SISTEMA PARA IDENTIFICACIÓN DE DIRECCIONES MAC Y PROXIMIDAD DE LOS DISPOSITIVOS MÓVILES.

ING. ALEX FERNANDO VILLANUEVA LOAIZA

TRABAJO DE PROFUNDIZACIÓN PARA OPTAR POR EL TÍTULO DE MAGISTER EN INGENIERÍA ELECTRÓNICA

DIRECTOR:

ING. LUIS CARLOS TRUJILLO ARBOLEDA

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

MAESTRÍA EN INGENIERÍA ELECTRÓNICA

BOGOTA

NOVIEMBRE DE 2017

Introducción.

Los servicios basados en localización, más conocidos como, LBS (location based service), han incrementado en gran proporción su importancia en el mercado debido al incremento exponencial del uso de dispositivos móviles como celulares, computadores portátiles y tabletas[1][2].

En general estos servicios de localización han estado más orientados a entornos exteriores, es decir, localización sobre grandes áreas, como por ejemplo ciudades, pero recientemente han aparecido soluciones para localización en interiores, como las oficinas de una empresa, centros comerciales o universidades. Esta aparición en gran medida se debe a que las redes WI-fi se hayan masificado, y es posible aprovechar esa infraestructura para otros servicios.

En el mercado existen diversas soluciones para localización en interiores, pero estas tienen dos tipos de limitaciones principales, la primera es que algunas de estas soluciones requieren acciones por parte del usuario, como instalar una aplicación, o agregar un elemento de hardware. La segunda limitación de estas soluciones comerciales es que funcionan sobre puntos de acceso con protocolos propietarios de los fabricantes, y no ofrecen interoperabilidad entre fabricantes, adicionalmente son muy costosas.

Teniendo en cuenta que las redes públicas de WI-FI [3] tienen una gran participación en entornos comerciales, se presenta la necesidad de encontrar nuevas formas de aprovecharlas, además del esquema tradicional de cobrar por tiempo o ancho de banda. Una posibilidad que está volviéndose más popular es la publicidad, y para que esta sea efectiva se necesita que sea oportuna en contenido, momento y lugar.

Bajo la característica técnica de Inserción de tráfico para usuarios conectados detrás del portal cautivo, se ha generado una serie de necesidades en métodos que ayuden a determinar qué tipo de contenidos y a qué tipo de usuarios se aplicaría este servicio de manera automática con un cierto grado de georreferenciación basado en la proximidad del dispositivo móvil. El objetivo principal del proyecto consiste en desarrollar un sistema embebido (punto de acceso WI-FI), que se pueda colocar en muchos lugares y que permita buscar dispositivos móviles, identificar direcciones MAC y enviar una solicitud de búsqueda al servidor de autenticación, el cual conoce que dispositivos están conectados a la red. Lo anterior con el objetivo de confirmar su conexión al sistema, estimar una distancia respecto al punto de acceso (WI-FI) y entregar al

servidor de autenticación la información para que este haga la inserción de trafico de manera oportuna.

Teniendo en cuenta la importancia de determinar la proximidad de un dispositivo móvil, este proyecto propone diseñar e implementar una solución que supere las limitaciones mencionadas, es decir, que no requiera acciones por parte del usuario, y que no dependa de la infraestructura de los puntos de acceso y tampoco de ningún fabricante en particular, para que sea una solución interoperable.

Marco teórico

Es necesario tener en cuenta las diferentes soluciones actuales dentro del mercado. Para este propósito, se encontraron diferentes mecanismos y tecnologías que se presentan a continuación:

La instalación de aplicaciones en los dispositivos móviles, que utilizan tecnologías como bluetooth. Es un mecanismo muy común, pero presenta las limitaciones ya mencionadas, un ejemplo de esto es el Chirp[4], un producto de la empresa Garmin. Es un dispositivo que funciona como faro, esperando peticiones de los dispositivos que se acerquen, este dispositivo, se deja pre configurado para trasmitir información específica, entre la cual se puede incluir una posición geográfica, luego la aplicación se conecta a un servidor, y con esto se realizan las acciones pertinentes. Esta opción se descarta porque no todos los usuarios están dispuestos a instalar una aplicación; adicionalmente, algunos usuarios no cuentan con un canal de datos disponible para descargar dicha aplicación. También se busca que la solución sea estándar, no tener que diseñar una aplicación para cada tipo de usuario. Otro punto importante, es ofrecer un proceso sencillo y ágil de acceso a los recursos protegidos detrás del NAS (Network Access Server).

Otra alternativa común, es utilizar aditamentos físicos como etiquetas de radio frecuencia que se adhieren a los dispositivos. Para esto, se asocia la etiqueta a un usuario determinado en una base de datos, y en este caso, la que se encarga de la localización es la estación base, que constantemente está buscando etiquetas, y reporta cuando una entra en el rango de proximidad. Esta opción no es funcional, porque el objetivo es manejar volúmenes grandes de usuarios, y el proceso del registro de los usuarios en una base de datos es muy tedioso. Además es costoso estar gastando en etiquetas por cada usuario. También se debe tener en cuenta que algunos usuarios no van a estar dispuestos a pegar una etiqueta a sus dispositivos móviles.

Después de realizar una revisión bibliográfica, se encontraron trabajos similares, que buscan ofrecer servicios de localización a través de canales inalámbricos [1][2], la diferencia principal radica en que los dispositivos móviles son los que analizan su nivel de señal con diferentes puntos de acceso, y usan una función de estimación de distancia con cada uno para poder triangular su posición, en estos casos se tiene la ventaja de que podían caracterizar muy bien el hardware, y habían hecho múltiples ensayos con el mismo dispositivo móvil, de manera que tenían parámetros muy precisos, y sus resultados presentaban márgenes de error relativamente pequeños.

Otra propuesta interesante fue un trabajo donde utilizaban etiquetas RFID [5], similares al modelo comercial mencionado anteriormente, pero en este, se debía agregar etiquetas a los dispositivos móviles, y el usuario era el beneficiado en utilizar los servicios de localización, en esta arquitectura, el dispositivo móvil se comunicaba con unos puntos de acceso diseñados para este propósito, el sistema está diseñado para localización en exteriores, los resultados obtenidos tenían un margen de error aceptable para aplicaciones con cierta tolerancia, pero la conclusión resalta que todavía no es viable para dispositivos en movimiento, donde la velocidad y precisión son necesarias, y por esto, la solución no es competencia para otros tipos de soluciones de LBS.

Una solución que logro buenos resultados de LBS basado en el protocolo IEEE 802.11[6], propone generar una base de datos con posiciones exactas de los puntos de acceso , donde el dispositivo móvil se encarga de estar buscando los mismos, y calcular su ubicación de acuerdo a los que encuentra en su rango, esta solución también está orientada a localización en área urbana, pero en este caso, áreas urbanas muy densas, y dio mejores resultados de desempeño que el GPS, y el AGPS, esto se debe a que en estas áreas tan densas, se presenta mucho ruido, además de factores de atenuación, multi trayectos y desvanecimiento de las señales. Además, se espera que esta solución siga mejorando su precisión con el tiempo, en la medida en que se registren nuevos puntos de acceso a la base de datos, de manera que la solución se va adaptando, e incrementado los puntos de referencia para los dispositivos móviles.

La solución más viable es utilizar el canal WI-FI que está disponible en todos los dispositivos móviles de los usuarios objetivos, pero el proceso normal es que los clientes buscan los puntos de acceso, por lo tanto, los puntos de acceso convencionales no tienen la funcionalidad de buscar clientes. Dentro del proceso de búsqueda se encontró que existen algunos dispositivos que pueden realizar esta función, como por ejemplo el “wifi pineapple” [7], que permite hacer una identificación del canal inalámbrico, y detectar otros puntos de acceso, clientes y sus respectivas direcciones MAC (media Access control). Este dispositivo es muy limitado en recursos computacionales, y su interfaz de programación de aplicaciones es muy limitada para el desarrollo, teniendo en cuenta que no está diseñado para este propósito, otro impedimento, es su costo, ya que no es rentable comprar un producto terminado, pensado para otro propósito, solo para ofrecer una funcionalidad dentro de otro sistema.

Las alternativas más viables que se encontraron, son diferentes software para varias plataformas como Windows o Linux, estos instalan un controlador para el radio WIFI disponible en el equipo, y con este se pueden buscar otros dispositivos que funcionan como clientes, es posible encontrar su dirección MAC y además,

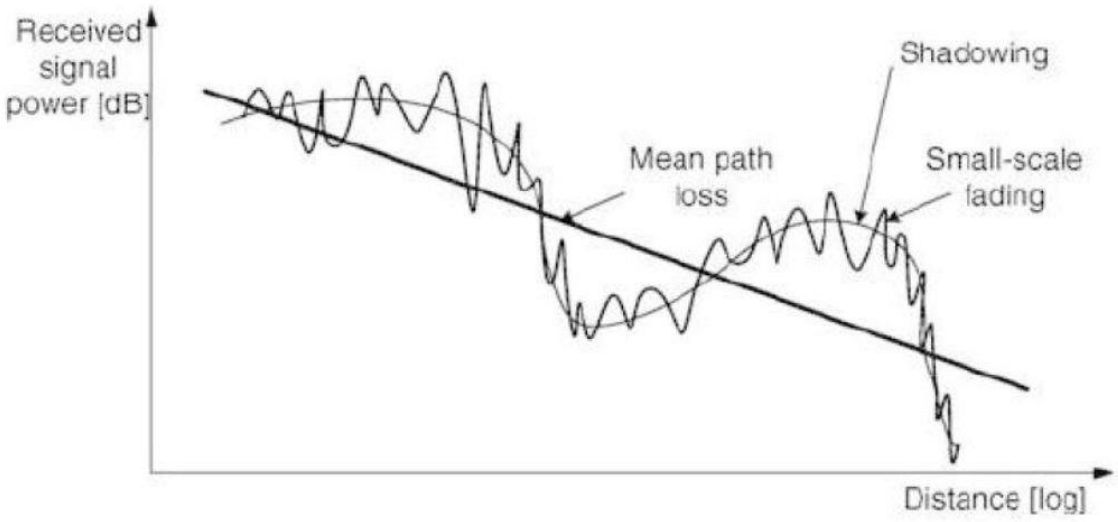
saber si están vinculados a algún punto de acceso. La gran ventaja de este método, es que también se puede tener información de los niveles de señal, de este modo, sería posible hacer un modelo para determinar la distancia en función de esos niveles de señal, para esto, se requiere hacer una buena caracterización del hardware.

Algunas de las herramientas comerciales populares son: “NetSurveyor” [8], y “Acrylic wifi” [9], estas tienen módulos que permiten hacer una búsqueda de los clientes, y sus direcciones MAC. El problema con este tipo de software, es que no se cuenta con un pc completo en cada uno de los puntos, y esto haría que la implementación fuera muy costosa.

Teniendo en cuenta que la solución más viable es utilizar el canal Wi-Fi, haciendo una estimación de la distancia a partir de la información de intensidad de la señal recibida de cada cliente. Se debe considerar que se va a tener un margen de error considerable debido a los fenómenos de propagación y las variaciones de hardware en los diferentes tipos de clientes que se conecten.

En el documento “The Limits of Localization Using Signal Strength: A Comparative Study” [10] se presentan los resultados de un estudio que busca comparar y caracterizas los límites de un proceso de localización en ambientes interiores con el enfoque de intensidad de la señal, así como los errores que se pueden considerar aceptables. Usando tecnología con protocolos IEEE 802.11 se puede esperar un error medio de 3 metros, y un error de 9 metros percentil 97. Se analizaron diferentes resultados, y se espera poder tener resultados en términos de precisión dentro de estos límites.

Estas variaciones se deben principalmente a los fenómenos de desvanecimiento de pequeña y gran escala [11], a continuación se muestra una gráfica de cómo son las variaciones de potencia recibida en términos de una escala logarítmica de la distancia.



Grafica 1, potencia recibida vs distancia (log) [12]

Como se puede ver, la media de perdidas tiene una relación lineal con el logaritmo de la distancia, pero se presentan variaciones debido a los desvanecimientos de pequeña y gran escala.

Se debe tener en cuenta que los desvanecimientos se dividen en dos grupos, de gran escala y de pequeña escala. Los de pequeña escala se dan por multi trayectos o varianzas en el tiempo. Los de gran escala se dividen en Pathloss y desvanecimientos.

Para el desarrollo de este proyecto, se espera tener condiciones de espacio libre entre el sistema a desarrollar y los dispositivos móviles. Por este motivo, se espera que el modelo se asimile a un path loss con variaciones de pequeña escala.

La ecuación para las pérdidas en espacio libre descrita en Wireless communications: principle and practices [11] es:

$$perdidas\ en\ espacio\ libre\ (path\ loss) = \left(\frac{4\pi df}{c}\right)^2$$

Dónde:

d= distancia

f= frecuencia

c= velocidad de la luz

Como se puede observar, las pérdidas son proporcionales a la frecuencia, por lo tanto se debe tener en cuenta cual estándar de IEEE802.11 se va a utilizar, ya que algunos trabajan a diferentes frecuencias.

Otros factores que se deben tener en cuenta en un modelo de propagación son las ganancias de las antenas trasmisoras y receptoras, es posible conocer la ganancia de la antena del sistema que se va a desarrollar, pero las ganancias de las antenas receptoras varían según los diferentes dispositivos de los clientes que se conectan, y esto va a introducir un margen de error en la estimación de la distancia.

Objetivos

Objetivo general

Diseñar, implementar y evaluar un sistema que permita detectar los dispositivos móviles que utilizan un canal inalámbrico para identificar sus direcciones MAC y estimar la distancia a la que se encuentran.

Objetivos Específicos

- Diseñar la plataforma hardware y software del sistema de detección de dispositivos móviles WI-FI
- Definir un modelo de propagación para estimar la distancia en función de la intensidad de la señal
- Implementar un prototipo funcional de la plataforma de detección de dispositivos móviles WI-FI
- Validar la funcionalidad del sistema teniendo en cuenta, una tolerancia de ocho metros en la estimación de la distancia, y un rango de operación de treinta metros.
- Evaluar los parámetros de desempeño mediante el diseño y la ejecución de pruebas en escenarios que simulen las condiciones reales de operación.

Descripción

El sistema a implementar tiene dos funciones principales, identificar la dirección MAC de los dispositivos que estén en rango, y medir la intensidad de la señal de los mismos, para poder estimar la distancia.

El software debe llevar a cabo estas funciones con la información que obtiene de los beacons, todo se va a implementar sobre un sistema embebido, de manera que el sistema sea compacto, y se pueda instalar fácilmente en diferentes locaciones, y buscando que sea de menor costo que las soluciones comerciales, para que pueda llegar a ser un producto competitivo en el mercado.

Este sistema va a ser parte de una red, que posee varios elementos, entre los más importantes se encuentra un servidor de autenticación AAA con protocolo Radius [10], donde están registradas las direcciones MAC de todos los usuarios que se conectan al sistema, este se va a encargar de hacer la validación, y definir si un usuario especifico hace parte o no de la red.

Los clientes de la red, van a estar conectados mediante los puntos de acceso, los cuales van a estar manejados por una controladora, y puede ser posible que cada uno tenga varias redes, y propague diferentes dominios de broadcast, por eso es tan relevante la dirección MAC, porque este es un identificador único de cada equipo, y es independiente de la red a las que se conecte.

Se necesita escoger un modelo de propagación que describa la distancia en función de la intensidad de la señal recibida por el sistema, como por ejemplo, la caracterización del canal de IEEE 802.11 en interiores [15] que se hizo mediante un modelo denominado “Log-distance path loss model and log-normal shadowing”.

Es posible que los diferentes dispositivos tengan diferentes niveles de señal debido a las ganancias de las antenas, el hardware y la potencia a la que trasmiten, de este modo, se deben establecer unas tolerancias, para contemplar los posibles tipos de dispositivos que se conecten. El propósito del sistema no es tener una posición exacta de los usuarios, y su correcto funcionamiento tampoco depende de una estimación muy cercana a la distancia real. Por eso se espera tener unas tolerancias flexibles, y un sistema que se pueda adaptar a todos los tipos de dispositivos comerciales.

De acuerdo a los resultados obtenidos en los estudios consultados [12], se espera que el sistema sea funcional en un rango de aproximadamente 30 metros, y teniendo en cuenta las consideraciones anteriores de la aplicación, se espera que

para esta primera aproximación, se tenga una tolerancia de 8 metros en la estimación de la distancia.

Para lograr un buen funcionamiento del sistema, y que sea compatible con la mayor cantidad posible de usuarios, se espera que la tarjeta inalámbrica del dispositivo funcione en las dos bandas de Wi-FI [3], la de 2,4Ghz, y la de 5Ghz.de este modo, el dispositivo podrá reconocer los protocolos más populares en la industria de IEEE802.11: a,b,g,n y ac. Lo que garantiza su compatibilidad con un gran número de dispositivos comerciales.

Definición de las especificaciones.

Para el proceso de diseño, es fundamental definir las necesidades y características del sistema, para poder traducirlas en especificaciones detalladas del dispositivo.

De acuerdo a lo planteado en los objetivos del proyecto, es necesario que el rango de operación sea de 30 metros, y que se tenga una tolerancia de 8 metros en la estimación de la distancia.

El dispositivo debe conectarse con un servidor Radius para reportar las direcciones MAC y la intensidad correspondiente a esos dispositivos.

El dispositivo debe ser compatible con los estándares IEEE 802.11: a,b,g,n. y debe operar con dispositivos de diferentes marcas que adopten estos estándares.

Se espera poder atender 10 usuarios simultáneamente.

El tiempo de respuesta va a depender del número de canales que se deban escanear, pero es necesario minimizarlo, ya que la información respecto a la proximidad debe ser entregada oportunamente al servidor Radius.

El sistema debe tener la capacidad de funcionar de forma continua durante largos periodos de tiempo.

El sistema debe ser fácil de instalar, y por eso su alimentación debe ser una toma de corriente estándar colombiana (110V, 60Hz, clavijas en configuraciones NEMA de cuchilla recta).

El sistema debe ser gestionable de manera remota, para propósitos de control y monitoreo.

Caja negra

A continuación se presenta el diagrama de caja negra, un método para plantear el Problema de manera muy general. Se muestran las entradas y salidas que se Requieren para el sistema, sin tener en cuenta el proceso de transformación de las mismas. Esto permite tener una idea general de lo que se espera del diseño final. A partir de este modelo, es más sencillo empezar a descomponer en subsistemas. Ya que se presenta como un conjunto de los mismos.

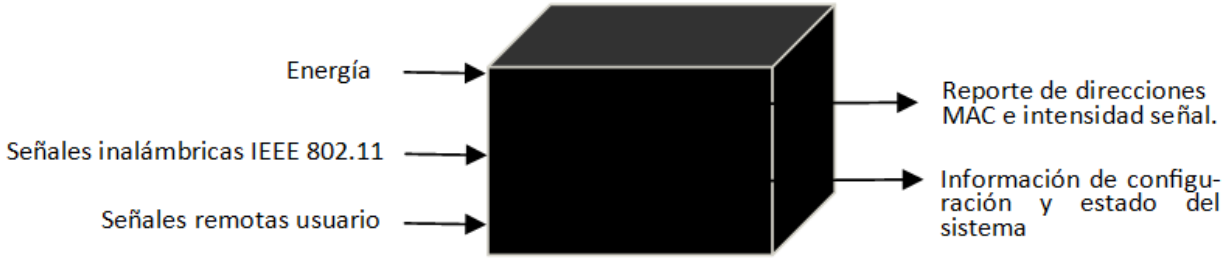


Diagrama caja negra. Se puede ver lo que se espera a nivel funcional de todo el sistema

DESARROLLO DEL DISEÑO

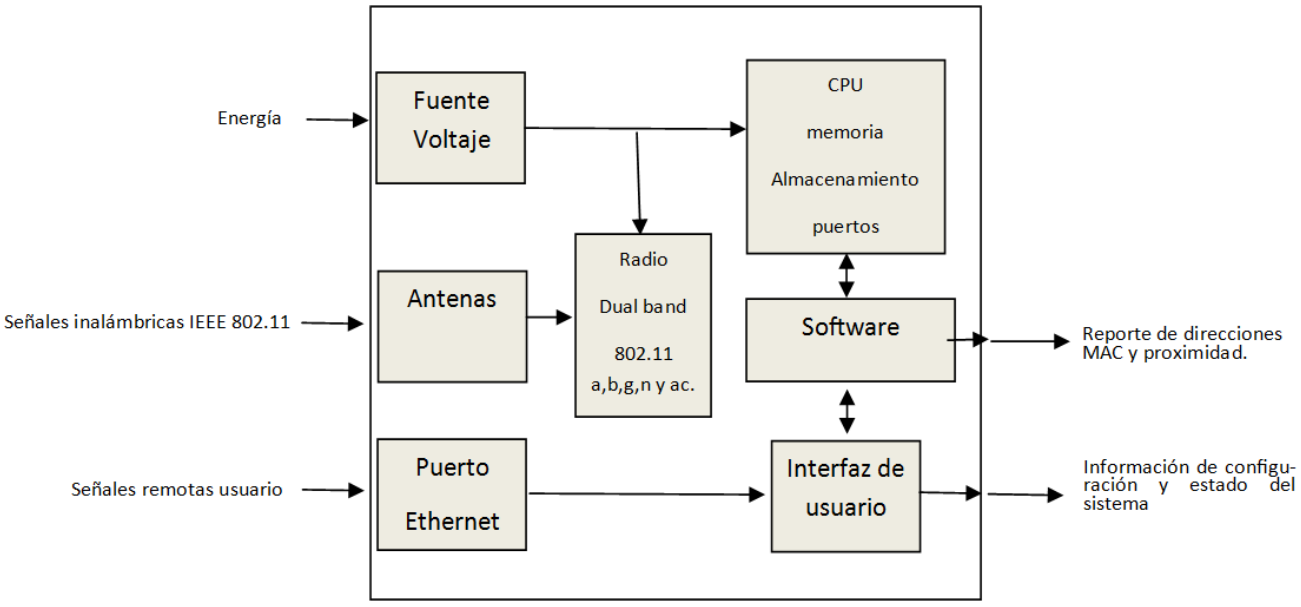
A continuación se describen los diferentes pasos de la metodología de diseño Estructurado y concurrente aplicados al diseño para llegar a la solución requerida.

Descomposición funcional

De acuerdo al análisis de las funciones, se definió una arquitectura funcional del sistema, de los componentes mínimos que son necesarios para cumplir con las necesidades.

Teniendo en cuenta estos elementos, es posible hacer una selección de los componentes para cumplir con las funciones requeridas.

Es posible que durante la implementación, se puedan integrar varios componentes de acuerdo a lo que se consiga en el mercado.



Especificaciones detalladas de la Primera Implementación.

De acuerdo al análisis de los conceptos y alternativas para cumplir con los requerimientos planteados, se presentan las especificaciones detalladas de los componentes del sistema.

Teniendo en cuenta que el sistema debe ser económico en relación a las soluciones existentes, además debe ser compacto, se seleccionó un sistema embebido con los componentes necesarios. Después de una búsqueda en el mercado, se optó por los sistemas embebidos de la marca Gatewroks, que ofrecen sistemas embebidos dedicados a funciones de networking.

Una de las razones de peso para su escogencia, es que ya se cuenta con conocimiento y manejo de esta plataforma, lo que reduce el tiempo de desarrollo porque se reduce la curva de aprendizaje.

Hardware:

Ventana GW5310 Single Board Computer

Este sistema cuenta con las siguientes características:

- Freescale™ i.MX6 800MHz Dual Core ARM® Cortex™ -A9 SoC Processor
- 1Gbyte DDR3-1066 SDRAM Memory
- 256Mbytes Flash System Memory
- Micro SD™ Flash Expansion Socket
- Four High-Power Gen 2.0 Mini-PCIe Sockets
- GbE Ethernet Port
- Power Through Ethernet or Barrel Jack
- Ethernet Power Supports Passive or 802.3af Compatible PoE

Sistema operativo:

Teniendo en cuenta los diferentes sistemas operativos compatibles con el sistema embebido GW5310, se debe tener en cuenta que es necesario tener una plataforma abierta, que facilite el desarrollo del software.

También se busca optimizar los recursos del hardware, por este motivo es necesario que el sistema operativo consuma una cantidad mínima de recursos.

Después de analizar las diferentes opciones, y siguiendo algunas recomendaciones del fabricante, así como la tabla de referencia:

Feature	Yocto	Android	OpenWrt	Ubuntu	Debian	WEC7	Notes
Pre-built images	Yes	Yes	Yes	No	No	Yes	
NAND supported	Yes	Yes	Yes	Yes	Yes	No	
Storage Needed	<256MB	2GB or larger	<256MB	1GB or larger	1GB or larger		1
Build-System	Yes	Yes	Yes	No	No	No	2
Toolchain	SDK/Native	No	SDK	Native	Native		3
GUI	X11/Qt/Qte	Yes	No	Future	Future		4
VPU/GPU	Yes	Yes	No	Yes	Yes		5
Web-Admin	No	No	Yes	No	No		6

Tabla de referencias de los sistemas operativos [17]

Se selecciona el sistema operativo Debían, instalado sin interfaz gráfica, y con los componentes mínimos de funcionamiento. Este sistema operativo permite configurarlo con los módulos que sean necesarios mediante compilación del Kernel, de manera que se optimizan los recursos, y solo se instala los componentes necesarios.

Radio y antenas.

De acuerdo a la recomendación del fabricante, y teniendo en cuenta los requerimientos ya descritos, se selecciona:

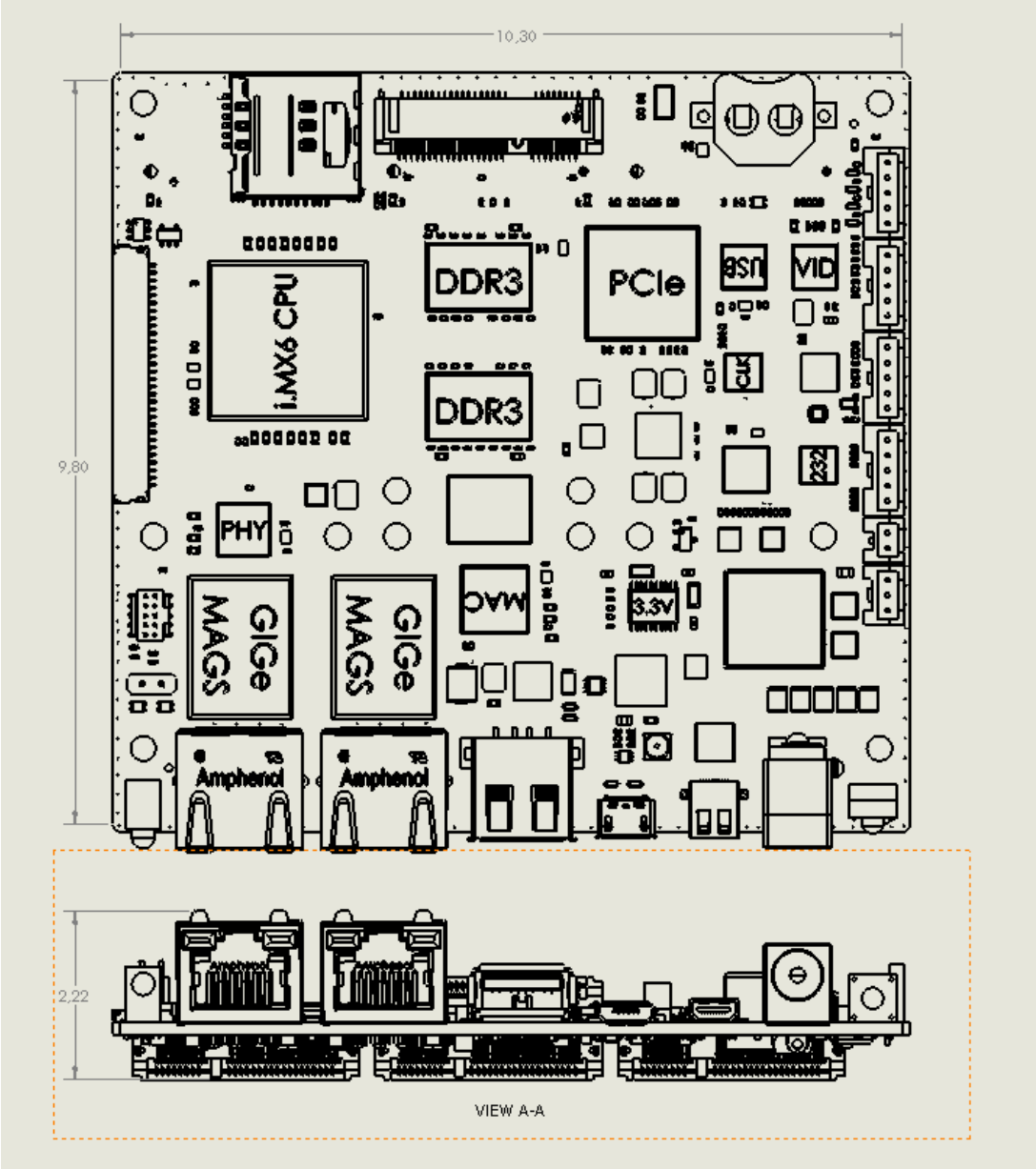
Doodle Labs Dual Band AC WiFi Radio Mini PCIe.

Esta tarjeta de radio es compatible con los puertos mini PCI del sistema embebido, y funciona con los protocolos IEEE 802.11 que se requieren. De manera que solo es necesario tener una tarjeta de radio.

Con un arreglo de tres antenas omnidireccionales de doble banda. Se maximiza la sensibilidad del sistema, y se pueden obtener mejores resultados en la medición de la intensidad de la señal debido a la diversidad espacial [12].

Dimensiones del hardware.

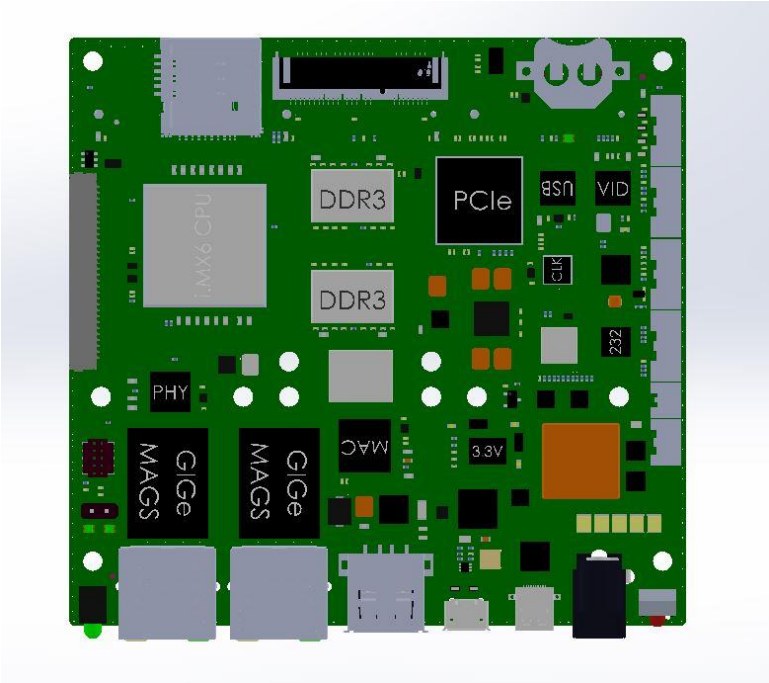
A continuación se presentan las dimensiones del sistema embebido obtenidas del modelo tridimensional que ofrece el fabricante [16]



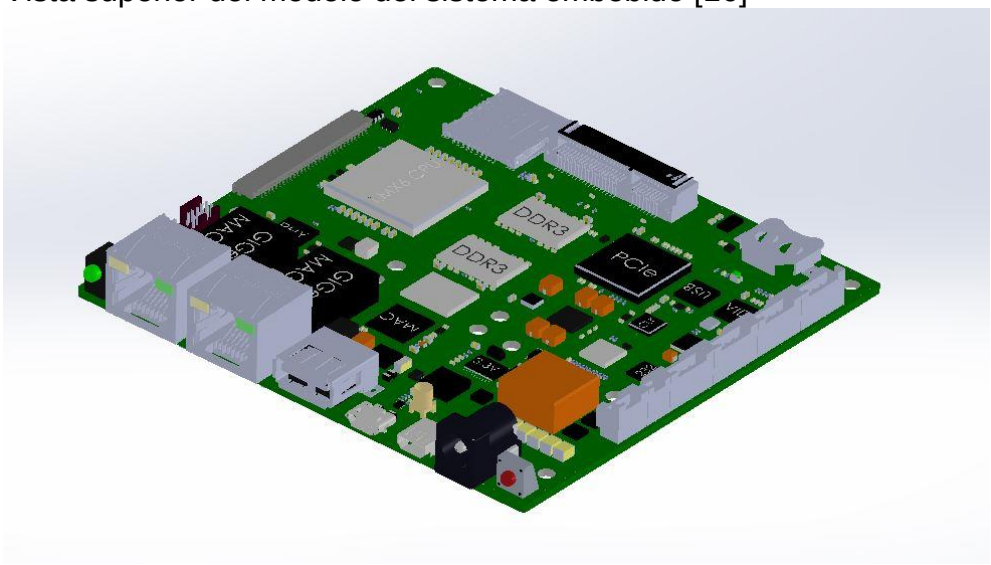
Dimensiones del sistema embebido, todas las unidades en centímetros.[16]

Vistas del modelo tridimensional.

Vistas principales del modelo tridimensional [16] ofrecido por el fabricante, en este se puede observar la ubicación de los componentes principales, así como la distribución de sus puertos.



Vista superior del modelo del sistema embebido [16]



Vista isométrica del modelo del sistema embebido [16]

Diagrama modular de los componentes del sistema embebido

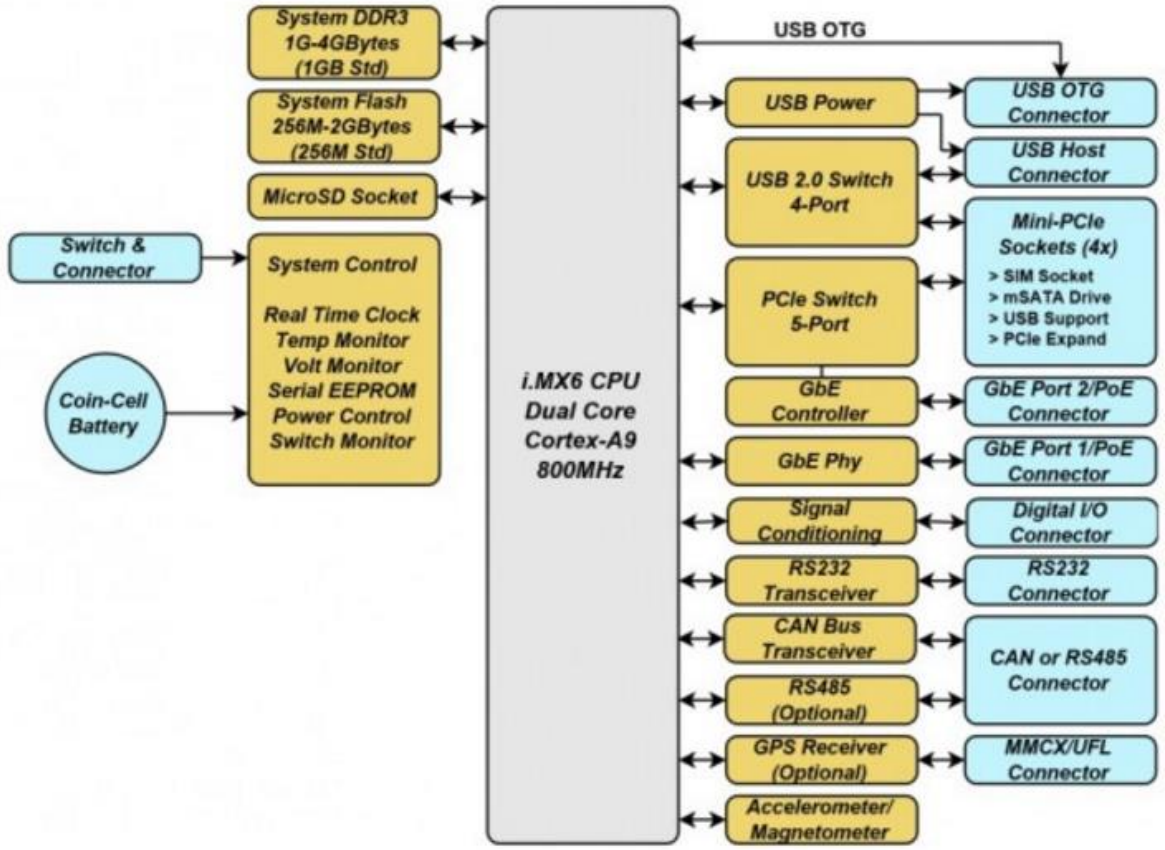


Diagrama de los componentes hardware del sistema embebido. [16]

Se debe tener en cuenta que el sistema tiene varias opciones para su arranque, entre las cuales se encuentran los puertos msata , la memoria interna o un socket para tarjetas Micro SD. El orden de inicio es configurable en el bootloader que viene incluido en el sistema.

También es importante tener en cuenta el hecho de que el sistema embebido cuenta con cuatro puertos mini-pci, teniendo en cuenta que se va a utilizar un radio dual band, quedan disponibles 3 puertos mini-pci para futuras expansiones de hardware del proyecto.

Especificaciones detalladas de la Segunda Implementación

Para la segunda implementación del sistema, se optó por un sistema embebido más económico, y de prestaciones similares a nivel de capacidad de cómputo, aunque un poco más limitado en opciones de hardware como los puertos mini PSI, y el puerto GbE.

Hardware:

RASPBERRY PI 3 MODEL B

Este sistema cuenta con las siguientes características:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM LPDDR2 (900 MHz)
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

Sistema operativo:

Teniendo en cuenta la compatibilidad con la solución anterior, y la necesidad de tener un sistema operativo ligero, se seleccionó Rasbian, una distribución de Linux basada en Debían Jessie.

Esta distribución tiene la ventaja de estar optimizada para el hardware del sistema, y viene con un soporte nativo para Phyton, y otras herramientas enfocadas al diseño de productos para internet de la cosas.

Radio y antenas.

El sistema embebido viene con una tarjeta de red inalámbrica incorporada en la board principal, con un chipset BCM43438 sin embargo, esta tiene una antena soldada sobre el chip, que provee una sensibilidad y ganancia muy bajas, adicionalmente los drivers de este radio inalámbrico no tienen la funcionalidad para monitoreo nativamente incorporada.

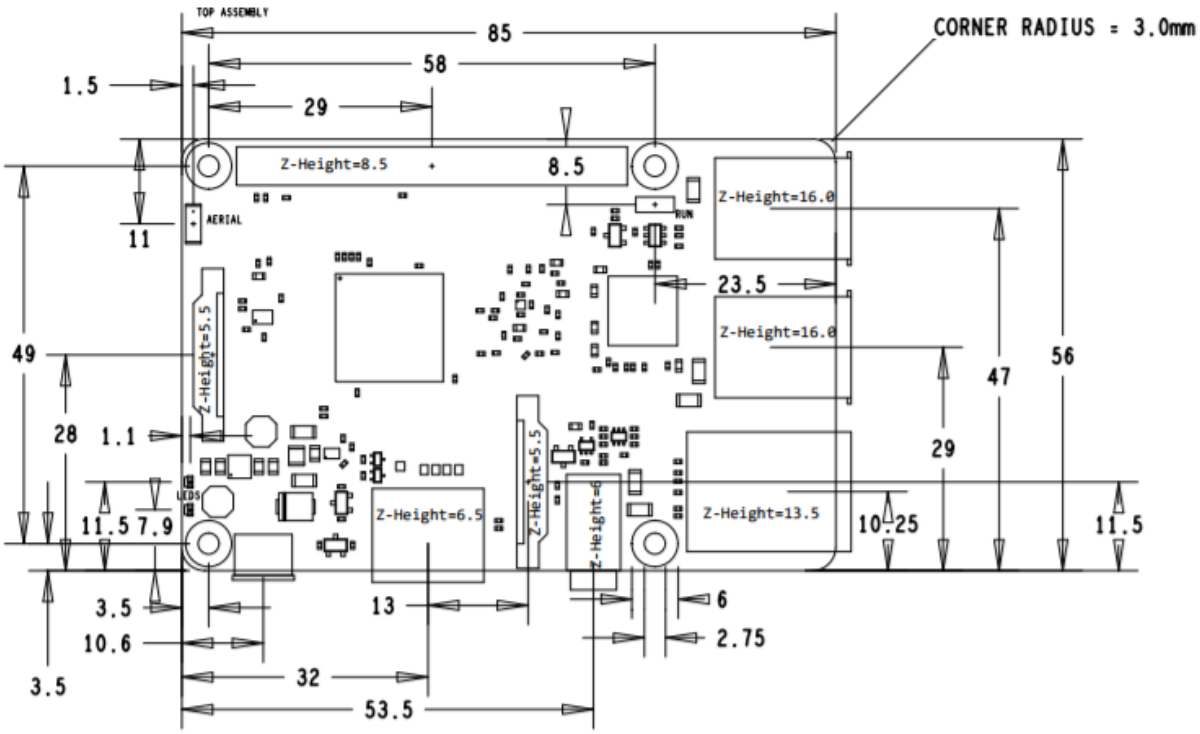
Dada esta limitación, se optó por usar un radio inalámbrico y antena externas al dispositivo. Después de revisar las alternativas con soporte nativo para Debian y que no superaran la limitación de corriente del puerto USB 2.0 del sistema de 500mA para funcionar.

Se seleccionó el modelo TL-WN722N V1 que viene con un chipset Atheros AR9271 con una antena omnidireccional de ganancia 4dBi en el rango de frecuencia de 2,4 Ghz. Y con soporte nativo para el modo monitoreo. Compatible con protocolos IEEE 802.11 que se requieren. [19]

Foto del dispositivo inalámbrico [19]

Dimensiones del hardware.

A continuación se presentan las dimensiones del sistema embebido obtenidas del plano que ofrece el fabricante [18]

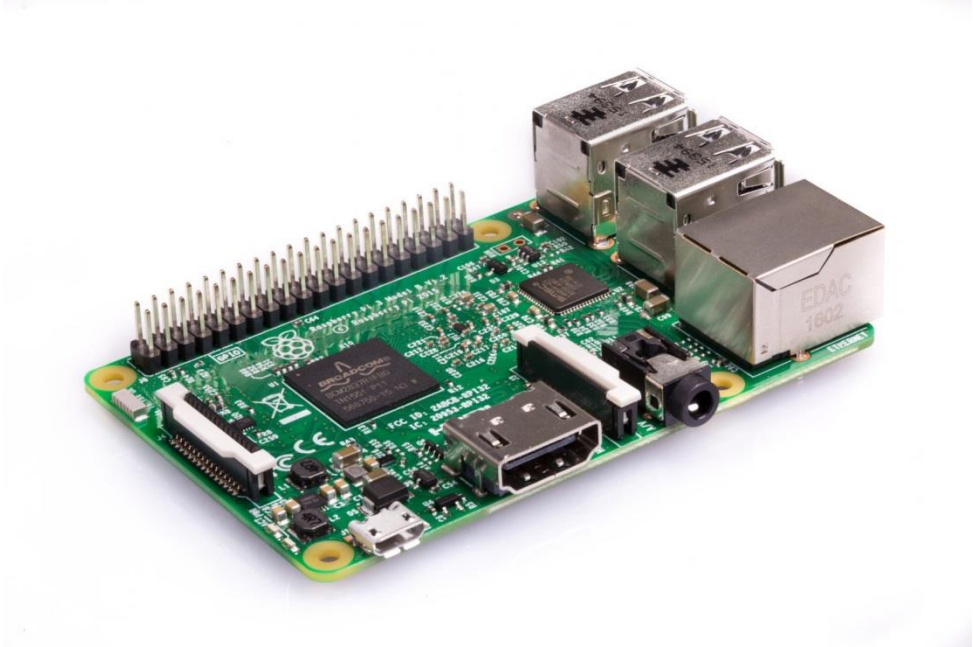


Dimensiones del sistema embebido, todas las unidades en milímetros.[18]

Vistas del modelo tridimensional.

Vistas principales del modelo tridimensional [16] ofrecido por el fabricante, en este se puede observar la ubicación de los componentes principales, así como la distribución de sus puertos.

Vista superior del modelo del sistema embebido [18]



Vista isométrica del modelo del sistema embebido [18]

Diagrama modular de los componentes del sistema embebido

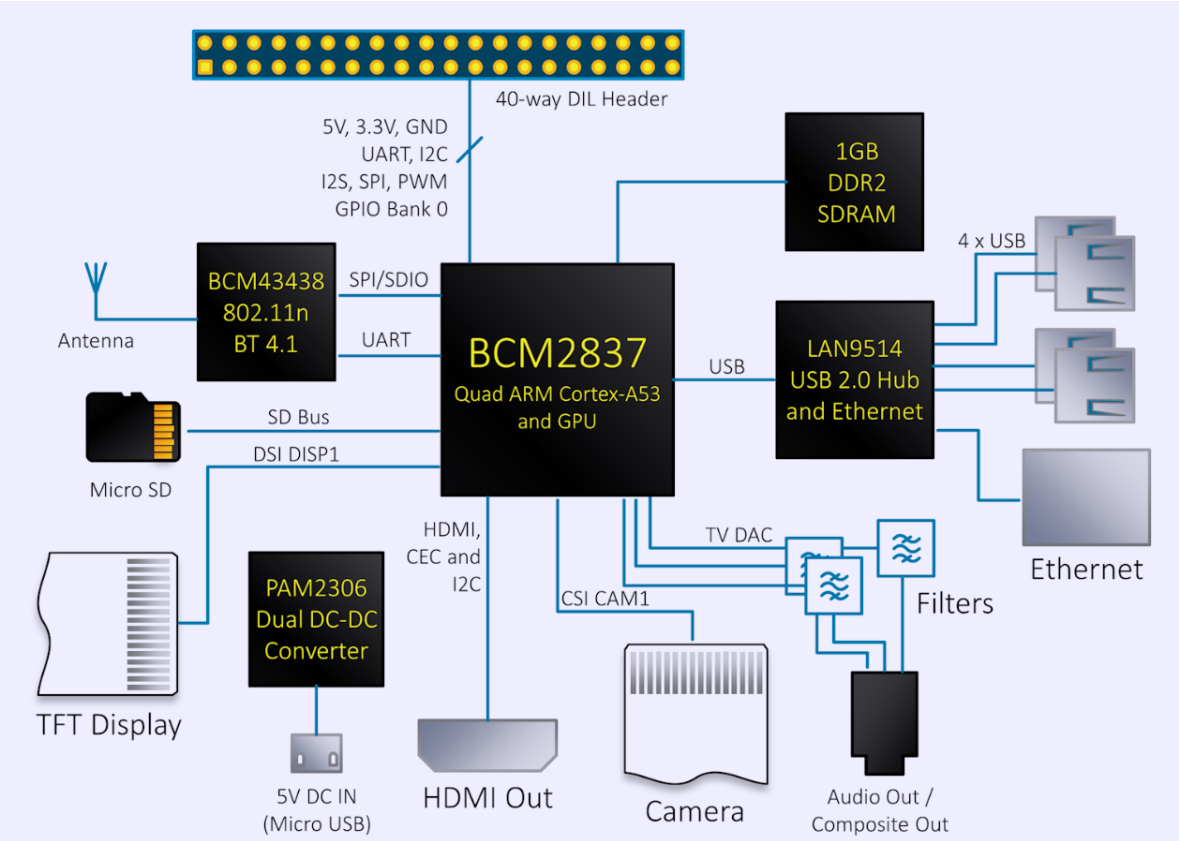
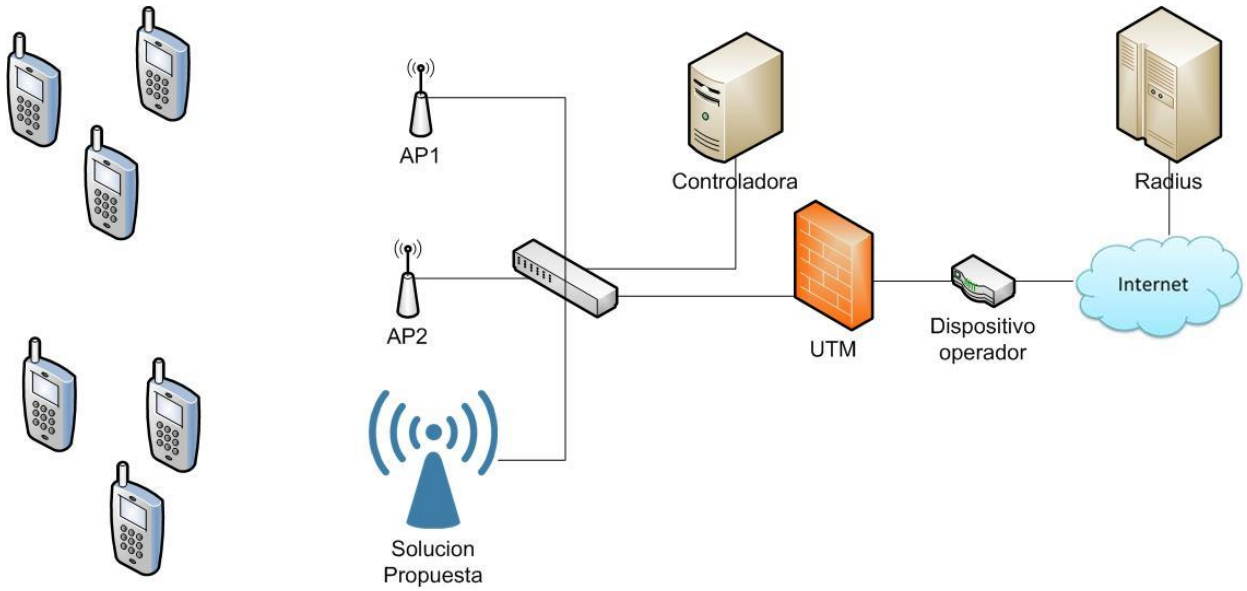


Diagrama de los componentes hardware del sistema embebido. [18]

En comparación con el sistema anterior, este carece de puertos mini PCI, lo cual podría presentar una limitación para expandir las funcionalidades del sistema, sin embargo cuenta con 4 puertos USB, y una interfaz de entradas y salidas de propósito general.

Arquitectura propuesta



Arquitectura propuesta de la solución.

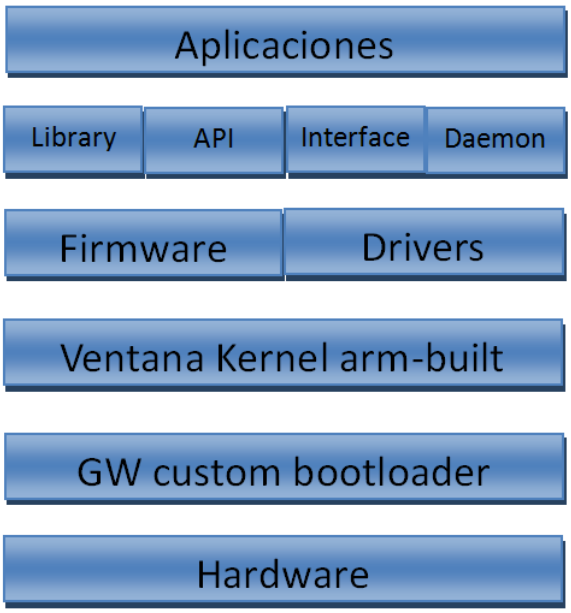
Arquitectura del software

La arquitectura del software es tradicional de un sistema embebido, donde un hardware que soporta todo el software es controlado por un sistema de arranque. Para este sistema, el Kernel está compilado específicamente para el hardware del sistema.

El sistema operativo es tipo UNIX, por lo que el Kernel funciona de interfaz entre las aplicaciones y los periféricos hardware, ninguna otra aplicación tiene acceso directo al hardware, esta característica hace que el sistema sea muy robusto.

Para que las aplicaciones puedan hacer peticiones sobre el control del hardware, están dispuestos los drivers y el firmware de los componentes, estos se utiliza mediante los servicios que el Kernel tiene disponibles.

Las aplicaciones de este sistema van a correr como sub procesos dentro del sistema operativo, y el mismo se encarga de la asignación de los recursos, la comunicación entre las aplicaciones se va a dar mediante interfaces que para el sistema operativo son archivos, y se debe regular el acceso a escritura de las mismas. Las aplicaciones van a estar soportadas por librerías, Apis, Interfaces y demonios todos con licencia de uso libre (GNU).



Aplicación.

Dadas las condiciones de uso del sistema, la aplicación principal es un Loop de Polling, que está diseñado para correr indefinidamente dentro de un subproceso del sistema. Esta aplicación es un clico que llama subrutinas que llevan a cabo las funciones del sistema. Las funciones principales de la aplicación son:

- Revisar interfaces: contantemente el dispositivo está disponible para recibir comandos externos mediante las interfaces de comunicación.
- Recibir Data frames 802.11: este proceso se encarga de recibir y almacenar todos los paquetes de dispositivos móviles cercanos.
- Procesar data frames 802.11: se deben aplicar los filtros correspondientes y extraer la información útil de estos data frames, para poder calcular la distancia del dispositivo móvil.
- Reportar al servidor Radius : de acuerdo a unas condiciones establecidas , el sistema debe reportar las direcciones MAC y las distancias estimadas a un servidor externo.

Funcionamiento de Subrutinas

A continuación se describe el funcionamiento detallado de cada una de las subrutinas y como se comunica con la función principal.

Revisar interfaces

Dado que el sistema embebido dispone de varias interfaces de red, una siempre estará disponible para recibir comandos externos del usuario. Para este propósito, está configurado un subproceso en el sistema operativo que siempre está corriendo y aloja el servicio nativo para SSH.

Para entrar a la consola de manera remota basta con conocer la dirección IP del dispositivo y tener las credenciales de acceso, esta interfaz es compatible con cualquier terminal con soporte para SSH.

Recibir Data Frames 802.11

Para recibir los data frames, la aplicación principal inicia un Daemon en un subproceso. Que corre continuamente y se encarga de recibir los paquetes de acuerdo a los criterios definidos. La interfaz con la aplicación principal se da mediante un archivo plano separado por comas (CSV), que se actualiza según un parámetro definido.

A continuación se presenta un ejemplo de cómo se guarda la información en el archivo plano, y a que datos se tiene acceso:

```
Station MAC, First time seen, Last time seen, Power, # packets, BSSID, Probed ESSIDs
E4:7D:BD:AF:C2:A9, 2017-10-07 00:51:48, 2017-10-07 00:53:48, -69,       17, C4:A3:66:FF:DD:1E,
84:8E:0C:3B:B2:BA, 2017-10-07 00:51:36, 2017-10-07 00:54:07, -47,      143, C4:A3:66:FF:DD:1E,
94:DB:C9:48:75:30, 2017-10-07 00:51:34, 2017-10-07 00:54:12, -42,      966, C4:A3:66:FF:DD:1E,
D0:FC:CC:C8:EA:BD, 2017-10-07 00:51:38, 2017-10-07 00:54:12, -21,      625, C4:A3:66:FF:DD:1E,
```

Ejemplo del archivo plano generado por el Daemon.

Como se puede observar, la información disponible incluye la dirección Mac del dispositivo, la primera y última vez que se detectó el dispositivo, la potencia

recibida del último data frame, el número de paquetes que se recibieron, y la dirección MAC del punto de acceso con el que se comunicó.

Procesar data Frames 802.11

Para procesar los data frames, es necesario leer el archivo plano generado por el Daemon, y convertir la información a un arreglo, para este propósito se utiliza la librería CSV nativa de Python.

Después es necesario almacenar esta información como un arreglo (lista) para poder procesarla. Para la funcionalidad propuesta, solo es relevante la dirección MAC del dispositivo, el momento de la última captura, y la potencia recibida.

Con la potencia recibida se llama la subrutina para estimar la distancia, que utiliza el siguiente modelo:

```
def estm(PRX):
    Prx= PRX #potencia recibida
    import math
    #Parametros de configuracion del modelo:
    n=6 #exponente de pathLoss
    Ptx=14 #potencia trasmitida media
    Freq=2400000000 # Frecuencia
    Gan= 20*n  # producto de las ganancias
    Cons=  32.4 #constante por la velocidad de propagacion.

    resul=  (Ptx - Prx - 2*n*math.log10(Freq)+ Gan-Cons)/ (n*10)
    d= math.pow(10, resul)
    d=format(d, '.2f')

    return d
```

A continuación se presenta un ejemplo de la salida de consola después de procesar el mismo archivo plano del ejemplo anterior:

```
   Direccion MAC      Ultima captura     Prx Distancia
 E4:7D:BD:AF:C2:A9  2017-10-07 00:53:48  -69 9.27
 84:8E:0C:3B:B2:BA  2017-10-07 00:54:07  -47 3.99
 94:DB:C9:48:75:30  2017-10-07 00:54:12  -42 3.29
 D0:FC:CC:C8:EA:BD  2017-10-07 00:54:12  -21 1.47
```

Ejemplo de salida de consola después de procesar.

Reportar al servidor Radius.

Para reportar al Radius, se utiliza el cliente de SOAP para Python ZEEP. Que se ejecuta después de procesar cada iteración del archivo plano. La estructura definida para el XML es la siguiente:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:ser="http://www.talend.org/service/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:Reporte>
         <DatosEstacion>
            <EstacionID>?</EstacionID>
            <clave>?</clave>
            <timeStamp>?</timeStamp>
            <cliente>?</cliente>
            <ubicacion>?</ubicacion>
            <ListaClientes>
<!--1 or more repetitions:-->
               <itemCliente>
                  <MAC>?</Mac>
                  <Distancia>?</Distancia>
                  <BSSID>?</BSSID>
                  <LastSeen>?</LastSeen>
               </itemCliente>
               <itemCliente>
                  <MAC>?</Mac>
                  <Distancia>?</Distancia>
                  <BSSID>?</BSSID>
                  <LastSeen>?</LastSeen>
               </itemCliente>
            </ListaClientes>
         </DatosEstacion>
      </ser:Reporte>
   </soapenv:Body>
</soapenv:Envelope>
```

Estructura definida para el XML.

La estructura está definida con unos datos de cabecera que incluyen datos de la estación que está reportando, estos datos son: un identificador, una contraseña, una marca de tiempo, la dirección MAC, y una referencia de la ubicación.

Dentro de cada mensaje, va una lista de los dispositivos móviles detectados, y de cada uno se incluye: dirección Mac, distancia estimada, y marca de tiempo de la última vez que se detectó.

Protocolo de pruebas

El protocolo de pruebas se divide en tres etapas, Gestión, distancia de un cliente, y medición de proximidad de múltiples clientes. A continuación se describen las diferentes etapas de la prueba y los resultados que se espera obtener.

Las condiciones son que los dispositivos estén trasmitiendo datos mediante Wifi (estándares 802.11 a/b/g/n) y estén en un rango de distancia menor a 30 metros.

El ambiente es un espacio amplio con línea de vista entre el dispositivo y los dispositivos móviles.

Los dispositivos móviles no deben tener opciones de ahorro de energía, deben estar trasmitiendo en los rangos de potencia normal constantemente.

El dispositivo está en una posición fija, mientras que los dispositivos móviles se desplazan variando la distancia, en una prueba exitosa se espera que el dispositivo reporte la distancia aproximada a la que se encuentran los dispositivos móviles durante el tiempo que se monitorea, de manera que se pueda determinar la proximidad.

Gestión.

Esta etapa sirve para probar que el sistema está operativo, y se debe ejecutar de primera. Durante esta etapa se revisan las interfaces estén siendo reconocidas, y estén activas, se verifica que el software este corriendo correctamente y que el dispositivo tiene conectividad. Los pasos a seguir son los siguientes:

1. acceder mediante una comunicación serial o ssh al dispositivo, (seguir instrucciones) para tener acceso a la consola de comandos.

2. ejecutar los comandos ipconfig, iwconfig, ifconfig, de manera que se pueda ver el estado de las interfaces y las conexiones. Las interfaces debe estar en estado “on”

3. Se debe validar que sea posible iniciar el daemon Airodump-ng. Esto garantiza que la interfaz está en modo monitoreo y se pueden recibir Data frames.

Distancia de un cliente.

Esta prueba sirve para determinar qué medidas de distancia se están obteniendo, se debe hacer con el mismo dispositivo, procurando que este con la misma orientación respecto al sistema durante toda la prueba. Y se debe buscar un espacio libre de obstáculos para llevar a cabo la prueba,

1. acceder mediante una comunicación serial o ssh al dispositivo, (seguir instrucciones) para tener acceso a la consola de comandos.

2. ejecutar el comando “iw dev */nombre de la interfaz/* station deump” cada vez que se quiera tomar una medida.

3. colocar el dispositivo a la distancia correcta. Y ejecutar el paso 2. Luego registrar el valor en la tabla. Se recomienda la siguiente tabla de distancias, y medir en el orden presentado con los números azules, de manera que se haga un recorrido alejándose, y luego acercándose, de modo que también se pueda evaluar la repetitividad.

Distancia (m)	valor "signal" (dBm) alejar	valor "signal" (dBm) acercar
1	1	21
3	2	20
6	3	19
9	4	18
12	5	17
15	6	16
18	7	15
21	8	14
24	9	13
27	10	12
28	11	11

4. convertir los valores de intensidad a distancia, usando el modelo propuesto mediante la ejecución de la subrutina de estimación.

Medición de proximidad.

Esta prueba sirve para determinar la funcionalidad del dispositivo, durante esta prueba, el sistema entra en un modo de escaneo y registra las diferentes

distancias de todos los dispositivos que tienen en rango. Para esta prueba, se recomienda tener varios dispositivos que se deben alejar y acercar del sistema.

1. Iniciar la aplicación principal, MAIn.py y validar que empiece correctamente, sin reportar errores en la consola.

2. acercar y alejar los diferentes dispositivos móviles, pasando por el umbral de proximidad que se definió.

3. Revisar los Logs en la consola, donde se pueden verificar los resultados obtenidos de diferentes mediciones.

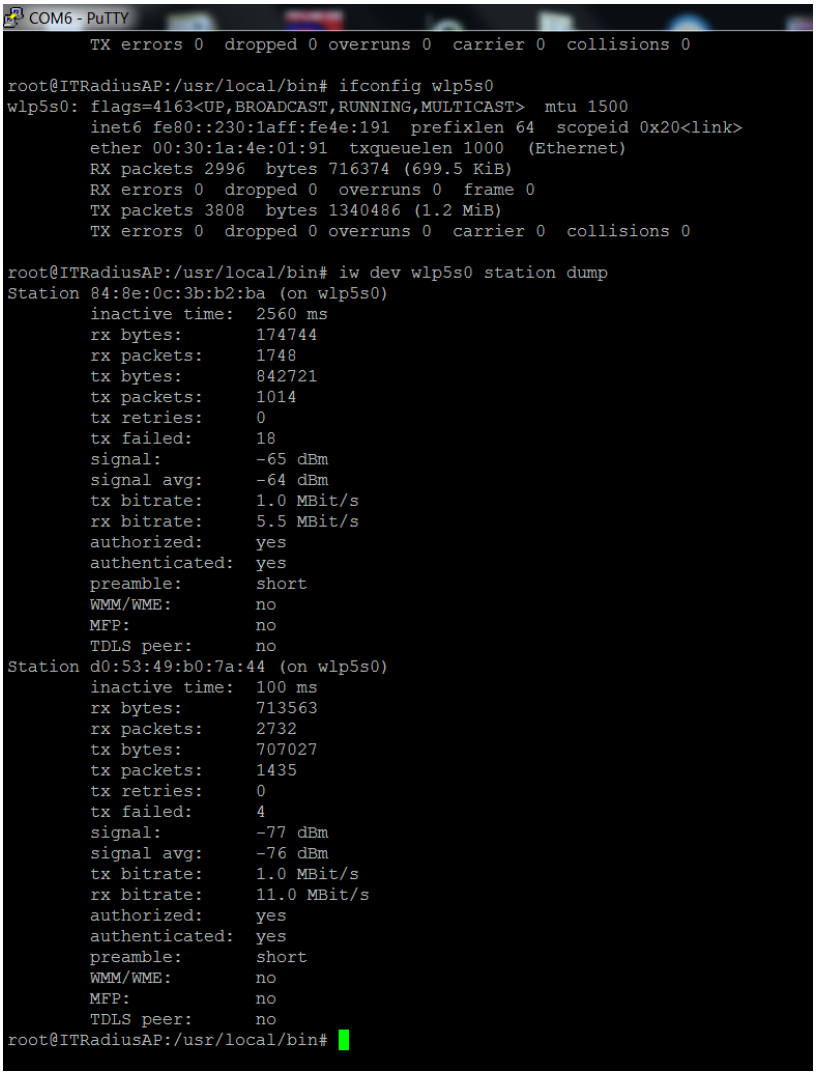
Calibración del modelo.

Cuando se instala el dispositivo en un espacio nuevo, es necesario hacer una calibración del modelo, en la cual se ajusta el exponente de Path loss, y si se hizo algún cambio de hardware como antena, también es necesario ajustar la ganancia.

Para este propósito, se deben medir al menos 3 puntos a distancias conocidas, y ajustar el exponente hasta aproximarse lo mejor posible a la distancia real. Este es un procedimiento de varias iteraciones donde se busca reducir el error, cuanto mejor se calibre el modelo, mas exactos van a ser los resultados de estimación de la distancia para diferentes distancias.

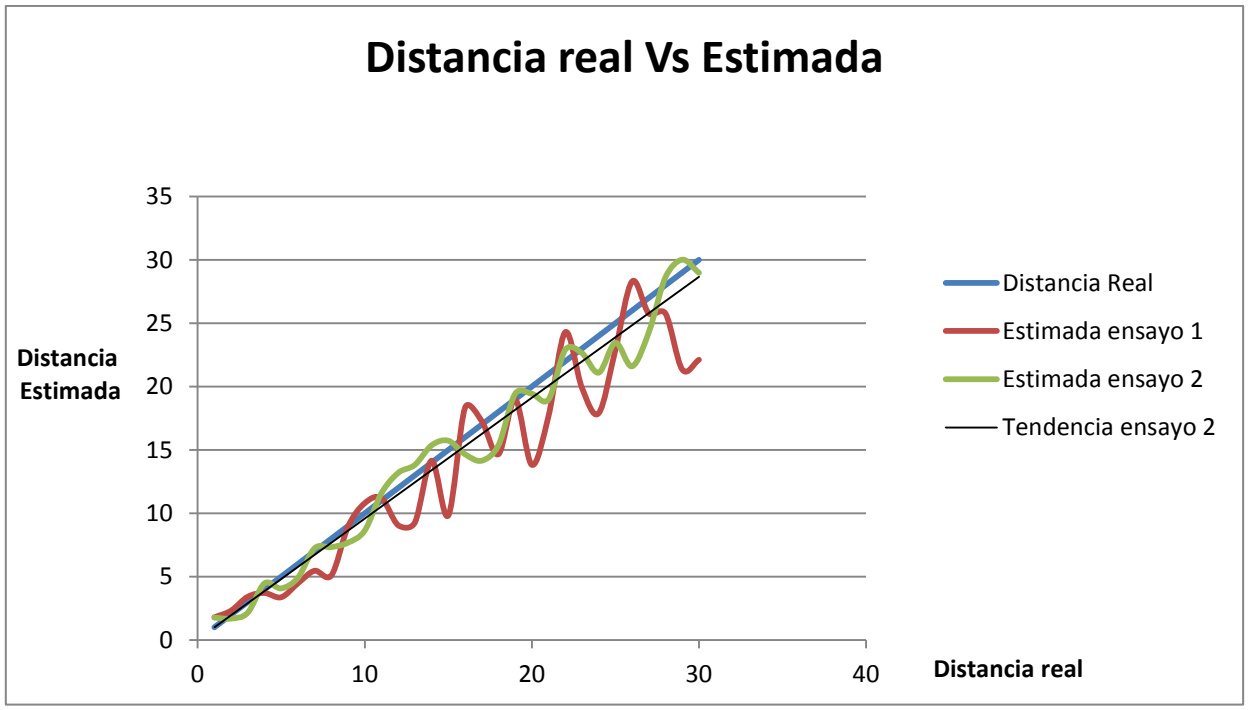
Resultados de la primera Implementación

Las primeras mediciones se hicieron con lecturas directas de los registros de los clientes. A continuación se presenta un ejemplo de dos lecturas, utilizando una conexión serial con el dispositivo.



Muestra de una toma de datos mediante la consola del sistema.

A continuación se presentan las gráficas de distancia real vs distancia estimada para dos ensayos, después de calibrar el modelo. Estas pruebas se hicieron en un entorno indoor, con línea de vista entre el sistema embebido y el dispositivo móvil. Y teniendo en cuenta las condiciones definidas en el protocolo de pruebas.

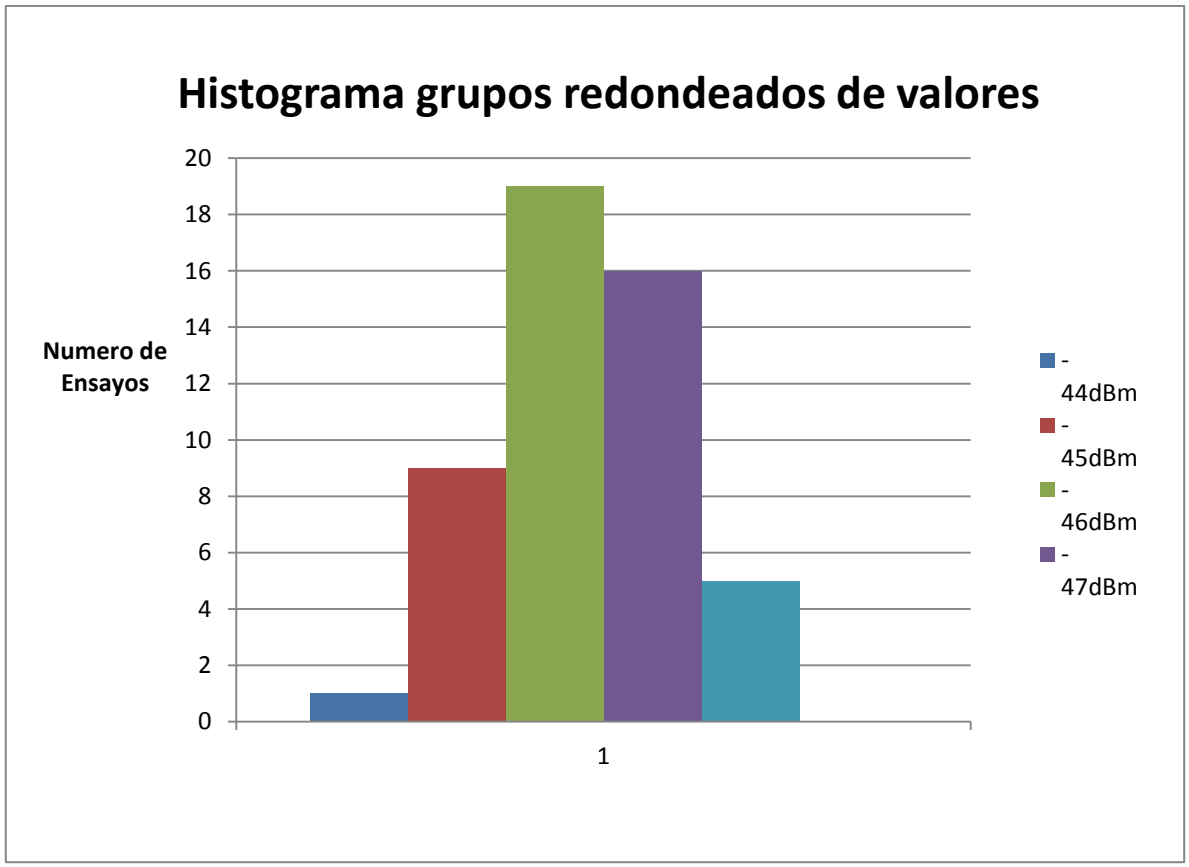


Grafica de ensayos, primera implementación, distancia estimada vs Real.

El error es relativamente alto en los extremos, es decir muy cerca del dispositivo o cuando la distancia se acerca a los 30 metros, pero dentro de un rango de trabajo más cerrado (2<Distancia<25m) se tiene un error medio de 13.1% para el ensayo 1 y 11.2% para el ensayo 2.

Si se tiene en cuenta que el error más alto dentro de ese rango es de 2.9 metros, se puede decir que el sistema cumple con las condiciones necesarias para determinar proximidad, y el error está dentro de los valores aceptables.

Durante los ensayos se observa que la repetición de las mediciones presenta fluctuaciones, por lo tanto se procede con un ensayo a distancia fija, del mismo cliente para analizar el comportamiento de estas fluctuaciones, A continuación se presentan los resultados del mismo:



Grafica de ensayo a distancia fija (10m) para el mismo cliente.

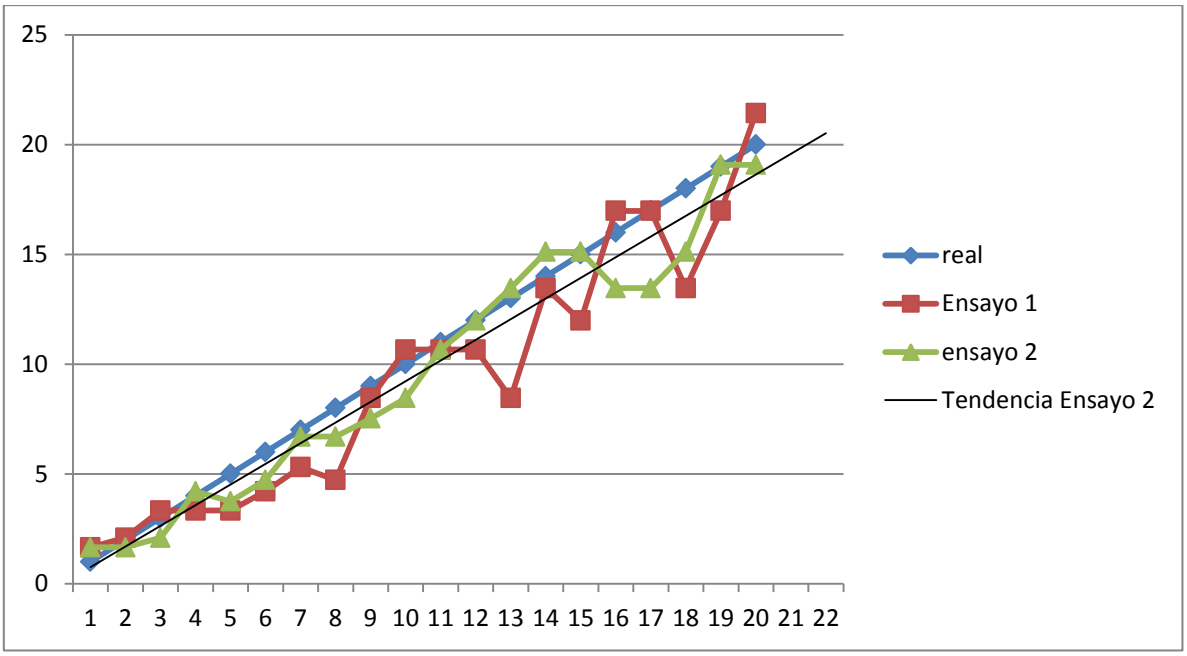
Para este ensayo se tomaron 50 medidas del mismo cliente, a una distancia fija de 10 metros, que en este caso se consideró como el umbral de proximidad. Si se toma un rango de potencias recibidas de -48dBm a -44dBm, el 100% de las muestras quedaron registradas, y si se reduce el rango a -46dBm a – 44dBm, 88% de los valores quedan registrados.

Teniendo en cuenta que el sistema constantemente está monitoreando los data frames de los dispositivos móviles, y que solo se necesita recibir uno con la intensidad de señal suficiente para determinar la proximidad, se puede afirmar que el sistema tiene un nivel de confiabilidad alto si se definen umbrales de proximidad. Con una tolerancia de más o menos 2dBm

Resultados de la segunda Implementación

Para la segunda implementación, se tenía una limitación en comparación a la primera y es que el driver de la tarjeta inalámbrica no reporta valores decimales de la potencia recibida, los valores son enteros, por lo tanto la resolución es menor, adicionalmente, el sistema solo cuenta con una antena, por lo tanto la sensibilidad también es menor.

Para los cálculos, fue necesaria ajustar la ganancia del modelo, ya que también es un poco menor. A continuación se presentan las gráficas de distancia real vs estimada para dos ensayos.



Grafica de ensayos, segunda implementación, distancia estimada vs Real.

El error medio para el ensayo 1 fue de 14,5%. Y para el ensayo 2 de 13.6%, por los factores mencionados anteriormente se obtiene más error en la estimación de la distancia, sin embargo, siguen siendo valores aceptables para el objetivo del sistema.

El error más grande que se presento fue de 3,5 metros, y de nuevo se dio a una distancia lejos del umbral de proximidad. Si se considera el umbral de proximidad como 10 metros, y se analiza un rango cercano a este punto, se presenta un error cercano al 10%.

Los resultados de la segunda implementación tenían un rango de error mayor, sin embargo, para el rango de trabajo propuesto, sigue estando dentro de los valores aceptables, y propuestos en la especificación del sistema.

Entre las causas más comunes de los errores se encuentran el hecho de que las mediciones no presentan buena repetitividad entre ellas, ya que un ligero cambio en las condiciones genera fluctuaciones notables en la potencia recibida, como por ejemplo la orientación del dispositivo, y aunque las antenas se asumen como omnidireccionales, en la práctica no lo son, y el mismo dispositivo puede obstruir la señal según la orientación con respecto al receptor.

Adicionalmente los dispositivos móviles no trasmiten con potencia constante, y aunque se tenía previsto un error porque los diferentes dispositivos trasmiten en los rangos de 12-15 dBi, no se esperaba que el mismo dispositivo varié su potencia de trasmisión a lo largo de los ensayos. Para reducir este error, se podrían promediar varias medidas a la misma distancia, para el mismo dispositivo, pero esto requeriría bar el tiempo de respuesta de la aplicación, ya que en un escenario real, los dispositivos móviles están en constante movimiento.

El dispositivo no tiene la capacidad de monitorear varios canales simultáneamente, por lo tanto tiene que ir escaneando los diferentes rangos de frecuencia, cuantos más canales tiene que analizar, mayor es la cantidad de paquetes que no se reciben, por lo tanto se requiere que los dispositivos móviles trasmitan más información antes de ser identificados. Para minimizar este error, se deben segmentar los canales en los cuales se va a trabajar.

Conclusiones

Después de analizar los resultados obtenidos con el prototipo funcional, y teniendo en cuenta que el objetivo del sistema es determinar la proximidad e identificar direcciones MAC de dispositivos móviles, se puede concluir que el proyecto es exitoso, ya que los resultados se aproximan a lo que se esperaba desde el diseño y concuerdan con la teoría sobre la cual está fundamentado el desarrollo.

El estado actual del proyecto da lugar para algunas mejoras en su desempeño mediante la optimización del software, una rigurosa caracterización del hardware que permita mejorar la precisión del modelo matemático actual, y el análisis de métodos alternos para obtener más información, como por ejemplo incluir fuentes de información adicionales, ya sea mediante diversidad temporal o espacial.

Los dispositivos móviles han obtenido un rol importante en la vida de las personas, y el sistema desarrollado permite obtener información no cifrada de los mismos, sin el consentimiento de los usuarios, por lo tanto es importante informar a las personas del propósito del uso de esta información y garantizar debido manejo de la privacidad de los datos de los usuarios.

Después de la segunda implementación, se demostró que la solución se puede migrar a diferentes sistemas, y no depende de ningún fabricante o componente en particular. Solamente se requiere un ajuste del modelo de acuerdo a las características del nuevo hardware.

Reducir el espectro que se va analizar mediante filtros permite mejorar el desempeño de la solución, ya que aumenta el tiempo efectivo que se reciben paquetes en cada canal, y reduce la cantidad de paquetes innecesarios a procesar.

Durante las pruebas no fue posible saturar el dispositivo por concurrencia de los clientes, dada la alta capacidad del sistema para recibir paquetes de diversas fuentes, pero antes comercializar una solución de este tipo, sería necesario

elaborar un escenario de pruebas con múltiples dispositivos para encontrar el límite real del sistema en cuanto a número de clientes.

Para obtener mejores resultados se debe trabajar en las causas de error mencionadas, sin embargo se debe tener en cuenta que todo representa un compromiso de ingeniería, y reducir el error implica reducir tiempos de respuesta o condiciones más estrictas para el escenario, que en algunas condiciones no son viables o no se pueden garantizar en una implementación.

Bibliografía

[1] D. Park and J. G. Park, "An Enhanced Ranging Scheme Using Wi-Fi RSSI Measurements for Ubiquitous Location," *Computers, Networks, Systems and Industrial Engineering (CNSI), 2011 First ACIS/JNU International Conference on*, Jeju Island, 2011, pp. 296-301. doi: 10.1109/CNSI.2011.29

[2] S. Papadakis and A. Traganitis, "Wireless positioning using the Signal Strength Difference on Arrival," *The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2010)*, San Francisco, CA, 2010, pp. 674-681. doi: 10.1109/MASS.2010.5663794

[3]IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture, IEEE Computer Society [en línea] http://standards.ieee.org/getieee802/download/802-2014.pdf [consultado 15 de marzo]

[4] Garmin, Chirp [en línea] https://buy.garmin.com/en-US/US/prod74811.html [consultado febrero 18]

[5] H. L. Moen and T. Jelle, "The Potential for Location-Based Services with Wi-Fi RFID Tags in Citywide Wireless Networks," *2007 4th International Symposium on Wireless Communication Systems*, Trondheim, 2007, pp. 148-152. doi: 10.1109/ISWCS.2007.4392319

[6] Xingchuan Liu, Sheng Zhang, Jinguo Quan and Xiaokang Lin, "The experimental analysis of outdoor positioning system based on fingerprint approach," *Communication Technology (ICCT), 2010 12th IEEE International Conference on*, Nanjing, 2010, pp. 369-372. doi: 10.1109/ICCT.2010.5689160

[7]Wifi pineapple, wifi pineapple [en línea] https://www.wifipineapple.com [consultado 10 de abril]

[8]Nutsaboutnets, Netsurveyor [en línea] http://nutsaboutnets.com/netsurveyor-wifi-scanner [consultado 10 de abril]

[9] Acrylicwifi, Wifi analyzer [en línea] https://www.acrylicwifi.com/en [consultado 10 de abril]

[10]The Limits of Localization Using Signal Strength: A Comparative Study, Eiman Elnahrawy, Xiaoyan Li, Richard P. Martin

[11] Cho, Yong Soo, et al. (20015) MIMO-OFDM Wireless Communication with MATLAB, John & Wiley, ProQuest. pp 1-22

[12] Rappaport, Theodore S, Wireless communications: principle and practices

[13] NetworkRadius, freeredius [en línea]
http://networkradius.com/freeradius-documentation [consultado 07 de abril]

[14] IETF, Remote Authentication Dial In User Service RFC 2865 [en línea]
https://tools.ietf.org/html/rfc2865 [consultado 18 febrero]

[15]Robert Akl and Dinesh TummalaIndoor “Propagation Modeling At 2.4 Ghz For Ieee 802.11 Networks”

[16] Ventana GW5310 Single Board Computer [en linea]

http://www.gateworks.com/product/item/ventana-gw5310-network-processor
[consultado 5 de octubre]

[17] gateworks ventana family support [en linea]

http://trac.gateworks.com/wiki/ventana [consultado 5 de octubre]

[18] Raspberry PI 3 Documentation [en linea]

https://www.raspberrypi.org/documentation [consultado 23 de septiembre]

[19] TL-WN722N_Datasheet [en linea]

http://static.tp-link.com/resources/document/datasheet/TL-WN722N_ds.zip

[Consultado 10 de octubre]