



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

**“EXTENSIÓN DE CAPACIDADES DE ANÁLISIS DE DATOS EN ROUTERS
CON OPENWRT”**

INFORME DE MATERIA DE GRADUACIÓN

Previa a la obtención del Título de:

**INGENIERO EN COMPUTACIÓN ESPECIALIZACIÓN SISTEMAS
TECNOLÓGICOS**

INGENIERO EN TELEMÁTICA

Presentado por:

VICTOR HUGO PEZO ORTIZ
AMBAR MICHELLE PIZA HERNÁNDEZ

GUAYAQUIL – ECUADOR

Año 2013

AGRADECIMIENTO

A Dios por haberme proporcionado entendimiento, inteligencia y sabiduría; a mis padres y demás familiares que aportaron en todos los aspectos de mi crecimiento, a mis profesores que han contribuido a mi desarrollo profesional y al CVR por los aportes a nuestro trabajo de investigación.

Ambar Piza Hernández

A Dios por guiarme en cada momento, a mi esposa por siempre brindarme su apoyo incondicional, a mis padres y a mi hermana, por su cariño, su fuerza y su aliento a lo largo de este camino.

Víctor Pezo Ortiz

DEDICATORIA

A mis hermanos para quienes deseo ser siempre un ejemplo y a todas las personas que estuvieron a mi lado impulsándome para alcanzar mis metas.

Ambar Piza Hernández

A mi familia por las que lucho cada día de mi vida ya que son la fuerza que me lleva a conseguir mis objetivos.

Víctor Pezo Ortiz

TRIBUNAL DE SUSTENTACIÓN

Ph. D. Daniel Ochoa

PROFESOR DE LA MATERIA DE GRADUACIÓN

Ing. Patricia Chávez MSEE

PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD

DECLARACIÓN EXPRESA

“La responsabilidad por los hechos, ideas y doctrinas expuestas en este trabajo, nos corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral”

(Reglamento de exámenes y títulos profesionales de la ESPOL)

Víctor Hugo Pezo Ortiz

Ambar Michelle Piza Hernández

RESUMEN

El tema expuesto en este documento describe como implementar un sistema de monitoreo de redes existentes como una forma de sobrellevar los problemas de las redes inalámbricas Wi-Fi ocasionados por el medio, al momento de decidir a qué red es mejor conectarse en base al porcentaje de señal recibido de las mismas.

En el primer capítulo se detallan las directrices del proyecto tales como: objetivos generales, objetivos específicos y la metodología que se va a emplear para el desarrollo del mismo. En el segundo capítulo se describen y desarrollan los fundamentos teóricos que son la base para la elaboración del proyecto de investigación.

La implementación y el funcionamiento de nuestro sistema son expuestos en el capítulo 3 incluidos las características de hardware de los equipos utilizados. Las pruebas y resultados de nuestra implementación son explicados en el capítulo 4

ÍNDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA	III
TRIBUNAL DE SUSTENTACION	IV
DECLARACIÓN EXPRESA	V
RESUMEN	VI
INDICE GENERAL	VII
ABREVIATURAS	X
INDICE DE FIGURAS	XI
INDICE DE TABLAS	XIII
INTRODUCCIÓN	XIV
CAPITULO 1. DESCRIPCIÓN GENERAL DEL PROYECTO	1
1.1. Justificación	2
1.2. Objetivos	4
1.2.1. Objetivos Generales	4
1.2.2. Objetivos Especificos	4
1.3 Metodología	5
CAPITULO 2. FUNDAMENTOS TEÓRICOS	6
2.1. Redes Inalámbricas Wi-Fi	6
2.1.1. Evolución de las Redes Wi-Fi	7
2.1.2. Fenómenos Físicos Intrínsecos a las Redes Wi-Fi	9
2.2. Métricas de Calidad de Servicio (QOS)	19
2.3. Sistemas Embebidos	21

2.3.1.	Sistema Embebido vs Computadora de uso general	26
2.3.2.	Enrutadores Inalámbricos	27
2.4.	El Firmware	29
2.5.	Openwrt	32
2.6.	Sistemas de Tiempo Real	35
2.6.1.	Soft Real Time	37
2.6.2.	Hard Real Time	38
CAPITULO 3.	IMPLEMENTACIÓN	39
3.1.	Descripción del Hardware	39
3.2.	Descripción del Software	41
3.3.	Instalación del Firmware Openwrt	42
3.4.	Gestión de Paquetes en OpenWrt	45
3.4.1.	OPKG	45
3.4.2.	Instalación de Paquetes	45
3.4.3.	Paquetes de instalación necesarios obtener de los datos	46
3.5.	Compilación Cruzada	47
3.6.	Programa Principal	48
3.6.1.	Diseño del código del programa	49
3.6.2.	Implementación de técnicas de programación en tiempo real	52
3.6.3.	Ejecución del paquete de instalación	54
CAPÍTULO 4.	PRUEBAS Y RESULTADOS	56
4.1.	Monitoreo de Redes Existentes	57
4.2.	Recursos consumidos por el enrutador al ejecutar el script	60
4.3.	Obtención del mapa de porcentaje de señal	62

CONCLUSIONES Y RECOMENDACIONES 69

ANEXOS

BIBLIOGRAFÍA

ABREVIATURAS

AP:	Access Point o Punto de acceso
ASIC:	Circuito Integrado para Aplicaciones Específicas
GP IO:	Entrada/Salida de Propósito General
HW-N:	Virtualización de Hardware
ISM:	Industrial, scientific and Medical
JTAG:	Puerto de Acceso de Acción de Pruebas
PLD:	Dispositivo Lógico Programable
RF:	Radio Frequency o Radio Frecuencia
RT:	Real Time o Tiempo Real
RTC:	Reloj en Tiempo Real
UART:	Transmisor-Receptor Asíncrono Universal
WDS:	Sistema de Distribución Inalámbrico
WECA:	Compatibilidad Inalámbrica-Ethernet
WLAN:	Red de Área Local Inalámbrica
Wi-Fi:	Fidelidad Inalámbrica

ÍNDICE DE FIGURAS

Figura 1.1	Ejemplo de uso en un estacionamiento	3
Figura 2.1	Dispositivos Wi-Fi	7
Figura 2.2	Disminución en la señal según el obstáculo traspasado	11
Figura 2.3	Ángulo de la onda incidente y la onda reflejada	13
Figura 2.4	Uso de la reflexión de las ondas	14
Figura 2.5	Desvío de la onda al pasar de un medio a otro	15
Figura 2.6	Propagación de la señal a través de un orificio	16
Figura 2.7	Propagación en multitrayectoria	18
Figura 2.8	Relación Señal a Ruido (SNR)	21
Figura 2.9	Interior de enrutador WRT160NL	22
Figura 2.10	Arquitectura de un sistema embebido	23
Figura 2.11	Arquitectura interna del enrutador	28
Figura 2.12	Arquitectura de una computadora	30
Figura 2.13	Tiempo de ejecución de una tarea	36
Figura 3.1	Enrutador Linksys WRT54GL	40
Figura 3.2	Batería recargable Li-on Y SN-05480	41
Figura 3.3	Ejemplo de grafico realizado en GNUPLOT	42
Figura 3.4	Diagrama de interacción del sistema	49
Figura 3.5	Diagrama de flujo del script de monitoreo	52
Figura 3.6	Diagrama ejecución del script en real time	53
Figura 4.1	Disposición de equipos para tomar los datos	57

Figura 4.2	Gráfico porcentaje de la señal promedio vs tiempo	59
Figura 4.3	Gráfico carga del procesador del enrutador antes de la ejecución del script	61
Figura 4.4	Gráfico carga del procesador del enrutador durante la ejecución del script	61
Figura 4.5	Esquema de ubicación de los enrutadores dentro de la locación	63
Figura 4.6	Esquema de coordenadas asignadas para generar el mapa	64
Figura 4.7	Mapa de porcentaje de señal Enrutador 1	67
Figura 4.8	Mapa de porcentaje de señal Enrutador 2	67
Figura 4.9	Mapa de porcentaje de señal Enrutador 3	68
Figura 4.10	Mapa de porcentaje de señal Enrutador 4	68

ÍNDICE DE TABLAS

Tabla I	Propiedades de los Medios	12
Tabla II	Índices de Refracción	15
Tabla III	Funciones de los componentes del Enrutador	29
Tabla IV	Archivos de Configuración de OpenWrt	34
Tabla V	Características del Linksys WRT54GL	40
Tabla VI	Paquetes adicionales necesarios para obtener los datos	47
Tabla VII	Porcentaje de la señal promedio vs Tiempo	58
Tabla VIII	Porcentaje de la de la señal vs ubicación	65

INTRODUCCIÓN

Un enrutador inalámbrico es un dispositivo utilizado en todo tipo de redes y permite la interconexión de las mismas; su función principal es la de guiar los paquetes de datos para que se transmitan a la red correcta y determinar que caminos deben seguir para arribar a sus destinos. La tecnología de comunicación que poseen estos dispositivos se basa en ondas de radio; sin embargo entre cada obstáculo que atraviesa la señal pierde intensidad reduciendo así su cobertura, esto afecta de forma negativa la transmisión de los datos.

Todos los enrutadores inalámbricos poseen procesadores, memoria de almacenamiento y memoria flash que tienen la misma función del CPU, disco duro y memoria RAM de una computadora normal con la diferencia fundamental en sus capacidades por ejemplo: el procesador del enrutador inalámbrico solo es capaz de procesar datos aproximadamente a un 10% de velocidad de lo que puede realizar un procesador de computadora personal de escritorio; el almacenamiento y la memoria de arranque del enrutador mencionado solo representan un 0.03% y 0.4% respectivamente del usado actualmente por una computadora personal.

Un enrutador inalámbrico es un dispositivo que por lo general funciona las 24 horas del día y como se mencionó dispone de un procesador y memorias de arranque y almacenamiento; por lo antes indicado podemos extender sus funcionalidades para realizar tareas distintas al enrutamiento de paquetes en beneficio del usuario. Con el uso de un firmware diferente al de fábrica se puede utilizar el enrutador inalámbrico para distintos usos como por ejemplo: servidores de voz sobre IP, servidores de archivos, servidor de impresión, servidor NTP. En nuestro caso realizar la captura de datos para el monitoreo de las redes existentes.

Nuestro sistema tiene la ventaja del uso de las técnicas de programación en tiempo real permitiendo crear un sistema con rendimiento confiable y mayor prioridad de ejecución que el resto de los sistemas normales. La programación en tiempo real nos permite obtener las respuestas del sistema de forma más rápida para agilizar el resultado final presentado al usuario.

CAPÍTULO 1

Descripción General del Proyecto

Según un estudio presentado por Pew Internet en el 2010 “el 59% de la población adulta accede a internet mediante conexión inalámbrica Wi-Fi o a través del móvil. Estos datos ponen de manifiesto la tendencia de la población a utilizar la conexión a internet en cualquier situación, convirtiendo su uso en algo primordial” [1].

El uso de las redes Wi-Fi va en aumento; esto se debe a varios motivos como por ejemplo: la necesidad de mantener conectividad a toda hora y en todo lugar, el soporte a la movilidad, etc. No resulta extraño encontrarse con personas sentadas en centros comerciales con ordenadores portátiles conectados a alguna red de libre acceso para navegar en internet.

Sin embargo el mayor inconveniente en la conectividad de las redes Wi-Fi es que realiza transferencias de datos a una menor velocidad en comparación a una conexión con cable, debido a las interferencias y pérdidas de intensidad en la señal recibida ocasionadas por el entorno, los cuales se detallan en el capítulo 2.

Ante esta desventaja y al ser imposible quitar toda interferencia que pueda existir, planteamos un método para sobrellevar de mejor manera el problema; proponemos un sistema que nos permita conocer de forma semi-automatizada cual es el porcentaje de señal recibida en el lugar que nos encontremos a través de un mapa que muestre por cada red existente los porcentajes de sus señales.

1.1. JUSTIFICACIÓN

El proyecto consiste en la implementación de un sistema portátil de monitoreo de redes que como resultado muestra mapas de porcentajes de las señales de redes Wi-Fi recibidas. Para visualizar su funcionamiento expondremos un ejemplo de uno de sus posibles usos:

Actualización de Bases de Datos en carros de monitoreo

Existen carros que monitorean incidentes tales como control de velocidad de la comisión de tránsito, control de robos de la policía

nacional, etc. Todos estos sistemas utilizan radares y necesitan actualizar continuamente sus bases de datos para realizar de forma eficiente sus funciones de control. La desventaja existe cuando realizan sus actualizaciones de datos de forma inalámbrica en redes Wi-Fi en los estacionamientos de sus respectivas instituciones, al ser parqueaderos cubiertos, con múltiples puntos de acceso y tener interferencias variables en el tiempo (autos estacionados) es muy difícil garantizar una correcta transferencia de datos; para la solución de esta situación planteamos el uso de nuestro sistema en el cual con el uso del enrutador inalámbrico monitoreamos las redes existentes en los lugares disponibles para estacionar, luego usando una computadora generamos un mapa con los datos obtenidos para saber cuál es el lugar de los monitoreados que tiene en ese momento mejor señal para garantizar la transferencia de datos.



Fig. 1.1 Ejemplo de uso en un estacionamiento

1.2. OBJETIVOS

1.2.1. Objetivo general

- Extender las funcionalidades de un dispositivo embebido (enrutador inalámbrico) para poder construir mapas de porcentajes de señal en tiempo real.

1.2.2. Objetivos específicos

- Instalar en un dispositivo embebido (enrutador inalámbrico) un firmware de distribución Linux.
- Investigar las llamadas al sistema necesarias para acceder al módulo de red inalámbrica en un enrutador inalámbrico.
- Implementar captura de datos de porcentaje de señal inalámbrica de las redes disponibles, mediante el uso de técnicas de programación concurrente.
- Analizar los datos para la construcción de mapas de porcentajes de señales Wi-Fi recibidas.
- Adaptar una fuente de alimentación portátil (batería) al enrutador inalámbrico para poder usarlo como un dispositivo móvil.

1.3. METODOLOGÍA

Los pasos para la realización del proyecto son:

- Seleccionar un enrutador inalámbrico que sea compatible con distribuciones Linux para sistemas embebidos.
- Seleccionar el firmware adecuado para el enrutador según su marca y modelo.
- Instalar el firmware seleccionado en el enrutador.
- Analizar que paquetes de instalación podemos utilizar que sean compatibles con el procesador de nuestro dispositivo.
- Realizar el ejecutable en lenguaje C para la obtención de los datos y su análisis en el enrutador mediante el uso de la técnica del cross compiling (compilación cruzada).
- Utilizar técnicas de programación en tiempo real para darle mayor velocidad de respuesta a nuestro sistema.
- Uso del software gnuplot en una distribución Linux para realizar las gráfica de los mapas de porcentajes de señal.
- Evaluación y pruebas.

CAPÍTULO 2

Fundamentos teóricos

En este capítulo se exponen los conocimientos sobre los que se fundamenta el desarrollo de nuestro proyecto. Trataremos temas como las redes inalámbricas Wi-Fi, el tipo de onda que usan para la comunicación y los fenómenos físicos por los que son afectadas en el medio en el que se transmiten. Luego se incluye una descripción acerca del firmware que usamos en el dispositivo, métricas de calidad de servicio y por último presentaremos conceptos de la optimización de sistemas en tiempo real.

2.1. REDES INALÁMBRICAS WI-FI

Una red inalámbrica es una conexión de equipos o nodos en la que se implementan protocolos, que permiten el intercambio de datos sin la

utilización de cables. En la figura 2.1 se muestran ejemplos de diferentes tipos de dispositivos que son capaces de implementar conectividad inalámbrica Wi-Fi[4][5]:

- Tarjetas de Red Wi-Fi
- Computadoras Portátiles
- Impresoras Wi-Fi
- Smartphone
- Cámaras fotográficas Wi-Fi

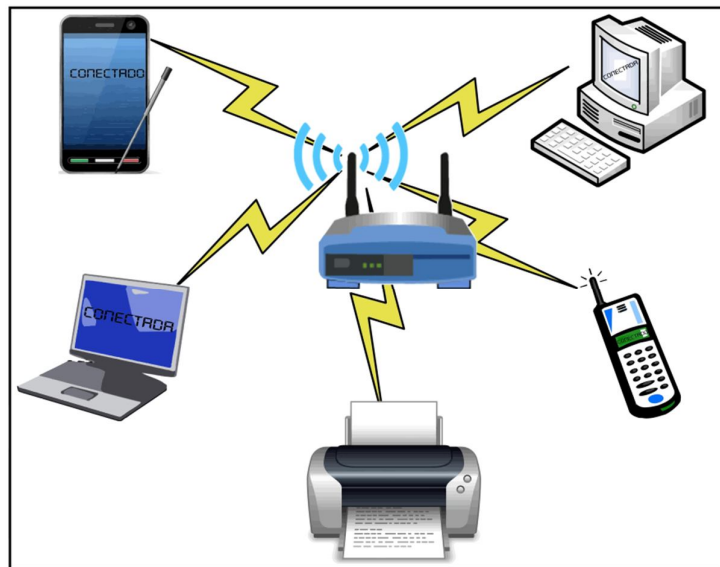


Fig. 2.1 Dispositivos Wi-Fi

2.1.1. Evolución de las Redes Wi-Fi

La tecnología inalámbrica Wi-Fi inició su desarrollo al principio de los años 90 de una forma desordenada a consecuencia de la

variedad de fabricantes que existían, quienes desarrollaban sus equipos incompatibles con los de otros fabricantes. Por este motivo a finales de los años 90 se creó una asociación conformada por empresas como: Lucent, 3Com, Cisco, Intersil, Agere, Nokia y Symbol Technologies; denominada como WECA cuyos objetivos se basaron en fomentar la tecnología Wi-Fi y establecer estándares para que los equipos que la usen fueran compatibles entre sí.

El primer estándar para esta tecnología fue establecido en el año 1999, y designado como IEEE 802.11b, utiliza la banda de los 2.4 GHz y alcanza una velocidad máxima de 11 Mbps. Luego en ese mismo año se presentó el estándar IEEE 802.11a que trabaja en la banda de los 5 GHz y alcanza una velocidad de 54 Mbps; esta norma presentó problemas en el continente Europeo porque utilizaba una banda de radio frecuencia reservada para fines militares, lo que paralizó en cierta medida el avance de la tecnología Wi-Fi en este continente, no fue sino hasta el año 2002 que se resolvió el problema cuando se liberó en Europa la banda de los 5GHz permitiendo el uso de los equipos Wi-Fi con este estándar.

En el año 2003 la WECA pasó a llamarse Wi-Fi Alliance. Después de varios estudios en Junio de ese año se certificó una nueva especificación, la norma IEEE 802.11g que trabaja en la banda de los 2.4 GHz y alcanza una velocidad máxima de 54 Mbps. Posteriormente al coexistir tres normas vigentes en el mercado se pudo comprobar que eran incompatibles entre sí y la solución que se dio a este problema fue la creación de hardware capaz de trabajar con los tres estándares. [3]

2.1.2. Fenómenos Físicos Intrínsecos a las Redes Wi-Fi

Las redes inalámbricas de tipo Wi-Fi utilizan ondas de radio frecuencia (RF) para realizar la comunicación, se transportan por el aire en forma de ondas electromagnéticas formadas por la unión de un campo eléctrico y otro magnético. Usan una banda ISM de tipo compartido de 2,4 a 5 GHz, por lo que se tienen interferencias con transmisores domésticos, cámaras inalámbricas, celulares, etc. Porque estos equipos trabajan con el mismo tipo de ondas que los dispositivos que implementan conectividad inalámbrica Wi-Fi.

La comunicación Wi-Fi está expuesta a los mismos fenómenos físicos capaces de modificar el patrón de las ondas que afectan

a las señales RF. En condiciones ideales las ondas RF tienden a desplazarse en línea recta pero en condiciones normales están expuestas a sufrir cambios debido a los siguientes fenómenos físicos:

- Absorción
- Reflexión
- Refracción
- Difracción

Absorción

La absorción es un fenómeno por el cual la onda RF disminuye su intensidad de señal cuando atraviesa un obstáculo; esto se debe a que parte de la señal se absorbe y se convierte en otro tipo de energía usualmente calórica, mientras que la parte restante disminuye su intensidad y sigue propagándose.

La disminución de la intensidad en una señal se mide en belios (B) o en su submúltiplo más utilizado el decibelio (dB), esta depende ampliamente del tipo de material que atraviesa la onda. La fórmula matemática usada para calcular la disminución de la señal es:

$$R(dB) = (10) * \log\left(\frac{S_2}{S_1}\right) \quad (1)$$

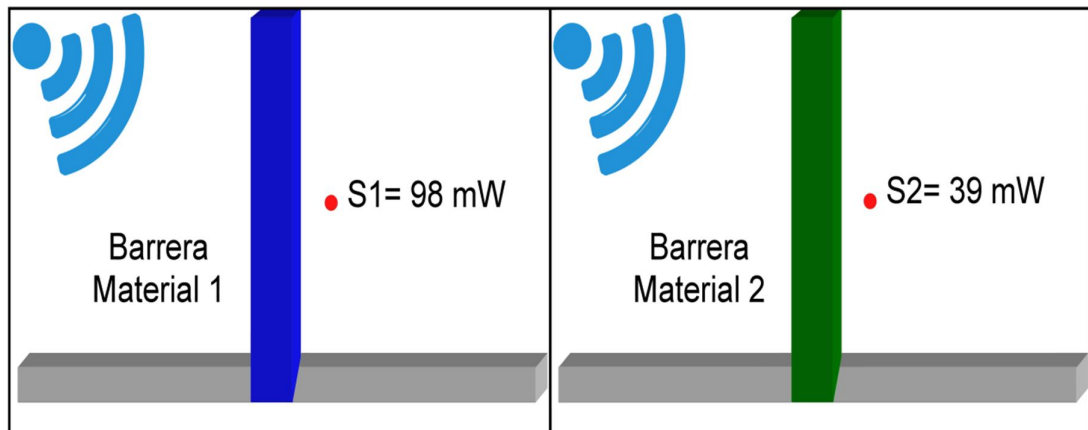


Fig. 2.2 Absorción: Disminución en la señal según el obstáculo traspasado.

$$R(dB) = (10) * \log\left(\frac{39}{98}\right) = -4 dB \quad (2)$$

En la ecuación (2) se modela la atenuación de la señal debido al obstáculo atravesado relacionado directamente con el material de la muro; ocasionado por la absorción que produce cada propiedad de los medios pasados. Las condiciones y las distancias del punto de acceso a la toma de datos son las mismas solo cambia el material de la barrera. A continuación se muestra una tabla comparativa del grado de absorción de los diferentes tipos de materiales.

Tabla I Propiedades de los Medios

Material	Grado de Atenuación	Ejemplo
Aire	Ninguno	Aire libre sin obstáculos
Plástico	Bajo	Anaqueles
Madera	Bajo	Puertas
Vidrio	Bajo	Ventanas
Agua	Medio	Peceras
Ladrillos	Medio	Paredes
Seres Vivos	Medio	Personas
Cerámica	Alto	Pisos
Concreto	Alto	Columnas
Vidrio Reforzado	Alto	Ventana a prueba de balas
Metal	Muy Alto	Armarios metálicos

Reflexión

Cuando una onda de radio choca con un obstáculo, parte o la totalidad de la onda se refleja observándose una pérdida de la intensidad en la propagación. El ángulo en que se refleja la onda es el mismo con el que incide en la superficie. En la figura 2.3 se muestra la igualdad de los ángulos que se forman entre la onda incidente y la superficie con el ángulo que se forma entre la onda reflejada y la superficie.

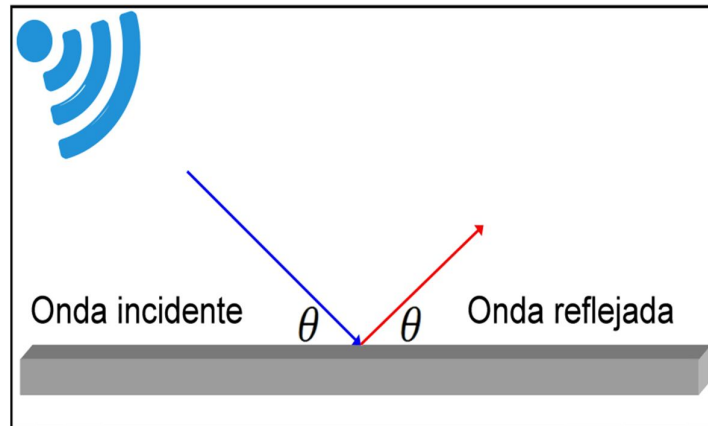


Fig. 2.3 Ángulo de la onda incidente y la onda reflejada

Una onda RF es capaz de propagarse en varias direcciones por lo que una señal de origen puede llegar a una estación o AP después de reflejarse muchas veces y haber tomado múltiples rutas diferentes. La propagación multirutas induce una diferencia temporal denominada retraso de propagación que interfiere en la recepción de los datos; ya que los paquetes se superponen entre sí. Esta diferencia aumenta a medida que se incrementa la velocidad de transmisión debido a que los intervalos de recepción de los paquetes de datos se acortan limitando la transmisión en las redes inalámbricas. Sin embargo este fenómeno es aprovechado en otros dispositivos como sonares, radares, etc. En la figura 2.4 se muestra un sistema de radar de medición de distancias, su funcionamiento se basa en

emitir una señal de radio frecuencia que se refleja en el objetivo y se recibe en la misma posición del emisor.

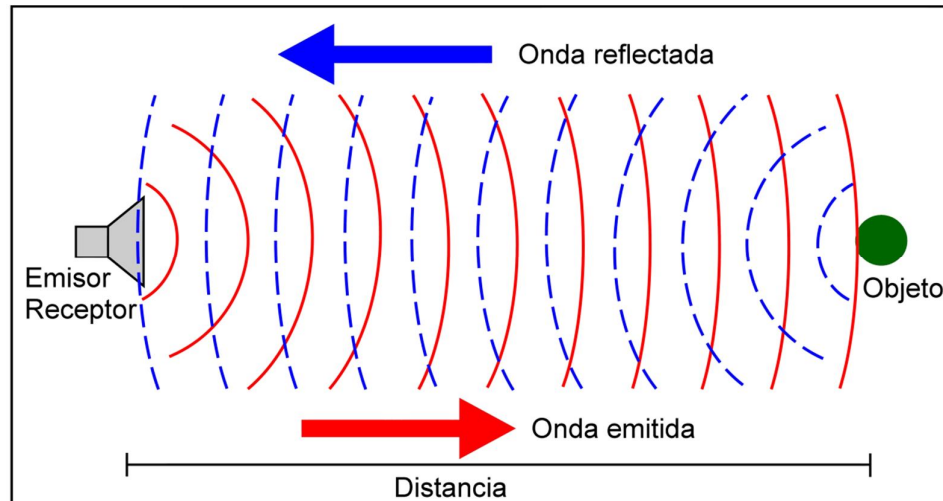


Fig. 2.4 Reflexión de las ondas

Refracción

Cuando las ondas RF pasan de un medio a otro están expuestas a sufrir desviación en su trayectoria produciéndose cambios en su velocidad y en la dirección de propagación. En cada medio en el que se propaga la onda la velocidad de propagación depende directamente del índice de refracción o características propias del medio en el que se encuentra. En la figura 2.5 se muestra el paso de una onda de un medio de propagación a otro, se muestra claramente la forma en que la onda cambia su dirección al cambiar de medio. [7]

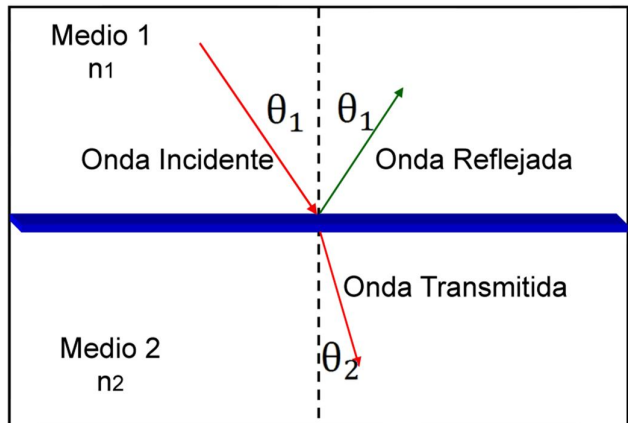


Fig. 2.5 Desvío de la onda al pasar de un medio a otro

Existe una relación entre el ángulo de incidencia y el ángulo de refracción conocida como Ley de Snell en la cual se enuncia que el producto del índice de refracción del primer medio por el seno del ángulo de incidencia es igual al producto del índice de refracción del segundo medio por el seno del ángulo de refracción.

$$n_1 \text{sen} \theta_1 = n_2 \text{sen} \theta_2 \quad (3)$$

Tabla II Índices de Refracción
Señales RF con longitudes de onda de 589 mu

Agua	1.33
Vidrio	1,46 – 1,96
Polietileno	1,59
Cuarzo	1.54

Difracción

El fenómeno de difracción se observa cuando en su trayectoria la onda se encuentra con una barrera que posee aberturas o con los bordes de un objeto sólido; la onda en lugar de seguir avanzando en línea recta rodea los bordes del objeto o atraviesa las aberturas formando nuevos frentes de ondas o tipos de propagación. Para que se observe difracción en una onda es necesario que el tamaño de los agujeros sean iguales o menores que la longitud de la onda; por lo que la difracción depende directamente de la longitud de la onda que se está propagándose. Como se observa en la figura 2.6 dependiendo del tamaño del orificio los frentes de onda o forma de la onda que se observan varían.

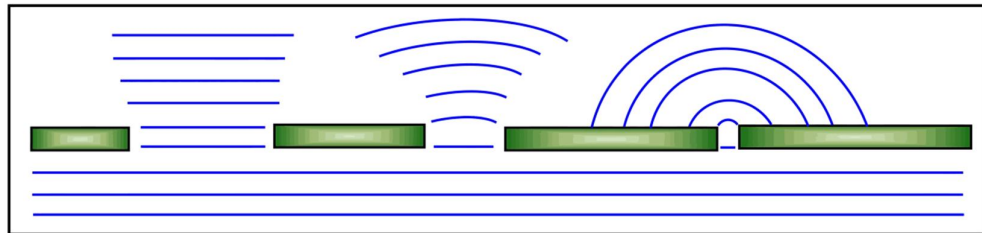


Fig. 2.6 Propagación de la señal a través de un orificio

Un ejemplo común de difracción en ondas mecánicas son las olas producidas por una embarcación en un río; si la embarcación es pequeña las olas lo bordean y se puede observar el oleaje detrás del barco, pero si la embarcación es

grande en comparación con el tamaño de las olas formadas solo se aprecia la difracción en el borde y detrás del barco se observa una zona sin oleaje. Este es un ejemplo para ondas mecánicas pero el efecto del fenómeno de difracción es igual para una onda electromagnética RF o para una onda bajo el estándar Wi-Fi solo hay que tener en cuenta la longitud de la onda que es de 12.248 cm.

Los fenómenos antes enunciados de reflexión, refracción y difracción son básicos para entender el efecto de la multitrayectoria entre la antena emisora y la antena de recepción, es decir que las señales que llegan al receptor a través de diferentes caminos y tiempos diferentes, esto es muy importante en las redes inalámbricas por los retardos de tiempo e interferencia que provoca. La señal que llega al receptor está compuesta por ondas de radios directas del emisor y por una gran cantidad de ondas reflejadas, refractadas o difractadas por diferentes objetos como paredes, vidrios, suelos, etc. Todas estas ondas que se forman se suman o restan vectorialmente por lo que la señal percibida nunca proviene en línea recta ni en el caso que el emisor y receptor estén alineados en visión directa. En la figura 2.7 (a) podemos observar el efecto de la

propagación multitrayectoria en la cual se muestra claramente las diferentes trayectorias que puede tomar una onda a pesar de que el emisor y receptor se encuentran uno en frente del otro y en la figura 2.7 (b) se observa el retraso en el tiempo provocado por la propagación en multitrayectoria y la onda resultante que llega al receptor.

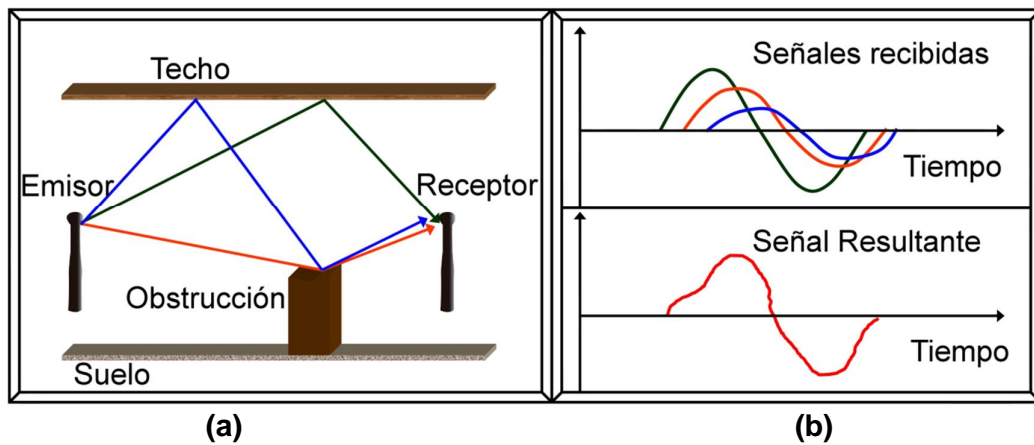


Fig. 2.7 Propagación en multitrayectoria
(a) Multitrayectoria entre el emisor y receptor **(b)** Efecto en el tiempo y en la onda resultante de la propagación en multitrayectorias

Existen dos clases de multitrayectoria: la multitrayectoria especular debido a las reflexiones y la multitrayectoria difusa ocasionada por las difracciones. Ambas multitrayectorias son interferencias para las comunicaciones inalámbricas. La multitrayectoria difusa ocasiona un nivel de ruido de fondo de interferencia o ruido aleatorio de pequeña magnitud; mientras que la multitrayectoria especular puede causar interrupciones

en la señal o crear aéreas con una intensidad de señal muy reducida como en el caso de túneles, habitaciones pequeñas, callejones, etc.[6][8][17]

2.2. MÉTRICAS DE CALIDAD DE SERVICIO (QOS)

Nivel de señal, relación señal a ruido y calidad de enlace son tres indicadores proporcionados por los equipos. Utilizados para realizar el análisis del porcentaje de la señal en un punto determinado y la calidad de la transmisión de la información. A pesar de estar mencionados dentro del estándar IEEE 802.11 no hay una forma específica de su cálculo por lo que cada fabricante de equipos inalámbricos posee su propia forma de calcularlos, a continuación definiremos a que se refieren cada una de estas métricas:

Nivel de Señal: Es la diferencia entre el nivel de una señal en un punto de transmisión y el nivel de la señal recibida en el punto receptor, este indicador mide la calidad de la transmisión al comparar la señal que se envía y la señal que se recibe; si la diferencia de estos dos valores es muy grande significaría que una gran cantidad de ruido (perturbación) se introdujo en el camino de la señal.

$$S_r = G_{se} - P_{ce} - P_{ae} + G_{ae} - P_p + G_{ar} - P_{cr} - P_{ar} - P_a \quad (4)$$

Gse: Ganancia de salida del equipo transmisor.

Gae: Ganancia de la antena del equipo transmisor.

Pce: Pérdida por cables del equipo transmisor.

Pae: Pérdida por conectores y puertos del equipo transmisor.

Pp: Pérdida por propagación.

Gar: Ganancia de la antena del equipo receptor.

Pce: Pérdida en cables del equipo receptor.

Par: Pérdida por conectores y puertos del equipo receptor.

Pa: Pérdidas adicionales ocasionadas por condiciones ambientales.

Calidad de Enlace: Este indicador está propuesto en la norma IEEE 802.15, pero su evaluación no está estandarizada, así que la forma de calcularlo depende de cada fabricante; el LQI por sus siglas en inglés (Link Quality Indicator) está basado en el número de paquetes de datos enviados y el número de paquetes de datos recibidos de forma que refleja la cantidad de señal en el canal formado por dos estaciones; por ejemplo un punto de acceso y un cliente.

Relación SNR: Es la diferencia entre la intensidad de la señal que se transmite y la intensidad del ruido que la corrompe. Este indicador se

medida en decibelios; proporciona una comparación de la intensidad de la señal con la cantidad de ruido de fondo que posee de tal forma que un indicador SNR mayor indica un ruido menos perceptible; este indicador es ampliamente usado para evaluar ondas electromagnéticas. En la figura 2.13 podemos observar la relación SNR. [9]

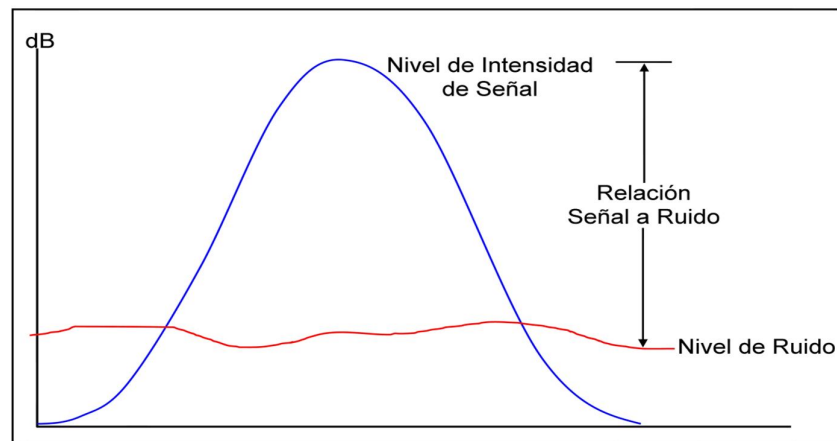


Fig. 2.8 Relación Señal a Ruido (SNR)

2.3. SISTEMAS EMBEBIDOS

Los sistemas embebidos o también conocidos como sistemas empotrados son sistemas informáticos construidos dentro de un dispositivo de mayor tamaño y capacidad, son diseñados para realizar soluciones específicas, es decir que están hechos para realizar una sola función; frecuentemente usados en procesamiento en el cual las restricciones de tiempo son específicas como por ejemplo en el ámbito

de la medicina el software encargado de controlar un respirador artificial o un marca pasos.



Fig. 2.9 Interior Enrutador Linksys WRT160NL

En un sistema embebido, los componentes se encuentran incluidos en la placa base y pueden resultar muy diferentes en aspecto a las computadoras. Ejemplo de ello son los taxímetros que a simple inspección no poseen dispositivos de entrada como un teclado. Estos sistemas se pueden programar directamente en el lenguaje ensamblador del microcontrolador o microprocesador que tengan incorporado o pueden usar compiladores específicos para poder utilizar lenguajes como C o C++ y luego utilizar un proceso denominado compilación cruzada para llevarlo al sistema embebido (ver sección 3.5).

Un dispositivo embebido posee básicamente los siguientes componentes (ver figura 2.10). [10][13]

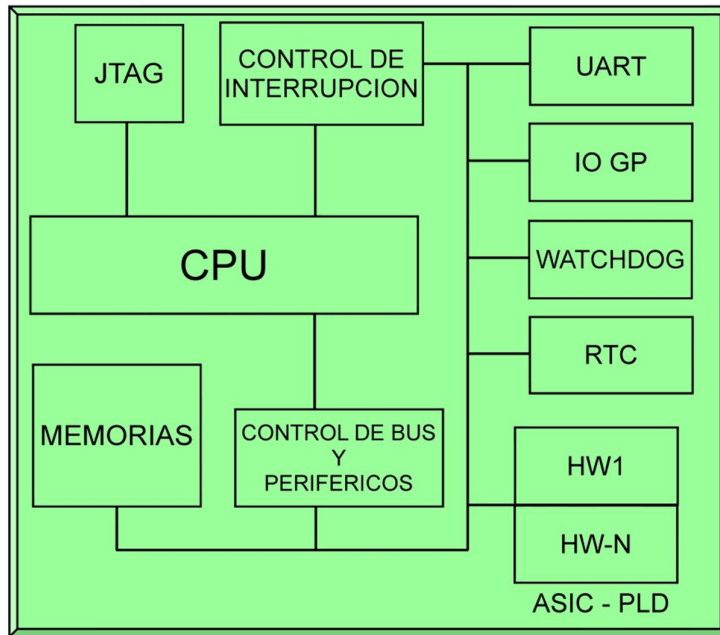


Fig. 2.10 Arquitectura de un Sistema Embebido

CPU: Es el microprocesador que realiza las operaciones de cálculo principales del sistema, dirige el funcionamiento de los demás elementos que conforman el dispositivo embebido.

Memorias (RAM, NVRAM, FLASH, ROM): Las memorias incluidas en un dispositivo embebido tienen diferentes usos entre estos almacenamientos, arranque, ubicaciones de archivos, etc.

JTAG: Es la norma IEEE 1149.1 creada para testar los puertos del circuito impreso y para probar los submódulos de circuitos integrados por el escáner de límites; es útil también para la depuración de

aplicaciones embebidas de forma que permite al programador corregir errores de código y lógica del sistema.

Controlador de Interrupciones: Es un componente de hardware dentro del circuito integrado (chipset) usado para combinar varias fuentes de interrupciones sobre una o más líneas del CPU.

Controlador de Buses y periféricos: Controla el uso y acceso a las líneas de datos y direcciones compartidas por todos los componentes, es decir es el que controla que no haya colisión en la información del sistema.

UART: Controla los puertos y dispositivos seriales está integrado en la placa base del dispositivo. Sus funciones principales dentro del sistema son manejar las interrupciones de los dispositivos conectados al puerto serie y convertir los datos en formato paralelo.

GP IO: Son puertos de señales de propósito general genéricos, no poseen un uso específico pero la idea de incluirlos dentro del sistema es poder tener nuevas líneas de control digital para funcionalidades adicionales. En el control vía GP IO el software se comunica con un hardware externo a través de salidas lógicas o entradas lógicas.

Watchdog: Es un temporizador que se utiliza para detectar y recuperar del mal funcionamiento al equipo. Cuando se activa este temporizador por una falla o problema de software el organismo central reinicia el sistema.

RTC: Es un reloj de ordenador incluido en el circuito integrado que mantiene la hora actual.

HW-N: Oculta las características físicas de una plataforma informática de los usuarios. Por ejemplo las máquinas virtuales nos permiten tener varios ordenadores virtuales ejecutándose sobre el mismo computador físico

ASIC O PLD: ASIC es un circuito integrado para aplicaciones específicas es decir poseen un uso particular como por ejemplo un chip diseñado únicamente para ser usado en un teléfono móvil es un ASIC. Un PLD es un componente electrónico programable que permite utilizarlo en funciones diversas según sea programado.

2.3.1. Sistema Embebido vs Computadora de uso general

Los sistemas embebidos tienen tres características principales que los diferencian de un sistema de cómputo de uso general:

Funcionamiento Específico: Un sistema embebido usualmente ejecuta de forma repetitiva el software para el que fue diseñado y una computadora de uso general ejecuta una amplia variedad de programas además de tener la capacidad de instalar nuevos software. Aunque existen excepciones en los que se puede actualizar el sistema embebido y extender sus capacidades, como en el caso del enrutador inalámbrico que se usa en este proyecto.

Fuertes Limitaciones: Todos los sistemas electrónicos o de computación poseen limitaciones en cuanto al tamaño, costo, desempeño y consumo de energía pero en lo que respecta a los sistemas embebidos estas limitaciones son mucho mayores por lo general deben de ser poco costosos, tener tamaño reducido, tener un buen desempeño para procesar datos dentro del tiempo de respuesta valido (tiempo real) y consumir poca energía para evitar la utilización de dispositivos adicionales para disipar el calor.

Reacción en intervalo de tiempo limitado: Muchos sistemas embebidos deben reaccionar ante cambios en el ambiente, realizar cálculos sin retrasos y tener resultados fijos. Por ejemplo los frenos ABS de un automóvil que monitorean las señales de deslizamiento para actuar ante cualquier inconveniente mientras que un sistema de computación de uso general se centra en realizar los cálculos con una frecuencia no determinada y el retraso de los mismos no producen que el proceso sea considerado como no útil .

2.3.2. Enrutadores Inalámbricos

Un enrutador inalámbrico es un dispositivo embebido utilizado en redes sin cableado que permite la interconexión entre los dispositivos. El propósito de un enrutador es examinar los paquetes de datos entrantes a la red, elegir la mejor ruta para luego conmutarlos hacia el puerto de salida adecuado, son los dispositivos que regulan el tráfico de la red; el enrutador puede tomar decisiones basadas en grupos de direcciones de red y conectar distintas tecnologías como por ejemplo Ethernet, Wi-Fi, Token-ring, etc. [2]

Los datos transmitidos desde un emisor a un receptor se dividen en paquetes de datos, cada paquete es de aproximadamente 1500 bytes; el enrutador se encarga de examinar todos los paquetes que conforman el mensaje y luego buscará el camino más corto y rápido en la red para enviar la información a su respectivo destino. Al ser el enrutador inalámbrico un dispositivo embebido tiene la arquitectura interna diseñada para que sus componentes efectúen papeles fundamentales como se mencionó en la sección 2.2. En la figura 2.11 se observa los módulos internos funcionales del enrutador inalámbrico.

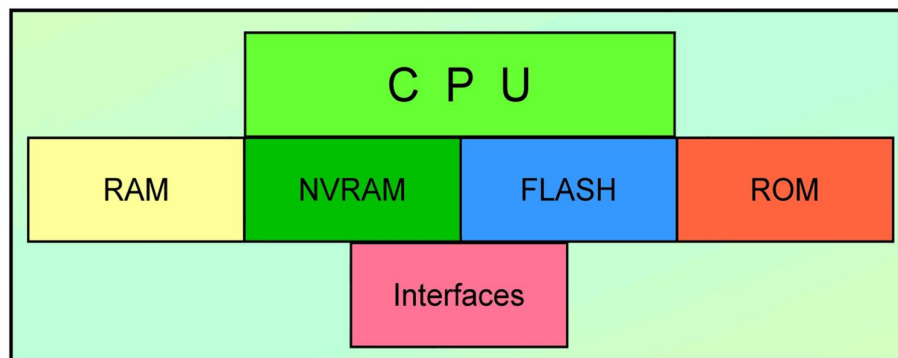


Fig. 2.11 Arquitectura interna del enrutador

En general los enrutadores inalámbricos poseen los módulos mostrados en la figura 2.11 divididos según su función específica dentro del dispositivo. En la tabla III se muestra el funcionamiento específico de cada módulo mostrado.

Tabla III Funciones de los componentes del Enrutador

Componente	Función en el enrutador inalámbrico
CPU	Función de enrutamiento y control de la interfaz de red
RAM	Almacenamiento de tablas de enrutamiento y colas de espera de los paquetes
NVRAM	Almacena copia del archivo de configuración
FLASH	Almacena el firmware
ROM	Almacena los diagnósticos de encendido
INTERFACES	Conexiones de red

2.4. EI FIRMWARE

Cuando se habla del firmware nos referimos al bloque de instrucciones de máquina, grabados en un dispositivo de lógica programable que controla los circuitos electrónicos de un dispositivo. El firmware provee la interacción entre el sistema operativo y el hardware. Muchos de los dispositivos electrónicos que utilizamos están controlados por un firmware almacenado en una memoria interna (ROM, EPROM, FLASH).

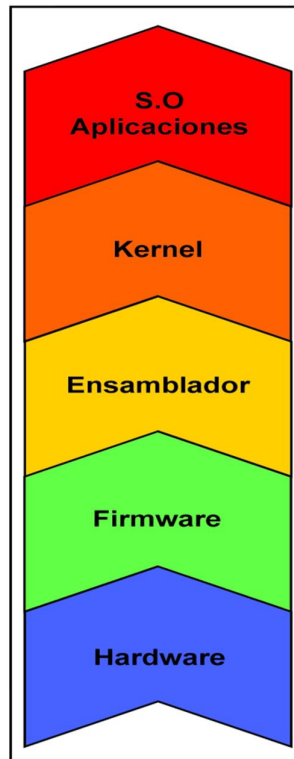


Fig. 2.12 Arquitectura de una computadora

El firmware de un sistema es igual de importante que el sistema operativo de una computadora por las relevantes funciones que realiza; sin embargo a diferencia del sistema operativo los firmwares en la mayoría de los casos no poseen un mecanismo automatizado para actualizarse y corregir problemas de funcionalidad detectados después de su puesta en el mercado, además ejecutan menos funciones como por ejemplo: el manejo de la memoria. El firmware es considerado parte del hardware por estar integrado dentro de un chipset y a su vez también es considerado parte del software al ser

desarrollado bajo un lenguaje de programación. Por ejemplo el denominado BIOS de una computadora es un firmware cuyo propósito es activar la maquina desde que se la enciende hasta el momento en que se carga el sistema operativo en la memoria RAM. La arquitectura de la computadora está compuesta por una serie de capas con distintos niveles como se muestra en la figura 2.13; el hardware es la base y luego aparece el firmware como intermediario entre las ordenes externas que recibe el dispositivo y sus componentes electrónicos además de controlar la secuencia de arranque e iniciación del sistema; luego se encuentran el ensamblador que es el encargado del manejo de interrupciones del sistema, el kernel que forma parte del sistema operativo y se encarga de la comunicación entre procesos, control de periféricos y el manejo de memoria; el sistema operativo y al final las aplicaciones. Muchos dispositivos incluyen la opción de actualizar el firmware para corregir errores o mejorar su desempeño como los enrutadores inalámbricos; pero se debe ser muy cauteloso al realizar actualizaciones del firmware ya que un fallo en el proceso deja al dispositivo inutilizable.

2.5. OPENWRT

OpenWrt es un firmware utilizado en dispositivos embebidos para conmutar el tráfico de red, es una distribución de Linux derivada directamente de Debian para dispositivos embebidos como enrutadores personales, con un sistema de archivos totalmente modificable permitiendo personalizar el dispositivo incorporando aplicaciones a través del uso de paquetes de instalación. El firmware OpenWrt nos permite adaptar el enrutador para que realice funciones propias de una herramienta mucho más especializada y de mayor costo; ya que el firmware estándar provee solo una interfaz de fácil configuración sin explotar el verdadero potencial del enrutador; algunas de las características generales del proyecto OpenWrt son:

- Posee la misma estructura de Unix: Kernel, Shell y sistemas de archivos.
- El Kernel, Shell y el sistema de archivos de OpenWrt realizan las mismas funciones que en el sistema Unix.
- El kernel de OpenWrt al igual que el kernel de Unix controla la memoria del dispositivo, distribuye las tareas realizadas por el procesador, ejecuta las instrucciones del Shell y se encarga de cumplir los permisos del sistema de archivos.

- El Shell de OpenWrt es el ash Shell muy parecido al bash Shell de Unix ambos permiten que se ejecuten los comandos y programas ejecutados.
- El sistema de archivos tiene la función de organizar los datos de archivos, directorios y dispositivos conectados. [12]

OpenWrt se configura a través de una interfaz de línea de comandos (ash) o un interfaz web (LUCI) y utiliza fuentes del núcleo oficiales de GNU/Linux por lo que solo añade parches para el sistema de chips y controladores de la interfaces de red de los diferentes dispositivos. Se puede ejecutar no solo en enrutadores sino también en teléfonos inteligentes por ejemplo Neo FreeRunner, ordenadores de bolsillo como Ben NanoNote y pequeños portátiles como One Laptop per Child. La última versión estable de OpenWrt es denominada bajo el nombre clave de Backfire 10.03, la última revisión fue lanzada en Diciembre del 2011 y corresponde a OpenWrt Backfire 10.03.1.

Un enrutador que tiene como firmware Openwrt puede extender sus funcionalidades de enrutamiento a través de la instalación y/o creación de scripts, programas y paquetes que le permitan al enrutador implementar nuevas funcionalidades como por ejemplo: Servidor VPN, control del porcentaje de señal de emisión Wi-Fi, servidor de impresión, servidor web, acceso a base de datos MySQL entre otras.

Archivos de configuración que posee OpenWrt

La configuración de OpenWrt está dividida en varios archivos ubicados en el directorio `/etc/config/`, cada archivo contenido en este directorio corresponde a la parte del sistema que configura. La edición de los archivos de configuración puede realizarse en un editor de texto, la sentencia UCI o mediante varios APIs de programación como Bash Shell y en los lenguajes de programación Lua o C.

Tabla IV Archivos de Configuración de OpenWrt

Archivo	Descripción
<code>/etc/config/dhcp</code>	Configuración de los parámetros DHCP: DNS, puerta de enlace de la dirección IP, ARP, etc.
<code>/etc/config/firewall</code>	Configuración del firewall.
<code>/etc/config/network</code>	Configuración de red, interfaces de red y tabla de enrutamiento.
<code>/etc/config/ucitrack</code>	Configuración de los servicios relacionados con los archivos.
<code>/etc/config/wireless</code>	Configuración inalámbrica y de la interfaz Wi-Fi.
<code>/etc/config/dropbear</code>	Configuración del servidor SSH.
<code>/etc/config/luci</code>	Configura los elementos esenciales de la interfaz web.

Archivo	Descripción
/etc/config/system	Configuración básica del sistema del enrutador: nombre del host, zona horaria, etc
/etc/config/uhttpd	Configura del servidor web

2.6. SISTEMAS DE TIEMPO REAL

Un sistema de tiempo real es un sistema informático que debe satisfacer restricciones explícitas en el tiempo de respuesta. Estos tipos de sistemas interactúan repetidamente con su entorno físico y deben responder a un estímulo externo dentro de un tiempo especificado. La eficiencia de los mismos no solo depende de la exactitud de los resultados, sino también del momento en que los entrega por lo que una de sus principales tareas es asegurar la distribución de recursos de tal forma que se cumplan los requerimientos de tiempo.

Toda actividad de un sistema de tiempo real se denomina tarea, estas a su vez tienen tres tipos de propiedades:

- **Funcionalidad:** La función que realizan.
- **Temporalidad:** En qué tiempo realizan su función.
- **Fiabilidad:** La seguridad de realizar su función en el tiempo especificado o antes del límite de tiempo.

En un sistema de tiempo real debe conocerse exactamente el tiempo que le toma al sistema responder a un determinado evento (deadline), este debe ser invariable y cada tarea debe ejecutarse lo suficientemente rápido para no producir fallas. En la figura 2.14 se muestra el tiempo de ejecución de una tarea y que este es igual al tiempo que transcurre hasta que se empieza a ejecutar la tarea más el tiempo de ejecución; para que un proceso o tarea sea valido se debe de terminar de ejecutar el proceso antes del tiempo límite del sistema (deadline).

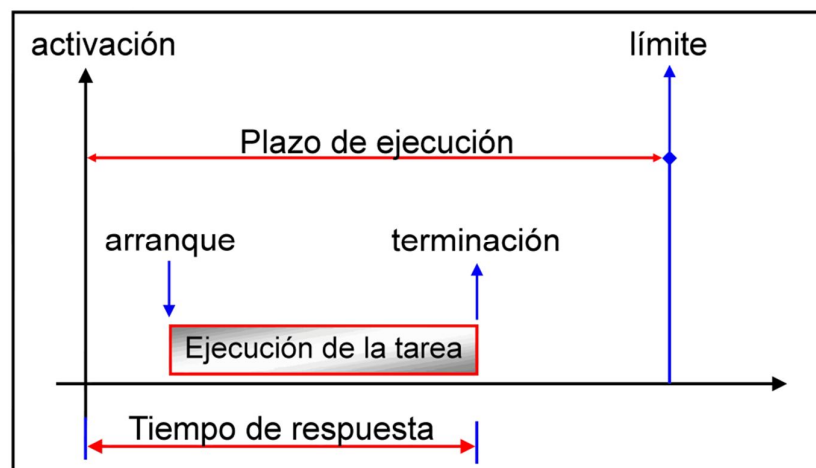


Fig. 2.13 Tiempo de ejecución de una tarea.

En la programación los sistemas RT se expresan en la reunión de técnicas y conceptos de análisis y diseño con el propósito de organizar el desarrollo del sistema y aprovechar al máximo los recursos de

hardware. A continuación mencionaremos las que utilizaremos en nuestro proyecto:

- **Prioridades:** Se establece la prioridad de un proceso por encima de otro acortando así el tiempo de ejecución sin importar si fue finalizado o no.
- **Hilos:** Son tareas de tiempo real que se ejecutan concurrentemente aplicando el concepto de pseudoparalelismo.

Los sistemas de tiempo real se pueden dividir en dos tipos diferentes, dependiendo de la severidad en el tratamiento de errores y estos son:

- Soft Real Time (Sistemas de tiempo real blandos).
- Hard Real Time (Sistemas de tiempo real duros).

2.6.1. Soft Real Time

Un sistema soft real time garantiza que las tareas RT van a tener mayor prioridad que otro tipos de tareas y su resultado es útil aun después del deadline o tiempo límite. Estos sistemas se usan especialmente cuando se debe mantener actualizados los datos de un sistema que trabaja con condiciones cambiantes del exterior.

2.6.2. Hard Real Time

Un RTOS hard es aquel que garantiza o limita que una tarea se realice antes de que termine el deadline. Si la tarea no se realiza dentro del límite de tiempo se considera que el sistema falló. La meta específica de estos sistemas es que todos los deadline sean cumplidos

Este tipo de sistemas de tiempo real se usan estrictamente para casos en lo que un fallo en el tiempo de ejecución y procesamiento sería catastrófico un ejemplo de esto son los sistemas de tráfico aéreo. [14]

CAPÍTULO 3

Implementación y Funcionamiento

En este capítulo se describen los dispositivos usados para la implementación de nuestro sistema. Se detallan las características de hardware y de software empleados para la realización del sistema y la configuración necesaria para su funcionamiento. Se presenta también el diagrama de flujo del sistema junto con sus funciones principales y el funcionamiento del mismo.

3.1. DESCRIPCIÓN DEL HARDWARE

Para la implementación de nuestro sistema utilizamos el enrutador Linksys WRT54GL, popular por ser el primer enrutador que liberó el código fuente del firmware debido a que estaba basado en fuentes

GNU. Es un enrutador inalámbrico de tipo G que posee procesador Broadcom con velocidad de procesamiento de 200MHz, dos antenas externas de polaridad inversa; funciona con cualquier voltaje entre 5 y 20 voltios; con su firmware de fábrica la configuración es a través de una simple interfaz web. En la tabla VI se muestran las principales características del enrutador inalámbrico Linksys WRT54GL.

Tabla V Características del Linksys WRT54GL

Memoria Flash – RAM - Procesador	4MB - 16MB - 200 MHz
Tecnología - Bandas	Wi-Fi Inalámbrica G - 2,4 GHz
Estándares	802.3, 802.3u, 802.11b, 802.11g
Puertos Ethernet	4 x 10/100
Velocidades	Hasta 54Mbps



Fig. 3.1 Enrutador Linksys WRT54GL

Para poder usar al enrutador como un dispositivo móvil requerimos el uso de un batería recargable como fuente de alimentación del sistema. La misma que tiene como voltaje de entrada 12.6 VDC y como voltaje de salida 5VDC y corriente de salida 4800mA. Las dimensiones de la batería son 57x95x20 mm y pesa 195 g. Posee botón de encendido/apagado, salida con conector USB.



Fig. 3.2 Batería recargable Li-on Y SN-05480

3.2. DESCRIPCIÓN DEL SOFTWARE

Para realizar los gráficos con los datos obtenidos usamos el software Gnuplot en Ubuntu una distribución Linux. Gnuplot es un programa que permite generar gráficas 2D y 3D de funciones y de archivos de datos. Es compatible con los sistemas operativos Windows, Linux, Unix y Mac Os X; este programa se distribuye bajo una licencia de software libre pero no tiene ningún vínculo con el proyecto GNU. Gnuplot puede presentar los gráficos directamente en pantalla o guardarlos en una multitud de formatos de imagen que posee en

acabados de una muy buena resolución como por ejemplo: PNG, EPS, SVG, JPEG, etc. Este programa nos permite realizar gráficos bidimensionales con diferentes estilos como: puntos, líneas, barras, flechas, etc. Gráficos tridimensionales de contornos y superficies. Gnuplot permite además la creación de scripts para la obtención de gráficos especiales. www.gnuplot.info [11]

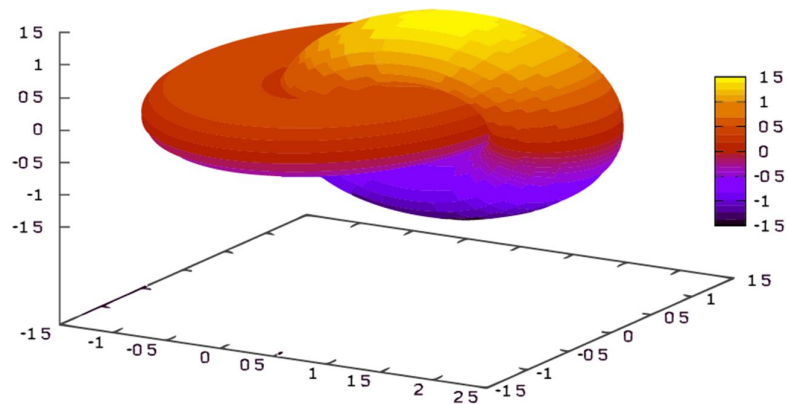


Fig. 3.3 Ejemplo de un gráfico en GNU PLOT

3.3. INSTALACIÓN DEL FIRMWARE OPENWRT

Hay muchas formas de instalar el firmware, se explicara los métodos más usados que son por medio de páginas web y a través de TFTP. Podemos usar cualquier procedimiento y al final el resultado es el mismo.

Por medio de la página web del firmware original:

Primero se abre la interfaz web de configuración del enrutador y usando la página de actualización del firmware del dispositivo procedemos a escoger el archivo de la imagen que descargamos. Se debe esperar unos minutos y por ningún motivo se debe parar la actualización ya que esto dejaría al enrutador inservible.

A través de TFTP:

Este método es utilizado comúnmente para reinstalar una instalación fallida.

Esta opción se realiza desde tftp en Linux. Se debe tener el cable de alimentación del enrutador desconectado y configurar al ordenador con una IP estática 192.168.1.X con máscara de subred 255.255.255.0. Luego accedemos a la ruta en donde se encuentra el firmware de OpenWrt descargado desde una terminal, y ejecutamos los siguientes comandos:

```
tftp 192.168.1.1
binary
rexmt 1
timeout 60
trace
put openwrt-XXXXXXXXX.bin
```

La opción rexmt 1 se utiliza para que se intente constantemente enviar el archivo a la dirección antes proporcionada. Luego de introducir los comandos se conecta el cable de alimentación del enrutador. Cuando

el cargador de arranque del enrutador comience a esperar el envío del firmware, automáticamente se le enviara. En ningún momento se debe apagar el enrutador, cuando el firmware este actualizado el enrutador se reiniciara. La forma de obtener y descargar el firmware OpenWrt se explica en el anexo A.

Inicio en OpenWrt

Luego de reiniciar el enrutador, se deberá poder realizar una conexión telnet para acceder al mismo, usando la dirección IP 192.168.1.1. Al entrar por primera vez vía telnet OpenWrt nos pide asignar una contraseña, luego de esto cualquier intento de telnet será fallido, porque se habrá deshabilitado en forma predeterminada este servicio para dar paso a SSH. Esto se realiza para exigir a los usuarios un mínimo nivel de seguridad. Si existiese algún problema al acceder por telnet la primera vez puede ser por algún problema con los ajustes del firewall, por lo que si hay algún firewall instalado en su ordenador se debe desactivar. Antes de ingresar por SSH luego de asignar la contraseña se debe esperar unos minutos para que OpenWrt ajuste el sistema de archivos y genere las claves de cifrado de SSH; el servidor no se iniciará hasta que se hayan generado las claves de cifrado.

3.4. GESTIÓN DE PAQUETES EN OPENWRT

La utilidad que usa el firmware OpenWrt para la instalación de paquetes es OPKG.

3.4.1. OPKG

OPKG es un gestor de paquetes muy ligero usado para descargar e instalar en OpenWrt a través de internet. Es similar al comando apt-get de las distribuciones Ubuntu. El firmware está diseñado por si solo para ocupar el menor espacio posible y proveer una interfaz razonable de comandos o web de administración. Sin paquetes instalados, OpenWrt configura las interfaces de red, el firewall NAT básico, el servidor seguro de Shell, el servidor DNS y DHCP. El archivo de configuración de OPKG es /etc/opkg.conf

3.4.2. Instalación de paquetes

Para instalar un paquete que está presente en un repositorio usamos el comando:

```
opkg install wireless-tools
```

Para realizar la instalación de un paquete desde una dirección URL o un archivo descargado:

```
opkg install
```

```
http://downloads.openwrt.org/kamikaze/8.09.1/brcm-  
2.4/packages/openssh-sftp-server_5.0p1-1_mipsel.ipk
```

En ambos casos opkg tratará de resolver las dependencias con los paquetes en los repositorios. Si no se logra la instalación, opkg informará un error y no se instalara ningún paquete.

Podemos ver los paquetes instalados con el comando:

```
opkg list_installed
```

Si deseamos borrar un paquete instalado lo podemos hacer con el comando:

```
opkg remove
```

3.4.3. Paquetes de instalación necesarios para la obtener los datos.

Al ser OpenWrt un firmware flexible del cual podemos crear nuestra propia imagen incluyendo los paquetes que necesitemos o deseamos, la imagen del firmware que descargamos solo incluye los paquetes básicos para su funcionamiento por lo cual debemos instalar los que hacen falta buscándolos en el repositorio.

Tabla VI Paquetes adicionales necesarios para la implementación

Nombre	Versión
bash	4.2-1
wl	4.150.10.5.3-9

3.5. COMPILACIÓN CRUZADA

El proceso de compilación cruzada significa crear un código ejecutable para una plataforma distinta a la que se ejecuta en donde se crea el código. Este tipo de compilación es útil cuando no se tiene acceso al entorno en la que se va a ejecutar el código o cuando es imposible compilar en la plataforma señalada por falta de recursos como memoria para poseer un compilador nativo como es el caso de los sistemas embebidos en especial en los enrutadores inalámbricos por sus grandes limitaciones de hardware.

Para realizar este tipo de compilación es necesario contar con una serie de programas y librerías denominadas en Linux como toolchain que está compuesto por un compilador, utilidades para el manejo de ficheros binarios (ensamblador, enlazador y depurador). Para OpenWrt el entorno de desarrollo lleva como nombre SDK y varía de acuerdo a la versión del firmware OpenWrt que este instalado en el enrutador y la

arquitectura de la computadora. El proceso utilizado para realizar compilación cruzada de un código escrito en lenguaje c se describe en el anexo B con el proceso de compilación del programa HolaMundo.

3.6. PROGRAMA PRINCIPAL

El script fue creado en el lenguaje interpretado bash. Es pasado al enrutador inalámbrico a través de SSH con uso del comando de transferencia de archivos SCP (Secure Copy) de Linux. Se ejecuta el script en el enrutador inalámbrico; el script empieza su ejecución capturando los datos, luego los ordena y clasifica según el punto de acceso al que pertenecen, al final de la ejecución los datos son guardados en archivos separados en el directorio /tmp, este directorio es temporal por lo que cada vez que se reinicia el enrutador los datos se van a borrar, de esta forma se trata de no sobrepasar la capacidad de almacenamiento del enrutador. Los archivos con los datos ordenados y clasificados son extraídos del enrutador a través del comando SCP ejecutado desde la computadora ya que el firmware del enrutador OpenWrt no posee la opción de ejecutar este comando.

Luego en el computador con distribución Linux se ejecuta en una terminal el comando gnuplot, se copia el código de graficación y los datos, esto dibuja 1 mapa por cada archivo copiado desde el enrutador

a la computadora. Los gráficos presentan con variación de tono de color los porcentajes de la señal en los puntos tomados de la siguiente forma: a mayor porcentaje de señal un tono de color más fuerte y a menor porcentaje de señal tonos de color más claros. En la figura 3.4 se muestra gráficamente lo antes explicado.

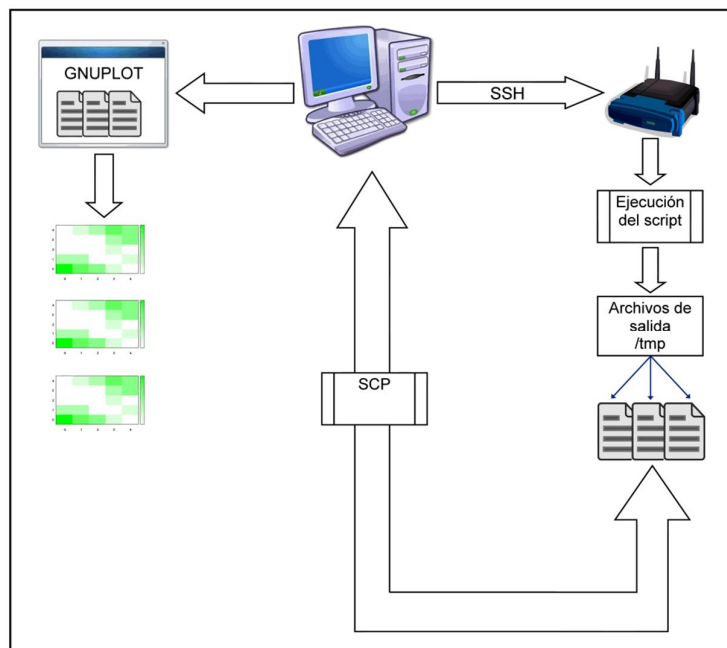


Fig. 3.4 Diagrama de interacción entre el enrutador y la computadora

3.6.1. Diseño del código del programa

El diseño del sistema para generar un mapa en el que se refleje los porcentajes de señal en cada ubicación se realiza en primera instancia como se mencionó en la sección 3.3 al

cambiar el firmware de fábrica por el firmware Openwrt. Luego se procedió a evaluar las funciones que presenta este nuevo firmware lo que nos mostró como resultado el comando específico para obtener información total de redes adyacentes como: SSID, calidad de la señal, dirección Mac, etc. El comando usado para este fin fue:

```
iwlist
```

El script de monitoreo de redes wireless (ver anexo C), posee 4 funciones principales: Scanning, revisa, calcula, salida.

Scanning: Esta es la función que realiza el monitoreo de las redes su sentencia principal es:

```
VAR=`iwlist wlo scan | grep "Cell\|SSID\|Quality" `
```

Esta sentencia realiza un escaneo de las redes y selecciona los parámetros Cell, SSID, Quality y los guarda en una variable. Lo siguiente que realiza esta función es guardar todos los datos contenidos en la variable en un archivo temporal y asignar a otra variable la dirección de ubicación del archivo temporal para poder acceder luego al mismo.

Revisa: Lo que realiza esta función es revisar el archivo temporal creado por la función scanning, ordenar los datos de

acuerdo al SSID y los guarda en otro archivo temporal.

Calcula: Esta función es la que nos proporciona el valor del porcentaje de la señal de las redes escaneadas por la función scanning y ordenadas por la función revisa, a través del valor de calidad de señal; es decir esta función trabaja sobre el segundo archivo temporal que fue creado en el script.

Salida: Esta función se encarga de guardar en un archivo creado en el directorio /tmp los datos de las redes ya ordenados por la función revisa y cada uno con su respectivo porcentaje de señal.

En la figura 3.5 se muestra la interacción de las funciones en el programa principal, lo que se realiza es lo siguiente: En el programa principal (main) se ingresa el intervalo de monitoreo que es las veces que se va a realizar la captura de los datos, a continuación se llama a la función scanning que se repetirá en un ciclo de acuerdo a la variable intervalo de monitoreo al terminar todos los ciclos la función scanning es detenida y se llama a la función revisa, seguidamente de la función calcula; luego se presentan los datos obtenidos, pero lo importante del programa es la creación de los archivos con los datos obtenidos para poderlos utilizar en el generador de gráficos.

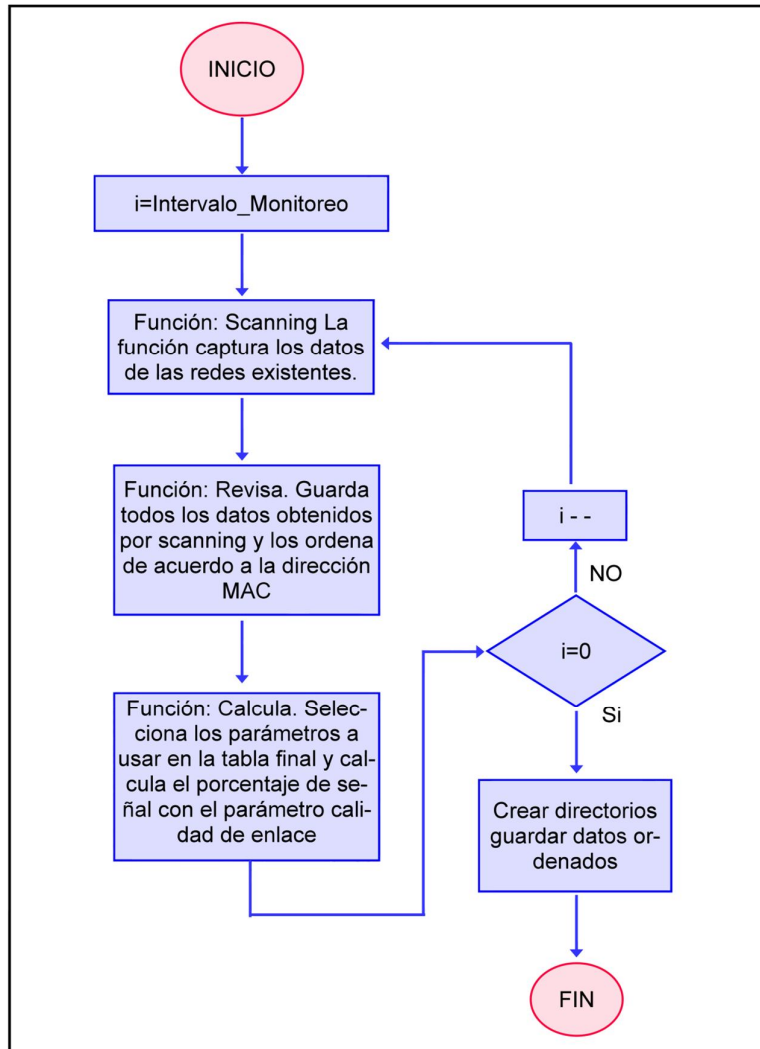


Fig. 3.5 Diagrama de flujo del script de monitoreo

3.6.2. Implementación de técnicas de programación en tiempo real.

El programa de monitoreo de redes hasta el momento es un programa monolítico que ejecuta un proceso a la vez lo que demora la respuesta entregada del sistema.

Como se mencionó en la sección 2.6 las ventajas de programar en tiempo real es mejorar el tiempo de respuesta del sistema. En nuestro caso hemos usado las sentencias wait y sleep para sincronizar un paralelismo de ejecución de procesos internos. Esto proporciona respuestas antes de que el monitoreo de redes termine y al final entrega los datos finales.

Al iniciar la ejecución del script se ejecuta el proceso 1 en primer plano que captura los datos de las redes existentes y los guarda en un archivo obteniendo la primera interacción de captura de datos, inmediatamente de finalizada la primera interacción se ejecuta el proceso 2 que realiza las mismas funciones del proceso 1 al finalizar la segunda interacción se ejecuta en segundo plano el proceso calculo 1 de forma paralela al proceso 3. El proceso de cálculo lee el archivo escrito por los anteriores procesos y calcula los porcentajes de señal recibidos este esquema es mostrado en la figura 3.6.

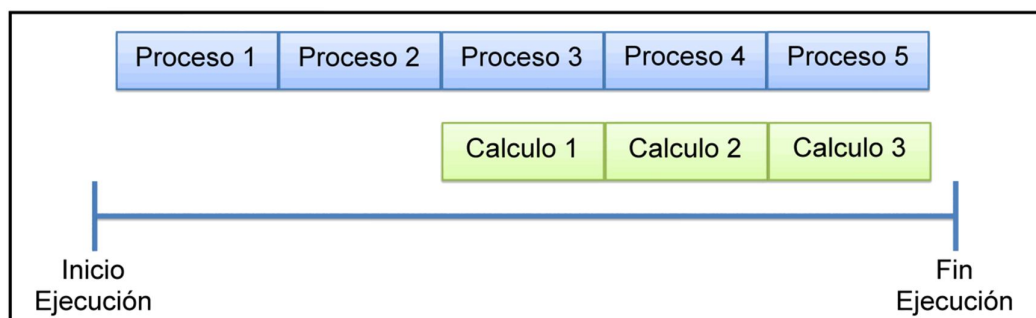


Fig. 3.6 Diagrama ejecución del script en real time

De esta forma se empiezan a obtener datos de información de las redes antes de terminar la ejecución del script y son considerados en tiempo real.

3.6.3. Ejecución del paquete de instalación

El programa realizado en lenguaje C que mediante la técnica de compilación cruzada se encuentra en un paquete de instalación para nuestro enrutador inalámbrico es enviado desde la computadora al dispositivo a través del comando:

```
scp Monitormap.ipkg.
```

Una vez que tenemos el programa en el enrutador ejecuta mediante la sentencia:

```
opkg install Monitormap.ipkg
```

El programa se ejecuta y crea los archivos con la información que sirve para realizar el mapa. Nos ubicamos en la carpeta /tmp con el comando:

```
cd /tmp
```

Copiamos los archivos a la computadora a través de una terminal con la sentencia:

```
scp -r /tmp:/192.168.1.3/datos ~
```

Esta instrucción copia la carpeta datos del directorio tmp a la carpeta /home del computador.

En una terminal se ejecuta: gnuplot. Y se copia las sentencias para la generación de los gráficos (ver anexo D).

CAPÍTULO 4

Pruebas y Resultados

En este capítulo se muestra los resultados a las pruebas realizadas para comprobar la validez del sistema al momento de obtener los datos. Para llegar a comprobar su utilidad se realizaron 3 pruebas con diferentes objetivos y forma de realización; en la primera prueba se mantuvo el enrutador de monitoreo fijo para evidenciar la capacidad de captar las redes; la segunda prueba fue realizada también con el enrutador fijo para mostrar la capacidad de procesamiento de los datos que posee el enrutador y la última prueba se realizó para mostrar el funcionamiento total del sistema y el cumplimiento de los objetivos del proyecto de investigación.

4.1. Monitoreo de Redes Existentes.

El objetivo de esta prueba es comprobar el funcionamiento del equipo y del script de monitoreo para la obtención de datos; así como observar experimentalmente las variaciones de los porcentajes de las señales ocasionados por los obstáculos atravesados. Para realizar esta prueba se ubicó el enrutador en un punto fijo durante cuarenta minutos y se ejecutó el script para que capture datos de las redes existentes en los alrededores. El enrutador que se mantuvo fijo durante la prueba es el que monitorea las redes Wi-Fi generadas por los demás enrutadores mostrados en la figura 4.1.



Fig. 4.1 Esquema de la disposición de los equipos al momento de tomar los datos

Cada 5 minutos se capturan 5 valores del porcentaje de señal de las redes existentes que se observa desde el enrutador de monitoreo y son procesados por el mismo para obtener como resultado el promedio del porcentaje señal en ese tiempo. .

Luego de transcurrido el tiempo de ejecución del programa, el enrutador muestra los datos presentados en la tabla IX en un archivo que se encuentra ubicado en el directorio /tmp. Se observan variaciones en los porcentajes de señal de las redes debido a los diferentes obstáculos atravesados por la señal y menor porcentaje de señal recibido de los enrutadores que se encontraban a una distancia mayor.

Tabla VII Porcentaje de la señal promedio vs Tiempo

Red	Tiempo transcurrido en minutos								
	0	5	10	15	20	25	30	35	40
HomeVipsVs	100	100	100	100	100	100	100	100	100
Enrutador1	68	65	68	68	67	67	68	68	68
Enrutador2	77	75	75	72	77	77	75	76	77
Celular1	47	47	45	47	47	45	45	47	45

La red que registra estar al 100% durante todos los intervalos de tiempo de la captura de datos es la red HomeVipsVs cuyo punto de acceso se encontraba a 3 metros del enrutador de monitoreo sin

obstáculos entre ellos. El celular1 es un Smartphone con la capacidad de vincularse a un enrutador y luego actuar como un punto de acceso para otros equipos. Se nota que la señal no es constante en tres de los equipos utilizados y se debe al ruido variable existente como personas u objetos en movimiento y puertas que fueron abiertas o cerradas luego de empezar la captura de datos.

Para observar lo que está sucediendo con las redes se realizó un gráfico con los ejes porcentaje de señal vs tiempo en el mismo se muestra la variación en el porcentaje de la señal recibida de la redes monitoreadas. Como se explico en la sección 2.1.2. Esto es ocasionado por fenómenos físicos como reflexión, difracción y multitrayectoria que disminuyen la señal.

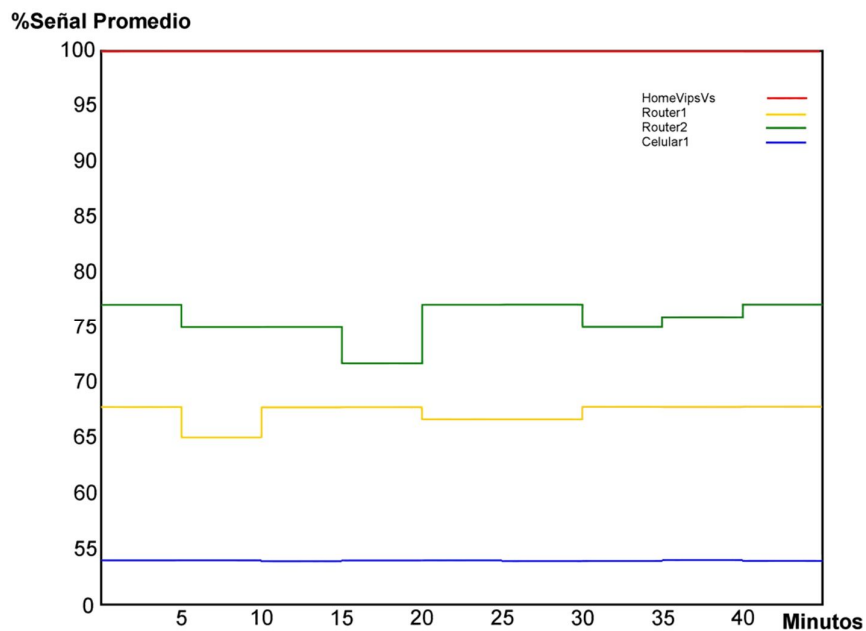


Fig. 4.2 Gráfico Porcentaje de la señal promedio vs tiempo

4.2. Recursos consumidos por el enrutamiento al realizar la captura de datos y procesamiento de datos.

El objetivo de esta prueba fue observar cuántos recursos se consumen en el enrutador al monitorear las redes: capturar, ordenar, clasificar los datos y guardarlos en archivos en el enrutador. Para esto fue necesario ingresar simultáneamente mediante la interfaz web de OpenWrt al enrutador y a través de línea de comandos en una terminal; esta medición se realizó en 8 minutos compartidos en 4 minutos antes de la ejecución y 4 minutos durante la ejecución del script.

Para observar lo que sucede con procesador el enrutador inalámbrico al capturar y clasificar los datos se utilizó la interfaz web de OpenWrt (Luci) en el apartado de gráficos en tiempo real en la pestaña carga (Load), donde se toman datos cada minuto automáticamente y se crean gráficos carga vs tiempo. La carga es valorada en porcentaje al igual que lo presenta el comando top de Unix.

El gráfico 4.3 fue tomado antes de la ejecución del script y el porcentaje máximo de uso del procesador era 38% y el promedio en los 4 minutos de la toma de datos fue 35 %.

Carga antes de la ejecución

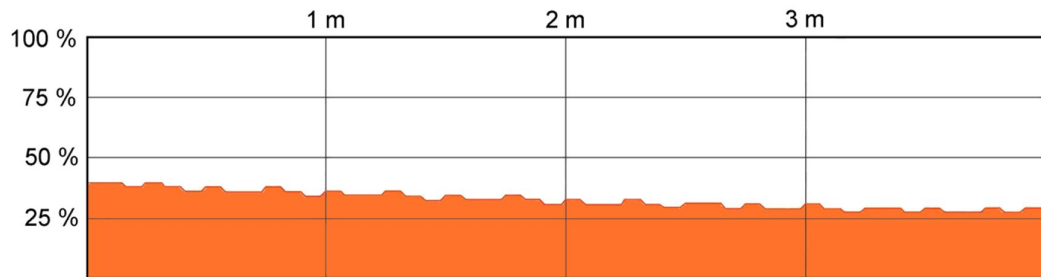


Fig. 4.3 Gráfico carga del procesador del enrutador antes de la ejecución

Carga durante la ejecución

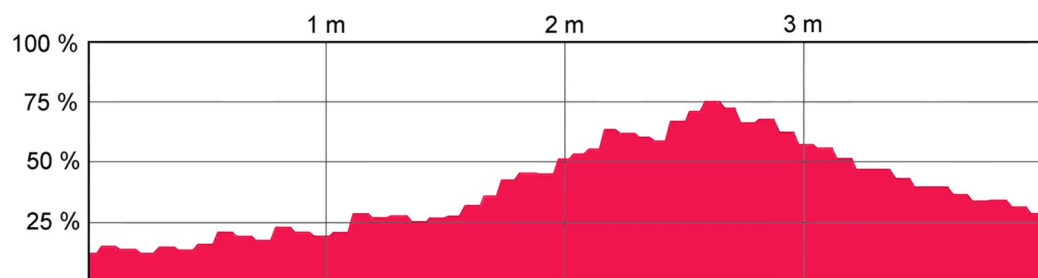


Fig. 4.4 Gráfico carga del procesador del enrutador durante la ejecución

El gráfico 4.4 se empezó a capturar 1 minuto antes de su ejecución para observar gráficamente la variación en el uso del cpu del enrutador; al ejecutarse se empieza a notar de forma muy rápida como aumenta la carga en el procesador, la carga máxima esta vez fue 75% y el promedio en los 4 minutos de carga fue 51%. A través de esta prueba se observó que el enrutador inalámbrico posee capacidad de procesamiento más allá del enrutamiento de paquetes; pero la misma es muy limitada y puede llegar a su límite de forma muy rápida

4.3. Obtención del mapa de porcentaje de señal para los datos capturados por el enrutador.

El objetivo de esta prueba es generar un mapa de porcentaje de señal con los datos obtenidos en cada cuadrante asignado a través del generador de gráficos gnuplot. Para esta prueba realizamos la captura de datos con el enrutador en 12 puntos diferentes para generar el mapa de porcentaje de señal; por lo que utilizamos la batería para poder movilizar el enrutador.

El esquema de cómo estuvieron ubicados los equipos se muestra en la figura 4.5; en esta prueba utilizamos 2 enrutadores inalámbricos Tp-Link, 2 enrutadores inalámbricos Linksys WRT160NL y como enrutador de monitoreo el enrutador Linksys WRT54GL. Los dos enrutadores Tp-Link tenían su firmware predeterminado, así también el enrutador Linksys designado en el esquema como enrutador 4 lo que demuestra que no existe ningún tipo de incompatibilidad entre el firmware original del enrutador con el firmware OpenWrt.



Fig. 4.5 Esquema de ubicación de los enrutadores dentro de la locación

Al ser el enrutador inalámbrico un dispositivo incapaz de manejar posicionamiento en el espacio decidimos resolver esta dificultad dividiendo el espacio total de la casa en 12 cuadrantes para así manejar un posicionamiento referencial como se muestra en la figura 4.6; lo ideal hubiera sido de alguna forma instalar en el enrutador algún sistema de posicionamiento como gps o dispositivos de localización de objetos pero eso se lo dejara como un trabajo a futuro.





2,0	2,1	 Router 1 2,2	2,3
1,0	1,1	1,2	1,3
 Router 4 0,0	0,1	0,2	 Router 3 0,3
 Router 2			

Fig. 4.6 Esquema de coordenadas asignadas para generar el mapa

Para obtener los datos en cada coordenada el enrutador inalámbrico de monitoreo fue movido por cada cuadrante capturando así los datos de porcentaje de señal de cada una de las redes. Estos datos son guardados en el directorio /tmp al cual debemos acceder para extraer la información necesaria para generar los mapas deseados. Los datos obtenidos por el enrutador en cada cuadrante son presentados a continuación en la tabla IX, en la que se muestra además el promedio de la señal mostrando el enrutador que tiene mayor cobertura.

Tabla VIII Porcentaje de la Señal vs ubicación

Ubicación (x, y)	Porcentaje de la señal			
	Enrutador 1	Enrutador 2	Enrutador 3	Enrutador 4
0,0	63	100	50	93
0,1	71	93	68	80
0,2	84	78	93	69
0,3	86	63	100	52
1,0	77	90	41	100
1,1	88	85	50	95
1,2	100	73	73	79
1,3	97	60	89	64
2,0	79	78	43	93
2,1	96	69	45	86
2,2	100	53	67	73
2,3	100	47	82	67
Promedio	86,75	74,08	66,41	79,25

Para generar los gráficos requeridos lo primero que realizamos es copiar el código de graficación que se encuentra en el anexo D; el mismo que nos muestra la opción para ingresar los datos columna a columna de la tabla. Al ingresar los datos dentro del graficador gnuplot se va a generar un mapa de porcentaje de señal por cada enrutador o AP existente; se muestra en intensidad de color en que cuadrante de

la locación el porcentaje de la señal es mayor. A mayor intensidad de color mayor es el porcentaje de la señal en ese punto.

Como se mostró en la tabla IX la red con mayor cobertura es la generada por el enrutador 1 al presentar un promedio de 86.75% de señal en los 12 cuadrantes. Esto se observa gráficamente en la figura 4.7; se observó que el enrutador 1 posee excelente señal en 8 cuadrantes de la casa. En el gráfico 4.8 se observa los porcentajes de señal para cada cuadrante generados por el enrutador 2; el promedio total del porcentaje de señal para esta red mostrado en la tabla IX es 74.08%.

En el gráfico 4.9 se ve los porcentajes de señal generados por el enrutador 3 que tiene como promedio de porcentaje de señal en los 12 cuadrantes 66,41% como se ve en la tabla IX. El gráfico 4.10 muestra los porcentajes de señal generados por el enrutador 4 con un promedio de porcentaje de señal del 79,25%, este valor nos dejó ver que el enrutador 4 tiene menor cobertura debido al lugar en el que se encuentra ubicado y a todos los obstáculos que la señal debe atravesar

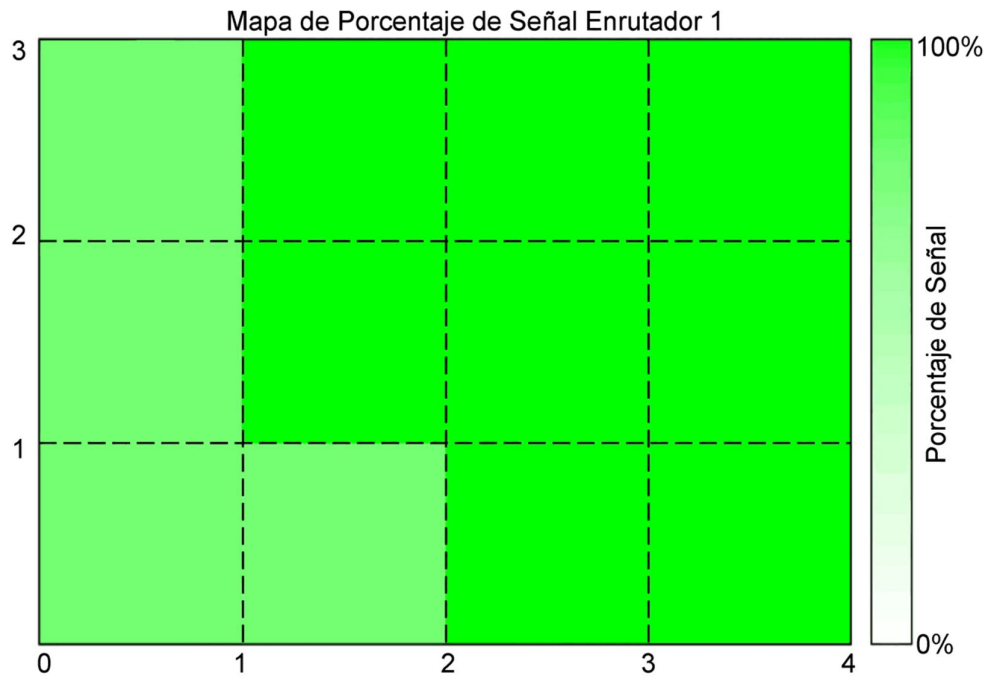


Fig. 4.7 Mapa de Porcentaje de señal Enrutador 1

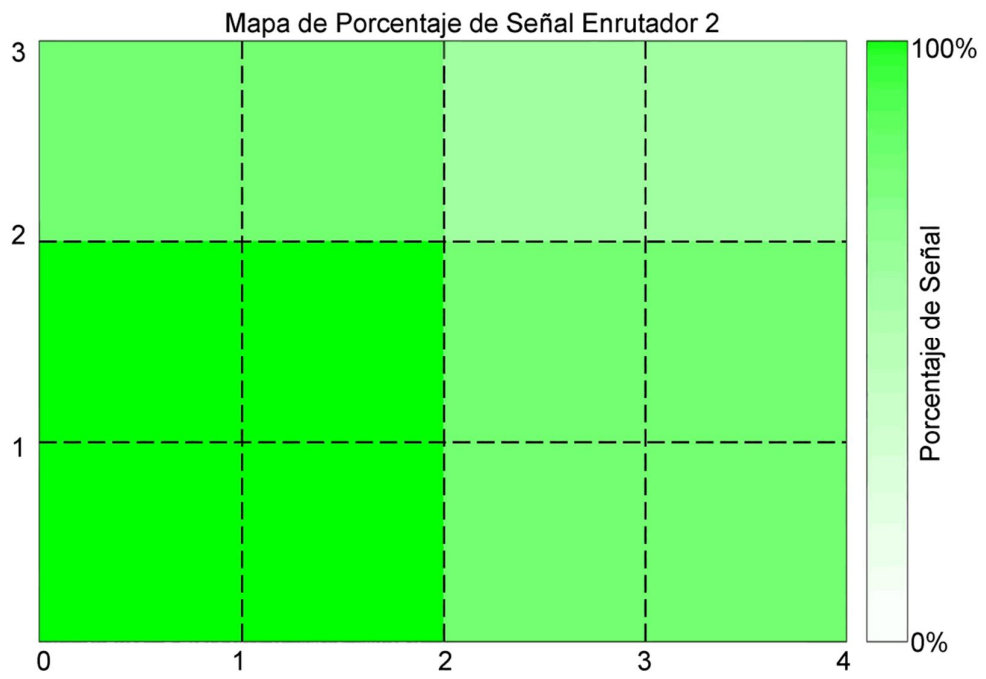


Fig. 4.8 Mapa de Porcentaje de señal Enrutador 2

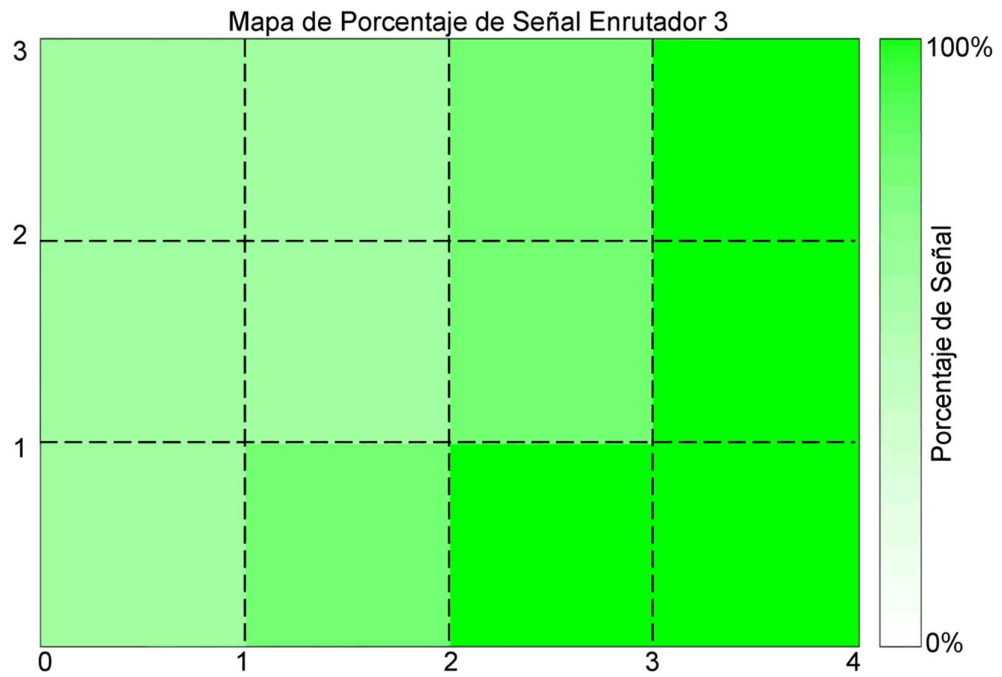


Fig. 4.9 Mapa de Porcentaje de señal Enrutador 3

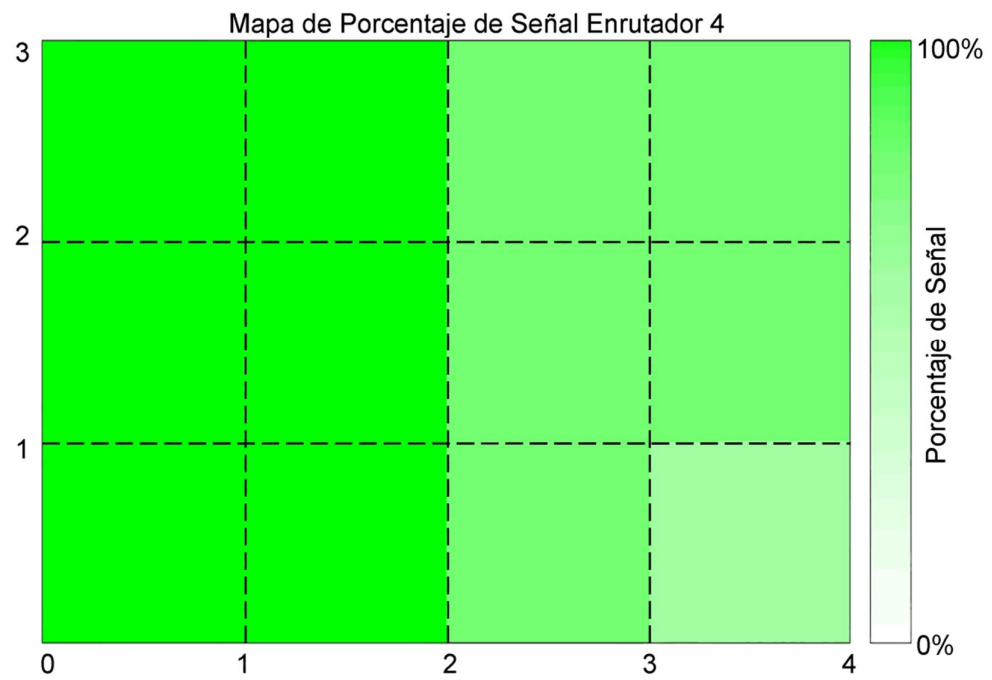


Fig. 4.10 Mapa de Porcentaje de señal Enrutador 4

CONCLUSIONES Y RECOMENDACIONES

Conclusiones:

- El proceso de cambio de firmware al enrutador inalámbrico es un proceso sencillo pero que necesita mucha atención y minuciosidad por la posibilidad de dejar inutilizable el dispositivo como fue sugerido en la sección 3.3 Instalación del firmware OpenWrt.
- Como se mostró en la sección 3.4.1. Instalación de paquetes; la implementación de nuevas funcionalidades para enrutadores inalámbricos que permiten cambios de firmware es un proceso sencillo; si el paquete de instalación para dicha funcionalidad ya existe y se encuentra en el repositorio de OpenWrt.
- Se pueden desarrollar nuevos paquetes de instalación si se tiene conocimiento de programación en lenguaje C y makefiles de Unix como se mostró en la sección 3.5 Compilación cruzada y el proceso de la misma en el anexo B logrando aportar al crecimiento del proyecto OpenWrt y aprovechar todos los recursos de hardware del dispositivo embebido (enrutador inalámbrico).
- En base a la prueba realizada en la sección 4.2 podemos concluir que el enrutador inalámbrico es un dispositivo en el que se pueden realizar muchas otras aplicaciones que no son necesariamente enrutamiento de datos; pero es necesario tener muy en cuenta sus fuertes

limitaciones en cuanto a procesamiento de datos y memoria de almacenamiento.

- Para graficar los datos se puede utilizar cualquier programa de software libre que nos permita insertar tablas de datos externos como se mostró en la sección 4.3.
- Como se mostró en la prueba de la sección 4.2, si se necesita monitorear redes por grandes intervalos de tiempo y aparte manejar un record histórico es mejor utilizar el enrutador inalámbrico sólo como un capturador de datos, pero que el procesamiento y almacenamiento sean realizados en una pc por las fuertes limitaciones que posee el dispositivo.

Recomendaciones:

- Al momento de realizar la instalación del firmware OpenWrt por ningún motivo desconectar al enrutador ya que esto provoca que resulte inutilizable como se vio en la sección 3.3.
- Es necesario leer bien la compatibilidad y documentación de un paquete de instalación antes de usarlo en el enrutador ya que muchos funcionan sólo con algún procesador en particular.

- Tener siempre en cuenta la capacidad de almacenamiento del enrutador inalámbrico fácilmente se puede exceder su capacidad y provocar fallos en su funcionamiento.

ANEXOS

ANEXO A

OBTENCIÓN DEL FIRMWARE OPENWRT

Lo primero que debemos hacer es obtener el firmware OpenWrt que corresponde a la marca y modelo de nuestro enrutador.

Hay dos formas de obtener el firmware de OpenWrt:

- Lo podemos descargar directamente desde la web de OpenWrt.

<http://downloads.openwrt.org>

- Podemos obtener el firmware mediante el repositorio de subversión (SVN) y nuestra propia compilación de la imagen del firmware con los paquetes que seleccionemos existentes en el repositorio.

TIPOS DE IMAGEN DEL FIRMWARE

Para nuestro enrutador nos encontramos con los siguientes archivos:

- <http://downloads.openwrt.org/backfire/10.03.1-rc4/ar71xx/openwrt-ar71xx-generic-wrt160nl-squashfs-factory.bin>
- <http://downloads.openwrt.org/backfire/10.03.1-rc4/ar71xx/openwrt-ar71xx-generic-wrt160nl-squashfs-sysupgrade.bin>
- <http://downloads.openwrt.org/backfire/10.03.1-rc4/ar71xx/openwrt-ar71xx-generic-wrt400n-jffs2-factory.bin>

- <http://downloads.openwrt.org/backfire/10.03.1-rc4/ar71xx/openwrt-ar71xx-generic-wrt400n-jffs2-sysupgrade.bin>

Debemos tener muy en cuenta el tipo de imagen que vamos a poner en el enrutador ya que cada una tiene su utilización específica:

- Los archivos que incluyen la palabra “factory” se utilizan para cambiar por primera vez el firmware del enrutador o para pasar de otro sistema operativo basado en Linux como por ejemplo DD-Wrt.
- Los archivos que incluyen la palabra “sysupgrade” se utilizan para realizar actualizaciones cuando ya tenemos instalado el firmware.
- Los archivos que incluyen la palabra “squashfs” tienen un sistema de archivo de solo lectura, por lo que no es recomendable si deseamos cambiar muchas opciones del firmware.
- Los archivos que incluyen la palabra “jffs2” se utilizan para poder realizar cambios en las opciones del firmware.

Para nuestra implementación vamos a usar el archivo de firmware que incluye las palabras “jff2-factory”, esto significa que tendremos la posibilidad de realizar sin problemas cambios en las opciones del firmware tales como: modificar archivos y configuraciones de OpenWrt para aprovechar a nuestro favor los recursos que el firmware nos ofrece.

ANEXO B

COMPILACIÓN CRUZADA DEL PROGRAMA HOLA MUNDO

Se debe escribir el código del programa en nuestra computadora con una versión Linux instalada. En nuestro caso, trabajamos con la distribución Ubuntu 12.04 LTS. A nivel de consola procedemos a crear un directorio para el programa HolaMundo y un subdirectorio src; ahí debemos colocar nuestro archivo extensión .c dentro del subdirectorio src:

```
/HolaMundo/src/HolaMundo.c:  
  
/*****  
* HolaMundo.c  
* Es el más simple programa escrito en lenguaje C  
*****/  
#include <stdio.h>  
int main (void)  
{  
    printf(" Hola Mundo \n\n ");  
    return 0;  
}
```

Luego debemos escribir un Makefile estándar de Unix para compilar el código; debemos colocarlo en el subdirectorio src:

```
/HolaMundo/src/Makefile:  
  
#Construye el ejecutable del programa HolaMundo al ejecutar el  
comando make  
HolaMundo: HolaMundo.o  
    $(CC) $(LDFLAGS) HolaMundo.o -o HolaMundo  
HolaMundo.o: HolaMundo.c  
    $(CC) $(CFLAGS) -c HolaMundo.c  
clean:  
    rm *.o HolaMundo
```

Ahora se debe realizar el proceso de compilación cruzada para poder hacer uso del ejecutable generado por la máquina Linux en el enrutador. Para realizar esto se debe tener un compilador y hacer uso de la herramienta SDK de OpenWrt, la misma que se puede descargar de <http://downloads.openwrt.org> dependiendo de la arquitectura de la computadora y de la versión de OpenWrt que está funcionando en el enrutador. El archivo que descargamos está comprimido por lo que debemos extraer los archivos contenidos en el mismo. Luego debemos entrar en el directorio SDK y copiar el directorio HolaMundo que se realizó con anterioridad. Luego debemos crear nuestro archivo de Makefile para OpenWrt en el directorio que se especifica a continuación:

```
/OpenWrt-SDK-Linux-i686-1/package/HolaMundo/Makefile:
#####
OpenWrt Makefile para programa HolaMundo en este programa se hacen
las configuraciones básicas que necesita el compilador para
relacionar el programa definido en el lenguaje C para así que la
nueva plataforma logre interpretar sin ningún inconveniente las
sentencias de código establecidas.
Básicamente se hace referencia al nombre del programa y la ruta
donde el mismo se encuentra.
#####

include $(TOPDIR) / rules.mk

PKG_NAME:=HolaMundo
PKG_RELEASE:=1

PKG_BUILD_DIR := $(BUILD_DIR) /$ (PKG_NAME)

include $ (INCLUDE_DIR) /package.mk

define Package/HolaMundo
    SECTION:=utils
    CATEGORY:=Utilities
    TITLE:=HolaMundo
    DESCRIPTION:=\
endef
```

```
define Build/Prepare
    mkdir -p $(PKG_BUILD_DIR)
    $(CP) ./src/* $(PKG_BUILD_DIR) /
endef

define Package/HolaMundo/install
    $(INSTALL_DIR) $(1)/bin
    $(INSTALL_BIN) $(PKG_BUILD_DIR)/helloworld $(1)/bin/
endef

$(eval $(call BuildPackage, HolaMundo))
```

Luego de esto ya podemos compilar el paquete HolaMundo, debemos ir al directorio del SDK desde una terminal y escribir:

```
make v=99
```

Al terminar el proceso el compilador, nuestro paquete HolaMundo se encuentra en el directorio bin/packages de la carpeta del SDK; este archivo es el ipkg ejecutable en el enrutador. Mediante la sentencia a continuación copiamos el paquete al enrutador:

```
scp HolaMundo_1_mipsel.ipkg
```

Cuando nuestro paquete ejecutable ya se encuentre en el enrutador procedemos instalamos mediante la sentencia:

```
opkg install HolaMundo_1_mipsel.ipkg
```

El ejecutable se ha instalado en el directorio /bin del enrutador, según nuestras instrucciones en el MaKefile para OpenWrt.

Luego de esto, debemos de simplemente correr nuestro código en el prompt de nuestro OpenWrt, llamándolo incluso sin necesidad de desplegar una ruta específica del mismo, ya que con solo el nombre del programa instalado se ejecuta el código inicialmente diseñado.

```
root@OpenWrt:~# helloworld  
Hola Mundo  
root@OpenWrt:~#
```

ANEXO C

CÓDIGO DEL SCRIPT

```
#!/bin/bash

# variables globales.
SCRIPT="Monitor_Wireless_CVR"
DIR="/tmp/${SCRIPT}"
FILE="${SCRIPT}.tmp"
FILEFIN="${SCRIPT}.txt"
FILECOMPARA="compara.txt"
FILEORD="ordena.txt"
#se declara una variable arreglo que toma los argumentos.
declare -a ARGUMENTOS=("$@")

#se declara la interfase de monitoreo.
Wireless_Interfase="wl0"

#comando de escaneo
comando_escaneo="iwlist wl0 scan | grep 'Cell\|SSID\|Quality'"
#echo $comando_escaneo

#intervalo default de monitoreo.
Intervalo_Monitoreo=60

#for (( i=0;i<${#ARGUMENTOS[@]};i++ )); do
#echo $i; echo ${#ARGUMENTOS[@]}

function Salida()
{
    # Muestra mensaje
    echo -e >&2 "$@\n"
    sleep 1
    #rm -f ${DIR}/* > /dev/null
    #sleep 1
    exit
}

function Scanning ()
{
    prefijo=$(date +%Y%m%d%H%M%S)
    echo "$prefijo" > ${DIR}/${FILE}

    VAR=`iwlist wl0 scan | grep "Cell\|SSID\|Quality"`
    echo "$VAR" >> ${DIR}/${FILE}

    #VAR1=`cat ${DIR}/${FILE}`
    #echo "$VAR1"
```

```

}

function CSQ ()
{

#echo -n "$prefijo;"
#se crea la variable que contiene la cantidad de registros de cada
arreglo
#varcount=`cat ${DIR}/${FILE} | grep -c "Address"`
#varcount=`expr $varcount`
#echo "$varcount"
varcount=`cat ${DIR}/${FILE} | grep -c "ESSID"`
#echo "la cantidad de registros a manejar es de $varcount"

direccionmac=`cat ${DIR}/${FILE} | grep "Address" | cut -d ' ' -f2`
#se asigna un arreglo a la variable arreglo vardireccion mac,
#se extrae solo la parte que se necesita del archivo tmp.

vardireccionmac=(`cat ${DIR}/${FILE} | grep "Address" | tr -s ' ' |
cut -d ' ' -f6`)

#echo ${#vardireccionmac[*]}
#echo "$vardireccionmac"

varssid=(`cat ${DIR}/${FILE} | grep "ESSID" | tr -s ' ' '@' | cut -d
' ' -f2`)
#echo "$varssid"
#echo "la cantidad de registros en el arreglo es ${#varssid[*]}"
#echo ${varssid[@]}
varssidmod>("${varssid[@]//@/ }")
#echo "la cantidad de registros en el arreglo modificado es
${#varssidmod[*]}"
#echo ${varssidmod[@]}

varquality=(`cat ${DIR}/${FILE} | grep "Quality" | tr -s ' ' | cut -d
' ' -f2 | cut -d ':' -f2 | cut -d '/' -f1`)
#echo "$varquality"

varsignal=(`cat ${DIR}/${FILE} | grep "Signal" | tr -s ' ' | cut -d
' ' -f4 | cut -d ':' -f2`)
#echo "$varsignal"

varnoise=(`cat ${DIR}/${FILE} | grep "Noise" | tr -s ' ' | cut -d '
' -f7 | cut -d ':' -f2`)
#echo "$varnoise"
#echo " $(date +%Y%m%d%H%M%S) "

varcomodin=`expr $varcount - 1`
#echo $varcomodin
for element in $(seq 0 $varcomodin)
do
varquality1=`expr ${varquality[$element]} \* 20`

echo -n "$ubicacion;"

```

```

echo -n "$prefijo;$varcount;"
echo
"${vardireccionmac[$element]};${varssidmod[$element]};$varquality1;${
varsignal[$element]};${varnoise[$element]};"

#almacena en el file fiin
echo -n "$ubicacion;" >> ${DIR}/${FILEFIN}
echo -n "$i" >> ${DIR}/${FILEFIN}
echo -n "$prefijo;$varcount;" >> ${DIR}/${FILEFIN}
echo
"${vardireccionmac[$element]};${varssidmod[$element]};$varquality1;${
varsignal[$element]};${varnoise[$element]};" >> ${DIR}/${FILEFIN}

#la asignacion del elemento del arreglo a una variable y luego la
multiplicacion.
#prueba="${varquality[$element]}"
#prueba=`expr $prueba \* 20`

#la multiplicacion * 20 pero hecho ya directamente y asignado a
variable
#prueba=`expr ${varquality[$element]} \* 20`
#echo "$prueba"
done
}

function revisa ()
{
if [ -e "${DIR}/${FILECOMPARA}" ]; then
rm ${DIR}/${FILECOMPARA} > /dev/null
fi
}

todossid=(`cat ${DIR}/${FILEFIN} | tr -s ' ' '@' | cut -d ';' -f5`)

#echo ${todossid[@]}
#echo "la cantidad de registros en el arreglo de revisa es
${#todossid[*]}"
todossidmod=("${todossid[@]//@/ }")
#echo "la cantidad de registros en el arreglo modificado es
${#todossidmod[*]}"

unset vararrayfin
vararrayfin[0]=${todossidmod[0]}
varcontador=${#todossidmod[@]}
#echo "$varcontador"
varconta=`expr $varcontador - 1`
for contador in $(seq 0 $varconta)
do
echo "${todossidmod[$contador]}"
for indiceconta in $(seq 0 ((${#vararrayfin[@]} - 1)))
do
#echo "${todossidmod[$contador]} ;
${vararrayfin[$indiceconta]}"
if [ "${todossidmod[$contador]}" =
"${vararrayfin[$indiceconta]}" ];

```



```

        then
            comodin1="1"
            vartmpvalor=${#vararrayfin[@]}
            indiceconta=`expr $vartmpvalor`
            #echo "escape con la cantidad de registros $indiceconta y el
comodin $comodin1"
            break
        else
            comodin1="0"
            #echo "no se encontro igualdad"
        fi

        #if ["${todossidmod[$contador]}" eq
"${vararrayfin[$indiceconta]}"]
            #indiceconta= ${(#vararrayfin[@]} - 1)
            #comodin1="1"
        #else
            #comodin1="0"
        #fi
#echo "$comodin1"
done
#echo "valor del $comodin1 por cada ciclo"

if [ "$comodin1" = "0" ]
then
#echo "nuevo registro ${todossidmod[$contador]}"
asignaconta=${#vararrayfin[@]}
prueba=`expr $asignaconta - 1`
#echo "cantidad inicial de registros del arreglo $asignaconta;
valor igual del ultimo registro ${vararrayfin[$prueba]}"
vararrayfin[$asignaconta]="${todossidmod[$contador]}"
asignacontal=${#vararrayfin[@]}
pruebal=`expr $asignacontal - 1`
#echo "cantidad final de registros del arreglo $asignacontal ;
valor igual del ultimo registro ${vararrayfin[$pruebal]}"
fi
done

#echo ${vararrayfin[@]} > ${DIR}/${FILECOMPARA}
#echo ${vararrayfin[@]}

for almacenatmp in $(seq 0 $asignacontal)
do
echo "${vararrayfin[almacenatmp]}" >> ${DIR}/${FILECOMPARA}
done
}

function calcula()
{
if [ -e "${DIR}/${FILEORD}" ]; then
rm ${DIR}/${FILEORD} > /dev/null
fi

```

```

varc1=(`cat ${DIR}/${FILEFIN} | cut -d ';' -f1`)
varc2=(`cat ${DIR}/${FILEFIN} | cut -d ';' -f2`)
varc3=(`cat ${DIR}/${FILEFIN} | cut -d ';' -f3`)
varc4=(`cat ${DIR}/${FILEFIN} | cut -d ';' -f4`)

varc5=(`cat ${DIR}/${FILEFIN} | tr -s ' ' '@' | cut -d ';' -f5`)
varc5mod=("${varc5[@]//@/ }")

varc6=(`cat ${DIR}/${FILEFIN} | cut -d ';' -f6`)
varc7=(`cat ${DIR}/${FILEFIN} | cut -d ';' -f7`)
varc8=(`cat ${DIR}/${FILEFIN} | cut -d ';' -f8`)

varst=(`cat ${DIR}/${FILECOMPARA} | tr -s ' ' '@'`)
varstmod=("${varst[@]//@/ }")

#echo "el tamaño de los arreglos es de ${#varc1[@]}, ${#varc2[@]},
${#varc3[@]}, ${#varc4[@]}, ${#varc5mod[@]}, ${#varc6[@]},
${#varc7[@]}, ${
#echo "tamaño de muestra de ssid ${#varstmod[@]}"
#echo ${varc5mod[@]}
echo
echo -n "registro unicos SSID: "
echo ${varstmod[@]}
echo

varc5conta=${#varc5mod[@]}
varstconta=${#varstmod[@]}
varc5contal=`expr $varc5conta - 1`
varstcontal=`expr $varstconta - 1`
varminimo=0
for ivarst in $(seq 0 $varstcontal)
do
echo "${varstmod[$ivarst]}"
varsumatp=0
varindtp=0
for ivarc5 in $(seq 0 $varc5contal)
do
do
if [ "${varc5mod[$ivarc5]}" = "${varstmod[$ivarst]}" ];
then
varsumatp=`expr $varsumatp + ${varc6[$ivarc5]}`
varindtp=`expr $varindtp + 1`
echo "${varc5mod[$ivarc5]}; ${varc6[$ivarc5]}"
fi
done

varprompto=`expr $varsumatp / $varindtp`
echo "$varindtp $varsumatp promedio es $varprompto"
echo "$varprompto;${varstmod[$ivarst]}" >> ${DIR}/${FILEORD}
echo
if (( varprompto > varminimo ))
then
#echo "el mayor es $varprompto"

```

```

varminimo=`expr $varpromto`
varmaxssid=${varstmod[$ivarst]}
fi
arrvarprom[ivarst]=`expr $varpromto`
done
#echo "La red mas optima de conexion es $varmaxssid con un promedio
de Signal Quality de $varminimo"

varsordena=(`cat ${DIR}/${FILEORD} | tr -s ' ' '@' | sort -nr | cut
-d ';' -f1`)
varsordenamod=("${varsordena[@]//@/ }")
varsordenal=(`cat ${DIR}/${FILEORD} | tr -s ' ' '@' | sort -nr | cut
-d ';' -f2`)
varsordenalmod=("${varsordenal[@]//@/ }")

varordenaconta=${#varsordenamod[@]}
varordenacontal=`expr $varordenaconta - 1`
for cuentate in $(seq 0 $varordenacontal)
do
    cuentatel=`expr $cuentate + 1`
    echo "El $cuentatel valor con un promedio de calidad de senal de
${varsordenamod[cuentate]} es ${varsordenalmod[cuentate]}"
#echo "${varsordenamod[cuentate]} ${varsordenamod1[cuentate]}"
done

#varsordena=`cat ${DIR}/${FILEORD}`
#echo "$varsordena"
#varordenal=`cat ${DIR}/${FILEORD} | sort -n`
#echo "$varordenal"

#echo ${arrvarprom[*]} > ${DIR}/${FILEORD}
#arrvarpromor=(`cat ${DIR}/${FILEORD} | uniq | sort`)
#arrvarpromor=(`echo "${arrvarprom[*]}" | uniq | sort`)
#echo "${arrvarpromor[*]}"
}

#Definiendo la funcion Main
function Main ()
{
#Definiendo la salida a traves de Ctrl+C
trap '{ echo -e "\nControl-C trap. Saliendo"; Salida; }' INT #traps
Ctrl-C
#echo $Wireless_Interfase
i=1
while [ "$i" -le "60" ]
do
do
Scanning
echo "$i"
CSQ
# echo " "
i=`expr $i + 1`
sleep $Intervalo_Monitoreo
done
done
revisa

```

```

calcula
}

if [ "${#ARGUMENTOS[@]}" != "0" ];
then echo "El script $SCRIPT no corre con ${#ARGUMENTOS[@]}
argumentos"
exit
fi

echo -n "Ingrese el nombre de la posicion: (Ej. 0)"
read ubicacion
if [ -z "$ubicacion" ]
then
ubicacion=0
echo "La posicion inicial es $ubicacion"
else
echo "La posicion inicial es $ubicacion"
fi

#echo ${ARGUMENTOS[$i]}
#done

# Verifica si existe el directorio
if [ -d "${DIR}" ]; then
rm -f ${DIR}/${FILE} > /dev/null
cd ${DIR}
Main
else
echo "Directory no existe"
echo "Creando directorio"
mkdir -p ${DIR}
cd ${DIR}
Main
fi

exit
#VAR=`iwlist wlo scan | grep "Cell\|SSID\|Quality"`
#echo "$VAR"
#for a in "$VAR"
#do
# echo "$a"
#done

```

ANEXO D

GNU PLOT SAMPLE GRÁFICO

```
# set terminal pngcairo transparent enhanced font "arial,10"
fontscale 1.0 size 450, 350

# set output 'heatmaps.2.png'

unset key

set view map

set xtics border in scale 0,0 mirror norotate offset character 0,
0, 0 autojustify

set ytics border in scale 0,0 mirror norotate offset character 0,
0, 0 autojustify

set ztics border in scale 0,0 nomirror norotate offset character 0,
0, 0 autojustify

set nocbtics

set rtics axis in scale 0,0 nomirror norotate offset character 0,
0, 0 autojustify

set title "Mapa de Porcentaje de Señal"

set xrange [ -0.500000 : 2.500000 ] noreverse nowriteback

set yrange [ -0.500000 : 3.500000 ] noreverse nowriteback

set cblabel "Porcentaje"

set cbrange [ 0.000000 : 5.000000 ] noreverse nowriteback

set palette rgbformulae -7, 2, -7

plot '-' using 1:2:3 with image
```

BIBLIOGRAFÍA

[1] Pew Internet, Smith Aaron – Research Specialist, (2010, Julio), “Mobile Access 2010”, The current state of wireless internet use [Online], Disponible en:

http://www.pewinternet.org/~media/Files/Reports/2010/PIP_Mobile_Access_2010.pdf

[2] Wikipedia.org – Estados Unidos, (2013, Junio), “Router”, Funcionamiento y Encaminadores inalámbricos [Online], Disponible en:

https://es.wikipedia.org/wiki/Router#Encaminadores_inal.C3.A1mbricos

[3] Wikipedia.org – Estados Unidos, (2013, Junio), “Wi-Fi”, Historia y desventajas [Online], Disponible en:

<http://es.wikipedia.org/wiki/Wi-Fi>

[4] Wikipedia.org – Estados Unidos, (2013, Julio), “Red Inalámbrica”, Características y Aplicaciones [Online], Disponible en:

http://es.wikipedia.org/wiki/Red_inal%C3%A1mbrica

[5] Wikipedia.org – Estados Unidos, (2013, Abril), “Red de área local inalámbrica”, Seguridad y velocidad [Online], Disponible en:

http://es.wikipedia.org/wiki/Red_de_%C3%A1rea_local_inal%C3%A1mbrica

[6] Kioskea.net – España, (2013, Junio), “Propagación de las ondas de radio (802.11)”, Absorción, reflexión y propiedades de los medios [Online], Disponible en:

<http://es.kioskea.net/contents/819-propagacion-de-las-ondas-de-radio-802-11>

[7] Wikipedia.org – Estados Unidos, (2013, Junio), “Refracción”, Refracción de ondas de radio [Online], Disponible en:

<http://es.wikipedia.org/wiki/Refracci%C3%B3n>

[8] Ecured.cu – Cuba, (2013, Julio), “Ondas de Radio”, Propagación de ondas de radio [Online], Disponible en:

http://www.ecured.cu/index.php/Ondas_de_radio

[9] Wikipedia.org – Estados Unidos, (2013, Marzo), “Sistema Embebido”, Arquitectura básica [Online], Disponible en:

http://es.wikipedia.org/wiki/Sistema_embebido

[10] Gnuplot.info – Estados Unidos, (2013, Abril), “GNU PLOT”, Demo Gallery y Heat Maps [Online], Disponible en:
<http://gnuplot.sourceforge.net/demo/heatmaps.html>

[11] Openwrt.org – Estados Unidos, (2013, Abril), “OPENWRT”, Documentation y downloads [Online], Disponible en:
<https://openwrt.org>

[12] Cisco CCNA – Estados Unidos, (2013, Julio), “CCNA EXPLORATION 4 Modulo 2”, Capitulo 1, Introducción al enrutamiento y envío de paquetes.

[13] ESPOL – Ecuador, Ph. D. Daniel Ochoa, (2012, Mayo), Sistemas Operativos, “Sistemas de tiempo Real”

[14] U. N. N. E. – Argentina, Dr. David Luis la Red Martínez, (2010, Julio), Sistemas Operativos, “Monografía de Redes e Interconexión de Redes”, Tipos de Redes [Online], Disponible en:
<http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/MonogSO/REDES02.htm>

[15] Solred.com – Argentina, Miguel Ricardo Ghezzi, (2008, Octubre), “Propagación de las Ondas de radio”, Formas de propagación [Online], Disponible en:
<http://www.solred.com.ar/lu6etj/tecnicos/handbook/propagacion/propagacion.htm>