

LOCALIZACIÓN DE OBJETOS EN RANGOS CORTOS EN REAL-TIME USANDO SEÑALES DE RADIOFRECUENCIA

Johnny Bravo M. ⁽¹⁾
José Gómez G. ⁽²⁾
PhD. Daniel Ochoa D. ⁽³⁾

Facultad de Ingeniería en Electricidad y Computación (FIEC)
Ingeniero en Telemática ^{(1) (2)}

Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 vía Perimetral
Apartado 09-01-5863. Guayaquil-Ecuador

jofrbrav@espol.edu.ec ⁽¹⁾
joangomez@espol.edu.ec ⁽²⁾

Escuela Superior Politécnica del Litoral (ESPOL)
PhD en Ciencias Computacionales, Profesor de Materia de Graduación ⁽³⁾
dochoa@espol.edu.ec ⁽³⁾

Resumen

El objetivo de este trabajo fue definir las razones por las que se estudia e implementan sistemas de localización de objetos con dispositivos de transmisión y recepción de señales de radiofrecuencia y computadores que procesen estas señales, para luego estudiar los métodos matemáticos existentes de localización de objetos por trilateración, tecnología en dispositivos de transmisión, estándares de comunicación, programación en bajo nivel y en tiempo real acrítico disponibles y modelos de funcionamiento y diseño del sistema, escoger los más adecuados, y verificar la viabilidad de implementar un sistema para localizar un objeto en tiempo real en un ambiente cerrado pequeño con la aplicación de los mismos. Para ello se realizaron varias pruebas de localización de un objeto en un ambiente cerrado con las que se analizaron parámetros específicos como la precisión en la posición estimada del objeto y el tiempo de respuesta que brinda el sistema, los efectos negativos de los fenómenos físicos presentes en el ambiente sobre estos parámetros y como mitigarlos por medio de un análisis del margen de error en la posición estimada del objeto mediante un escalamiento del mismo y de técnicas de programación en real-time que optimicen el tiempo de respuesta del sistema, para obtener un sistema que pueda ser utilizado como un complemento al Sistema de Posicionamiento Global (GPS).

Palabras Claves: Sistema de localización, Trilateración, Tiempo de respuesta, Precisión, Tiempo real acrítico.

Abstract

The objective of this study was to define the reasons why study and implement systems that locate objects with transmission and reception devices with signals of radiofrequency and computers that process these signals, and then study the existing mathematical methods of locating objects by trilateration, technology in transmission devices, communication standards, low level and soft real time programming available, and models of operation and system design, choose the most suitable ones, and verify the feasibility of implementing a system to locate an object in real time in a small closed environment with the implementation of the concepts mentioned. In order to reach this objective there were several tests of locating an object in an closed environment that permit to analyze specific parameters such as the precision in the estimated position of the object and the response time provided by the system, the negative effects of the physical phenomena present in the environment on these parameters and how to mitigate them through an analysis of the margin of error in the estimated position of the object by an scaling of the margin of error and using techniques of real-time programming that optimize the response time of the system, to get a system that can be used as a complement to the Global Positioning System (GPS).

Keywords: Positioning system, Trilateration, Response time, Accuracy, Soft real time.

1. Introducción

Los sistemas de localización de objetos han sido estudiados por años y el método que más aplican es la trilateración que utiliza 3 puntos de referencia conocidos: $P1(x_1,y_1,z_1)$, $P2(x_2,y_2,z_2)$ y $P3(x_3,y_3,z_3)$, y distancias entre el objeto y cada punto: d_{10} , d_{20} , d_{30} , con los que se grafican los centros y radios de 3 esferas intersectadas en el punto $P:(x,y,z)$, como se ve en la figura 1.1, que genera un sistema de ecuaciones que al resolverse da la ubicación del objeto. [1]

$$d_{10}^2 = (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 \quad (1)$$

$$d_{20}^2 = (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 \quad (2)$$

$$d_{30}^2 = (x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 \quad (3)$$

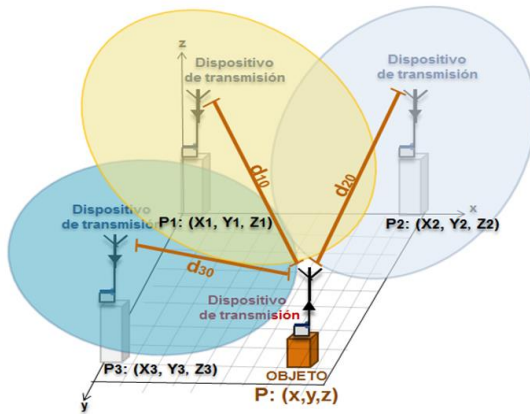


Figura 1.1. Localización de objeto por trilateración

El Sistema de Localización Global de objetos es el más utilizado en lugares abiertos por su margen de error promedio de la posición correcta de 20 metros, brindado por 24 satélites en puntos de referencia conocidos que reciben la señal de un transmisor en el objeto y miden su tiempo de propagación con la que se estima la distancia satélite-objeto a aplicarse en trilateración, sin embargo en lugares cerrados el error es superior a causa de fenómenos físicos que atenúan la señal, por lo que como solución se han implementado sistemas con transmisores ubicados en los puntos de referencia y computadores que procesan la potencia de la señal emitida para estimar la distancia punto-objeto a aplicarse en trilateración, sin embargo si el objeto se mueve rápidamente suelen darse lecturas imprecisas pues el tiempo de respuesta del sistema no es tan rápido para estimar la posición en tiempo real, es decir apenas el objeto se mueva. [2]

En este proyecto se utilizaron computadores con kernel Linux 2.4 que permite integrar tiempo real y representa los procesos como archivos simples con un número descriptor, con lenguaje C y la librería POSIX que presenta funciones de bajo nivel para implementar programas que permitan a los dispositivos transmitir o recibir señales a una red de computadores que las procesan y técnicas de programación en tiempo real que permitan estimar la posición del objeto. [3]

2. Exactitud en medidas experimentales.

Un parámetro a considerar en el diseño de un sistema de localización es la reducción de las pérdidas de la potencia “ P_{TIN} ” de señales electromagnéticas de frecuencia f y velocidad c , transmitidas en una o varias direcciones por dispositivos con antenas direccionales u omnidireccionales respectivamente, puesto que estas pérdidas implican pérdidas de datos, producidas por el material de fabricación de las antenas que es el metal y absorbe mucha potencia, por lo que la potencia entrante en cada antena se reduce a una potencia de salida, y estas pérdidas se cuantifican como ganancias $G_t = P_{TOUT}/P_{TIN}$ en la transmisora y $G_R = P_{ROUT}/P_{RIN}$ en la receptora; además que la receptora presenta una sensibilidad de recepción de señales que se define como su capacidad de captar toda la potencia de una señal transmitida, es decir todos sus datos, expresada en [Decibelios] y en signo negativo. [4]

De igual manera durante la propagación de la señal entre antenas, los objetos conductores provocan que se reflejen o difracten señales no deseadas a la receptora que generen pérdidas de potencia cuantificadas con la ecuación de Friss de una ganancia G , en términos de las ganancias en las antenas G_t y G_R , y distancia entre antenas D , como: $G = 10 \cdot \log [P_{RIN}/P_{TOUT}] = 10 \cdot \log [G_t \cdot G_R \cdot (c/4f \cdot D)^2]$, expresada en [Decibelios], que establece una relación inversa y cuadrática entre la potencia P_{RIN} y la distancia D a la que la señal no presenta pérdidas, como se ve en la figura 2.1. [5]

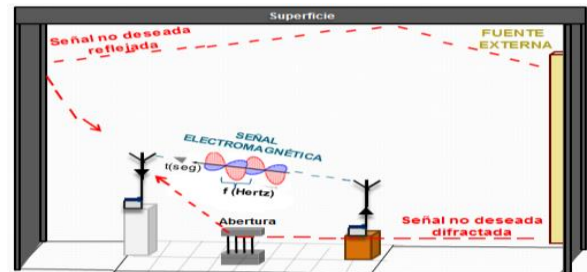


Figura 2.1. Pérdidas de potencia en el entorno

Estas pérdidas se mitigan mediante la ubicación de la receptora en una dirección de línea de vista en la que no se presenten objetos conductores, que en la práctica es en los extremos altos del lugar de instalación de las antenas, y con un análisis aplicable en comunicaciones de la incertidumbre de la P_{RIN} , denominado: teoría de propagación de errores, en la que P_{RIN} se considera una variable aleatoria gaussiana de media P_{RIN} y desviación σ_{PRIN} , y se puede calcular la incertidumbre de una variable dependiente de esta, como lo es D representada por su desviación σ_D en términos de la derivada de P_{RIN} como P_{RIN}' y desviación σ_{PRIN} , como se ve en la ecuación (4). [6]

$$\sigma_D = \sigma_{PRIN} / P_{RIN}' \quad (4)$$

3. Dispositivos de radio xbee.

Los dispositivos utilizados son los radios xbee, creados por la empresa estadounidense de soluciones tecnológicas inalámbricas Digi, los que presentan una antena omnidireccional que opera en el rango entre 2.400 y 2.480[GHz], como se ve en la tabla 3.1. [7]

S1 ESTÁNDAR	S1 PRO
Alcance: Con Línea de Vista: 90 m Sin Línea de Vista: 30 m	Alcance: Con Línea de Vista: 1.6 km Sin Línea de Vista: 90 m
Antena tipo Chip: Potencia (PRIN): 1[mW] Sensibilidad: - 92 [dBm]	Antena tipo Cable: Potencia (PRIN): 10[mW] Sensibilidad: - 98 [dBm]
Símbolo de S1:	Símbolo de S1-PRO:

Tabla 3.1. Datos Teóricos de los xbee S1

Estos radios transmiten o reciben señales entre ellos con el estándar IEEE 802.15.4, basado en la capa física y de enlace de datos del modelo OSI, que es una alternativa de bajo costo a los estándares IEEE 802.11 y Bluetooth, con una baja tasa de transmisión de datos de 250 kbps entre radios, y una repartición eficiente de los 80 [MHz] en 16 canales de 5 [MHz] numerados del 11 al 26, que no genera interferencia entre canales como el IEEE 802.11, como se ve en la figura 3.1. [8]

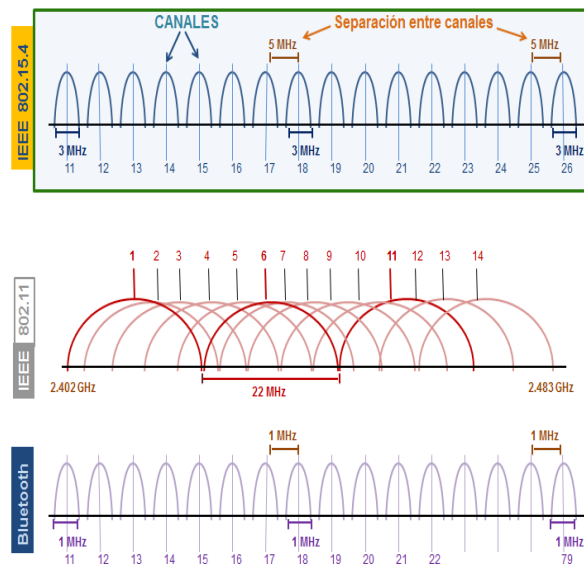


Figura 3.1. Distribución de canales en 802.15.4

En la capa de enlace de datos del xbee se establece cada xbee presenta una dirección propia MY, de destino DL-DH de su dato almacenado, e ID del grupo de xbee que este integra, de 4 bits hexadecimales. [8]

La capa física del xbee describe de su distribución de pines como se ve en la figura 3.2, en la que los pines más importantes son: el pin 2 de un buffer para ingresar un dato y transmitirlo a otro xbee, el pin 3 de un buffer para ingresar y ejecutar un comando enviado por un computador, y el pin 6 de Received Stenght Signal Intensity (RSSI) que reporta la RSSI del dato, definida como la relación entre la potencia recibida P_{RIN} real y la potencia recibida teórica en su antena.

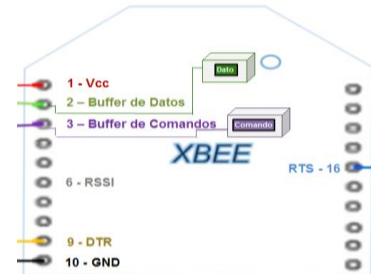


Figura 3.2. Estructura de los radios xbee S1

Un xbee no puede polarizarse, recibir o transmitir datos por sí mismo, por lo que la empresa Digi diseño la Xbee Explorer, que es una tarjeta que incorpora un microcontrolador que polariza al xbee al conectarse a un computador y le permite comunicarse vía USB al computador, como se ve en la figura 3.3.

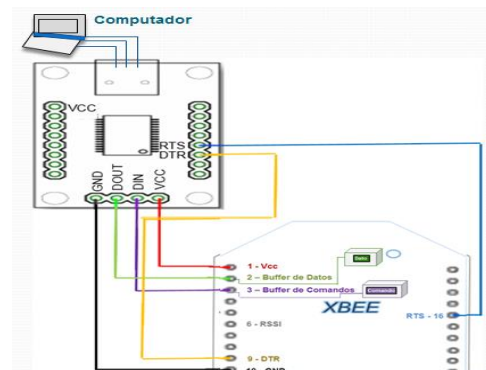


Figura 3.3. Tarjeta de Conexión Xbee Explorer

El computador ejecuta un programa escrito en lenguaje C que utiliza la librería <fcntl.h> para que el xbee pueda recibir un comando o transmitir un dato a otro, para lo que se accede al buffer con la función **int open (char* path, int oflag)**, que retorna su descriptor, en el que path es la ruta del puerto y oflag la constante que indica si quiere leer (O_RDONLY), ingresar (O_WRONLY) datos o ambas acciones (O_RDWR), y luego si desea ingresar un dato o comando en su buffer respectivo se utiliza la función **int write (int fd, void* buf, int n)**, que toma los primeros n bytes de la cadena buf y los almacena en el buffer de descriptor fd, o si se desea leer un dato o comando en su buffer respectivo se utiliza la función **int read (int fd, void* buf, int n)**, que lee los n bytes del dato del xbee de descriptor fd, que permite al xbee operar en uno de sus 2 modos. [9]

El primer modo de operación de un xbee es el de configuración en formato hexadecimal de las direcciones y canal de un xbee con el ingreso de un comando de la tabla 3.2 en su buffer de comandos, mediante la función write, que responde con una cadena de "OK" en este buffer.

Ingresar a modo comando	+++
- Asignar dirección origen	ATMY 3BF1
- Asignar dirección destino	ATDL 5FCC
- Asignar dirección del grupo	ATID 1234
- Asignar Canal	ATCH 10
- Tiempo de espera para ingreso a modo comando [mseg]	ATGT 3E8
- Obtener RSSI	ATDB
Salir de modo comando	ATCN

Tabla 3.2. Comandos de configuración de xbee S1

El otro modo es el de transmisión que está habilitado por defecto en un xbee y le permite transmitir un dato de su buffer de datos vía broadcast, hacia varios xbee con el mismo ID y canal CH, como se puede observar en la figura 3.4, o vía Unicast, hacia un xbee específico de dirección MY, que está asignada como dirección destino DL del transmisor, como se puede observar en la figura 3.5. [10]

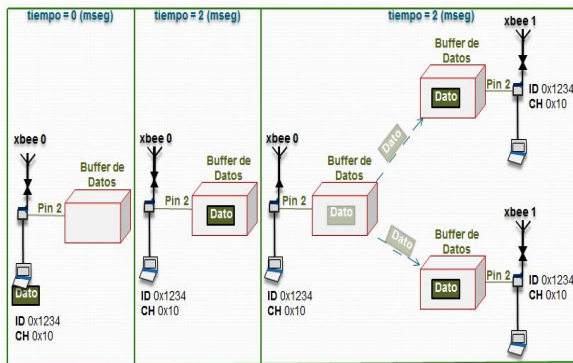


Figura 3.4. Modo de Transmisión por Broadcast.

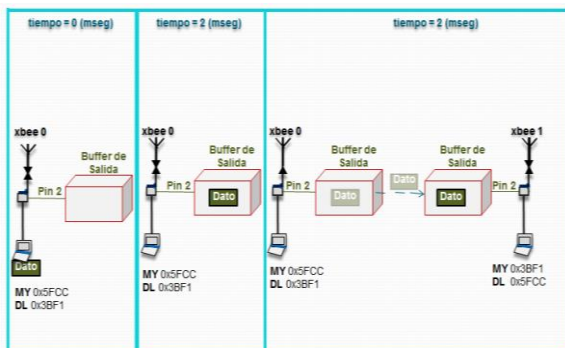


Figura 3.5. Modo de Transmisión por Unicast.

4. Comunicación entre procesos.

En un sistema de localización se necesita que un proceso cliente en un computador comunique datos, ya sea tiempo o potencia de las señales transmitidas, a un proceso servidor en otro computador, por medio de una estructura llamada: socket, constituida por una IP que identifica al computador en una red, un número de puerto que identifica al proceso y un protocolo de transporte que brinda garantía de llegada de los datos transmitidos, que puede ser: De Stream, en la que el cliente establece una conexión con el servidor que garantiza la transmisión de los datos en orden con un mensaje de confirmación ACK que implica un retardo, o De Datagram, en la que el cliente no establece la conexión de garantía, pero es más rápido que Stream.

En lenguaje C las librerías <sys/socket.h>, <sys/types.h> y <sys/time.h> permiten comunicar un proceso cliente con un servidor por sockets, como se ve en la figura 4.1, para lo que se obtiene el descriptor del socket en cada computador con la función **int socket (int dom, int tp, int prt)**, en el que dom define el protocolo de red IPv4 (AF_INET), tp el protocolo de transporte, ya sea SOCK_STREAM o SOCK_DGRAM, y prt un valor constante de cero; se crea una estructura sockaddr_in [family, address, port], en la que family define el protocolo de red IPv4 (AF_INET), port define el número de puerto del proceso en un computador, y address la dirección IPv4, parámetros que por medio de la función **int bind (int s_fd, sockaddr_in*serv, int ln)**, son asignados al socket servidor de descriptor s_fd y estructura serv, y queda a la espera de peticiones de conexión con un máximo de nc clientes con la función **int listen (int fd, int nc)**; un cliente de descriptor c_fd envía su petición de conexión por medio de la función **int connect (int c_fd, sockaddr_in *serv, int ln)**, y acepta la conexión al cliente de estructura clnt con **int accept (int s_fd, sockaddr_in*clnt, int ln)** y con **int select (int c_fd, fd_set* rwd, timeval* t)** se habilita al cliente para que comparta datos al servidor durante un tiempo t, por medio de las funciones read y write y después negarle la conexión, y se vuelve a la espera de peticiones; el servidor deja de escuchar peticiones con la función **int close (int s_fd)**. [11]

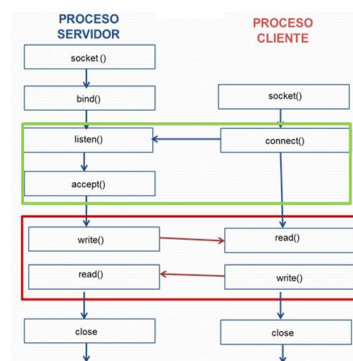


Figura 4.1. Comunicación por sockets TCP.

5. Respuesta en tiempo real acrítico.

Otro parámetro que se debe considerar en el diseño de un sistema de localización es su tiempo de respuesta, que se puede determinar que es predecible mediante su puesta en funcionamiento, pero debido al retardo de tiempo en la transmisión de señales de los radios producido por las ganancias de las antenas y los fenómenos físicos de difracción y reflexión en el ambiente este tiempo muchas veces no se cumple, sin embargo si se puede reducir este retardo mediante un análisis de escalamiento del margen de error en la posición estimada del objeto por medio de la teoría de propagación de errores vista en la sección 2.

Así mismo existe un retardo en el procesamiento de las señales en los computadores, por parte de sus procesadores (CPU), reducido mediante la gestión de memoria, dispositivos de entrada y salida, y CPUs, por parte del kernel del sistema bajo el concepto de **multiprocesamiento**, en el que varios CPUs comparten la memoria y dispositivos de entrada y salida del sistema, y cada CPU ejecuta varios procesos, lo que puede causar conflictos entre los procesos cuando acceden a la vez a un dato, para lo que se aplican variables semáforos que comunican y sincronizan procesos; además con el concepto de **multihilo**, se divide un proceso en instancias de este llamadas hilos, que cooperan en realizar sus instrucciones, y se crean con la función `int pthread_create (pthread_t*rt, pthread_attr_t*ra)` que retorna el descriptor de un hilo del proceso con atributos ra, sin embargo un hilo también puede entrar en conflicto con otro al acceder a un dato, para lo que se utilizan semáforos creados con la función `pthread_mutex_t PTHREAD_MUTEX_INITIALIZE` y por medio de la función `int pthread_mutex_lock (pthread_mutex_t* mut)` se bloquear temporalmente las acciones del semáforo mut de bloquear la ejecución de un hilo, como se ve en la figura 5.1. [12]

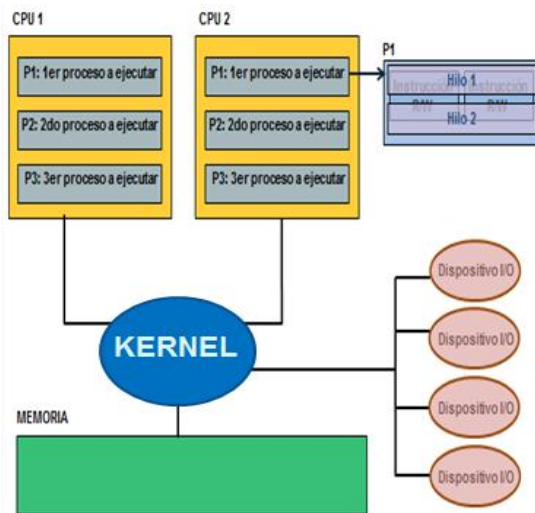


Figura 5.1. Gestión del kernel del sistema

Luego surgió el concepto de **tiempo real acrítico**, en el que el kernel utiliza algoritmos de planificación más eficientes en la repartición del CPU del sistema para priorizar la ejecución de ciertos procesos importantes sobre otros no tan importantes, lo que brinda un tiempo de respuesta predecible en el que puede aceptar un pequeño retardo, como en este caso permite que el tiempo de procesamiento de las señales se reduzca; para obtener tiempo real acrítico se debe instalar un sistema de kernel compatible con funciones de tiempo real, como en los sistemas Linux en los que se puede priorizar la ejecución de procesos llamados: procesos en tiempo real, que presentan mayor prioridad de ejecución que procesos normales que intervienen en sistemas con multiprocesamiento y multihilos, como se ve en la figura 5.2. [13]



Figura 5.2. Modo de Transmisión por Unicast.

Se han creado diversos algoritmos de planificación de ejecución eficiente de las instrucciones de los procesos o hilos en tiempo real para que el kernel del sistema priorice eficientemente su ejecución, entre los que los más utilizados son: First In, First Out (FIFO), en la que los procesos se ejecutan de acuerdo al tiempo en el que se cargan en la memoria del sistema, de modo que el primer proceso que se carga en memoria ejecuta primero todas sus instrucciones para luego ejecutar el siguiente cargado en memoria, sin embargo como el proceso obtiene el CPU hasta ejecutar todas sus instrucciones, esto puede provocar que procesos largos no importantes hagan esperar a procesos cortos importantes; el otro algoritmo de planificación es conocido como: Round Robin (R.R), en el que a cada proceso se asigna un tiempo máximo de uso del CPU denominado: quantum, tiempo que finalizado deja al proceso pendiente de la ejecución de sus instrucciones y se ejecuta el siguiente proceso.

Bajo el mismo principio de la planificación de los procesos de un sistema se puede priorizar la ejecución de un hilo sobre otros hilos, por medio de la función `void pthread_setschedparam (pthread_t th, int policy, sched_param* param)`, que asigna la planificación definida por la constante "param" que puede ser `SCHED_FIFO` o `SCHED_RR` para establecer al hilo thread, y policy definido por defecto como cero. [14]

6. Diseño e Implementación del Sistema.

Para diseñar un sistema de localización de objetos en tiempo real este aplica localización por trilateración para estimar la ubicación (x,y,z) de un objeto sobre el cual se encuentra un xbee, en el que para facilitar el análisis matemático del sistema y evitar en su mayoría el efecto negativo de los fenómenos físicos de difracción y reflexión, se considera que este objeto se desplaza únicamente en forma horizontal, por lo que la altura z a la que se encuentra no interesa, con lo que se asume las coordenadas z_1, z_2, z_3 como ceros, y se ubican los dispositivos de radio en puntos de referencia conocidos en los extremos más altos del lugar $P_1(0,0)$, $P_2(X_2,0)$ y $P_3(0,Y_3)$, con lo que se obtiene el siguiente sistema de ecuaciones:

$$d_{10}^2 = x^2 + y^2 + z^2 \quad (4)$$

$$d_{20}^2 = (x - x_2)^2 + y^2 + z^2 \quad (5)$$

$$d_{30}^2 = x^2 + (y - y_3)^2 + z^2 \quad (6)$$

Cada xbee fijo tiene conectada una computadora, que ejecuta un programa de transmisión ó recepción de datos, que obtienen las RSSI de los datos transmitidos entre el xbee sobre el objeto y cada fijo RSSI1, RSSI2, RSSI3 y se envían por sockets TCP a un computador central, que ejecuta un proceso que recibe las tres RSSI y genera un hilo que basado en cada RSSI halla la distancia del xbee sobre el objeto a cada xbee fijo d_{10} , d_{20} y d_{30} , y un semáforo sincroniza estas distancias para que se envíen a un proceso que estima la posición del objeto por trilateración, que con lo considerado inicialmente para trilateración calcula las coordenadas del objeto definidas como:

$$x = [d_{10}^2 - d_{20}^2 + x_2^2 / 2.x_2] \quad (7)$$

$$y = [d_{10}^2 - d_{30}^2 + y_3^2 / 2.y_3] - [x / y_3] \quad (8)$$

El sistema se puede implementar mediante un modelo de transmisor sobre el objeto, como se ve en la figura 6.1, en el que el computador conectado al xbee sobre el objeto ejecuta un programa de transmisión vía broadcast de un dato hacia cada xbee fijo, y el computador conectado a cada xbee fijo ejecuta un programa de recepción de este dato, que extrae su RSSI, y la envía a un computador que aplica los procesos de creación de hilos que estiman la distancia de cada xbee fijo al objeto en función de cada RSSI, y el proceso de trilateración que por medio de un semáforo recibe estas distancias y aplica trilateración con las ecuaciones (7) y (8); así mismo el sistema se puede implementar mediante un modelo de receptor sobre el objeto, como se ve en la figura 6.2, en el que el computador conectado a cada xbee fijo ejecuta un programa de transmisión vía unicast de un dato hacia el xbee sobre el objeto, y el computador conectado a este xbee ejecuta un programa de recepción de los datos, extrae sus RSSI y las envía al computador que de igual manera aplica los procesos de creación de hilos y de trilateración mencionados.

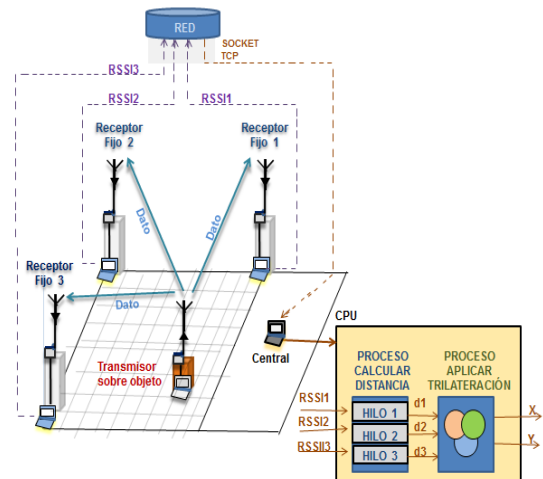


Figura 6.1. Modelo del transmisor sobre el objeto

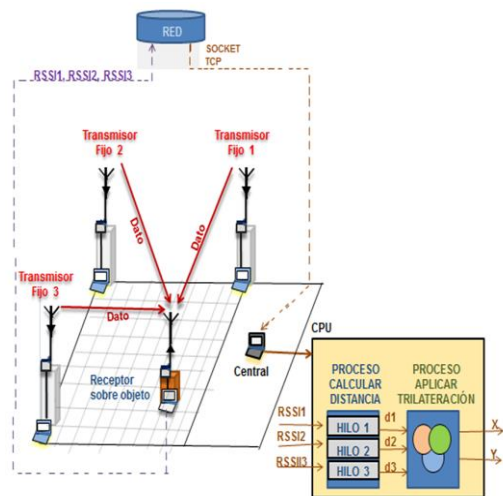


Figura 6.2. Modelo del receptor sobre el objeto

Los modelos de transmisor y receptor sobre el objeto utilizan un computador que ejecuta un programa central que genera hilos que calculan las distancias basados en las RSSI recibidas y con estas aplica trilateración; con el modelo del receptor sobre el objeto estas acciones pueden ser realizadas por el mismo programa receptor en el computador conectado al xbee sobre el objeto, lo que quita la necesidad de un computador central y la comunicación por sockets TCP, sin embargo las RSSI transmitidas por cada xbee fijo pueden interferirse entre ellas y para solucionar esto el programa receptor sobre el objeto sería más complejo en su implementación; con el modelo del transmisor sobre el objeto cada xbee fijo recibe una RSSI que se almacena en su respectivo computador, por lo que las RSSI no se interfieren entre sí, a pesar de que en este modelo si se necesita un computador central y comunicación por sockets TCP, sin embargo se hace más simple el desarrollo del programa receptor en cada xbee fijo, por lo que se escoge el modelo del transmisor sobre el objeto para implementar el sistema de localización en tiempo real del proyecto.

El sistema se implementó en el laboratorio del Centro de Visión y Robótica de la ESPOL que tiene un área de 5×10 [m²] y 5[m] de altura, para localizar una silla de 0.5 [m] de altura, en el que una portátil conectada a un xbee ubicado sobre la silla ejecuta un programa transmisor para que el xbee transmita un dato a tres xbee fijados en las esquinas altas del laboratorio en posiciones: F1(0,0), F2(0,10) y F3(5,0), como se ve en la figura 6.3, que tiene cada uno conectado un computador que ejecuta un programa receptor que extrae la RSSI del dato recibido y envía por socket TCP la RSSI a un programa central en una portátil, que recibe las RSSI y basadas en estas ejecuta hilos que calculan las distancias de cada xbee fijo al objeto y las aplica en trilateración para estimar la posición del objeto en un plano x versus y, como se ve en la figura 6.4, en la que se divide al laboratorio en cuadros de 1 metro de ancho y largo, para representar las posibles posiciones (x,y) del objeto.



Figura 6.3. Vista Panorámica del modelo en el CVR

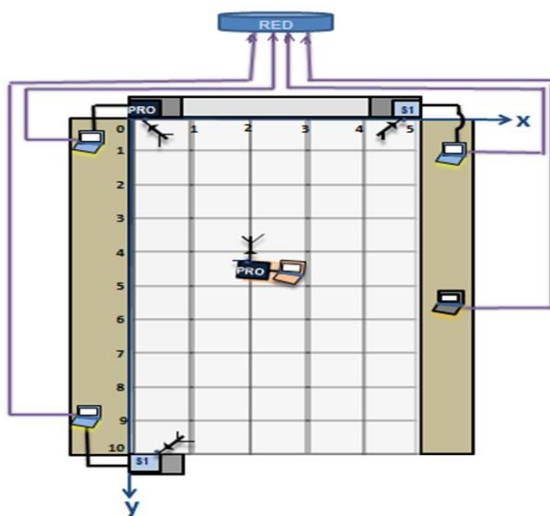


Figura 6.4. Vista Panorámica del modelo en el CVR

Los programas que permiten implementar el sistema de localización son cargados en los computadores de modo que el programa transmisor se carga en el computador conectado al xbee ubicado sobre el objeto, el programa receptor se carga en los computadores conectados a los xbee fijados en las esquinas altas del laboratorio y el programa central se carga en el computador que está en red con los computadores conectados a los xbee fijados en las esquinas altas del laboratorio.

El programa transmisor del sistema permite al xbee ubicado sobre el objeto almacenar un dato "Xi" cada 50 [mseg] en su buffer de datos, y que lo transmita automáticamente vía broadcast a los xbee fijados en las esquinas altas del laboratorio para recibir el dato, en la que $i = 1, 2, 3, \text{NUM_DATOS}$, y la constante NUM_DATOS definida en el programa establece la cantidad máxima de datos que puede transmitir un radio xbee conectado al computador que tiene implementado este programa, como se puede observar en la figura 6.5.

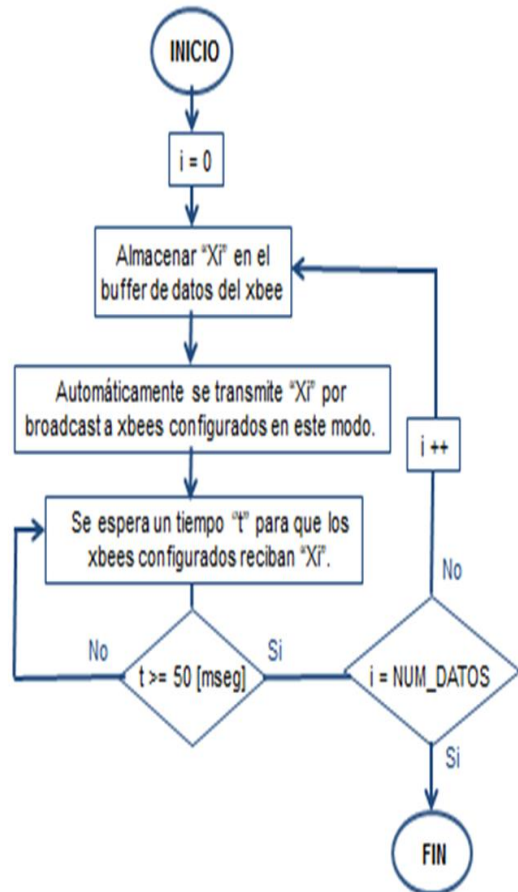


Figura 6.5. Flujoograma de programa transmisor

El programa receptor del sistema permite a cada uno de los xbee fijados en los extremos altos del laboratorio que si no presenta un dato X_i en su buffer de datos, se quede en un modo de espera por un dato, donde X_i es el dato enviado por el programa transmisor, y por otro lado si hay un dato X_i en el buffer de datos de un xbee fijo receptor, el programa lee el dato por medio de la función read, en el buffer de comandos ingresa “+++” para ingresar a modo de configuración por comandos y en el mismo ingresa el comando ATDB para extraer la RSSI del dato X_i , luego ingresa el comando ATCN para volver al modo de transmisión, y luego se envía esta RSSI de X_i por sockets TCP a un computador central en red con los computadores conectados a los receptores, todas estas acciones en 50 [mseg], como se ve en la figura 6.6.

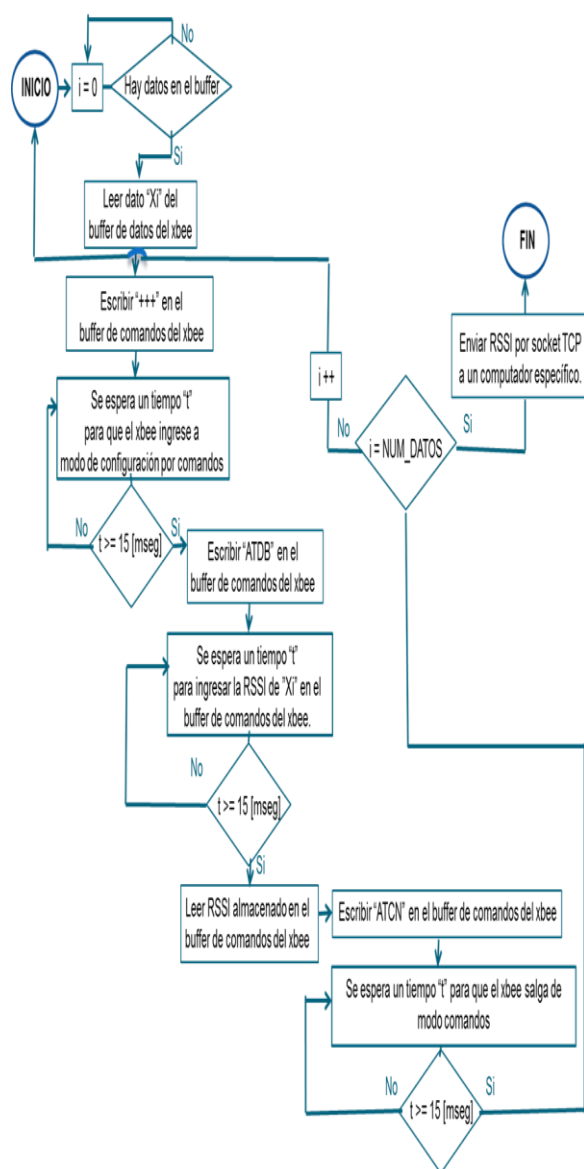


Figura 6.6. Flujograma de programa receptor

Finalmente el programa central del sistema permite al computador en red con los computadores conectados a los xbee fijados en los extremos altos del laboratorio, esperar por recibir las tres RSSI provenientes de los xbee fijados, por medio de sockets TCP, para luego generar y sincronizar tres hilos por medio de un semáforo, bajo planificación en tiempo real Round Robin, que calculen las distancias de cada xbee fijo al objeto, y aplicar trilateración para estimar la posición del objeto, como se ve en la figura 6.7.

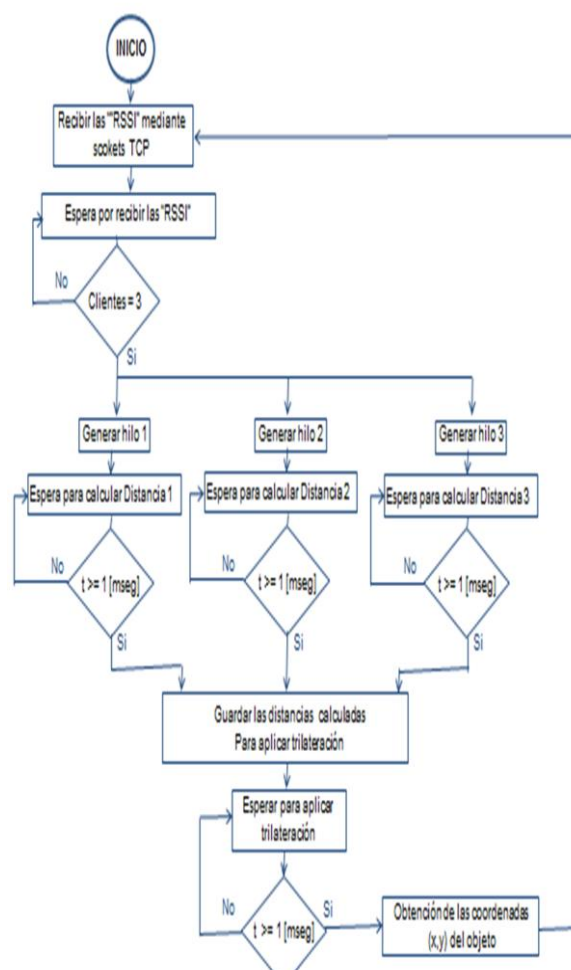


Figura 6.7. Flujograma de programa central

Para implementar el sistema se realizó una prueba que determina la relación Distancia versus RSSI que utiliza cada hilo para aplicar en trilateración para localizar el objeto, y una prueba de localización, en la que se estima la posición de un objeto, que no se espera que sea precisa, pues el laboratorio presenta varios objetos metálicos que producen muchas pérdidas de potencia que afectan a las distancias utilizadas en trilateración, pero si se esperó que con los hilos y planificación en tiempo real Round Robin implementados en el programa central el sistema, el sistema pueda funcionar lo suficientemente rápido, para realizar un análisis de corrección de errores en un futuro proyecto que mejore la precisión del sistema.

7. Prueba de Distancia versus RSSI.

El xbee S1-PRO sobre la silla transmite una trama de datos $[X0, X1]$ hacia un xbee fijo S1-PRO ubicado a una distancia $D1$ que varía desde 20[cm] hasta 5[m]; este proceso se repite 10 veces, y el programa receptor en el xbee fijo S1-PRO obtiene la RSSI promedio de los datos para cada distancia $D1$, para generar una gráfica de RSSI versus Distancia: $RSSI1 = f(D1)$, como se ve en la figura 7.1, y luego se realiza este procedimiento con un xbee fijo S1 ubicado a una distancia $D2$ similar a $D1$, para generar una gráfica: $RSSI2 = f(D2)$, como se ve en la figura 7.2, y con la teoría de propagación de errores, obtener la incertidumbre $SD1$ y $SD2$ en el valor de las distancias $D1$ y $D2$ respectivamente generado en el laboratorio en términos de las incertidumbres $SRRSSI1$ y $SRSSI2$ de las $RSSI1$ y $RSSI2$ respectivamente, que es útil para un análisis de corrección de errores en posición a realizarse en un futuro proyecto.

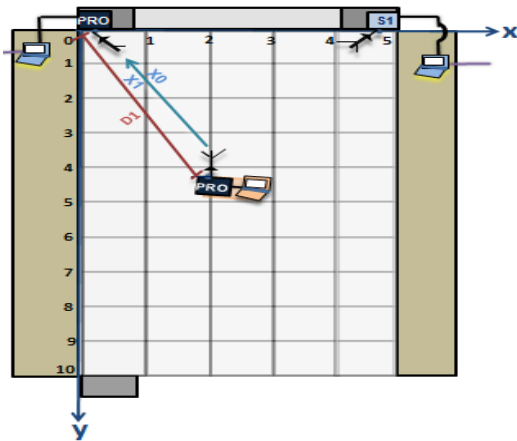


Figura 7.1. Prueba de transmisión a xbee fijo

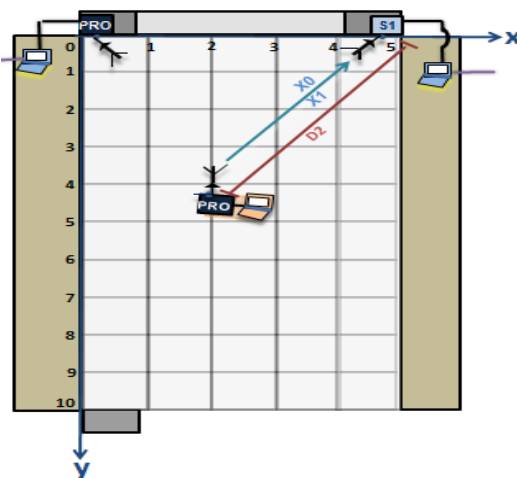
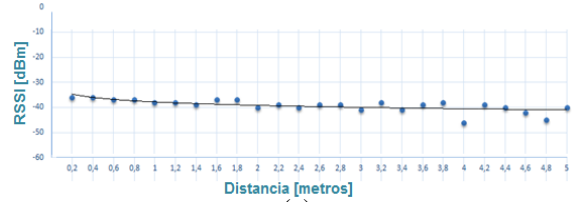
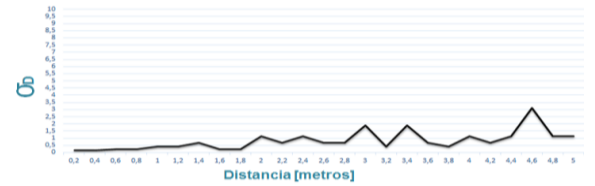


Figura 7.2. Prueba de transmisión a xbee fijo

Primero se obtuvo como resultado la figura 7.2a $RSSI1 = f(D1) = -1.975 \cdot \ln(D1) - 37.837$, y la figura 7.2b: $SD1 = f(D1) = SRSSI1(D1) / RSSI1'(D1)$, donde $SRSSI1(D1)$ y $RSSI1'(D1)$ es la incertidumbre de $RSSI1$ y derivada de $RSSI1$ evaluada en $D1$.



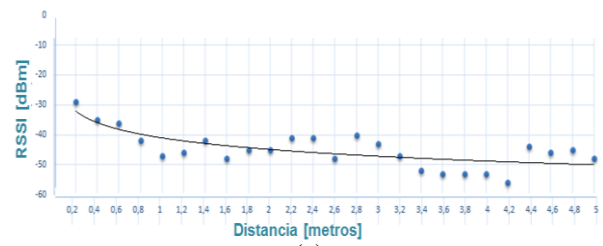
(a)



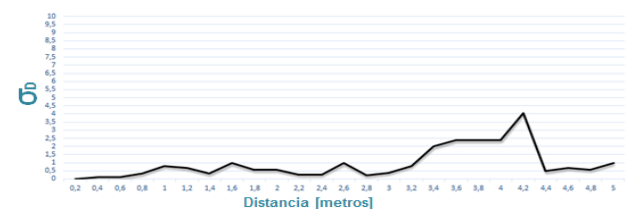
(b)

Figura 7.2. Resultado de prueba xbee fijo S1-PRO

Luego se obtuvo como resultado la figura 7.3a $RSSI2 = f(D2) = -5.579 \cdot \ln(D2) - 41.035$, y la figura 7.3b: $SD2 = f(D2) = SRSSI2(D2) / RSSI2'(D2)$, donde $SRSSI2(D2)$ y $RSSI2'$ es la incertidumbre de $RSSI2$ y derivada de $RSSI2$ evaluada en $D2$.



(a)



(b)

Figura 7.3. Resultado de prueba xbee fijo S1

En las gráficas 7.2b y 7.3b se puede ver que con un xbee S1 como xbee fijo, la RSSI presenta un error en la posición mayor que con un S1-PRO, lo cual ocurre por efecto de la difracción y reflexión causados por los objetos metálicos del laboratorio, pero también debido a la sensibilidad de recepción de las antenas de los xbee, que en el xbee S1 es menor, por lo que tiende a recibir menos datos que un S1-PRO, y a distorsionarse más los valores de RSSI obtenidos.

8. Prueba de localización de objeto.

Se implementa el sistema de localización completo con el modelo de transmisor sobre el objeto y funciona automáticamente de forma que el xbee sobre la silla transmite “NUM_DATOS” datos X_i hacia los 3 xbee receptores, fijados en los extremos altos del laboratorio, que extraen la RSSI promedio de los datos X_i recibidos y la envía por sockets TCP a la computadora central en la misma red, que ejecuta 3 hilos a la vez que aplican las relaciones $D1 = f(RSSI1)$ y $D2 = f(RSSI2)$ para calcular las distancias de cada xbee fijo al objeto y aplicar trilateración para hallar la posición (x,y) del objeto, como se ve en la figura 8.1.

Para esta prueba primero se establece en el programa transmisor y receptor a la constante NUM_DATOS con el valor de 1 y se ubica la silla en la posición $(2,4)$ para enviar un solo dato $[X_0]$ y determinar el tiempo de respuesta y precisión del sistema por el programa receptor y central que presentan los tiempos de realización de los procesos para estimar la posición; así mismo luego se establece en el programa transmisor y receptor a la constante NUM_DATOS con el valor de 10 y se ubica la silla en la misma posición para enviar una trama de datos $[X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9]$ y de igual forma con el programa receptor y central determinar el tiempo de respuesta y precisión del sistema, y verificar como este cambia respecto al envío de un solo dato.

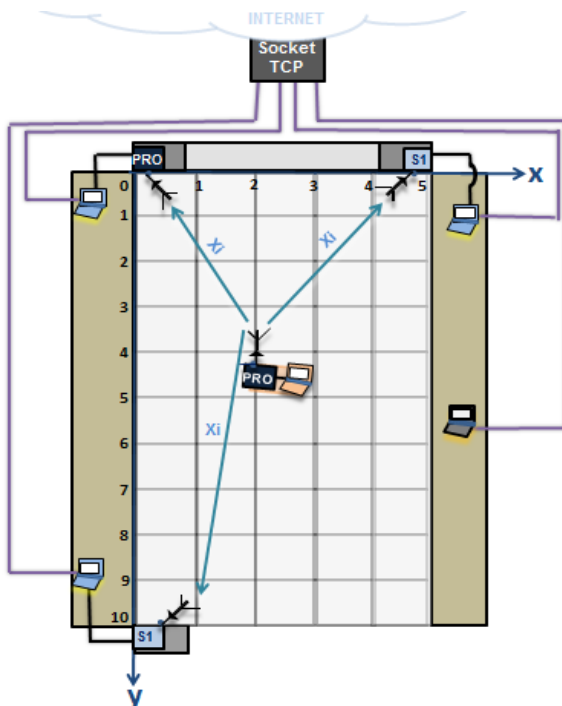


Figura 8.1. Esquema de localización de objeto

El tiempo de respuesta del sistema con la transmisión de 1 dato es de 70 [mseg], y el programa central determinó que la silla está ubicada en la posición $(5,5)$ lo que implica una precisión con error de 3[metros] y 1[metro] en las coordenadas (x,y) .

Tabla 8.1. Comandos de configuración de xbee S1

Tiempos de Respuesta Promedio del Programa Transmisor	
Ingreso de dato	Envío del dato por broadcast
2 [mseg]	1 [mseg]

Tiempos de Respuesta Promedio del Programa Receptor			
Lectura del dato	Ingresar a modo comando	Obtener la RSSI	Salir del modo comando
2 [mseg]	15 [mseg]	15 [mseg]	15 [mseg]

Tiempo de Respuesta Promedio del Programa Central		
Comunicación por Socket TCP	Cálculo de distancia por hilo	Trilateración
10 [mseg]	1 [mseg]	2 [mseg]

El tiempo de respuesta del sistema con la transmisión de 10 datos es de 513[mseg], y el programa central determinó que la silla está ubicada en la posición $(3,4)$ que implica una precisión con error de 1[metro] y 0[metros] en coordenadas (x,y) .

Tabla 8.2. Comandos de configuración de xbee S1

Tiempos de Respuesta Promedio del Programa Transmisor	
Ingreso de dato	Envío del dato por broadcast
2 [mseg]	1 [mseg]

Tiempos de Respuesta Promedio del Programa Receptor			
Lectura del dato	Ingresar a modo comando	Obtener la RSSI	Salir del modo comando
2 [mseg]	15 [mseg]	15 [mseg]	15 [mseg]

Tiempo de Respuesta Promedio del Programa Central		
Comunicación por Socket TCP	Cálculo de distancia por hilo	Trilateración
10 [mseg]	1 [mseg]	2 [mseg]

9. Conclusiones.

Es posible utilizar a los xbee para implementar un sistema de localización en el laboratorio del CVR, puesto que los xbee S1 y S1-PRO presentan alcances lo suficientemente grandes de 30 y 100 [metros] respectivamente para abarcar el laboratorio, y tiempos de respuesta cortos de [mseg] como se vieron en la prueba de localización de la sección 8 que permiten obtener un tiempo de respuesta rápido, y por medio de sincronización de hilos y programación en soft real time, específicamente con planificación de hilos round robin, se obtuvo un tiempo de respuesta predecible.

La precisión en la prueba de localización no fue tan regular y se quería concluir que la precisión del sistema era directamente proporcional a la cantidad de datos transmitidos entre los xbee, debido a que una gran cantidad de datos disminuye la probabilidad de que la pérdida de un dato afecte a la RSSI promedio y a la ubicación, y esto se debe a la presencia de varios objetos metálicos que bloquean la línea de vista de las señales emitidas y a la sensibilidad de recepción de los xbee que disminuye su capacidad de captar toda la potencia de las señales transmitidas y que esto ocasiona que se altere la posición estimada del objeto, sin embargo el proyecto estableció la posibilidad de mejorar la precisión mediante un análisis más extenso de las incertidumbres $SD = f(D)$ obtenidas en la prueba de RSSI versus Distancia de la sección 7, y la incorporación de dispositivos de GPS en las posiciones de referencia conocidas de los receptores.

10. Referencias

- [1] Wikipedia.org - Estados Unidos, (2013, Marzo), “Trilateración”, Localización de Objetos [Online], Disponible en: www.wikipedia.org/wiki/Trilateración
- [2] Blauden Electronics - España, (2008, Febrero), “¿Qué es el GPS?”, Soluciones de conectividad y movilidad [Online], Disponible en: www.zonagps.com/que-es-el-gps/
- [3] Univ. de Zaragoza - España, (2012, Mayo), “Sistemas de tiempo real”, Definición y Clases de sistemas de tiempo real [Online], Disponible en: www.webdiis.unizar.es/joseluis/STR.pdf
- [4] Univ. de Córdoba - Argentina, (2013, Mayo), “Antenas de Comunicación”, Tipos de Antenas [Online], Disponible en: www.cordobawireless.net/portal/descargas/Conceptos_basicos_sobre_antenas.pdf
- [5] Solred Radiodifusión - Argentina, (2008, Octubre), “Propagación de Ondas de Radio”, Propagación por Línea Visual [Online], Disponible en: www.solred.com.ar/lu6etj/tecnicos/handbook/propagacion/propagacion.htm
- [6] Wikipedia.org - Estados Unidos, (2013, Junio), “Propagación de Errores”, Error en combinaciones No Lineales [Online], Disponible en: www.wikipedia.org/wiki/Propagación_de_errores
- [7] Univ. de Colima - México, (2001, Mayo), “Modelo OSI”, Las 7 capas del modelo OSI y sus funciones principales [Online], Disponible en: www.docente.ucol.mx/al950441/public_html/osi1hec_B.htm
- [8] Libelium: Gascón - España, (2010, Septiembre), “Dispositivos Inalámbricos, ¿Como conviven? ”, Banda de 2.4 Ghz [Online], Disponible en: www.blogs.heraldo.es/ciencia/?p=1417
- [9] Blogspot Ingeniería Aplicada al Área de la Mecatrónica: Unal - Colombia, (2012, Julio), “Modulos xbee”, Configuración de módulos xbee [Online], Disponible en: www.alvarounal.blogspot.com/2011/12/modulos-xbee-parte-6-configuraciony.html
- [10] Spark Fun Inc. - Estados Unidos, (2012, Mayo), “Modos de operación de un xbee”, Funcionamiento de xbee [Online], Disponible en: www.sparkfun.com/products/8664
- [11] HispaLinux - España, (2007, Junio), “Sockets”, Programación-sockets [Online], Disponible en: www.tldp.org/Tutoriales/prog-sockets
- [12] Escuela Superior de Ingenieros de Bilbao: Dpto. de Sistemas - España, (2012, Mayo), “Sistemas Operativos”, Historia de los Sistemas Operativos [Online], Disponible en: www.disa.bi.ehu.es/spanish/asignaturas/ii/3_SO.pdf
- [13] YoLinux - Estados Unidos, (2013, Junio), “POSIX Threads”, POSIX Thread Libraries [Online], Disponible en: www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html
- [14] Univ. de Pennsylvania - Estados Unidos, (2006, Junio), “Real Time Scheduling”, Soft Temporal Constraints [Online], Disponible en: www.informatica.uv.es/it3guia/ARS/practicas/Funciones.pdf