

**Universidad de la República
Facultad de Ingeniería**

Proyecto Yacaré

Analizador de RF para Plan Ceibal



Martín Irazoqui Ahlig
Juan Andrés Rodríguez Palombo
Hernán Susunday D'Ambrosio

Noviembre 2008

Tutor: Ing. Eduardo Cota

Agradecimientos

A nuestras familias y amigos, por su apoyo constante en todo momento a lo largo del proyecto y del proceso de formación de todos estos años.

A nuestro tutor, el Ing. Eduardo Cota.

Al Plan Ceibal, en particular a la Ing. Fiorella Haim, Ricardo Medina y a todo el departamento de conectividad.

Resumen

Yacaré es una aplicación diseñada para el Plan Ceibal. Es una herramienta que integra escaneos de redes WiFi y Mesh orientada a los relevamientos. Para el posicionamiento geográfico de los puntos escaneados utilizamos un receptor GPS, de esta forma la información puede ser desplegada en mapas utilizando el sistema de información geográfico de Ceibal.

Para el desarrollo del proyecto se estudiaron a fondo el estándar de la IEEE 802.11b/g (WiFi) y el draft 802.11s (Mesh) para comprender así el funcionamiento de las capas inferiores y la comunicación con las interfaces de red. También se estudiaron los mensajes intercambiados con el dispositivo GPS para la adquisición de coordenadas y funciones de control.

La aplicación fue desarrollada con una estructura modular que permite adaptarse fácilmente a nuevas tecnologías. Se diseñó una interfaz gráfica amigable orientada al usuario, en este caso los técnicos de Ceibal, desde la cual se pueden manejar todas las funciones relativas a la configuración de interfaces, adquisiciones en relevamientos, manejo de datos y despliegue en pantalla de la información en tiempo real.

Visto el crecimiento de las redes wireless en ciudades y el desarrollo vertiginoso del Plan Ceibal a nivel nacional resulta prioritario este tipo de herramienta para la planificación y optimización de los sitios. Dada la novedad tecnológica que son las redes Mesh, formadas por las laptop XO entregadas por el Plan Ceibal, surge también la necesidad de contar con una aplicación que analice estadísticamente la comunicación de estos equipos con los puntos de acceso wireless.

Contenido

1.	Introducción	3
1-1	Plan Ceibal	3
1-1.1	Contexto	3
1-1.2	Necesidad Técnica de Plan Ceibal	4
1-2	Proyecto Yacaré - Analizador de RF para Plan Ceibal	5
1-2.1	Presentación	5
1-2.2	Objetivos del Proyecto Yacaré	6
1-2.3	Conclusiones generales	6
1-3	Estructura de la documentación	7
2.	Desarrollo de la Aplicación	8
2-1	Introducción	8
2-2	Elección del lenguaje y estructura de programación	8
2-3	Estructura de Yacaré	9
2-4	Requerimientos de la aplicación	10
3.	Escaneo de redes 802.11b/g y pruebas de ancho de banda	11
3-1	Introducción a 802.11	12
3-1.1	Definiciones y entidades que participan de 802.11	15
3-1.2	Acceso al medio	17
3-2	Escaneo pasivo y activo	18
3-2.1	Escaneo activo	18
3-2.1.1	Probe Request	20
3-2.1.2	Probe Response	20
3-2.1.3	Escaneo activo y tecnología MIMO	21
3-2.2	Escaneo pasivo	22
3-3	Relevamientos previos de Ceibal y requerimientos	23
3-3.1	Wireless-Tools: el comando iwlist	23
3-3.2	Módulo Python-WiFi	24
3-3.3	Comunicación con el hardware: iocctl	24
3-4	Módulo WiFi	26
3-4.1	Objetivos y funcionamiento	26
3-4.2	Escaneos	26
3-4.3	Parámetros de las adquisiciones	27
3-4.4	Flujo de eventos en una adquisición	27
3-5	Pruebas de ancho de banda	29
3-5.1	Antecedentes y pruebas con Iperf	29
3-5.2	Capturas y eficiencia en el uso de la red	30
3-5.3	Funcionamiento	31
4.	Búsqueda de portales Mesh	33
4-1	Introducción al draft 802.11s	34
4-2	Ruteo en capa de enlace	35
4-3	Active Antenna	37
4-4	Mesh Portal Points	39
4-4.1	Las XO como Mesh Portal Points	39
4-4.2	Búsqueda de Portales	39
4-4.3	El módulo pydhcplib	40
4-5	Módulo Mesh	41
4-5.1	Objetivos y funcionamiento	41
4-5.2	Búsqueda de portales	41

5.	Posicionamiento geográfico por GPS	43
5-1	Introducción al GPS	44
5-2	El sistema GPS	45
5-2.1	Arquitectura del sistema GPS	45
5-2.2	Principios de funcionamiento del sistema GPS	45
5-3	Sistema de referencia de coordenadas	46
5-3.1	Sistemas de referencia ROU-USAMS	46
5-4	Sistema de información geográfico de Ceibal	47
5-5	Receptor GPS	48
5-5.1	Elección del dispositivo	48
5-5.2	GPS 18 USB de Garmin	48
5-5.3	Funcionamiento del receptor GPS	52
5-6	Módulo GPS	54
5-6.1	GPSD	54
5-6.2	Formato de los mensajes	55
6.	Interfaz gráfica de la aplicación	58
6-1	Diseño de la interfaz	59
6-1.1	Barra de estados GPS	60
6-1.2	Solapa WiFi	61
6-1.3	Solapa Adquisiciones	65
6-1.4	Solapa Log	69
6-1.5	Ventana de Opciones	70
7.	Resultados	72
7-1	Etapa de desarrollo	72
7-1.1	Relevamiento en Durazno	72
7-1.2	Comparativo de antenas	73
7-2	Etapa de pruebas de la aplicación	75
7-2.1	Relevamiento en Mendoza	76
7-2.2	Relevamiento en Villa Cardal	77
7-2.3	Relevamiento en Sarandí Grande	80
7-3	Desempeño de la aplicación	91
8.	Conclusiones	93
9.	Anexos	95
9-1	802.11 Capa Física: Modulación y Codificación	95
9-2	802.11 Acceso al medio	101
9-3	Escaneos: Mensajes Intercambiados	103
9-4	Protocolo DHCP	107
9-5	Sistemas de Referencia	110
9-6	GPS Diferencial	113
9-7	Contenido del CD	114
10.	Referencias	116

Acrónimos

AP	Access Point
BSA	Basic Service Area
BSS	Basic Service Set.
BSSID	Basic Service Set Identifier
CTS	Clear To Send
DA	Destination Address
DGPS	Differential GPS
DHCP	Dynamic Host Configuration Protocol
DS	Distribution System
DSSS	Direct Sequence Spread Spectrum
EGNOS	European Geostationary Navigation Overlay Service
ESA	Extended Service Area
ESS	Extended Service Set
ESSID	Extended Service Set Identifier
FHSS	Frequency Hopping Spread Spectrum
FTP	File Transfer Protocol
GIS	Geographic Information System
GPS	Global Position System
HWMP	Hybrid Wireless Mesh Protocol
IBSS	Independent Basic Service Set
IEEE	Institute of Electrical and Electronics Engineers
IR	Infrared
ISM	Industrial, Scientific and Medical
LAN	Local Area Network
LLC	Logical Link Layer

MAC	Medium Access Control
MAP	Mesh Access Point
MP	Mesh Point
MPP	Mesh Portal Point
MSAS	Multi-functional Satellite Augmentation System
NAT	Network Address Translation
NMEA	National Marine Electronics Association
OFDM	Orthogonal Frequency Division Multiplexing
PHY	Physical Layer
PND	Portable Navigation Device
QAM	Quadrature Amplitude Modulation
RF	Radio Frequency
RTS	Request To Send
SNR	Signal Noise Rate
SSH	Secure Shell Transfer Protocol
STA	Station
TCP	Transmission Control Protocol
TPC	Transmit Power Control
UDP	User Datagram Protocol
USERE	User Estimated Range Error
WAAS	Wide Area Augmentation System
WGS84	World Geodesic System 1984
WLAN	Wireless LAN
WM	Wireless Medium

1. Introducción

1- 1 Plan Ceibal



1- 1.1 Contexto

El Plan Ceibal es un programa creado por el Gobierno de la República en Diciembre de 2006. Este programa tiene como objetivo central promover la inclusión digital con el fin de disminuir la brecha existente respecto a otros países y de los ciudadanos del país entre sí, de manera de posibilitar un mayor y mejor acceso a la educación y a la cultura.

Se define que entre 2007 y 2009 todos los niños y maestros de escuelas públicas recibirán una computadora portátil, así como brindar al colectivo docente la capacitación, los materiales, las orientaciones y el apoyo necesario para lograr los objetivos mencionados anteriormente.

Para implementar el programa el Plan Ceibal comienza en mayo del 2007 una prueba piloto en la escuela pública N° 24 de Villa Cardal, Florida. Se entregaron equipos XO donados por OLPC y diseñadas por el Instituto Tecnológico de Massachussets (MIT). Luego de una larga etapa de pruebas se realizó un llamado a licitación para la compra de 100.000 laptops.

El Plan Ceibal accedió finalmente a la compra de laptops XO. Éstas poseen acceso a redes inalámbrica WiFi y son capaces de relacionarse una máquina con otra sin necesidad de equipos adicionales mediante la formación de redes Mesh. Estas características permiten el trabajo en red, favoreciendo la interactividad entre los niños y la co-construcción del aprendizaje así como el acceso a contenidos a través de Internet.

Nuevos llamados a licitación fueron surgiendo para brindar soluciones de conectividad inalámbrica WiFi y servidores en las escuelas en los departamentos de Colonia, Durazno, Soriano, Río Negro, Paysandú, Salto y Artigas.

Desde la primera experiencia piloto en mayo de 2007, el Plan Ceibal se ha extendido a más del 50% del territorio nacional a mediados de 2008, cubriendo Florida, Flores, Colonia, Artigas, Salto, Paysandú, Río Negro, Durazno, Soriano, Tacuarembó, Rivera y Cerro Largo.

Ya hay instalados planes piloto en Montevideo y Canelones, y se prevé para el año 2009 que el Plan se extienda a todo el territorio nacional.

1 -1.2 Necesidad técnica de Plan Ceibal

El sistema de conectividad de Ceibal, consiste en brindar a los usuarios conexión a Internet a través de Access Points WiFi ubicados en las escuelas y en diferentes puntos de las localidades a donde se ha extendido.

El desarrollo del Plan Ceibal se ha dado de forma vertiginosa en los últimos 18 meses, se encuentra en plena fase de crecimiento pero a su vez el mantenimiento y optimización de las instalaciones ya existentes se realiza en paralelo.

Para facilitar estas últimas tareas, aparece como una necesidad el contar con una herramienta que permita relevar los datos *Señal WiFi - Posición*. Dicha información, es de mucha utilidad para los técnicos del Plan Ceibal ya que luego de relevada puede ser presentada sobre un mapa de la localidad y desde ahí planificar los pasos a seguir para lograr los objetivos de cobertura.

Dadas las propiedades de las redes Mesh, las laptops XO permiten a los usuarios la conexión a Internet sin necesidad de estar en zona de cobertura de los Access Points de Ceibal. Es decir, estas XO pueden lograr el acceso a Internet mediante otras máquinas XO que sí estén en zona de cobertura, éstas funcionan como portales Mesh de las primeras.

Por lo tanto, el estudio referido a las redes Mesh, busca conocer en mejor medida cuanto puede extenderse el acceso a Internet a través de la misma. Es de interés para los técnicos de Plan Ceibal, contar con una herramienta que permita relevar *Portales Mesh - Posición*.

1- 2 Proyecto Yacaré - Analizador de RF para Plan Ceibal

1- 2.1 Presentación

Dadas las necesidades planteadas anteriormente, surge el *Proyecto Yacaré*.

Yacaré es una herramienta que unifica los relevamientos, permite analizar señales de redes WiFi y realiza simultáneamente la búsqueda de portales de redes Mesh en un área metropolitana, mientras que el posicionamiento geográfico de los relevamientos es realizado con la ayuda de un GPS.

Posteriormente, la información es guardada en una base de datos y a su vez puede ser exportada en un formato conocido, facilitando su presentación en forma de capas sobre el sistema de información geográfico de Ceibal.

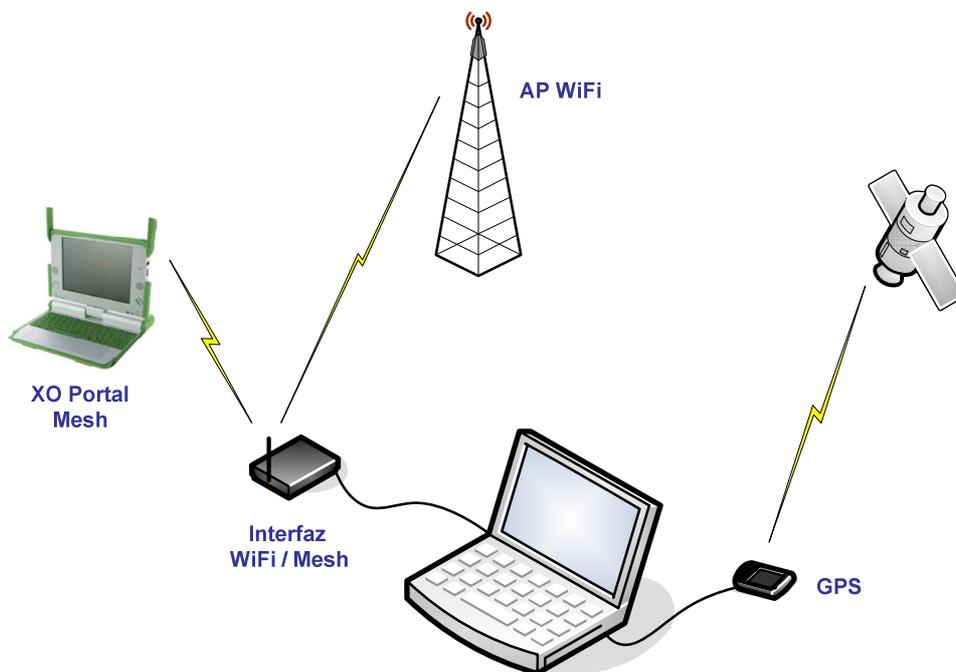


Fig 1.1: Esquema general

1- 2.2 Objetivos del Proyecto Yacaré

Al iniciar el desarrollo de la aplicación, trazamos seis condiciones de éxito del proyecto. Éstas son:

- Adquisición de señales inalámbricas de redes WiFi (802.11b/g) y Mesh (802.11s)
- Procesamiento de los datos adquiridos, y a partir de estos generar una base de datos
- Dar la posibilidad a los técnicos del Plan Ceibal de extraer los resultados obtenidos de las mediciones y desplegarlos en mapas, utilizando el software del sistema de información geográfico de Ceibal
- Crear una interfaz gráfica de fácil manejo para los usuarios
- Definir las interfaces de la aplicación de manera de poder adaptarse fácilmente a nuevas tecnologías y a cambios propuestos por el cliente
- A la fecha de la entrega debe haber pasado por una etapa de testing de aceptación por el cliente.

1- 2.3 Conclusiones generales

Al finalizar el proyecto podemos decir que se cumplieron con las condiciones de éxito.

Logramos crear una aplicación con una interfaz gráfica amigable y completa que integra la comunicación con los dispositivos de Mesh, WiFi y GPS, teniendo en cuenta los requerimientos del Plan Ceibal planteados al inicio del proyecto y otros que fueron sugiriendo en el desarrollo del mismo.

Se realizaron pruebas sobre la aplicación que nos permiten asegurar que Yacaré es una herramienta que cumple satisfactoriamente con los objetivos trazados.

1- 3 Estructura de la documentación

De aquí en más se desarrollaran las diferentes áreas de estudio que se abarcaron durante el proyecto. Es decir, las herramientas de programación, el estándar 802.11, las redes Mesh, el posicionamiento geográfico por GPS, su integración con la interfaz gráfica para el usuario, la etapa de pruebas y conclusiones finales sobre el Proyecto Yacaré.

Iniciaremos el desarrollo centrándonos en la elección del lenguaje de programación así como las definiciones en cuanto a la estructura de la aplicación.

Luego, se encontrarán los capítulos dedicados a los diferentes dispositivos utilizados, éstos contienen un estudio teórico inicial enfocado a los requerimientos dentro de Yacaré y luego una descripción del módulo que se diseñó para controlarlo dentro de la aplicación del sistema.

Mas adelante, dedicaremos un capítulo al módulo integrador de la aplicación que corresponde a la interfaz gráfica. Allí describiremos a fondo las funcionalidades de Yacaré y cómo se realiza la interacción entre los submódulos.

Finalmente se encontrará la etapa de pruebas, donde se describe los resultados obtenidos con Yacaré luego de los relevamientos piloto, así como las pruebas realizadas sobre el sistema operativo para estimar el consumo de recursos por parte de la aplicación.

En las conclusiones sobre el proyecto, nos enfocaremos en cada una de las condiciones de éxito así como en las diferentes áreas que abarca el Proyecto Yacaré.

2. Desarrollo de la aplicación

2- 1 Introducción

En este capítulo nos enfocaremos en el desarrollo de la aplicación integrada, la elección de Python como lenguaje de programación y la estructura de desarrollo utilizada. El área de programación fue una parte muy importante en este proyecto por lo que le dedicamos una etapa previa al desarrollo para diseñar la arquitectura del programa. También incluimos una sección referente a los requerimientos de hardware y software para la óptima ejecución de Yacaré.

2- 2 Elección del lenguaje y estructura de programación

A la hora de elegir el lenguaje de programación pusimos sobre la mesa los factores que serían determinantes. La posibilidad de interacción con los dispositivos con los que íbamos a trabajar era prioritaria, por lo que necesitábamos un lenguaje que disponga de librerías para esta tarea. La facilidad con la que se aprende el manejo de las estructuras para programar así como la disponibilidad de manuales de ayuda para desarrolladores fue determinante para la elección dado que nuestro equipo es principiante en el área del desarrollo de software

Python es un lenguaje de programación dinámico orientado a objetos que puede ser utilizado de diversas maneras a la hora de desarrollar software. Con su versatilidad, ofrece al programador un sólido soporte en la interacción con dispositivos, herramientas y otros lenguajes ya que cuenta con librerías y manuales extensos que permiten aprender el manejo en muy poco tiempo.

Por todo esto, Python fue el lenguaje elegido para desarrollar Yacaré, manejamos una serie de librerías para interactuar con el GPS, con la tarjeta WiFi y con la active antenna (dispositivo descrito en capítulo 4-3) que facilitaron el trabajo de comunicación con los dispositivos. Python también nos brindó la posibilidad de crear la interfaz gráfica con el usuario mediante el diseñador de interfaces gráficas Glade. Además las laptops XO poseen un sistema operativo basado en Python lo que nos permite creer que a futuro Yacaré pueda llegar a ser instalado en éstas.

Elegimos un modelo de programación modular, de esta manera subdividimos la aplicación en módulos independientes con el fin de hacerla mas manejable y adaptable a cambios. Cada uno de estos módulos tiene una tarea bien definida dentro de Yacaré, éstos se describirán a fondo mas adelante, en cuanto a la comunicación con el GPS, el manejo de los datos relevados, las adquisiciones de Mesh, los escaneos de la red WiFi y el módulo integrador que maneja la interfaz con el usuario. En caso de que un módulo necesite de otro, puede comunicarse con éste mediante interfaces bien definidas. Esto brinda una aplicación escalable en arquitectura y funcionalidades hacia el usuario adaptable a futuras tecnologías.^[1]

2- 3 Estructura de Yacaré

La aplicación fue subdividida en módulos con el fin de separar las diferentes tareas que realiza Yacaré. Las cuatro funcionalidades bien definidas son: la comunicación con el GPS, las adquisiciones en busca de Portales Mesh, los escaneos de la red WiFi y el manejo de los datos relevados. Existen cuatro paquetes de la aplicación que controlan las tareas: *modulo_GPS*, *modulo_mesh*, *modulo_wifi* y *modulo_datos*.

Luego de desarrolladas las subdivisiones, se creó un módulo integrador de la aplicación que también controla la interfaz gráfica, llamado *yacare_GUI*. En dicho módulo son mapeadas todas las acciones que realiza el usuario sobre las diferentes funcionalidades del resto de los submódulos (mas adelante detallaremos como *yacare_GUI* realiza estas asociaciones) así como las diferentes operaciones que se realizan en background.

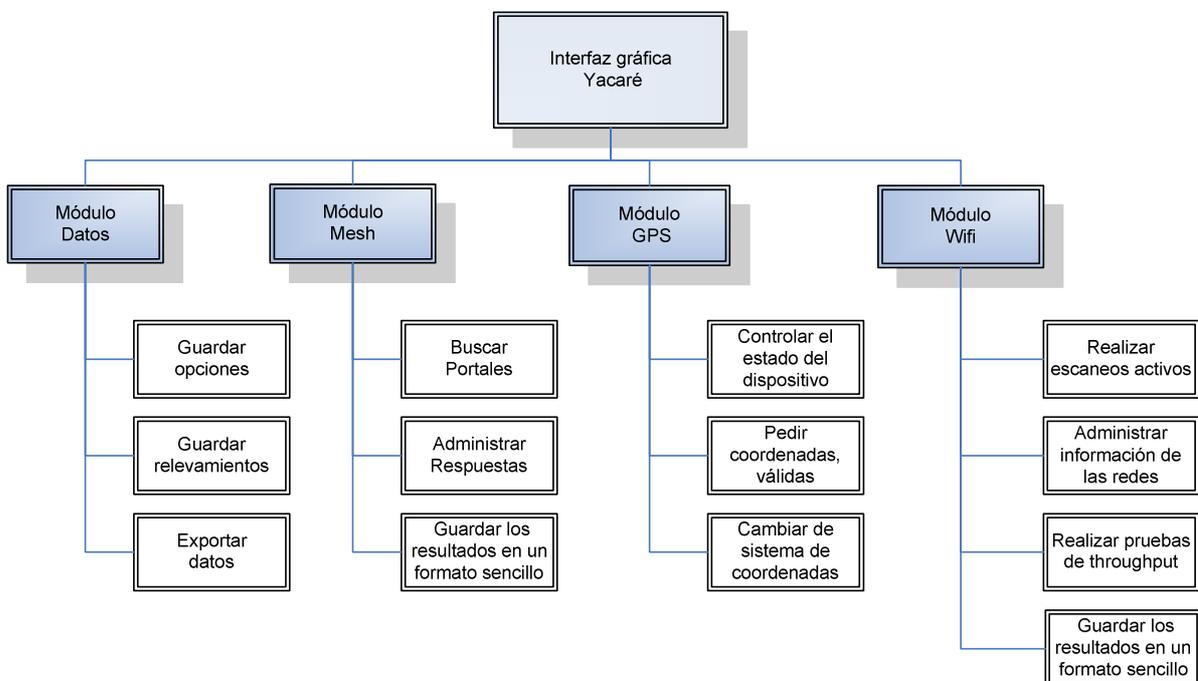


Fig 2.1: Diagrama modular de Yacaré

Como realizar varias tareas en paralelo es algo primordial, debimos diseñar un sistema de threads que nos ayuda a dinamizar la ejecución del programa. Otro problema con el que nos encontramos fue el control de acceso a la interfaz gráfica para la actualización de tablas y gráficas. Utilizamos funciones de bloqueo para que cada thread pida permisos para actualizar la interfaz. Este sistema evita errores en tiempo de ejecución típicos, como los de querer actualizar dos tablas en simultaneo.

2- 4 Requerimientos de la aplicación

Para que Yacaré funcione correctamente debemos cumplir los requisitos mínimos detallados a continuación:

Hardware

- Pentium Dual Core 1.6GHz, 1GB de RAM
- Tarjeta de red inalámbrica compatible con 802.11b/g para las adquisiciones WiFi con drivers que soporten wireless extensión v22 o superior (Recomendamos fuertemente utilizar la interfaz WiFi de la OLPC active antenna)
- OLPC active antenna (conocida por Ceibal como *Radio Mesh*) para las adquisiciones Mesh
- GPS compatible con *gpsd* y *gpsd-clients*

Software

Sistema Operativo:

- Fedora Core 8 o superior
- Kernel 2.6-23 o superior

Software adicional:

- Gnome
- Python 2.5
- Gpsd
- Pygarmin
- Gpsd-clients
- Firmware usb8388.bin
- Pcap
- Paramiko
- Numpy
- Matplotlib

3. Escaneo de redes 802.11b/g y pruebas de ancho de banda

En este capítulo estudiaremos las formas en las que podemos realizar escaneos de redes 802.11b/g, y las pruebas para el estudio de ancho de banda. Para comprender el concepto de escaneo, y definir los actores que participan de la tecnología, realizamos un estudio del estándar 802.11, donde se detallan tanto la capa física como la subcapa de acceso al medio, MAC. Muchos de los conceptos serán tenidos en cuenta a la hora del estudio de redes Mesh para la búsqueda de Portales.

Al existir dos formas de realizar los escaneos, debimos optar, y para ello se realizaron distintas pruebas y se tomaron en cuenta los antecedentes de los relevamientos de cobertura de Ceibal. A pesar de la sencillez de los escaneos pasivos, optamos por realizarlos activos, ya que ofrece la posibilidad de interrogar al Access Point por información. Para la aplicación que realizamos, no es necesario solicitar información adicional, pero se tiene en cuenta que Ceibal puede llegar a necesitarlo a futuro.

Para la realización de las pruebas de ancho de banda, se priorizó la seguridad de la red y sus integrantes. Esto es, Servidor, Access Points y Laptops. Nos decidimos por generar conexiones Secure Shell con el servidor, y generar un flujo de datos en el sentido cliente servidor, y luego en el sentido contrario. Conociendo el tráfico generado, y capturando el tráfico total cursado, podemos evaluar tanto la percepción del usuario en cuanto a la tasa de transferencia, así como el impacto de la transferencia en la red.

3- 1 Introducción a 802.11

El estándar de 802.11 en su revisión de 2007 define las capas física y de control de acceso al medio. El mismo fue realizado por la organización IEEE, Institute of Electrical and Electronics Engineers, y tiene como objetivo *brindar conectividad wireless a elementos sencillos, portables que requieren rápido desarrollo*. El estándar original fue publicado en 1999 y reafirmado en 2003 y 2007. La versión del año 2007, es la que utilizamos para el estudio de la tecnología.

El estándar 802.11:

- Describe las características de un dispositivo que cumple la norma 802.11 para operar en redes de infraestructura como en redes ad-hoc.
- Define los procedimientos de la subcapa MAC para soportar servicios de entrega de datos de forma asíncrona.
- Define distintas formas de señalización de la capa física y funciones controladas por la subcapa MAC.
- Permite la coexistencia de redes inalámbricas de área local (WLAN) 802.11 con otras WLANs que posiblemente estén superpuestas.
- Describe los requerimientos y procedimientos para brindar transporte de datos de forma confiable y segura sobre el medio inalámbrico (WM) y de autenticación con otros dispositivos que cumplen la norma 802.11.
- Define los mecanismos para selección dinámica de frecuencia (DFS) y control de potencia de transmisión (TPC) que pueden ser utilizados para cumplir las regulaciones locales.
- Define los procedimientos de la subcapa MAC para soportar redes de área local (LAN) con requerimientos de calidad de servicio (QoS), incluyendo el transporte de voz, audio y video. ^[2]

La arquitectura de la capa física y de enlace se puede ver en la figura 3.1. Viendo la división de la capa física (PHY), se define la subcapa PMD, dependiente del medio físico, y la PLCP que es la subcapa de convergencia. En esta última, se toman las tramas desde la subcapa MAC y se le agregan los campos referentes a sincronismo, velocidad de transmisión y delimitadores de tramas para la comunicación entre estaciones mediante la PMD. La subcapa MAC, como lo indica su sigla, se encarga del control del acceso al WM. Utiliza servicios de la PHY para conocer el estado del WM, y cuenta con mecanismos para evitar colisionar con las distintas STAs que comparten el espectro.

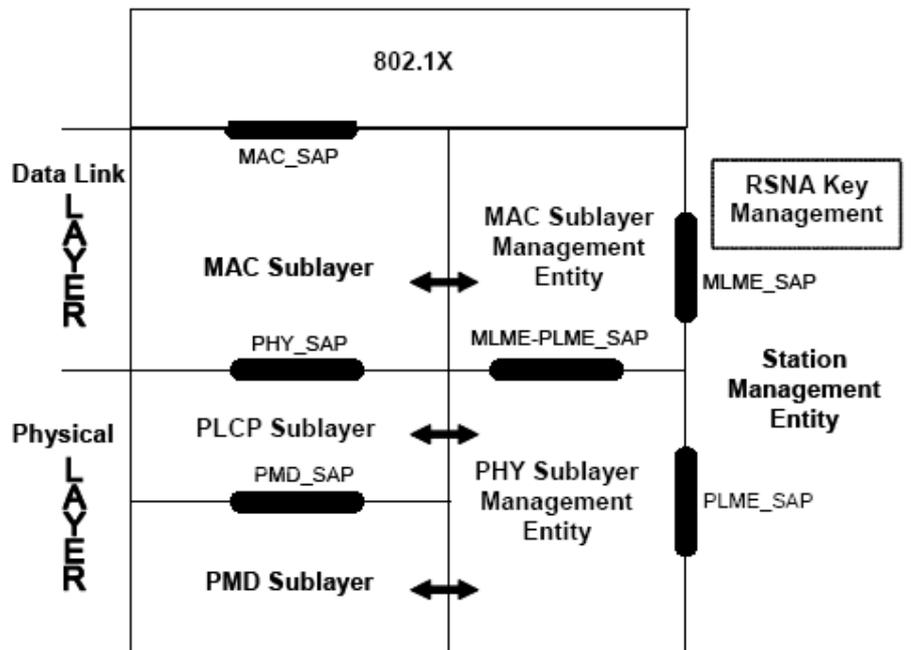


Fig 3.1: Estructura de capas del estándar 802.11

Una segunda subcapa de la capa de enlace, se encarga de mantener la independencia de capas. A los efectos, la LLC (subcapa de control de capa de enlace), realiza las tareas que permiten que desde la capa de red, no se distinga entre la red 802.11, o una 802.3 (Ethernet). No es una tarea sencilla, dado que el WM se puede ver congestionado por muchos, y muy variados motivos.

Debemos diferenciar las distintas versiones de 802.11 según tres tipos: a, b y g. Todos ellos trabajan en bandas no licenciadas, pero el primero trabaja en la banda de 5GHz, mientras que los modos b y g son para 2.4GHz. En la primera versión del estándar, se lograban alcanzar velocidades de 2Mbps. Distintas mejoras en las técnicas de modulación, junto con el desarrollo más económico de los dispositivos de red, permitieron lograr velocidades de transmisión de hasta 54Mbps. En el *anexo 802.11: Capa Física, capítulo 9-1*, podemos encontrar como el bit rate se modifica utilizando diferentes combinaciones de codificación y modulación.

Los factores fundamentales que afectan el diseño, son tanto la movilidad de las estaciones, como la coexistencia con redes en las mismas bandas de frecuencia. Los mecanismos de acceso al medio que implementa la subcapa MAC, tienen técnicas que permiten evitar las consecuencias de tener estaciones que no logran “verse”.

Otras dificultades que afectan el desempeño de las redes 802.11, son los distintos dispositivos electrónicos como hornos microondas y teléfonos inalámbricos que utilizan la banda de 2.4GHz.

La cantidad de canales en las bandas para 802.11b/g está determinada por las regulaciones locales. En Uruguay, la banda de 2.4GHz se divide en once canales de 22MHz separados 5MHz desde los 2.412GHz hasta 2.462GHz, correspondientes a las frecuencias centrales de los canales uno y once. En la siguiente figura 3.2 podemos ver el solapamiento de canales del 1 al 14.

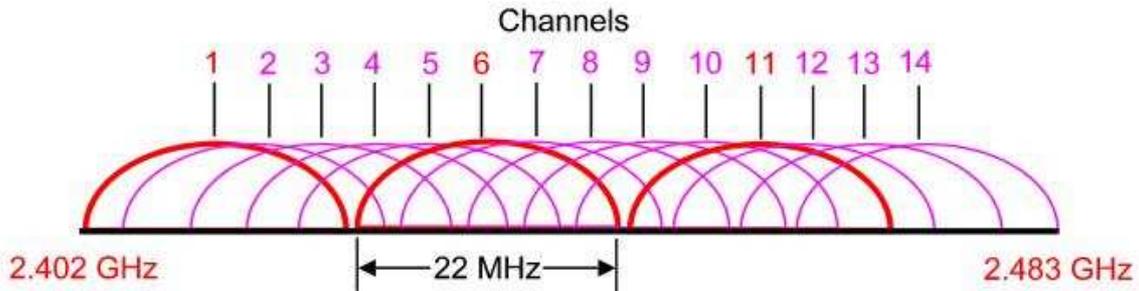


Fig 3.2: División de canales para 802.11b/g

Ante la necesidad de utilizar tres canales o más, y teniendo en cuenta que la Mesh se establece en los canales 1, 6 y 11, se eligen éstos últimos para el despliegue de la red outdoor. Esto no quiere decir que Ceibal no instale Access Points en otros canales, sino que para que la red Mesh extienda efectivamente la cobertura, los equipos de cobertura outdoor deben estar en los canales mencionados. A la hora de realizar el radio planning, se deberá intentar que las celdas del mismo canal estén alejadas la mayor distancia posible. El solapamiento de las celdas es parte de la información que debemos ser capaces de desplegar en mapas, lo que nos lleva a tener que exportar archivos que permitan realizar esta observación.

3- 1.1 Definiciones y entidades que participan en 802.11

El estándar define muchos actores y funcionalidades de forma sumamente clara. Si bien, es una arquitectura de datos muy utilizada actualmente, esencialmente se emplea en hogares y empresas, para tráfico de Internet y redes de área local inalámbrica. El número de estaciones en una red como las mencionadas es muy reducido comparado con las redes desplegadas por Ceibal. El despliegue en un área metropolitana, tiene en cuenta las limitantes de la tecnología, en cuanto a propagación radioeléctrica, los terminales de los usuarios (laptops XO) y la pequeña porción de canales que ofrece la banda de 2.4GHz. Si además tenemos en cuenta que la Mesh impacta fuertemente en la cantidad de tráfico que van a cursar en cada canal, obtenemos un panorama sumamente complejo.

En la norma 802.11 se definen los siguientes actores que son de interés para el proyecto:

- **PHY:** Capa física.
- **MAC:** Subcapa de control de acceso al medio.
- **WM:** Medio inalámbrico utilizado para transmitir tramas entre PHYs de una misma red de área local inalámbrica.
- **STA:** Cualquier dispositivo que cumpla con 802.11, MAC y PHY.
- **BSS:** Conjunto de STA sincronizados utilizando las primitivas START y JOIN. Quien comienza el BSS, utiliza la primitiva START y el resto de las STAs usan JOIN. Las primitivas están definidas dentro de la capa MAC, en la MAC Sublayer Management Entity (MLME).
- **BSA:** Es el área física que contiene los miembros de un BSS.
- **AP:** Cualquier entidad que tiene funcionalidad de STA y a la vez brinda acceso al sistema de distribución (DS).
- **DS:** Para interconectar BSSs e integrar LANs, logrando ESS.
- **ESS:** Conjunto de uno o más BSSs interconectados que son vistos por el LLC de cualquier STA como una única entidad.
- **DA:** MAC de destino para una trama.
- **TPC:** Control de potencia en transmisión.

Viendo la figura 3.3:

- Las STAs son los notebooks
- Los BSS se componen de todos los notebooks asociados a un AP
- DS es el backbone en la red Ceibal (Ethernet, WDS, punto a punto, etc.)
- ESS es el conjunto de BSS definidos por cada AP (interconectados por un mismo DS)

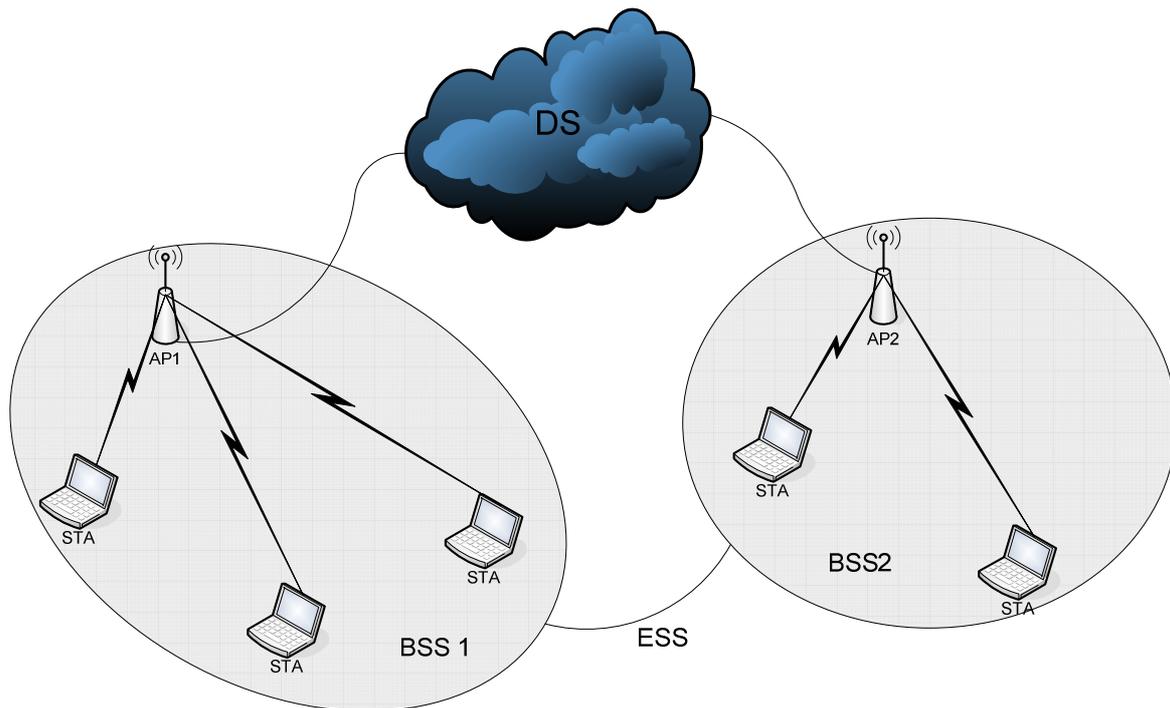


Fig 3.3: Red 802.11

El conjunto de BSSs define una nueva área de servicio, la Extended Service Area (ESA). La misma es el espacio físico en el cual las STA de distintos BSSs en el ESS pueden comunicarse.

Cada BSS es identificada unívocamente por el BSSID, que es la dirección MAC del AP que inicializa el BSS. El ESS tiene su identificador (SSID, Service Set Identifier) y cuando la red incorpora APs se lo llama ESSID (Extended Service Set Identifier). Este último es el conocido como “nombre” de la red, debido a que es quien aparece como visible al usuario, cuando se desea conectar a una red WiFi a través de un AP. Sin embargo, el ESSID no identifica el ESS unívocamente ya que es elegido por el usuario al configurar un AP. ^{[2] [3]}

3- 1.2 Acceso al medio

En un medio donde las entidades comparten recursos (en este caso el espectro radioeléctrico), hay que establecer “orden” para que todos puedan utilizarlos. La subcapa MAC es quien lleva adelante el orden, por medio de dos métodos posibles: DCF y PCF. El primero es el que se utiliza comúnmente, Función de Coordinación Distribuida y consiste en que todas las STA que participan del BSS realizan el mismo procedimiento para acceder al medio. PCF, es el caso en que un solo STA (llamado PC: Point Coordinator) coordina el acceso al medio de los demás STAs en el BSS.

Nos interesa conocer el funcionamiento de DCF, ya que PCF no especifica las funciones de coordinación que debe ejecutar el PC, y además no es utilizado. Los detalles sobre el acceso al medio en DCF se pueden ver en el *anexo 802.11: Acceso al medio, capítulo 9- 2*.

3- 2 Escaneo Pasivo y Activo

Los escaneos son parte fundamental del proceso de sincronización para la posterior asociación a un BSS. Permiten identificar los BSSs al alcance de la STA que realiza el escaneo por su BSSID, y devuelve la información sobre el SSID, canal, potencia de transmisión y sincronismo entre otros.

Una tarjeta de red 802.11b/g no puede establecerse en un canal y “escuchar” en otros canales al mismo tiempo, por lo que el proceso de escaneo debe pasar por los 11 canales de interés, aún cuando están solapados. En cada canal, debe establecerse un tiempo suficiente como para poder detectar las redes al alcance. Nuestra tarea, es desencadenar los escaneos, leer los resultados, y lograr una interpretación de los mismos. La persona que realiza el relevamiento no debe esperar a terminar el relevamiento completo para tener una idea de lo que está sucediendo en cuanto a las redes WiFi al alcance, sino que Yacaré pretende ser también una herramienta que sirva para diagnósticos sencillos en cuanto a la selección de canales. La herramienta ideal para esta tarea es un analizador de espectro.

La diferencia fundamental entre el escaneo Pasivo y Activo, es que el primero se dedica a “escuchar” por mensajes llamados Beacons cambiando de canal, y el segundo se envía un mensaje al AP y espera por la respuesta.

3- 2.1 Escaneo Activo

El procedimiento para este tipo de escaneo se puede resumir de la siguiente forma:

- Se realiza la técnica de acceso al medio mencionada anteriormente
- Se envía un mensaje Probe Request
- Se inicializa el Probe Timer (contador que junto con las reglas que vemos a continuación permite acelerar y controlar el proceso de escaneo activo)
- Se sensa el canal, y si no se detecta ninguna respuesta antes de que el Probe Timer alcanza el valor MinChannelTime (tiempo mínimo de canal) se cambia al siguiente canal y se comienza desde el primer paso
- Si por medio del sensado del canal se detecta que existen redes en actividad, se procesan los mensajes Probe Response luego de que el Probe Timer alcanza MaxChannelTime (tiempo máximo de canal)
- Se cambia al siguiente canal y se comienza desde el primer paso a menos que ya se hayan recorrido todos los canales ^[2]

Seguendo el ejemplo de la figura 3.4, la STA 1 realiza un escaneo activo y las estaciones STA 2 y STA 3 le responden. En el caso de STA 3, luego de recibir el Probe Request sensa el canal, esperar un tiempo DIFS, y envía el Probe Response (con ventana de contención =1).

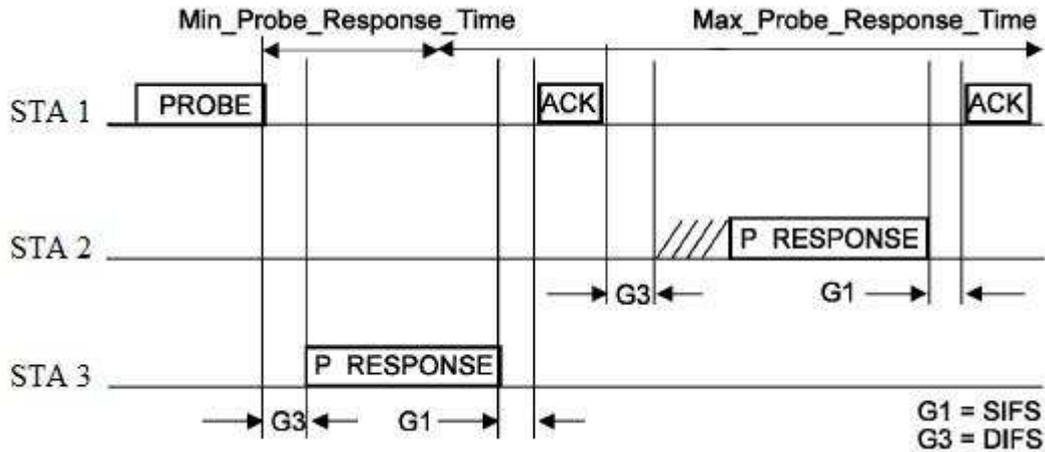


Fig 3.4: diagrama de tiempos para escaneos activos

La STA 2, cuando recibe el Probe Request, sensa el canal pero lo encuentra ocupado por STA 3. Continúa con la técnica de acceso al medio habitual, con una ventana de contención de varios TS hasta que logra acceder. Envía entonces el Probe Response.

A pesar de que los mensajes son de pocos bytes, y el tiempo que se ocupa en el aire es de unos pocos microsegundos, debemos tener en cuenta el factor de que el acceso al medio es conflictivo y produce colisiones en redes de muchas STA. Es una de las razones por las cuales siempre conviene realizar varios escaneos para descubrir las redes al alcance.

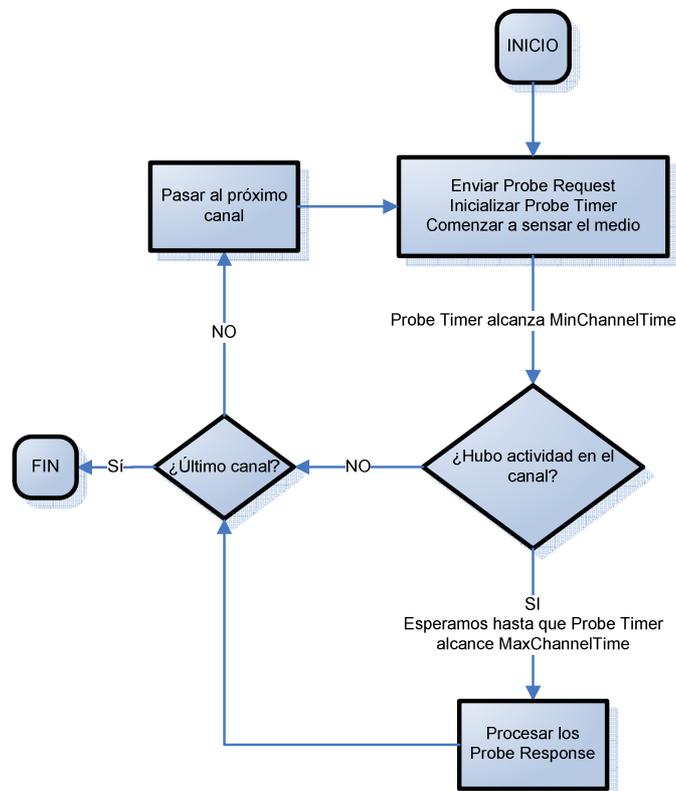


Fig 3.5: Flujo de eventos de escaneos Activos

3- 2.1.1 Probe Request

El mensaje Probe Request es una solicitud de información. Tiene como dirección de destino (DA) la MAC del AP (o la dirección de Broadcast), como SSID el SSID de la STA (o el wildcard SSID) y como BSSID el BSSID de la STA (o el wildcard BSSID). El wildcard SSID es un identificador de que será reconocido por todas las STA (identifica cualquier ESS) y el wildcard BSSID es el identificador equivalente para todos los BSSs. El formato del mensaje se puede encontrar en el *anexo Mensajes Escaneos*. En un IBSS, quien envió el último Beacon es quien debe responder el Probe Request.

Cuenta con el campo *Request Information* lista la información adicional que espera la estación que escanea. Actualmente no lo utilizamos, pero pueden ser de utilidad a futuro, como mencionáramos anteriormente.

3- 2.1.2 Probe Response

El mensaje Probe Response envía información muy similar a la del Beacon, pero además incluye la información solicitada por quien envió el Probe Request, si cuenta con ella. Es enviado en unicast, con la dirección de destino de quien envió el Probe Request. El mensaje es enviado si y solo si el Probe Request cumple las siguientes condiciones:

- El SSID es el SSID de la STA, o es el wildcard SSID
- El BSSID es el BSSID de la STA, o es el wildcard BSSID
- DA es la MAC de la STA o Broadcast

Una diferencia fundamental con los Beacons, es que al ser enviados en unicast, los Probe Response se envían a la máxima tasa de transferencia posible, mientras que los Beacons se envían siempre a 1Mbps. Si bien parece una diferencia no muy relevante, a la hora de relevar equipos MIMO, tiene un peso muy importante, como se verá mas adelante.

El formato del mensaje Probe Response se presenta en la *figura 9.8 del anexo Escaneos: Mensajes intercambiados, capítulo 9- 3*, y como se puede ver tiene información que envía siempre, y algunos campos los envía únicamente si hay una solicitud explícita en el Probe Request.

3- 2.1.3 Escaneo Activo y tecnología MIMO

Algunos APs de la red Ceibal, utilizan tecnología MIMO (múltiples entradas, múltiples salidas) para reducir las pérdidas por obstáculos. Los mismos cuentan con un array de seis antenas que tiene la capacidad de decidir cuan desfasadas tiene que estar las señales a transmitir en cada antena, según como recibe las señales de las STA. Cuanto más tráfico genera la STA, el equipo puede “aprender” más de las señales recibidas en cada antena, y tener una ganancia de hasta 10dB en recepción. A este proceso se le denomina *beamforming*.

Al escanear de forma pasiva, el AP no realiza el mencionado procedimiento, sino que envía por el proceso habitual los Beacons, sin aprovechar el array de antenas. Es un mensaje de Broadcast, el destino son todas las STA al alcance, y no hay *beamforming*. Los relevamientos de cobertura con escaneos pasivos nos entregan cobertura mucho menor a la que podemos lograr con el AP. Sin embargo, al realizar escaneos activos, los equipos MIMO pueden ir “aprendiendo” como realizar el *beamforming* en base a los Probe Requests, y enviar Probe Responses ajustando la señal lo mejor posible. Sin embargo, no podemos decir a priori que se mejore mucho con los escaneos activos, ya que la tecnología necesita de tráfico significativo, y no únicamente de unos pocos bytes como lo es el Probe Request.

Las formas que consideramos pueden solucionar este problema, es escanear de forma activa sucesivamente hasta que el AP aprenda como debe enviar la información, y al recibir los Probe Response hacer un cálculo mas real sobre los parámetros de Potencia y Calidad en recepción. Por este tipo de razones es que la cantidad de escaneos por punto es un parámetro de las opciones del programa. Sin embargo, al promediar estamos teniendo en cuenta los primeros valores, que tienen un *beamforming* menor a los de los últimos escaneos. El equilibrio entre la cantidad de escaneos y el uso de las pruebas de ancho de banda para el relevamiento de cobertura, son factores que bajan la movilidad.

Por las razones mencionadas, Yacaré es capaz de generar tráfico y a la vez realizar escaneos activos para relevar este tipo de Access Points. No es uno de los objetivos con los que comenzó Yacaré, al ver la necesidad consideramos que era muy sencillo adaptar el software para realizar los procesos en simultáneo. ^[4]

3- 2.2 Escaneo Pasivo

Como mencionamos anteriormente, consiste fundamentalmente en escuchar Beacons en los distintos canales. La información de los Beacons permite la sincronización entre STAs, para lograr el correcto acceso al medio. Los mismos son enviados periódicamente por los APs (salvo que se deshabilite) con un intervalo de tiempo denominado Beacon Interval (típicamente 102,4ms).

El procedimiento de la STA es básicamente el siguiente:

- Se establece en un canal durante un tiempo máximo definido (MaxChannelTime), típicamente 102,4ms
- Una vez alcanzado este tiempo, procesa los Beacons capturados (si los hay)
- Cambia al siguiente canal

De manera de minimizar el tiempo de escaneo, MaxChannelTime debe ser igual a Beacon Interval.

La información que vemos en los Beacons es muy similar a la que podemos encontrar en los Probe Responses. La diferencia fundamental radica en que en estos últimos se debe pedir la información. El formato del Beacon se puede ver en la *figura 9.9 del anexo Escaneos: Mensajes intercambiados, capítulo 9- 3*. Otra diferencia, es que el mensaje se envía a un Mbps. Es un mensaje de Broadcast, y como tal, debe enviarse en el menor nivel de modulación de forma que todas las STA al alcance puedan recibirlo.

3- 3 Relevamientos previos de Ceibal y requerimientos

La idea original con la que surge el proyecto, tiene que ver con las pruebas de conexión durante el plan piloto en el año 2007 en la localidad de Villa Cardal, Florida. Distintos equipos se instalaron en el pueblo y se diseñaron varias formas de ir "midiendo conectividad". Una de las formas, que se sigue usando al día de hoy, es utilizar en comando de Linux *iwlist scan* que despliega escaneos completos de los canales de 802.11 b/g. Se recorre la localidad ejecutando scripts que básicamente ejecutan el comando varias veces, lo promedian y lo guardan en un archivo de texto plano. Con la incorporación del GIS y mapas completos de las localidades del país, se logró generar capas proyectadas en los mapas a partir de los puntos relevados. El proyecto Yacaré pretende dar automatismo a forma de relevar de manera mas rápida e integrada con los dispositivos de Mesh y GPS, así como un procesamiento y guardado de los datos mas eficiente.

El comando *iwlist scan* devuelve varios parámetros que han representado para Ceibal de forma fiel las posibilidades de conexión a una red, por eso comenzamos nuestro desarrollo del módulo enfocados en el uso del mismo, su documentación y su funcionamiento.

3- 3.1 Wireless-tools: El comando *iwlist*

El paquete *wireless-tools*, que puede ser instalado en cualquier distribución de Linux, contiene, entre otros comandos, el *iwlist*. Dentro del rango de opciones con las que puede ser ejecutado, la más utilizada es la funcionalidad *scanning*. La opción mencionada despliega información de una red inalámbrica WiFi: frecuencia, bit rate, tipo de autenticación, si utiliza encriptación, potencia de transmisión, etc. En este caso, se desencadena un escaneo activo en cada canal. Un ejemplo es el siguiente:

```
wlan0    Scan completed :
          Cell 01 - Address: 00:17:3F:BB:D5:4A
            ESSID:"Casa"
            Mode:Managed
            Frequency:2.412 GHz (Channel 1)
            Quality=85/100   Signal level=-65 dBm   Noise
            level=-86dBm
            Encryption key:off
            Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 6
            Mb/s
                    9 Mb/s; 12 Mb/s; 18 Mb/s; 24 Mb/s;
            36 Mb/s
                    48 Mb/s; 54 Mb/s
```

Fig 3.6: Resultado *iwlist scanning*

La documentación de *wireless-tools*, en conjunto con el código fuente, nos permitió conocer como la aplicación manipula los mensajes que permiten ejecutar un escaneo y leer los resultados del mismo. Los mensajes intercambiados con el dispositivo de red son llamados *ioctls* (controles de entrada/salida), en el caso de los escaneos, se definen los *SIOCSIWSCAN* y *SIOCGIWSCAN* *ioctls* que veremos en 3- 3.2.

El modo de escaneo, es un parámetro que puede ser cambiado en algunos casos. Distintas tarjetas, permiten cambiar dinámicamente el modo, con la utilización de *iwpriv* o directamente con la implementación de los *ioctl*s SIOCSIWPRIV, SIOCGIWPRIV. De todas maneras, encontramos que las tarjetas utilizadas por nosotros funcionan del mismo modo.

Se realizan escaneos activos, sin embargo, el resultado del escaneo devuelve datos sobre redes que no enviaron Probe Responses. Se debe a que se capturan los Beacons al cambiar por los distintos canales durante el escaneo. Tampoco son capturadas todas las redes inalámbricas con un solo escaneo, ya que como vimos anteriormente, si durante un tiempo MinChannelTime no se detecta actividad en el canal, se cambia al siguiente.

3- 3.2 Módulo Python-WiFi

Nuestra intención desde un principio fue comunicarnos de forma directa con el hardware, sin la necesidad de comandos de Linux existentes. Al estudiar el funcionamiento de *iwlist* nos pareció bueno poder tomar control sobre la comunicación, y hacer más eficiente el software, y menos dependiente de factores externos. En este contexto, buscamos herramientas y el módulo Python-Wifi, en una versión en desarrollo, logra los objetivos mencionados. Implementa los *ioctl*s definidos para realizar y ver resultados de escaneos y además cuenta con una batería de clases que permiten manipular los resultados con facilidad.

Se realizaron modificaciones mínimas del código fuente, de forma de no tener errores en tiempo de ejecución. Por esta razón el módulo no es instalado, sino que se utiliza únicamente como librería. ^[24]

3- 3.3 Comunicación con el hardware: ioctl

Como fue mencionado, los *ioctl*s permiten la interacción usuario-hardware (también usuario-kernel) y evitan la dependencia con el driver del hardware, y sus formas de consultarlo. También son una forma de consultar o modificar parámetros del driver dinámicamente, y si el driver del hardware es removido, no se ven afectados otros dispositivos. Las extensiones wireless que introduce wireless-tools necesitan que los drivers de los dispositivos de red soporten estos mecanismos. ^[5]

Con este concepto sencillo de *ioctl*, podemos entender el código fuente de *iwlist* y, como consecuencia, el funcionamiento de *iwlist scanning*. Para pedir al dispositivo de red que realice un escaneo, podemos resumir la secuencia de eventos de la siguiente forma:

1. Se envía un *ioctl* SIOCSIWSCAN al dispositivo
2. Se chequean las banderas de error y si el escaneo es exitoso
3. Se envía el *ioctl* SIOCGIWSCAN y se lee el buffer donde se almacenó la información recolectada

Este procedimiento está simplificado ya que no es del interés del proyecto implementar este tipo de herramientas, sino aprovecharlas de manera eficiente.

El comando *iwlist* tiene una rutina como la que mencionamos anteriormente, y realizamos un módulo WiFi que lo utiliza. Sin embargo, consideramos que debíamos interactuar directamente con el hardware de la tarjeta de red, y no tener el pasaje software – *iwlist* – hardware, ya que la experiencia nos demostró grandes puntos de falla en la dependencia con la versión de wireless-tools. Además, luego de testear repetidas veces Python-Wifi y realizar las modificaciones necesarias para acoplarlo al módulo, resultó sumamente estable.

3- 4 Módulo WiFi

3- 4.1 Objetivos y funcionamiento

Los objetivos del módulo son los siguientes:

- Realizar escaneos y devolver los resultados en un formato sencillo de administrar
- Realizar pruebas de ancho de banda.

Cuando se le pide al programa un escaneo (por presionar Escanear, o Adquirir), el programa invoca el método *test_wifi* de *modulo_wifi.py* para llevar a cabo los escaneos en busca de APs, mientras que la prueba de ancho de banda es únicamente en algunos puntos del recorrido llamando el método *test_bw_subida* y *test_bw_bajada*. Esto se debe a que para realizar la prueba, se debe establecer primero la conexión, y puede que los Access Points con los que hicimos la prueba anteriormente, no estén mas al alcance.

Es tarea de las rutinas de control el invocar los métodos de forma correcta. El módulo no es un ejecutable en si mismo, sino que como el resto de los módulos es un conjunto de métodos y clases.

Los datos nos permitirán identificar tanto los BSA como los ESA.

3- 4.2 Escaneos

El Módulo WiFi se basa en un método llamado *test_wifi*. El mismo utiliza los parámetros: número de adquisiciones, interfaz de WiFi y ESSIDs “conocidos”. Con estos datos, somos capaces de desplegar información periódicamente al usuario, y realizar los escaneos completos en los puntos a relevar. Por más que se realicen escaneos a intervalos cortos de algunos segundos, éstos son con el único propósito de desplegar información al usuario aunque de todas maneras se podrán guardar y desplegar al igual que el resto.

Algunas tarjetas de red no son muy estables en cuanto a los resultados de los escaneos, esto nos lleva a tomar varios escaneos y promediarlos. En particular, si se quisiera realizar los escaneos utilizando la interfaz de WiFi de la active antenna, sería importante tomar varios escaneos, ya que el tiempo que pasa en cada canal durante los escaneos es menor a 102.4ms. No captura algunos Beacons y Probe Responses, ya que cambia de canal muy rápido, y además, varía mucho el valor de Potencia y Calidad en dos escaneos consecutivos desde el mismo lugar. En un mismo entorno, escaneando con la active antenna y varias tarjetas inalámbricas (Broadcom, Ralink y Atheros), notamos que el tiempo que le lleva a la radio el escaneo de los 11 canales es la mitad de lo que le lleva al resto de las tarjetas.

El escanear con la active antenna, resulta muy útil para procesar los datos. La ganancia de la misma es un poco menor al de la XO, ya que cuenta únicamente con

una antena de 2dbi, y las antenas de la XO tienen una excelente ganancia comparando con la gran mayoría de los laptops. La diferencia entre la active antenna y la XO al hacer algunas pruebas, resultó bastante menor de lo esperada. Además el cálculo de calidad realizado por el driver no debe ser modificado, ya que utilizamos el mismo dispositivo de red que la XO. Recomendamos la utilización de estos dispositivos para los relevamientos, con un número de escaneos mayor a dos.

Para administrar los datos de los Access Points de forma cómoda y ordenada, se define la clase AP. Los atributos de la clase son:

- Canal (*número*)
- Potencia (*dBm*)
- Ruido (*dBm*)
- Calidad (%)
- ESSID
- MAC (*BSSID*)
- Modo (*Si es AP, será modo Managed*)
- Conocida (*Boolean*)
- Cantidad

El atributo “Conocida” es verdadero en el caso en que el usuario haya seleccionado el ESSID como conocido en las opciones del programa. Permite un seguimiento de cobertura en la localidad al momento del relevamiento, comparando la potencia de las redes propias e interferentes y desplegándolo al usuario periódicamente. El procesamiento de los datos es igual tanto para las redes conocidas como para las interferentes. Cantidad tiene que ver con el parámetro de opciones llamado número de escaneos y representa la cantidad de apariciones del AP en los escaneos realizados en una posición dada. El campo es únicamente para el mejor tratamiento de los datos

Cada una de las instancias AP, una vez concluidos los escaneos en el punto de relevamiento, promedia los valores de los atributos Potencia, Ruido y Calidad. Los resultados de *test_wifi* se guardan en dos listas (clases *list* de Python) para ser administrados y guardados por el programa que controla la adquisición.

El módulo *Python-Wifi* cuenta con la implementación de los *ioctl*s necesarios para la comunicación con el hardware. Utilizamos *iwlibs.py* que cuenta con escaneos iguales a los de *iwlist*. El tipo de escaneo (activo y/o pasivo) depende del driver y no de la implementación que realicemos. En nuestro caso, con las tarjetas de red que trabajamos, los escaneos son activos y pasivos como veremos más adelante, y algunos chips permiten setear el modo de escaneo con el comando *iwpriv*, también de *wireless-tools*.

3- 4.3 Parámetros de las adquisiciones

Para comenzar a tomar muestras con el programa, se deben elegir un grupo de opciones que incluyen las referidas al escaneo de redes WiFi. Como primer parámetro importante, se debe elegir la interfaz de red con la que se realizarán los escaneos. La ventana de opciones cuenta con la lista de interfaces 802.11 disponibles. Como segundo parámetro se necesita conocer cuantos escaneos se deben tomar por punto de adquisición, y por último la lista de ESSIDs denominados conocidos. La cantidad de escaneos, como mencionamos anteriormente, aporta estabilidad a las adquisiciones, y los ESSIDs conocidos permiten desplegar información útil al momento de la

adquisición. Se pueden configurar hasta tres ESSID distintos para identificar. Por como está implementado el módulo, un ESSID va a ser identificado como conocido si contiene alguno de los tres seleccionados en las opciones. Por ejemplo, si elijo únicamente el ESSID “CEIBAL” y el resultado del escaneo arroja una red con ESSID “Ceibal-Montevideo”, la clase AP asociada a la red tendrá el atributo Conocida en True. Esto se ve reflejado en la gráfica de la aplicación, donde se distinguen las que se identificaron como conocidas

3- 4.4 Flujo de eventos en una adquisición

Cuando el usuario presiona el botón adquirir, o el programa va a ejecutar un escaneo para desplegar al usuario, se crea un thread que llama al método *test_wifi* de *modulo_wifi.py*. Los parámetros necesarios para la ejecución, el número de escaneos y la interfaz, ya que puede darse que en un futuro se desee realizar escaneos en dos interfaces de red distintas, y sería más sencillo de realizar modificaciones al código.

La función *test_wifi* crea la interfaz en la cual se realizará el escaneo, utilizando la clase *Wireless* de *iwlibs.py*. La clase tiene implementado el método *scan* que realiza el proceso de: enviar el SIOCSIWSCAN, que desencadena el escaneo activo, luego chequea que no haya habido errores, y comienza a enviar SIOCGIWSCAN esperando a que el hardware le permita obtener los resultados.

El proceso de escaneo activo es el mismo detallado en 3- 2.1. Las tareas restantes que realiza *test_wifi*, tienen que ver con adaptar el resultado del escaneo a clases AP, distinguiendo las redes propias e interferentes y separándolas en redes distintas. Estas listas son las salidas de *test_wifi*.

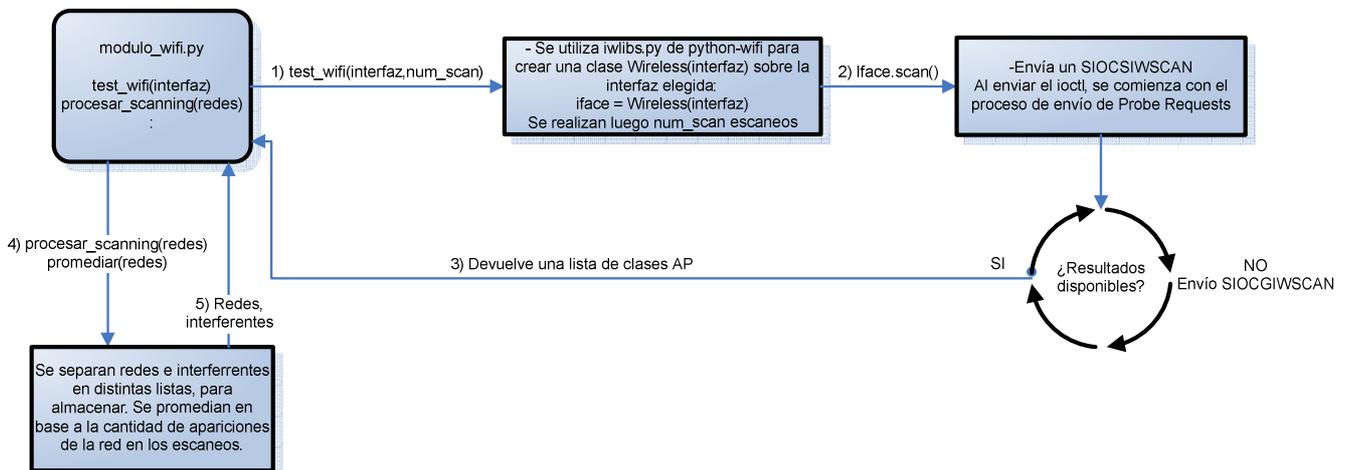


Fig 3.7: Flujo de eventos de escaneos

No es despreciable el consumo de recursos del laptop en el proceso de escaneos. Los sucesivos escaneos, en conjunto con la actualización de las gráficas y tablas hacen notar el uso del CPU. Para bajar el efecto, es importante elegir bien el tiempo entre escaneos continuos en el menú opciones. También es recomendable, parar los escaneos continuos cuando se realiza una adquisición que requiere muchos escaneos, esto se detallará en el capítulo 7- 3.

3- 5 Pruebas de ancho de banda

Los objetivos con los que comenzamos a construir las pruebas, eran únicamente conocer el ancho de banda que puedo lograr al momento que se realiza el relevamiento en determinadas posiciones. Los antecedentes de pruebas similares en el LATU, son con el programa Iperf que veremos a continuación. La modalidad de la prueba debe ser tanto en el sentido del uplink como downlink con conexiones TCP.

3- 5.1 Antecedentes y pruebas con Iperf

El Iperf es una herramienta muy utilizada para testing sencillo de enlaces, generar tráfico en una red, cálculo de ancho de banda y pérdidas. Funciona en la modalidad Cliente-Servidor y debe ejecutarse en ambos. Los resultados son muy útiles cuando se quieren hacer pruebas específicas en enlaces punto a punto y ver los niveles de modulación, throughput y pérdidas. El problema surge cuando la conexión se vuelve inestable y el programa no responde de la mejor manera. Nuestra idea inicial era incorporarlo al software Yacaré, pero encontramos muy inconveniente la modalidad en la que funciona.

Una forma para utilizar el Iperf, sería la siguiente:

- Se ejecuta en el servidor, el Iperf en modo servidor escuchando TCP y UDP en distintos puertos.
- Se ejecuta en el laptop el Iperf en modo cliente durante un lapso de tiempo determinado generando tráfico TCP. Se obtiene información sobre las velocidades que se pueden alcanzar en la red.
- A esas velocidades, se genera tráfico UDP. El Iperf tiene una modalidad de trabajo donde el servidor sabe a priori que es lo que le va a enviar el cliente, y a partir de eso calcula las pérdidas en las comunicaciones, ya que UDP no tiene recuperación de errores. Luego le envía al cliente el reporte de pérdidas y velocidad de transferencia (puede no alcanzar la velocidad elegida)

Como mencionábamos anteriormente, la prueba se vuelve ineficiente e inestable cuando la conexión con el AP también lo es. ^[6]

3- 5.2 Capturas y eficiencia en el uso de la red

Un segundo planteo que nos hicimos surgió de las primeras pruebas que realizamos transfiriendo archivos en un sentido y otro. Utilizando Wireshark para capturar las transferencias, notamos muchas retransmisiones, al punto que para transmitir archivos de tamaño 1MB, las capturas tenían “diálogos” de hasta 1,5MB entre el PC y el laptop que realizaban el FTP. Es de esperar que exista overhead en un entorno del 10 % a 12% en una transferencia de un archivo teniendo encabezados de 802.11, IP, TCP y SSH. Si únicamente tomáramos el throughput como el tamaño del archivo a transferir sobre el tiempo que demoramos en hacerlo, no veríamos reflejadas las retransmisiones a nivel TCP o a nivel 802.11. La prueba de ancho de banda incluye una captura de la actividad en la interfaz, en la que se calcula el tamaño de las tramas completo, tanto para las entrantes como salientes. Quedan incluidas las confirmaciones de las conexiones TCP, y retransmisiones.

Definimos el valor de “eficiencia de la red”, de la siguiente forma:

- E = eficiencia
- T = tamaño en bytes del archivo a transferir
- N = número de transferencias
- C = tamaño en bytes de la captura sobre la interfaz durante la transferencia

$$E(\%) = N \times \frac{T}{C} \times 100$$

3- 5.3 Funcionamiento

Por los antecedentes mencionados, es que decidimos buscar una forma más transparente para el servidor para hacer la prueba. Por transparente, nos referimos a no tener que ejecutar aplicaciones especialmente para la prueba, ya que implican tiempo al momento del relevamiento, o el trabajo previo de ejecutar el programa en todos los servidores con los que se va a interactuar. Para poder ejecutar la prueba con el servidor de Ceibal, se debe conocer un usuario por el cual pueda establecer una conexión SSH (Secure Shell) y su contraseña, además del IP y el puerto en el cual se escuchan las conexiones SSH. Sobre la conexión SSH, es que se envía y recibe un archivo de 256KB, según el diagrama de flujo de la figura 3.8.

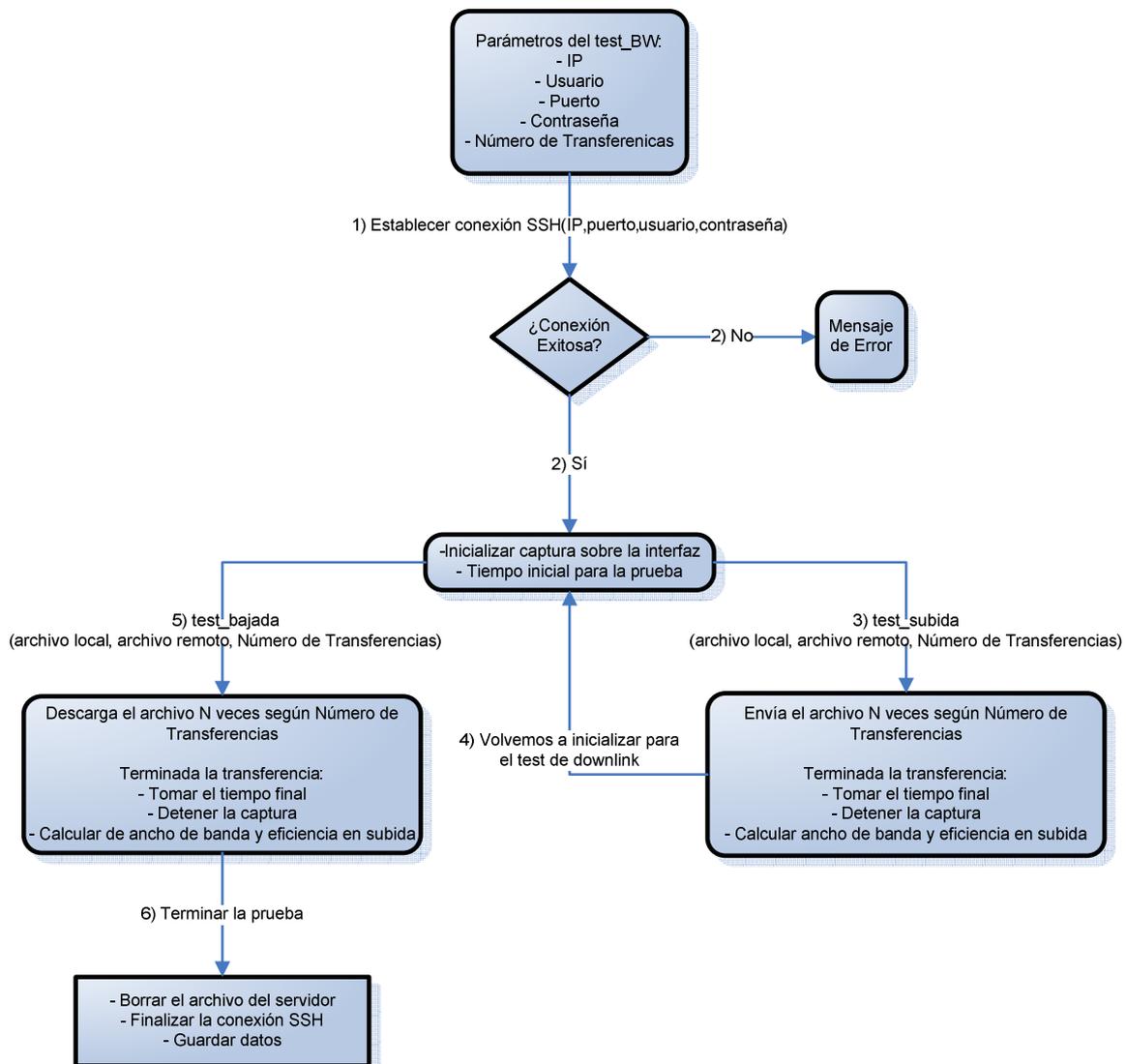


Fig 3.8: Flujo de eventos en prueba de ancho de banda

El tamaño del archivo a transferir lo decidimos en base a un estudio publicado en Web Site Optimization, donde se exponía que en los últimos años, el tamaño promedio de una página web había alcanzado los 312KB. Tomamos 256kB como tamaño de archivo ya que es sencillo trabajar con múltiplos de este tamaño (es fácil pasar a unidades de Mbps).

Creemos que la modalidad de la prueba de ancho de banda nos permite evaluar la percepción de lo que vería una XO desde el lugar, y además podríamos utilizar la eficiencia para evaluar ambientes congestionados con muchas retransmisiones.

4. Búsqueda de Portales Mesh

Este capítulo contiene un estudio de redes Mesh, presentando las principales características del draft 802.11s y la implementación de OLPC. Utilizando conceptos del estándar 802.11 se agregan los conceptos de Mesh Points y Mesh Portal Points así como las nuevas posibilidades que brindan las redes Mesh, como formar topologías de red sin necesidad de infraestructura externa y configurar laptops como portales para proveer de salida a Internet a vecinos que no tengan acceso.

Se estudiarán los mensajes que son intercambiados por los dispositivos que implementan 802.11s para descubrir Mesh Portal Points y formar adyacencias. Nos centraremos en el protocolo DHCP y los mensajes provistos que conforman el mismo, que serán los utilizados por el *Módulo Mesh* de la aplicación para la búsqueda de portales.

4- 1 Introducción al draft 802.11s

Los laptops XO entregados por el Plan Ceibal cuentan con una implementación del draft 802.11s: WLAN Mesh. Las redes Mesh, se caracterizan básicamente por nodos que se comunican entre sí sin necesidad de infraestructura, es decir, Access Points. Forman IBSSs, que son BSSs independientes. Esto quiere decir que las STA se comunican directamente entre sí, y no a través de quien inicia el BSS. Hasta este momento, no hay diferencias con las redes ad-hoc, sin embargo, las STA de la Mesh pueden tener comportamientos distintos, funcionando como Portales o Access Points.

Las XO forman caminos para que quienes no se ven directamente (vecinos), lo hagan a través de un conjunto de STA que pertenezcan a la Mesh. La complejidad radica en el descubrimiento de los vecinos, y en establecer caminos, denominados Mesh Paths, que permitan alcanzar los nodos de la Mesh. En la implementación de las XO, se crean caminos en la Mesh de hasta cuatro saltos.

Cuando una XO logra asociarse a una red externa, automáticamente se comporta como Portal en la Mesh (MPP). Las distintas STAs de la Mesh (denominadas Mesh Points) pueden usar el Portal Mesh para lograr tener acceso a la red externa. En el contexto de Ceibal, la formación de redes Mesh extiende en cierta medida la cobertura de los APs, o en el caso de no contar cobertura en algún lugar, se pueden realizar actividades compartidas utilizando los laptops.

4- 2 Ruteo en capa de enlace

Hay que diferenciar el ruteo al que haremos referencia en la Mesh, del ruteo IP. En la Mesh, los nodos se descubren y establecen caminos en capa de enlace, de forma transparente a las capas superiores. A partir de la información colectada por el protocolo de ruteo (*HWMP, Hybrid Wireless Mesh Protocol*) construyen su tabla, denominada tabla de forwarding. En base a la misma, deciden a que dirección MAC deben enviar las tramas para llegar a un determinado destino, el costo (métrica del camino), cantidad de saltos hasta el destino y tiempo de vida de las rutas.

El protocolo por el cual se construye la tabla consiste básicamente en dos mensajes: Route Request (RREQ) y Route Reply (RRPLY). Para el descubrimiento de vecinos, utilizan los métodos mencionados en el estudio de 802.11 para sincronización: escaneos pasivos y activos. Para buscar MPPs, se envían mensajes MPP Request (MPPREQ), el mensaje se propaga por la Mesh, y los portales responden con un MPP Reply. Luego elige el portal con menor métrica en la tabla.

En el caso de la figura 4.1, vemos como XO1 y XO2 a pesar de estar conectadas a distintos Access Points, se comunican por la Mesh. La XO3, a pesar de no pertenecer a los BSSs de XO1 y XO2 logra comunicarse con ambas. Con XO2, logra hacerlo directamente, mientras que para comunicarse con XO1, en su tabla de forwarding aparece como mejor camino enviárselo a XO2, y que este último lo envíe a XO1. La tabla guarda varios caminos hasta el destino, lo que le permite lograr un gran dinamismo en la topología de las redes Mesh.

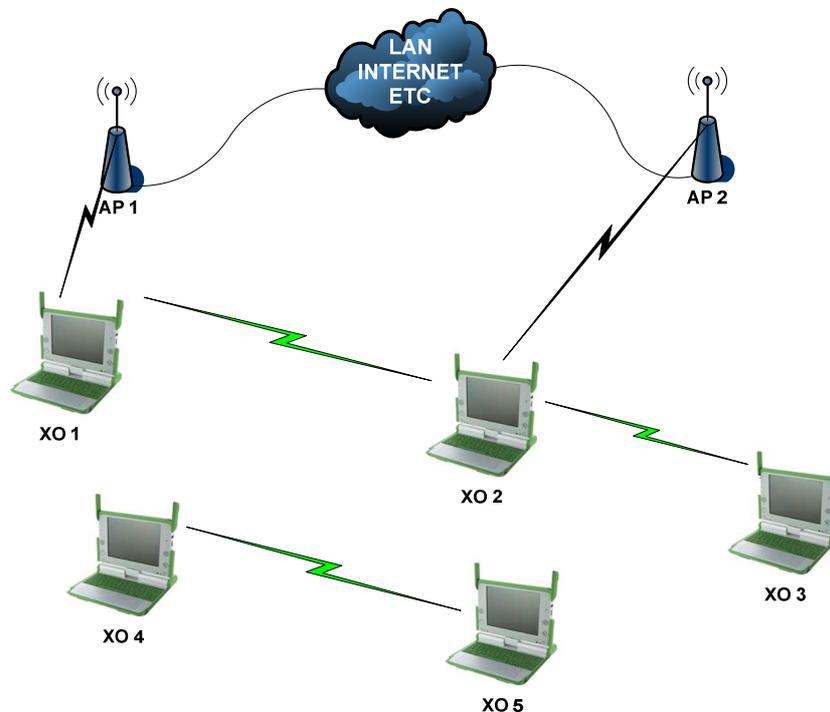


Fig 4.1: Topologías de redes Mesh

Los laptops XO4 y XO5 de la figura 4.1 forman una Mesh aislada. El IBSS al que pertenecen se identifica por el IBSSID que es una dirección MAC elegida por quien

inicia el IBSS, como en cualquier red ah-hoc. Si XO1 se comporta como portal para XO3, este último tiene acceso a la red WLAN, Internet o a la red que XO1 esté asociada, con XO2 como salto en la Mesh.

Cabe destacar que la formación de éstos IBSS, es capturada por el módulo WiFi, ya que responden a los escaneos activos realizados por el mismo.

Quedan definidos entonces los dos actores de interés en el draft 802.11s:

- **Mesh Points (MP).** Todas aquellas STA que emplean tanto la capa MAC como física de 802.11s.
- **Mesh Portal Points (MPP).** Todos los MP que permiten que otros MP se interconecten con otras redes.

Un tercer actor que participa de 802.11s son los Mesh Access Points (MAP), que son nodos capaces tomar STAs que no son MP, y servir de acceso a la red Mesh. Las XO aún no funcionan como MAP. ^{[7][8]}

4- 3 Active Antenna

La active antenna, en Ceibal conocida como radio Mesh, es un periférico USB que utiliza la misma tarjeta de red que utilizan los laptops XO (Marvell 8388 USB). OLPC desarrolló un servidor denominado XS, que brinda servicios de presencia en la Mesh entre otros. Como interfaz de Mesh, el XS utiliza varias active antenna, funcionando como MPPs. Nuestro objetivo es utilizar esta radio para encontrar portales en la Mesh, y buscar la cantidad de saltos a los mismos. Utilizamos una versión prototipo del dispositivo.

El driver Libertas desarrollado para las tarjetas está incluido en los módulos de kernel superiores a 2.6.22, para todas las distribuciones. Cuenta con ioctl's que permiten desplegar la forwarding table, y de esa forma registrar los saltos hasta cualquier destino de la Mesh. Para las distribuciones de Fedora 8 y 9, y también en Ubuntu se ha probado que la radio funciona correctamente.

Una vez que los módulos están funcionando, se instala el firmware USB8388.bin de la radio, tenemos dos nuevas interfaces de red: una de 802.11b/g y otra de Mesh en la implementación de OLPC. No debemos olvidarnos que OLPC implementa parte del draft, y como mencionamos antes, no hay capacidad aún de tener MAP.

Características

Se conecta a un receptor o fuente de energía a través de un cable con interfaz USB de 3 metros de largo. Se diseñó para funcionar como interfaz en la Mesh del servidor XS, o como repetidor en la Mesh (stand alone Mesh repeater) o MP.

Existe un dispositivo llamado Repetidor Solar, que consta de una active antenna integrada con una celda solar y una unidad de batería. Se trata de un dispositivo para trabajar en exteriores, de bajo costo, que permite que se extiendan fácilmente las redes Mesh.

La unidad requiere alrededor de 200mA con una tensión de +5 V para un funcionamiento normal. Si está conectado a un puerto USB (y no sólo una fuente de alimentación), requiere un driver específico que cargue el firmware y controle las funcionalidades de Mesh. Este driver está disponible para Linux (Libertas) y Windows.

Los primeros prototipos de la active antenna se construyó con piezas de ingeniería proporcionadas por Marvell. Estos primeros prototipos tienen un problema, no se reinician consistentemente, pero permitió el desarrollo de los Servidores de las Escuelas y pruebas de uso para su producción.^[9]

Al igual que las XO, sobre un mismo dispositivo de red, la active antenna cuenta con una interfaz de Mesh y una de 802.11b/g.



Figura 4.2: Prototipo de active antenna

Se han fabricado algunas centenas de estas radios. Éstas no tienen ninguna aprobación reglamentaria, y son exclusivamente para fines de ensayo, como es el caso para proyecto Yacaré y del Plan Ceibal

La figura 4.3 muestra en detalle la placa de ingeniería Marvell y su conexión USB

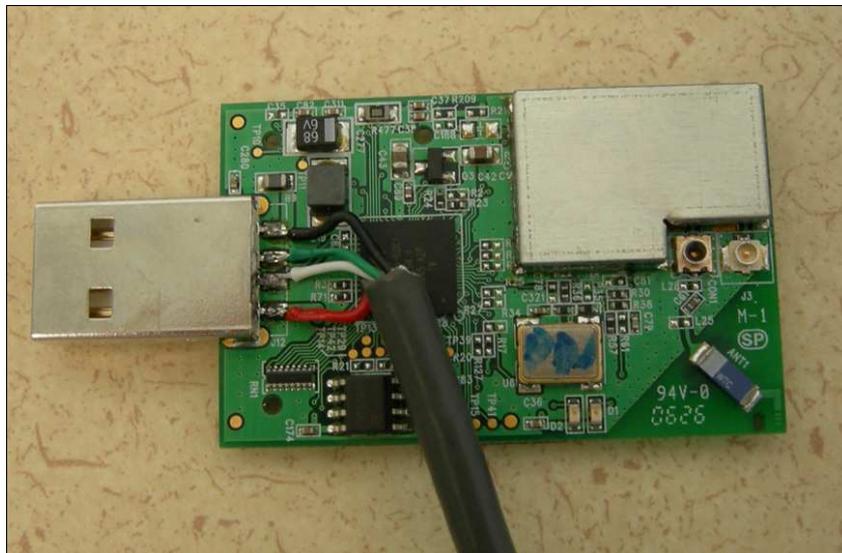


Figura 4.3: Placa active antenna

Para este modelo de prototipo toda la placa queda en el interior de la caja, a excepción de la antena externa que se extiende desde el conector que se puede ver a la derecha en la figura 4.3 y el cable de extensión USB.

4- 4 Mesh Portal Points

4- 4.1 Las XO como Mesh Portal Points

Para entender el funcionamiento de estos nodos, iremos viendo como se configuran las interfaces de las XO. Tomemos el caso de una XO que se asocia a un AP de Ceibal, se le asigna vía DHCP una dirección IP en su interfaz eth0 (802.11b/g), por ejemplo: la dirección 192.168.1.2 en la red 192.168.1.0/24, con ruta por defecto 192.168.1.1. Cuando logra la conexión a nivel IP con la red externa, la XO se asigna un IP en la Mesh (actualmente por AutoIP). Luego inicia el servicio de servidor DHCP, que consiste básicamente en ofrecer la configuración IP a quienes la soliciten mediante los mensajes DHCPREQUEST. Se está utilizando actualmente el espacio de direcciones 169.254.0.0/16 de IPv4. Los detalles del protocolo DHCP se pueden ver en el *anexo Protocolo DHCP - RFC 2131, capítulo 9- 4*.

Dado que la interfaz de Mesh y de 802.11b/g operan físicamente sobre el mismo dispositivo inalámbrico, ambas están siempre en el mismo canal, y no podemos decir que una es independiente de la otra. A nivel de capa de enlace, entre las interfaces se establece un bridge, o puente que traduce tramas de 802.11s a 802.11b/g.^[7]

4- 4.2 Búsqueda de Portales

La búsqueda de Portales puede realizarse de dos formas: enviando MPPREQ y escuchando MPPREPLY y procesando los datos o enviando DHCDISCOVER y escuchando DHCPOFFER. El primer camino implica una manipulación del hardware que es bastante inestable, ya que los protocolos son modificados en las imágenes de OLPC. Por esto es que elegimos basar nuestra búsqueda de portales en el protocolo DHCP, siendo que es más sencillo al ser éste un protocolo con muchos años de desarrollo.

Frente a cambios en los protocolos a nivel de capa de enlace, es más difícil realizar modificaciones. Si la configuración IP se realiza por otro protocolo, o se cambia a IPv6, el cambio en el software es más sencillo.

Hasta la versión B2 de las XO, todos los portales usaban una dirección de Multicast para recibir pedidos DHCPREQUEST. Las XO clientes tomaban la IP que le asignaba el Portal, lo que producía que en ambientes con muchos MPP y clientes, se dieran conflictos de direcciones IP. Esto quedó solucionado con la inclusión de AutoIP.

Llamaremos cliente Mesh al MP que se quiere asociar a un MPP. Luego de que un cliente envía el mensaje DHCPDISCOVER en la Mesh, el Portal responde con un DHCPOFFER. En este mensaje desde el MPP son enviados los siguientes datos:

- Gateway: su IP en la red Mesh
- IP asignada al cliente: un IP de la red 169.254.0.0 / 16

- IP del servidor DNS: envía al cliente el IP del servidor DNS que obtuvo al conectarse a la red WiFi.
- Mascara de red: 255.255.0.0

El cliente sigue el flujo de la figura 4.4 y toma una configuración IP como es explicitado en el protocolo DHCP, salvo que descarta la dirección que le envía el MP y se asigna una por AutoIP. Luego si, agrega en su tabla de rutas la red, y define su ruta por defecto como el IP del portal en la Mesh. El Network Manager de las XO es el encargado de llevar adelante la configuración mencionada.

El Portal Mesh, trabaja tanto a nivel de capa de enlace como capa de red (IP). Si la situación es como la que vemos en la figura 4.4, el MP intermedio no participa a nivel IP en la comunicación entre el MP cliente y el MPP.

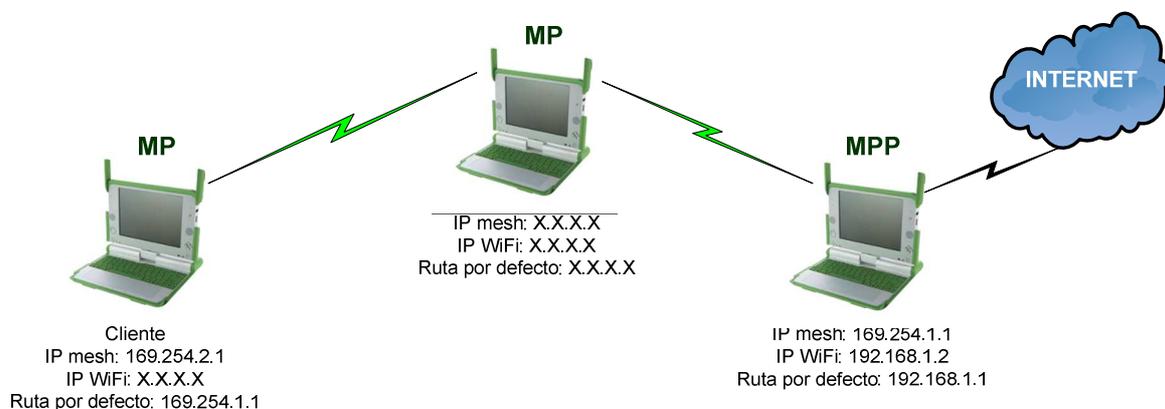


Fig: 4.4: MP y MPP

Cuando el MP cliente genera tráfico hacia la red externa (nube Internet en este caso), mira en su tabla de rutas IP el próximo salto y se la envía al MPP. Luego a nivel de capa de enlace, busca en la forwarding table el mejor Mesh Path hasta el MPP, y envía la trama vía el MP intermedio. El MPP puentea las tramas, y a nivel IP, realiza NAT y encamina el tráfico hacia Internet como si fuera generado por él. El camino desde la red externa hacia el cliente es exactamente el inverso.

4- 4.3 El módulo pydhcplib

Para crear, enviar y recibir mensajes del protocolo DHCP (ver anexo *Protocolo DHCP - RFC 2131, capítulo 9- 4*), utilizaremos la biblioteca *pydhcplib* de Python. Enviaremos mensajes DHCPDISCOVER creados según las especificaciones de la RFC 2131, y luego recibiremos e interpretaremos las respuestas DHCPOFFER utilizando todas las herramientas de la biblioteca. Al modificar varios de los scripts de la biblioteca *pydhcplib* preferimos no instalar el módulo, sino que lo ejecutamos localmente. Esto evita que en caso de tener el módulo ya instalado en el PC, éste no sea alterado por nuestra versión y así evitamos generar conflictos con otras aplicaciones que lo utilicen.^[25]

4- 5 Módulo Mesh

4- 5.1 Objetivos y funcionamiento

Los objetivos del módulo son los siguientes:

- Realizar escaneos en búsqueda de portales
- Buscar el número de saltos hasta cada Portal
- Guardar los datos de cada portal (direcciones MAC, IP y saltos)

4- 5.2 Búsqueda de portales

La búsqueda de portales es una controlada por el método *test_mesh* de *modulo_mesh.py*. No es posible cambiar la interfaz de Mesh de canal directamente, este proceso se debe realizar por medio del cambio de canal de la interfaz 802.11b/g de la active antenna. Por esto es que debimos tomar como parámetro para la búsqueda esta interfaz. Otros parámetros utilizados en el método adquirir es la cantidad de DHCPDISCOVER que queremos mandar y la cantidad de DHPPOFFER que queremos procesar por adquisición.

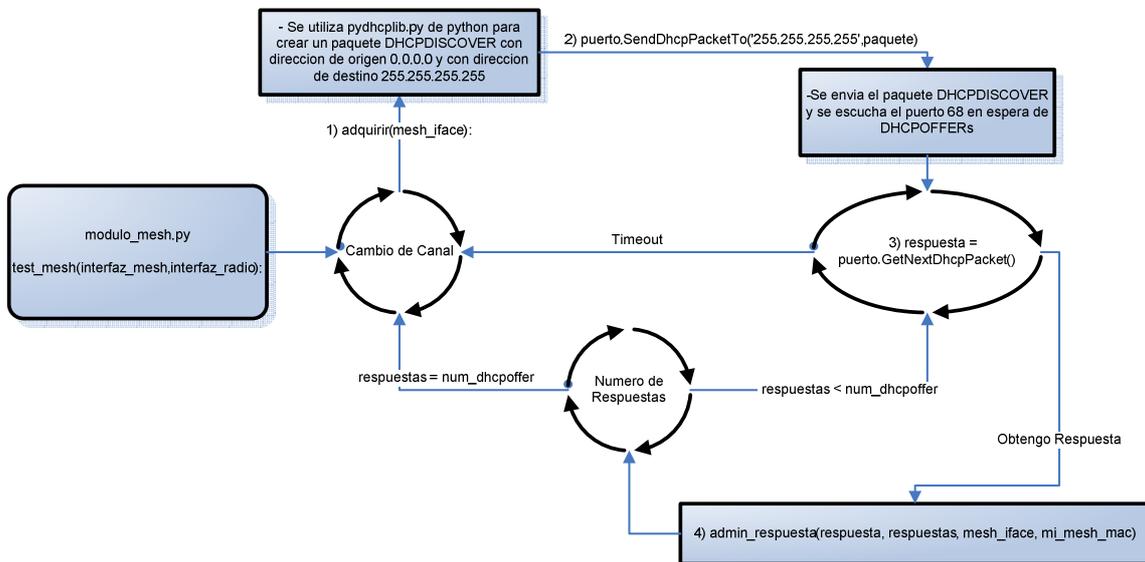


Fig 4.5: Diagrama de flujo

El método adquirir realiza la misma rutina en los canales 1, 6, 11. Crea un DHCPDISCOVER con dirección IP de origen 0.0.0.0 y lo envía con destino a broadcast 255.255.255.255 y el sistema queda escuchando el puerto 68 en espera de DHCPOFFER. Para crear el paquete realizamos el paquete *yhcp.py*, donde se setean las opciones y encabezado del mismo.

Creamos una funcionalidad administradora de respuestas (`admin_respuesta`), método clave de este módulo, encargada de procesar los DHCP OFFER recibidos. En principio chequea que efectivamente el paquete haya sido enviado a nosotros mediante el campo `chaddr` correspondiente a la dirección MAC del cliente (nosotros) y el `xid` (identificador de la comunicación). Toma el IP del servidor del header, luego envía consultas ARP (ARPing) para que el servidor DHCP le diga su dirección MAC.

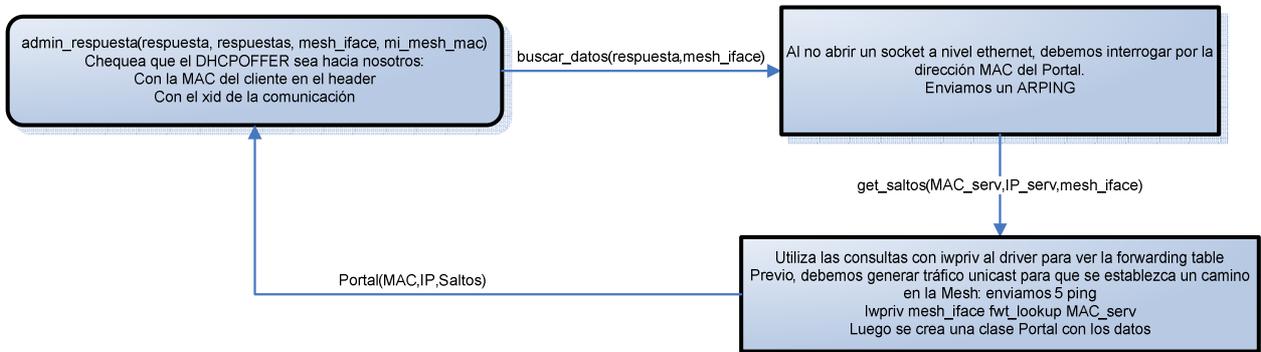


Fig 4.6: flujo para administrar mensajes DHCP OFFER

Para conocer el número de saltos hasta el portal, debemos generar tráfico entre servidor y cliente, ya que hasta el momento únicamente habíamos generado tráfico hacia la dirección de broadcast. Por este motivo, se envían algunos ping al servidor, y luego si se consulta la tabla de forwarding. Para realizar esto utilizamos el comando `iwpriv <Mesh_iface> fwd_lookup <MAC>`, donde `Mesh_iface` es la interfaz de Mesh, y `MAC` es la dirección MAC del destino que queremos alcanzar.

La salida del comando `msh0 fwd_lookup` es la siguiente:

```
00:17:C4:0C:EC:3C1 00:17:C4:0C:21:4F2 13 134 15 12 28 25 16 07 1543780 0 65
00:00:00:00:00:00
```

Donde:

1. 00:17:C4:0C:EC:3C = Dirección de destino
2. 00:17:C4:0C:21:4F = Dirección del próximo salto
3. 1 = Validez de la ruta
4. 13 = métrica
5. 1 = DIR, dirección de la ruta (guardan también la ruta hacia atrás)
6. 1 = Hop count, número de saltos
7. 0 = TTL, tiempo de vida. Es la cantidad de saltos restantes para la trama

Si no llega ningún DHCP OFFER al puerto 68 luego de un tiempo determinado, se corta la espera por timeout y volvemos al loop para enviar tantos DHCP DISCOVER como se haya elegido previamente. Si es el último, la rutina devuelve a la aplicación principal la lista con los portales descubiertos indexados por canal. ^[10]

5. Posicionamiento Geográfico por GPS

En este capítulo se presentan los conceptos relativos al posicionamiento geográfico utilizando un dispositivo GPS. Se presenta la arquitectura de este sistema y se describe el funcionamiento del receptor GPS en cuanto a la información que recibe de los satélites para la obtención de las coordenadas y sus modos de funcionamiento.

Se estudian los distintos sistemas de coordenadas y el concepto de punto Datum, llegando al sistema utilizado en Uruguay, ROU-USAMS mas conocido como Yacaré. Este sistema es el utilizado por Ceibal para la representación de las coordenadas en planos. Ceibal cuenta con el software QGIS, que es un sistema de información geográfica con una importante base de datos de mapas de Uruguay. De esta forma logramos desplegar los puntos obtenidos de los relevamientos de una localidad en mapas con una gran exactitud.

Estos conceptos están integrados en el módulo GPS desarrollado, el cual realiza funciones de control y consulta al receptor GPS mediante un cliente en el puerto del PC. Se implementó una función para la conversión de coordenadas al sistema Yacaré y posteriormente el método de proyección adecuado para poder almacenarlas también como coordenadas planas. De esta forma podemos desplegarlas correctamente en los mapas junto con los datos de interés de las redes WiFi y Mesh.

5- 1 Introducción al GPS

El Sistema de Posicionamiento Global (GPS) es un sistema de localización para determinar la posición en forma de latitud, longitud y altitud, de cualquier objeto en la superficie terrestre. Fue diseñado por el Departamento de Defensa de los Estados Unidos con fines militares para proporcionar estimaciones precisas de posición, velocidad y tiempo. Está operativo desde 1995, utiliza conjuntamente una red de PCs y una constelación de 24 satélites.

En el ámbito civil por razones de seguridad sólo se permite el uso de un subconjunto degradado de señales GPS. Sin embargo, se han encontrado alternativas para obtener una excelente precisión en la localización.

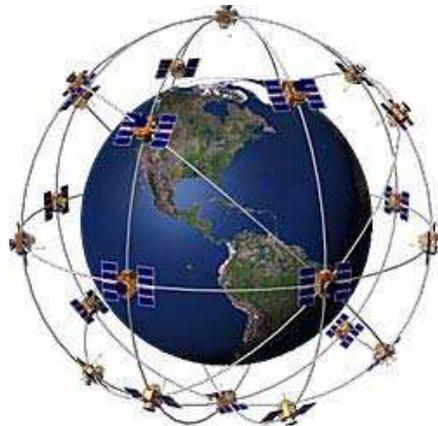


Fig 5.1: Constelación - Red de satélites GPS

Se ha implementando para las correcciones el sistema GPS Diferencial (DGPS), gracias a ello las aplicaciones han experimentado un gran crecimiento en el ámbito diario como los smartphones, que presentan el mayor crecimiento en ventas el año pasado, dispositivos de navegación portátil para automóviles (PND), hasta aplicaciones relativas al servicio de localización de ganado con receptores en los collares.

Como consecuencia existen actualmente más de 70 fabricantes de receptores GPS en el mercado mundial. Las ventas a nivel mundial del 2007 fueron del orden de 39 millones de unidades, duplicando la cifra alcanzada el año anterior.

5- 2 El Sistema GPS

5- 2.1 Arquitectura del sistema GPS

El sistema se descompone en tres segmentos básicos, los dos primeros de responsabilidad militar: *segmento espacio*, formado por 27 satélites GPS (24 operativos y 3 de respaldo) con una órbita de 20.300km de radio y un periodo aproximado de 12 horas; *segmento control*, que consta de cinco estaciones monitoras encargadas de mantener en órbita los satélites y supervisar su correcto funcionamiento, tres antenas terrestres que envían a los satélites las señales que deben transmitir y una estación experta de supervisión de todas las operaciones; y *segmento usuario*, formado por las antenas y los receptores pasivos situados en tierra. Los receptores GPS, a partir de los mensajes que provienen de cada satélite visible (señal de radio en 1517.42Mhz, llamada banda UHF), calculan distancias y proporcionan una estimación de posición y tiempo, además de contener un identificador de satélite, parámetros de su estado de funcionamiento y la fecha y hora con una altísima precisión. ^[11]

5- 2.2 Principios de funcionamiento del sistema GPS

El sistema GPS tiene por objetivo calcular la posición de un punto cualquiera en un espacio de coordenadas (x,y,z), partiendo del cálculo de las distancias del punto a un mínimo de tres satélites cuya localización es conocida. La distancia entre el usuario, receptor GPS y un satélite se mide multiplicando el tiempo de vuelo de la señal emitida desde el satélite por su velocidad de propagación.

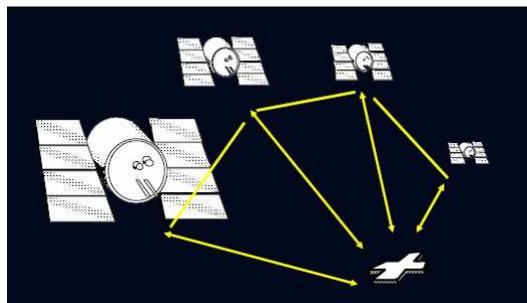


Fig 5.2: Estimación de posición

Para medir el tiempo de vuelo de la señal de radio es necesario que los relojes de los satélites y de los receptores estén sincronizados, pues deben generar simultáneamente el mismo código. La desviación en los relojes de los receptores GPS hace necesario un mínimo de cuatro satélites para estimar correctamente las posiciones.

5- 3 Sistema de Referencia de Coordenadas

La localización geográfica de un punto relevado se expresa en un determinado sistema de coordenadas respecto a un sistema de referencia geodésico.

Para el desarrollo del proyecto Yacaré surge la necesidad de utilizar un sistema de referencia adecuado para Uruguay, con lo cual es necesario convertir los datos obtenidos del GPS, del sistema de referencia global WGS84 a un sistema de referencia local para así poder desplegar la información en mapas de las localidades, ver *Apéndice Sistemas de Coordenadas*.

Para Uruguay se utiliza el sistema de referencia ROU-USAMS, conocido como sistema de referencia Yacaré, que se ajusta a las características geográficas de nuestro país.

5- 3.1 Sistema de referencia ROU-USAMS

Es el sistema de referencia elegido en el proyecto para desplegar la información en mapas de Uruguay utilizando el Sistema de Información Geográfico de Ceibal.

El ajuste al sistema de referencia Yacaré se hace determinando el *punto fundamental* donde se hace coincidir el geode con el elipsoide de referencia elegido, ver *anexo Sistema de referencia geodésico local, 9- 5.1*. El Datum es el conjunto de parámetros que definen este punto fundamental. Define entre otras cosas, la posición de origen y la orientación de las líneas de latitud y longitud del sistema de coordenadas.

Por definición, el Datum Yacaré queda definido por los siguientes parámetros:

- **Elipsoide de referencia:** Hayford 1924 (Internacional)
- **Punto fundamental:** Vértice I-Yacaré, coordenadas astro-geodésicas (definidas en el año 1939):
 - Latitud: 30° 35' 53.68" S
 - Longitud: 57° 25' 01.30" W
 - Acimut a: I-La Quisilla
- **Sistema de Proyección:** Gauss-Krügger modificado, el meridiano de contacto es 55° 48 W origen Y=0 en el Polo Sur y X=500km. al oeste del meridiano de contacto. ^[12]

Fue necesario realizar la conversión de los datos adquiridos desde el GPS al Sistema de Referencia Yacaré.

Primero realizamos una conversión del elipsoide WGS84 (sistema de referencia geocéntrico) al la elipsoide de Hayford 1924. Luego aplicamos una transformación para ajustar el sistema de referencia geodésico al punto Datum Yacaré.

Por último, para la representación de las coordenadas en un plano utilizamos el sistema de Proyección Gauss-Krügger (Transversa Mercator, ver *Apéndice Sistema de Proyecciones Cartográficas, capítulo 9- 5.3*), de esta forma podemos representar los puntos relevados en mapas de Uruguay utilizando el sistema de información geográfico como mencionamos anteriormente.

5- 4 Sistema de Información Geográfico de Ceibal

Un Sistema de Información geográfico (GIS) particulariza un conjunto de procedimientos sobre una base de datos no gráfica o descriptiva de objetos del mundo real que tienen una representación gráfica y que son susceptibles de algún tipo de medición respecto a su tamaño y dimensión relativa a la superficie de la tierra. A parte de la especificación no gráfica el GIS cuenta también con una base de datos gráfica con información georeferenciada o de tipo espacial y de alguna forma ligada a la base de datos descriptiva. La información es considerada geográfica si es mesurable y tiene localización.

La mayor utilidad de un sistema de información geográfico es la capacidad de construir modelos o representaciones del mundo real a partir de las bases de datos digitales. ^[13]

El Plan Ceibal utiliza el ArcGIS y el Quantum GIS (QGIS), este último de código libre en Python y utilizado por el proyecto Yacaré para la representación de la información adquirida de los relevamientos georeferenciada en mapas de Uruguay.

El QGIS cuenta con una importante base de datos con planos de Uruguay en el sistema de referencia de coordenadas Yacaré.

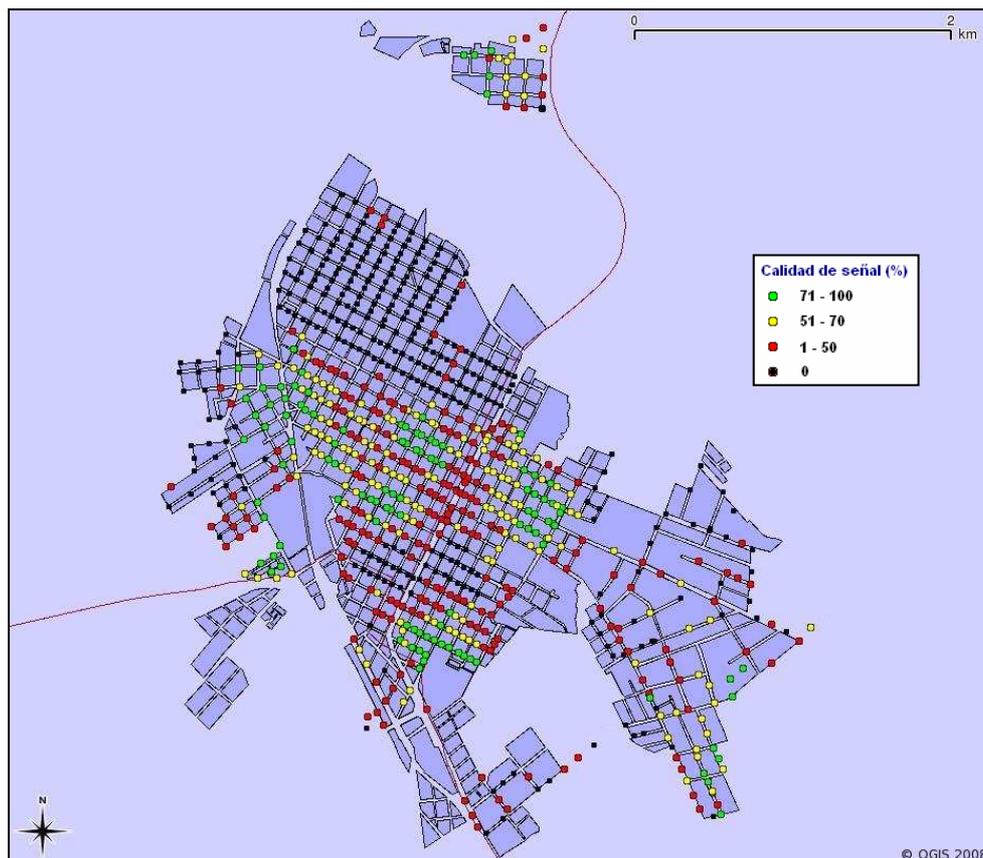


Fig 5.3: Mapa de Durazno en QGIS

La figura 5.3 muestra un relevamiento WiFi de la ciudad de Durazno donde los datos fueron desplegados utilizando el QGIS.

5- 5 Receptor GPS

5- 5.1 Elección del dispositivo GPS

Existe una amplia variedad de dispositivos GPS en el mercado.

Debimos enfocarnos en los requisitos necesarios para nuestro proyecto, por lo tanto optamos por un GPS que tuviese interfaz con el PC. No era necesario que posea un display, si que tenga buena exactitud, sea robusto y con cable largo para poder realizar los relevamientos desde un auto y moverlo con facilidad.

Nos concentramos en los GPS de Garmin, ya que contamos con bibliotecas para Python que interactúan con las sentencias NMEA, protocolo propio de los receptores GPS.

La figura 5.4 muestra dos modelos de la serie GPS 18 de Garmin, el USB y el PC



Fig 5.4: GPS 18 USB vs. GPS 18 PC

Ambos receptores GPS poseen las mismas características en cuanto a exactitud de posición, tiempos de adquisición, canales de GPS para la recepción de las señales desde los satélites y sensibilidad.

El GPS 18 PC posee interfaz DB-9 (RS-232) y alimentación de 12V, utilizando el adaptador del auto. La diferencia más importante es que el modelo GPS 18 USB cuenta con interfaz USB por lo que resulta más fácil conectarlo a cualquier PC, la alimentación es por el propio puerto USB con niveles de tensión entre 4.4 y 5.5V.

Por ello decidimos que el modelo USB era más adecuado a las necesidades del proyecto.

5- 5.2 GPS 18 USB de Garmin

Características del dispositivo GPS 18 USB

- 12 canales GPS para recepción de datos de los satélites.
- Posee GPS diferencial (DGPS), utiliza correcciones en tiempo real WAAS obteniendo así errores en la posición menores a 3 metros. La infraestructura para esta tecnología no está implementada en Uruguay.
- No requiere inicialización del usuario. Una vez conectado y en estado Fix, el receptor está listo para transmitir información.

- Parámetros configurables como posición esperada, hora y fecha actual y elegir tipo de posición Fix (2 dimensiones, 3 dimensiones o automático)
- Modo de navegación configurable por el usuario (2D o 3D Fix)
- Memoria flash interna, que permite al GPS guardar datos críticos como parámetros de las órbitas de los satélites, última posición conocida, fecha y hora.
- Posee una batería de respaldo recargable onboard para almacenar la hora real hasta 3 semanas
- Posee una base magnética, pesa 100 gramos y tiene un cable de 2 metros de largo para la interfaz USB, que facilita su integración en todo tipo de aplicaciones, en particular para realizar los relevamientos y colocarlo en el techo de un auto.
- Diseño resistente al agua, útil para los días de lluvia. [14]

En la figura 5.5 se muestran las características técnicas principales del receptor GPS

Protocolo	Sensibilidad en Rx	Canales	Readquisición	Int. de actualización	Exactitud posición
Garmin Binary Protocol	-165 dBW	12	<2 seg	1 muestra/seg	<15 mts

Fig 5.5: Características - Garmin GPS 18 USB

A cada una de las medidas de posición que nos ofrece el receptor GPS se la denomina *Fix*. Se clasifican en dos tipos:

- **2D Fix**

El sistema GPS permite obtener posiciones en el espacio tridimensional, pero para ello se requiere tres satélites y un reloj local preciso o, sencillamente y lo habitual, cuatro satélites a la vista.

Si solo disponemos de tres satélites, se puede obtener una posición (latitud y longitud) si suponemos que estamos en la superficie terrestre a nivel del mar. Si no es el caso, la posición obtenida puede ser muy imprecisa (por ejemplo, si estamos en una montaña).

Hay que tener en cuenta que la posición obtenida en estos casos sólo es válida si nos movemos en un entorno bidimensional, ya que recibiremos exclusivamente datos de latitud y longitud.

- **3D Fix**

Si tenemos a la vista cuatro satélites, podemos proporcionar una posición tridimensional precisa.

En teoría eso supone tres satélites a nivel del horizonte, separados por 120 grados, y otro satélite sobre nuestro cénit. Si hay más de cuatro satélites, se pueden utilizar también para mejorar la precisión del Fix.

Pero los satélites cercanos al horizonte son ocultados con mucha facilidad por cualquier obstáculo, además de estar sujetos a una mayor dispersión troposférica y a multirrayecto, por lo que los receptores GPS suelen exigir un mínimo de altura (entre 5 y 15 grados) sobre el horizonte para utilizar un satélite.

Hay que recordar que el sistema GPS, por diseño, proporciona una mayor precisión horizontal que vertical. El error asociado a la posición vertical suele ser en torno al doble de la horizontal.

Tiempos de adquisición del GPS

Los GPS tienen un tiempo de adquisición determinado, lo que habitualmente llamamos posicionamiento después de un arranque en frío (cold start). Los satélites envían información con su ubicación y es almacenada en los receptores GPS, la cual se utiliza luego en las inicializaciones y determinar la posición. Se denomina constelación al conjunto de los satélites GPS, pero también es utilizado para denominar al conjunto de satélites de la constelación visibles desde una posición dada del receptor.

Información enviada por los satélites

- **Almanaque**

Es enviada de forma periódica por los satélites de la constelación, informando sobre ellos mismos y el resto de los satélites miembros del sistema, su estado, etc. Esta información suele variar con poca frecuencia de año en año.

- **Efemérides**

Se envía dando la posición precisa de los satélites GPS. Esta información sí cambia frecuentemente, siendo actualizada por las estaciones de seguimiento de la Tierra. Los parámetros orbitales de los satélites se van actualizando a medida que su movimiento se ve alterado por la atracción del Sol y la Luna, la diferencia de gravedad entre distintas zonas de la corteza terrestre, viento solar, etc. Un período de cambio típico sería de 4 horas. La colección de efemérides de toda la constelación se completa en 12 minutos y se guarda en el receptor GPS. ^[15]

Tipos de arranque para la adquisición inicial del GPS

- **Cold Start**

Si la unidad GPS tiene un almanaque válido (recordemos que la validez normal de un almanaque es de varios meses), posición inicial y tiempo, tan solo deberá recibir las efemérides de los satélites, el tiempo normal de adquisición para el modelo GPS 18 USB es de aproximadamente 45 segundos (transmitiendo a 50 bps y dependiendo de en qué punto de la transmisión se arranque el GPS).

Las efemérides de los satélites tienen una validez de varias horas, por lo que un encendido "cercano" de la unidad GPS puede necesitar un simple "warm start".

- **Warm Start**

Si la unidad GPS tiene un almanaque y unas efemérides recientes (menos de cuatro horas), necesita obtener una referencia de tiempo y sincronización con la transmisión de "espectro ensanchado" de los satélites. El tiempo de adquisición, en este caso, es de unos 7 a 15 segundos.

- **Hot Start**

Si la unidad GPS tiene un almanaque, unas efemérides actualizadas y un tiempo local razonablemente preciso (por ejemplo, acabamos de pasar por debajo de un techo), el tiempo de adquisición es muy bajo del orden de algunos segundos.

Las figuras a continuación muestran la constelación de satélites que ve el receptor GPS 18 USB. Corresponden a la aplicación XGPS de Linux incluida dentro del paquete *GPSD-Clients*.

En la figura 5.6, el dispositivo todavía no ha adquirido una posición. Todavía no ha encontrado a los satélites en gris, pero conoce donde deberían estar según la información almacenada.

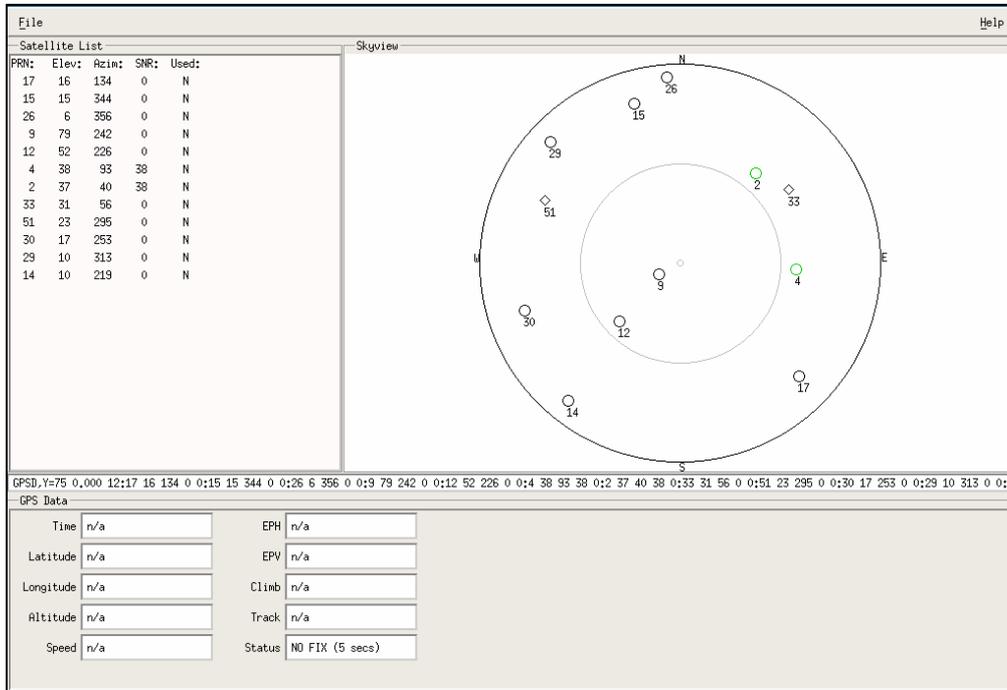


Fig 5.6: Situación No Fix

Luego de la inicialización se tiene una situación 3D Fix en el ejemplo de la figura 5.7. A partir de 3 satélites se puede obtener los parámetros de latitud, longitud, pero no altitud (no modo 2D Fix). En este caso hay 6 satélites y ya con un mínimo de 4 sería suficiente para el modo 3D Fix.

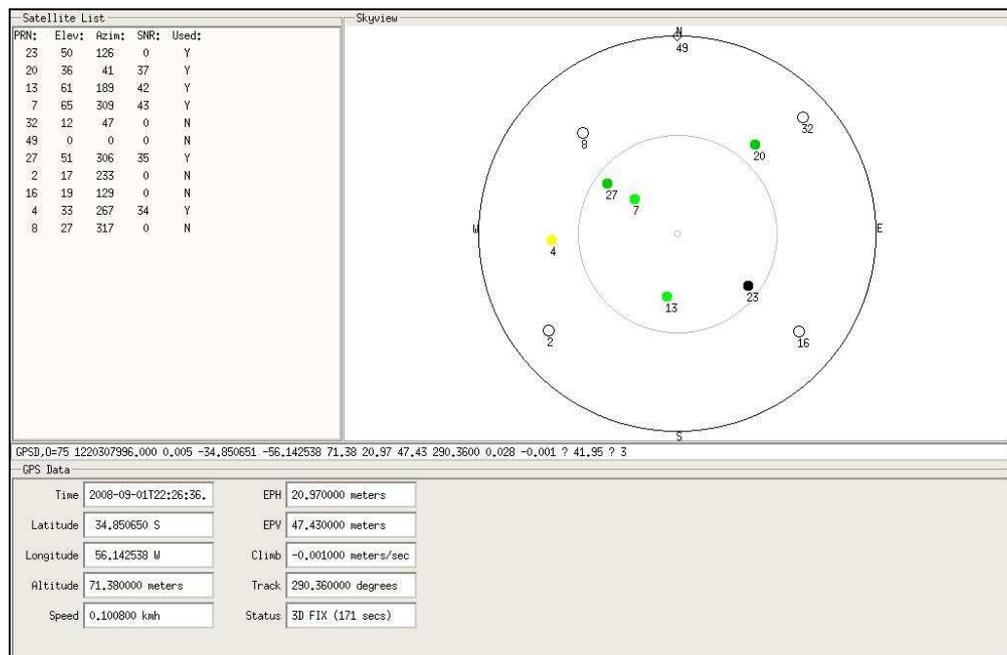


Fig 5.7: Situación 3D Fix

Los otros satélites son usados por el GPS 18 USB para corregir las medidas y así determinar la posición con una mejor exactitud. Los satélites que aparecen en amarillo corresponden a $SNR < 35$ y en negro es que se perdió solo por un instante la comunicación. Es normal que tengamos 6 o 7 satélites en una zona libre de obstáculos.

5- 5.3 Funcionamiento del receptor GPS

A continuación se describe el funcionamiento del receptor GPS 18 USB y el proceso para determinar la posición geográfica



Fig 5.8: Procedimiento del receptor GPS

1. La situación de los satélites es conocida por el receptor con base en las efemérides recibidas (5 parámetros orbitales Keplerianos), parámetros que como ya mencionamos son transmitidos por los propios satélites.
2. El receptor GPS funciona midiendo su distancia a los satélites, y usa esta información para calcular su posición. Esta distancia se mide calculando el tiempo que la señal tarda en llegar al receptor. Conocido ese tiempo y basándose en el hecho de que la señal viaja a la velocidad de la luz, salvo algunas correcciones que se aplican, se puede calcular la distancia entre el receptor y el satélite.
3. Cada satélite indica que el receptor se encuentra en un punto en la superficie de la esfera, con centro en el propio satélite y de radio la distancia hasta el receptor.
4. Obteniendo información de dos satélites se nos indica que el receptor se encuentra sobre la circunferencia que resulta cuando se intersectan las dos esferas.

5. Si adquirimos la misma información de un tercer satélite notamos que la nueva esfera solo corta la circunferencia anterior en dos puntos. Uno de ellos se puede descartar porque ofrece una posición absurda. De esta manera ya tendríamos la posición en 3D. Sin embargo, los satélites cuentan con relojes atómicos muy precisos y el del receptor es un oscilador de cuarzo y por lo tanto impreciso.
6. Como consecuencia los códigos no están sincronizados entre el receptor y los satélites, los dos puntos determinados no son precisos y se lo llama modo de posición 2D. Las distancias con errores debidas al sincronismo se denominan pseudodistancias.
7. La desviación en los relojes de los receptores añade una incógnita más que hace necesario un mínimo de cuatro satélites para estimar correctamente las posiciones.
8. Teniendo información de un cuarto satélite, eliminamos el inconveniente de la falta de sincronización entre los relojes de los receptores GPS y los relojes de los satélites. Y es en este momento cuando el receptor GPS puede determinar una posición 3D exacta (latitud, longitud y altitud). Al no estar sincronizados los relojes entre el receptor y los satélites, la intersección de las cuatro esferas con centro en estos satélites es un pequeño volumen en vez de ser un punto. La corrección consiste en ajustar la hora del receptor de tal forma que este volumen se transforme en un punto.

5- 6 Módulo GPS

5- 6.1 GPSD

Para la interacción con el receptor GPS utilizamos el paquete *gpsd* (Demonio de GPS).

Gpsd es un demonio de servicio que monitorea uno o más receptores GPS conectados a una computadora a través de los puertos serie o USB, haciendo que todos los datos de la ubicación, dirección y velocidad de los sensores se puedan consultar en un puerto TCP conocido de la computadora.

Con *gpsd*, múltiples aplicaciones GPS clientes pueden compartir el acceso a los GPS sin contención o pérdida de datos. También responde a peticiones con un formato que es más fácil de interpretar que el formato NMEA 0183, Garmin Binario Serial o USB. Los cuales están compilados en la instancia *gpsd*.^[16]

Para el Proyecto Yacaré el receptor GPS se conecta a través del puerto USB del PC. El demonio de servicio es capaz de reconocer el Protocolo Binario de Garmin utilizado por el GPS 18 USB y de determinar cual es la velocidad correcta del puerto.

Básicamente, el módulo GPS que desarrollamos se encarga, a partir del cliente *gpsd*, de adquirir las coordenadas desde el receptor GPS y guardarlas en 3 tipos de formatos que usaremos en distintas fases del proyecto Yacaré:

- **Referenciadas al sistema WGS84 en formato decimal.** Son las adquiridas por el receptor GPS
- **Referenciadas al sistema WGS84 en formato de grados-minutos-segundos.** Son útiles para una rápida visualización de las coordenadas y se despliegan en la interfaz gráfica con el usuario
- **Referenciadas al sistema ROU-USAMS (Yacaré).** Son representadas en un sistema de coordenadas planas, utilizadas para desplegar la información en los mapas del Sistema Cartográfico del LATU. Como ya lo mencionamos es el sistema de referencia de coordenadas que se utiliza en todo Uruguay

Para estas últimas, fue necesario implementar una función que permita corregir el par de coordenadas desde el elipsoide de referencia WGS84 al elipsoide de Hayford Internacional 1924 utilizada por el Datum Yacaré en dicho sistema de referencia.

Se implementaron funciones de control para detectar que el receptor GPS esté conectado y recibiendo coordenadas válidas de los satélites. El módulo implementa métodos capaces de iniciar o finalizar el cliente *gpsd*, con la opción de iniciar una conexión indicando el host (dirección IP) y puerto (TCP 2947, inicializado por defecto).

Gpsd acepta las siguientes opciones:

- **-n**
No espera a que se conecte el cliente para obtener información del GPS. La opción de espera puede ser útil para GPS seriales que aguardan en modo standby, sin utilizar energía, de esta forma esperan por la primer consulta al GPS.
- **-D**
Setea el nivel de fallos. Para nivel 2 y superior (número entero), *gpsd* avisa de sentencias entrantes y acciones como un reporte de error.

Consideramos que la mejor opción para inicializar *gpsd* es *-n* combinada con *-D2*, para así habilitar el monitoreo del flujo de datos del GPS y que comience a obtener información: `gpsd -D 2 -n /dev/ttyUSB0`

2- 6.2 Formato de los mensajes

Internamente, *gpsd* tiene una lista de dispositivos con los pathnames de los GPS conocidos.

Para la inicialización del demonio se debe especificar el dispositivo GPS y socket. El cliente estará escuchando en todo momento a este dispositivo.

El protocolo de consultas para clientes *gpsd* es muy sencillo. Cada consulta consiste de un carácter ASCII seguido de una línea nueva, y devuelve una línea con un texto de respuesta finalizando con CR/LF (retorno de carro y salto de línea).

Tanto consultas como respuestas, son seguidas de **%f** para un número flotante decimal y por **%d** para un número entero decimal.

Una respuesta que consiste del carácter **?** seguido de **=** significa que no hay formación válida disponible. Puede significar o que el receptor consultado está desconectado o que las consultas de posición, velocidad o tiempo están disponibles pero no fijadas.

Las respuestas pueden ser concatenadas y enviadas como un string; *gpsd* va a devolver entonces una lista de respuestas separada por comas.

Cada respuesta de *gpsd* va a comenzar con el string "GPSD" seguido de las respuestas.

A continuación se describen algunos ejemplos de mensajes intercambiados entre el receptor GPS y el cliente *gpsd*.

Devuelve la posición en formato de grados (latitud, longitud)	consulta: "P\n"
	respuesta: "GPSD,P=36.000000 123.000000\r\n"
Devuelve la hora UTC en el formato ISO 8601	consulta: "D\n"
	respuesta: "GPSD,D=2008-06-16T02:45:05.12Z\r\n"
Devuelve concatenado velocidad (en km/h) y altitud (metros) respecto al nivel del mar	consulta: "VA\n"
	respuesta: "GPSD,V=0.000000,A=37.900000\r\n"

Fig 5.9: Mensajes intercambiados con GPSD

Cuando el cliente está activo pero el receptor de GPS no responde, *gpsd* tratará de abrir el dispositivo GPS una vez por segundo. Por lo tanto, puede estar corriendo en background y tener un GPS que se desconectó y conectó nuevamente y así detectarlo.

Las consultas se envían a *GPSD* utilizando el método *query*, y se le pasa como parámetros los datos que estamos interesados en recibir. En concreto para Yacaré, le pasamos un string con varias consultas: *gps.query("MOQSY")*

Las respuestas corresponden a:

- **M**
Modo NMEA como "M=%d"
0 = Aún no se ha obtenido el modo
1 = No Fix
2 = Modo 2D (sin parámetro de altitud)
3 = Modo 3D
- **O**
Devuelve un reporte completo de tiempo/posición/velocidad. Cualquier campo que no tenga información disponible se envía como ?.
Si el modo es *No Fix*, entonces la respuesta es simplemente "O=?", de otro modo la posición y el timestamp son siempre enviados.
Los campos se describen a continuación y tienen el siguiente orden:

-etiqueta: identifica la última sentencia recibida. Para dispositivos NMEA es justamente el nombre de la sentencia NMEA (útil para identificar dispositivos)

-timestamp: segundos desde el "Unix epoch", UTC.

-error de tiempo: error estimado en timestamp (%f, en segundos, 95% nivel de confianza)

-latitud / longitud: valor igual que en la consulta "P" (%f, grados)

-altitud: valor igual que en la consulta "A" (%f, metros). Si no se está en modo 3D, se envía un valor estimado pero no debe tomarse en cuenta.

-error horizontal estimado / error vertical estimado: valor igual que en la consulta "E" (%f, metros)

-curso a nivel de superficie: track al igual que en la consulta "T" (%f, grados respecto al norte verdadero)

-velocidad a nivel de superficie: velocidad (%f, metros/segundos)

-pendiente: velocidad vertical al igual que en la consulta "U" (%f, m/s)

-error estimado para curso a nivel de superficie: (%f, grados, 95% nivel de confianza)

-error estimado para velocidad a nivel de superficie: (%f, m/s, 95% nivel de confianza)

-modo: modo NMEA al igual que en la consulta "M" (descripta anteriormente)

A continuación mostramos un ejemplo de un mensaje recibido a partir de la consulta:

```
"GPSD,O=75 1220306837.00 0.005 -34.850677 -56.142487 55.53 27.28 61.47
45.4637 0.003 -0.014 ? ? 3"
```

Fig 5.10: Respuesta a consulta 'O'

- **Q**
Devuelve "Q=%d %f %f %f %f %f": datos de los satélites utilizados en el último Fix (de aquí se desprende el número de satélites usados), y cinco número adimensionados llamados dilution-of-precision (DOP) que son: esférico, horizontal, vertical, tiempo y geometría total. Son usados para establecer la geometría de los satélites, como factores para multiplicar el UERE estimado (user error, en metros con un cierto nivel de confianza debido al retardo de la ionosfera, recepción por multicamino, etc.) para obtener los valores circulares de error en metros (o segundos) con el mismo nivel de confianza.
- **S**
El status NMEA como "S=%d"
0 = No Fix
1 = Fix
2 = Fix corregido por DGPS
- **Y**
Devuelve **Y=**, seguido de una sentencia tag, seguido por un timestamp (segundos del "Unix epoch", UTC) y un número hasta 12, seguido por los datos de los satélites, elevación/azimut (elevación como un entero en formato %d en el rango 0-90 y azimut también como un entero como %d en el rango 0-359), nivel de señal en dB, y un 0 o 1 si el satélite fue o no usado en este último Fix. Cada número es seguido de un espacio.

La biblioteca *gps.py* también nos ofrece métodos para poder consultar el estado del receptor GPS una vez que se actualizaron los datos con las respuestas que se describieron anteriormente.

Podemos consultar si está recibiendo de los satélites (STATUS=1, si hay datos válidos), es útil para poder sensarlo cada un periodo de tiempo determinado y asegurarnos que se están recibiendo coordenadas válidas. En el caso que esté implementado el sistema de correcciones DGPS (*ver apéndice GPS Diferencial*) la respuesta es STATUS=2. Actualmente no existe a nivel nacional una infraestructura para DGPS, pero se podría llegar a implementar en Montevideo y algunas ciudades.

También implementamos otro método que devuelve si el modo es 2D o 3D, utilizando *fix.mode*, y podemos consultar por el método *satellites_used* el número de satélites visibles por el receptor, los cuales son utilizados para mejorar la exactitud de las coordenadas recibidas.

De esta forma, obtuvimos un módulo capaz de extraer los datos de interés del receptor GPS de forma confiable ya que implementamos funciones que monitorean el correcto funcionamiento del dispositivo y que los datos que se reciben en los mensajes son válidos. Vale mencionar que el módulo es escalable respecto a la información que se recibe del dispositivo GPS, simplemente realizando otras consultas y chequeando los datos de las respuestas con los métodos adecuados. También es flexible a la utilización de otros modelos de receptores GPS que soporte el cliente GPSD.

6. Interfaz gráfica de la aplicación

En este capítulo detallaremos a fondo la aplicación gráfica realizada, desde la cual se deciden y controlan los distintos tipos de relevamientos. También explicaremos el manejo que se realiza con los datos adquiridos para la presentación de resultados. Cabe mencionar que este capítulo detalla los diferentes casos de uso de la aplicación funcionando como un solo módulo consolidado, los diferentes submódulos serán descriptos más adelante de una manera mas especifica.

6- 1 Diseño de la interfaz

El diseño gráfico de la aplicación fue realizado por el diseñador de interfaces Glade 3.0. Glade es una herramienta que facilita al desarrollador el diseño de interfaces graficas gtk. Este programa guarda el diseño realizado en un archivo de formato XML que luego es llamado desde Python mediante funciones de las bibliotecas gtk y gtk.glade.

Los diferentes elementos que componen una interfaz gráfica son conocidos como widgets, luego de crearlos en Glade, debemos asignarle una función a cada una de las diferentes acciones que podemos realizar sobre estos. Dichas funciones serán creadas en código Python.^[17]

La aplicación tiene una interfaz gráfica orientada a las necesidades del Plan Ceibal, por lo que tuvimos en cuenta los principales eventos que deben ser desplegados en pantalla al usuario que realiza un relevamiento.

Si bien no era prioritario agregar complejidad, durante el transcurso del proyecto nos surgieron nuevas ideas y requerimientos sobre interfaz gráfica. Entendimos que la herramienta resulta más útil de la forma en que la creamos, con varias gráficas y tablas que brindan mucha información al usuario.

Como podemos apreciar en la figura 6.1, la aplicación está dividida en solapas y resultados en la parte superior de la vista y por otro lado los comandos de adquisición con la barra de estado del GPS en la parte inferior.

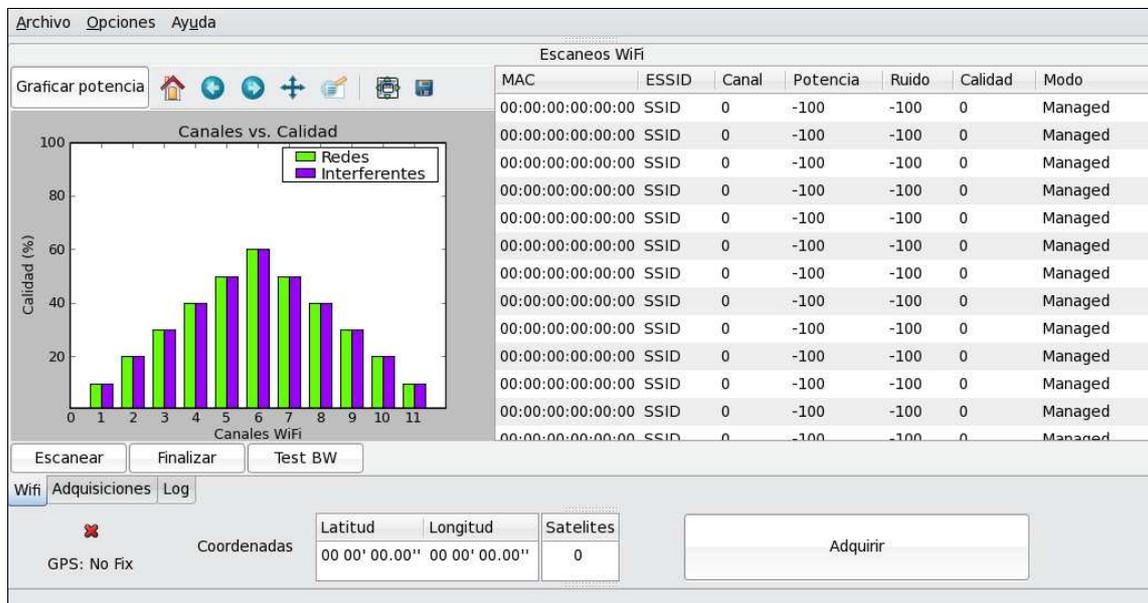


Fig 6.1: Pantalla principal Yacaré

La idea principal para este diseño, es que el usuario en todo momento debe tener la posibilidad de chequear el estado del GPS (activo o no) o de adquirir una muestra, mientras puede elegir entre las solapas para ver los últimas adquisiciones o los resultados de los escaneos de redes WiFi así como informarse sobre los diferentes eventos ocurridos chequeando el log.

Vamos a detallar las funcionalidades del panel principal, recorriendo las diferentes solapas y explicando en profundidad los eventos que son procesados por la aplicación.

6- 1.1 Barra de Estado GPS

El posicionamiento es un componente fundamental en la aplicación, cuando el GPS es activado en las opciones, el sistema comienza la etapa de inicialización del dispositivo. Como podemos ver en la figura 6.2, dependiendo del estado en que se encuentre el GPS, un icono indicara la disponibilidad de los datos del dispositivo. Los diferentes estados del GPS son explicados en el capítulo dedicado enteramente al dispositivo.

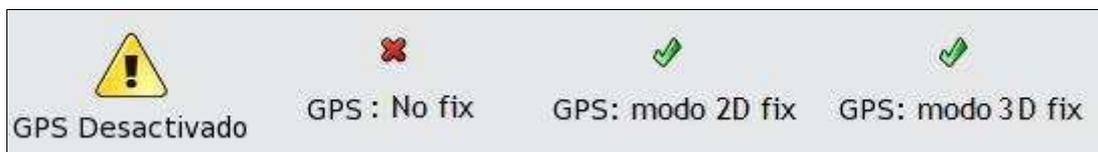


Fig 6.2: diferentes estados del GPS

Cuando la aplicación comienza a recibir coordenadas del GPS, éstas aparecen en pantalla refrescándose cada 1 segundo, lo que permite al usuario un control de los escaneos mucho mas preciso. La cantidad de satélites visibles al GPS es otro de los datos que se le brindan al usuario, el cual es un índice de la exactitud de las coordenadas.

Latitud	Longitud	Satelites
-34 25' 01.54"	-56 12' 53.01	4

Fig 6.3: Información desplegada por el GPS

6- 1.2 Solapa WiFi

Separaremos la descripción de esta solapa en las 2 funcionalidades principales, la presentación en tiempo real de los escaneos y las pruebas de ancho de banda.

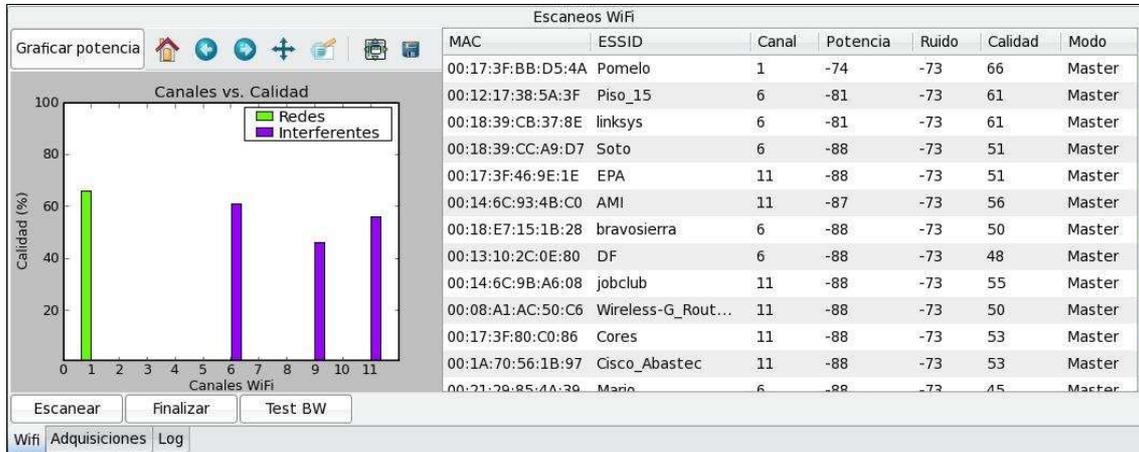


Fig 6.4: solapa WiFi

Escaneos

En esta solapa nos pareció interesante brindar la posibilidad de escanear en todo momento en busca de APs. De esta forma el usuario puede tener una idea de las redes que hay y tomar la decisión de adquirir. Al hacer click en el botón escanear (ver figura 6.5), comienza el escaneo y una tabla muestra la información completa, desplegando MAC, ESSID, IP, Canal, Potencia, Ruido y Calidad de cada AP encontrado.

Esta tabla (ver figura 6.5) nos da la posibilidad de ordenar los resultados de cada columna. Hay que tener en cuenta que estos son refrescados luego de cada escaneo completo, por lo que debemos detener el escaneo para una mejor lectura de la tabla, si el tiempo entre escaneos es corto.

MAC	ESSID	Canal	Potencia	Ruido	Calidad	Modo
00:17:3F:BB:D5:4A	Pomelo	1	-74	-73	66	Master
00:12:17:38:5A:3F	Piso_15	6	-81	-73	61	Master
00:18:39:CB:37:8E	linksys	6	-81	-73	61	Master
00:18:39:CC:A9:D7	Soto	6	-88	-73	51	Master
00:17:3F:46:9E:1E	EPA	11	-88	-73	51	Master
00:14:6C:93:4B:C0	AMI	11	-87	-73	56	Master
00:18:E7:15:1B:28	bravosierra	6	-88	-73	50	Master
00:13:10:2C:0E:80	DF	6	-88	-73	48	Master
00:14:6C:9B:A6:08	jobclub	11	-88	-73	55	Master
00:08:A1:AC:50:C6	Wireless-G_Rout...	11	-88	-73	50	Master
00:17:3F:80:C0:86	Cores	11	-88	-73	53	Master
00:1A:70:56:1B:97	Cisco_Abastec	11	-88	-73	53	Master

Fig 6.5: Tabla de resultados de los Escaneos

Para detener el escaneo con resultados en tiempo real, el usuario debe presionar *Finalizar*.

Hay que tener en cuenta que la funcionalidad escanear es independiente de la base de datos, por lo tanto para guardar los resultados WiFi debemos hacer click en *Adquirir*.

Dentro de las opciones de Yacaré, tenemos la posibilidad de que el sistema muestre en tiempo real una gráfica comparativa, entre la “mejor” red propia (barras en verde) y la “peor” red interferente (barras en violeta) en cada canal. Podemos intercambiar la gráfica y así poder desplegar *potencia de señal por canal* o *calidad de señal por canal*, como se puede ver en la *figura 6.6*. La aplicación diferencia las redes propias de las interferentes con la lista de ESSID conocidos que el usuario ingresa en la Ventana de Opciones.

Estas gráficas pueden ser guardadas en un formato de imagen en cualquier momento que el usuario lo desee.

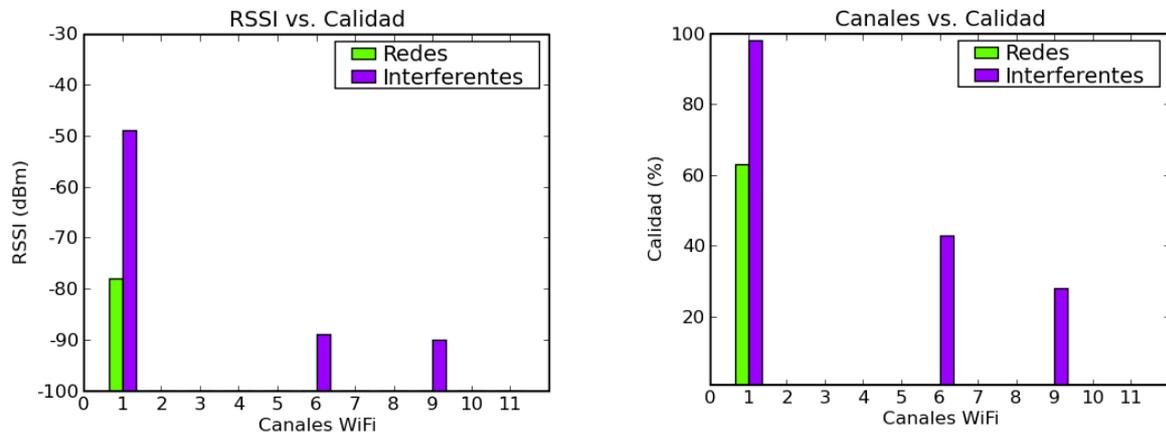


Fig 6.6: Gráfica potencia vs. canal y grafica calidad vs. canal

Esta opción está inestable hasta el momento de la entrega ya que consume los recursos del sistema de manera excesiva. En la etapa de desarrollo se comprobó su funcionamiento de manera correcta pero preferimos dejarlo opcional.

Test BW

Podemos acceder al test de ancho de banda al hacer click en *Test BW*, una ventana como la de la figura 6.7 se abre en pantalla. Para comenzar la prueba debemos ingresar el IP del servidor sobre el cual queremos realizar el test, ingresar el usuario y password para la autenticación y seleccionar el múltiplo de 256KB, para determinar el tamaño del archivo que queremos traficar.

Datos para prueba de ancho de banda

IP servidor: 169 . 254 . 35 . 226

Usuario: root

Password:

Tamaño de archivo (N x 256kB): 1

BW test

Prueba	BW(kbps)	Eficiencia(%)
Subida	0	0
Bajada	0	0

Aceptar Cancelar

Figura 6.7: Test BW

Al hacer click en *Aceptar*, comienza la rutina y la aplicación empieza por el test de subida, calcula la tasa de cada archivo parcialmente y lo imprime en pantalla. Inmediatamente comienza la segunda etapa correspondiente al cálculo de la tasa de bajada. Al término del test se muestran en pantalla los resultados parciales y luego son presentados los resultados finales junto con la eficiencia del ancho de banda.

Datos para prueba de ancho de banda

IP servidor: 192 . 168 . 1 . 17

Usuario: ceibal

Password:

Tamaño de archivo (N x 256kB): 3

Resultados

Prueba	BW(kbps)	Eficiencia(%)
Subida	52	20
Bajada	132	87

Aceptar Cancelar

Figura 6.8: Resultados del test BW

Cada vez que se realiza un test, es generada en la base de datos una entrada. Esta entrada es guardada simultáneamente con las adquisiciones. Es posible luego exportar estos datos desde la opción *Exportar* del menú de Archivo.

Al exportar los archivos BW, se generan los archivos *.txt* que son desplegados en el QGIS con el siguiente formato:

IP Serv	Subida	Bajada	Ef. Subida	Ef. Bajada	Lat	Long
IP 1	68.6511128594	157.38108032	70	85	-34.28849	-56.39379
IP 1	68.3509660568	153.098601009	72	79	-34.288482	-56.394143
IP 2	52.2195873103	160.413566729	69	87	-34.288419	-56.394479
IP 2	64.8367996792	142.10380297	71	80	-34.288129	-56.394455

Fig 6.9: Archivos BW

6- 1.3 Solapa Adquisiciones

Función Adquirir

Como podemos ver en la figura 6.10, dos tablas son creadas con solo hacer click en el botón *adquirir*, una presentando los resultados de WiFi y otra con los de Mesh. Estos relevamientos son guardados simultáneamente ya que se crea una nueva entrada en la base de datos con los resultados de WiFi y Mesh junto con la información de posicionamiento geográfico brindado por el GPS.

The screenshot shows a software interface with a menu bar (Archivo, Opciones, Ayuda) and a main window titled 'Adquisiciones'. The window is divided into two panes: 'Wifi' and 'Mesh'. Below the panes, there are tabs for 'Wifi', 'Adquisiciones', and 'Log'. At the bottom, there is a status bar with a warning icon and 'GPS Desactivado', a 'Coordenadas' section with input fields for 'Latitud' and 'Longitud' (both showing '00 00' 00.00'' and '00 00' 00.00''), a 'Satelites' field showing '0', and a large 'Adquirir' button.

Wifi							Mesh			
MAC	ESSID	Canal	Potencia	Ruido	Calidad	Modo	MAC	IP	Salto	Canal
00:17:3F:BB:D5:4A	Pomelo	1	-85	-96	50	Managed	00:17:C4:0C:D2:E4	169.254.9.128	n/a	1
00:12:17:38:5A:3F	Piso_15	6	-87	-96	46	Managed	00:17:C4:00:00:00	169.254.35.226	n/a	1
02:21:31:89:E9:31	olpc-mesh	11	-31	-96	98	Ad-Hoc	00:17:C4:XX:XX:XX	169.254.XXX.XXX	XX	6
00:1E:E5:34:EC:4C	PLESIR	11	-92	-96	33	Managed	00:17:C4:0C:D2:E4	169.254.9.128	n/a	11
							00:17:C4:0C:EC:3C	169.254.35.226	n/a	11

Fig 6.10: Solapa Adquisiciones

Manejo de las adquisiciones

Una de las condiciones de éxito de nuestro proyecto se refiere al manejo de los datos luego de relevados. Para dicha tarea utilizamos *shelve*, que es un tipo de base de datos Python. Permite esencialmente que cualquier instancia de objetos pueda ser guardada en la base. Las bases de datos *shelve* son de tipo persistente.

Shelve nos brinda la posibilidad de guardar las listas de adquisiciones realizadas por el usuario. Tiene comandos simples para acceder a la información almacenada así como para el ingreso de nuevos elementos en la misma.

El hecho de haber creado clases AP, Portal y Coordenada, posibilita guardar una instancia de cada una de ellos en cada adquisición. Cada vez que el usuario adquiere, una entrada en la base de datos temporal llamada *yacare.yac* es creada. Esta entrada contiene la adquisición en su completitud, es decir, los datos relevados Mesh, WiFi y Coordenadas. Para manejar los elementos utilizamos un número de adquisición que será la clave de la base de datos. En las figuras 6.11 y 6.12 podemos ver como queda estructurada las adquisiciones y pruebas de ancho de banda respectivamente. ^[18]

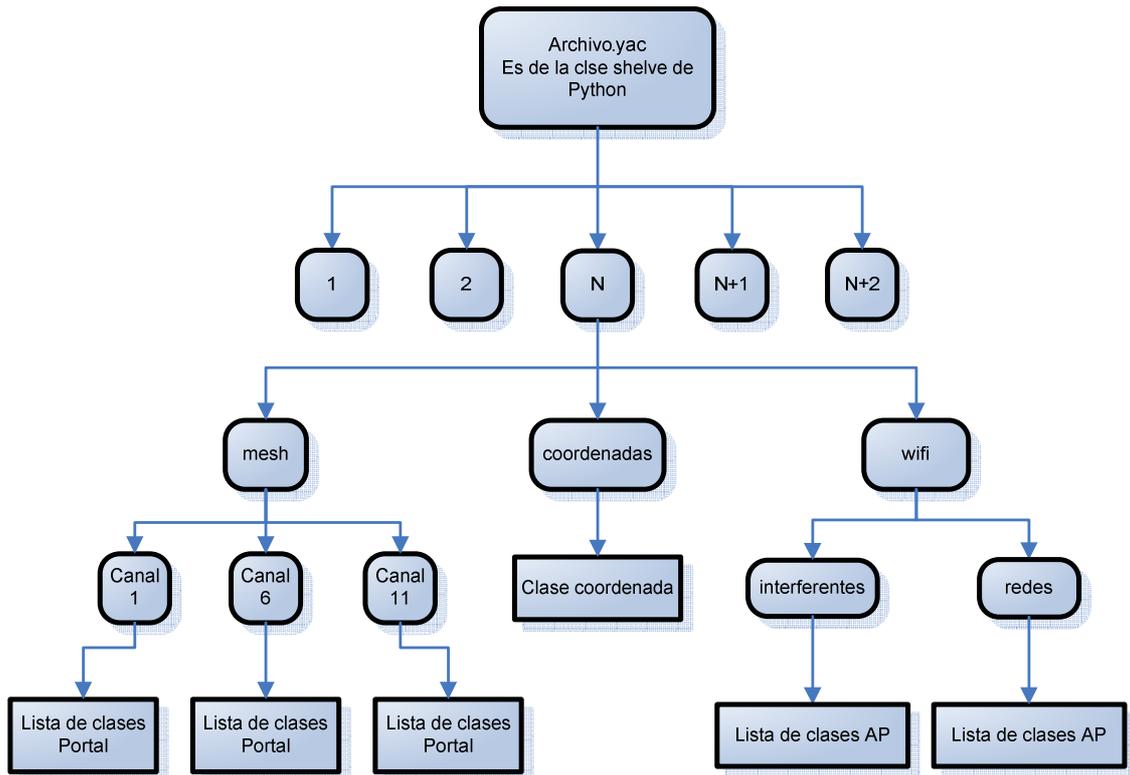


Fig 6.11: Diagrama de archivo de adquisiciones

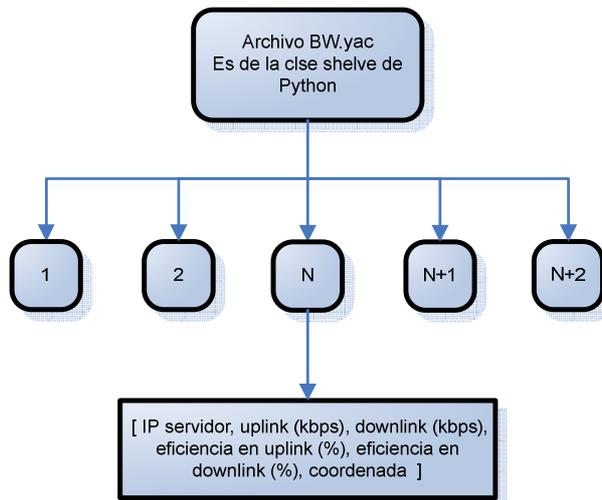


Fig 6.12: Diagrama de archivo de test BW

Guardar datos

Al término de una adquisición, el usuario tiene la posibilidad de guardar los relevamientos desde el menú de archivo, opción *Guardar*. Esta opción guarda la base de datos temporal en un archivo de extensión '.yac' propio de Yacaré. Después de guardado, la opción '*guardar como..*' es habilitada en el mismo menú de Archivo.

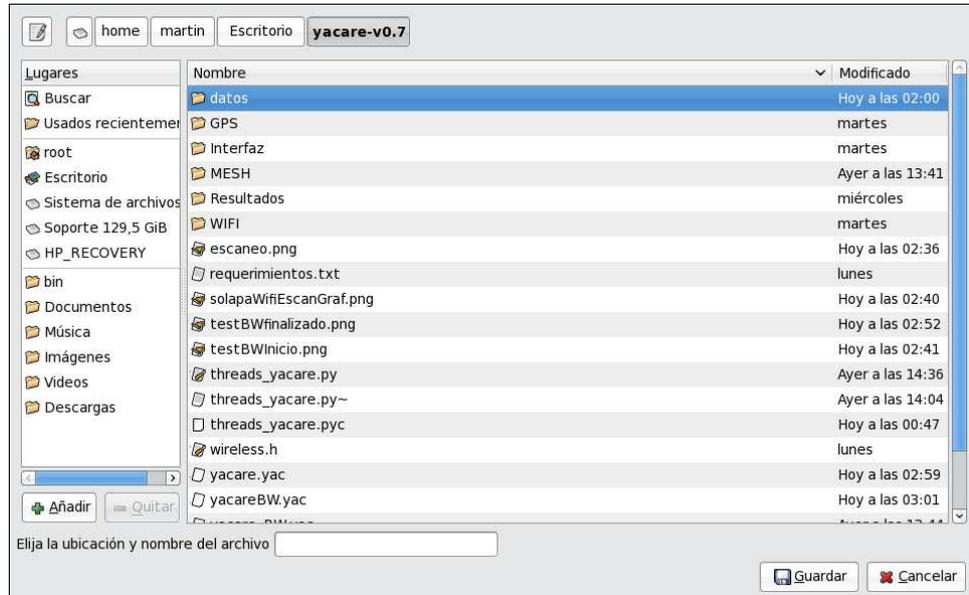


Fig 6.13: Guardar datos

Exportar datos

Luego de terminados los relevamientos fácilmente podemos exportar toda la información guardada por la base de datos *shelve* en archivos de formato *.txt*. Desde el menú de Archivo solo tenemos que seleccionar *Exportar* y un menú emergente nos pedirá seleccionar el archivo *'yac'* que queremos utilizar. Como vemos en la figura 6.14, podemos seleccionar en el menú emergente como queremos exportar los datos, podemos exportar cobertura WiFi o cobertura separada por AP y portales Mesh. Como detallamos anteriormente, los datos son exportados en archivos de texto plano tienen una estructura de tablas bien definida dependiendo del relevamiento.

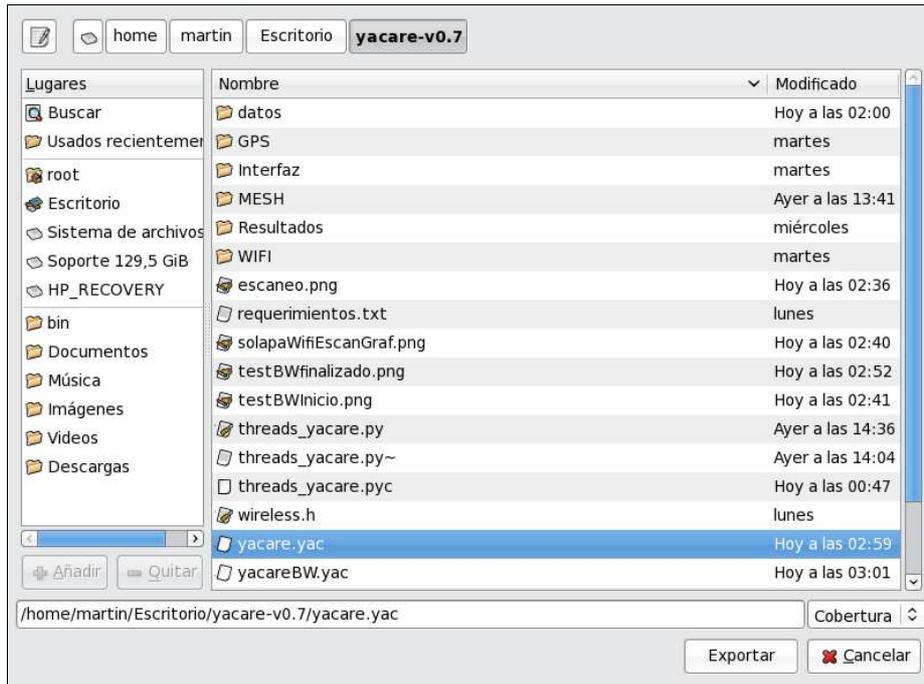


Fig 6.14: Exportar datos

Cuando el usuario elige exportar cobertura, 2 archivos *'txt'* son creados, uno con el nombre de la adquisición más *'_cobertura'* y otro con el agregado de *'_interferentes'*. En los cuales se separan las mejores redes por punto de adquisición y las peores redes interferentes.

Si el usuario elige exportar por AP, en el directorio donde se ubique el archivo fuente *'yac'* será creado un directorio por AP, en donde se guardará la tabla de texto plano con el nombre del archivo fuente más *'_portales'*.

6- 1.4 Solapa Log

Como la aplicación interactúa con varios periféricos, nos pareció interesante incluir un log donde el usuario puede chequear en tiempo real el estado de los dispositivos así como los mensajes devueltos por el sistema y los mensajes de error logueados por la aplicación.

En la figura 6.15 vemos como estos mensajes aparecen como entradas en una tabla, imprimiendo fecha y hora del mismo.

Log	
Tiempo	Eventos
02:59:02	Log yacare
02:59:02	Inicializando...
02:59:20	Comenzando el escaneo continuo
02:59:49	Finalizado el escaneo continuo
03:00:21	Comenzando test BW de subida con servidor 192.168.1.17
03:00:44	322 KB capturados
03:00:44	Eficiencia 79%
03:01:02	289 KB capturados
03:01:02	Eficiencia 88%

Fig 6.15: Solapa log

6- 1.5 Ventana de Opciones

Hay que tener en cuenta desde un principio que muchas de las funcionalidades de Yacaré están sujetas a las opciones que estén activas al momento de realizar las adquisiciones.

Para que el módulo completo, ya sea escaneo de redes WiFi o búsqueda de portales Mesh, funcione correctamente, es necesario que el usuario haya seleccionado las interfaces de WiFi y Mesh previamente. Si se va a relevar WiFi, en las opciones debe estar WiFi Activado. Para realizar la búsqueda de Portales debe estar Mesh activado en las Opciones así como correctamente seleccionada la interfaz de radio de la active antenna, ya que ésta cobra importancia a la hora de cambiar de canal.

El posicionamiento geográfico con GPS debe ser activado también desde la pantalla de Opciones para asociar los relevamientos a coordenadas.

Dependiendo del tipo de relevamiento que el usuario quiera realizar, las opciones deben ser correctamente seleccionadas previamente. Es importante tener un conocimiento completo de las opciones brindadas al usuario para realizar las adquisiciones.

Al hacer click en *Opciones*, la ventana que aparece en pantalla es la que podemos ver en la figura 6.16.

Fig 6.16: Ventana Opciones

A continuación detallaremos las diferentes opciones y las posibilidades que brinda cada una:

- **Número de escaneos:** esta opción indica la cantidad de escaneos que realiza la aplicación en cada canal en los relevamientos de WiFi.
- **GPS Activado:** indica si el GPS va a ser utilizado a la hora de relevar resultados. Si no está activado tampoco se desplegará en pantalla ningún tipo de información acerca del dispositivo. Esta opción desactivada fue pensada para los relevamientos indoor.

- WiFi Activado: cuando esta opción es activada los resultados WiFi son guardados en la base de datos al hacer click en *adquirir*.
- Mesh Activado: opción similar a la anterior, cuando ésta es activada los resultados Mesh son guardados al hacer click en *adquirir*.
- Interfaz WiFi: permite elegir de la lista de interfaces Wireless disponibles la que será utilizada para el escaneo WiFi.
- Interfaz Mesh: permite elegir de la lista de interfaces Wireless disponibles la que será utilizada para los relevamientos Mesh. Es imprescindible que la interfaz sea compatible con 802.11s
- Interfaz de Radio: permite configurar la interfaz WiFi de la active antenna, es muy importante que sea correctamente seleccionada ya que para las funcionalidades de Mesh es imprescindible.
- Essid Conocidos: al realizar los escaneos WiFi es importante conocer una lista de ESSID propios de manera de separarlos del resto, así se puede separar la información de los mismos a la hora de guardar los relevamientos y graficar en tiempo real.
- Tiempo entre muestras: esta opción permite al usuario decidir el tiempo entre escaneos en las adquisiciones WiFi.
- Graficar escaneos: al activar esta opción, junto con la actualización de la tabla al realizar escaneos, se realiza el grafico en tiempo real de calidad de la mejor red conocida y la peor red interferente por canal.

Si el usuario quiere solamente ver las opciones seteadas sin editarlas, también puede hacer click en *Información* dentro del menú de archivo, en donde se listarán en pantalla las opciones que están activas en este momento.

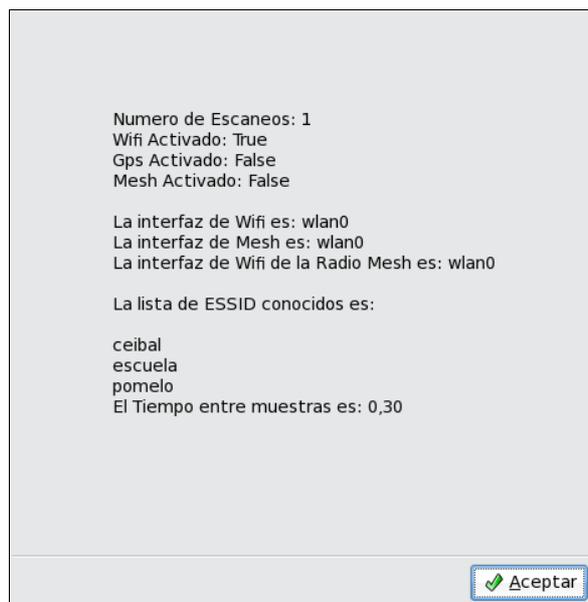


Figura 6.17: Listado de opciones activas

7. Resultados

A lo largo de las distintas etapas del proyecto Yacaré la metodología de ensayo y error fue fundamental para el desarrollo de la aplicación final, desde corroborar el buen funcionamiento de cada módulo hasta su integración en el programa principal de la interfaz gráfica.

7- 1 Etapa de desarrollo

Aún en etapa de desarrollo de la aplicación, se realizó un relevamiento de la ciudad de Durazno el cual fue hecho por Ceibal, donde se han incluido parte de los scripts generados en los módulos WiFi y GPS.

7- 1.1 Relevamiento en Durazno

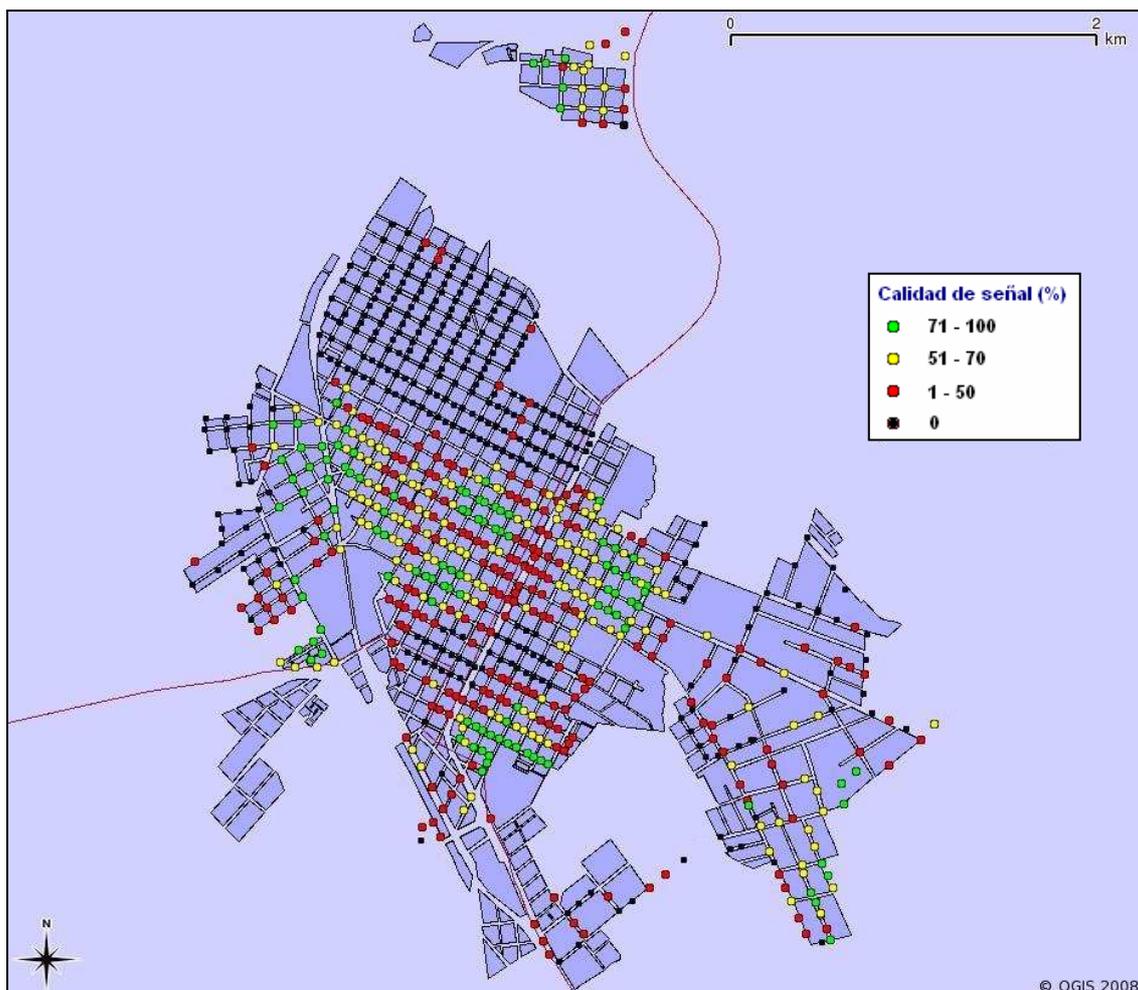


Fig 7.1: Mapa de cobertura de Durazno

En este caso, como muestra la referencia de la figura 7.1, se optó por desplegar el *Factor de Calidad*.

El relevamiento fue realizado con un laptop XO y un GPS Global Sat. Las zonas de cobertura mayoritariamente en verde corresponden a AP instalados en escuelas públicas de la ciudad.

7- 1.2 Comparativo de antenas

Esta prueba se basa en estudiar los niveles de potencia para un laptop utilizando la active antenna (radio Mesh) como interfaz inalámbrica de WiFi, en comparación a la señal de un laptop XO en recepción.

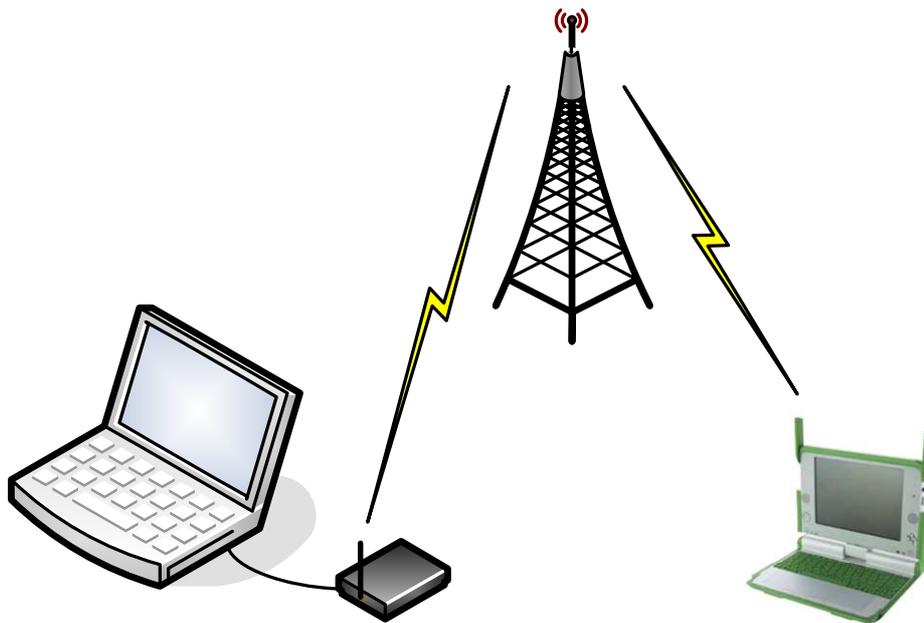


Fig 7.2: Comparación active antenna - antenas XO

Realizamos relevamientos con Yacaré y además generamos un script para ejecutar en un laptop XO. Al adquirir con la aplicación realizábamos escaneos en simultáneo desde la XO para poder luego comparar las muestras relevadas.

Los resultados de una de las pruebas son los que se muestran en la figura 7.3

AP (ESSID)	Potencia en Rx: XO / Active Antenna (dBm)																	
	1		2		3		4		5		6		7		8		9	
“Pomelo”	-75	-81	-78	-83	-72	-77	-76	-79	-78	-80	-76	-79	-80	-77	-78	-89	-80	-79
“Soto”	-83	-92	-84	-90	-84	-89	-82	-90	-82	-89	-89	-88			-83	-90	-86	-91
“Piso”					-89	-90			-89	-90			-89	-90	-86	-90	-87	-90
“WirelessG”	-90	-90	-81	-90	-85	-89	-88	-89					-87	-93				

Fig 7.3: XO vs. active antenna

Concluimos que en promedio la diferencia es de 5dBm mas para la XO en comparación a la laptop con la active antenna.

No hay datos de OLPC sobre la ganancia de las antenas integradas de las XO. Pero esperábamos que fuese mayor si consideramos 3dB de ganancia por diversidad en Rx y también que la ganancia de estas antenas es mayor a la antena omnidireccional que posee la active antenna (estimamos 2dBi para esta última).

En cuanto a los niveles de calidad de señal, éstos también dependen de la tarjeta WiFi utilizada. Se miden en la escala de 1 a 100 y están obviamente relacionados con la potencia de la señal recibida.

Por la experiencia adquirida y las pruebas realizadas podemos decir que el “peor caso” de calidad de señal queda determinado por la laptop con la active antenna, ya que tiene una menor potencia en recepción en relación a la XO como pudimos ver anteriormente.

De esta forma, aseguramos una cota inferior para los distintos valores de calidad de señal cuando realizamos los relevamientos con la active antenna y así poder garantizar niveles en cuanto a las conexiones de red WiFi para las laptops XO.

La clasificación para la calidad de señal queda entonces determinada en los siguientes niveles:

	71 - 100: Bueno
	51 - 70: Aceptable. Suficiente para la establecer conexión
	1 - 50: Malo. Muy difícil de establecer una conexión
	0: No hay señal

Estos valores son los que se utilizan actualmente en Ceibal para desplegar la información adquirida en mapas.

7- 2 Etapa de pruebas de la aplicación

Con el programa finalizado, se realizaron relevamientos en tres localidades del departamento de Florida, en Mendoza, Villa Cardal y Sarandí Grande. Villa Cardal tiene la particularidad de ser el pueblo piloto en el despliegue de Ceibal a comienzos del año 2007.

Para las pruebas se utilizó un laptop, el receptor GPS y la active antenna como interfaz de red inalámbrica para WiFi y Mesh. Vale destacar que los periféricos son ideales para este tipo de relevamientos. Ambos con cables lo suficientemente largos y el receptor GPS se puede colocar en el techo al poseer una base magnética y además es resistente al agua. En la foto de la figura 7.4 se puede ver el equipo necesario para el relevamiento.



Fig 7.4: Equipo para relevamiento

Se puede ver que también hay una interfaz de WiFi con una antena omnidireccional y un laptop XO que se utilizó para corroborar el buen funcionamiento de la aplicación en cuanto a la búsqueda de MPP y niveles de señal WiFi.

Una vez adquiridos los puntos de los distintos relevamientos, utilizamos el QGIS para desplegar los resultados en mapas. Exportamos los datos desde Yacaré en distintos tipos para analizar mapas de cobertura WiFi, cobertura por AP, redes interferentes y portales Mesh.

7- 2.1 Relevamiento en Mendoza

Los relevamientos se realizaron en zonas donde hay viviendas, muchas de las manzanas que aparecen en el mapa corresponden a campos.

En la figura 7.5 se puede ver la ubicación del AP en la torre de Antel a 25 metros de altura, donde también hay equipos de telefonía celular. El equipo es marca Wavion con una potencia de máxima de salida igual a 42.5dBm considerando beamforming.

El relevamiento de la red WiFi es el que se muestra a continuación en la figura 7.5



Fig 7.5: Mapa de cobertura de Mendoza

Todos los mapas exportados del QGIS de aquí en más poseen como escala de referencia "grados" que a los efectos de los mismos deberían ser "metros".

Los resultados son muy buenos, teniendo en cuenta las distancias que hay entre el AP y las zonas relevadas, donde hay más de 500 metros al punto más lejano.

Los niveles de potencia son aceptables y permiten una buena conexión para las XO, corroborado en el relevamiento.

7- 2.2 Relevamiento en Villa Cardal

Al igual que para el relevamiento de Mendoza, nos concentramos en la zona de viviendas.

En la figura 7.6 se puede ver la ubicación de los APs de Ceibal, junto con los resultados del relevamiento de cobertura de la red WiFi. Para el mapa se exportó cobertura para redes “conocidas”.

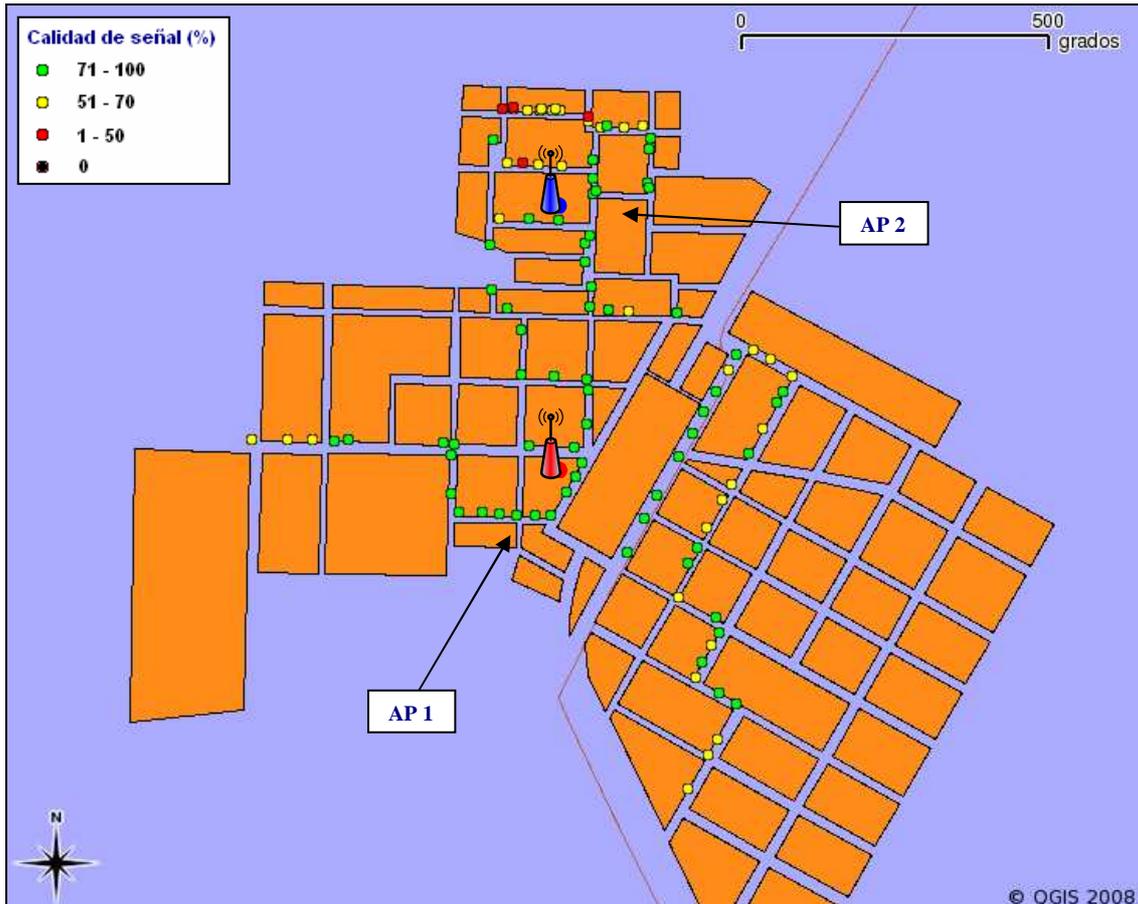


Fig 7.6: Mapa de cobertura de Cardal - Florida

El AP 1 está ubicado en una torre de Antel a 25 metros de altura, el equipo es marca Wavion al igual que en Mendoza mientras que el AP 2 corresponde a un equipo marca Meraki ubicado en un mástil a una altura de 5 metros con una potencia de salida de 18dBm.

Se pueden observar niveles de señal muy buenos y uniformes. Los puntos rojos son críticos y no se puede establecer conexión con una XO, pero también hay que considerar que son distancias de más de 400 metros al AP 1 Wavion.

Respecto al posicionamiento geográfico, se puede concluir una excelente precisión en las muestras obtenidas del receptor GPS, con un error de menos de 2 metros si comparamos puntos consecutivos.

A continuación desplegaremos los mapas exportando los datos de cobertura para cada AP y seleccionando los layers adecuados para mostrar desde el QGIS.

Cobertura AP 1

Como se puede ver en el mapa de la figura 7.7, la cobertura WiFi del AP 1 Wavion es casi la misma que la total brindada por el conjunto de los 2 APs, a menos de las zonas más lejanas donde la calidad de señal pasa a estar entre 51 y 70%.



Fig 7.7: Mapa de cobertura AP 1

Cobertura AP 2

Como se puede ver en el mapa de la figura 7.8, la cobertura del AP 2 se reduce a la esquina donde está ubicado el AP en el mástil y unas pocas cuadras más.



Fig 7.8: Mapa de cobertura AP 2

7- 2.3 Relevamiento en Sarandí Grande

A diferencia de las localidades anteriores, Sarandí Grande es más grande, posee una población mucho mayor y tiene 3 escuelas, por lo cual Ceibal consideró instalar 4 APs en distintos puntos de la ciudad.

Los equipos son MikroTik con potencia de salida 22dBm y están ubicados en los siguientes puntos:

- AP 1: Tanque de OSE
- AP 2: Escuela N° 38
- AP 3: Escuela N° 7
- AP 4: MEVIR

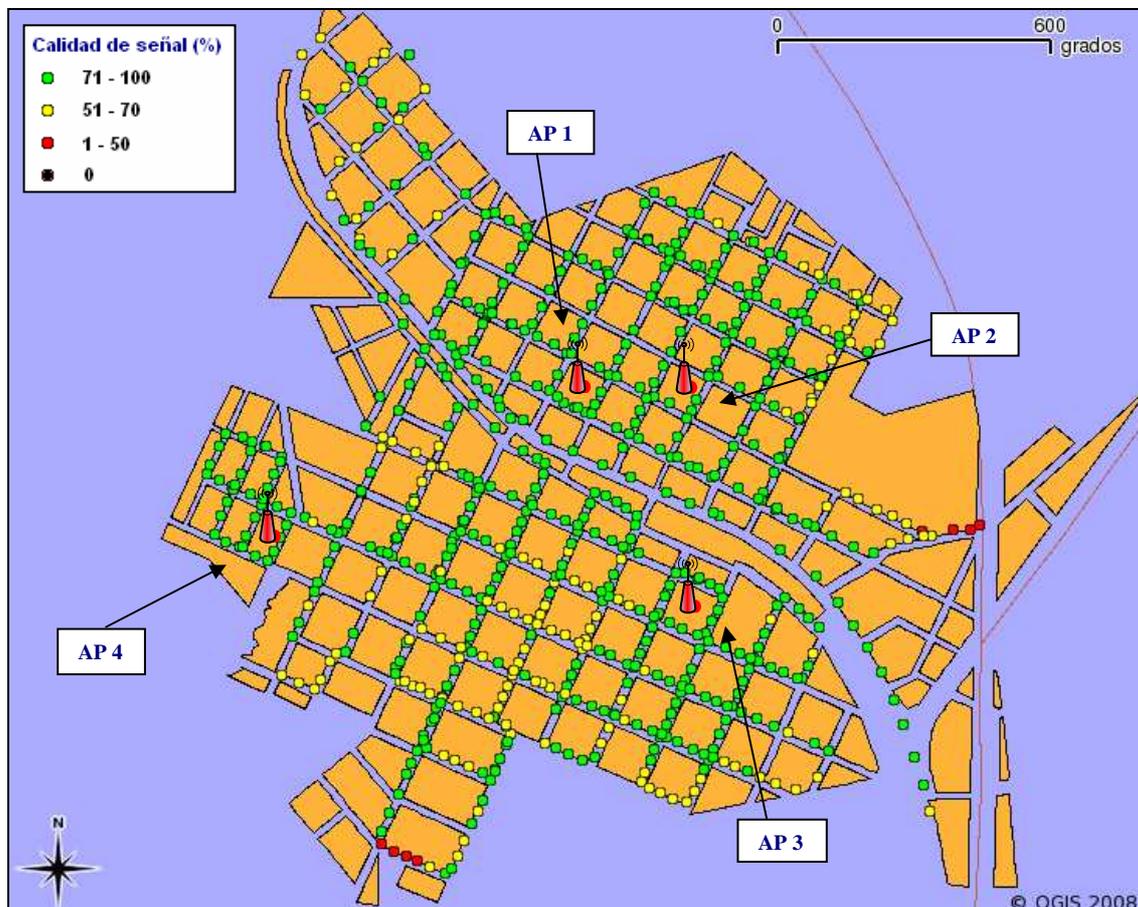


Fig 7.9: Mapa de cobertura WiFi de Sarandí Grande

Se aprecia una excelente cobertura en todo Sarandí Grande, debido a las características geográficas de la ciudad y una buena ubicación de los APs en los puntos más altos.

En los relevamientos se utilizó también una laptop XO para corroborar que los niveles de señal fueran coherentes con las muestras adquiridas, donde se observó la diferencia en los niveles de potencia analizada en 7- 1.2.

A continuación se muestran los mapas exportando los datos de cobertura para cada AP.

Cobertura AP 1

Está ubicado en el tanque de OSE a una altura aproximada de 20 metros en uno de los puntos más altos de la ciudad.

La figura 7.10 corresponde a la cobertura WiFi de un radio que posee una antena omnidireccional con 12dBi de ganancia.

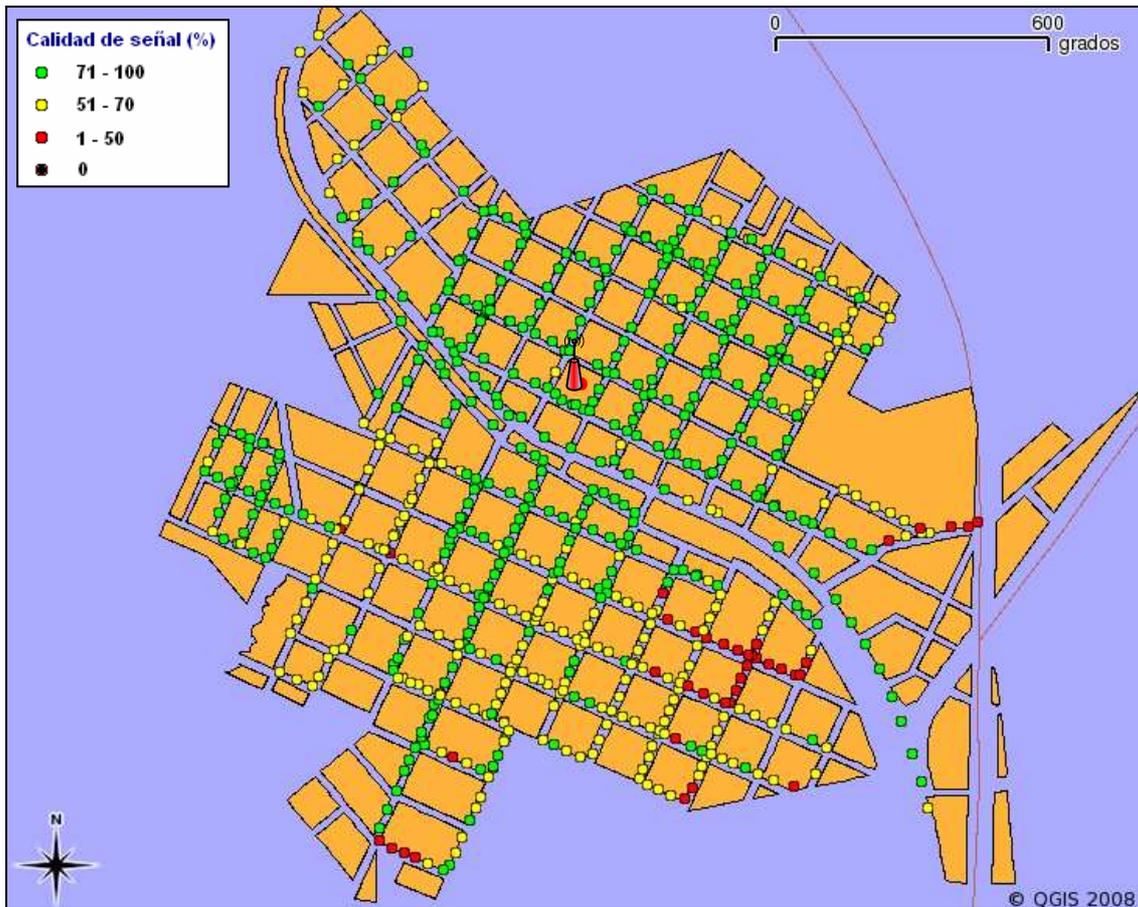


Fig 7.10: Mapa de cobertura AP 1 - antena omnidireccional

El área que cubre este AP es excelente con niveles óptimos de señal en el 80% de la zona.

En la figura 7.11 se puede apreciar la zona de cobertura que corresponde al radio del AP 1 que posee una antena sectorial con 12dBi de ganancia orientada hacia la zona de viviendas.

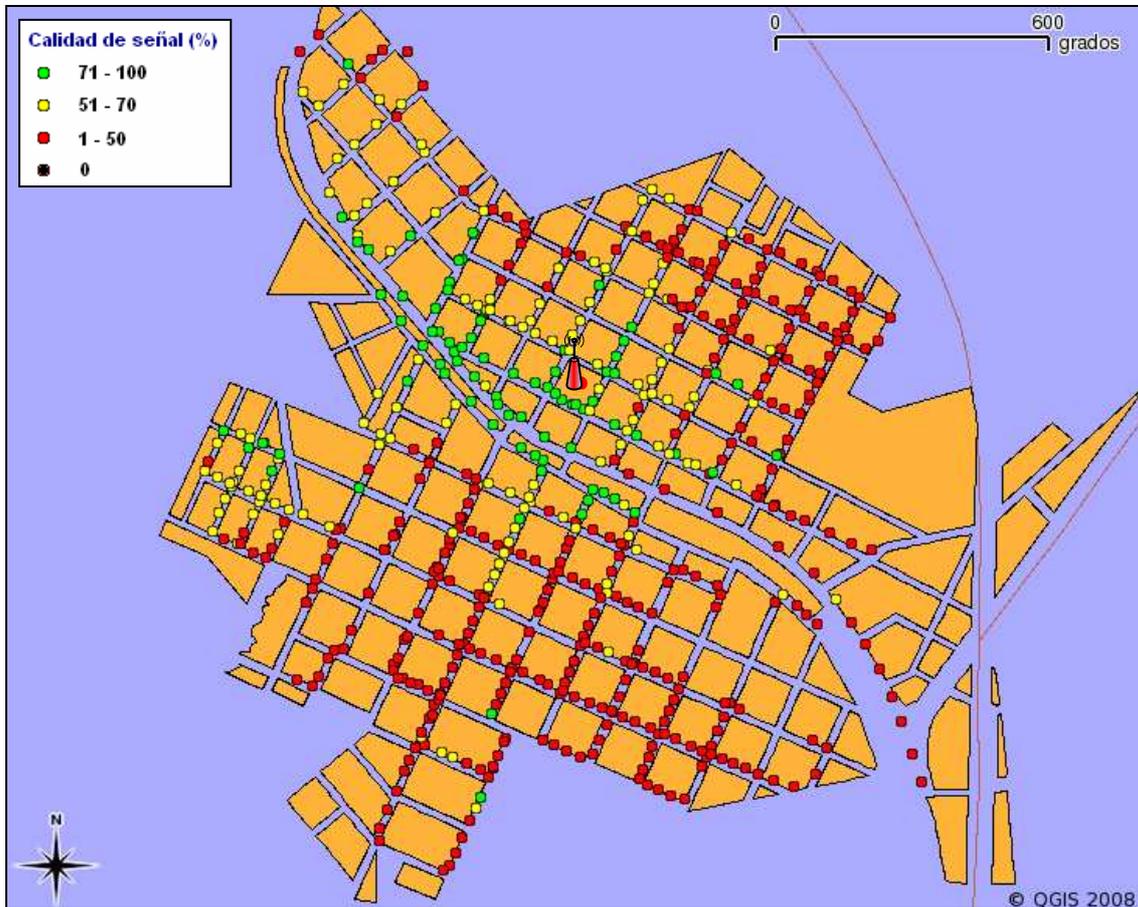


Fig 7.11: Mapa de cobertura AP 1 - antena sectorial

Del mapa de cobertura se concluye efectivamente el tipo de antena y que Ceibal buscó cubrir la zona oeste de viviendas de la ciudad.

Cobertura AP 2

Está ubicado en la escuela pública N° 38 y posee una antena omnidireccional de 12dBi.

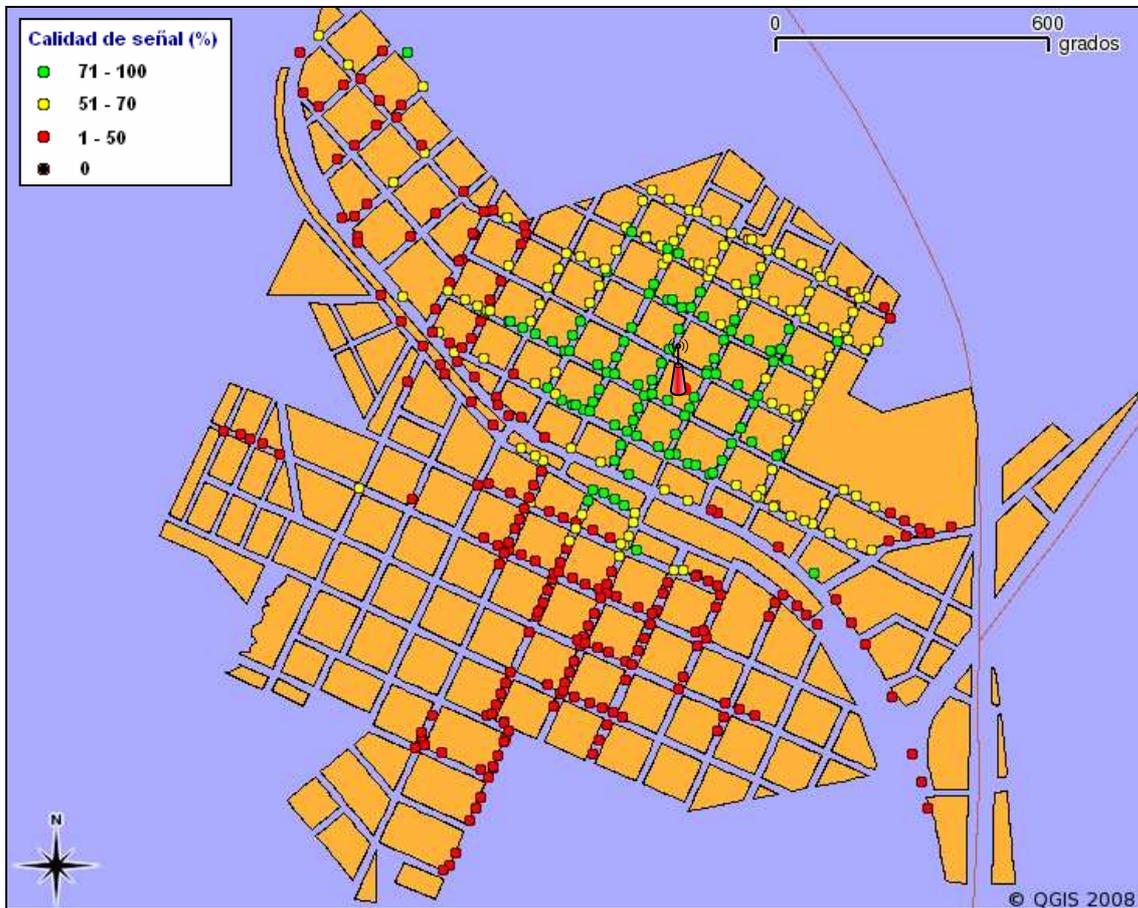


Fig 7.12: Mapa de cobertura AP 2 - Escuela N° 38

Cobertura AP 3

Está ubicado en la escuela pública N° 7 y posee una antena omnidireccional de 12dBi.

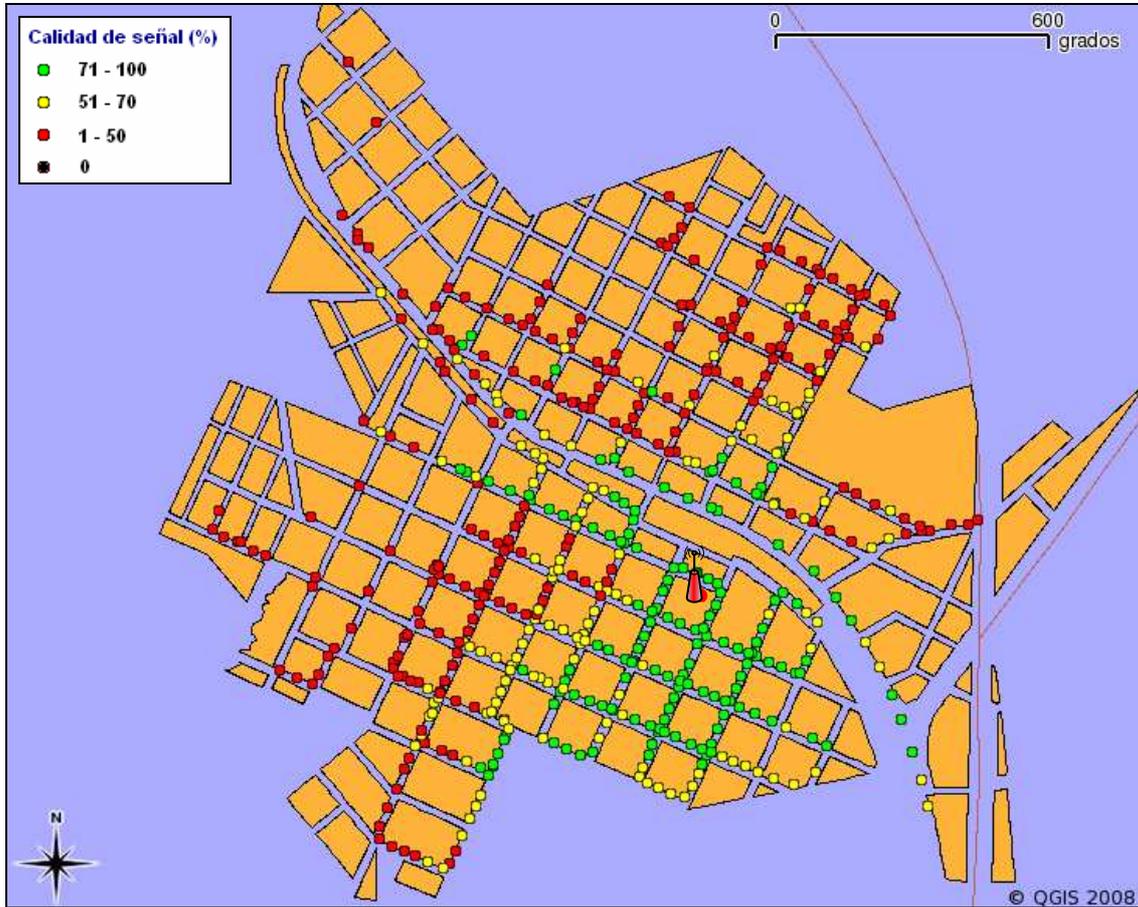


Fig 7.13: Mapa de cobertura AP 3 - Escuela N° 7

Cobertura AP 4

Está ubicado en el MEVIR. Posee una antena omnidireccional de 12dBi al igual que las antenas anteriores.

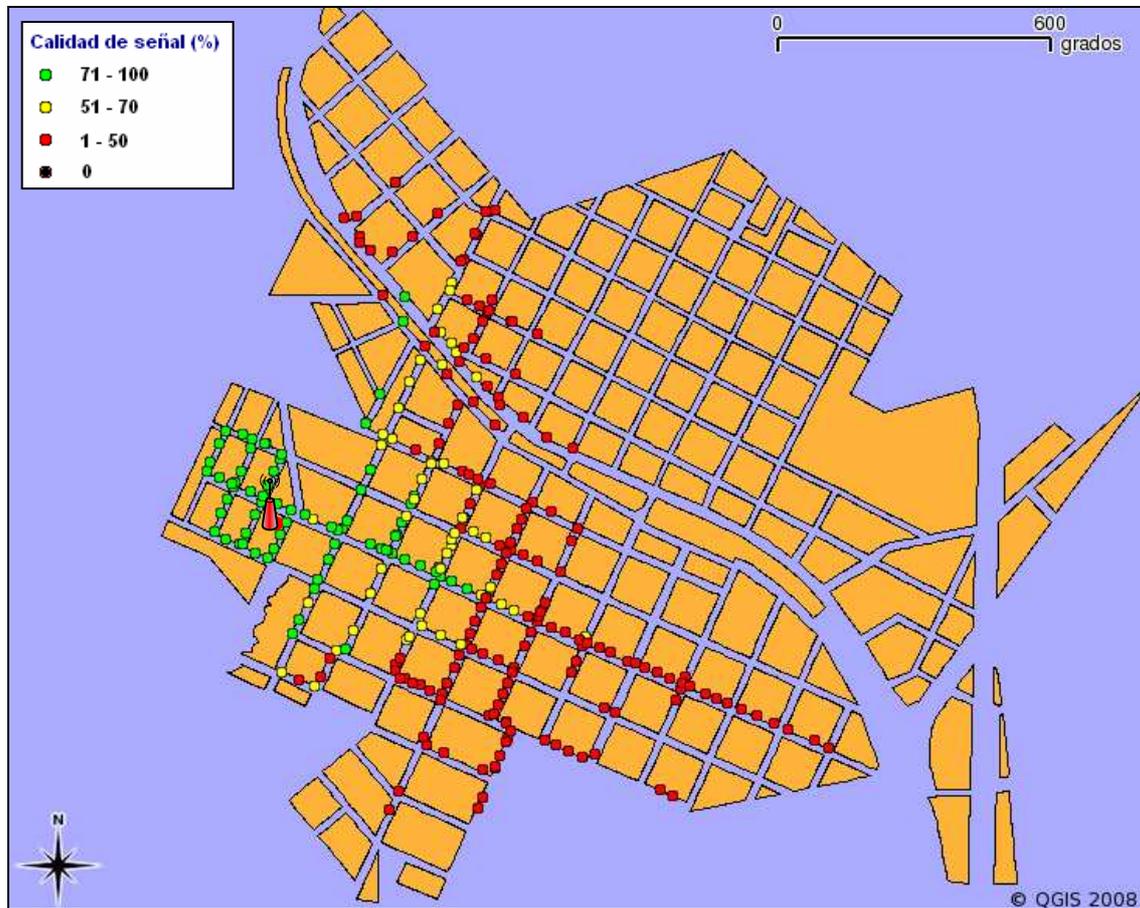


Fig 7.14: Mapa de cobertura AP 4 - MEVIR

Redes interferentes

Al exportar los resultados de cobertura, se genera también un archivo con la información de las redes interferentes a los AP de Ceibal.

De esta manera los técnicos poseen información valiosa para la planificación de los canales para nuevos AP o la optimización de la red ya instalada.

En la figura 7.15 se puede ver, para cada punto donde se realizó un escaneo, los valores de cobertura para los AP interferentes.



Fig 7.15: Mapa de cobertura de redes interferentes

Luego de exportados los datos al QGIS, podemos generar layers para realizar el análisis por canal o potencia.

De manera similar, cuando se exportan los resultados de cobertura por AP desde Yacaré, se generan simultáneamente archivos con datos por AP de redes interferentes.

Mapa de cobertura por potencia

Los mismos análisis que hicimos anteriormente de mapas de cobertura en función de la calidad de la señal, pueden ser realizados en función de la potencia.

Podemos ver en la figura 7.16 un mapa de cobertura de Sarandí Grande exportado del QGIS, donde se seleccionó un layer de potencia, donde los niveles de señal son los que se muestran en la referencia.

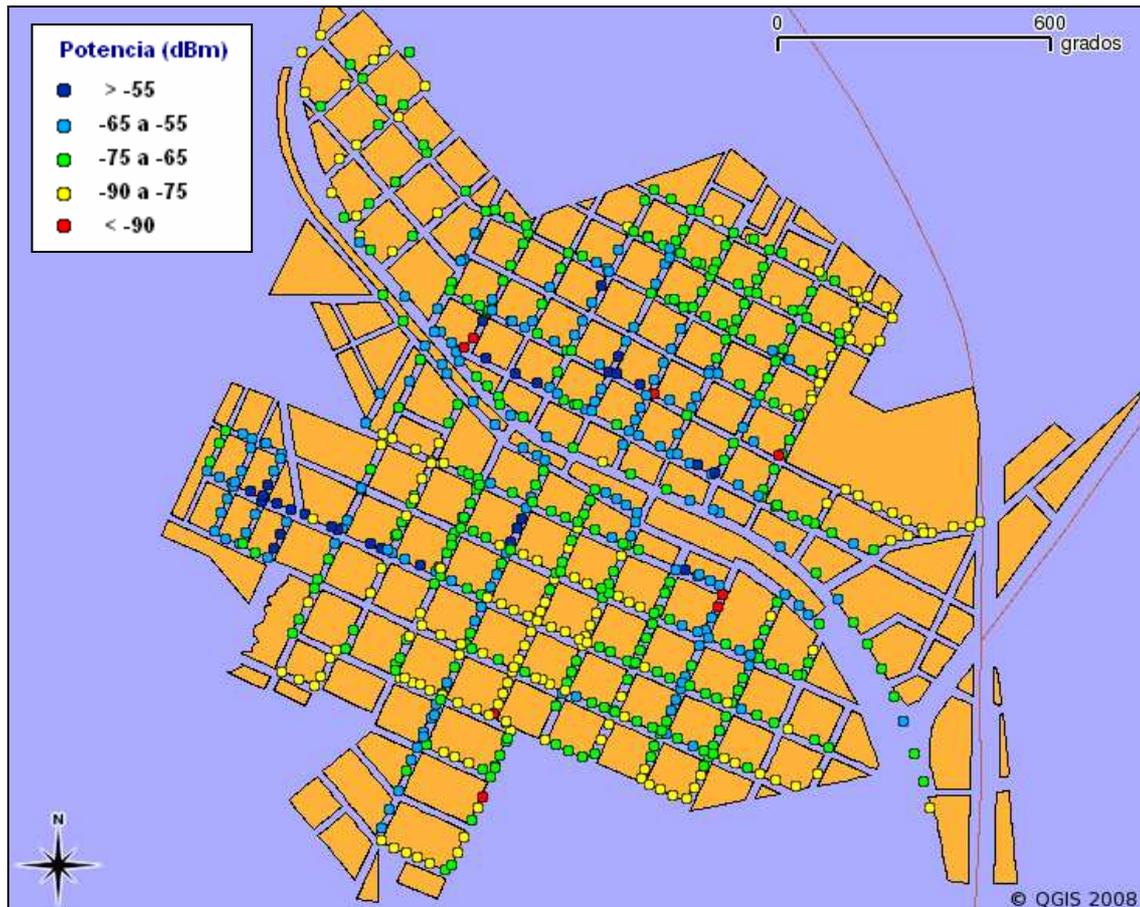


Fig 7.16: Mapa de cobertura por potencia de Sarandí Grande

Podemos apreciar claramente la similitud entre los layers exportados por calidad y el anterior mapa con el layer por potencia. Este tipo de mapas nos permite independizarnos del driver ya que factor de calidad de señal WiFi es proporcionado por el fabricante de cada tarjeta de red.

Búsqueda de portales Mesh

La búsqueda de MPP se realizó en la zona donde hay mayor concentración de viviendas, en la zona oeste de la ciudad.

Elegimos una zona reducida porque el tiempo de adquisición de un escaneo completo (WiFi y Mesh) es 4 veces mayor al de un escaneo WiFi, como se detallará en 7- 3.

En la figura 7.17 se muestra por punto la cantidad de XO que están como MPP.

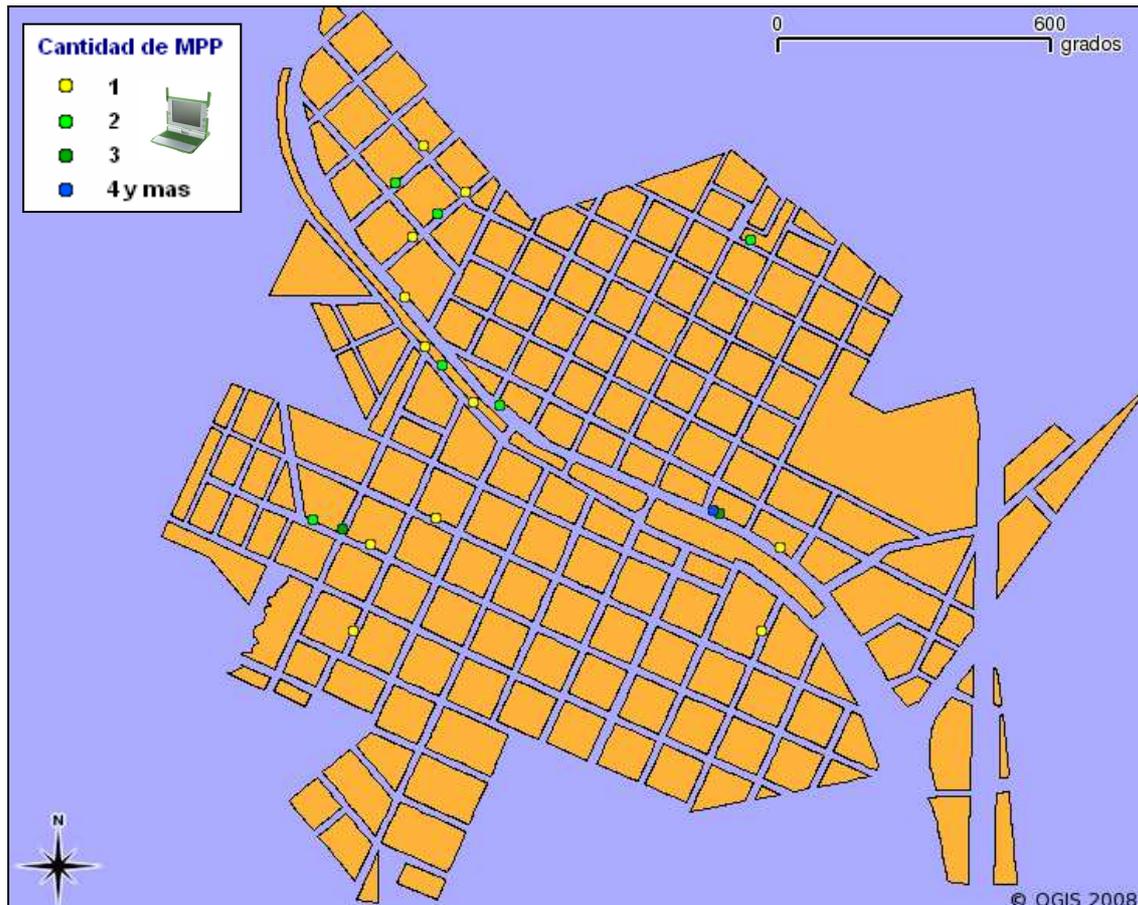


Fig 7.17: Búsqueda de MPP

Simultáneamente con la etapa de adquisiciones utilizamos también una laptop XO para corroborar los MPP encontrados. Los puntos aislados que aparecen en el mapa (fuera de la zona oeste) fueron tomados al ver en la calle niños con XO.

Cabe destacar que en los relevamientos de WiFi observamos redes Mesh aisladas (Ad-hoc), que aparecen en los escaneos con SSID "olpc-mesh". Estas redes solo tienen MPs.

El relevamiento de búsqueda de MPP fue realizado un sábado, teniendo en cuenta que los resultados son de carácter estadístico y representa una “foto” de la actividad Mesh del momento, la mejor opción para realizarlo y encontrar la mayor cantidad de MPP hubiese sido un día de semana en horario de clase o posterior.

Mostramos algunas fotos donde había niños con XO en Sarandí Grande.

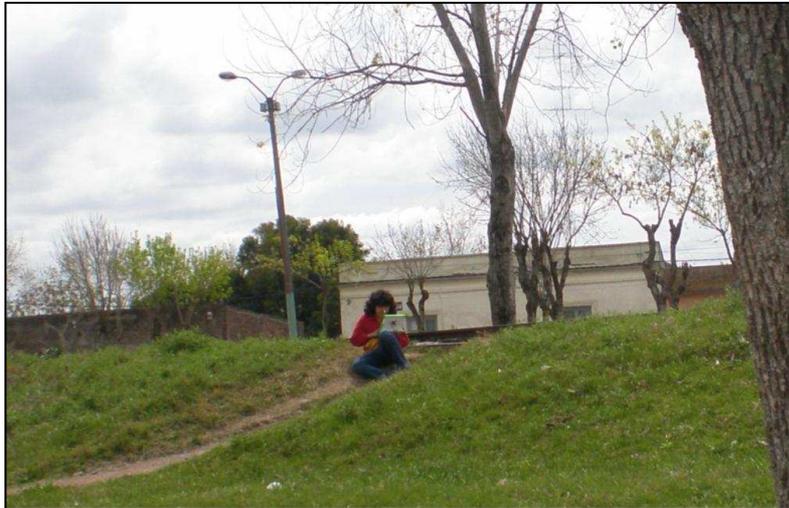


Fig 7.18: Sobre Av. Artigas

En la siguiente foto se puede ver el AP del MEVIR a tope en el mástil con su antena omnidireccional.



Fig: 7.19: Centro comunal del MEVIR

Ancho de Banda

Se realizaron tests de ancho de banda para la zona norte de Sarandí Grande.

La conexión ssh se realiza contra el servidor local de Ceibal, que requiere conocer dirección IP y contraseña para autenticación. A nivel de acceso de la red WiFi, la conexión se establece contra el AP Ceibal con mayor calidad de señal, para esta zona son el AP 1 Tanque de OSE y el AP 2 Escuela N° 38, como pudimos observar en los mapas de cobertura WiFi anteriores.

En la figura 7.20 se muestran los resultados obtenidos en la pruebas de bajada, pudiéndose desplegar también subida y eficiencia del uso de red.

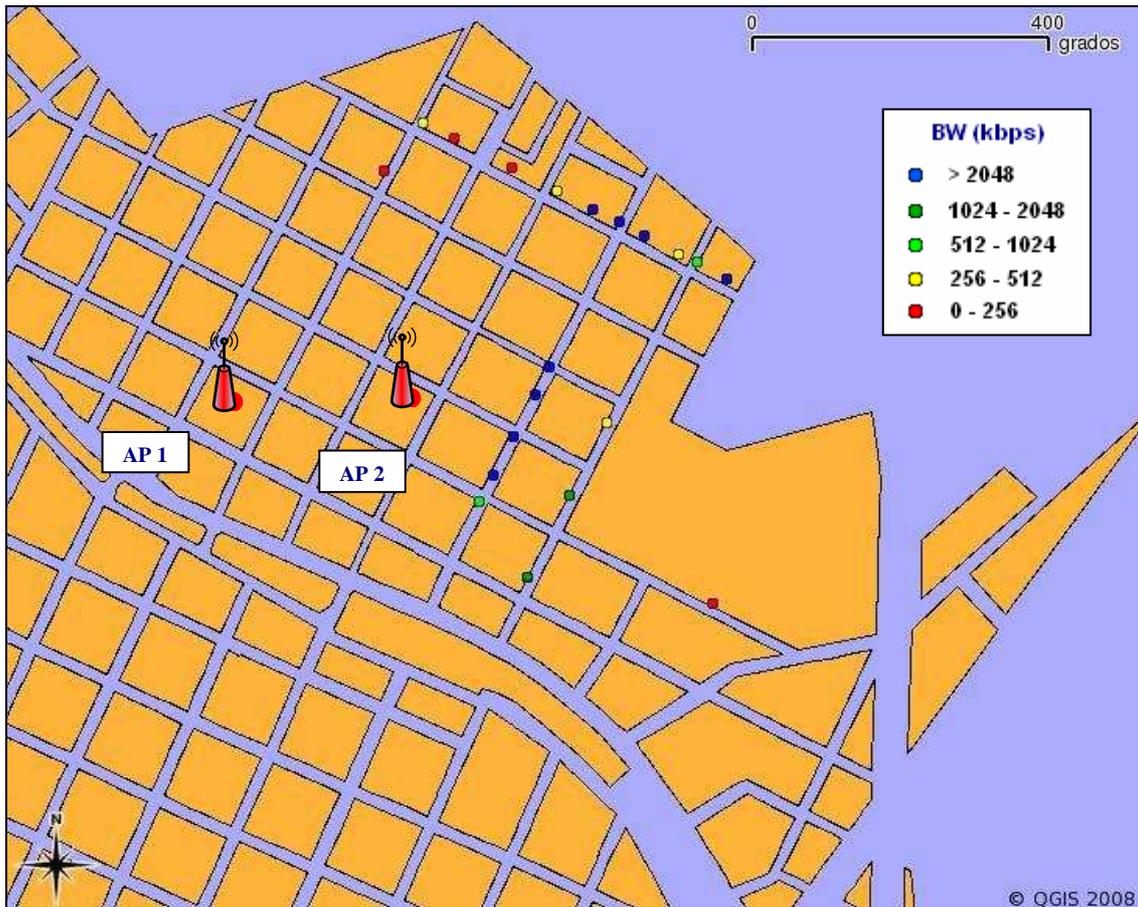


Fig 7.20: Resultados test de ancho de banda

Optamos por traficar archivos de tamaño 256kB. Para los puntos que obtuvimos buen throughput realizamos nuevamente el test traficando archivos de 512kB para corroborar los resultados. La eficiencia del uso de la red fue de 80 a 90%, si consideramos retransmisiones y payload respecto a la carga útil en la transferencia de los datos.

Para puntos lejanos (más de 400 metros del AP), algunos resultados del test fueron distintos para pruebas consecutivas, a pesar de tener un buen nivel de señal WiFi el ancho de banda varía debido a distintos factores.

El resultado de la prueba de ancho de banda permite detectar problemas si encontramos una zona con buen nivel de cobertura y una baja tasa de transferencia.

7- 3 Desempeño de la aplicación

Creímos conveniente desarrollar una serie de pruebas para determinar el consumo de recursos del sistema por parte de Yacaré. Para esto probamos las funcionalidades que realizan la mayor cantidad de procesos en paralelo que son los escaneos y las adquisiciones. La primera es realizada en forma continua mientras que la segunda es realizada puntualmente pero corre muchas tareas en background sobre el manejo de los datos.

Las pruebas fueron realizadas con una equipo que contaba con procesador AMD Turion 64x2, CPU Clock 1.8GHz y 2Gb de memoria RAM.

Las primeras pruebas consistieron en realizar escaneos durante un tiempo largo y relevar el estado del procesador. Con este test intentamos manejar la utilización de recursos de los procesadores por parte de Yacaré.

En la figura 7.21 se muestra el monitor del sistema de Fedora ante la realización de los escaneos, como podemos ver se realiza un consumo de ambos procesadores bastante exigente, en donde se observan los picos de consumo cuando la interfaz de red realiza el escaneo completo.

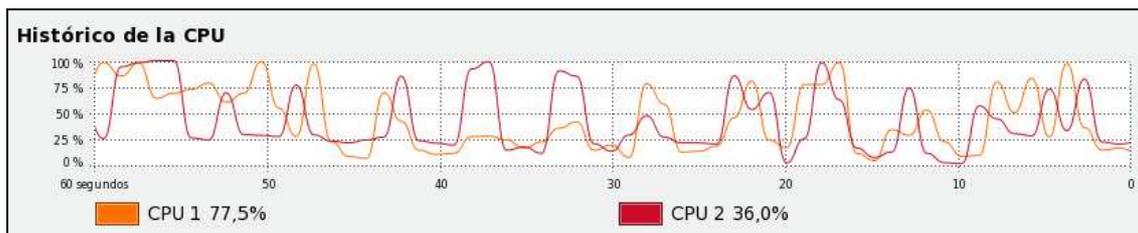


Fig 7.21: Monitor del sistema con Yacaré realizando escaneos

La aplicación luego de un tiempo prueba de 20 minutos se siguió comportando de manera óptima, los tiempos de respuesta de las funcionalidades tampoco se vieron afectados por la duración de la prueba.

La segunda prueba consistió en realizar un número de adquisiciones elevado en una zona en donde los escaneos determinaron la presencia promedio de 8 AP por escaneos. Con esta prueba se intento realizar un cálculo aproximado del tamaño de los archivos con la información de las adquisiciones y a su vez relevar el consumo del procesador al realizar la toma de muestras.

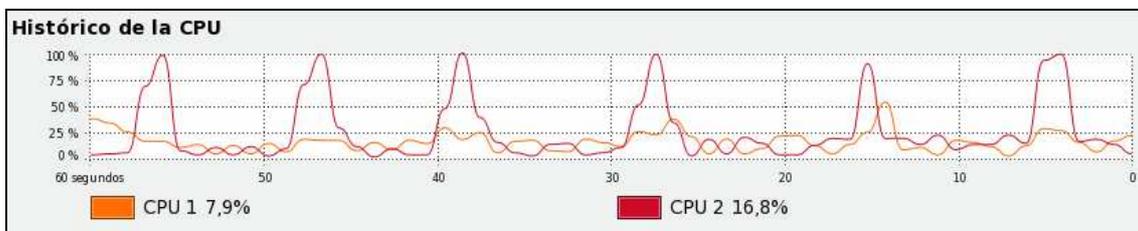


Fig 7.22: Monitor del sistema con Yacaré realizando adquisiciones

Como podemos apreciar en la figura 7.22, al realizar una adquisición solo se ve exigido un procesador. Pudimos estimar que el tamaño de los archivos tiene una tasa de 500KB/(100 Adquisiciones). Los archivos de relevamientos no son de tamaño excesivo por lo que no consideramos críticas las operaciones con los archivos

temporales. Cuando éstos son exportados en tablas *.txt* baja considerablemente el tamaño de los mismos ya que no superan los 50Kb.

Finalmente realizamos también pruebas para estimar los tiempos de respuesta del sistema frente a la toma de adquisiciones. Tomamos como referencia un relevamiento típico, con 3 escaneos WiFi. Los resultados fueron relevantes ya que el tiempo que toma una adquisición completa con escaneos de Access Points WiFi y búsqueda de portales Mesh es de 20 segundos, mientras que una adquisición utilizando solo las funcionalidades WiFi es de 5 segundos. Es recomendable que el usuario tenga muy en cuenta estos tiempos de respuesta a la hora de realizar un relevamiento.

Los escaneos se hicieron a un promedio de 20km/h para WiFi y para el completo nos deteníamos en una posición hasta completar el proceso.

Luego de realizadas estas pruebas podemos asegurar un buen desempeño de la aplicación observándose una performance óptima cuando ésta es ejecutada con la menor cantidad de aplicaciones en paralelo.

8. Conclusiones

Como conclusión general del proyecto tenemos que decir que estamos sumamente conformes con el resultado final. Algo fundamental para nosotros es que la aplicación sea útil para Ceibal, y creemos que logramos adaptarnos a sus necesidades. Al estar uno de nosotros trabajando en Ceibal, tuvimos siempre un buen feedback de parte del departamento técnico, y pudimos llegar a tiempo a nuevos requerimientos. Un ejemplo claro, es el adaptar las pruebas de ancho de banda para tener buenos relevamientos de cobertura de los equipos MIMO. Nuestro error, quizás fue olvidarnos por momentos de que Yacaré es un proyecto de fin de carrera, y además de la aprobación del cliente, debíamos cumplir cronogramas y objetivos trazados en el plan de proyecto.

Relativo al cumplimiento del cronograma, podemos dividir el proceso en dos etapas claras. La primera, donde cumplimos en tiempo y forma el cronograma, que abarca hasta el segundo entregable, presentado el 23 de Junio. El haber definido módulos claros en cuanto a la programación, donde intentamos tener la menor dependencia uno de otros, permitió desarrollar los módulos Mesh, WiFi y GPS, prácticamente idénticos a los entregados finalmente en el proyecto. La segunda etapa, desde la fecha mencionada hasta la entrega final, en donde el esfuerzo realizado fue mucho mayor al planificado. La integración de todas las partes a la interfaz gráfica demandó una carga horaria importante.

La complejidad que encontramos en la etapa final no fue la interacción con los dispositivos, sino el control sobre el estado de los procesos que son desencadenados. Fue entonces donde comenzamos a notar que los tiempos trazados para esta etapa de desarrollo de Yacaré fueron un poco ingenuos. Como las herramientas con las que contábamos hasta el momento no eran suficientes, tuvimos que dedicar mucho más tiempo del esperado al estudio de Python.

Fue fundamental para nosotros lograr una interfaz amigable e intuitiva, que mantenga informado al usuario de que es lo que está sucediendo, que sea una buena herramienta para relevar, y que no necesite ser complementada con otras aplicaciones.

El contar con otros programas de código abierto, sobre todo con un programa de monitoreo desarrollado por Ceibal en Python, fue una de las herramientas más útiles a la hora del desarrollo de la aplicación. Diversos manuales de Python, foros y la documentación de los módulos que utilizamos, nos demostraron que las tareas que nos trazamos en un principio implicaban una dedicación más grande de la planificada. Ninguno de los tres contaba con grandes conocimientos en programación por lo que nos enfrentamos con varios errores debido a la falta de experiencia. Más allá de esto, creemos que gracias a la elección del lenguaje pudimos solucionarlos y progresar.

Más allá de las dificultades con las que nos encontramos, creemos que la elección de Python como lenguaje de programación fue acertada. La estructura simple de las sentencias de programación, la gran disponibilidad de manuales y el gran desarrollo que tiene este lenguaje lograron que el proceso de aprendizaje se acelerara.

La etapa de pruebas hizo darnos cuenta el potencial que tiene la herramienta diseñada. Estamos convencidos que el análisis que es posible realizar a partir de la gran cantidad de información adquirida por Yacaré será de suma utilidad para el Plan Ceibal.

El proyecto como experiencia en sí mismo, nos hizo crecer en aspectos técnicos, de investigación, desarrollo y áreas de la ingeniería en las que no habíamos tenido la posibilidad de profundizar como estudiantes. Creemos que luego de haber estudiado estándares y protocolos de comunicación en detalle, comprendemos un poco mejor el desafío que implica el desarrollo de los mismos. Por más de que se logró de manera “sencilla” la comunicación con los dispositivos y la adquisición de los datos necesarios, se requirió para su procesamiento una mayor comprensión de lo que realmente se estaba realizando y los resultados que queríamos obtener.

En este proceso encontramos un crecimiento personal de cada uno de nosotros. El documentar y compartir información dentro del grupo nos sirvió para darnos cuenta realmente de los conocimientos que habíamos adquirido. No nos consideramos expertos en ninguna de las tecnologías que utiliza Yacaré, pero si creemos que contamos con herramientas valiosas para continuar nuestro desarrollo en estas áreas o profundizar en el estudio de otras.

9. Anexos

9- 1 802.11: Capa Física - Modulación y Codificación

El Standard original 802.11 define 3 diferentes tipos de capa física, Frequency Hopping Spread Spectrum (FHSS), Direct Sequence Spread Spectrum (DSSS) y Infrared (IR). Estos 3 tipos están definidos para la banda de frecuencias de 2.4Ghz (ISM).

El Standard 802.11b/g está definido para tecnologías de Spread Spectrum e introduce la división de la banda ISM de 2,4GHz en 14 canales, definiendo también la técnica HR-DSSS en 802.11b para alcanzar velocidades de hasta 11Mbps y OFDM en 802.11g alcanzando así velocidades de transmisión de 54Mbps.

Vamos a enfocaremos en las tecnologías de Spread Spectrum y recorreremos cada una de ellas haciendo un repaso de los conceptos mas importantes que se definen en la norma.

9- 1.1 FHSS

Fue definida en el Standard original 802.11, provee 1 y 2 Mbps en la banda ISM de 2.4GHz. En Norte America, la IEEE especifica en 802.11 que FHSS usa los 79MHz que van de 2.402GHz a 2.480GHz. FHSS transmite los datos con una portadora que utiliza una pequeña banda de frecuencia, luego de un tiempo corto previamente definido salta a otra portadora para continuar la comunicación y así sucesivamente como muestra la figura 9.1.

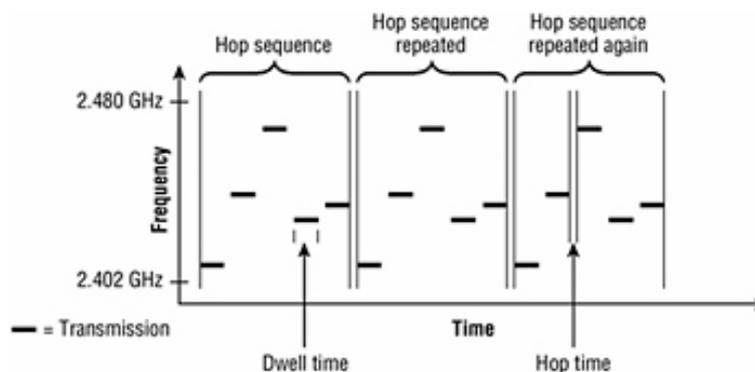


Fig 9.1: FHSS

Los atributos que define FHSS son: el patrón de secuencias, el tiempo predefinido en cada portadora (dwell time), tiempo de salto (Hop Time) y la modulación utilizada.

Patrón de secuencias

FHSS utiliza un patrón de secuencias predefinido, una serie de portadoras de pequeño ancho de banda llamados saltos. Los standards IEEE 802.11 definen que cada salto es de tamaño 1MHz. En Norteamérica y en la mayoría de los países de Europa el patrón de secuencias utiliza un set de 75 saltos. Para que la transmisión se realice de manera satisfactoria, los transmisores y receptores FHSS deben estar sincronizados en el mismo salto en el mismo instante de tiempo. En el Standard 802.11 se define que el patrón de secuencias puede ser configurado por el AP y luego esta información es entregada a las estaciones mediante los beacon management frames.

Dwell time

Es el tiempo en que los sistemas FHSS transmiten en una frecuencia específica antes de saltar a la próxima portadora en el patrón de secuencia. Típicamente esta cantidad de tiempo está entre los 100ms y 200ms, pero puede llegar a un máximo de 400ms. Para un ancho de banda de 79MHz, el máximo número de portadoras es 79. Entonces con un patrón de secuencia de 75 portadoras, lleva como máximo 30 segundos recorrer la secuencia completa.

Hop Time

El tiempo de salto no es una cantidad previamente definida pero es la medida de tiempo que le lleva al trasmisor cambiarse de una frecuencia a la otra. Típicamente el tiempo de salto es un numero chico en comparación con el dwell time, se encuentra cerca de los 300 microsegundos. Cuanto mayor es el dwell time, menos frecuentemente el transmisor tiene que cambiar de frecuencia, por lo que menos tiempo desperdicia en hop time, esto se traduce en mayor throughput. Por lo tanto si el dwell time es corto, el transmisor debe malgastar una cantidad de tiempo mayor en hop time haciendo decrecer el throughput.

Modulación

FHSS utiliza Gaussian Frequency Shift Keying (GFSK) para codificar los datos. Puede utilizar 2-GFSK, que usa 2 frecuencias para modular un bit o 4-GFSK, que utiliza 4 frecuencias para representar un patrón de 2 bits.

9- 1.2 DSSS

Direct Sequence Spread Spectrum fue originalmente especificada en la primera versión de 802.11. El Standard provee 1Mbps y 2Mbps a las comunicaciones de RF en la banda ISM de 2.4GHz. DSSS también fue especificada en la 802.11b brindando 5.5Mbps y 11Mbps utilizando también la banda ISM de 2,4GHz, conocida como High Rate DSSS.

Los equipos 802.11b son compatibles hacia atrás con los dispositivos DSSS de 802.11 originales. De cualquier manera los equipos DSSS no son capaces de transmitir FHSS.

A diferencia de FHSS en donde el transmisor saltaba entre frecuencias, DSSS se establece en un solo canal. Los datos entonces son transmitidos a través de todo el rango de frecuencias que define el canal. El proceso de enviar datos en el canal es conocido como DSSS Data Encoding.

9- 1.3 Canales en 802.11b/g

Para entender como es utilizado DSSS 802.11b/g es importante saber que el Standard IEEE 802.11 divide la banda ISM de 2.4GHz en 14 canales que se solapan entre si, dejando libres de solapamiento los canales que distan 5, el ejemplo mas utilizado es el de los canales 1, 6 y 11.

DSSS Data Encoding

Como el medio RF utilizado por 802.11 está expuesto potencialmente a una enorme interferencia, las señales tienen que ser diseñadas a favor de minimizar la corrupción de los datos.

Para minimizar los errores se agrega información adicional mediante la inclusión de bits. El sistema convierte x bits de datos en una serie de bits que llamaremos chips, este proceso de convertir bits de datos en chips es conocido como expansión o spreading.

Hay muchas técnicas para crear chips, DSSS originalmente utilizaba códigos de Baker, que consistía en expandir 1 bit en un chip de 11 bits mediante el XOR del bit de información y un código de 11 bits como podemos ver a continuación en el ejemplo:

- Binary data 1 = 1 0 1 1 0 1 1 1 0 0 0
- Binary data 0 = 0 1 0 0 1 0 0 0 1 1 1

Esta secuencia de chips es enviada a través del espacio de frecuencias. Como 1 bit de datos necesita 2MHz de espacio, los 11 Chips utilizan el total de 22MHz del canal. Luego de transmitidos el receptor convierte la secuencia del chip en 1 bit. Cuando los chips no son recibidos de manera correcta, el receptor aun es capaz de interpretar los datos mediante el conocimiento del código de Baker usado. En el caso de este código de 11 bits, el receptor puede reconocer la información mandada hasta cuando 9 de los 11 bits están corruptos.

El código de Baker usa chips de 11 bits, a pesar de esto, el largo del código es irrelevante. Para ayudar a brindar mayores velocidades, en HR-DSSS es utilizado Complementary Code Keying (CCK), que utiliza chips de 8 bits diferentes para frecuencias distintas y permite codificar 4 bits de datos con 8 chips (5.5Mbps) y 8 bits de datos con 8 chips también (11Mbps).

Modulación

Luego de que los datos fueron codificados, el transmisor necesita modular la señal para crear una señal portadora que contenga los chips. Differential Binary Phase Shift Keying (DBPSK) utiliza dos fases para representar 1 bit, para throughputs mayores, se utiliza Differential Quaternary Phase Shift Keying (DQPSK). En la figura 9.2 podemos ver como el data rate se modifica utilizando diferentes combinaciones de codificación y modulación.

Standard	Velocidad (Mbps)	Codificación	Largo del Chip	Bits Codificados	Modulación
802.11	1	Barker Coding	11	1	DBPSK
802.11	2	Barker Coding	11	1	DQPSK
802.11b	5.5	CCK Coding	8	4	DQPSK
802.11b	11	CCK Coding	8	8	DQPSK

Fig 9.2: Combinaciones de modulación y codificación

9- 1.4 OFDM

Orthogonal Frequency Division Multiplexing (OFDM) es una de las tecnologías más populares en comunicaciones, es utilizado tanto en medios cableados como wireless. Como parte de una tecnología 802.11, OFDM es especificado en 802.11a y 802.11g y permite transmitir a velocidades de 54Mbps. Aunque no es una tecnología de spread spectrum, comparte propiedades similares como bajas potencias de transmisión y la utilización de mayor ancho de banda del que es requerido para transmitir datos. OFDM transmite a través de 52 canales separados, muy cercanos y precisamente espaciados como podemos ver en la figura 9.3.

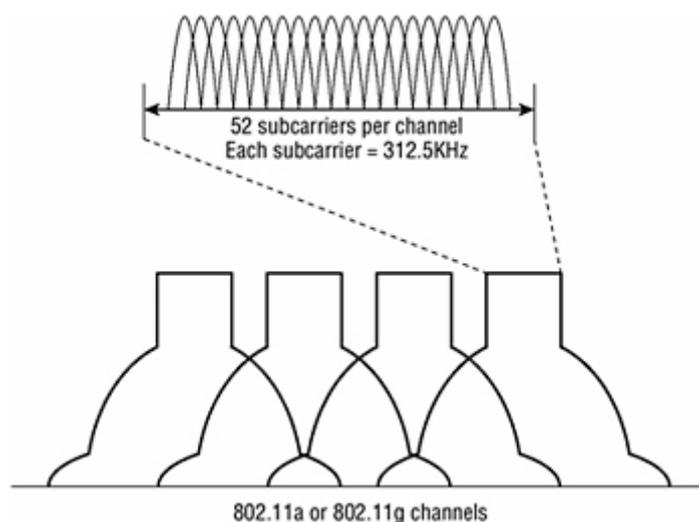


Fig 9.3: Portadoras OFDM

El ancho en frecuencia de cada subportadora es de 312.5KHz. Éstas son transmitidas a su vez a una velocidad más lenta, pero al tratarse de un número tan grande de subportadoras, las velocidades de transmisión son mayores que en tecnologías que utilizan los mismos canales. OFDM también es más resistente a los efectos negativos de multicamino e interferencia intersimbólica observados en las técnicas DSSS y FHSS.

Las 52 subportadoras se numeran del -26 al +26, 48 de ellas son utilizadas para transmitir datos y las 4 restantes son utilizadas como piloto. Éstas son usadas como referencias de fase y amplitud por el demodulador, permitiendo así al receptor compensar la distorsión de la señal OFDM.

Códigos de Convolución

Para hacer OFDM mas resistente a las interferencias de banda angosta, se utiliza una técnica de corrección de errores llamada códigos de convolución. Esta técnica no es precisamente parte de OFDM pero si es parte de los estándar 802.11a y 802.11g. Esta técnica es de tipo forward error correction (FEC) y permite al receptor detectar y reparar bits corruptos.

La figura 9.4 muestra una comparación entre las diferentes tecnologías usadas para crear diferentes velocidades de transmisión en 802.11g.

Velocidad (Mbps)	Método de Modulación	Bits Transmitidos	Transmitidos	Bits Codificados
6	DBPSK	1	1	2
9	DBPSK	1	3	4
12	DQPSK	2	1	2
18	DQPSK	2	3	4
24	16-QAM	4	1	2
36	16-QAM	4	3	4
48	64-QAM	6	2	3
54	64-QAM	6	3	4

Fig 9.4: Velocidades en 802.11g

Throughput vs. Ancho de banda

Las comunicaciones wireless son realizadas en una banda restringida de frecuencias. Uno de los aspectos más sorprendentes en las tecnologías de red wireless, es el verdadero throughput que logran alcanzar las redes.

El ancho de banda juega un papel principal a la hora del cálculo de la velocidad de la señal y el throughput de los datos transferidos, pero hay otros factores influyentes. Al ancho de banda se le suma, la codificación y la modulación, como vimos anteriormente, en 802.11g, OFDM puede transmitir a 6, 9, 12, 18, 24, 36, 48 o 54 Mbps, mientras el ancho de banda de la comunicación es exactamente el mismo para todas las velocidades.

El control de acceso al medio conocido como Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) ayuda a asegurar que solo una tarjeta de radio pueda transmitir en el medio en un tiempo dado, pero el overhead generado por CSMA/CA es el 50% del throughput aproximadamente, y a esto se le suma por naturaleza que el medio es half duplex.

En adición a estos problemas, no hay que olvidar que la banda de frecuencias utilizada es un medio compartido y que 802.11 no trabaja bien ante la presencia de interferencia en 2.4GHz.

Hay muchas otros factores como la encriptación de los datos, el agregado de seguridad y la fragmentación de tramas que incrementan el overhead en la comunicación.

Por lo tanto, hay muchas variables en todas las capas del modelo OSI que afectan el throughput en las comunicaciones 802.11, por eso es importante tener en cuenta cada una de las diferentes causas y sus efectos para así saber que está al alcance de nuestra mano a la hora de minimizarlos. ^[19]

9- 2 802.11: Acceso al medio

DCF: Distributed coordination function

DCF es esencialmente el protocolo CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). Como lo indica su sigla en inglés, consiste en sensar la portadora y un mecanismo para evitar colisiones. Naturalmente, si el único criterio para transmitir fuese detectar la portadora “libre” (nadie está transmitiendo) todas las estaciones que desean utilizar el medio en ese instante lo harían inmediatamente y existirían las mencionadas colisiones.

Para implementar la detección de portadora (CS), el estándar define dos formas: física y virtual. La física, es un servicio de la capa física en el que no entraremos en detalle, y tenemos en cuenta únicamente que nos dice únicamente si el medio está siendo utilizado o si está disponible. La virtual (es opcional) es la implementación de los mensajes RTS y CTS que son el pedido para utilizar el medio, y el permiso de utilizarlo. La detección de portadora virtual es recomendada en situaciones donde hay muchas STA a transmitir, y posiblemente muchas de ellas estén ocultas.

En la figura vemos el funcionamiento con DCF sin RTS/CTS.

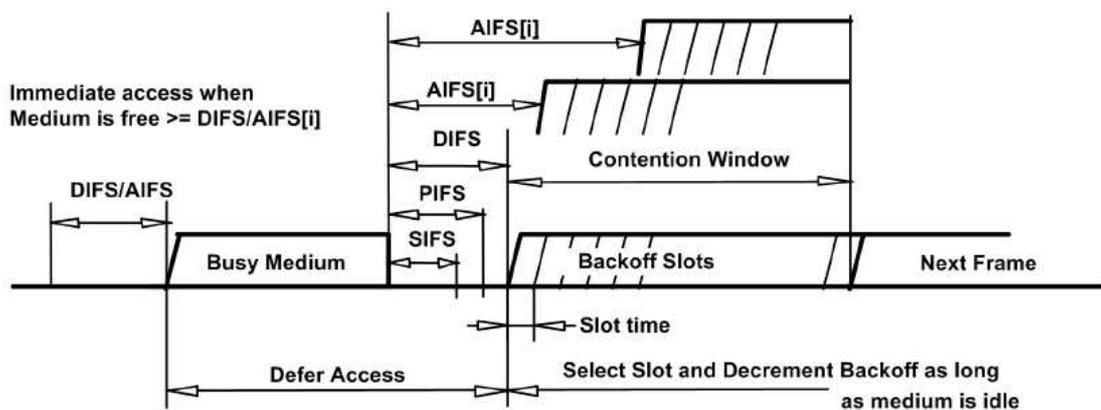


Fig 9.5: Diagrama de tiempos DCF

Se define el DIFS (DCF interframe space) como mínimo tiempo entre tramas. Cada estación que quiere transmitir realiza el siguiente procedimiento:

- CS físico para detectar si el medio está siendo utilizado
- Si está “libre” espera un tiempo DIFS
- Si luego del DIFS sigue libre, genera una ventana de contención: un contador en un número aleatorio por el tamaño de un Time Slot. Cada un Time Slot el contador se decrementa en uno si el CS devuelve que el medio está libre
- Cuando se alcanza el 0 la STA transmite la trama.

Si se implementa además la detección de portadora virtual con RTS/CTS, el diagrama de tiempos es el siguiente, donde SIFS (Short Interframe Space) es un tiempo mas corto que el DIFS que se espera para transmitir los CTS, RTS y el ACK de la trama.

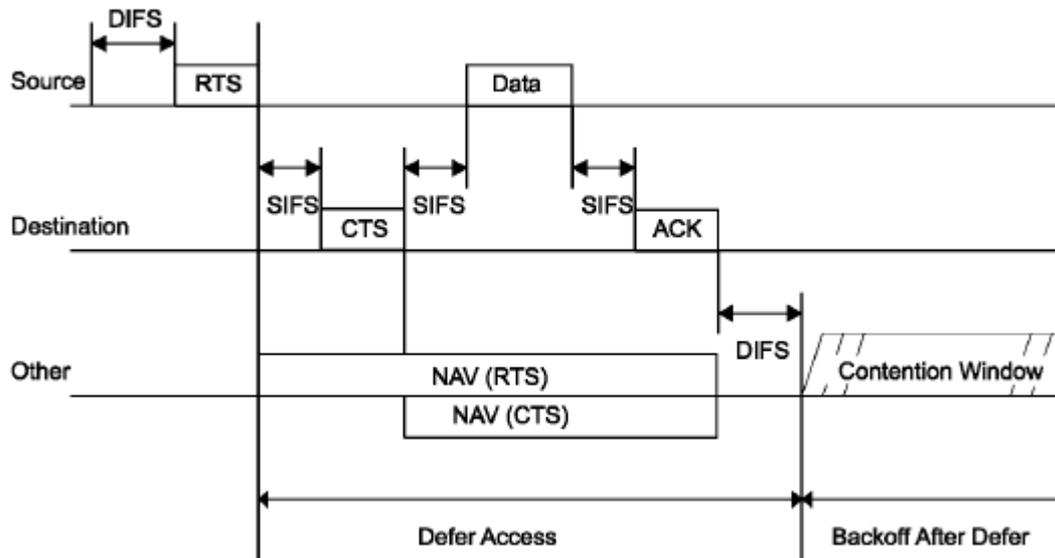


Fig 9.6: Diagrama de tiempos DCF con detección de portadora virtual

En este funcionamiento, las pérdidas en las colisiones son mínimas, ya que los mensajes que pueden colisionar son los RTS y no tramas con datos (teóricamente). En la práctica, soluciona problemas por ambientes congestionados. La ventana de contención es en general de 16 Time Slots, y la probabilidad de colisionar en una red de muchas STA crece mucho. Si tenemos en cuenta que, por ejemplo, una clase desea buscar contenidos en la Web, la simultaneidad en el acceso al medio es seguramente bastante más alta que en el uso habitual de las conexiones WiFi a las que estamos acostumbrados. Cuando las probabilidades son más bajas, como en una red doméstica, con 4 o 5 laptops, es más eficiente realizar únicamente la detección física de portadora. ^[2]

9- 3 Escaneos: Mensajes intercambiados

Realizaremos una breve descripción de los mensajes Probe Request, Probe Response y Beacon. Son parte fundamental de los proceso de escaneo activo y pasivo en 802.11.

Comenzando con los mensajes para el escaneo activo, la STA que realiza el escaneo envía un Probe Request, lo hace según el formato de la figura 9.7, donde en el campo *Request information* lista la información solicitada. Esta última es respondida en el campo *Requested information elements* del Probe Response.

Order	Information	Notes
1	SSID	
2	Supported rates	
3	Request information	May be included if dot11MultiDomainCapabilityEnabled is true.
4	Extended Supported Rates	The Extended Supported Rates element is present whenever there are more than eight supported rates, and it is optional otherwise.
Last	Vendor Specific	One or more vendor-specific information elements may appear in this frame. This information element follows all other information elements.

Fig 9.7: Campos de Probe Request

El mensaje Probe Response, es más extenso, ya que lista información sobre la STA, tiempos de beaconing, parámetros para FHSS y DSSS, bit rates soportados, reporte TPC, etc. La información es muy similar a la del Beacon, comparando las figuras 9.8 y 9.9 sólo que Probe Response incluye el campo *Requested information elements*.

Order	Information	Notes
1	Timestamp	
2	Beacon interval	
3	Capability	
4	SSID	
5	Supported rates	
6	FH Parameter Set	The FH Parameter Set information element is present within Probe Response frames generated by STAs using FH PHYs.
7	DS Parameter Set	The DS Parameter Set information element is present within Probe Response frames generated by STAs using Clause 15, Clause 18, and Clause 19 PHYs.
8	CF Parameter Set	The CF Parameter Set information element is present only within Probe Response frames generated by APs supporting a PCF.
9	IBSS Parameter Set	The IBSS Parameter Set information element is present only within Probe Response frames generated by STAs in an IBSS.
10	Country	Included if dot11MultiDomainCapabilityEnabled or dot11SpectrumManagementRequired is true.
11	FH Parameters	FH Parameters, as specified in 7.3.2.10, may be included if dot11MultiDomainCapabilityEnabled is true.
12	FH Pattern Table	FH Pattern Table information, as specified in 7.3.2.11, may be included if dot11MultiDomainCapabilityEnabled is true.
13	Power Constraint	Shall be included if dot11SpectrumManagementRequired is true.
14	Channel Switch Announcement	May be included if dot11SpectrumManagementRequired is true.
15	Quiet	May be included if dot11SpectrumManagementRequired is true.
16	IBSS DFS	Shall be included if dot11SpectrumManagementRequired is true in an IBSS.
17	TPC Report	Shall be included if dot11SpectrumManagementRequired is true.
18	ERP Information	The ERP Information element is present within Probe Response frames generated by STAs using ERPs and is optionally present in other cases.
19	Extended Supported Rates	The Extended Supported Rates element is present whenever there are more than eight supported rates, and it is optional otherwise.
20	RSN	The RSN information element is only present within Probe Response frames generated by STAs that have dot11RSNA-Enabled set to TRUE.
21	BSS Load	The BSS Load element is present when dot11QosOptionImplemented and dot11QBSSLoadImplemented are both true.
22	EDCA Parameter Set	The EDCA Parameter Set element is present when dot11QosOptionImplemented is true.
Last-1	Vendor Specific	One or more vendor-specific information elements may appear in this frame. This information element follows all other information elements, except the Requested Information elements.
Last-n	Requested information elements	Elements requested by the Request information element of the Probe Request frame.

Fig 9.8: Campos de Probe Response

Order	Information	Notes
1	Timestamp	
2	Beacon interval	
3	Capability	
4	Service Set Identifier (SSID)	
5	Supported rates	
6	Frequency-Hopping (FH) Parameter Set	The FH Parameter Set information element is present within Beacon frames generated by STAs using FH PHYs.
7	DS Parameter Set	The DS Parameter Set information element is present within Beacon frames generated by STAs using Clause 15, Clause 18, and Clause 19 PHYs.
8	CF Parameter Set	The CF Parameter Set information element is present only within Beacon frames generated by APs supporting a PCF.
9	IBSS Parameter Set	The IBSS Parameter Set information element is present only within Beacon frames generated by STAs in an IBSS.
10	Traffic indication map (TIM)	The TIM information element is present only within Beacon frames generated by APs.
11	Country	The Country information element shall be present when dot11MultiDomainCapabilityEnabled is true or dot11SpectrumManagementRequired is true.
12	FH Parameters	FH Parameters as specified in 7.3.2.10 may be included if dot11MultiDomainCapabilityEnabled is true.
13	FH Pattern Table	FH Pattern Table information as specified in 7.3.2.11 may be included if dot11MultiDomainCapabilityEnabled is true.
14	Power Constraint	Power Constraint element shall be present if dot11SpectrumManagementRequired is true.
15	Channel Switch Announcement	Channel Switch Announcement element may be present if dot11SpectrumManagementRequired is true.
16	Quiet	Quiet element may be present if dot11SpectrumManagementRequired is true.
17	IBSS DFS	IBSS DFS element shall be present if dot11SpectrumManagementRequired is true in an IBSS.
18	TPC Report	TPC Report element shall be present if dot11SpectrumManagementRequired is true.
19	ERP Information	The ERP Information element is present within Beacon frames generated by STAs using extended rate PHYs (ERPs) defined in Clause 19 and is optionally present in other cases.
20	Extended Supported Rates	The Extended Supported Rates element is present whenever there are more than eight supported rates, and it is optional otherwise.
21	RSN	The RSN information element shall be present within Beacon frames generated by STAs that have dot11RSNAEnabled set to TRUE.

Order	Information	Notes
22	BSS Load	The BSS Load element is present when dot11QosOptionImplemented and dot11QBSSLoadImplemented are both true.
23	EDCA Parameter Set	The EDCA Parameter Set element is present when dot11QosOptionImplemented is true and the QoS Capability element is not present.
24	QoS Capability	The QoS Capability element is present when dot11QosOptionImplemented is true and EDCA Parameter Set element is not present.
Last	Vendor Specific	One or more vendor-specific information elements may appear in this frame. This information element follows all other information elements.

Fig 9.9: Campos de Beacon

9- 4 Protocolo DHCP – RFC 2131

Dynamic Host Configuration Protocol es un protocolo muy utilizado para la configuración IP de LANs y WLANs. Está construido sobre la base de un modelo cliente-servidor y cumple dos funciones:

1. Permitir la entrega de parámetros de configuración de un host desde un servidor DHCP hasta dicho equipo.
2. Proveer un mecanismo para la asignación de direcciones de red IP a los host que así lo requieran.

El servidor DHCP será el encargado de entregar estos dos servicios mencionados a lo host o clientes.

El flujo de mensajes intercambiados se puede ver en la figura 9.10

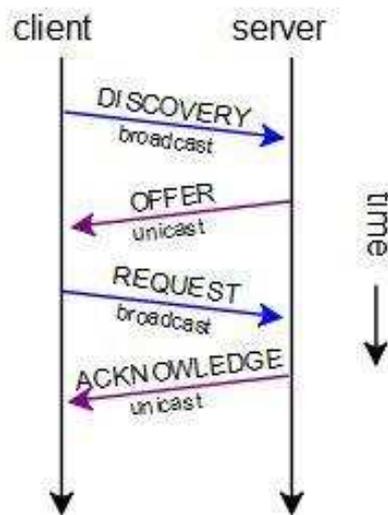


Fig 9.10: Mensajes Cliente-Servidor

- **DHCP Discovery**

El cliente envía un paquete DHCPDISCOVER. Las direcciones IP origen y destino de dicho paquete serán 0.0.0.0 y de broadcast 255.255.255.255 respectivamente. El servidor almacena los campos del paquete CHADDR (que contiene la dirección Ethernet origen y MAC) y el de identificación del cliente.

- **DHCP Offer**

El servidor determina la configuración basándose en la dirección del soporte físico de la computadora cliente especificada en el registro CHADDR. El servidor especifica la dirección IP en el registro YIADDR.

- **DHCP Request**

El cliente selecciona la configuración de los paquetes recibidos de *DHCP Offer*. Una vez más, el cliente solicita la dirección IP específica que indicó el servidor

- **DHCP Acknowledge**

El servidor DHCP recibe el paquete DHCPREQUEST del cliente, que inicia la fase final del proceso de configuración. Esta fase implica el reconocimiento DHCPACK con envío de un paquete al cliente. Este paquete incluye el arrendamiento de

duración y cualquier otra información de configuración que el cliente pueda tener solicitada. En este punto, queda completado el proceso de configuración TCP/IP. El sistema en su conjunto espera que el cliente configure su interfaz de red con las opciones suministradas.^[20]

Desde el punto de vista del cliente, DHCP es una extensión del mecanismo utilizado en el protocolo de asignación automática de dirección IP, BOOTP (Bootstrap Protocol). Lo anterior permite que los clientes BOOTP puedan interactuar con el servidor DHCP sin tener que alterar su software de inicialización. En este sentido, DHCP fue desarrollado diferenciándose de BOOTP en ciertos aspectos como por ej. el que algunos campos en el formato de mensaje han sido renombrados como "options" y se ha ampliado la función de otros campos.^[21]

A continuación se muestra en la figura 9.11 el formato de un mensaje DHCP con sus campos bien definidos (entre paréntesis se muestra el número de octetos correspondiente a cada campo):

op (1)	htype (1)	hlen (1)	hops (1)
xid (4)			
secs (2)		flags (2)	
ciaddr (4)			
yiaddr (4)			
siaddr (4)			
giaddr (4)			
chaddr (16)			
sname (64)			
file (128)			
options (variable)			

Fig 9.11: Formato de mensaje DHCP

La siguiente figura 9.12 es una breve descripción de los distintos campos presentados

Campo	Descripción
Op	Indica tipo de mensaje. Request o respuesta BOOTP.
htype	Tipo de dirección de hardware (ej. 1= 10MbEth).
hlen	Largo de dirección de hardware (ej. 6= 10MbEth).
hops	Cientes lo setean en 0.
xid	ID de transacción. N° aleatorio para identificarse entre cliente y servidor.
secs	Llenado por el cliente. Segundos transcurridos desde que el cliente ha solicitado dirección IP o renovación.
flags	Se utilizan para trabajar con clientes que no aceptan unicast.
ciaddr	Dirección IP del cliente. (Se llena para estados especiales).
yiaddr	Dirección IP del cliente.
siaddr	Dirección IP del próximo servidor en uso.
giaddr	Dirección IP de agente relacionador.
chaddr	Dirección de hardware de cliente.
sname	Nombre opcional del host servidor.
file	Nombre de archivo de BOOT.
options	Campo de parámetros opcional.

Fig 9.12: Descripción de los campos de mensaje DHCP

Asignación de dirección IP

La segunda función del protocolo DHCP es la asignación de direcciones IP. Para ello existirán tres mecanismos distintos. Se utilizará uno o varios de estos mecanismos según sea la política del administrador de red:

- **Asignación automática.** El servidor DHCP asigna una dirección IP permanente al cliente, se lo conoce como autoIP.
- **Asignación dinámica.** En este caso el servidor asigna la dirección IP por un periodo de tiempo determinado o hasta que el cliente deje de utilizarla.
- **Asignación manual:** La dirección es asignada por el administrador de red en forma manual. En este caso el servidor DHCP se utiliza simplemente para enviar dicha dirección.

De los tres mecanismos anteriores uno de los más útiles es la asignación dinámica debido a que es la única que permite la reutilización de una dirección IP que un cliente deja de utilizar. Así es posible distribuir un número acotado de direcciones IP entre clientes según las vayan requiriendo. El mecanismo de operación de la asignación dinámica es el siguiente:

- Un cliente hace Request de dirección IP por un periodo determinado de tiempo.
- El servidor DHCP garantiza que no se reasignará dicha dirección dentro del tiempo especificado y asume que entregará esa única dirección al cliente cada vez que éste solicite una.
- El cliente puede extender dicho tiempo con Request sucesivos.
- Si el cliente deja de ocupar la dirección puede publicar un mensaje dando aviso de esto al servidor.
- El cliente puede solicitar una dirección IP permanente, definiendo un tiempo de uso "infinito". El servidor por su parte establecerá tiempos extendidos pero finitos, asegurándose así que el uso de la dirección IP sea el adecuado.
- Cuando exista déficit de direcciones el servidor deberá reasignar de acuerdo a los parámetros de configuración del repositorio (uso de direcciones cuyo tiempo ha expirado).

9- 5 Sistemas de Referencias

9- 5.1 Sistema de referencia geodésico local

Un sistema de referencia local queda definido por la elección de un elipsoide de referencia (superficie más simple que el geode), y un acimut de partida, punto origen conocido como *Punto Datum*. De esta forma se establece su ubicación en relación con la forma física de la Tierra que es el geode. Este elipsoide así elegido y posicionado se adapta bien al geode en la zona próxima al punto Datum, creciendo la posibilidad de que a medida que me aleje del punto Datum esta adaptación no sea tan buena.

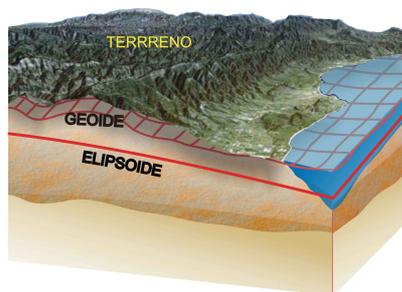


Fig 9.13: Geode vs. Elipsoide

Estos sistemas locales tienen un ámbito de aplicación dentro de un país, o una región del mismo y cada uno trata de que la superficie de su elipsoide se aproxime lo mejor posible con el geode. El ajuste se hace determinando el llamado *punto fundamental* donde se hace coincidir el geode con el elipsoide de referencia elegido.

El Datum es el conjunto de parámetros que definen este punto fundamental. Define entre otras cosas, la posición de origen y la orientación de las líneas de latitud y longitud del sistema de coordenadas.

Estos tipos de sistemas de referencia son definidos en 2 dimensiones (latitud y longitud sobre el elipsoide) y los sistemas de referencia para las alturas se establecen a superficies equipotenciales de la gravedad, en general se trata de que esta superficie sea el geode. ^[12]

El elipsoide queda definido por los siguientes elementos:

- semieje mayor **a**
- semieje menor **b**
- y el aplastamiento **f**, dado por $(a-b)/a$

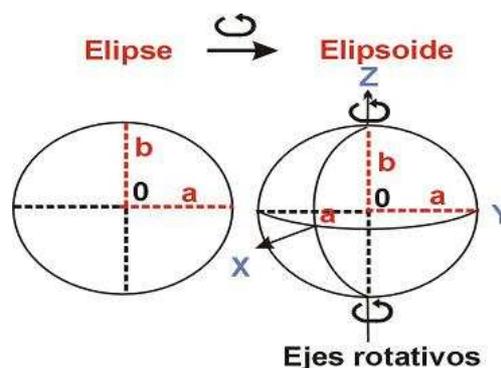


Fig 9.14: Elipsoide de referencia

Hay dos tipos de Datum:

- **Locales.** Alinea su elipsoide lo mas próximo a la superficie de la Tierra y en un área en particular, como es el caso del sistema de referencia Yacaré
- **Geocéntrico.** Usa el centro de masa de la Tierra como origen (WGS84).

9- 5.2 Sistema de referencia geocéntrico

Un sistema de referencia geocéntrico queda definido por los tres ejes cartesianos directos XYZ, fijando el origen en el centro de masas de la tierra, y de tal forma que el plano XOZ contiene al meridiano origen, y el eje OZ es muy cercano al eje instantáneo de rotación terrestre.

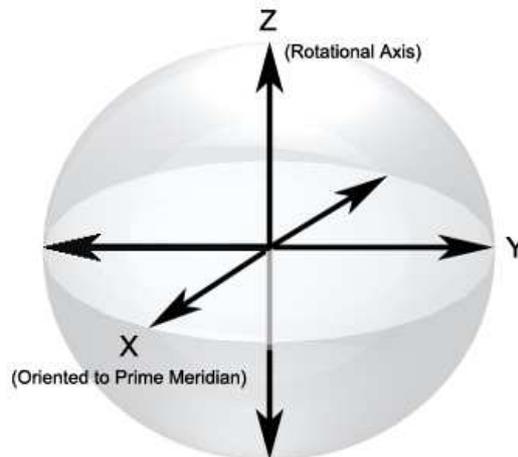


Fig 9.15: Sistema de referencia geocéntrico

El concepto de Datum cambia y ya no es asociado a un punto Datum, si no al origen y orientación de los ejes cartesianos. También es claro que el sistema pasa de ser un sistema 2D a ser un sistema 3D, tridimensional. El WGS84 corresponde a estos tipos de sistemas.^[12]

9- 5.3 Sistema de proyecciones cartográfica

Los sistemas de coordenadas planas o rectangulares surgen como respuesta a los sistemas angulares por la dificultad que estos tienen en medir distancias constantes.

Las proyecciones cartográficas son las que proporcionan una representación en un plano, de la Tierra o una región de ésta, es por lo tanto una relación biunívoca entre coordenadas de un sistema de referencia terrestre (XYZ o latitud, longitud y altura según sea el caso) y coordenadas *Este Norte* de una cuadrícula de un plano.

Obviamente al realizar esta transformación existen deformaciones, y se trata de preservar algunas de las cualidades, según sea la posición que ocupe la parte representada con respecto a la proyección definida, existen deformaciones lineales y defasajes en los ángulos acimutales, de una línea en el sistema real y en el proyectado.

Las proyecciones mas usadas son las que mantienen la forma y se le denominan proyecciones conformes, dentro de las cuales está el Sistema Gauss-Krügger (Transversa Mercator). Todos los ángulos en cada punto son preservados. Las formas de las grandes regiones se distorsionan hacia fuera del meridiano central.

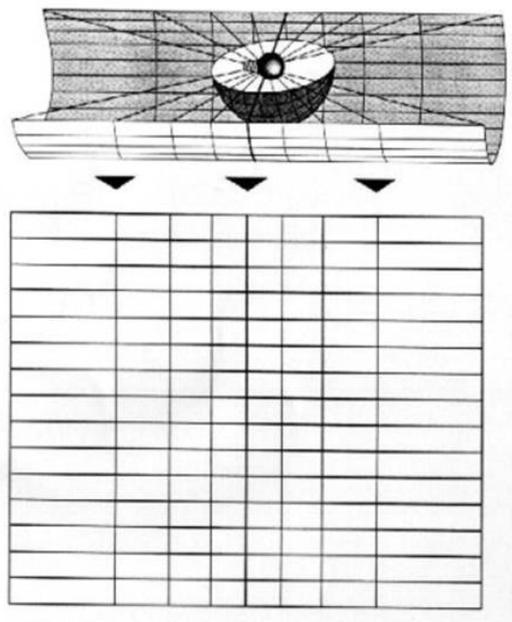


Fig 9.16: Proyección Transversa Mercator

En cuanto a las alturas o cotas en una proyección cartográfica estas pasan a ser un atributo puntual, o se representan a través de líneas de igual cota conocidas como curvas de nivel. ^[22]

9- 6 GPS Diferencial

El DGPS (Differential GPS) es un sistema que proporciona a los receptores de GPS correcciones de los datos recibidos de los satélites GPS, con el fin de proporcionar una mayor precisión en la posición calculada. Se concibió fundamentalmente debido a la introducción de la disponibilidad selectiva (SA).

El fundamento radica en el hecho de que los errores producidos por el sistema GPS afectan por igual, o de forma muy similar, a los receptores situados próximos entre sí. Los errores están fuertemente correlacionados en los receptores próximos.

Un receptor GPS fijo en tierra (referencia) que conoce exactamente su posición basándose en otras técnicas, recibe la posición dada por el sistema GPS, y puede calcular los errores producidos por el sistema GPS, comparándola con la suya, conocida de antemano. Este receptor transmite la corrección de errores a los receptores próximos a él, y así estos pueden, a su vez, corregir también los errores producidos por el sistema dentro del área de cobertura de transmisión de señales del equipo GPS de referencia.

En suma, la estructura DGPS quedaría de la siguiente manera:

- **Estación monitorizada (referencia)**, que conoce su posición con una precisión muy alta. Esta estación está compuesta por:
 - Un receptor GPS
 - Un microprocesador, para calcular los errores del sistema GPS y para generar la estructura del mensaje que se envía a los receptores.
 - Transmisor, para establecer un enlace de datos unidireccional hacia los receptores de los usuarios finales.
- **Equipo de usuario**, compuesto por un receptor DGPS (GPS + receptor del enlace de datos desde la estación monitorizada).



Fig 9.17: Estación de referencia DGPS en edificio de ANP

Existen varias formas de obtener las correcciones DGPS. Las más usadas son:

- Recibidas por radio, a través de algún canal preparado para ello, como el RDS en una emisora de FM.

- Descargadas de Internet, o con una conexión inalámbrica.
- Proporcionadas por algún sistema de satélites diseñado para tal efecto. En Estados Unidos existe el WAAS, en Europa el EGNOS y en Japón el MSAS, todos compatibles entre sí.

En los mensajes que se envían a los receptores próximos se pueden incluir dos tipos de correcciones:

- Una corrección directamente aplicada a la posición. Esto tiene el inconveniente de que tanto el usuario como la estación monitora deberán emplear los mismos satélites, pues las correcciones se basan en esos mismos satélites.
- Una corrección aplicada a las pseudodistancias de cada uno de los satélites visibles. En este caso el usuario podrá hacer la corrección con los 4 satélites de mejor relación señal-ruido. Esta corrección es más flexible.

El error producido por la disponibilidad selectiva (SA) varía incluso más rápido que la velocidad de transmisión de los datos. Por ello, junto con el mensaje que se envía de correcciones, también se envía el tiempo de validez de las correcciones y sus tendencias. Por tanto, el receptor deberá hacer algún tipo de interpolación para corregir los errores producidos.

Si se deseara incrementar el área de cobertura de correcciones DGPS y, al mismo tiempo, minimizar el número de receptores de referencia fijos, será necesario modelar las variaciones espaciales y temporales de los errores. En tal caso estaríamos hablando del GPS diferencial de área amplia.

Con el DGPS se pueden corregir en parte los errores debidos a:

- Disponibilidad selectiva (eliminada a partir del año 2000).
- Propagación por la ionosfera - troposfera.
- Errores en la posición del satélite (efemérides).
- Errores producidos por problemas en el reloj del satélite.

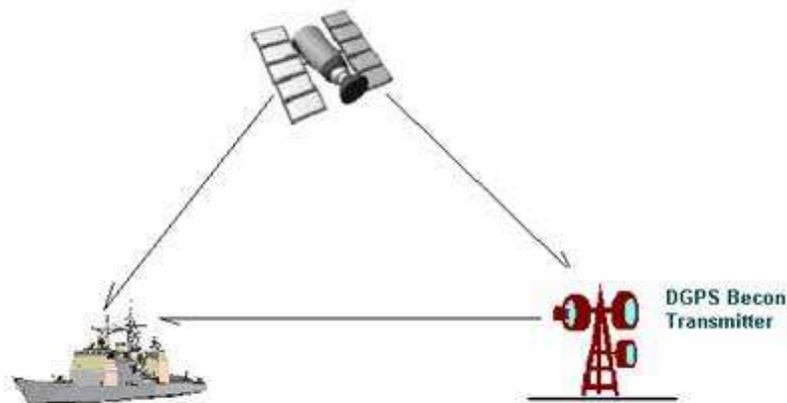


Fig 9.18: Esquema DGPS

Para que las correcciones DGPS sean válidas, el receptor tiene que estar relativamente cerca de alguna estación DGPS; generalmente, a menos de 1 km. La precisión lograda puede ser de unos dos metros en latitud y longitud, y unos 3 m en altitud.^[23]

9- 7 Contenido del CD

En la contratapa del documento hay un CD conteniendo:

- Aplicación (*Yacaré*)
- Documentación del proyecto en formato de archivo *.pdf*
- Artículo del proyecto en formato de la IEEE
- Poster

Los requisitos mínimos para el correcto funcionamiento de la aplicación, anteriormente detallados en el *capítulo 2 -4, pag. 10*, son los siguientes:

Hardware

- Pentium Dual Core 1.6GHz, 1GB de RAM
- Tarjeta de red inalámbrica compatible con 802.11b/g para las adquisiciones WiFi con drivers que soporten wireless extensión v22 o superior (Recomendamos fuertemente utilizar la interfaz WiFi de la OLPC active antenna)
- OLPC active antenna (conocida por Ceibal como *Radio Mesh*) para las adquisiciones Mesh
- GPS compatible con *gpsd* y *gpsd-clients*

Software

Sistema Operativo:

- Fedora Core 8 o superior
- Kernel 2.6-23 o superior

Software adicional:

- Gnome
- Python 2.5
- Gpsd
- Pygarmin
- Gpsd-clients
- Firmware usb8388.bin
- Pcap
- Paramiko
- Numpy
- Matplotlib

10. Referencias

- [1] Programación estructurada.
http://es.wikipedia.org/wiki/Programaci%C3%B3n_estructurada
- [2] 802.11. IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.
- [3] A Technical Tutorial on the 802.11 Protocol. Pablo Brenner.
- [4] Beamforming. Wavion.
<http://www.wavionnetworks.com/Beamforming.html>
- [5] Wireless-tools.
- [6] Iperf. Distributed Applications Support Team.
<http://dast.nlanr.net/projects/Iperf/>
- [7] The IEEE 802.11s Extended Service Set Mesh. Networking Standard.
<http://networks.rice.edu/papers/mesh80211s.pdf>
- [8] Monografía de redes Mesh - 802.11s. Proyecto de Fin de Carrera. Facultad de Ingeniería. UDELAR
- [9] Active Antenna. OLPC.
http://wiki.laptop.org/go/Active_Antenna
- [10] Iwpriv. Wireless driver. OLPC.
http://wiki.laptop.org/go/Wireless_Driver_README#IWPRIV_COMMAND
- [11] GPS: Descripción, Análisis de errores, Aplicaciones y Futuro. Dpto. de Tecnología Electrónica. Universidad de Málaga.
- [12] Parámetros de Transformación entre el Sistema SIRGAS95 y los Sistemas Locales CDM y ROU-USAMS (Yacaré). Facultad de Ingeniería. UDELAR.
<http://www.fing.edu.uy/ia/deptogeo/proyectos/parametros.pdf>
- [13] Sistema de Información Geográfico.
<http://www.procoopsrl.com.ar/sistgis.htm>
- [14] GPS 18 Technical Specifications. Garmin.
http://www.garmin.com/manuals/425_TechnicalSpecification.pdf
- [15] Definiciones GPS.
<http://www.jcea.es/artic/gps-definiciones.htm>
- [16] GPSD - a GPS service Daemon.
<http://gpsd.berlios.de/gpsd.html>
- [17] Glade - a User Interface Designer for GTK+ and GNOME.
<http://glade.gnome.org>

- [18] Shelve - Python object persistence. Python Library Reference.
<http://www.python.org/doc/lib/module-shelve.html>

- [19] CWNA: Certified Wireless Network Administrator Study Guide. David D. Coleman, David A. Westcott. Sybex. 2006.

- [20] DHCP.
<http://es.wikipedia.org/wiki/DHCP>

- [21] Estudio RFC 2131 (DHCP). Sebastián Peñaloza.

- [22] Conceptos generales de Georeferenciación y Cartografía. Sistema de Información sobre Biodiversidad. Gustavo Iglesias.

- [23] DGPS.
http://es.wikipedia.org/wiki/DGPS#GPS_Diferencial

- [24] Python_wifi. Python Library
<http://pypi.python.org/pypi/python-wifi>

- [25] Pydhcplib. A dhcp client/server library in pure python.
<http://pydhcplib.tuxfamily.org>

Proyecto Yacaré – Analizador de RF para Plan Ceibal

Martín Irazoqui

UDELAR. Facultad de Ingeniería
Montevideo, Uruguay
mirazoqui17@gmail.com

Juan Andrés Rodríguez

UDELAR. Facultad de Ingeniería.
Montevideo, Uruguay
juanandres84@gmail.com

Hernán Susunday

UDELAR Facultad de Ingeniería.
Montevideo, Uruguay
hsusunday@gmail.com

Abstract

En este artículo, describiremos el desarrollo del Proyecto Yacaré, una herramienta que unifica el análisis y los relevamientos de señales para redes WiFi y Mesh, con el posicionamiento geográfico mediante un receptor GPS

INTRODUCTION

El Proyecto Yacaré es un proyecto de fin de carrera del Instituto de Ingeniería Eléctrica de la Facultad de Ingeniería de la Universidad de la República. Éste surge como una necesidad técnica del Plan Ceibal, para facilitar las tareas de relevamientos en las etapas de planificación, mantenimiento y optimización de las instalaciones.

Historia Previa

El Plan Ceibal es un programa creado en Diciembre de 2006 por el Gobierno de la República. Éste programa es una implementación del plan creado por OLPC y tiene como objetivo central promover la inclusión digital, entregando una laptop a cada niño de las escuelas públicas de manera de promover el trabajo en red de manera colaborativa así como abrir la ventana al mundo que es el acceso a Internet. Para llevar a cabo el plan, el sistema de conectividad de Ceibal brinda a los usuarios conexión a Internet a través de Access Points WiFi ubicados en las escuelas y en diferentes puntos de las localidades a donde se ha extendido. Uno de los atributos más importantes de las laptops distribuidas por Ceibal es el uso de redes Mesh permiten a los usuarios la conexión a Internet sin necesidad de estar en zona de cobertura de Access Points, o formar redes aisladas donde pueden compartir programas, documentos y juegos. Cabe destacar que las redes Mesh implementadas por los laptops XO se basan en el draft 802.11s de la IEEE.

Necesidad Técnica

El desarrollo del Plan Ceibal se ha dado de forma vertiginosa en los últimos 18 meses, se encuentra en plena fase de crecimiento pero a su vez el mantenimiento y optimización de las instalaciones ya existentes se realiza en paralelo.

Para facilitar estas últimas tareas, aparece como una necesidad el contar con una herramienta que permita relevar los datos Señal WiFi - Posición. Dicha información, es de mucha utilidad para los técnicos del Plan Ceibal ya que luego de relevada puede ser presentada sobre un mapa de la localidad y desde ahí planificar los pasos a seguir para lograr los objetivos de cobertura.

Dadas las características de las redes Mesh, las XO que no se encuentran en la ESA de una red 802.11b/g pueden lograr el acceso a la WLAN mediante otras laptops XO que sí estén zona de cobertura. Éstos últimos son llamados Portales Mesh.

Por lo tanto, el estudio referido a las redes Mesh, busca conocer en mejor medida cuanto puede extenderse el acceso a Internet a través de la misma. Es de interés para los técnicos de Plan Ceibal, contar con una herramienta que permita relevar Portales Mesh - Posición.

Definición del Proyecto Yacaré

Luego de entender la necesidad técnica del Plan Ceibal, se crea Yacaré. Es una herramienta que realiza simultáneamente los relevamientos de señales de redes WiFi y la búsqueda de portales de redes Mesh en un área metropolitana, mientras que el posicionamiento geográfico de los relevamientos es realizado con la ayuda de un GPS.

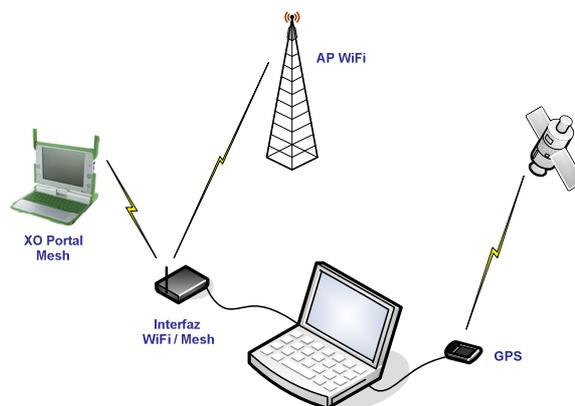


Figura 1: Esquema general

En la figura 1 podemos ver en qué lugar de la topología de la red de Ceibal entra en juego Yacaré.

Posteriormente, la información es guardada en una base de datos y a su vez puede ser exportada en un formato conocido, facilitando su presentación en forma de capas sobre el sistema de información geográfico de Ceibal.

Tiene como objetivos extra definir las interfaces de la aplicación con los actores externos de manera de poder adaptarse fácilmente a nuevas tecnologías y a cambios propuestos por el cliente.

APLICACIÓN

En esta sección nos enfocaremos en el desarrollo de la aplicación integrada y describiremos la arquitectura del software.

Elección del Lenguaje y Estructura de Programación

Python fue elegido como lenguaje de programación ya que es dinámico, orientado a objetos y puede ser utilizado de diversas maneras a la hora de desarrollar software. Con su versatilidad, ofrece al programador un sólido soporte en la interacción con dispositivos, herramientas y otros lenguajes ya que cuenta con librerías y manuales extensos que permiten aprender el manejo en muy poco tiempo.

Manejamos una serie de librerías para interactuar con el GPS, con la tarjeta WiFi y con la active antenna Mesh que facilitaron el trabajo de comunicación con los dispositivos. Python también nos brindó la posibilidad de interactuar de forma sencilla con la interfaz gráfica creada mediante el diseñador de interfaces gráficas Glade. Además las laptops XO poseen un sistema operativo basado en Python lo que nos permite creer que a futuro Yacaré pueda llegar a ser instalado en éstas.

Estructura de Yacaré

Elegimos un modelo de programación modular, de esta manera subdividimos la aplicación en módulos independientes con el fin de hacerla más manejable, adaptable a cambios y a futuras tecnologías. Las cuatro funcionalidades bien definidas, son las que componen estos módulos: la comunicación con el GPS, las adquisiciones en busca de Portales Mesh, los escaneos de la red WiFi y el manejo de los datos relevados. Existen cuatro paquetes de la aplicación que controlan las tareas: *modulo_gps*, *modulo_mesh*, *modulo_wifi* y *modulo_datos*.

Luego de desarrolladas las subdivisiones, se creó un quinto módulo, éste funciona como integrador de la aplicación que también controla la interfaz gráfica, llamado *yacare_gui*. En dicho módulo son mapeadas todas las acciones que realiza el usuario sobre las diferentes funcionalidades del resto de los submódulos.[1]

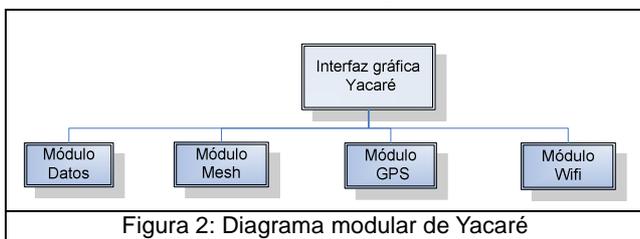


Figura 2: Diagrama modular de Yacaré

Interfaz gráfica de la aplicación

En este capítulo detallaremos a fondo la aplicación gráfica realizada, desde la cual se deciden y controlan los distin-

tos tipos de relevamientos. También explicaremos el manejo que se realiza con los datos adquiridos para la presentación de resultados. Cabe mencionar que este capítulo detalla los diferentes casos de uso de la aplicación funcionando como un solo módulo consolidado, los diferentes submódulos serán descritos más adelante de una manera más específica.

El diseño gráfico de la aplicación fue realizado por el diseñador de interfaces Glade 3.0. Glade es una herramienta que facilita al desarrollador el diseño de interfaces gráficas gtk. Este programa guarda el diseño realizado en un archivo de formato XML que luego es llamado desde Python mediante funciones de las bibliotecas gtk y gtk.glade.

Los diferentes elementos que componen una interfaz gráfica son conocidos como widgets, luego de crearlos en Glade, debemos asignarle una función a cada una de las diferentes acciones que podemos realizar sobre estos. Dichas funciones serán creadas en código Python. [2]

La aplicación tiene una interfaz gráfica orientada a las necesidades del Plan Ceibal, por lo que tuvimos en cuenta los principales eventos que deben ser desplegados en pantalla al usuario que realiza un relevamiento.

Si bien no era prioritario agregar complejidad, durante el transcurso del proyecto nos surgieron nuevas ideas y requerimientos sobre interfaz gráfica. Entendimos que la herramienta resulta más útil de la forma en que la creamos, con varias gráficas y tablas que brindan mucha información al usuario.

La idea principal para este diseño, es que el usuario en todo momento debe tener la posibilidad de chequear el estado del GPS (activo o no) o de adquirir una muestra, mientras puede elegir entre las solapas para ver los últimos adquisiciones o los resultados de los escaneos de redes WiFi así como informarse sobre los diferentes eventos ocurridos chequeando el log.

Como podemos apreciar en la figura 3, la aplicación está dividida en solapas y resultados en la parte superior de la vista y por otro lado los comandos de adquisición con la barra de estado del GPS en la parte inferior.

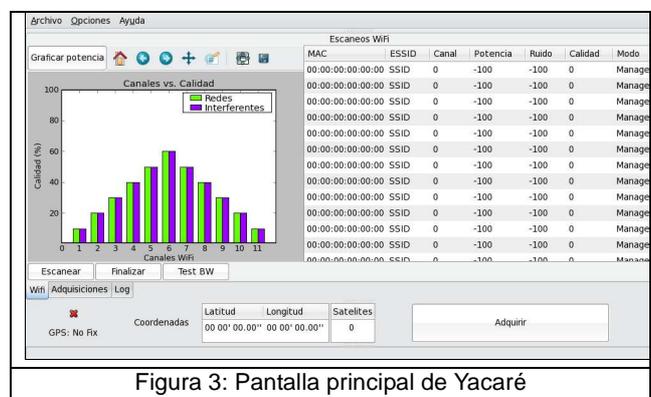


Figura 3: Pantalla principal de Yacaré

ESCAHEO DE REDES 802.11B/G Y PRUEBAS DE ANCHO DE BANDA

En esta sección estudiaremos las formas en las que podemos realizar escaneos de redes 802.11b/g, y las pruebas para el estudio de ancho de banda. Para comprender el concepto de escaneo, y definir los actores que participan de la tecnología, realizamos un estudio del estándar 802.11, donde se detallan tanto la capa física como la subcapa de acceso al medio, MAC. Muchos de los conceptos serán tenidos en cuenta a la hora del estudio de redes Mesh para la búsqueda de Portales.

Al existir dos formas de realizar los escaneos, debimos optar, y para ello se realizaron distintas pruebas y se tomaron en cuenta los antecedentes de los relevamientos de cobertura de Ceibal. A pesar de la sencillez de los escaneos pasivos, optamos por realizarlos activos, ya que ofrece la posibilidad de interrogar al Access Point por información. Para la aplicación que realizamos, no es necesario solicitar información adicional, pero se tiene en cuenta que Ceibal puede llegar a necesitarlo a futuro.

Para la realización de las pruebas de ancho de banda, se priorizó la seguridad de la red y sus integrantes. Esto es, Servidor, Access Points y Laptops. Nos decidimos por generar conexiones Secure Shell con el servidor, y generar un flujo de datos en el sentido cliente servidor, y luego en el sentido contrario. Conociendo el tráfico generado, y capturando el tráfico total cursado, podemos evaluar tanto la percepción del usuario en cuanto a la tasa de transferencia, así como el impacto de la transferencia en la red.

Introducción al Estándar 802.11

El estándar original fue publicado en 1999 por la organización IEEE, Institute of Electrical and Electronics Engineers, y tiene como objetivo brindar conectividad wireless a elementos sencillos, portables que requieren rápido desarrollo. Fue reafirmado en 2003 y 2007. La versión del año 2007, es la que utilizamos para el estudio de la tecnología y:

- Describe las características de un dispositivo que cumple la norma 802.11 para operar en redes de infraestructura como en redes ad-hoc.
- Define los procedimientos de la subcapa MAC para soportar servicios de entrega de datos de forma asíncrona.
- Define distintas formas de señalización de la capa física y funciones controladas por la subcapa MAC.
- Permite la coexistencia de redes inalámbricas de área local (WLAN) 802.11 con otras WLANs que posiblemente estén superpuestas.
- Describe los requerimientos y procedimientos para brindar transporte de datos de forma confiable y segura sobre el medio inalámbrico (WM). y de autenticación con otros dispositivos que cumplen la

norma 802.11.

- Define los mecanismos para selección dinámica de frecuencia (DFS) y control de potencia de transmisión (TPC) que pueden ser utilizados para cumplir las regulaciones locales.
- Define los procedimientos de la subcapa MAC para soportar redes de área local (LAN) con requerimientos de calidad de servicio (QoS), incluyendo el transporte de voz, audio y video.[3]

Debemos diferenciar las distintas versiones de 802.11 en especial entre a, b y g. Todos ellos trabajan en bandas no licenciadas, pero el primero trabaja en la banda de 5GHz, mientras que los modos b y g son para 2.4GHz. En la primera versión del estándar (Legacy), se lograban alcanzar velocidades de 2Mbps. Distintas mejoras en las técnicas de modulación, junto con el desarrollo más económico de los dispositivos de red, permitieron lograr velocidades de transmisión de hasta 54Mbps.

Los factores fundamentales que afectan el diseño, son tanto la movilidad de las estaciones, como la coexistencia con redes en las mismas bandas de frecuencia. Los mecanismos de acceso al medio que implementa la subcapa MAC, tienen técnicas que permiten evitar las consecuencias de tener estaciones que no logran “verse” y evitar colisiones por la utilización simultánea del WM.

Otras dificultades que afectan el desempeño de las redes 802.11, son los distintos dispositivos electrónicos como hornos microondas y teléfonos inalámbricos que utilizan la banda de 2.4GHz. Yacaré no detecta estos dispositivos ya que para relevar el impacto de los mismos deberíamos utilizar un analizador de espectro.

La cantidad de canales en las bandas para 802.11b/g está determinada por las regulaciones locales. En Uruguay, la banda de 2.4GHz se divide en once canales de 22MHz separados 5MHz desde los 2.412GHz hasta 2.462GHz, correspondientes a las frecuencias centrales de los canales uno y once. En la siguiente figura 4 podemos ver el solapamiento de canales del 1 al 14.[4]

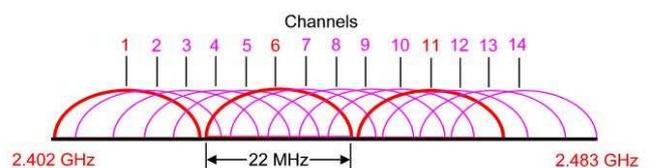


Figura 4: División de canales para 802.11b/g

Escaneo Pasivo y Activo

Los escaneos son parte fundamental del proceso de sincronización para la posterior asociación a un BSS. Permiten identificar los BSSs al alcance de la STA que realiza el escaneo por su BSSID, y devuelve la información sobre el SSID, canal, potencia de transmisión y sincronismo

entre otros. La diferencia fundamental entre el escaneo Pasivo y Activo, es que el primero se dedica a “escuchar” por mensajes llamados Beacons cambiando de canal, y el segundo se envía un mensaje al AP y espera por la respuesta.

Escaneo Activo

El procedimiento para este tipo de escaneo se puede resumir de la siguiente forma:

- Se realiza la técnica de acceso al medio mencionada anteriormente
- Se envía un mensaje Probe Request
- Se inicializa el Probe Timer (contador que junto con las reglas que vemos a continuación permite acelerar y controlar el proceso de escaneo activo)
- Se sensa el canal, y si no se detecta ninguna respuesta antes de que el Probe Timer alcance el valor MinChannelTime (tiempo mínimo de canal) se cambia al siguiente canal y se comienza desde el primer paso
- Si por medio del sensado del canal se detecta que existen redes en actividad, se procesan los mensajes Probe Response luego de que el Probe Timer alcance MaxChannelTime (tiempo máximo de canal)
- Se cambia al siguiente canal y se comienza desde el primer paso a menos que ya se hayan recorrido todos los canales

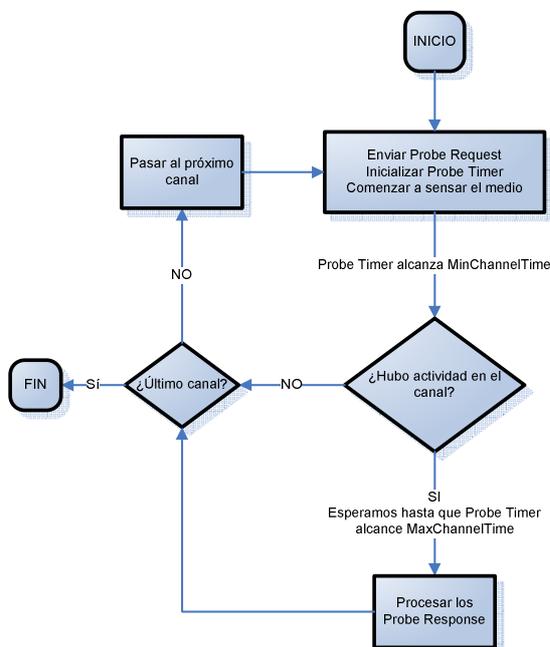


Figura 5: Flujo de eventos de escaneos Activos

A pesar de que los mensajes son de pocos bytes, y el tiempo que se ocupa en el aire es de unos pocos microsegundos, debemos tener en cuenta el factor de que el acceso al medio es conflictivo y produce colisiones en redes de muchas STA. Es una de las razones por las cuales siempre conviene realizar varios escaneos para descubrir las redes al alcance.

Escaneo Pasivo

Como mencionamos anteriormente, consiste fundamentalmente en escuchar Beacons en los distintos canales. La información de los Beacons permite la sincronización entre STAs, para lograr el correcto acceso al medio. Los mismos son enviados periódicamente por los APs (salvo que se deshabilite) con un intervalo de tiempo denominado Beacon Interval (típicamente 102,4ms).

El procedimiento de la STA es básicamente el siguiente:

- Se establece en un canal durante un tiempo máximo definido (MaxChannelTime), típicamente 102,4ms
- Una vez alcanzado este tiempo, procesa los Beacons capturados (si los hay)
- Cambia al siguiente canal

De manera de minimizar el tiempo de escaneo, MaxChannelTime debe ser igual a Beacon Interval.

La información que vemos en los Beacons es muy similar a la que podemos encontrar en los Probe Responses. La diferencia fundamental radica en que en estos últimos se debe pedir la información.. Otra diferencia, es que el mensaje se envía a un Mbps. Es un mensaje de Broadcast, y como tal, debe enviarse en el menor nivel de modulación de forma que todas las STA al alcance puedan recibirlo.

BÚSQUEDA DE PORTALES MESH

Esta sección contiene un estudio de redes Mesh, presentando las principales características del draft 802.11s y la implementación de OLPC. Utilizando conceptos del estándar 802.11 se agregan los conceptos de Mesh Points y Mesh Portal Points así como las nuevas posibilidades que brindan las redes Mesh, como formar topologías de red sin necesidad de infraestructura externa y configurar laptops como portales para proveer de salida a Internet a vecinos que no tengan acceso.

Se estudiarán los mensajes que son intercambiados por los dispositivos que implementan 802.11s para descubrir Mesh Portal Points y formar adyacencias. Nos centraremos en el protocolo DHCP y los mensajes provistos que conforman el mismo, que serán los utilizados por el Módulo Mesh de la aplicación para la búsqueda de portales.

Introducción al Draft 802.11s

Las redes Mesh, se caracterizan básicamente por nodos que se comunican entre sí sin necesidad de infraestructura (Access Points). Forman IBSSs, que son BSSs independientes. Esto quiere decir que las STA se comunican directamente entre sí, y no a través de quien inicia el BSS. Hasta este momento, no hay diferencias con las redes ad-hoc, sin embargo, las STA de la Mesh pueden tener comportamientos distintos, funcionando como Portales o Access Points.

Las XO forman caminos para que quienes no se ven directamente (vecinos), lo hagan a través de un conjunto de STA que pertenezcan a la Mesh. La complejidad radica en el descubrimiento de los vecinos, y en establecer caminos, denominados Mesh Paths, que permitan alcanzar los nodos de la Mesh. En la implementación de las XO, se crean caminos en la Mesh de hasta cuatro saltos.

Cuando una XO logra asociarse a una red externa, automáticamente se comporta como Portal en la Mesh (MPP). Las distintas STAs de la Mesh (denominadas Mesh Points) pueden usar el Portal Mesh para lograr tener acceso a la red externa. En el contexto de Ceibal, la formación de redes Mesh extiende en cierta medida la cobertura de los APs, o en el caso de no contar cobertura en algún lugar, se pueden realizar actividades compartidas utilizando los laptops.[5]

Ruteo en Capa de Enlace

Hay que diferenciar el ruteo al que haremos referencia en la Mesh, del ruteo IP. En la Mesh, los nodos se descubren y establecen caminos en capa de enlace, de forma transparente a las capas superiores. A partir de la información colectada por el protocolo de ruteo (HWMP, Hybrid Wireless Mesh Protocol) construyen su tabla, denominada tabla de forwarding. En base a la misma, deciden a que dirección MAC deben enviar las tramas para llegar a un

determinado destino, el costo (métrica del camino), cantidad de saltos hasta el destino y tiempo de vida de las rutas.

El protocolo por el cual se construye la tabla consiste básicamente en dos mensajes: Route Request (RREQ) y Route Reply (RRPLY). Para el descubrimiento de vecinos, utilizan los métodos mencionados en el estudio de 802.11 para sincronización: escaneos pasivos y activos. Para buscar MPPs, se envían mensajes MPP Request (MPPREQ), el mensaje se propaga por la Mesh, y los portales responden con un MPP Reply. Luego elige el portal con menor métrica en la tabla.

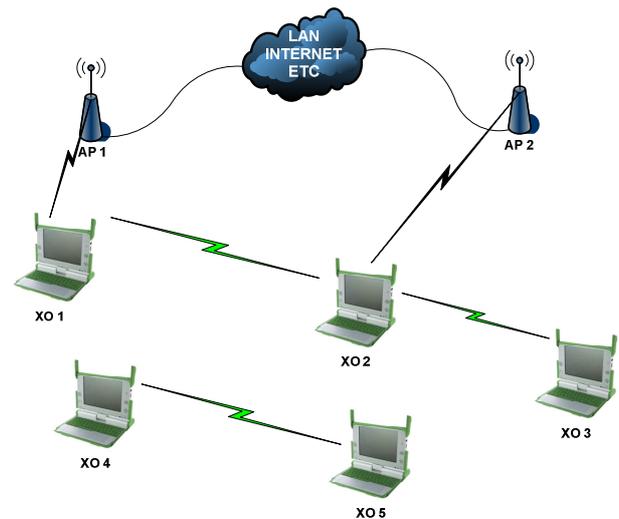


Figura 6: Topologías de redes Mesh

En el caso de la figura 6, vemos como XO1 y XO2 a pesar de estar conectadas a distintos Access Points, se comunican por la Mesh. La XO3, a pesar de no pertenecer a los BSSs de XO1 y XO2 logra comunicarse con ambas. Con XO2, logra hacerlo directamente, mientras que para comunicarse con XO1, en su tabla de forwarding aparece como mejor camino enviárselo a XO2, y que este último lo envíe a XO1. La tabla guarda varios caminos hasta el destino, lo que le permite lograr un gran dinamismo en la topología de las redes Mesh.

Los laptops XO4 y XO5 de la figura 6 forman una Mesh aislada. El IBSS al que pertenecen se identifica por el IBSSID que es una dirección MAC elegida por quien inicia el IBSS, como en cualquier red ah-hoc. Si XO1 se comporta como portal para XO3, este último tiene acceso a la red WLAN, Internet o a la red que XO1 esté asociada, con XO2 como salto en la Mesh.

Cabe destacar que la formación de éstos IBSS, es capturada por el módulo WiFi, ya que responden a los escaneos activos realizados por el mismo.[6]

Quedan definidos entonces los dos actores de interés en el draft 802.11s:

- Mesh Points (MP). Todas aquellas STA que emplean tanto la capa MAC como física de 802.11s.
- Mesh Portal Points (MPP). Todos los MP que permiten que otros MP se interconecten con otras redes.

Mesh Portal Points

Para entender el funcionamiento de estos nodos, iremos viendo como se configuran las interfaces de las XO. Tomemos el caso de una XO que se asocia a un AP de Ceibal, se le asigna vía DHCP una dirección IP en su interfaz eth0 (802.11b/g), por ejemplo: la dirección 192.168.1.2 en la red 192.168.1.0/24, con ruta por defecto 192.168.1.1. Cuando logra la conexión a nivel IP con la red externa, la XO se asigna un IP en la Mesh (actualmente por AutoIP). Luego inicia el servicio de servidor DHCP, que consiste básicamente en ofrecer la configuración IP a quienes la soliciten mediante los mensajes DHCPREQUEST. Se está utilizando actualmente el espacio de direcciones 169.254.0.0/16 de IPv4.

Dado que la interfaz de Mesh y de 802.11b/g operan físicamente sobre el mismo dispositivo inalámbrico, ambas están siempre en el mismo canal, y no podemos decir que una es independiente de la otra. A nivel de capa de enlace, entre las interfaces se establece un bridge, o puente que traduce tramas de 802.11s a 802.11b/g. [5]

Búsqueda de Portales

La búsqueda de Portales puede realizarse de dos formas: enviando MPPREQ y escuchando MPPREPLY y procesando los datos o enviando DHCDISCOVER y escuchando DHCPOFFER. El primer camino implica una manipulación del hardware que es bastante inestable, ya que los protocolos son modificados en las imágenes de OLPC.

Llamaremos cliente Mesh al MP que se quiere asociar a un MPP. Luego de que un cliente envía el mensaje DHCPDISCOVER en la Mesh, el Portal responde con un DHCPOFFER. Al ser un mensaje enviado en Broadcast, no se establece un MP hasta los Portales, sino que la trama es forwardada por la Mesh hasta que su tiempo de vida (cantidad de saltos máxima) supera los cuatro saltos. Debemos luego, generar un Mesh Path hasta cada Portal.

En el paquete DHCPOFFER enviado por el MPP se encuentran los siguientes datos:

- Gateway: su IP en la red Mesh
- IP asignada al cliente: un IP de la red 169.254.0.0 / 16
- IP del servidor DNS: envía al cliente el IP del servidor DNS que obtuvo al conectarse a la red WiFi.

- Mascara de red: 255.255.0.0

El cliente sigue el flujo de la figura 7 y toma una configuración IP como es explicitado en el protocolo DHCP, salvo que descarta la dirección que le envía el MP y se asigna una por AutoIP. Luego si, agrega en su tabla de rutas la red, y define su ruta por defecto como el IP del portal en la Mesh. El Network Manager de las XO es el encargado de llevar adelante la configuración mencionada.

El Portal Mesh, trabaja tanto a nivel de capa de enlace como capa de red (IP). Si la situación es como la que vemos en la figura 7, el MP intermedio no participa a nivel IP en la comunicación entre el MP cliente y el MPP.

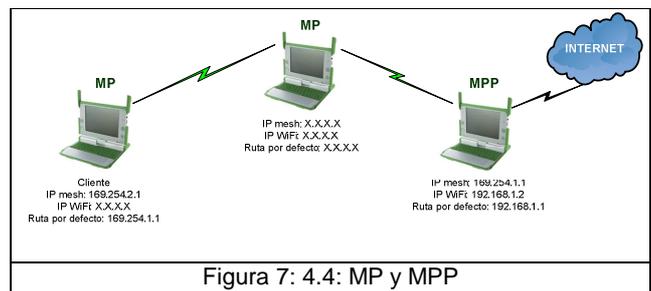


Figura 7: 4.4: MP y MPP

Cuando el MP cliente genera tráfico hacia la red externa (nube Internet en este caso), mira en su tabla de rutas IP el próximo salto y se la envía al MPP. Luego a nivel de capa de enlace, busca en la forwarding table el mejor Mesh Path hasta el MPP, y envía la trama vía el MP intermedio. El MPP puentea las tramas, y a nivel IP, realiza NAT y encamina el tráfico hacia Internet como si fuera generado por él. El camino desde la red externa hacia el cliente es exactamente el inverso.

Active Antenna

La active antenna, en Ceibal conocida como radio Mesh, es un periférico USB que utiliza la misma tarjeta de red que utilizan los laptops XO (Marvell 8388 USB). OLPC desarrolló un servidor denominado XS, que brinda servicios de presencia en la Mesh entre otros. Como interfaz de Mesh, el XS utiliza varias active antenna, funcionando como MPPs. Nuestro objetivo es utilizar ésta radio para encontrar portales en la Mesh, y buscar la cantidad de saltos a los mismos. Utilizamos una versión prototipo del dispositivo.

El driver Libertas desarrollado para las tarjetas está incluido en los módulos de kernel superiores a 2.6.22, para todas las distribuciones. Cuenta con ioctl's que permiten desplegar la forwarding table, y de esa forma registrar los saltos hasta cualquier destino de la Mesh. Para las distribuciones de Fedora 8 y 9, y también en Ubuntu se ha probado que la radio funciona correctamente.

POSICIONAMIENTO GEOGRÁFICO POR GPS

En esta sección se presentan los conceptos relativos al posicionamiento geográfico utilizando un dispositivo GPS. Se presenta la arquitectura de este sistema y se describe el funcionamiento del receptor GPS en cuanto a la información que recibe de los satélites para la obtención de las coordenadas y sus modos de funcionamiento.

Se estudian los distintos sistemas de coordenadas y el concepto de punto Datum, llegando al sistema utilizado en Uruguay, ROU-USAMS mas conocido como Yacaré. Este sistema es el utilizado por Ceibal para la representación de las coordenadas en planos. Ceibal cuenta con el software QGIS, que es un sistema de información geográfica con una importante base de datos de mapas de Uruguay. De esta forma logramos desplegar los puntos obtenidos de los relevamientos de una localidad en mapas con una gran exactitud.

Estos conceptos están integrados en el módulo GPS desarrollado, el cual realiza funciones de control y consulta al receptor GPS mediante un cliente en el puerto del PC. Se implementó una función para la conversión de coordenadas al sistema Yacaré y posteriormente el método de proyección adecuado para poder almacenarlas también como coordenadas planas. De esta forma podemos desplegarlas correctamente en los mapas junto con los datos de interés de las redes WiFi y Mesh.

Arquitectura del sistema GPS

El sistema se descompone en tres segmentos básicos, los dos primeros de responsabilidad militar: segmento espacio, formado por 27 satélites GPS (24 operativos y 3 de respaldo) con una órbita de 20.300km de radio y un periodo aproximado de 12 horas; segmento control, que consta de cinco estaciones monitoras encargadas de mantener en órbita los satélites y supervisar su correcto funcionamiento, tres antenas terrestres que envían a los satélites las señales que deben transmitir y una estación experta de supervisión de todas las operaciones; y segmento usuario, formado por las antenas y los receptores pasivos situados en tierra. Los receptores GPS, a partir de los mensajes que provienen de cada satélite visible (señal de radio en 1517.42Mhz, llamada banda UHF), calculan distancias y proporcionan una estimación de posición y tiempo, además de contener un identificador de satélite, parámetros de su estado de funcionamiento y la fecha y hora con una altísima precisión. [7]

Principios de funcionamiento del sistema GPS

El sistema GPS tiene por objetivo calcular la posición de un punto cualquiera en un espacio de coordenadas (x,y,z), partiendo del cálculo de las distancias del punto a un mínimo de tres satélites cuya localización es conocida. La distancia entre el usuario, receptor GPS y un satélite se

mide multiplicando el tiempo de vuelo de la señal emitida desde el satélite por su velocidad de propagación.

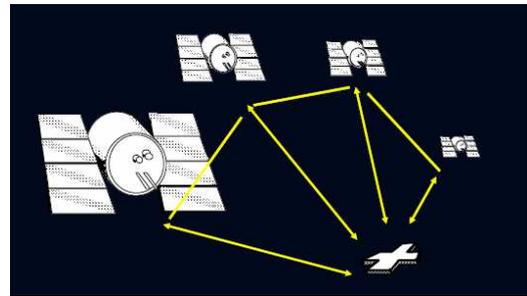


Figura 8 : Estimación de posición

Para medir el tiempo de vuelo de la señal de radio es necesario que los relojes de los satélites y de los receptores estén sincronizados, pues deben generar simultáneamente el mismo código. La desviación en los relojes de los receptores GPS hace necesario un mínimo de cuatro satélites para estimar correctamente las posiciones.

Sistema de Referencia de Coordenadas

La localización geográfica de un punto relevado se expresa en un determinado sistema de coordenadas respecto a un sistema de referencia geodésico.

Para el desarrollo del proyecto Yacaré surge la necesidad de utilizar un sistema de referencia adecuado para Uruguay, con lo cual es necesario convertir los datos obtenidos del GPS, del sistema de referencia global WGS84 a un sistema de referencia local para así poder desplegar la información en mapas de las localidades.

Para Uruguay se utiliza el sistema de referencia ROU-USAMS, conocido como sistema de referencia Yacaré, que se ajusta a las características geográficas de nuestro país.

Sistema de Información Geográfico de Ceibal

Un Sistema de Información geográfico (GIS) particulariza un conjunto de procedimientos sobre una base de datos no gráfica o descriptiva de objetos del mundo real que tienen una representación gráfica y que son susceptibles de algún tipo de medición respecto a su tamaño y dimensión relativa a la superficie de la tierra. A parte de la especificación no gráfica el GIS cuenta también con una base de datos gráfica con información georeferenciada o de tipo espacial y de alguna forma ligada a la base de datos descriptiva. La información es considerada geográfica si es medible y tiene localización.

La mayor utilidad de un sistema de información geográfico es la capacidad de construir modelos o representaciones del mundo real a partir de las bases de datos digitales. [8]

El Plan Ceibal utiliza el ArcGIS y el Quantum GIS (QGIS), este último de código libre en Python y utilizado por el proyecto Yacaré para la representación de la información adquirida de los relevamientos georeferenciada en mapas de Uruguay.

El QGIS cuenta con una importante base de datos con planos de Uruguay en el sistema de referencia de coordenadas Yacaré.

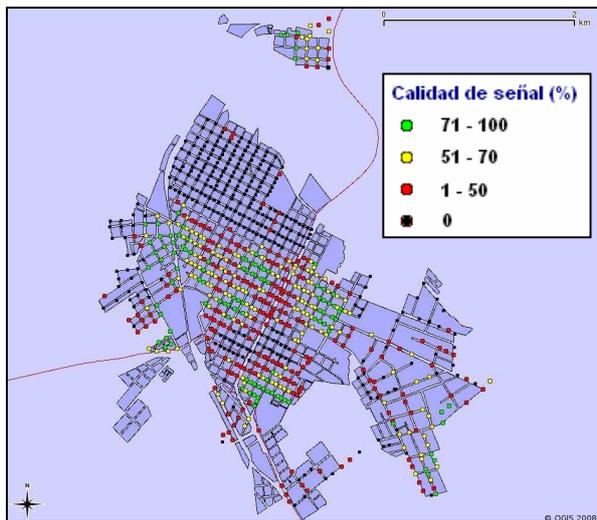


Figura 9: Mapa de Durazno en QGIS

La figura 9 muestra un relevamiento WiFi de la ciudad de Durazno donde los datos fueron desplegados utilizando el QGIS.

REFERENCIAS

- [1] Programación estructurada.
http://es.wikipedia.org/wiki/Programaci%C3%B3n_estructurada

- [2] Glade - a User Interface Designer for GTK+ and GNOME.

- [3] 802.11. IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

- [4] A Technical Tutorial on the 802.11 Protocol.
Pablo Brenner.

- [5] The IEEE 802.11s Extended Service Set Mesh Networking Standard.
<http://networks.rice.edu/papers/mesh80211s.pdf>

- [6] Monografía de redes Mesh - 802.11s. Proyecto de Fin de Carrera. Facultad de Ingeniería. UDELAR

- [7] GPS: Descripción, Análisis de errores, Aplicaciones y Futuro. Dpto. de Tecnología Electrónica. Universidad de Málaga.

- [8] Sistema de Información Geográfico.
<http://www.procoopsrl.com.ar/sistgis.htm>