



FACULTAD DE INFORMÁTICA
UNIVERSIDAD POLITÉCNICA DE MADRID

UNIVERSIDAD POLITÉCNICA DE MADRID
FACULTAD DE INFORMÁTICA

Trabajo fin de carrera
Sistema de posicionamiento en interiores

Autor: Miguel Villalba Capilla
Tutor: José Crespo del Arco

Resumen

El uso de sistemas de posicionamiento como el GPS se ha convertido en algo muy cotidiano hoy en día gracias, en parte, a su implantación de serie en la mayoría de teléfonos móviles. Sin embargo en cuanto a localización en interiores no existe a día de hoy una tecnología que ofrezca las mismas prestaciones.

Sin embargo, existen muchos servicios que podrían ofrecerse en base a la localización en un entorno interior como navegación en aeropuertos, estudios de pautas de conducta en supermercados, envío de ofertas desde las tiendas dentro de un centro comercial, etc.

El principal problema del porqué los sistemas de posicionamiento en interiores no acaban de despegar se debe a la complejidad que presentan éstos, sobre todo por la degradación que sufren las señales ante la presencia de obstáculos tales como muros, paredes, mobiliario, personas etc.

A pesar de ello el estudio y desarrollo de estos sistemas ha tenido un gran auge en los últimos años. La tecnología WiFi parece la más propicia debido fundamentalmente a su fácil acceso y bajo coste. Por otra parte los algoritmos fingerprinting parecen ofrecer buenos resultados en entornos interiores debido principalmente a que no necesitan línea de visión directa entre emisor y receptor.

El presente trabajo de fin de carrera pretende evaluar que rendimiento ofrecen actualmente estos sistemas. Para ello se ha desarrollado uno utilizando WiFi como tecnología y una red neuronal como algoritmo fingerprinting de localización.

Índice de contenidos

1. Introducción.....	5
1.1 Objetivos.....	7
1.2 Organización de la memoria.....	8
2. Revisión del estado del arte.....	10
2.1 Características.....	10
Representación de la localización.....	10
Localización simbólica.....	11
Localización real.....	11
2.2 Tecnologías subyacentes.....	14
Luz.....	14
Sonido.....	16
Radiofrecuencia.....	16
2.3 Algoritmos de localización.....	18
2.3.1 Trilateración.....	19
2.3.2 Multilateración.....	23
2.3.3 Triangulación.....	25
2.3.4 Proximidad.....	26
2.3.5 Método análisis de la escena o fingerprinting.....	27
3. Descripción del sistema.....	31
3.1 Resumen del funcionamiento.....	31
3.2 Entorno y material utilizados.....	32
3.3 Configuración y disposición de la arquitectura física.....	34
3.4 WiFi como tecnología.....	36
3.4.1 Redes 802.11 (WiFi) de infraestructura.....	36
3.4.2 Escaneo de redes.....	37
3.4.3 Control de acceso al medio.....	52
3.5 RSSI.....	54
3.5.1 Definición.....	54
3.5.2 Retos que presenta el uso del RSSI en sistemas de localización.....	55
3.6 Algoritmo de localización: Perceptrón multicapa.....	59
3.6.1 Perceptrón multicapa.....	60
3.6.2 Nuestra red neuronal.....	65
4. Diseño del sistema.....	70
4.1 Diagrama de bloques.....	70
4.2 Módulo sensor.....	70
4.3 Módulo localizador.....	72
4.4 Fichero de test.....	72
5. Caso de prueba. Resultados.....	73
5.1 Caso de prueba 1.....	73
6. Evaluación del sistema.....	75
6.1 Caso de prueba 1.....	75
7. Conclusiones y trabajo futuro.....	76
8. Bibliografía y recursos web.....	79
Bibliografía.....	79

1.Introducción

Los sistemas de posicionamiento tales como el GPS han propiciado la posibilidad de determinar la posición de cualquier objeto con una precisión de unos pocos metros (centímetros si se utiliza el GPS diferencial).

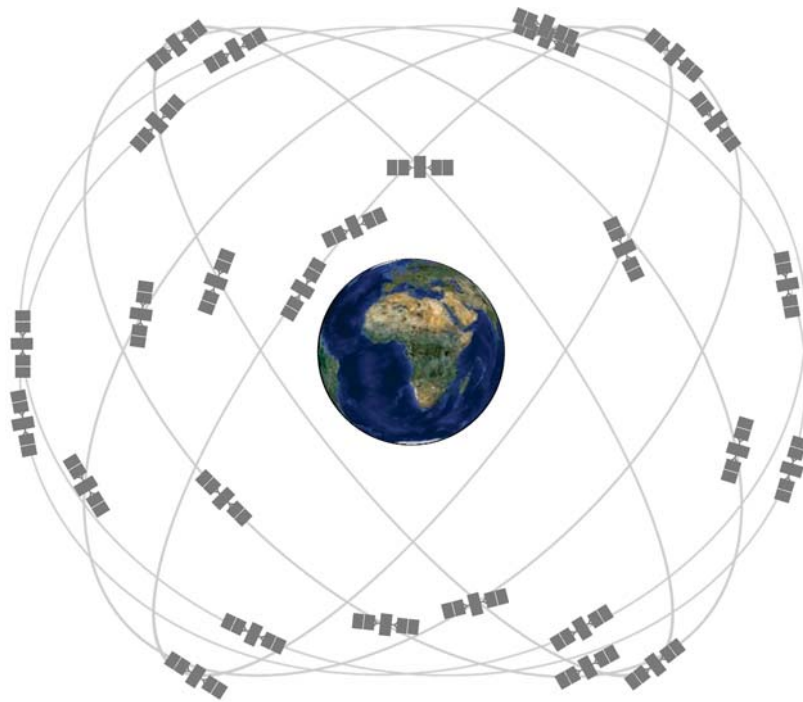


Figura 1.1: Sistema GPS

Esa exactitud ha traído consigo la aparición o mejora de servicios como la navegación aérea, la navegación marítima, la navegación terrestre, tareas de socorro y rescate, servicios de cronometría, topografía, cartografía, etc. Por otra parte la fuerte irrupción de los dispositivos móviles con receptores GPS ha convertido su uso en algo muy popular y cotidiano.

Los sistemas de posicionamiento global funcionan muy bien en exteriores, con una precisión muy alta, sin embargo en entornos interiores la señal sufre una gran atenuación al tener que atravesar paredes o muros y su precisión se ve muy deteriorada. Es por esto que en los últimos años se ha producido un gran auge en el desarrollo de nuevos sistemas que sean capaces de localizar en entornos interiores.

El poder localizar un dispositivo o persona que se halle en el interior de una estructura permite el desarrollo de infinidad de servicios:

Sistema de posicionamiento en interiores

Navegación en interiores

Sería posible poder ubicarnos en aeropuertos, hospitales, museos, centros comerciales y recibir información de navegación en tiempo real para por ejemplo acceder a una puerta de embarque determinada o acceder a los servicios de un centro comercial.

Monitorización

Podríamos monitorizar la ubicación de los niños en un campamento, o conocer la ubicación y constantes vitales de los pacientes en un hospital, la ubicación en una residencia de personas mayores, etc.

Estudios de pautas de conducta

Sabiendo la ubicación y movimientos de los usuarios sería posible elaborar estudios sobre la pauta de compra en supermercados, etc.

Información en tiempo real en base a la ubicación

Podríamos recibir información de ofertas al pasar por una determinada tienda en un centro comercial, recibir información del cuadro que estamos viendo en un museo, recibir los horarios de actividades al pasar por un gimnasio, etc.

Sin embargo, a pesar de las innumerables funciones y servicios que la localización en interiores proporciona su uso no está extendido y no contamos ni de lejos con un sistema de la popularidad y rendimiento como el que ofrece el GPS. Esto es debido fundamentalmente a que la localización en interiores plantea problemas técnicos de más envergadura y difícil solución.

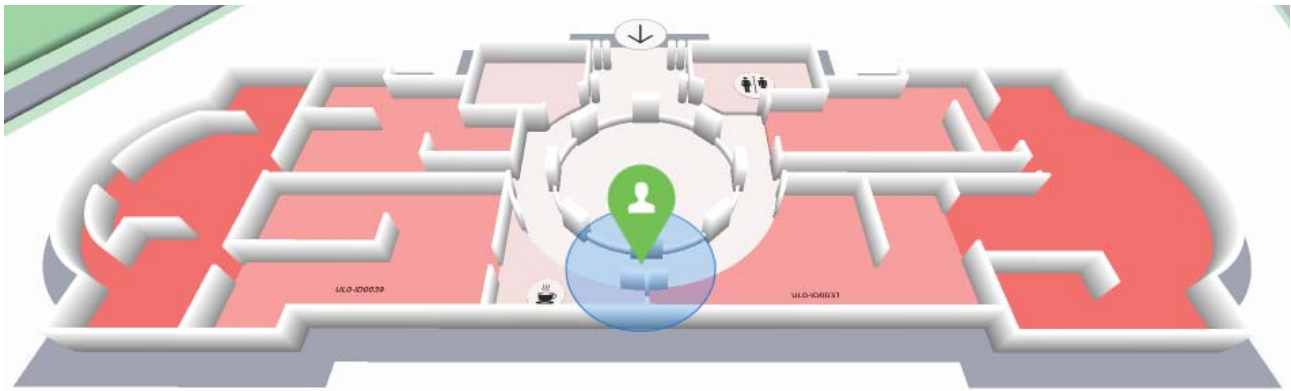


Figura 1.2: Posible ejemplo de un sistema de localización en un centro comercial

No obstante en los últimos años se ha producido un gran auge en el estudio y desarrollo de estos sistemas. Actualmente la tecnología WiFi se impone como la tecnología mas adecuada y utilizada en el desarrollo de sistemas de localización en entornos interiores. Esto es debido principalmente a dos factores:

- **Factores de accesibilidad:** Debido a su fuerte implantación en la sociedad es posible ver redes WiFi en hoteles, museos, aeropuertos, autobuses, centros comerciales y un largo etcétera, además por supuesto de en la práctica totalidad de los hogares. Por otra parte existen multitud de dispositivos que permiten acceder “de serie” a este tipo de redes como son los ordenadores, los teléfonos móviles, videoconsolas, televisores, libros electrónicos, etc. Esto quiere decir que tanto desde el punto de vista del que ofrece el servicio, como desde el punto de vista del que usa el servicio como desde el punto de vista del que desarrolla el servicio la tecnología WiFi se erige como la más accesible de todas con un gran margen de diferencia.
- **Factores económicos:** Por otra parte y como consecuencia del factor anterior su utilización para el desarrollo de sistemas de localización permitiría reutilizar la infraestructura actual y abaratar mucho los costes de desarrollo.

El principal problema de los sistemas de posicionamiento en interiores, como hemos comentado antes, viene principalmente impuesto por las características intrínsecas de los mismos: paredes, muros, mobiliario, personas, etc. Estos obstáculos en el camino de las señales ocasionan degradación en las mismas dando lugar a errores importantes en el cálculo de la posición usando los algoritmos tradicionales.

Los algoritmos denominados de análisis de la escena o fingerprinting están consiguiendo buenos resultados en este campo. Se basan en la intensidad o fuerza de la señal que se recibe en cada punto del entorno. Guardan para cada posición los valores de intensidad que se reciben de los nodos que forman el sistema. De esta forma crean un mapa de radio del entorno. Cuando posteriormente el sistema mida la intensidad recibida en un punto la comparará con el mapa de radio y mediante algún cálculo obtendrá la posición del mapa cuyos valores de intensidad más se parezcan a los medidos.

Viendo las bondades de la tecnología WiFi y los algoritmos fingerprinting la pregunta es si es posible obtener un buen rendimiento combinando ambos en un sistema de localización.

Por ello el presente trabajo fin de carrera aborda el análisis de un sistema de localización en interiores prestando especial interés al rendimiento que puede ofrecer.

Para ello se utilizará la tecnología WiFi y un algoritmo de localización de estilo fingerprinting como son las redes neuronales.

1.1 Objetivos

El objetivo principal de este proyecto es analizar el rendimiento que ofrece un sistema de localización en entornos interiores que use la tecnología WiFi y un

algoritmo de localización de tipo fingerprinting. Diseñaremos uno con el que realizar la evaluación.

El listado de objetivos del proyecto es el siguiente:

- ✓ Realizar un estudio sobre las tecnologías y algoritmos que se utilizan en la actualidad en el desarrollo de estos sistemas. Comentar sus ventajas e inconvenientes y valorar su idoneidad en sistemas de localización.
- ✓ Estudiar el concepto de RSSI y los retos que presenta a la hora de desarrollar un sistema de estas características.
- ✓ Estudiar la arquitectura que ofrece el sistema operativo Windows para el manejo de redes WiFi.
- ✓ Estudiar las APIS que ofrece el sistema operativo Windows para el manejo de redes WiFi y la obtención del valor RSSI.
- ✓ Realizar la implementación de un pequeño módulo sensor capaz de obtener las intensidades de señal de redes WiFi.
- ✓ Estudiar la arquitectura y funcionamiento de las redes neuronales, en especial, del perceptrón multicapa.
- ✓ Diseñar una red neuronal con ayuda del software NeuroSolutions.
- ✓ Probar el sistema desarrollado.
- ✓ Evaluar el rendimiento del sistema.
- ✓ Extraer las conclusiones que se deriven del trabajo realizado.

1.2 Organización de la memoria

La memoria desde este punto se organiza de la siguiente manera:

Capítulo 1

Se expone una introducción al tema.

Capítulo 2

Muestra una revisión del estado actual de los sistemas de localización en interiores.

Capítulo 3

Se aborda una profunda descripción del sistema.

Capítulo 4

Describe el diseño del sistema.

Sistema de posicionamiento en interiores

Capítulo 5

Se muestran los casos de prueba utilizados y los resultados obtenidos.

Capítulo 6

Se realiza una evaluación del sistema.

Capítulo 7

Se exponen las conclusiones obtenidas y las líneas de trabajo futuro.

Capítulo 8

Se describe la bibliografía y los recursos web utilizados.

2. Revisión del estado del arte

2.1 Características

En general podemos asumir que un sistema de localización ya sea para exteriores o para interiores se compone de una serie de dispositivos que emiten una señal al medio (acústica, óptica, radiofrecuencia, etc) y que llamaremos nodos de comunicación y de otra serie de dispositivos que captan esa señal del medio y miden alguna de sus características (tiempo de llegada, energía de la señal, etc) a los que denominaremos dispositivos móviles. A través de esas mediciones y por medio de algún algoritmo (trilateración, triangulación, fingerprinting, etc) establecen su localización.

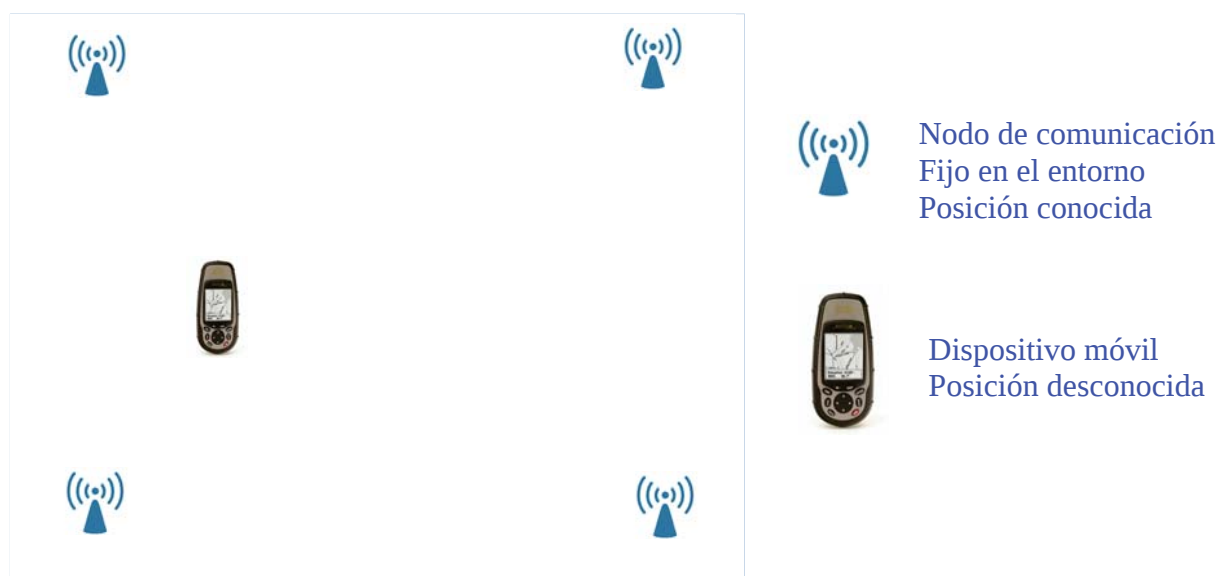


Figura 2.1 Esquema básico de un sistema de localización

En estos sistemas los nodos de comunicación tienen una posición fija y conocida de antemano mientras que los dispositivos móviles como su nombre indica pueden moverse libremente por el entorno y su posición es la que debe ser determinada.

Representación de la localización

La forma de presentar la localización también difiere de unos sistemas a otros. Podemos establecer que existen dos tipos o formas de presentar la localización de un objeto: Localización simbólica o localización real.

Localización simbólica

Como su nombre indica la localización simbólica establece la posición de un objeto en referencia a símbolos.

De esta forma la posición de un objeto puede presentarse como una región del entorno (la biblioteca, el salón, etc) o como un punto concreto del entorno (frente a la nevera, encima del sofá, etc) pero siempre utilizando referencias simbólicas.

En muchas ocasiones debido a limitaciones de cómputo o simplemente por exceder a los objetivos del sistema no es necesario establecer una localización real o exacta ofreciendo en su lugar un posicionamiento simbólico.

Localización real

La localización física o real establece la posición del objeto en base a unas coordenadas. Estas coordenadas pueden hacer referencia a una posición en el plano (coordenadas 2D) o en el espacio (Coordenadas 3d).

La localización usando coordenadas 3D (más costoso de calcular) sólo se realiza en aquellos sistemas en los que sea necesario diferenciar entre diferentes alturas como pudieran ser entornos de varias plantas.

La localización física puede ser absoluta o relativa a algún punto del entorno. El sistema de coordenadas dado por las variables latitud, longitud y altitud ofrece un posicionamiento absoluto y es el utilizado por los sistemas de localización en exteriores.

Sin embargo este esquema de coordenadas no se utiliza en interiores por varias razones. En primer lugar, dos ubicaciones alejadas en un entorno interior varios metros o incluso decenas de metros lo que supone una gran diferencia en este tipo de entornos resultaría en una variación de alrededor 0,0001 grados usando coordenadas de longitud y latitud lo que complica el cálculo de la posición. Además desde el punto de vista semántico estas coordenadas le dicen muy poco al usuario en cuanto a su localización en el interior del entorno.

Los sistemas de posicionamiento en interiores suelen utilizar coordenadas relativas para establecer la posición. Estas coordenadas pueden indicar una posición relativa al entorno o indicar una posición relativa a una región del mismo.

Lugar de cálculo de la localización

Dependiendo de donde se realiza la localización podemos distinguir entre:

Sistema de posicionamiento en interiores

- Autolocalización o posicionamiento local: En este caso el cómputo del posicionamiento es llevado a cabo por el propio dispositivo móvil. Éste recibe la señal, mide las características oportunas, aplica el método que corresponda y establece su propia localización.

Las aplicaciones donde el usuario necesita conocer su ubicación utilizan el posicionamiento local (navegación, demanda de información en base a la localización, etc). El usuario es consciente en todo momento de su posición.

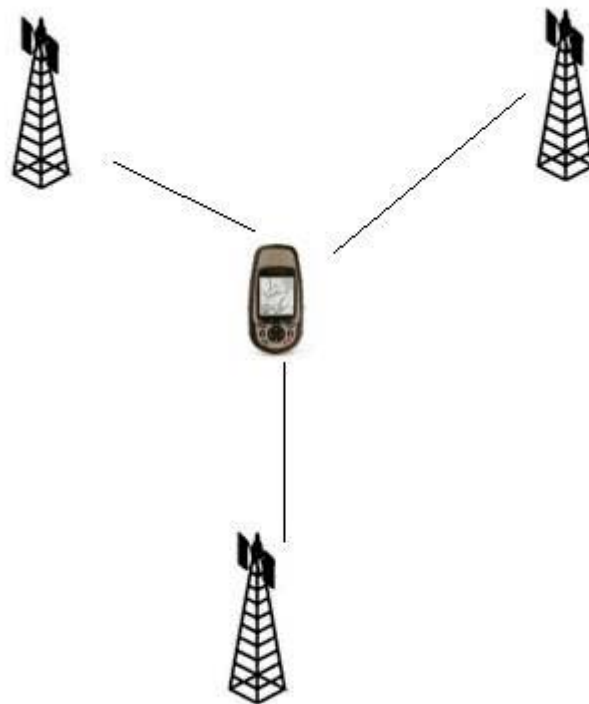


figura 2.2: Autolocalización

- Localización remota: En este caso el cómputo de la localización es realizado fuera del dispositivo móvil. Puede ser llevado a cabo por los nodos del sistema o por un servidor externo.

Normalmente en una localización remota son los nodos del sistema los reciben la señal del medio (en este caso proveniente del dispositivo móvil), miden alguna de sus características y envían las mediciones a un servidor externo que las procesa y obtiene la localización. Sin embargo existen sistemas donde es el dispositivo móvil quien recibe la señal del

Sistema de posicionamiento en interiores

medio y la reenvía a un servidor externo para que realice el cómputo de su localización.

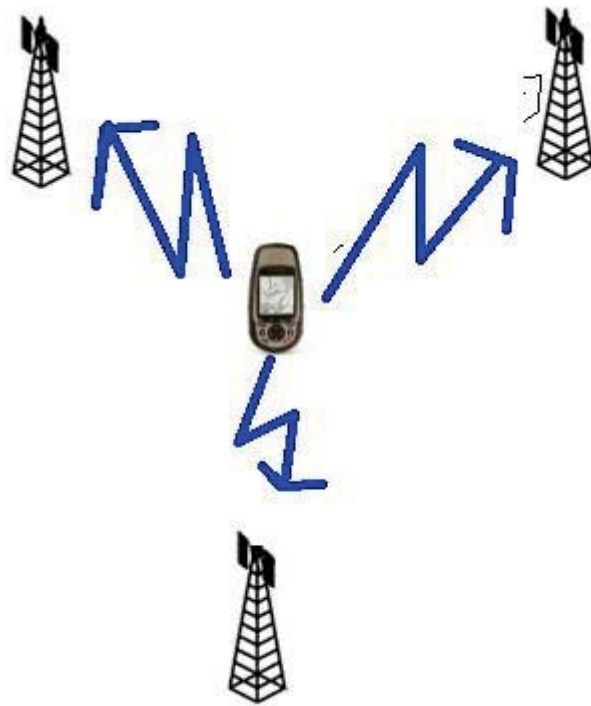


Figura 2.3: Localización remota

La localización remota se utiliza cuando el objetivo del sistema es la monitorización. El objetivo puede ser monitorizar la ubicación de individuos (guarderías, residencias, etc) o utilizar esta información para otros servicios (estudios de compra en supermercados, estudio de salas mas visitadas de un museo, etc). En general, el usuario objeto del posicionamiento no está interesado en conocer su ubicación, no obstante es conveniente que sea consciente de que está siendo monitorizado.

Arquitectura

Los sistemas de localización presentan en líneas generales la siguiente arquitectura:

- Una red de nodos distribuidos por el entorno. Su posición es fija y conocida de antemano. En general cuantos más nodos existan mayor será la precisión del sistema, no obstante demasiados nodos podrían

saturar la red de comunicaciones. Pueden estar sincronizados si el algoritmo de posicionamiento lo requiere. Pueden ser los encargados de enviar la señal al medio o de recibirla de éste. Pueden apoyarse en un servidor externo para calcular la localización en una configuración remota.

- Uno o varios dispositivos móviles que pueden moverse libremente por el entorno. Su posición es desconocida y utilizan el sistema para obtenerla. Si son los encargados de enviar la señal al medio (localización remota) la privacidad de los usuarios puede verse comprometida. Un exceso de dispositivos móviles en el entorno puede ocasionar excesiva latencia en las localizaciones y sobrecargar la red de comunicaciones.
- Un medio de transmisión. Pueden ser señales ópticas, acústicas, radiofrecuencia, magnéticas, etc.
- Un servidor externo (opcional) que se encargue de dar apoyo de cómputo al sistema. Normalmente utilizado en sistemas de localización remota, también puede ser interesante su uso cuando se requiere de mucha potencia de cálculo en la localización.

2.2 Tecnologías subyacentes

El gran auge que los sistemas de posicionamiento en interiores está teniendo en los últimos años ha llevado a la investigación de un gran número de tecnologías en su desarrollo. Cada tecnología presenta sus propias ventajas e inconvenientes y su elección depende del proyecto concreto que se aborde.

Podemos asumir que al hablar de tecnologías subyacentes nos estamos refiriendo principalmente al medio de comunicación o tipo de señal utilizado (luz, ondas de radio, etc) y al hardware que lo soporta (fotodiodos, antenas de radio, etc).

Las tecnologías más utilizadas en la actualidad son las siguientes:

Luz

Los sistemas ópticos de localización se caracterizan principalmente por el uso de señales electromagnéticas cuya longitud de onda se encuentra dentro del espectro de la luz. Los más utilizados se basan en la luz infrarroja y en la luz visible.

Generalmente suelen funcionar modulando las señales en forma de pulsos codificados. Cada nodo de comunicación envía su propio código, el receptor al

Sistema de posicionamiento en interiores

recibirlo sabe que nodo se lo ha enviado y puede resolver la posición en la que se encuentra.

En el caso de los sistemas basados en la luz visible los nodos de comunicación pueden ser cualquier tipo de bombilla aunque lo más común es utilizar bombillas led. Estas bombillas se encuentran adaptadas para poder emitir parpadeos que son imperceptibles para el ojo humano pero que llevan consigo un código que identifica a la bombilla que lo ha emitido. Los dispositivos móviles suelen llevar fotodiodos o sensores de imagen (cámaras) para la recepción.

Los sistemas basados en esta tecnología pueden reutilizar el alumbrado actual adaptándolo como se ha comentado anteriormente, permitiendo la reducción de costes. Además el hecho de poder utilizar una cámara (presente en la práctica totalidad de teléfonos móviles) para utilizarlo hace posible que el acceso a estos sistemas sea muy fácil y sin la necesidad de adquirir hardware adicional o específico.

Los sistemas basados en luz infrarroja utilizan diodos de emisión de luz infrarroja para la transmisión y fotodiodos para la recepción de los pulsos de luz.

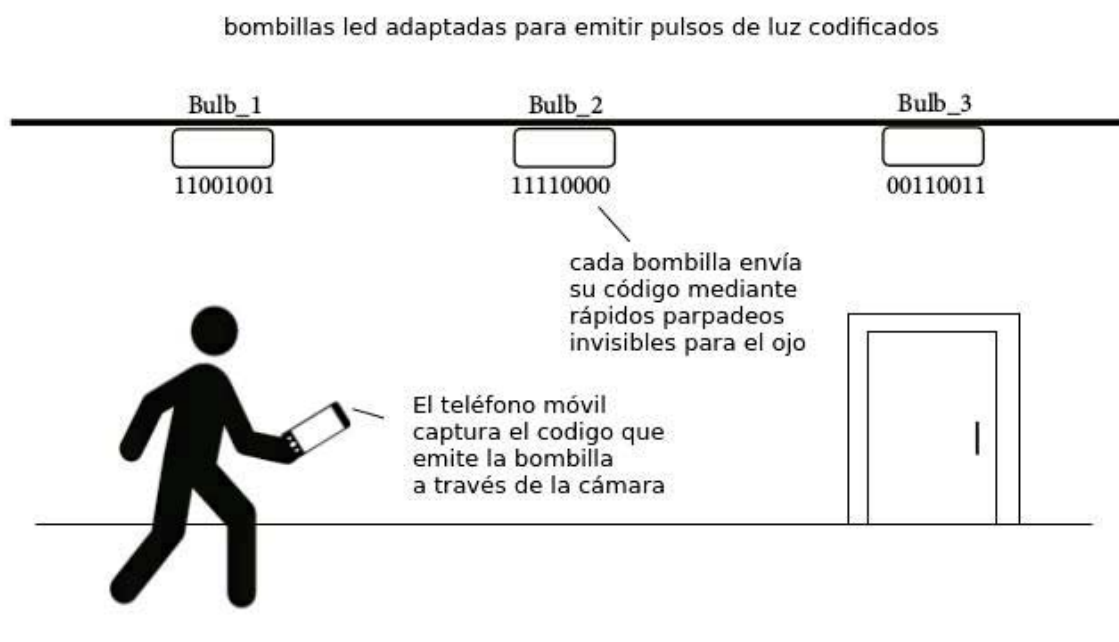


Figura 2.4: Sistema de localización mediante luz visible

Sistema de posicionamiento en interiores

La principal desventaja de los sistemas ópticos es que necesitan línea de visión directa entre el emisor y el receptor y en entornos interiores esto no siempre es fácil de conseguir. Además, en general necesitan de una gran número de nodos de comunicación para conseguir gran precisión.

Sonido

Los sistemas sonoros utilizados en localización se caracterizan por el uso de ondas sonoras en las comunicaciones.

Suele utilizarse tanto señales de ultrasonido (frecuencias por encima de los 20000 Hz y por tanto inaudibles para el ser humano) como señales sonoras audibles (rango de frecuencias comprendido entre los 20 Hz y los 20000 Hz).

Generalmente basan su algoritmo de localización en el tiempo que tarda la señal en su viaje desde el emisor al receptor y a partir de este valor poder calcular distancias.

Se utilizan altavoces para la transmisión y micrófonos para la recepción.

Una de las principales ventajas del uso de esta tecnología es que permite que con un teléfono móvil (la practica totalidad de estos dispositivos dispone de un micrófono y un altavoz) sea posible utilizar el sistema sin tener que adquirir hardware específico o adicional. Además cualquier sistema de sonido puede ser utilizado como sistema altavoz con lo que sería posible reutilizar la infraestructura actual abaratando costes (por ejemplo en centros comerciales).

Uno de los inconvenientes que presentan estos sistemas es la presencia de sonidos propios de la aplicación que pudieran ser molestos para los usuarios. Con los sistemas basados en ultrasonidos se resuelve este problema debido a la imposibilidad de poder captar los sonidos que emiten. Si se trata de sistemas de sonido audible una posible solución es embeber los sonidos del sistema en un hilo musical.

Una solución empleada consiste en utilizar un conjunto de altavoces que emiten el mismo hilo musical pero cada uno tiene asociada una secuencia propia pseudo aleatoria que modula en amplitud algunas de las frecuencias que componen la musica. Los usuarios reciben este hilo musical a través de un micrófono y mediante un algoritmo extrae la secuencia de cada hilo identificando de que altavoz proviene. Posteriormente calcula el retardo de cada altavoz para obtener la distancia del dispositivo a cada uno de los altavoces.

Radiofrecuencia

Los sistemas de localización basados en esta tecnología usan las señales de radiofrecuencia en sus comunicaciones (frecuencias comprendidas entre los 3Khz y los 300 Ghz).

Sistema de posicionamiento en interiores

En la actualidad existen gran cantidad de tecnologías que hacen uso de la radiofrecuencia. Estas tecnologías ofrecen gran cantidad de servicios que ya se encuentran implementados en multitud de dispositivos y que pueden ser reutilizados en sistemas de posicionamiento. Esta es la razón principal por la que la mayoría de los sistemas de localización hacen uso de la radiofrecuencia.

Las tecnologías mas usadas son las siguientes:

IEEE 802.11 (Wi-Fi)

Este estándar surgió para implementar redes de comunicaciones inalámbricas. Utiliza señales de radiofrecuencia en las bandas de 2,4ghz y 5ghz. Se utilizan tarjetas de red 802.11 para la transmisión y recepción de las señales de radiofrecuencia.

La configuración más utilizada consiste en usar puntos de acceso WiFi como nodos de comunicaciones y terminales WiFi (smartphones, laptops, netbooks, etc) como dispositivos móviles.

La tecnología Wí-Fi es una tecnología fuertemente implantada en todo el mundo. Proporciona una especificación de los niveles físico y de enlace incluyendo un protocolo de acceso al medio.

Esto significa que aquellos dispositivos que integren una tarjeta 802.11 tendrán implementados “de serie” todos los servicios que ofrece esta tecnología además de un protocolo de acceso al medio lo que permite a los desarrolladores centrarse en el algoritmo de localización y despreocuparse en gran medida de temas como las posibles colisiones en las transmisiones.

Actualmente esta tecnología se encuentra tan difundida que existe cobertura WiFi prácticamente en cualquier lugar por lo que ofrecer servicios de localización sería posible con reutilizar la infraestructura actual en muchas de las ocasiones.

Otra de las grandes ventajas del uso de esta tecnología en sistemas de localización derivada de la fuerte implantación que presenta es la cantidad de dispositivos que llevan incorporada una tarjeta 802.11 (móviles, cámaras, videoconsolas, impresoras, laptops, etc). Esto posibilita que el acceso al uso de estos sistemas sea increíblemente fácil.

Bluetooth

Originalmente ideado para compartir datos entre dispositivos sin la necesidad de cables. Utiliza señales de radiofrecuencia en la banda 2,4ghz. Bluetooth es una tecnología de comunicaciones inalámbricas basadas en el estándar 802.15.1. Se emplean tarjetas de red bluetooth para la transmisión y recepción de señales.

Su configuración más habitual en sistemas de localización es la siguiente: se utilizan unos dispositivos llamados beacons como nodos de comunicaciones y terminales bluetooth (smartphones, laptops, netbooks) como dispositivos móviles.

Al igual que WiFi ofrece la especificación de la capa de nivel físico y nivel de enlace. A partir de la especificación BLE (bluetooth de bajo consumo) su uso en sistemas de localización se ha vuelto más interesante debido sobre todo a su alta disponibilidad (prácticamente todos los móviles modernos disponen de bluetooth), su bajo coste y por supuesto su bajo consumo energético.

Un inconveniente en el uso de esta tecnología es la necesidad de adquirir beacons para su infraestructura lo que eleva su coste. Además

Al utilizar la misma banda de frecuencias que Wi-Fi pueden interferirse resultando en colisiones y problemas de recepción de tramas.

Zigbee

Zigbee es un estándar de comunicaciones desarrollado por la Zigbee Alliance. Está pensado para redes inalámbricas que necesiten baja transferencia de datos y bajo consumo de energía. Además los dispositivos que usan esta tecnología tienen un coste de fabricación muy reducido. Esta tecnología se usa principalmente en productos de domótica.

Adopta el estandar ieee 802.15.4 como especificación para sus niveles físico y de enlace. Incorpora CSMA/CA como protocolo de control de acceso al medio.

En un sistema de posicionamiento la configuración es la siguiente: una serie de nodos zigbee esclavos que actúan como nodos de comunicaciones y un nodo maestro que actúa como dispositivo móvil.

El uso de esta tecnología presenta como ventajas su bajo consumo y bajo coste. Además con cada paquete de datos que se envía se adjunta el indicador de fuerza de la señal. Esto facilita mucho la labor de desarrollo.

Como desventaja mencionar que es necesario adquirir tanto la infraestructura como un hardware específico para usar este tipo de sistemas.

2.3 Algoritmos de localización

El algoritmo de localización se encarga de obtener la posición de un dispositivo en base a alguna característica de la señal recibida. Actualmente existen varios métodos de estimación utilizados para localización en interiores, cada uno con sus ventajas e inconvenientes que difieren principalmente en la característica observada de la señal. Aquí presentamos los más utilizados.

2.3.1 Trilateración

El método de trilateración obtiene la posición del dispositivo móvil en base a la distancia que existe entre éste y los nodos de comunicación.

Este método necesita conocer la distancia de al menos tres nodos con respecto al dispositivo móvil para poder ubicarlo con precisión en un entorno de dos dimensiones y de 4 nodos para ubicarlo en tres dimensiones.

La distancia del nodo al dispositivo móvil se puede representar como una circunferencia en cuyo centro se ubica el nodo y cuyo radio es la distancia existente entre ellos.

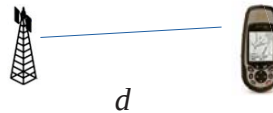


Figura 2.5 Distancia del nodo al dispositivo móvil

Si representamos de esta manera la distancia de al menos tres nodos respecto al dispositivo móvil tendríamos una imagen como la que sigue.

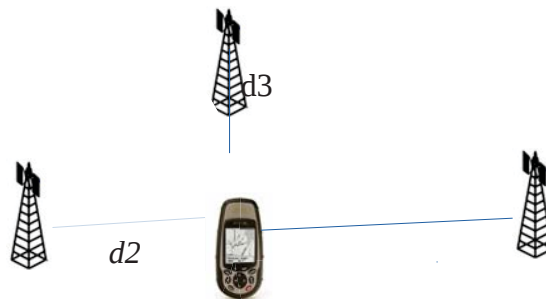


Figura 2.6: Método de trilateración

En esta imagen se aprecia que la intersección de las tres circunferencias identifica el punto en el que de forma unívoca se encuentra el dispositivo. Analíticamente, la resolución de este sistema de ecuaciones (teniendo en cuenta que la posición de los nodos es conocida) daría con la posición del dispositivo.

Este método necesita conocer las distancias a los nodos para poder funcionar.

En general existen tres técnicas para obtener la distancia:

1. A partir del tiempo que tarda la señal.
2. Mediante modelos de propagación de la señal

Los métodos más utilizados son los siguientes:

Tiempo de Llegada

La técnica de tiempo de llegada (también llamada tiempo de vuelo) se basa en conocer el tiempo que tarda la señal en recorrer la distancia existente entre el emisor y el receptor. Sabiendo esto y la velocidad de propagación de la señal se puede deducir la distancia “d” entre ambos a través de la conocida fórmula:

$$\text{distancia} = \text{velocidad} \times \text{tiempo}$$

En este momento podemos situar el dispositivo móvil en cualquier posición a una distancia “d” con respecto al dispositivo emisor. Si obtenemos el tiempo de llegada de tres o más nodos de comunicación obtendremos las distancias de éstos y mediante el método de trilateración hallaremos la ubicación del dispositivo móvil.

Para calcular el tiempo que tardó la señal en llegar al dispositivo móvil éste debe conocer el instante de tiempo en el que se envió la señal y ambos deben estar sincronizados. Por ello el emisor envía en cada señal una marca de tiempo que indica el instante de envío.

En aquellos sistemas que utilizan señales electromagnéticas (que son los mayoritarios) hay que tener en cuenta que las velocidades de propagación de éstas son muy altas (viajan a la velocidad de la luz $C=300000000\text{m/s}$) por lo que son necesarios relojes de una grandísima precisión y resolución. Tengamos en cuenta que un error de medida de 0,00000001 segundos en una señal electromagnética supone 3 metros de desviación con respecto a la distancia real, cifra que en entornos interiores y dependiendo el sistema puede ser inaceptable. Si se utiliza el sonido como señal transmisora al tener una velocidad de propagación mucho menor (unos 300 m/s) los requisitos son mucho más permisivos en este sentido.

El hecho de necesitar relojes de gran precisión y resolución además de sincronismo entre nodos de comunicación y el dispositivo móvil obliga a utilizar un hardware específico (y posiblemente caro) que dificulta el acceso a la utilización del sistema.

Tiempo de ida y vuelta

Tiempo de ida y vuelta es una técnica que se basa en conocer el tiempo que tarda la señal en ir desde el dispositivo móvil al nodo de comunicación y volver.

El dispositivo móvil envía la señal al medio y el nodo de comunicación al recibirla la vuelve a enviar al dispositivo móvil. Una vez de vuelta éste calcula el tiempo que tardó en ir y volver y sabiendo la velocidad de propagación de la señal puede calcular la distancia que existe entre el emisor y el receptor mediante la fórmula:

$$\text{distancia} = (\text{velocidad} \times \text{tiempo})/2$$

Si se obtiene el tiempo de ida y vuelta de al menos tres nodos de la red obtendremos sus distancias y mediante trilateración podremos obtener la ubicación del dispositivo móvil.

Sistema de posicionamiento en interiores

Esta técnica sigue precisando de un reloj en el dispositivo móvil para poder calcular el tiempo de ida y vuelta. Además, si usa señales electromagnéticas éste debe ser de gran precisión y resolución debido a las altas velocidades de propagación que presentan este tipo de señales. Además es necesario algún tipo de mecanismo que calcule los tiempos de procesamiento de los nodos (desde que reciben la señal hasta que la envían) y restárselo al tiempo de ida y vuelta calculado.

Al igual que con la técnica de tiempo de llegada la necesidad de un reloj de gran precisión en el dispositivo móvil obliga al usuario a adquirir un hardware específico para utilizar el sistema.

Intensidad de la señal recibida

Este método se basa en conocer la intensidad de la señal recibida, que en esencia es la energía que tiene la señal en el momento de su recepción y suele venir determinada en dBm (decibelios/ miliWatio).

Teniendo en cuenta que la potencia de la señal disminuye en función de la distancia recorrida podemos determinar la distancia entre el nodo de comunicación y el dispositivo móvil sabiendo la potencia de señal emitida y la potencia de señal recibida (entre otros parámetros).

Según el modelo de propagación de Friis la potencia recibida en el espacio libre viene determinada por la ecuación:

$$Pr(d) = P_t * G_t * G_r * \lambda^2 / (4\pi)^2 * d^2 * L$$

Pr (d) – potencia recibida en función de la distancia entre transmisor y receptor

Pt – potencia transmitida

Gt – ganancia de antena transmisora

Gr – ganancia de la antena receptora

λ – longitud de onda en metros

d – la distancia de separación de T-R en metros

L – pérdidas del sistema no achacables a la propagación

De esta forma obtenemos la distancia entre el nodo de comunicación y el dispositivo móvil. Obteniendo la potencia de señal recibida de tres o más nodos de comunicación podemos calcular sus distancias y mediante trilateración podemos ubicar el dispositivo móvil.

Este método necesita de algún mecanismo para que el dispositivo móvil pueda medir la potencia de señal que recibe de los nodos de comunicación. Tecnologías como WiFi, Bluetooth o Zigbee permiten obtener este indicador de forma sencilla.

2.3.2 Multilateración

La multilateración es otro método utilizado para obtener la posición del dispositivo móvil a partir de la diferencia de distancias que existe entre el mismo y al menos dos nodos de comunicación.

Si la distancia que existe entre el dispositivo móvil y un nodo de comunicación es d_1 y la que existe con respecto a otro nodo de comunicación es d_2 podemos deducir que el dispositivo se encuentra en algún punto de la hipérbola cuyos puntos cumplen la misma diferencia de distancias $|d_1 - d_2|$.

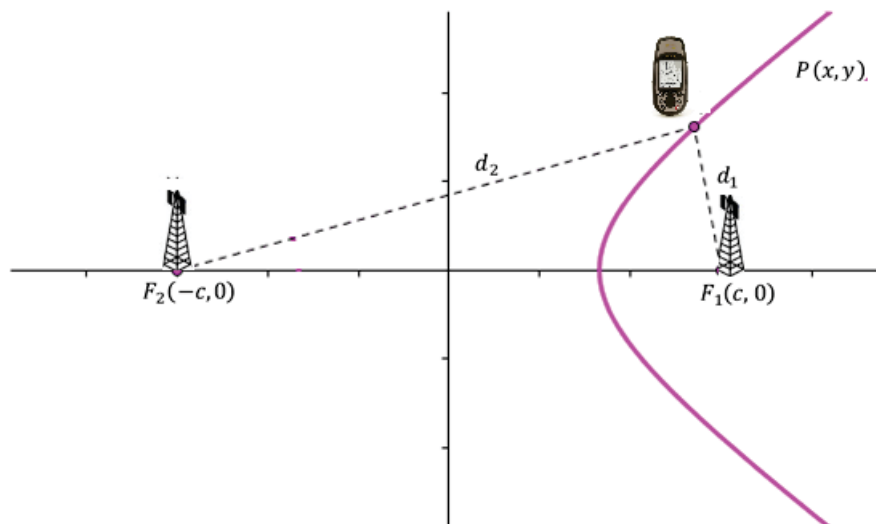


Figura 2.7: $H = \{P(x, y) / |d(P; F_1) - d(P; F_2)| = |d_1 - d_2|\}$

Con un tercer nodo de comunicación podemos obtener otra diferencia de distancias $|d_1 - d_3|$ y por lo tanto otra curva hipérbola donde podría estar el

Sistema de posicionamiento en interiores

dispositivo. El punto de intersección de ambas hipérbolas corresponde al punto en el plano donde se ubica el dispositivo móvil.



Figura 2.8 Método de multilateración

Para hallar las distancias a los nodos de comunicaciones pueden usarse los métodos vistos anteriormente como son el tiempo de llegada, el tiempo de ida y vuelta o el método basado en el cálculo del RSSI y posteriormente calcular las diferencias entre pares de nodos.

Existe otra técnica que directamente te calcula las diferencias de distancias entre el dispositivo móvil y cada par de nodos y recibe el nombre de diferencia de tiempo de llegada.

Diferencia de tiempos de llegada

La técnica de diferencia de tiempo de llegada se basa en conocer el tiempo que tarda la señal en llegar a un nodo de comunicación en relación a otro nodo de comunicación, es decir, si la señal llega al “nodo1” en el instante de tiempo 15 y esa misma señal llega al “nodo2” en el instante de tiempo 18, según esta técnica la diferencia de tiempo de llegada es 3.

Conociendo la diferencia de tiempo de llegada entre dos nodos de comunicación podemos establecer la distancia que existe entre ellos con respecto al dispositivo móvil ($|d1 - d2|$). Obteniendo la diferencia de tiempo de

llegada entre otro par de nodos ($|d_1-d_3|$) podemos establecer mediante multilateración la posición del dispositivo móvil.

Esta técnica sólo es realizable usando una configuración de localización remota. Necesita sincronización entre los receptores y relojes de gran precisión y resolución. Esto puede encarecer la infraestructura necesaria y supone una desventaja en el uso de este tipo de sistemas.

2.3.3 Triangulación

El método de triangulación obtiene la posición del dispositivo móvil a partir del ángulo de incidencia de la señal.

Teniendo en cuenta que la posición de los nodos es conocida sabemos la longitud del segmento que los une. Obteniendo el ángulo de llegada de la señal de al menos dos nodos podemos determinar un triángulo del que conocemos todos sus ángulos.

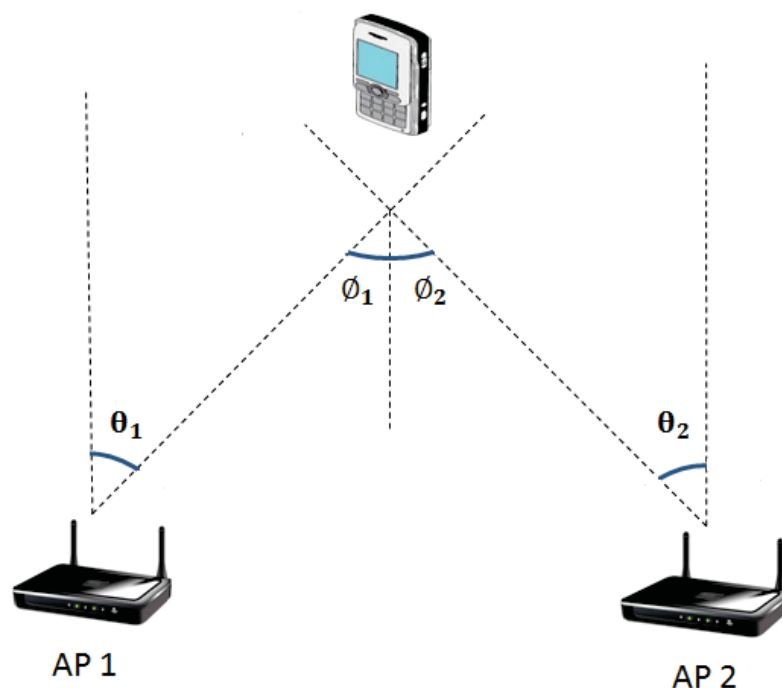


Figura 2.9: Método de triangulación

De esta forma podemos obtener la posición del dispositivo móvil como el punto de intersección de los segmentos.

Este método necesita de hardware que pueda calcular el ángulo de incidencia de la señal que encarece el coste en infraestructura.

2.3.4 Proximidad

El método de proximidad obtiene la posición del dispositivo móvil a partir de la distancia al nodo más cercano.

Una vez conocemos la identidad de ese nodo el método establece la posición del mismo como la posición del dispositivo móvil.

Si la localización utilizada es simbólica, el sistema ubicará al dispositivo móvil en la región a la que pertenece el nodo más cercano, si por el contrario el sistema utiliza localización física entonces lo ubicará en las coordenadas donde se encuentra el nodo.

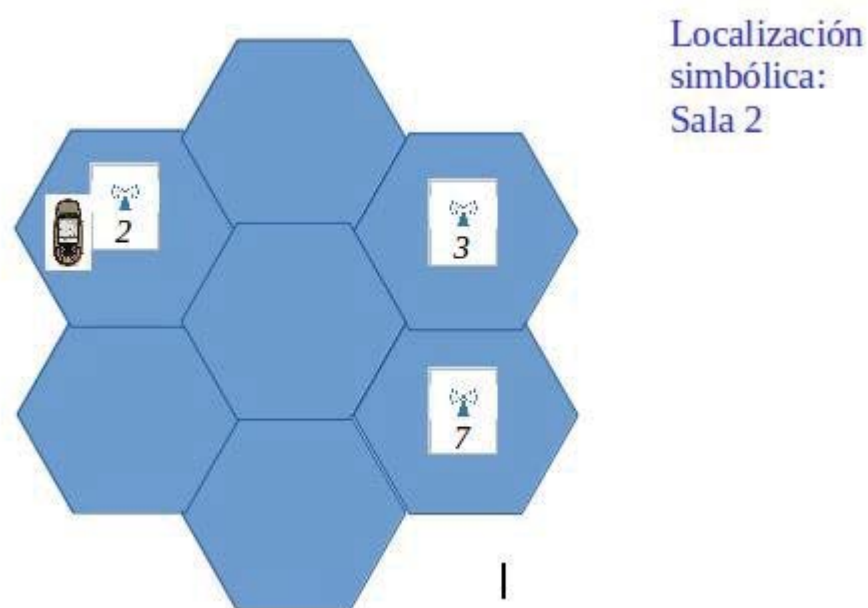


Figura 2.10: Método de proximidad para localización simbólica

El método necesita conocer cuál es el nodo más cercano al dispositivo móvil. Como hemos visto hasta ahora existen unas cuantas técnicas para calcular distancias y poder determinar cuál es el nodo más cercano (tiempo de llegada, tiempo de ida y vuelta, intensidad de la señal recibida, etc), sin embargo este método no suele utilizar las distancias para su funcionamiento, sino intensidades.

El método de proximidad utiliza la intensidad de señal recibida para estimar el nodo más cercano basándose en la propia fuerza de la señal, es decir, el nodo cuya señal se reciba con más fuerza será el nodo estimado como el más cercano.

Entre las virtudes de este método está su fácil implementación y su baja complejidad reduciendo los requisitos de cómputo del sistema. Admite tanto configuración remota como local.

El principal inconveniente de este método es la necesidad de desplegar un gran número de nodos para obtener una buena precisión, sin embargo para aquellos sistemas cuya localización no necesite de una gran precisión o precisen de una localización simbólica puede resultar una buena opción.

Consideraciones sobre los algoritmos presentados

El problema de los algoritmos basados en el cálculo de distancias y ángulos es que necesitan línea de visión directa entre emisor y receptor. En entornos interiores la presencia de obstáculos como muros, paredes, mobiliario e incluso personas dificulta que haya esa línea de visión directa entre emisor y receptor.

La señal al chocar con un obstáculo cambia sus condiciones de velocidad y energía por lo que los algoritmos basados en tiempo o intensidad no son válidos.

Otros problema que plantean los entornos interiores tiene que ver con el efecto multicamino. Este consiste en lo siguiente: Cuando la señal choca con un obstáculo ésta puede reflejarse, difractarse o dispersarse de forma que varían sus condiciones de velocidad y dirección. Esto puede ocasionar que lleguen al receptor varios haces de la misma señal pero por diferente camino y en diferentes tiempos. Al llegar al receptor estos haces se suman y ocasionan distorsión en la señal lo que también plantea un problema en aquellos algoritmos que usen la intensidad de la señal como referencia.

2.3.5 Método análisis de la escena o fingerprinting

El método de análisis de la escena (también conocido como método fingerprinting) consiste en hacer corresponder el valor de alguna característica de la señal con una posición dentro del entorno.

La idea general es crear un mapa del entorno donde se asocia a cada posición del mismo (o por lo menos a un gran número de posiciones) un valor correspondiente a alguna característica de la señal medida en esa posición. Esa característica debe depender de la posición ocupada, generalmente suele usarse la potencia de señal recibida.

Sistema de posicionamiento en interiores

Una vez creado el mapa del entorno (mapa de radio si se utilizan señales de radio) el sistema está listo para ser usado en tiempo real. Al medir desde una posición concreta la característica de la señal se compara con las medidas almacenadas en el mapa y se escoge aquella posición cuyo valor almacenado se asemeje más al valor medido.

Este método no necesita línea de visión directa entre emisor y receptor, tampoco necesita de hardware adicional o sincronización, además tiene en cuenta el ruido producido por el entorno así como el efecto multicamino. Esto lo convierte actualmente en el método más indicado para el desarrollo de sistemas de localización en entornos interiores.



Consta de dos fases:

- Una primera fase de calibración donde se crea el mapa con los valores discriminantes de la señal para todas las posiciones del entorno.
- Una segunda fase de ejecución donde en tiempo real se realiza la medición y se obtiene la posición cuyo valor almacenado sea el más cercano al medido.

Fase de calibración

Durante esta fase se crea el mapa del entorno con los valores de la señal para las diferentes posiciones consideradas en el mismo.

Es necesario decidir que puntos del entorno se van a considerar. En principio cuantos más se consideren más precisión tendrá el sistema pero más recursos de memoria se consumirán. Normalmente suele dividirse el entorno en rejillas y utilizarse coordenadas cartesianas para identificar las posiciones.

							
(0,1)							
(0,0)	(1,0)						

También es necesario establecer que característica de la señal se va a tener en cuenta y cuantas señales se van a recibir. En principio cuantas más señales se tengan en cuenta mas preciso será el sistema pero también más recursos de cómputo y tiempo serán necesarios. Con estos datos es necesario asignar a cada posición un vector cuyos componentes son los valores de las diferentes señales y la identificación de esa posición (coordenadas cartesianas, número de celda, etc.).

Finalmente es necesario rellenar las distintas posiciones con los valores requeridos. Existen dos formas de hacer esto: Mediante inspección del lugar o mediante modelos de propagación.

Inspección del lugar

La inspección de lugar consiste en recorrer una a una las posiciones que se han considerado en el entorno y realizar las mediciones anotando en una tabla la posición donde se realiza la medida y los valores obtenidos.

Modelos de propagación

Los modelos de propagación permiten mediante el ajuste de ciertos parámetros simular el valor que tiene la señal en una determinada posición.

Fase de ejecución

La fase de ejecución se corresponde con el uso del sistema en casos reales.

El modo de funcionamiento es el siguiente:

Sistema de posicionamiento en interiores

- El dispositivo móvil obtiene los valores de señal requeridos.
- Consulta el mapa del entorno y mediante alguna técnica escoge aquél punto cuyos valores de señal sean los más cercanos. Este punto será el resultado o posición que devolverá el sistema.

Técnica KNN

Se utiliza la distancia euclídea entre el vector de medidas obtenido y los vectores almacenados en el mapa del entorno. Los k vectores con menor distancia son seleccionados.

Posteriormente se devuelve la posición promedio de los k vecinos seleccionados.

El parámetro k se ajusta dependiendo de las características del problema. En general un valor de k grande obliga a más recursos de cómputo y memoria no siempre obteniendo mejor precisión. Es necesario probar varias configuraciones hasta obtener la deseada.

Redes neuronales

Las redes neuronales son algoritmos capaces de aprender a través de ejemplos. Pueden ser utilizadas en problemas de clasificación como en problemas de aproximación funcional.

Necesitan de una fase de entrenamiento para adquirir el conocimiento requerido.

Cuando actúan como algoritmos fingerprinting los vectores del mapa del entorno actúan como los patrones de entrenamiento de la red neuronal. Los valores de señal se corresponden con las entradas de la red y los valores de posición con las salidas.

Son algoritmos muy rápidos en su ejecución y ofrecen muy buenos resultados en problemas donde la solución no está completamente definida o es difícil de obtener.

Métodos probabilísticos

Los métodos probabilísticos infieren la posición en base a una medida de probabilidad. Al principio todas las posiciones son igual de probables pero a medida que se van realizando localizaciones las probabilidades van cambiando.

3. Descripción del sistema

3.1 Resumen del funcionamiento

Se trata de un sistema capaz de obtener la posición que ocupa un dispositivo móvil dentro de una sala a partir de los valores de intensidad que recibe de 4 redes WiFi.

Se basa en la relación que existe entre el valor de intensidad de una señal y la distancia recorrida por la misma. En el espacio libre esta relación es bien conocida y da buenos resultados. Sin embargo, en entornos interiores la presencia de obstáculos entre emisor y receptor provoca la pérdida de parte de la energía de la señal al atravesarlos por lo que la regla mencionada no tiene validez. El problema sigue siendo modelizable debido a que la energía perdida depende del material del obstáculo, pero es difícil obtener estos modelos.

Las redes neuronales ofrecen buenos resultados ante problemas cuya solución no está completamente definida o es difícil de obtener como es el que nos ocupa por lo que se ha decidido utilizar una red neuronal como algoritmo de localización.

Se ha utilizado la tecnología y servicios que ofrecen las redes WiFi (ieee 802.11) para la infraestructura del sistema. Para medir valores de intensidad se ha utilizado el indicador proporcionado por WiFi (RSSI).

El sistema puede dividirse en dos módulos:

- Un módulo sensor cuya función es obtener el valor RSSI que se recibe de las redes WiFi y realizar un pequeño procesamiento para adaptar los datos a la entrada del módulo localizador. El desarrollo de este módulo se ha llevado a cabo usando el lenguaje c++ y la API Native Wifi propia del sistema operativo Windows.
- Un módulo localizador cuya función es obtener la posición del dispositivo móvil a partir de los valores RSSI devueltos por el módulo sensor. Para el desarrollo de este módulo se ha utilizado un software de terceros específico para creación de redes neuronales.

Módulo sensor

El módulo sensor utiliza el mecanismo de escaneo para obtener los valores RSSI de las redes WiFi.

El escaneo de redes es una técnica creada con la idea de poder obtener las redes WiFi que se encuentran accesibles y luego poder conectarse a alguna de ellas, generalmente a la que presenta una intensidad de señal más fuerte.

Mediante esta técnica el módulo sensor descubre todas las redes WiFi accesibles. Posteriormente realiza un filtrado para obtener las que pertenecen al sistema y finalmente obtiene el valor RSSI de cada red, le aplica un pequeño procesamiento para adaptarlo al módulo localizador y devuelve el resultado.

Para ello hemos utilizado las primitivas `wlanScan` y `wlanGetNetworkBssList` pertenecientes a la API Native Wifi. La primera realiza el escaneo y obtiene las redes accesibles, la segunda recupera esas redes y permite la consulta posterior del valor RSSI.

Módulo localizador

El módulo localizador se encarga de obtener la posición del dispositivo móvil a partir de los valores RSSI devueltos por el módulo sensor. Para ello hemos utilizado una red neuronal de tipo perceptrón multicapa.

La red consta de una capa de entrada, una capa oculta y una capa de salida. La capa de entrada consta de 4 neuronas que reciben los valores RSSI y la capa de salida consta de dos neuronas que se corresponden con las coordenadas de la posición que ocupa en el entorno el dispositivo móvil.

La red ha sido entrenada con el algoritmo de retropropagación y se ha utilizado la técnica de validación cruzada para conseguir que la red sea capaz de generalizar el conocimiento introducido en los cerca de 600 patrones de entrenamiento utilizados.

La posición del dispositivo se representa mediante coordenadas cartesianas. Estas coordenadas son relativas a la sala donde se va a utilizar el sistema.

3.2 Entorno y material utilizados

Para la realización del sistema hemos utilizado 4 puntos de acceso que actuarán como nodos de comunicación y un netbook que actuará como estación móvil. La descripción detallada del material utilizado es la siguiente:

Estación móvil

Como estación móvil hemos elegido un netbook cuyas características se detallan a continuación.

Sistema de posicionamiento en interiores

Sistema operativo	Windows 7
CPU	Intel core atom N550 x 2
Memoria	1GB
Almacenamiento	200 GB
Tarjeta inalámbrica	Qualcom atheros 802.11 b/g/n

Puntos de acceso

Como puntos de acceso hemos utilizado los siguientes:

Huawei E5172 4G Router

Tipo de antena	Omnidireccional desmontable
Estandares inalámbricos	IEEE 802.11 b/g/n
Frecuencia	2.401-2.483 GHz
Potencia de transmisión	17 dbm
Sensibilidad de recepción	802.11n: -64 dBm@65 Mbit/s 802.11g: -65 dBm@54 Mbit/s 802.11b: -76 dBm@11 Mbit/s
Modos inalámbricos	Infraestructura / Ad-Hoc

Tp-link TL-WR940N (X2)

Tipo de antena	3 antenas omnidireccionales
Estandares inalámbricos	IEEE 802.11n, IEEE 802.11g, IEEE 802.11b
Frecuencia	2.4-2.4835GHz
Potencia de transmisión	CE:<20dBm(2.4GHz) FCC:<30dBm
Sensibilidad de recepción	270M: -68dBm@10% PER 130M: -68dBm@10% PER 108M: -68dBm@10% PER 54M: -68dBm@10% PER 11M: -85dBm@8% PER 6M: -88dBm@10% PER 1M: -90dBm@8% PER
Modos inalámbricos	Infraestructura / Ad-Hoc

Tp-link archer mr200

Tipo de antena	2 antenas omnidireccionales
Estandares inalámbricos	IEEE 802.11ac/n/a 5GHz, IEEE 802.11b/g/n 2.4GHz
Frecuencia	2.4GHz y 5GHz
Potencia de transmisión	CE: <20dBm(2.4GHz), <23dBm(5GHz)
Sensibilidad de recepción	5GHz: 11a 54M: -74dBm 11ac HT20: -67dBm 11ac HT40: -64dBm 11ac HT80 : -60dBm 11n HT20: -71dBm 11n HT40: -70dBm 2.4G: 11g 54M: -76dBm 11n HT20: -73dBm 11n HT40: -67dBm
Modos inalámbricos	Infraestructura / Ad-Hoc

Entorno

El entorno utilizado para el desarrollo del sistema ha sido una sala rectangular de 7,40 x 4,40 metros, sin apenas mobiliario.

3.3 Configuración y disposición de la arquitectura física

El sistema desde el punto de vista de la arquitectura física se compone de 4 puntos de acceso y un dispositivo WiFi móvil.

Los puntos de acceso se han dispuesto en forma de cruz en una sala de 7,40 metros de largo por 4,40 metros de ancho. Su posición es fija y conocida.

Cada punto de acceso está configurado para actuar como una red WiFi de tipo infraestructura y enviar tramas beacon (o tramas balizas) cada 100 ms indicando la presencia de la red al dispositivo móvil. Operan en los canales 1,6,11 y 13 del espectro de frecuencias de la banda de 2.4 GHZ. Están preparados para trabajar con las capas físicas del estándar 802.11 b,g y n.

El Dispositivo móvil por su parte, integra una tarjeta de red WiFi configurada para trabajar como un dispositivo móvil en redes de tipo infraestructura. Está preparada para trabajar en las capas físicas 802.11 b, g y n del estándar. El dominio de regulación de la tarjeta le permite operar en España.

Sistema de posicionamiento en interiores

Cada red WiFi dispone de un identificador propio. Los designaremos con los nombres AP1, AP2, AP3 y AP4. La correspondencia con los puntos de acceso es la siguiente:

La red AP1 corresponde al punto de acceso Huawei y emite en el canal 1 del espectro WiFi.

La red AP2 corresponde al punto de acceso Tp-link TL-WR940N 1 y emite en el canal 11 del espectro WiFi.

La red AP3 corresponde al punto de acceso Tp-link TL-WR940N 2 y emite en el canal 6 del espectro WiFi.

La red AP4 corresponde al punto de acceso Tp-link archer y emite en el canal 13 del espectro WiFi.

La configuración y disposición física del sistema queda de la siguiente forma:

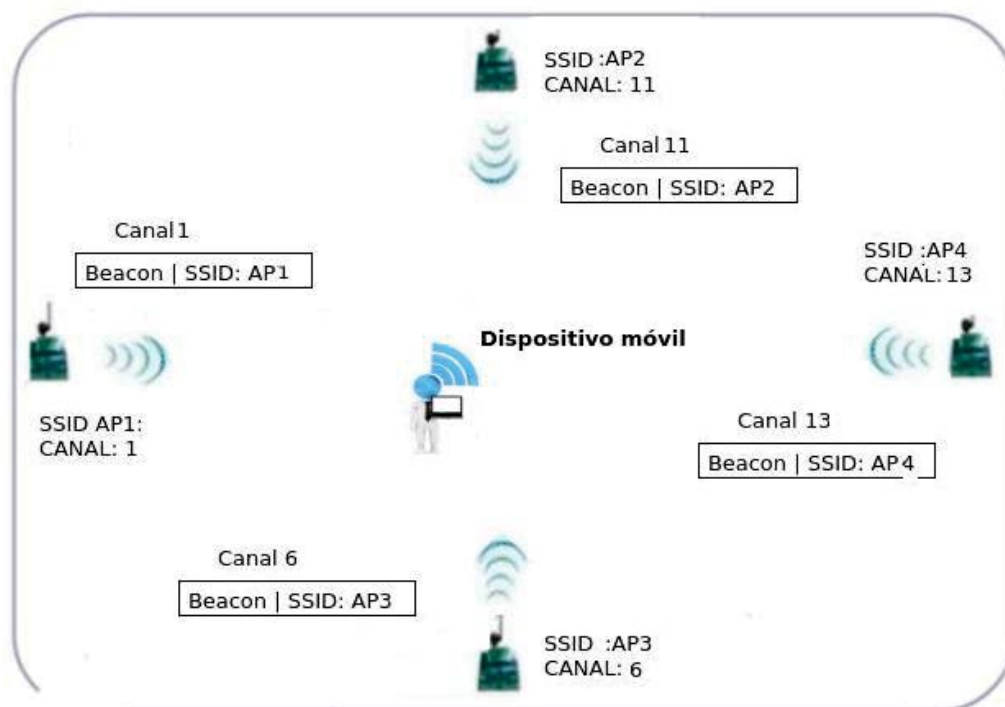


figura 3.1: Configuración y disposición física del sistema

3.4 WiFi como tecnología

Hemos escogido la tecnología WiFi para la realización de este sistema fundamentalmente por tres motivos: El primero es que es una tecnología de muy fácil acceso. Ha sido muy fácil y relativamente barato montar la infraestructura de nuestro sistema y esto es porque existen infinidad de dispositivos que integran una tarjeta de red WiFi (ordenadores, smartphones, videoconsolas, cámaras fotográficas, etc.) y por otra parte la adquisición de puntos de acceso es relativamente barata además de la posibilidad de reutilizar los que ya dispones.

Desde un punto de vista más amplio y con vistas a sistemas comerciales el hecho de la amplia difusión de la tecnología WiFi permite que las infraestructuras actuales puedan ser reutilizadas para ofrecer sistemas de localización lo que ha constituido otro punto a favor para la utilización de esta tecnología.

Un segundo motivo es la facilidad de configuración. Con sólo adquirir un punto de acceso y enchufarlo a la red eléctrica ya dispones de una red WiFi plenamente funcional con la que desarrollar tu sistema de localización. Es verdad que esta configuración es muy básica pero ya sería suficiente para poder desarrollar algún sistema básico de localización simbólica.

Por último y no menos importante es toda la funcionalidad que ofrece una red WiFi “de serie”. Tanto los puntos de acceso como la tarjeta de red y el consiguiente sistema operativo ofrecen una serie de primitivas o funcionalidades que hemos utilizado para el desarrollo de nuestro sistema de localización.

3.4.1 Redes 802.11 (WiFi) de infraestructura

Cada punto de acceso se ha configurado para actuar como una red 802.11 de tipo infraestructura. En total disponemos de 4 redes WiFi en el sistema. Cada una dispone de un identificador propio, el SSID, que también es configurable. Denominaremos a las redes del sistema como AP1, AP2, AP3 y AP4.

Estas redes han sido configuradas para transmitir en canales que interfieran lo mínimo posible entre ellos. El estándar WiFi proporciona 14 canales de transmisión dentro del espectro de la frecuencia 2.4Ghz que es el que hemos utilizado y como se puede apreciar en la imagen sólo existen tres canales que no solapen sus frecuencias que son el 1, el 6 y el 11. En nuestro sistema hemos utilizado además el canal 13.

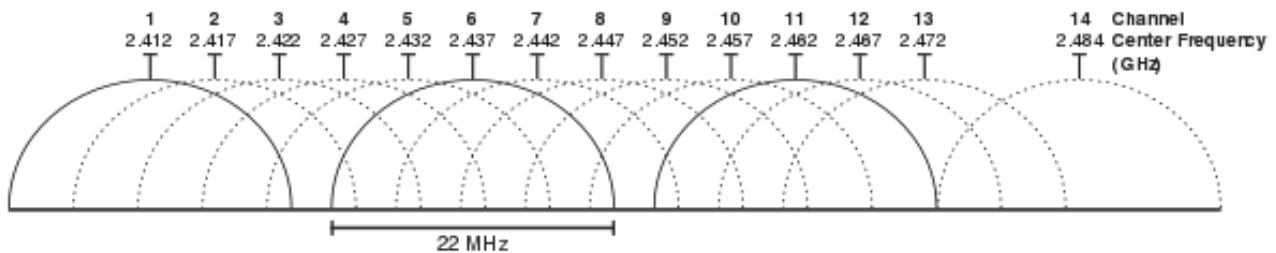


Figura 3.2: Canales WiFi en la banda de 2.4Ghz

Tanto la tarjeta de red utilizada como los puntos de acceso están preparados para usar las capas físicas de los estándares 802.11 b/g/n en la frecuencia de 2.4GHZ.

3.4.2 Escaneo de redes

La tecnología WiFi proporciona un mecanismo de escaneo de redes para que las estaciones cliente puedan descubrir que redes se encuentran accesibles y poder optar a la conexión por alguna de ellas.

Este mecanismo ha resultado fundamental en el desarrollo del módulo sensor puesto que nos ha permitido una comunicación entre las redes WiFi y el dispositivo móvil sin necesidad de una conexión explícita.

Gracias a esta comunicación hemos podido medir el valor RSSI de las tramas recibidas. Existen dos tipos de escaneo: Escaneo pasivo y escaneo activo.

Escaneo pasivo

Las tramas beacon o tramas baliza pertenecen al grupo de tramas de gestión del estándar 802.11. Son unas tramas de difusión que incluyen en su cuerpo diferentes campos con información variada. Entre ésta destacamos el campo SSID que como hemos mencionado antes identifica a la red que envía la trama.

Las redes WiFi (a través de los puntos de acceso) envían periódicamente tramas beacon al entorno para informar de su presencia (por defecto se envían cada 100 ms aunque este periodo es configurable). Las estaciones cliente al recibir una trama beacon saben por su campo SSID que la red con este identificador se encuentra accesible.

Un escaneo pasivo consiste en escuchar las tramas beacon que llegan del entorno y de esta forma conocer que redes están disponibles.

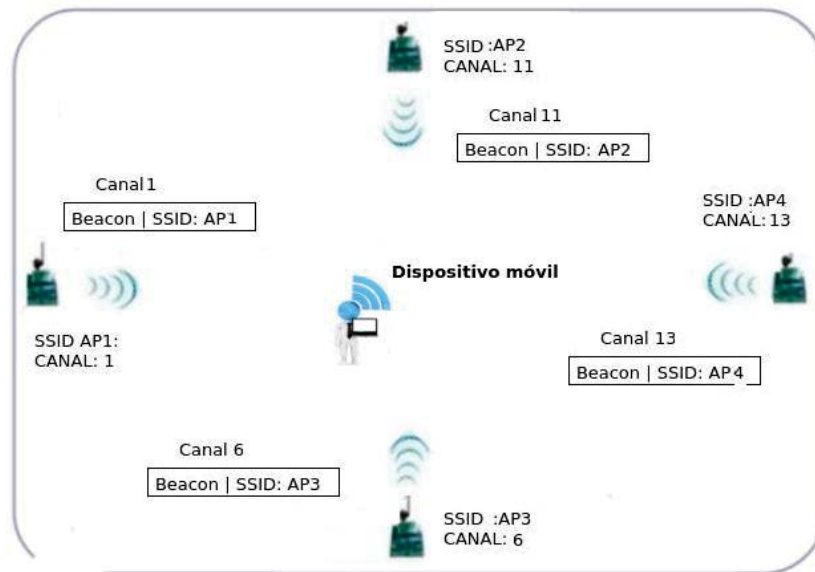


Figura 3.3: Ejemplo de escaneo pasivo

Escaneo Activo

En el escaneo activo la estación cliente pregunta directamente por la existencia de la red o redes requeridas. Se realiza a través de las tramas probe request.

Las tramas probe request son tramas pertenecientes al grupo de tramas de gestión del estándar 802.11 e incorporan un campo SSID con el nombre de la red por la que se pregunta. Normalmente se trata de tramas de difusión aunque pueden estar dirigidas a un punto de acceso en concreto. Si se quiere preguntar por todas las redes que se encuentren accesibles debe enviarse el campo SSID con valor "vacío" lo que se considera como un SSID de "difusión".

Cuando un punto de acceso recibe una trama probe request que pregunta por su propio SSID éste tiene la obligación de contestar con una trama probe response indicando su presencia. Si el campo SSID lleva valor de "difusión" todos los puntos de acceso que la reciban deben contestar con una trama probe response.

Las tramas probe response incorporan diversos campos de información que son similares a los incluidos en las tramas beacon (incluido el campo SSID).

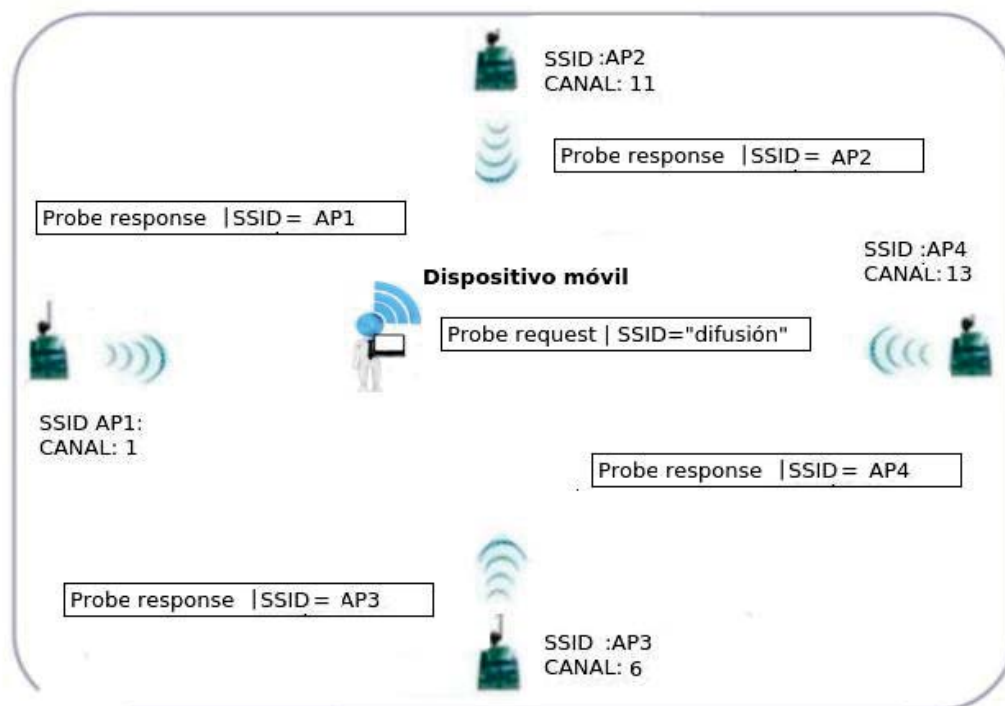


figura 3.4: Ejemplo simplificado de escaneo activo

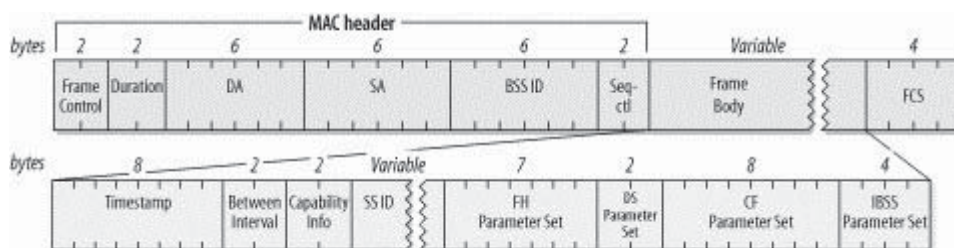


Figura 3.5: Formato de una trama probe request

3.4.2.1 Escaneo de redes en el sistema operativo windows 7

Una vez hemos visto como funciona el mecanismo de escaneo y sus posibles variantes es hora de explicar como podemos llevar a la práctica un escaneo de redes en el sistema operativo Windows 7 así como de que manera podemos obtener el valor RSSI de las diferentes redes escaneadas.

Para ello introduciremos en este capítulo la arquitectura Native 802.11 encargada del manejo de redes WLAN en el sistema operativo Windows 7.

Arquitectura Native 802.11

El sistema operativo Windows implementa desde su versión Vista una arquitectura de software específica para el manejo de redes WiFi (Native 802.11 Wireless LAN Design Guide).

Native 802.11 constituye además un marco de trabajo para el desarrollo de servicios y drivers pues especifica la funcionalidad que deben implementar éstos y la interfaz de comunicaciones que deben seguir.

La siguiente imagen muestra los módulos principales y la comunicación existente entre ellos. Podemos discernir dos bloques principales: un bloque de módulos que opera en modo usuario y otro bloque de módulos que conforman los drivers y que operan en modo sistema.

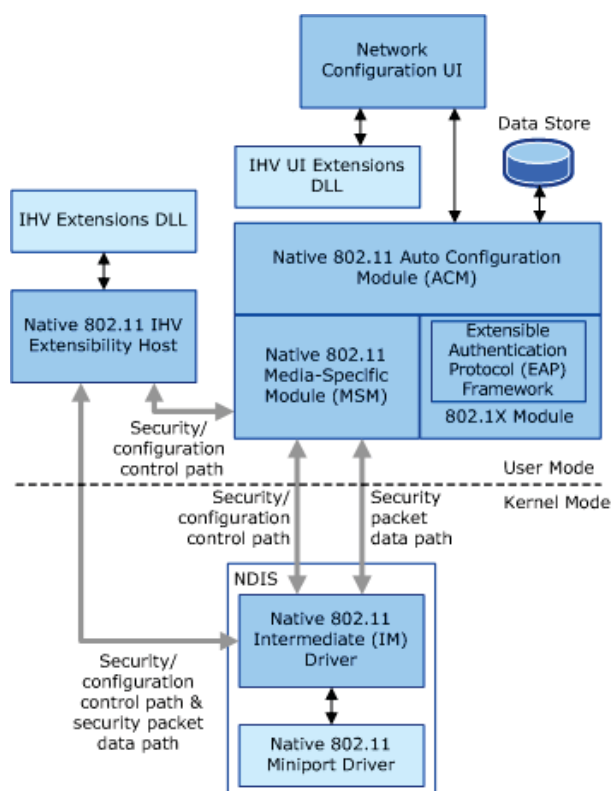


figura 3.6: Arquitectura Native 802.11

Bloque en modo sistema

Está formado por los drivers (controladores software) NDIS del dispositivo 802.11.

Los sistemas operativos Windows ofrecen lo que se denomina la Network driver interface specification (NDIS), algo así como una especificación de la funcionalidad que debe implementar un driver atendiendo a su tipología. De esta forma los fabricantes, si quieren que su dispositivo sea compatible con alguna versión de NDIS, deben implementar la funcionalidad que ésta refiera en su especificación.

Cada versión de NDIS especifica un conjunto de funciones que deben ser implementadas. Cada función se define mediante un OID (objeto identificador). De esta forma si un controlador quiere ser compatible con una especificación NDIS debe implementar los OID que establezca esa especificación. De esta forma gracias a NDIS los fabricantes de dispositivos tienen una guía de referencia para el desarrollo de los controladores compatibles con los sistemas Windows.

Con la aparición de la arquitectura Native 802.11 (a partir de la versión Vista de Windows) aparecen los OIDS Native 802.11 (incluidos a partir de la versión NDIS 6.0). Estos OIDS deben ser implementados por los controladores software de los dispositivos 802.11 si quieren ser compatibles con la arquitectura.

La arquitectura 802.11 está constituida por dos tipos de drivers o controladores:

- Native 802.11 Intermediate (IM): Este controlador constituye el punto de entrada al modo sistema de la arquitectura. Recoge las peticiones del MSM (en forma de OIDS) y las redirige al controlador de minipuerto.
- Native 802.11 Miniport Driver: El controlador de minipuerto es desarrollado por el fabricante del dispositivo y proporciona acceso a las interfaces inalámbricas 802.11. Debe implementar los Native 802.11 OIDS para ser compatible con la arquitectura.

Bloque en modo usuario

Los módulos pertenecientes a este bloque gestionan la funcionalidad 802.11 enviando los OID necesarios a las capas más bajas de la arquitectura. Destacan los siguientes módulos:

- Native 802.11 Auto Configuration Module (ACM): Este es el módulo principal del bloque. Se encarga de gestionar el acceso a las redes WiFi así como de gestionar los perfiles de red. Además Provee una API pública de desarrollo que proporciona las siguientes funcionalidades:
 - Procesamiento de los eventos que genera la tarjeta de red (NIC).
 - Enumeración de los diferentes interfaces (NICS) WiFi disponibles.
 - Consulta de las capacidades y estado de las interfaces.
 - Escaneo de las distintas redes WiFi accesibles.
 - Conexión y desconexión a una red WiFi.
 - Creación, modificación y validación de perfiles de red.
 - Recepción de notificaciones (conexión o desconexión a una red, notificaciones de escaneo de redes completado, etc.).
- Native 802.11 Media Specific Module (MSM) : Este módulo recibe peticiones del modulo ACM a través de una API privada que proporciona entre otros los siguientes servicios:
 - Emitir solicitudes al controlador Native 802.11 intermedio (IM) para descubrir redes WiFi, así como solicitudes para conectarse o desconectarse de una red. Estas solicitudes como se ha comentado antes se hacen a través del establecimiento de un OID por cada solicitud o petición.
 - Emitir solicitudes al controlador Native 802.11 (IM) para consultar o configurar los parámetros de conectividad, seguridad y configuración del controlador Native 802.11 de minipuerto subyacente.

Escaneos en la arquitectura Native 802.11

La arquitectura Native 802.11 permite o está definida para que los dispositivos WiFi puedan escanear las redes inalámbricas del entorno.

Permite tanto escaneos activos como escaneos pasivos. En este punto es necesario tener en cuenta el dominio de regulación que admite el dispositivo y que está relacionado con la zona para la que está pensado que opere. Si este dominio no se encuentra definido sólo podrá realizar escaneos pasivos, aunque podrá realizarlos en cualquier canal. Si por el contrario si se encuentra definido, podrá realizar escaneos activos en los canales que su dominio de regulación le permita y escaneos pasivos en cualquier canal.

En la documentación se refleja que durante la operación de escaneo, los dispositivos WiFi deben permanecer a la escucha en cada canal un tiempo que se considere razonable para poder recibir todas las tramas requeridas (Native 802.11 Wireless LAN Design Guide) . Este periodo de espera depende del tipo de escaneo utilizado y de la implementación que haya realizado el fabricante.

Un punto muy importante a destacar es el siguiente: Según se indica en la documentación, si el dispositivo está operando en modo “extensible station” (estación móvil) debe guardar las redes detectadas durante el escaneo en una caché y actualizar la lista con cada escaneo. Esto es de vital importancia pues nos permitirá consultar esa lista que además contiene el RSSI recibido de cada red.

La operación de escaneo se encuentra definida en 2 OIDS que pertenecen al grupo de los Native 802.11 OIDS. Se trata del OID_DOT_11_SCAN_REQUEST que se utiliza para solicitar al controlador la operación de escaneo y el OID_DOT11_ENUM_BSS_LIST que se utiliza para solicitar al controlador la lista de redes descubiertas.

OID_DOT11_SCAN_REQUEST

Este OID solicita un escaneo desde el módulo de autoconfiguración (ACM) a las capas más bajas de la arquitectura. Especifica de forma completa que opciones de escaneo existen y que parámetros deben indicarse para llevarlas a cabo. Estos parámetros se encuentran en la estructura DOT_11_SCAN_REQUEST_V2 que acompaña al OID en la solicitud de escaneo.

Cuando el controlador de minipuerto recibe este OID solicita a la estación (tarjeta de red) que lleve a cabo el escaneo según los parámetros indicados. Si el escaneo se inicia correctamente el controlador no espera por la finalización de la operación sino que retorna el valor NDIS_STATUS_SUCCESS dando por concluida la solicitud de escaneo.

Cuando el escaneo finaliza el dispositivo lo notifica al controlador que a su vez debe enviar la indicación NDIS_STATUS_DOT11_SCAN_CONFIRM al módulo de autoconfiguración para indicarle si el escaneo se ha realizado con éxito o por el contrario no ha podido realizarse.

Si todas las interfaces están apagadas ya sea mediante interruptor HW o mediante software el escaneo se cancela. De igual forma ocurre si se recibe el OID_DOT11_RESET_REQUEST.

Los parámetros de la petición indicados en la estructura DOT_11_SCAN_REQUEST_V2 son los siguientes:

Dot11ScanType: Este parámetro indica el tipo de escaneo que se va a llevar a cabo. Puede tomar tres valores: activo, para indicar que se quiere llevar a cabo un escaneo activo, pasivo, para indicar que se quiere llevar a cabo un escaneo pasivo y auto en cuyo caso podría llevarse a cabo un escaneo activo, pasivo o una combinación de ambos dependiendo de la implementación que haya realizado el fabricante.

UNumOfdot11SSID: Este parámetro indica el número de redes que se quieren localizar.

UDOT11SSIDOFFSET: Este parámetro contiene la lista de las redes que se quieren localizar. Cada red contenida en la lista viene representada por su SSID.

Si el tipo de escaneo es activo la estación mandará una trama probe request por cada SSID que haya en la lista. Si el tipo de escaneo es pasivo y la lista es vacía (el parámetro UNumOfdot11SSID es cero) se enviará una única trama con un elemento de información SSID de “difusión”.

Dot11BSSType: Este parámetro indica el tipo de las redes que se quieren localizar. Puede tomar los valores de “infraestructura”, “Ad Hoc” o “cualquiera”.

Dot11BSSID: Este parámetro indica la dirección MAC que pondrá en la cabecera BSSID. Si se indica valor nulo establecerá una dirección de difusión MAC.

Además de estos parámetros también podemos especificar que capas físicas y que configuración de éstas queremos utilizar para el escaneo siempre y cuando sean soportadas por la interfaz (tarjeta de red). Una configuración de capa física se define entre otras cosas por la técnica de transmisión a utilizar, los canales que se van a emplear, el tiempo de permanencia en cada canal, etc..

La configuración de capa física que se utilizará en el escaneo se establece en la estructura DOT_PHY_TYPE_INFO que se incluye como un miembro de la estructura DOT11_SCAN_REQUEST_V2. Así que cuando se solicite un escaneo la estructura anterior contendrá las configuraciones de las capas físicas que se quieren emplear en el mismo. Los parámetros más importantes a especificar en la capa física son los siguientes.

UnumOfPhyTypeInfo: Este parámetro indica el número de configuraciones físicas que se utilizarán en el escaneo.

UphyTypeInfoOffset: Este parámetro contiene la lista con las configuraciones físicas que se va a utilizar en el escaneo. Si el controlador de minipuerto opera en modo “extensible station”, las configuraciones de cada capa física

soportada serán establecidas por el propio controlador y no se tendrán en cuenta las configuraciones establecidas en la estructura. Además el controlador establecerá como canales donde escanear todos aquellos que permita la regulación que le corresponda para cada capa física soportada.

OID_DOT11_ENUM_BSS_LIST

Este OID sirve para solicitar al controlador software que devuelva la lista de redes descubiertas en el último escaneo.

3.4.2.2 Native WiFi API

Hasta ahora hemos visto como se estructura la arquitectura Native 802.11, que escaneos permite y como solicitarlos. El problema es que solo el sistema operativo puede solicitar Native OIDS así que nuestro módulo sensor no puede enviar el OID de escaneo sino que debe solicitarlo a través de la Native WiFi API.

El módulo ACM (componente de configuración automática) proporciona una API pública con funcionalidad para gestionar redes y perfiles 802.11. A este API se le denomina Native WiFi API.

Este API ofrece una primitiva llamada wlanScan que se encarga de solicitar escaneos y otra primitiva llamada WlanGetNetworkBssList que se encarga de recuperar las redes descubiertas en el escaneo. Además de estas primitivas existen otras que también hemos utilizado en el desarrollo de nuestro sistema.

Función wlanScan

La función wlanScan es la función que proporciona el API para el escaneo de redes inalámbricas.

```
C++  
  
DWORD WINAPI WlanScan(  
    _In_          HANDLE          hClientHandle,  
    _In_          const GUID       *pInterfaceGuid,  
    _In_opt_      const PDOT11_SSID pDot11Ssid,  
    _In_opt_      const PWLAN_RAW_DATA pIeData,  
    _Reserved_    PVOID            pReserved  
);
```

Parámetros

Sistema de posicionamiento en interiores

hClientHandle [entrada]: Es necesario introducir un objeto manejador de la sesión.

PInterfaceGuid [entrada]: Este parámetro hace referencia al interfaz de red (tarjeta de red) que se va a usar para realizar el escaneo.

PDot11Ssid [entrada, opcional]: Este parámetro indica la red por la que debe filtrar el escaneo. Si el parámetro se rellena con valor nulo entonces se hará un escaneo de todas las redes disponibles.

PleData [entrada, opcional]: Este parámetro indica los elementos de información que se van a incluir en las tramas. (el envío de este parámetro implica el envío de tramas probe request).

pReserved: Parámetro reservado para uso futuro. Debe tener valor nulo.

Retorno

Si la función se ha ejecutado con éxito se devolverá el valor "ERROR_SUCCESS".

Si por el contrario la función no ha podido ejecutarse devolverá alguno de los siguientes códigos:

Valor devuelto	Descripción
ERROR_INVALID_PARAMETER	hclientHandle tiene valor nulo o inválido, pInterfaceGuid tiene valor nulo, pReserved tiene valor no nulo
ERROR_INVALID_HANDLE	El manejador hClientHandle no se encuentra en la tabla de manejadores
RPC_STATUS	Errores varios
ERROR_NOT_ENOUGH_MEMORY	Memoria insuficiente para asignar a la lista de resultados

Descripción del funcionamiento

Sistema de posicionamiento en interiores

La función wlanScan solicita al controlador software del dispositivo que realice un escaneo de redes inalámbricas, seguramente mediante el envío del `OID_DOT11_SCAN_REQUEST`.

Es necesario introducirle el manejador de la sesión y la interfaz por la que queremos que se realice el escaneo.

Podemos preguntar por una red determinada introduciendo en el parámetro `PDot11Ssid` la red que queremos buscar. No se indica en la especificación (Native WiFi API) si la indicación de este parámetro supone un escaneo activo o no. Si queremos preguntar por la existencia de varias redes debemos realizar una llamada a wlanScan por cada red por la que queremos preguntar. Hemos hecho algunas pruebas y el tiempo que suele consumir el pc para realizar una petición es de alrededor de un segundo, por lo tanto si quisiéramos realizar una petición por cada red de nuestro sistema el tiempo requerido, sólo para escanear, sería de unos 4 segundos lo que sería un tiempo inviable para un uso real. Por lo tanto utilizaremos el parámetro `pDot11Ssid` con valor a nulo. De esta forma haremos un escaneo de todas las redes que sean accesibles desde el interfaz indicado.

La función permite introducir elementos de información al escaneo. Esto sólo es posible si se envían tramas probe request lo que implica un escaneo activo. No necesitamos usar elementos de información por lo que dejaremos este parámetro vacío.

Una vez iniciada la función wlanScan no espera a que concluya el escaneo. Si la solicitud se ha realizado con éxito termina su ejecución devolviendo el valor `"ERROR_SUCCESS"` (esto coincide con lo visto en la especificación del `OID_DOT11_SCAN_REQUEST`).

Es necesario llamar a la función wlanRegisterNotification con el parámetro `WLAN_NOTIFICATION_SOURCE_ACM` para recibir una notificación cuando el escaneo haya finalizado. De esta forma podremos saber cuando el escaneo ha terminado y si lo ha hecho satisfactoriamente.

La documentación también indica que el uso de esta función no implica que el resultado del escaneo se vaya a añadir al resultado de escaneos previos. Menciona que es posible que al iniciar un escaneo el driver borre la lista de redes del escaneo anterior.

En las pruebas que hemos realizado (con pDot11Ssid y pleData con valores nulos) cada vez que se realiza un escaneo se actualiza por completo la lista de redes. Las redes que ya no se encuentran accesibles no se mantienen en la lista. De esta forma cada vez que el sistema realice un escaneo sólo devolverá aquellas redes que son accesibles en el momento del escaneo.

WlanGetNetworkBssList

Esta función devuelve la lista de BSS de una red o redes accesibles desde un interfaz determinado.

Prototipo

```
DWORD WlanGetNetworkBssList(  
    HANDLE          hClientHandle,  
    CONST GUID      *pInterfaceGuid,  
    CONST PDOT11_SSID pDot11Ssid,  
    DOT11_BSS_TYPE  dot11BssType,  
    BOOL            bSecurityEnabled,  
    PVOID           pReserved,  
    PWLAN_BSS_LIST  *ppWlanBssList  
);
```

Parámetros

hclientHandle [entrada]: Es necesario introducir un objeto manejador de la sesión.

pInterfaceGuid [entrada]: Este parámetro hace referencia al interfaz de red (tarjeta de red) de la que se quiere obtener la lista de BSS.

pDot11Ssid [entrada, opcional]: Este parámetro indica la red de la cuál quiere obtenerse la lista de BSS. Si el parámetro se envía con valor nulo la función devuelve la lista de BSS de todas las redes accesibles desde la interfaz.

Dot11BssType [entrada]: Este parámetro indica el tipo de la red indicada en el parámetro pDot11Ssid. Este parámetro es ignorado si pDot11Ssid es nulo. Puede tomar uno de los siguientes valores.

Valor	Significado
dot11_BSS_type_infrastructure	Red de tipo infraestructura
dot11_BSS_type_independent	Red Ad-Hoc
dot11_BSS_type_any	Cualquier red

BsecurityEnabled [entrada]: Indica si la red identificada en el parámetro pDot11Ssid tiene la seguridad habilitada.

Preserved [entrada]: Este parámetro está reservado para uso futuro. Debe tener valor nulo.

PpWlanBssList [salida]: Este parámetro devuelve la lista de BSS solicitada.

Retorno

Si la función se ha ejecutado con éxito se devolverá el valor "ERROR_SUCCESS".

Si por el contrario la función no ha podido ejecutarse devolverá alguno de los siguientes códigos:

Código de error	Descripción
ERROR_INVALID_HANDLE	El manejador introducido no se ha encontrado en la tabla de manejadores.
ERROR_INVALID_PARAMETER	El parámetro hClientHandle, pInterfaceGuid, o ppWlanBssList es nulo. El parámetro pReserved no es nulo. El parámetro hClientHandle, pDot11Ssid o dot11BssType tiene un valor incorrecto.

La interfaz esta apagada.

ERROR_NDIS_DOT11_POWER_STATE_INVALID

ERROR_NOT_ENOUGH_MEMORY	No hay suficiente memoria para almacenar la lista de resultados.
ERROR_NOT_FOUND	La interfaz especificada no se encuentra.
ERROR_NOT_SUPPORTED	Funcionalidad no soportada (solo en versiones windows xp) o servicio de autoconfiguración de wlan deshabilitado.
ERROR_SERVICE_NOT_ACTIVE	El servicio de autoconfiguración de wlan no está activo.
RPC_STATUS	Errores varios

Descripción del funcionamiento

Cuando hablamos de BSS nos estamos refiriendo a puntos de acceso en el caso de redes de infraestructura o compañeros de red en el caso de redes Ad-Hoc. Esta función devuelve todos los puntos de acceso y máquinas compañeras de red de la red o redes que se indiquen y que han sido descubiertas en el último escaneo con el interfaz especificado. Seguramente se realice mediante el envío del `OID_DOT11_ENUM_BSS_LIST` al controlador software de la interfaz.

Es necesario indicar un manejador de sesión y el interfaz de donde queremos obtener los bss descubiertos en el último escaneo.

Si indicamos una red mediante el parámetro `pDot11Ssid` la función devolverá la lista de BSS correspondiente a esa red. Si indicamos un valor nulo para ese parámetro la función devolverá todos los BSS correspondientes a todas las redes accesibles desde el interfaz introducido. Nuestro propósito es obtener todos los puntos de acceso que pertenecen a las redes del sistema y que han sido descubiertas en el ultimo escaneo del interfaz, por lo tanto llamaremos a la función con el parámertro `pDot11Ssid` con valor nulo.

El resto de parámetros al indicar `pDot11Ssid` con valor nulo se ignoran así que los enviaremos con alguno de los valores válidos.

Si la función termina con éxito devuelve el valor `"ERROR_SUCCESS"` y el parámetro `PpWlanBssList` relleno con la lista de BSS requerida. Cada elemento

Sistema de posicionamiento en interiores

de la lista consiste en una estructura de tipo `wlanBssEntry`. Esta estructura en nuestro caso se corresponde con cada punto de acceso y tiene entre sus miembros el valor RSSI que se recibió del mismo y la red SSID a la que pertenece. Si filtramos por los SSID AP1, AP2, AP3 y AP4 obtendremos los puntos de acceso que forman nuestro sistema. Consultando su atributo RSSI tendremos los valores que devolverá el módulo sensor.

Otras funciones del API

Pasamos a enumerar otras funciones del API que hemos utilizado en el desarrollo del sistema.

WlanEnumInterfaces

Esta función devuelve las interfaces WiFi que se encuentran disponibles en el pc local.

Prototipo

```
DWORD WlanEnumInterfaces(  
    HANDLE                hClientHandle,  
    PVOID                 pReserved,  
    PWLAN_INTERFACE_INFO_LIST *ppInterfaceList  
);
```

Esta función la usaremos para determinar que interfaces tenemos disponibles y poder escoger alguna para el escaneo.

WlanQueryInterface

Esta función permite consultar características de una interfaz de red.

Prototipo

```
DWORD WlanQueryInterface(  
    HANDLE                hClientHandle,  
    CONST GUID            *pInterfaceGuid,  
    WLAN_INTF_OPCODE      OpCode,  
    PVOID                 pReserved,  
    PDWORD                 pdwDataSize,  
    PVOID                 *ppData,  
    PWLAN_OPCODE_VALUE_TYPE pWlanOpcodeValueType  
);
```

Utilizaremos esta función para comprobar que el interfaz cumple los requisitos necesarios para nuestro trabajo. En concreto se realizarán cuatro comprobaciones al inicio de la ejecución del módulo sensor. Estas comprobaciones son:

- Que la interfaz a utilizar trabaja en modo infraestructura.
- Que la interfaz está preparada para operar.
- Que el modo de operación que presenta la interfaz es extensible station.
- Que en el dominio de regulación de la interfaz está incluido España.

Si alguna de estas comprobaciones no resulta satisfactoria el sistema indicará la imposibilidad de realizar localizaciones y se cerrará.

3.4.3 Control de acceso al medio

Las redes Wi-Fi al compartir el medio de transmisión entre las estaciones necesitan de un protocolo que controle el acceso al mismo para evitar colisiones.

El mecanismo de acceso al medio para la transmisión de una trama es el siguiente:

La estación cliente quiere transmitir una trama y para ello escucha el canal donde va a transmitir.

- Si el canal está libre durante un tiempo predeterminado (DIFS) entonces envía la trama.
- Si el canal por el contrario se encuentra ocupado:

Sistema de posicionamiento en interiores

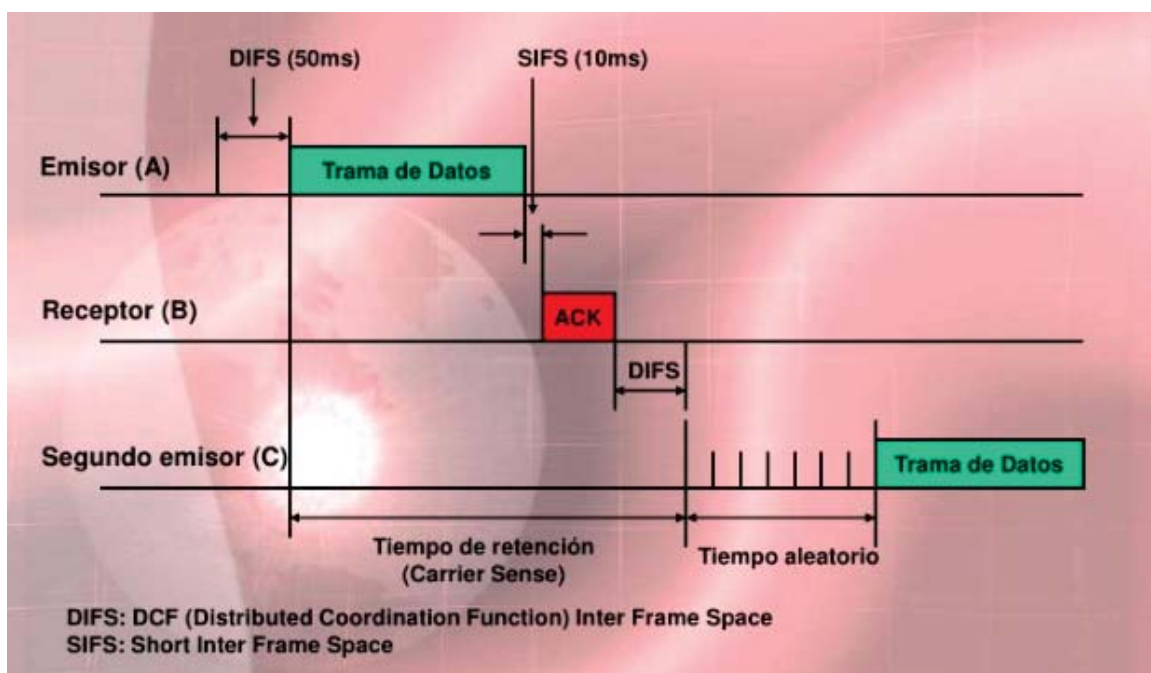
- Espera a que la transmisión en curso termine (esto incluye tanto la trama de emisión como la trama de confirmación de la misma).
- Vuelve a sondear el canal de modo que si éste está libre durante un tiempo predeterminado IFS entonces inicia un periodo de espera aleatorio después del cual transmite la trama.

Este periodo de espera aleatorio consiste en un número de slots de tiempo. Este número puede tener valores entre 0 y $(CW - 1)$.

CW puede tomar valores entre CW_{min} y CW_{max} ambos dependientes del medio físico utilizado en las transmisiones. La elección del valor CW concreto depende de cada estación cliente. De igual forma el tiempo correspondiente a cada slotTime depende del medio físico utilizado.

	CWmin	CWmax	SlotTime
FHSS	15	1023	50 μs
DSSS	32	1023	20 μs
OFDM	15	1023	9 μs

De esta manera dos estaciones que han encontrado el medio ocupado será muy difícil que transmitan la trama a la vez puesto que cada una tendrá un periodo de espera aleatorio distinto.



Al terminar de enviar la trama la estación espera a que el receptor le envíe una confirmación (ACK).

No obstante pueden existir colisiones en los siguientes casos:

- A) Dos estaciones detectan el canal libre y transmiten prácticamente a la vez.
- B) Dos estaciones que habían detectado el medio ocupado eligen el mismo periodo de espera aleatorio por lo que transmiten a la vez.
- C) Una estación detecta el canal libre y transmite una trama mientras otra a una cierta distancia esta transmitiendo de modo que ambas tramas colisionan en el receptor.

Detección de colisiones

En el protocolo CSMA/CA no hay detección de colisiones como tal sino que al no recibir confirmación de la trama enviada se considera que ha habido una colisión.

En ese caso la estación inicia el periodo de espera aleatoria modificando el valor de CW que pasa a ser $CW = CW * 2 + 1$. Cada vez que se produce una colisión se modifica el valor CW de esta forma hasta alcanzar el máximo CW_{max} . Por el contrario si la trama se envía correctamente el valor CW toma el valor de CW_{min} .

Nuestro sistema de localización hará uso del protocolo CSMA/CA ya que este funciona de forma implícita en cualquier red 802.11.

Gracias a esto todas las comunicaciones de las estaciones móviles tendrán un acceso controlado al medio y de igual forma las colisiones que se produzcan, en principio no provocarán la pérdida de las tramas puesto que estas se retransmitirán.

3.5 RSSI

3.5.1 Definición

El RSSI es un parámetro que utiliza el estandar ieee 802.11 para indicar a la entidad MAC el nivel de energía que tienen las tramas recibidas (Bardwell, J. 2004). Cuando el dispositivo físico recibe una trama realiza una medición de la energía o fuerza que tiene la señal de esa trama y la convierte a valores RSSI. Si la trama es aceptada se envía a la capa MAC (controlador) indicándole el valor RSSI obtenido.

Según en la citado estándar el RSSI se define simplemente como un número entero que puede tomar valores entre 0 y 255 (8 bits). Además menciona que no es necesario utilizar todo el rango de valores por lo que cada fabricante define que valores RSSI utilizará para indicar la fuerza de la señal denominándose al valor máximo RSSI_MAX. Por ejemplo, Cisco utiliza 101 valores para indicar la energía de la señal recibida, desde 0 a RSSI_MAX = 100.

También menciona que no se especifica que sensibilidad debe ser cubierta por el indicador RSSI, dejándolo de nuevo en manos del fabricante. Siguiendo con el ejemplo de Cisco, su indicador RSSI cubre el rango [-113 dbm, -10 dbm] haciendo corresponder el valor de -10 dbm al valor RSSI_MAX y el valor -113 dbm al valor RSSI = 0. La conversión o mapeo entre RSSI y dbm (decibelio-miliwatio) es también dependiente del fabricante (Bardwell, J. 2002) y (Bardwell, J. 2004).

Durante el escaneo, al recibir una trama beacon o probe response el dispositivo mide la fuerza de señal de la trama y se la notifica al controlador. Éste realiza una conversión a unidades dbm (decibelio-miliwatio) que es dependiente del fabricante y guarda en memoria la nueva red descubierta junto con su valor RSSI. Al finalizar el escaneo el controlador envía una notificación de escaneo terminado al módulo de configuración automática. De este modo mediante el mecanismo de escaneo se han obtenido las redes accesibles junto con su valor RSSI asociado.

3.5.2 Retos que presenta el uso del RSSI en sistemas de localización

El uso del indicador RSSI en sistemas de posicionamiento presenta bastantes retos: Unos derivados del hw y otros derivados de las propias características que presenta la señal.

3.5.2.1 Problemas derivados del hardware

Intervalo de sensibilidad insuficiente

Un intervalo de sensibilidad insuficiente puede suponer un problema a la hora de discernir una posición. Imaginemos, por ejemplo, que un dispositivo es capaz de indicar mediante RSSI el intervalo [-50 dbm (RSSI_MAX), -100 dbm] y que a una distancia de 3 metros la intensidad de señal recibida es de -50dbm (lo que hemos reportado en alguna de nuestras medidas) por lo que el parámetro RSSI en esa posición tendría un valor RSSI_MAX. Si nos acercáramos más al punto de acceso, a pesar de que la intensidad de la señal recibida sería mayor, el dispositivo seguiría indicando RSSI_MAX ya que es la sensibilidad máxima que puede reportar. Esto significa que a partir de una determinada

distancia al acercarnos más al punto de acceso el indicador RSSI dará el mismo valor imposibilitando al algoritmo discernir con claridad la posición auténtica de ubicación y con ello disminuyendo la exactitud del sistema de localización.

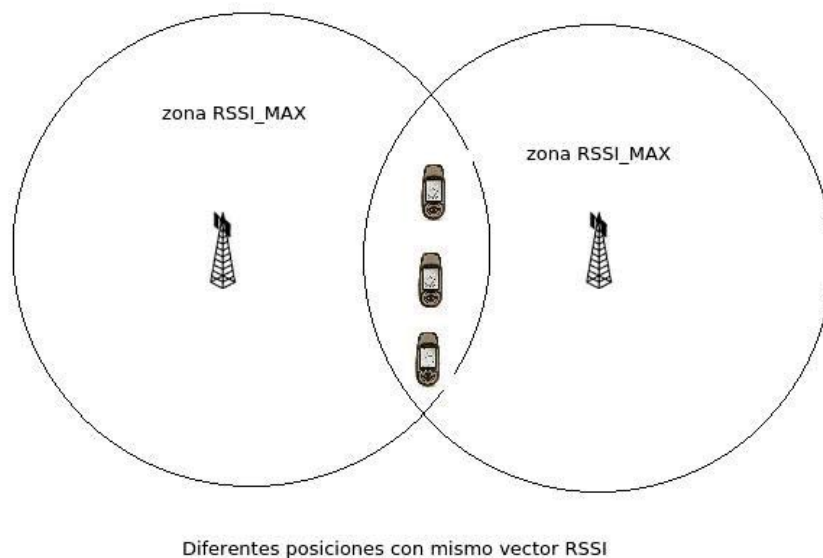


Figura 3.8: Insuficiente rango de sensibilidad

En las pruebas realizadas con nuestra tarjeta de red hemos observado un rango de sensibilidad desde -15 dbm estando a escasos centímetros del punto de acceso hasta -65 dbm situados a 7 metros del mismo lo que consideramos un buen rango de sensibilidad.

Escala de valores RSSI insuficiente

El número de valores diferentes que puede reportar el RSSI es importante a la hora de valorar la exactitud de un sistema. Si este número es pequeño da lugar a una baja exactitud. Por ejemplo, supongamos un dispositivo con un rango de sensibilidad que va desde -100dbm hasta 0dbm pero que solo cuenta con 10 valores RSSI para representarlos. Esto podría significar que el dispositivo reporte un valor RSSI para cada 10 valores de intensidad dbm. De esta forma llegamos al mismo problema que anteriormente en el que existen varios puntos distintos del entorno en los que el valor RSSI es el mismo.

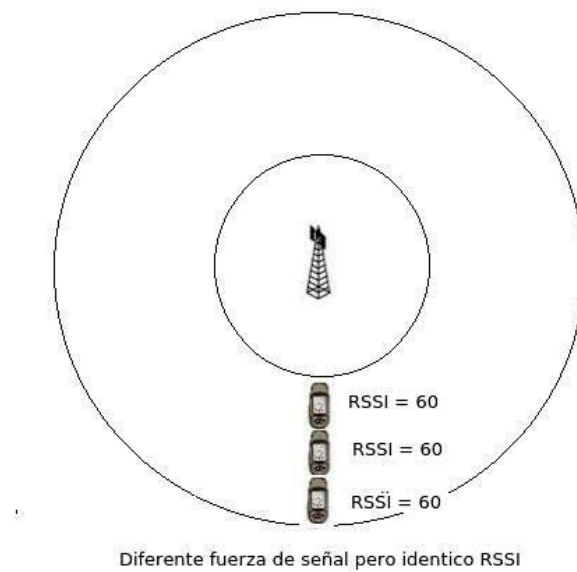


figura 3.9: Rango de valores RSSI insuficiente

Distinto fabricante, distinto RSSI

Si cada fabricante establece el intervalo de sensibilidad del dispositivo, el rango de valores RSSI que utilizará para indicarlo y la correspondencia entre el valor RSSI y el valor en dbm significa que lo más probable es que cada dispositivo dé un valor diferente de RSSI aún estando en la misma posición lo que dificulta implementar un sistema global en el que cualquier dispositivo con tarjeta 802.11 pueda usar el sistema para localizarse.

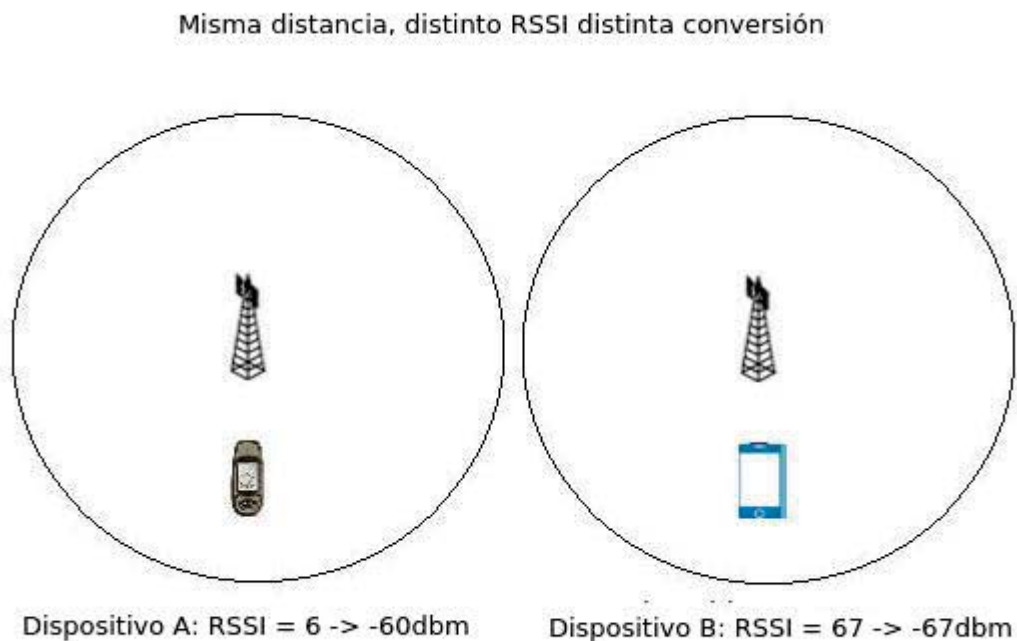


Figura 3.10: RSSI entre distintos fabricantes

A raíz de este punto hicimos unas pruebas con la tarjeta de red externa y encontramos que los valores RSSI que reportaba eran algo diferentes y sobre todo la fluctuación era mayor.

Orientación de la antena

Según refieren (Krishnamurthy, K. K. 2004) y (Zhan, J. L. (2014) la orientación de la antena respecto al punto de acceso puede influir en el valor RSSI recibido. Esto es debido a que según la orientación de ésta puede estar en línea de visión directa con el punto de acceso o por el contrario bloqueada por el propio dispositivo que la integra respecto a éste. Esto provoca variación en las mediciones dependiendo de la orientación.

En las pruebas realizadas hemos comprobado que la orientación de la antena influye en el valor RSSI recibido.

3.5.2.2 Problemas derivados del entorno

Interferencias con otros dispositivos WiFi

Es posible que otros dispositivos WiFi que emitan en el mismo canal puedan alterar el nivel de intensidad de la señal y con ello el valor del RSSI. Esto es muy poco probable debido a la acción del protocolo de acceso al medio CSMA/CA que impide a un dispositivo transmitir si otro está usando el canal. Si a pesar de ello hubiera colisiones la propia capa MAC o la capa física detectaría la corrupción de la trama y la invalidaría.

Fluctuaciones en el RSSI

Un problema a destacar es la fluctuación que presenta la señal en cuanto a su intensidad incluso estando parado en un mismo punto lo que provoca fluctuaciones en el RSSI. Esto es debido a tres factores: Variaciones en el entorno (movimiento de personas, movimiento de mobiliario, etc...) que provoquen obstaculizaciones temporales de la señal dando lugar a fluctuaciones, ruido en el entorno creado por otros dispositivos de radiofrecuencia, efecto multicamino.

En las pruebas que hemos realizado hemos comprobado que el valor RSSI de la señal fluctúa estando el dispositivo fijo en una posición. Esta fluctuación se ha dado sin presencia de personas en el entorno por lo que creemos que se debe en gran medida al efecto multicamino. Estas fluctuaciones han llegado a ser de mas de 10 dbm.

3.6 Algoritmo de localización: Perceptrón multicapa

Para realizar la estimación de la posición hemos escogido un algoritmo de tipo fingerprinting, en concreto una red neuronal perceptrón multicapa.

La elección de este algoritmo se debe a varias cuestiones relacionadas con la naturaleza del problema y que pasamos a explicar a continuación.

El problema a resolver plantea que a partir del valor RSSI recibido de varias señales pueda obtenerse la posición ocupada en relación a un entorno interior. Esto tiene su razón de ser en que la intensidad o fuerza de la señal varía con la distancia siguiendo una regla bien conocida.

El primer problema que surge es que esta regla sólo es válida si hay línea de visión directa entre emisor y receptor. En el espacio libre esta regla da buenos resultados pero en entornos interiores donde la presencia de obstáculos es habitual esta línea de visión directa es difícil de conseguir. Además, el ruido procedente del entorno (efecto multicamino, ruido de otros dispositivos de

radiofrecuencia, movimiento de personas, etc.) dificulta aún más la aplicación de modelos de estimación.

Esto no significa que no siga existiendo una regla en entornos interiores que relacione la fuerza de señal con la distancia. Existen modelos que tienen en cuenta todos estos factores y que son totalmente dependientes del entorno donde se vaya a desarrollar la aplicación. El principal problema de estos modelos es su dificultad para obtenerlos.

Las redes neuronales constituyen un método que proporciona buenos resultados en problemas donde el conocimiento subyacente no está completamente definido o es difícil de obtener. Esto encaja con el problema a resolver pues aunque existe una función que relaciona potencia recibida con posición en el entorno, ésta no está completamente definida y además es bastante difícil de obtener.

Además las redes neuronales trabajan muy bien con ruido y son capaces de obtener muy buen rendimiento aún con la presencia de éste. Esto presenta a las redes neuronales como una muy buena opción pues el problema a resolver presenta ruido en los valores de RSSI debido principalmente al efecto multicamino y a los cambios que ocurren en el entorno.

Por otra parte las redes neuronales son algoritmos muy rápidos en su fase de ejecución algo muy importante en sistemas de localización que exigen tiempos bajos de latencia.

Por ello decidimos emplear una red neuronal como algoritmo de localización para nuestro problema, en concreto un perceptron multicapa.

3.6.1 Perceptrón multicapa

Descripción

El perceptrón multicapa es un modelo de red unidireccional compuesto por una capa sensorial o de entrada, una o más capas ocultas y una capa de salida. Puede usarse tanto para problemas de clasificación como para problemas de aproximación funcional.

De hecho una de las características más importantes del perceptrón multicapa es su utilización como aproximador universal de funciones.

Como se recoge en (Molina, B. m. 2006) sea $f(x)$ una función no constante, acotada y monótona creciente. Sea K un subconjunto compacto (acotado y cerrado) de \mathbb{R}^n . Sea un número real ε en \mathbb{R} , y sea k en \mathbb{Z} , tal que $k \geq 3$, que fijamos. En estas condiciones se tiene que:

Sistema de posicionamiento en interiores

Cualquier mapping $g: x \text{ en } K \rightarrow (g_1(x), g_2(x), \dots, g_m(x)) \text{ en } R^m$, con $g_i(x)$ sumables en k , puede ser aproximado en el sentido de la topología L_2 en K por el mapping entrada-salida representado por una red unidireccional (perceptrón multicapa) de k capas ($k-2$ ocultas), con $f(x)$ como función de transferencia de las redes ocultas, y funciones lineales para las de las capas de entrada y de salida.

Esto significa que para todo $\varepsilon > 0$ existe un perceptron multicapa de las características anteriores, que implementa el mapping:

$$g': x \text{ en } K \rightarrow (g'_1(x), g'_2(x), \dots, g'_m(x)) \text{ en } R^m$$

de manera que:

$$d_{L_2(K)}(g, g') < \varepsilon$$

Esto significa que cualquier función continua en un intervalo puede ser aproximada hasta el nivel deseado por un perceptrón multicapa con una única capa oculta.

De este teorema podemos concluir que es posible encontrar un perceptrón multicapa con una única capa oculta que aproxime el mapping que buscamos hasta el error que deseemos. Para ello es necesario encontrar la arquitectura adecuada, seleccionar un conjunto de patrones de entrenamiento representativo del problema y emplear un método de aprendizaje que proporcione una buena capacidad de generalización.

Arquitectura

La arquitectura del perceptrón multicapa consta de una capa de entrada, una o varias capas ocultas y una capa de salida.

Cada capa consta de un conjunto de neuronas. Cada neurona consta de un conjunto de entradas (x_i), un conjunto de pesos sinápticos asociados a las entradas (w_{ij}) y una salida (y_j).

Las neuronas pertenecientes a las capas ocultas o a la capa de salida poseen una función de potencial sináptico, una función de activación o transferencia y una función de salida.

El funcionamiento de la red es el siguiente: la capa de entrada actúa como una capa sensora que recibe los datos del exterior y los envía a la capa siguiente. No realiza ningún procesamiento sobre éstos.

Las capas ocultas procesan los datos recibidos de la capa anterior y los envían a la capa siguiente.

La capa de salida procesa los datos recibidos de la capa anterior (última capa oculta) y obtiene el resultado de la red neuronal.

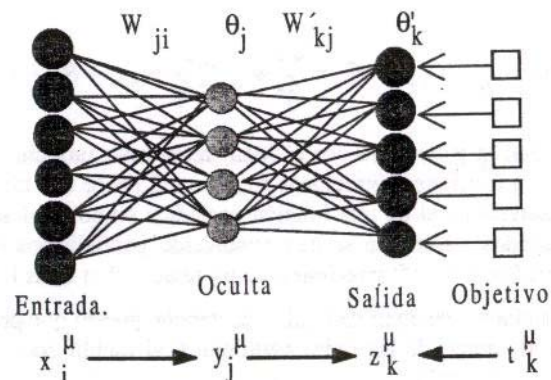


figura 3.11: Arquitectura de un perceptrón multicapa con una capa oculta obtenido de (Molina, B. m. 2006)

Las neuronas de las capas ocultas y de salida al recibir valor en las entradas calculan el potencial sináptico (h_i). La función de potencial sináptico se calcula de siguiente manera:

$$h_i = \sum w_{ij} * x_j$$

donde w_{ij} hace referencia al peso asociado a la entrada j de la neurona i . Se suele añadir una entrada extra (x_0) cuyo valor es siempre -1 y con un peso asociado para tener más grados de libertad a la hora del entrenamiento.

Posteriormente calculan la función de activación (f_i) sobre el potencial sináptico obtenido (h_i). En las capas ocultas la función de activación debe ser no constante, acotada y monótona creciente. En el perceptrón multicapa suelen utilizarse funciones de activación sigmoidea mientras que en la capa de salida suele utilizarse la función identidad.

Función sigmoidea: $f_i = 1 / (1 + e^{-x})$

Función tangente hiperbólica: $f_i = (e^x - e^{-x}) / (e^x + e^{-x})$

Finalmente las neuronas de estas capas aplican la función de salida al valor obtenido por la función de activación. En el perceptrón multicapa suele utilizarse la función identidad.

$$y_i = f_i = F(\sum w_{ij} * x_j)$$

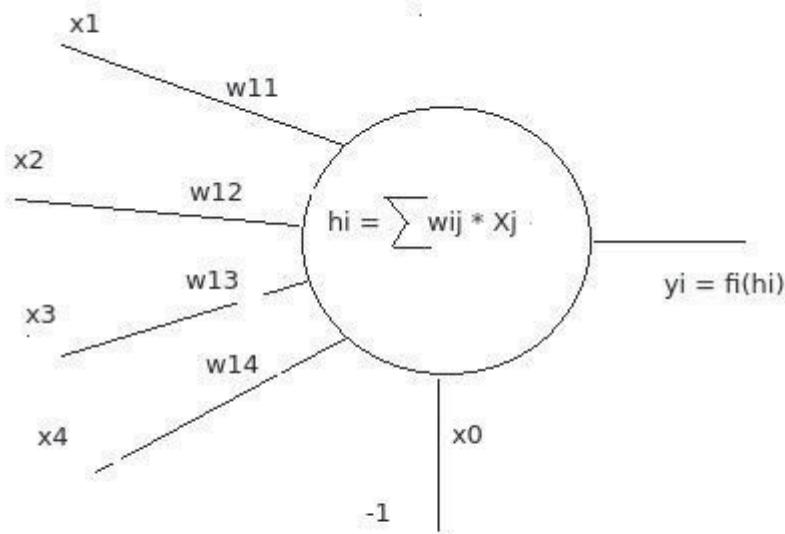


figura 3.11: Modelo de neurona en el perceptrón multicapa

Fase de aprendizaje

Tal y como se describe en (Molina, B. m. 2006) La fase de aprendizaje tiene por objetivo encontrar el conjunto de pesos óptimo de modo que el error cometido por la red para el conjunto de patrones de entrenamiento sea mínimo.

Se trata de un problema de optimización de la función error. Se aplica un método iterativo donde se van ajustando los pesos de la red hasta conseguir el error deseado.

Para ello se parte de una arquitectura y se le pasa el conjunto de patrones de entrenamiento. Se calcula el error que comete la red para todo el conjunto, normalmente mediante la función de error cuadrático medio. Si el error es inferior o igual al deseado, la red da por concluido el entrenamiento estando preparada para ser utilizada en casos reales. Si por el contrario el error es superior al deseado se modifican los pesos siguiendo alguna regla de aprendizaje y se ejecuta una nueva iteración del proceso. La función error debe ser dependiente de los pesos de la red $E(W)$.

La regla de aprendizaje utilizada en el perceptrón multicapa recibe el nombre de regla de retropropagación y se basa en el método de descenso por el gradiente. Se parte de una cierta configuración de los pesos $W(0)$ y se calcula

el sentido de la máxima variación de la función error $E(W)$ en $W(0)$ que vendrá dado por su gradiente en $W(0)$. El gradiente apuntará a un máximo de la función error por lo que los pesos se actualizarán siguiendo el sentido contrario al indicado por el gradiente. De esta forma el método siempre avanza hacia un mínimo de la función. Matemáticamente se expresa de la siguiente forma:

$$W(t + 1) = W(t) - \xi \nabla E(W)$$

donde ξ denota el ritmo de aprendizaje, esto es, el tamaño del paso empleado en cada iteración sobre la función error. Conviene usar un valor lo suficientemente alto para que el proceso de entrenamiento no sea excesivamente lento y que además no provoque oscilaciones que no nos permitan llegar al mínimo.

En nuestro proyecto utilizaremos una variante del algoritmo de retrpopropagación denominada regla de Levenberg–Marquardt.

Generalización: validación cruzada

Uno de los aspectos más interesantes de las redes neuronales es su capacidad de generalización a partir de ejemplos. Una red correctamente entrenada para generalizar es capaz de aprender la relación que existe entre las entradas y salidas presentadas. Es capaz de extraer el conocimiento subyacente que existe en esas relaciones. Por ello responderá correctamente ante patrones nunca vistos con anterioridad.

El problema surge cuando la red no aprende el mapping sino que lo memoriza. Esto ocurre cuando se entrena una red para conseguir un error muy pequeño (por ejemplo, inferior al 1%) en el que la red después de un gran número de iteraciones ya sólo memoriza la asociación entrada – salida pero no el conocimiento o relación que existe entre ambos y por tanto el error ante patrones nuevos es grande (error de generalización).

Para conseguir una buena generalización es necesario el uso de la validación cruzada. La validación cruzada consiste en usar dos conjuntos de patrones, ambos representativos del problema a solucionar, en la fase de aprendizaje. Uno para el entrenamiento en sí que se llamara conjunto de aprendizaje y otro para la validación que se llamará conjunto de validación.

De la misma manera ahora deberemos calcular tanto el error que se produce en el conjunto de aprendizaje (error de aprendizaje) como el error que se produce en el conjunto de validación (error de validación). El procedimiento consiste en ir presentando a la red los patrones del conjunto de aprendizaje (calculando el error de aprendizaje) e ir intercalando en determinadas

iteraciones los patrones del conjunto de validación (calculando el error de validación).

Al principio tanto el error de aprendizaje como el error de validación disminuirán, pero a partir de una determinada iteración el error de aprendizaje seguirá bajando mientras que el error de generalización empezará a subir, en este momento la red ya no está aprendiendo la relación sino que está memorizando la asociación. Si lo que buscamos es una buena capacidad de generalización debemos parar el entrenamiento en este momento.

Finalmente para calcular el error de generalización es necesario emplear un conjunto de test final no utilizado durante la fase de aprendizaje y medir el error que la red obtiene con él.

En nuestro proyecto hemos utilizado la validación cruzada. Nuestro criterio de parada ha sido que el error de validación aumente. La proporción de patrones de entrenamiento asignada a conjunto de aprendizaje, conjunto de validación y conjunto de test ha sido de 60%, 20% y 20% respectivamente.

3.6.2 Nuestra red neuronal

No existe una metodología clara para el diseño de la red neuronal que mejor se adecúe al problema a resolver, más bien existen una serie de recetas que en base a la experiencia han demostrado proporcionar buenos resultados. En general para la obtención de la red neuronal suele jugarse con varias arquitecturas hasta encontrar una que proporcione los resultados deseados o por lo menos se acerque a ellos.

En nuestro caso hemos optado por probar con diferentes arquitecturas hasta encontrar aquella cuyo error de validación fuera inferior a 0,1 o se acercara bastante, siempre escogiendo la que con idénticos resultados tuviera menos neuronas.

Para ello hemos utilizado el software NeuroSolutions. Se trata de una aplicación específica para el diseño y desarrollo de redes neuronales. Dispone de asistentes que te van guiando en el proceso indicándole los parámetros de la arquitectura, los parámetros de entrenamiento, los patrones de entrenamiento así como la condición de parada.

Hemos partido de un conjunto común de datos:

Parámetros red neuronal	Valor
Modelo de red neuronal	Perceptrón multicapa
Neuronas de entrada	Hemos utilizado 4 neuronas de entrada. Cada una de ellas recibe el valor RSSI de cada una

Sistema de posicionamiento en interiores

	de las redes WiFi que conforman el sistema. Los valores que pueden recibir se encuentran en el intervalo $[0, +1]$
Neuronas de salida	Hemos utilizado 2 neuronas de salida. Representan las coordenadas cartesianas de la posición que ocupa el dispositivo móvil en el entorno. Los valores que pueden devolver se encuentran en el intervalo $[-1, +1]$.
Datos de entrenamiento	Hemos proporcionado tres conjuntos de datos de entrenamiento: un conjunto de aprendizaje, un conjunto de validación y un conjunto de test. La proporción es 60%, 20%, 20%.
Regla de aprendizaje	Hemos utilizado la regla de aprendizaje Levenberg-Marquardt
Función de error utilizada	Error cuadrático medio
Condición de parada	El error de validación aumenta o el número de iteraciones máximas se ha alcanzado

Partiendo de estas premisas hemos ido probando diferente número de capas ocultas, diferentes configuraciones de capa oculta, diferentes funciones de activación y diferente número de iteraciones de terminación.

La configuración que mejor resultado nos ha dado ha sido utilizar una única capa oculta de 7 neuronas con función de activación sigmoidea mientras que la capa de salida ha utilizado la función tangente hiperbólica como función de activación. El número de iteraciones ha sido de 1000.

El error de aprendizaje ha sido inferior al 1% al igual que el error de validación. Al aumentar el número de neuronas de la capa oculta a partir de esta configuración o el número de iteraciones no suponía una mejora en el error de validación y apenas mejoraba algo en el error de aprendizaje.

Sistema de posicionamiento en interiores

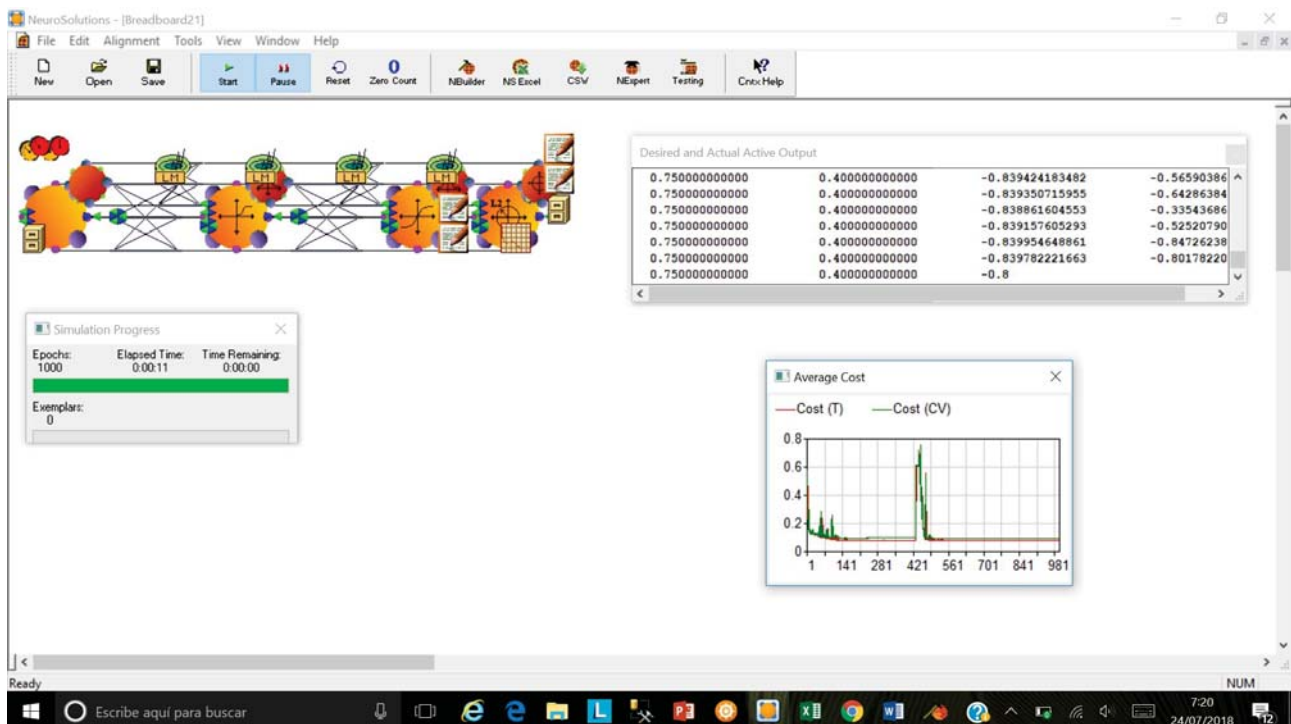


Figura 3.2: Resultado del entrenamiento de la red neuronal óptima

Patrones de entrenamiento utilizados

El conjunto de patrones que ha servido como entrenamiento a la red neuronal lo hemos obtenido de la siguiente manera:

1. Primeramente hemos dividido el entorno en cuadrículas de 90 centímetros cuadrados.
2. Luego establecimos distintos puntos donde realizar mediciones. Desde cada uno de ellos obtuvimos 20 mediciones del valor RSSI utilizando 4 orientaciones distintas. El valor se representa en unidades dbm.
3. Durante las mediciones sólo estuvo presente el usuario del sistema.
4. En aquellas cuadrículas donde había mobiliario no se realizaron medidas.

El siguiente gráfico muestra los puntos de medición del entorno así como aquellos puntos de los que por presencia de mobiliario no se pudo obtener una medición.



figura 3.12: Mediciones realizadas

Filtrado y preprocesamiento de los datos

Hemos realizado un pequeño filtrado en los datos eliminando aquellos que presentaban un valor fruto de una pérdida de la señal.

Además ha sido necesario un pequeño procesamiento de los datos para poder ajustarse al rango de valores permitido por las neuronas de entrada. En concreto hemos multiplicado por -1 los valores de intensidad y los hemos dividido entre 100 para tener el rango $[0,+1]$.

También hemos transformado los datos correspondientes a las posiciones para ajustarse al rango $[-1,+1]$. En concreto hemos realizado un escalado asociando el máximo valor del eje x (4 cuadrículas) al valor 1 y el menor valor de este eje (-4 cuadrículas) al valor -1. De igual forma se ha procedido con la neurona correspondiente al eje y siendo en este caso 2 y -2 cuadrículas los valores máximo y mínimo.

Red neuronal resultante

La red neuronal resultante fue la siguiente:

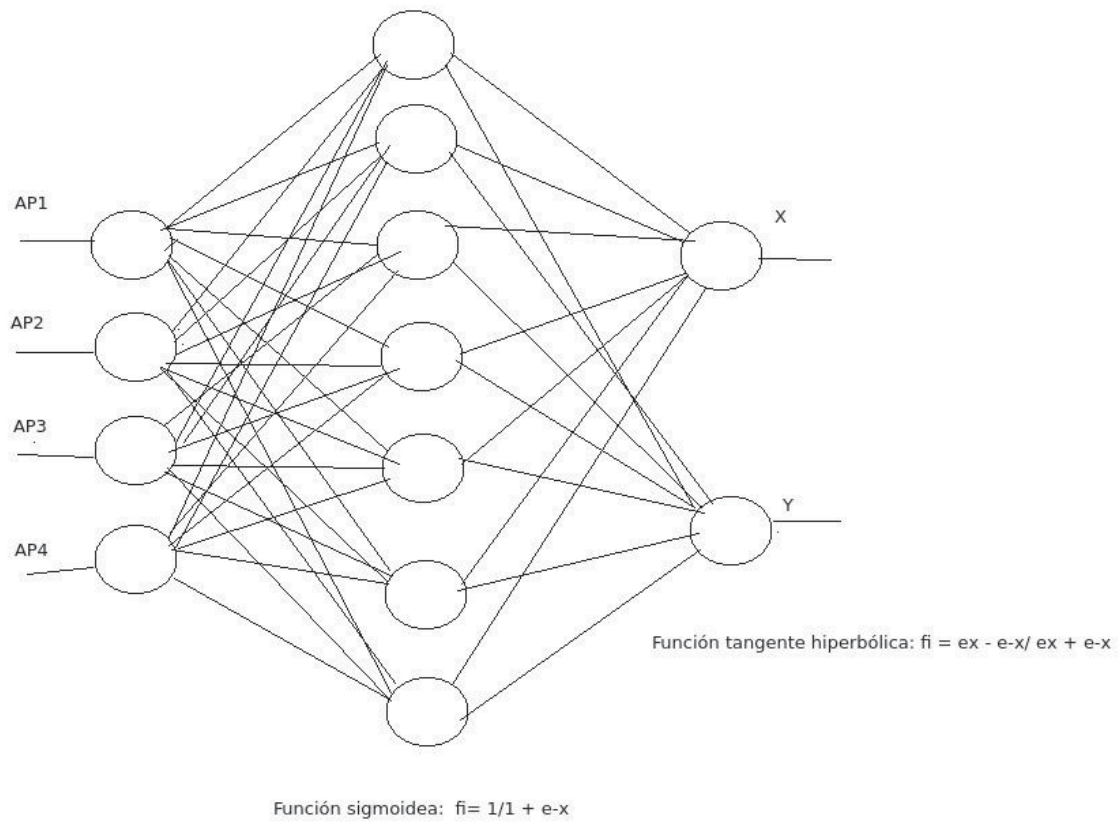


Figura 3.13: Red neuronal obtenida

4. Diseño del sistema

En este capítulo abordaremos el diseño del sistema.

4.1 Diagrama de bloques

La funcionalidad del sistema puede dividirse y/o describirse mediante un diagrama de bloques:



En él podemos distinguir claramente dos bloques o módulos: un módulo sensor que se encarga de obtener los valores RSSI de las redes WiFi y un módulo localizador que se encarga de obtener la posición que ocupa el dispositivo a partir de los datos suministrados por el módulo sensor.

Ambos bloques son totalmente independientes. La forma de uso del sistema es la siguiente: El módulo sensor devuelve los valores RSSI obtenidos tanto por pantalla como en un fichero. Posteriormente el usuario puede introducirlos manualmente en el módulo localizador o mediante el fichero generado por el módulo sensor.

El módulo sensor se ha desarrollado en lenguaje c++ bajo el sistema operativo Windows. El módulo localizador se ha desarrollado utilizando el software de terceros NeuroSolutions que permite la creación y uso de redes neuronales.

4.2 Módulo sensor

El módulo sensor se encarga de obtener los valores RSSI de las redes que conforman el sistema, adaptarlos a los requerimientos del módulo localizador y grabarlos en un fichero. Para ello se apoya en la Native WiFi API del sistema operativo Windows y en las técnicas de escaneo de redes inalámbricas.

Se trata de una pequeña aplicación escrita en lenguaje C++ cuyo esquema de funcionamiento de forma resumida es el siguiente:

- Enumera las interfaces de red disponibles mediante la primitiva `wlanEnumInterfaces(...)`.

- Filtra por aquellas interfaces que cumplan las siguientes condiciones:
 - La interfaz está preparada para usarse.
 - La interfaz opera en modo extensible station. De esta forma como se comentó en el capítulo anterior cada vez que realice un escaneo guardará los resultados en memoria y podrán ser recuperados.
 - El dominio de regulación de la interfaz incluye a España. De esta forma podrá realizar escaneos activos si aplicara.

Todas estas consultas se hacen a través de la primitiva `wlanQueryInterface(...)`.

- Si existe alguna interfaz que cumpla estos requisitos procede a realizar el escaneo mediante la primitiva `wlanScan(...)`. Como se comentó en el capítulo anterior se hace un escaneo de todas las redes accesibles por lo que el parámetro `pDot11Ssid` se introduce con valor `NULL`.
- Para recibir una notificación cuando el escaneo termine registra la recepción de la notificación de escaneo_terminado mediante la primitiva `wlanRegisterNotification(...)`.
- Una vez recibida y si ha ido bien llama a la primitiva `wlanGetBssNetworksList(...)` que devuelve los puntos de acceso correspondientes a las redes descubiertas en el escaneo.
- Filtra la lista devuelta por aquellos puntos de acceso que pertenezcan al sistema. Para ello utiliza el parámetro `SSID`.
- Una vez tenemos filtrados los puntos de acceso que conforman el sistema obtiene el valor del atributo `Irssi` de cada uno de ellos.
- Realiza un pequeño procesamiento de éstos para adaptarlos a los requerimientos del módulo localizador.
- Finalmente muestra los datos obtenidos por pantalla y los graba en un fichero `txt`.

4.3 Módulo localizador

El módulo localizador se encarga de obtener la posición que ocupa el dispositivo móvil en base a los datos aportados por el módulo sensor.

Se trata de una red neuronal con una capa de entrada de 4 neuronas, una capa oculta de 7 neuronas y una capa de salida de 2 neuronas. La capa de entrada recibe los valores RSSI obtenidos por el módulo sensor y devuelve las

Sistema de posicionamiento en interiores

coordenadas cartesianas de la posición que ocupa el dispositivo móvil en relación al entorno.

Para su desarrollo hemos utilizado el programa NeuroSolutions tal y como se explicó en el capítulo anterior. Esta aplicación permite usar las redes neuronales creadas mediante ficheros de test.

4.4 Fichero de test

El formato del fichero de test lo conforman una serie de registros que incorporan los valores RSSI de cada una de las redes WiFi así como la posición donde fueron obtenidos.

AP2	AP3	AP1	AP4	X	Y
Valor RSSI	Valor RSSI	Valor RSSI	Valor RSSI	Valor Eje X	Valor Eje y

5. Caso de prueba. Resultados

Se ha realizado un caso de prueba para comprobar el rendimiento del sistema.

$$d(\text{Posicion_estimada}, \text{posicion_real}) = \sqrt{(X_{\text{est}} - X_{\text{real}})^2 + (Y_{\text{est}} - Y_{\text{real}})^2}$$

Se mostrará la posición donde se realizaron y los resultados que devuelve el sistema. También se indicará el error de localización que ha tenido lugar mediante la distancia euclídea.

5.1 Caso de prueba 1

Descripción

Nombre	Caso de prueba 1
Descripción	Consiste en usar el sistema para localizar el dispositivo móvil en 10 puntos del entorno.
Tarjeta de red utilizada	Qualcomm atheros integrada
Condiciones de ejecución	Se realizará con la sola presencia en el entorno del usuario del sistema.
Objetivos	Comprobar el funcionamiento del sistema utilizando el hardware con el que ha sido entrenado.
Resultado	Por cada localización que lleve a cabo el sistema se indicará la posición donde se llevó a cabo, la posición que devolvió el sistema y el error de localización que hubo.

Tabla de resultados

Posición real		Posición estimada		Error de localización (metros)
x	y	x	y	
1,13	0,45	0,91	0,11	0,40
-2,24	0,45	0,85	-0,61	3,26
0,00	0,90	0,91	0,77	0,92
2,71	1,80	0,32	1,53	2,41
1,80	-0,45	0,38	1,46	2,38
-1,35	-0,90	0,69	-0,41	2,11
-0,44	0,45	-0,50	1,40	0,95
1,80	0,00	0,76	1,01	1,45

Sistema de posicionamiento en interiores

-1,80	1,80	0,44	1,40	2,27
-0,91	-0,90	0,35	0,61	1,96

6. Evaluación del sistema

En este apartado hemos evaluado el rendimiento del sistema en el caso de prueba realizado.

Para la evaluación del sistema utilizaremos tres medidas: Error de localización medio, error de localización el 90% de las veces y precisión en la medida del error.

Error de localización medio

Indica la media del error de localización del caso de prueba.

Error de localización el 90 % de las veces

Indica el mínimo límite máximo al que el error de localización no llega al menos el 90% de las veces.

Precisión

Indica la desviación estándar producida en la medida del error de localización.

6.1 Caso de prueba 1

Error de localización medio	1,81
Error de localización el 90% de las veces	<2,50
Precisión	0,87

Comentarios: Los resultados obtenidos muestran un error de localización medio de 1,81 metros. El 90% de las veces el error es inferior a 2,50 metros siendo el 50% de las veces inferior a 2 metros y el 30% de las veces inferior a 1 metro. Son resultados interesantes, quizás no sean viables para un sistema que necesite de una gran exactitud para su funcionamiento pero para la gran mayoría de sistemas y servicios basados en localización son datos más que aceptables.

7. Conclusiones y trabajo futuro

El uso de sistemas de posicionamiento como el GPS se ha convertido en algo muy cotidiano hoy en día gracias, en parte, a su implantación de serie en la mayoría de teléfonos móviles. Sin embargo en cuanto a localización en interiores no existe a día de hoy una tecnología que ofrezca las mismas prestaciones.

Durante el presente trabajo se ha pretendido analizar el rendimiento que puede ofrecer un sistema de posicionamiento en entornos interiores. Para ello se ha diseñado un sistema que utiliza la tecnología WiFi para las comunicaciones y una red neuronal como algoritmo de localización.

La lista de objetivos planteada inicialmente ha sido la siguiente:

- ✓ Realizar un estudio sobre las tecnologías y algoritmos que se utilizan en la actualidad en el desarrollo de estos sistemas. Comentar sus ventajas e inconvenientes y valorar su idoneidad en sistemas de localización.
- ✓ Estudiar el concepto de RSSI y los retos que presenta a la hora de desarrollar un sistema de estas características.
- ✓ Estudiar la arquitectura que ofrece el sistema operativo Windows para el manejo de redes WiFi.
- ✓ Estudiar las APIS que ofrece el sistema operativo Windows para el manejo de redes WiFi y la obtención del valor RSSI.
- ✓ Realizar la implementación de un pequeño módulo sensor capaz de obtener las intensidades de señal de redes WiFi.
- ✓ Estudiar la arquitectura y funcionamiento de las redes neuronales, en especial, del perceptrón multicapa.
- ✓ Diseñar una red neuronal con ayuda del software NeuroSolutions.
- ✓ Probar el sistema desarrollado.
- ✓ Evaluar el rendimiento del sistema.
- ✓ Extraer las conclusiones que se deriven del trabajo realizado.

Hemos realizado un estudio de las tecnologías y algoritmos que actualmente se utilizan en el desarrollo de estos sistemas. Muchos de ellos exigen la adquisición por parte del usuario de un hardware específico que dificulta su acceso a éstos. Por otra parte las características de los entornos interiores dificultan que exista línea de visión directa entre emisor y receptor por lo que

los algoritmos basados en cálculo de tiempos y distancias no son efectivos. Los algoritmos de análisis de la escena o fingerprinting ofrecen buenos resultados en entornos interiores puesto que tienen en cuenta el ruido de estos entornos y no necesitan línea de visión directa entre emisor y receptor.

La tecnología WiFi se ha erigido como la más usada en este tipo de sistemas. Su amplia difusión e implantación la convierten en una tecnología barata, accesible y reutilizable. Además proporciona de serie ventajas como puede ser un protocolo de control de acceso al medio.

WiFi proporciona un indicador llamado RSSI que indica la fuerza de señal de las tramas que se reciben. Se ha realizado un estudio de este indicador así como de los problemas que plantea en el desarrollo de sistemas de localización en interiores. Según el estándar IEEE 802.11 el parámetro RSSI es un entero que puede tomar valores comprendidos entre 1 y 255. Cada fabricante establece su propio rango de valores, su propia regla de conversión a dbm y su propio rango de sensibilidad. Esto ocasiona que diferentes dispositivos obtengan diferentes medidas en la misma posición.

Se ha realizado un estudio de la arquitectura Native 802.11 que proporciona a los sistemas operativos Windows un marco de trabajo para el desarrollo de drivers y servicios relacionados con las redes WiFi. Esta arquitectura mediante el componente de autoconfiguración (ACM) proporciona una API pública que permite gestionar las redes WiFi así como los perfiles de red. Esta API recibe el nombre de Native WiFi.

Mediante la citada API hemos desarrollado un pequeño módulo sensor que captura el RSSI de 4 redes WiFi. Se basa en el mecanismo de escaneo de redes que permite a una estación cliente detectar que redes se encuentran disponibles. Este procedimiento implica intercambio de tramas lo que permite poder obtener el RSSI que la tarjeta de red mide cada vez que el dispositivo recibe una. Destacan principalmente dos primitivas: wlanScan que permite realizar un escaneo de redes inalámbricas y wlanGetBssNetworksList que permite recuperar las redes detectadas en el escaneo.

Dentro de los algoritmos de fingerprinting las redes neuronales se plantean como un buen candidato para este tipo de sistemas. Son capaces de aprender casi cualquier mapping y ofrecen buenos resultados en condiciones de ruido. Además son muy rápidos en su fase de ejecución, algo muy interesante en sistemas que por su naturaleza necesitan buenos tiempos de respuesta.

Se ha realizado un estudio de la arquitectura y funcionamiento de varios modelos de redes neuronales en especial del perceptrón multicapa. Se ha

diseñado y entrenado este modelo mediante un software específico para estos propósitos. Actuará como nuestro algoritmo de localización recibiendo los valores RSSI medidos por el módulo sensor y devolviendo la posición que ocupa el dispositivo móvil en el entorno. Para entrenarla se han obtenido cerca de 600 muestras en diferentes posiciones del entorno. Cada muestra contiene los valores de RSSI medidos por el módulo sensor y la posición que ocupábamos durante esa medición.

Hemos realizado un caso de prueba que contenía el valor RSSI de 10 localizaciones para evaluar el rendimiento del sistema. Hemos evaluado el caso de prueba obteniendo un error medio de localización de 1,81 metros siendo además inferior a 2,50 metros el 90% de las veces. La desviación típica del error tuvo un valor de 0,87 metros.

De los resultados podemos extraer algunas conclusiones. Aunque en un principio me parecieron errores elevados teniendo en cuenta las medidas de la sala, recapacitando un poco creo que un error medio de 1,81 metros es más que suficiente para la gran mayoría de servicios basados en localización.

En cuanto a líneas de trabajo a seguir vistos los resultados obtenidos hay muchas. Podría ser interesante ver que rendimiento ofrece aumentar el número de puntos de acceso, en principio debería dar más exactitud al sistema aunque quizás el coste empieza a ser elevado. Por otra parte sería interesante probar el sistema con otras tarjetas de red con las que no se ha entrenado para ver cuánto influye el hardware y su independiente forma de medir RSSI. También sería interesante ver cómo influye la presencia de personas en el entorno sobre los valores de RSSI medidos.

Aunque en un primer momento realicé dos pequeños casos de prueba al respecto no los he tenido en cuenta en la memoria final por los pocos ejemplos con los que contaba. Sería interesante poder realizar estas pruebas en trabajos futuros.

8. Bibliografía y recursos web

Bibliografía

Bardwell, J. (2002). Converting Signal Strength Percentage to dBm Values. Wild Packets.

Cisco Connected Mobile Experiences (CMX) CVD. (2004). location-fundamentals.

Krishnamurthy, K. K. (2004). Properties of Indoor Received Signal Strength for WLAN Location Fingerprinting.

Received Signal Strength Indicator and Its Analysis in a Typical WLAN System . (2013).

Zhan, J. L. (2014). Characterization of Smart Phone Received Signal Strength Indication for WLAN Indoor Positioning Accuracy Improvement.

Bardwell, J. (2004). The Truth About 802.11 Signal And Noise Metrics. Connect802 Corporation.

Tsai, M. (2011). Path loss and shadowing.

Cisco Connected Mobile Experiences (CMX) CVD. (2014). Join Process—Association. Fundamentals

Lin, T.-N. (2005). Performance Comparison of Indoor Positioning Techniques based on Location Fingerprinting in Wireless Networks. International Conference on Wireless Networks, Communications and Mobile Computing.

Testing WLAN Devices According to IEEE 802.11 Standards. (2013). Agilent Technologies.

Shixiong Xia 1, Y. L. (2017). Indoor Fingerprint Positioning Based on Wi-Fi: An Overview. international journal of geo-information.

Molina, B. m. (2006). redes neuronales y sistemas borrosos. Rama.

A Discussion Clarifying Often-Misused 802.11 WLAN Terminologies. (2004). Connect802 Corporation.

Abusing the Windows WiFi native API to create a Covert Channel. (2011).

Hu, Y. (2015). A Visible Light-based Positioning System.

Javier Martín Moreno, D. R. (2007). Informe Técnico: Protocolo ZigBee (IEEE 802.15.4).

Battiti, R., Nhat, T. L., & Villani, A. (2002). LOCATION-AWARE COMPUTING: A NEURAL NETWORK MODEL FOR DETERMINING LOCATION IN WIRELESS LANS.

Dai, H., Ying, W.-h., & Xu, J. (2015). Multi-layer neural network for received signal strength-based indoor localisation. The Institution of Engineering and Technology.

Nowicki, M., & Wietrzykowski, J. (2017). Low-effort place recognition with WiFi fingerprints using deep learning. Institute of Control and Information Engineering.

Saleem, F., & Wyne, S. (2016). WLAN-BASED INDOOR LOCALIZATION USING NEURAL NETWORKS. de gruyter open.

Referencias web

Native 802.11 Wireless LAN Design Guide. Obtenido de <https://docs.microsoft.com/es-es/previous-versions/windows/hardware/wireless/native-802-11-wireless-lan4>

Native WiFi. Obtenido de <https://docs.microsoft.com/es-es/windows/desktop/NativeWiFi/portal>

