold 2500
  Mac/802_11 DSSS_AirPropagationTime 0.000003
  Mac/802_11 set dataRate_ 5.5mb
  Mac/802_11 set basicRate_ 5.5mb
```

```
  # La siguiente sentencia es usada para activar      modulación BPSK.
  # Sino se la desea usar se la setea a cero.
  # Por defecto se encuentra seteado a cero.
  Phy/WirelessPhy set modulationscheme_ 1
```

```
  set newnode [$ns_ node]
  $newnode random-motion 0
  $newnode set X_ $x
  $newnode set Y_ $y
```

```
    $newnode set Z_ $z

    return $newnode
}

proc create_cbr_connection { from to startTime interval pACKetSize } {
    global ns_
    set udp0 [new Agent/UDP]
    set src [new Application/Traffic/CBR]
    $udp0 set pACKetSize_ 1500
    $src set pACKetSize_ 1500
    $src set interval_ 0.005

    set sink [new Agent/Null]

    $ns_ attach-agent $from $udp0
    $src attach-agent $udp0
    $ns_ attach-agent $to $sink

    $ns_ connect $udp0 $sink
    $ns_ at $startTime "$src start"
    return $udp0
}

proc create_tcp_connection { from to startTime } {
    global ns_ par
    set tcp [new Agent/TCP]
    $tcp set pACKetSize_ 1500
    $tcp set class_ 2
    set sink [new Agent/TCPSink]
    $ns_ attach-agent $from $tcp
    $ns_ attach-agent $to $sink
    $ns_ connect $tcp $sink
    set ftp [new Application/FTP]
    $ftp set pACKetSize_ 1500
    $ftp attach-agent $tcp
    $ns_ at $startTime "$ftp start"
    $tcp attach $par

    $tcp trace cwnd_
    $tcp trace maxseq_
    $tcp trace rtt_
    $tcp trace dupACKs_
    $tcp trace ACK_
    $tcp trace ndatabytes_
    $tcp trace ndatapACK_
    $tcp trace nrexmit_
    $tcp trace nrexmitpACK_
    return $tcp
}
```

```
#####
```

```
$ns_ node-config -numif 1
set node_(0) [create_node 10 10 0]

[$node_(0) set netif_(0)] set channel_number_ 1

[$node_(0) set netif_(0)] set Pt_ 0.07943

set a [new Antenna/DirAntenna]
$a setType 1
$a setAngle 90
```

```
[$node_(0) set netif_(0)] dir-antenna $a
```

```
#####
```

```
$ns_ node-config -numif 1
set node_(1) [create_node 10 66710 0]
[$node_(1) set netif_(0)] set Pt_ 0.07943
[$node_(1) set netif_(0)] set channel_number_ 1
set a [new Antenna/DirAntenna]
$a setType 1
$a setAngle 265
```

```
[$node_(1) set netif_(0)] dir-antenna $a
```

```
#####
```

```
#Seteo manual de la tabla de enrutamiento para "wlstatic"
#La sintaxis es la siguiente
#[$node_(0) set ragent_] addstaticroute <número de saltos> <siguiente salto> <nodo de destino>
<interfaz usada>
[$node_(0) set ragent_] addstaticroute 1 1 1 0
[$node_(1) set ragent_] addstaticroute 1 1 0 0
```

```
#####
```

```
set tcp0 [create_cbr_connection $node_(0) $node_(1) 1.0 0.005 1500]
```

```
#
# Finalización de la simulación de los nodos
#
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ at 110.0 "$node_($i) reset";
}
$ns_ at 110.0 "stop"
$ns_ at 110.01 "puts \"NS EXITING...\" ; $ns_ halt"
proc stop {} {
```



```
exec nam -a sim12.nam &
global ns_ tracefd ;#par
$ns_ flush-trace
close $tracefd
}

puts "Starting Simulation..."
$ns_ run
```

ANEXO 8

CÓDIGO EN *ns-2* PARA EL ENLACE SALINAS – CABUYAS

```
#####
#                               ns-2 (TeNS)
#####

set val(chan)      Channel/Channel_802_11    ;# Tipo de canal
set val(prop)      Propagation/TwoRayGround  ;# Modelo de propagación
set val(netif)     Phy/WirelessPhy/Wireless_802_11_Phy ;# Tipo de interfaz de red
set val(mac)       Mac/802_11                ;# Tipo de MAC
set val(ifq)       Queue/DropTail/PriQueue   ;# Tipo de interfaz de cola
set val(ll)        LL                        ;# Capa de enlace
set val(ant)       Antenna/DirAntenna        ;# Modelo de antena
set val(ifqlen)    50                        ;# max pACKet in ifq
set val(nn)        2                        ;# Número de nodos

#También se puede utilizar el protocolo AODV
set val(rp)        WLSTATIC                  ;# Protocolo de enrutamiento

set val(ni)        1                        ;# Número de interfaces

#Nueva adicción para el tipo de modulación
set opt(mod)       Modulation/BPSK

# =====
# Programa principal
# =====

# Inialización de variables globales
#

set ns_            [new Simulator]
set tracefd        [open simple1.tr w]
#set par           [open param.tr w]
$ns_ trace-all $tracefd

#$ns_ use-newtrace

# *** Inialización del Animador de Red ***
set namtrace       [open sim12.nam w]
$ns_ namtrace-all-wireless $namtrace 10000 10000

# Seteo de la topografía del objeto
set topo           [new Topography]
```

```
$topo load_flatgrid 10000 10000
```

```
# Crea un número específico de nodos móviles[$val(nn)] y los adjunta al canal
```

```
# Son creados los nodos : node(0) y node(1)
```

```
# Configuración de nodo
```

```
  $ns_ node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -channel [new $val(chan)] \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace ON \
    -numif $val(ni) \
```

```
create-god $val(nn)
```

```
proc create_node { x y z } {
```

```
  global ns_
  Phy/WirelessPhy set RXThresh_ 2.07454e-16
  Phy/WirelessPhy set CStresh_ 2.07454e-16
  Phy/WirelessPhy set CPhresh_ 10.0
  Phy/WirelessPhy set L_ 1.0
  Phy/WirelessPhy set freq_ 2.4e+9
  Phy/WirelessPhy set bandwidth_ 5.5mb
  Mac/802_11 set CWMin_ 31
  Mac/802_11 set CWMax_ 1023
  Mac/802_11 MAC_RTSThreshold 2000
  Mac/802_11 MAC_FragmentationThreshold 2500
  Mac/802_11 DSSS_AirPropagationTime 0.000003
  Mac/802_11 set dataRate_ 5.5mb
  Mac/802_11 set basicRate_ 5.5mb
```

```
  # La siguiente sentencia es usada para activar la modulación BPSK.
```

```
  # Sino se la desea usar se la setea a cero.
```

```
  # Por defecto se encuentra seteado a cero.
```

```
  Phy/WirelessPhy set modulationscheme_ 1
```

```
  set newnode [$ns_ node]
```

```
  $newnode random-motion 0
```

```
  $newnode set X_ $x
```

```
  $newnode set Y_ $y
```

```
  $newnode set Z_ $z
```

```

        return $newnode
    }

proc create_cbr_connection { from to startTime interval pACKetSize } {
    global ns_
    set udp0 [new Agent/UDP]
    set src [new Application/Traffic/CBR]
    $udp0 set pACKetSize_ 1500
    $src set pACKetSize_ 1500
    $src set interval_ 0.005

    set sink [new Agent/Null]

    $ns_ attach-agent $from $udp0
    $src attach-agent $udp0
    $ns_ attach-agent $to $sink

    $ns_ connect $udp0 $sink
    $ns_ at $startTime "$src start"
    return $udp0
}

proc create_tcp_connection { from to startTime } {
    global ns_ par
    set tcp [new Agent/TCP]
    $tcp set pACKetSize_ 1500
    $tcp set class_ 2
    set sink [new Agent/TCPSink]
    $ns_ attach-agent $from $tcp
    $ns_ attach-agent $to $sink
    $ns_ connect $tcp $sink
    set ftp [new Application/FTP]
    $ftp set pACKetSize_ 1500
    $ftp attach-agent $tcp
    $ns_ at $startTime "$ftp start"
    $tcp attach $par

    $tcp trace cwnd_
    $tcp trace maxseq_
    $tcp trace rtt_
    $tcp trace dupACKs_
    $tcp trace ACK_
    $tcp trace ndatabytes_
    $tcp trace ndatapACK_
    $tcp trace nrexmit_
    $tcp trace nrexmitpACK_
    return $tcp
}

```

```
#####
```

```

$ns_ node-config -numif 1
set node_(0) [create_node 10 10 0]

[$node_(0) set netif_(0)] set channel_number_ 1

[$node_(0) set netif_(0)] set Pt_ 0.07943

set a [new Antenna/DirAntenna]
$a setType 1
$a setAngle 90

[$node_(0) set netif_(0)] dir-antenna $a

#####

$ns_ node-config -numif 1
set node_(1) [create_node 10 77510 0]
[$node_(1) set netif_(0)] set Pt_ 0.07943
[$node_(1) set netif_(0)] set channel_number_ 1
set a [new Antenna/DirAntenna]
$a setType 1
$a setAngle 265

[$node_(1) set netif_(0)] dir-antenna $a

#####

#Seteo manual de la tabla de enrutamiento para "wlstatic"
#La sintaxis es la siguiente
#[$node_(0) set ragent_] addstaticroute <número de saltos> <siguiente salto> <nodo de destino>
<interfaz usada>
[$node_(0) set ragent_] addstaticroute 1 1 1 0
[$node_(1) set ragent_] addstaticroute 1 1 0 0

#####

set tcp0 [create_cbr_connection $node_(0) $node_(1) 1.0 0.005 1500]

#
# Finalización de la simulación de los nodos
#
for {set i 0} {$i < $val(nn) } {incr i} {
    $ns_ at 110.0 "$node_($i) reset";
}
$ns_ at 110.0 "stop"
$ns_ at 110.01 "puts \"NS EXITING...\" ; $ns_ halt"
proc stop {} {

    exec nam -a sim12.nam &
    global ns_ tracefd ;#par

```

```
    $ns_ flush-trace
    close $tracefd
}

puts "Starting Simulation..."
$ns_ run
```

REFERENCIAS BIBLIOGRAFICAS

- [1] Elías, José, Nuevo Récord De Distancia WiFi,
http://elias.com/index.cfm?post_id=3242, publicación: 21 de junio del 2007, consulta:
enero 2012.
- [2] Tamayo, Viviana; Barrionuevo, Evelin, Análisis Del Desempeño De Una Red Con
Tecnología Wifi Para Largas Distancias En Un Ambiente Rural De La Región Sierra,
Escuela Politécnica del Ejército, 2011.
- [3] Secretaría Nacional de Planificación y Desarrollo, Semplades, Consejo Nacional
de Planificación 2009, Plan Nacional para el Buen Vivir 2009-2013: Construyendo un
Estado Plurinacional e Intercultural, segunda edición, ISBN: 978-9978-92-794-6.
- [4] Stephens, Adrian, IEEE 802.11TM WIRELESS LOCAL AREA NETWORKS:
The Working Group for WLAN Standards, <http://www.ieee802.org/11/> , disponible en
junio del 2012.
- [5] Creative Commons, Adslfaqs, Historia Y Actualidad Del WiFi, publicación: 23
marzo 2012, consulta: mayo del 2012.
- [6] Valadez, Fernando, Tipos de estándares WiFi, publicación: 21 de enero del 2011,
consulta: mayo 2012.
- [7] Miguel Ponce, Enrique; Molina Enrique. Redes Inalámbricas: IEEE 802.11,
<http://www.canal-ayuda.org/a-informatica/inalambrica.htm>, junio 2012.

- [8] Armentis, Tecnologías de propagación, <https://sites.google.com/site/amentis81/wifi-propagacion>, disponible en junio del 2012.
- [9] Penton media, Building future networks with MIMO and OFDM, http://connectedplanetonline.com/wireless/technology/mimo_ofdm_091905/ 2005, disponible en junio 2012. .
- [10] Ing, Jara, Pablo; Ing Nazar, Patricia. Estándar IEEE 802.11X de las Wlan, Editorial de la Universidad Tecnológica Nacional Argentina.
- [11] ACK-*timeout* y sus efectos en enlaces de larga distancia. http://www.air-stream.org/ACK_Timeouts, consulta: junio del 2012.
- [12] Comunidad de redes Inalámbricas, ACK *Timeouts* and the effects on distance links, publicación: 17 de Julio del 2011, consulta: junio del 2012.
- [13] Hidalgo, María, Asociación para el progreso de las comunicaciones, <http://www.apc.org/es/news/ecuador-llegar-donde-los-cables-y-los-intereses-co>, publicado en septiembre del 2009, disponible en julio del 2012.
- [14] Chebrolu, Kameswari; Raman, Bhaskaran; Sen, Sayandeep, Long-Distance 802.11b Links: Performance Measurements and Experience, IEEE 2005, 84 páginas.
- [15] Chebrolu, Kameswari; Raman, Bhaskaran, Design and Evaluation of a new MAC Protocol for Long-Distance 802.11 Mesh Networks, IEEE 2005, 14 páginas.

[16] Simó, Francisco, Modelado y optimización de IEEE 802.11 para su aplicación en el despliegue de redes extensas en zonas rurales aisladas de países en desarrollo, Universidad Politécnica de Madrid, Enero del 2007.

[17] Ludeña, Patricia, Estudio de aplicabilidad del estándar 802.11n para redes de larga distancia para entornos rurales en América Latina, Universidad Rey Juan Carlos de Madrid, 2011.

[18] Agila, Ricardo; Sánchez Jimmy, Análisis de Estándares 802.11e Y 802.11n para Largas Distancias, Universidad Técnica Particular de Loja, Ecuador, 2009.

[19] Grupo de Telecomunicaciones Rurales Pontificia Universidad Católica del Perú , WiFi base on long distance, <http://gtr.telecom.pucp.edu.pe/system/files/WiLD.pdf>, disponible en junio del 2012, primera edición, agosto del 2009, disponible en junio del 2012.

[20] Adi, Engineering, Inc, Pronghorn SBC Hardware Manual, 21 páginas. http://site.microcom.us/PronghornSBC_Hardware_Manual.pdf , disponible en junio del 2012

[21] *Soekris engineering. Inc., net4826 Product Range, <http://soekris.com/products/net4826.html> , disponible en junio del 2012.*

[22] Punkytse, Voyage Linux 0.6.5 released <http://linux.voyage.hk/content/voyage-linux-065-released>, publicado el 22 de febrero del 2010, disponible en junio del 2012.

[23] Open Course Ware, Universidad Politécnica de Madrid, Radio Mobile, <http://ocw.upm.es/teoria-de-la-senal-y-comunicaciones->

1/radiocomunicacion/contenidos/utilidades/RadioMobile/leeme/ , disponible en julio del 2012.

[24] <http://usuarios.multimania.es/sirenet2/sirenet2.html>, disponible en junio del 2012.

[25] Escuela Politécnica Nacional, Network Simulator, <http://dspace.epn.edu.ec/bitstream/15000/8650/3/T10148CAP3.pdf>, disponible en junio del 2012

[26] ns-3 Reference Manual, <http://www.nsnam.org/docs/release/3.9/manual.pdf> , publicado el 20 de agosto del 2010, disponible en junio del 2012.

[27] Borrero , Diego , Simulación del algoritmo *HandOff* vertical basado en preferencias de usuario, mediante la herramienta NCTUns, <http://bdigital.uao.edu.co/bitstream/10614/332/1/T0003563.pdf> , Cali 2009, disponible en junio del 2012.

[28] Departamento de Computación, ciencia e ingeniería, Instituto tecnológico de India, The Enhanced Network Simulator (Release Version 1.2), <http://www.cse.iitk.ac.in/users/braman/TeNS/>, disponible en junio del 2012.

[29] Camacho, Fabián, Universidad tecnológica Metropolitana, Trace Graph y sus numerosas utilidades, <http://es.scribd.com/doc/47268207/TRACEGRAPH>, pagina 51, disponible en junio del 2012.

[30] Naranjo, Sofía, Escuela Politécnica Nacional, Análisis del comportamiento del atecnologia Wimax y Wimax Mobile usando el simulador *ns-2*, <http://bibdigital.epn.edu.ec/bitstream/15000/4268/1/CD-3898.pdf>, disponible en junio del 2012.

[31] Abzug, Morty, Iperf 1.7.0 <http://iperf.updatestar.com/es>, disponible en junio del 2012.

[32] Velásquez, Karima; Gamess, Erick, Universidad central de Venezuela, Analisis comparativo de herramientas de evaluación en redes de Computadores, ISSN 1316-6239, Noviembre de 2009, página 4.

[33] Laine, Juha; Saaristo, Sampo; Prior, Rui, Rude & Crude, <http://www.atm.tut.fi/rude>, disponible en julio del 2012.

[34] García, Salvador, Radio Mobile para Windows, <http://es.scribd.com/doc/60812917/7/PARAMETROS-ESPECIFICOS-PARA-EL-MODELO-DE-LONGLEY-RICE> , publicado en octubre del 2010, disponible en junio del 2012., pagina 9.

[35] Apolo, Diana, Escuela Politécnica del Ejército, Diseño de la red de telecomunicaciones de la Espe para supervisión volcánica, enero del 2011. 312 páginas.

[36] Ubiquiti Netwoks, [_http://www.ubnt.com/powerstation](http://www.ubnt.com/powerstation), disponible en junio del 2012.