

Arquitecturas para la adquisición y distribución de la información de localización en sistemas de seguimiento en interiores basados en UWB

Trabajo Fin de Máster

Máster en Tecnologías de la Información y Comunicaciones en Redes Móviles

Programa Oficial de Posgrado en Ingeniería de Telecomunicación

Curso 2009/2010

Septiembre 2010

Juan Chóliz Muniesa

Director: Ángela Hernández Solana

Departamento Ingeniería Electrónica y Comunicaciones
Ingeniería Superior de Telecomunicación
Centro Politécnico Superior



Universidad de Zaragoza



Ingeniería de Telecomunicaciones



Centro Politécnico Superior

Anexos

ANEXO A Descripción del simulador

A.1 Introducción

El principal objetivo del simulador es evaluar las prestaciones que un sistema de localización UWB real puede proporcionar en entornos interiores. De cara a que esta evaluación sea realista, se deben utilizar las especificaciones tanto a nivel físico como de acceso a la red (topología de la red, estructura de la supertrama, duración de los slots) de un sistema UWB real, en concreto del diseñado dentro del propio proyecto EUWB, además de un modelo de estimación de distancias en interiores. Sobre esta herramienta se analizarán diversas alternativas de diseño tales como la arquitectura del sistema de localización, posibles estrategias para la obtención y distribución de la información de localización, métodos de selección de los anchors utilizados para posicionar a un usuario, algoritmos de posicionamiento, etc.

A.2 Escenario

El escenario que se plantea es el de seguimiento de dispositivos móviles en entornos interiores relativamente amplios tales como centros comerciales, aeropuertos, estaciones, centros de congresos, etc. Se trata de escenarios relativamente grandes por lo que se deberán cubrir por varias estaciones fijas. La gestión de recursos de la red es crítica ya que serán usados por un elevado número de usuarios todos ellos con movilidad. Este escenario representado en el simulador tiene unas dimensiones *longitud* x *anchura* cuyo valor será seleccionado en el fichero de entrada del simulador.

Sobre ese escenario se despliega una picocelda UWB con un conjunto de nodos fijos dispuestos en cuadrícula separados por una distancia d . Por simplicidad se considera una única picocelda, con lo que no será necesario realizar handover intercelular. La picocelda dispone de N_a anchors, elementos fijos y de posición conocida, y N_m targets, elementos móviles a localizar.

Uno de los elementos fijos será el coordinador de la picocelda, que marca la sincronización de la supertrama y gestiona el acceso de los nodos a los recursos. El resto de nodos serán "esclavos" y pueden tener conexión directa con el coordinador (si están en cobertura) o a través de otros nodos hasta un número máximo de saltos N_{hops} .

En la picocelda se definen uno o más controladores de localización (LC), que realizan la función del cálculo de la posición de los elementos a localizar. Según la estrategia que se defina, esta función puede localizarse físicamente en uno o más nodos fijos (estrategia centralizada en la red), en los nodos móviles (estrategia centralizada en el móvil) o ser asumida dinámicamente por cualquier nodo de la red (estrategia distribuida).

A.3 Modelos implementados

Antes de analizar el simulador en detalle, se describen algunos de los modelos implementados.

A.3.1 Topología de la red

La topología de la picocelda es mallada-centralizada. El coordinador de la picocelda transmite tramas piloto (beacon frames) para la sincronización. Después se construye el árbol de distribución y se utiliza para transportar las tramas piloto y los comandos, que son enviados desde el coordinador a cualquier nodo de la picocelda. Finalmente, la topología se extiende a un árbol mallado (meshed tree), permitiendo la transmisión fuera del árbol de tramas de datos y ranging, así como de las tramas de 'hello' que son enviadas periódicamente a todos los vecinos.

A.3.2 Estructura de la supertrama MAC

La estructura de la supertrama define la temporización de slots utilizados para cada propósito (descubrimiento de vecinos, comunicación, ranging...). La supertrama MAC se divide en slots temporales (timeslots) en los que se envían las distintas tramas (tramas piloto, tramas hello, tramas de datos, tramas de ranging, solicitudes de slots...).

La supertrama MAC se divide en los siguientes periodos:

- Beacon period, utilizado para alineamiento.
- Topology management period, usados para mandar las tramas hello.
- CFP (Contention Free Period), compuesto por slots GTS (Guaranteed Time Slots), que pueden utilizarse tanto para transmisión de datos como para ranging y un GTS Request Period, usado para solicitar la asignación de slots GTS.
- CAP (Contention Access Period), que se utilizaría para enviar comandos desde los nodos al coordinador de la celda.

A.3.3 Arquitecturas del sistema de localización

El simulador tiene implementados varias arquitecturas del sistema de localización, que serán seleccionados desde el fichero de entrada.

- Centralizada en la red. Uno o varios nodos fijos de la red actúan como controladores de localización. Esos nodos están prefijados.

- Distribuida. Cualquier nodo anchor puede asumir el papel de controlador de localización para un target dado y calcular el dato de posicionamiento. El anchor lo elige de forma dinámica el móvil.
- Centralizada en el móvil. La función de controlador de localización está implementada en los propios nodos target, por lo que es el propio móvil el que calcula su posición.

A.3.4 Modelo dinámico de los targets

Para simular el movimiento de los targets se define una velocidad máxima y mínima en el fichero de entrada, y el simulador elige una velocidad aleatoria en ese rango, además de una dirección aleatoria. Pasado un intervalo de tiempo también definido en el fichero de entrada, se fijan una nueva dirección y velocidad aleatorias.

A.3.5 Modelo de ranging

Este modelo determina la distribución del error en la estimación de las distancias y se compone de un determinado sesgo que es función de la distancia real y de la probabilidad de tener un determinado canal LOS/NLOS/NLOS2 (que a su vez también es función de la distancia), y de un determinado nivel de error residual, que está relacionado con la resolución en la estimación del TOA y es independiente de la distancia. La distribución está definida en el simulador en el fichero ranging_model, salvo el error residual que se añade posteriormente y cuya varianza se puede seleccionar en el fichero de parámetros.

A.3.6 Modelo de cobertura

El modelo de cobertura se utiliza para identificar los anchors “vecinos” de cada target con los que realizará el ranging. También para identificar las relaciones de vecindad entre los anchors y determinar el número de saltos necesarios para llegar al controlador desde cada nodo. A falta de mayor información sobre el comportamiento de los nodos se ha implementado un modelo básico, si la distancia es menor a 15 metros se considera en cobertura y si está más lejos no.

A.3.7 Algoritmo de seguimiento

En el simulador están implementados dos algoritmos de seguimiento. La trilateración es un algoritmo no paramétrico que calcula la posición en base a la distancia estimada a tres anchors realizando cálculos geométricos. Por tanto cada posición se calcula independientemente sin tener en cuenta la posición anterior. El filtro de Kalman es un método paramétrico que permite combinar tanto la información de la dinámica del target (como su posición y velocidad

anteriores) con la de múltiples observaciones, por lo que permite utilizar la distancia estimada a un número ilimitado de anchors. Para ello es necesario conocer ciertos parámetros tanto del modelo dinámico del móvil como del modelo de error en las estimaciones.

A.3.8 Modos de funcionamiento de ranging

La adquisición de la información de localización se hace a través del procedimiento de ranging entre los que el simulador implementa dos modos de funcionamiento.

- Two way ranging: El iniciador del procedimiento (target o anchor) transmite un ranging request a otro nodo, que estima el tiempo de llegada y envía un ranging response después de un tiempo predefinido. El iniciador mide el tiempo de llegada de la respuesta y puede estimar el retardo de transmisión y la distancia entre nodos. Se necesitan dos slots.
- Three way ranging: Similar al two way ranging, pero en este caso se envían dos ranging responses para compensar la deriva del reloj (clock drift). Tanto el target como el anchor pueden iniciar el procedimiento, siendo el que ha iniciado el intercambio el que finalmente estima la distancia. Se necesitan tres slots, pero la estimación de distancia es mucho más exacta. Por defecto, se utilizará este método.

A.3.9 Adquisición y distribución de la información de localización

De cara a reducir la cantidad de recursos utilizados para la adquisición y distribución de la información de localización, se han implementado diversos modos.

- Solicitud de ranging single/multicast/broadcast: Se puede enviar una solicitud de ranging a un único anchor (single), o a un grupo de ellos (multicast) o a todos los nodos vecinos (broadcast)
- Respuesta de ranging single/multicast: Se puede enviar una respuesta de ranging única (single) inmediatamente después de recibir una solicitud, o por el contrario enviar una respuesta multicast después de recibir varias solicitudes de ranging de distintos iniciadores.
- Agregación de datos: Las distancias estimadas se pueden enviar por separado en distintos paquetes de datos o bien agruparse en un único paquete de datos (agregación de datos)

A.3.10 Método de selección de anchors

De cara a escoger los anchors más cercanos a un determinado target para realizar el posicionamiento, se han implementado diversos modos de selección

- Selección por mínima distancia en base a conocimiento ideal: Selecciona los anchors más cercanos en base a la posición real de los targets
- Selección por mínima distancia en base a las distancias estimadas con broadcast periódico: Realiza periódicamente actualizaciones con todos los anchors vecinos y escoge los más cercanos en base a las distancias estimadas, que se mantendrán hasta la siguiente actualización periódica
- Selección por mínima distancia en base a la posición estimada del target: Selecciona los anchors más cercanos en base a la posición estimada de los targets
- Selección por mínima distancia en base a la posición estimada del target con broadcast periódico: Similar al método anterior, pero incluyendo actualizaciones periódicas con todos los anchors vecinos.

A.4 Documentación del simulador

En este apartado se va a describir el funcionamiento del simulador. El simulador se ha desarrollado en C++, lenguaje orientado a objetos, lo que se adapta muy bien a la situación real, ya que cada clase se corresponderá con un elemento real (target, nodos fijos, location controller, red, etc.), y en ella se implementarán sus funcionalidades. Además, esto también favorece a que el simulador sea más cercano al escenario real, ya que una clase sólo tiene su información o la que se le envía desde otras clases, que es lo que sucede en la situación real.

A.4.1 CSimulator

En esta clase se definen las variables globales del sistema y se implementa el programa principal del simulador. En el programa principal se ejecuta el bucle de simulación donde, a cada paso de simulación, se van llamando a las funciones que gestionan el escenario y las transmisiones.

A.4.2 CNetwork

Esta clase representa a la red, tiene acceso a todos los elementos de la red como son los anchor nodes, target nodes, location managers y picocell manager. Contiene numerosas variables para almacenar la información relevante.

- CTarget_node* target_nodes[MAX_NUMBER_OF_TARGETS]: Mediante este vector se apunta a los targets.

- `CAnchor_node* anchor_nodes[MAX_NUMBER_OF_ANCHORS]`: Este vector permite acceder a los anchors.
- `CLocation_manager* location_managers[MAX_NUMBER_OF_LOCATION_MANAGERS]`: Este vector apunta a los location managers.
- `CPicocell_manager picocell_manager`: Se utiliza para acceder al elemento que actúa como coordinador de la picocelda.
- `double distance_between_anchors`: En esta variable se almacena la distancia entre anchors, que determinará el número de anchors.
- `int distance_between_location_managers`: Este dato sólo se necesita en caso de utilizar la arquitectura centralizada en la red, y determina el número de location managers.
- `int num_anchor_nodes`: Almacena el número de anchor nodes.
- `int num_target_nodes`: Contiene el número de targets introducidos en el fichero parameters.
- `int num_location_managers`: Esta variable guarda el número de location managers.

Además de estas variables, la clase `CNetwork` contiene todos los datos introducidos en el fichero `parameters.txt` ya que esta clase es la que realiza la lectura de los parámetros. A continuación, se explican las funciones implementadas en la clase `CNetwork`.

- `CNetwork(void)`: Constructor de la clase `CNetwork`.
- `bool Init_network()`: Se encarga de iniciar la red, se ocupa de leer los parámetros, crear los targets y los anchors, para ello llama a las siguientes funciones:
 - `bool Load_parameters()`: Realiza la lectura del fichero `parameters.txt`.
 - `bool Load_ranging_model()`: Realiza la lectura del modelo de ranging.
 - `void Distribute_anchors()`: Se encarga de calcular las posiciones de los anchors.
 - `void Compute_neighbourhood_relations()`: Calcula las relaciones de vecindad entre los distintos nodos, tanto entre los anchors como entre los anchors y los targets.
 - `void Distribute_location_managers()`: En el caso de la arquitectura centralizada en la red, esta función calcula los anchors en que deben implementarse la función de location manager.

- void Assign_targets_to_location_managers(): En el caso de la arquitectura centralizada en la red, se encarga de asociar los targets a los distintos location manager existentes.
- void Calculate_number_of_hops(): Para la arquitectura centralizada en la red, calcula el número de saltos necesarios para transmitir un paquete desde cada anchor hasta el location manager más cercano.
- void Update_real_positions(): Llama a la función Update_real_position de cada target.
- void Update_location_stats(): Llama a la función Update_location_error de cada target.
- void Check_position_updates(): Se encarga de llamar a la función Check_position_update de cada target.
- void Update_assignment_targets_to_location_managers(): En la estrategia centralizada en la red, actualiza la asignación de los targets a los location managers.
- void Update_neighbourhood_relations(): Actualiza las relaciones de vecindad entre los targets y los anchors.
- void Collect_statistics(int n_tramas): Recoge los resultados de la simulación y los escribe en los ficheros de salida.

A.4.3 Picocell manager

Representa al coordinador de la picocelda, esta clase controla todo lo relacionado con la transmisión y el uso de slots. Para ello necesita una serie de variables y funciones.

- CAnchor_node** anchor_list: Permite acceder a los anchors.
- CTarget_node** target_list: Permite acceder a los targets.
- int num_target_nodes, int num_anchor_nodes: En estas variables se almacenan el número de targets y de anchors.
- ranging_request* pending_rangings_buffer[MAX_RANGING_BUFFER_SIZE]: Para simular el proceso de ranging, se ha creado un buffer donde se almacenan y se extraen los ranging request.
- location_info_packet* pending_transmissions_buffer[MAX_TRANSMISSIONS_BUFFER_SIZE]: Para simular el proceso de transmisión, se ha creado un buffer donde se almacenan y se extraen los paquetes de datos.

- `int num_of_pending_rangings`: Contiene el número de rangings pendientes.
- `int num_of_pending_transmissions`: En esta variable se almacena el número de transmisiones pendientes.

Además, una serie de variables recogen las estadísticas acerca del uso de recursos:

- `int number_of_ranging_requests`: Número de slots utilizados para la transmisión de tramas de solicitud de ranging.
- `int number_of_ranging_responses`: Número de slots utilizados para la transmisión de tramas de respuesta de ranging.
- `int number_of_measurement_reports`: Número de slots utilizados para la transmisión de tramas de datos con información de las distancias.
- `int number_of_position_update`: Número de slots utilizados para la transmisión de tramas de datos con la posición actualizada.
- `int number_of_location_data_packets`: Número de slots utilizados para la transmisión de tramas de datos, que es la suma de las dos variables anteriores.
- `int total_number_of_slots_for_location`: Almacena el número total de slots utilizados para la localización.
- `double total_bytes_for_location`: Similar a la variable anterior, pero en bytes.
- `double total_time`: Contiene el tiempo total de simulación.
- `double total_time_for_location`: Contiene el tiempo utilizado para la localización.

En cuanto a las funciones realizadas por la clase `CPicocell_manager`, tenemos las siguientes:

- `CPicocell_manager(void)`: Crea el controlador de la picocelda.
- `void Add_pending_ranging(ranging_request *pkt)`: Esta función es la encargada de almacenar una solicitud de ranging en la cola de tramas de ranging.
- `void Add_pending_transmission(location_info_packet *pkt)`: Añade los paquetes de datos a la cola de tramas de datos.
- `ranging_request Extract_pending_ranging()`: Esta función extrae el primer paquete de ranging de la cola de tramas de ranging.
- `location_info_packet Extract_pending_transmission()`: Se encarga de extraer el primer paquete de datos de la cola de tramas de datos.

- void Process_transmission(): Se encarga de procesar los paquetes de datos que haya en la cola de tramas de datos.
- void Process_ranging(): Esta función procesa las solicitudes de ranging que haya en la cola de tramas de ranging, creando las tramas de respuesta correspondientes en función del modo seleccionado.
- void Collect_stats(): Recoge los datos estadísticos relativos al uso de los recursos para presentarlos en el fichero resultados.

A.4.4 CLocation_manager

Esta clase representa a los controladores de localización, que son los elementos que realizan el cálculo de la posición, en función de las distancias estimadas. Dependiendo de la estrategia de localización utilizada, puede localizarse físicamente en uno o varios nodos fijos o en los targets. Para poder definir este elemento se necesitan las siguientes variables.

- int lm_ID: Representa el identificador del location manager.
- CAnchor_node* associated_anchor: En el caso de que el controlador se localice en un nodo anchor, apunta al anchor correspondiente.
- CTarget_node* associated_target: En el caso de que el controlador se localice en un nodo target, apunta al target correspondiente.
- int num_targets, int num_anchors: Indica el número de targets y de anchors de la picocelda.
- int target_list[MAX_NUMBER_OF_TARGETS]: Vector donde se almacenan los identificadores de los targets asociados al controlador.
- target_info target_info_array[MAX_NUMBER_OF_TARGETS]: Contiene información relativa a cada uno de los targets asociados al controlador. La estructura target_info se define en esta clase y contiene toda la información relevante de un target.
- double anchors_position_x[MAX_NUMBER_OF_ANCHORS], double anchors_position_y[MAX_NUMBER_OF_ANCHORS]: En estos dos vectores se almacenan las posiciones de los anchors, que son fijas y conocidas.
- int num_anchors_for_positioning: Indica el número de anchors que se utilizarán para calcular la posición.

En cuanto a las funciones realizadas por esta clase, cabe destacar las siguientes.

- CLocation_manager(int ID): Esta función se encarga de crear el location manager.

- `void Update_estimated_position(int targetID)`: Esta función es la encargada de llamar a la función `Calculate_position`, cuando ya tiene todas las medidas necesarias para el cálculo.
- `void Calculate_position(int target_ID, bool trilaterate)`: En esta función se realiza el cálculo de la posición para un target determinado, para ello se utiliza el filtro de Kalman o la trilateración, dependiendo de lo que se seleccione en el fichero `parameters`. La variable `trilaterate` indica si es la primera vez que se calcula la posición, en cuyo caso se utiliza obligatoriamente trilateración ya que el filtro de Kalman requiere disponer de una posición anterior.
- `void Start_Kalman_filter(int targetID)`: Esta es la función que se utiliza para inicializar el filtro de Kalman. Es llamada dentro de la función `Calculate_position` previamente a utilizar el filtro por primera vez.
- `void select_anchors(int target_ID, int *anchor_list, double position_x, double position_y)`: Selecciona los anchors más cercanos a un target, tiene distintos métodos de funcionamiento implementados, que se seleccionan en `parameters`.
- `void sort_anchor_list_by_distance(int target_ID, int *anchor_list, int list_size)`: Se encarga de ordenar una lista de anchors de menor a mayor distancia con respecto al target introducido.
- `void Start_update(int target_ID)`: Inicia el proceso de actualización de la posición. Determina cuantos anchors se utilizarán para calcular la posición, lo que depende del número de anchors para posicionamiento y del número anchors en cobertura, comprobando además la necesidad de actualización broadcast periódica en el caso de los métodos de selección que la implementan. Se ejecuta la función `select_anchors` para que los elija y se llama a `Start_ranging` del target para que cree los request correspondientes.

A.4.5 CAnchor_node

Este elemento representa a los nodos fijos distribuidos por el escenario, para ello necesitan una serie de variables que almacenen sus datos.

- `int anchor_ID`: Esta variable es el identificador del anchor.
- `double position_x, double position_y`: En estas variables se almacena la posición del anchor, que será fija durante toda la simulación.
- `int num_of_neighbour_anchors`: Aquí tenemos el número de anchors vecinos que tiene un anchor determinado.
- `CAnchor_node* neighbour_anchors[MAX_NUMBER_OF_ANCHORS]`: Este vector se apunta a los anchors vecinos de un anchor dado.

- `double estimated_distances[MAX_NUMBER_OF_TARGETS]`: En esta variable tenemos las distancias estimadas desde un anchor determinado a cada uno de los targets.
- `int targets_for_location[MAX_NUMBER_OF_TARGETS]`: En este vector se almacena el identificador de los targets que utilizan un anchor determinado para la localización.
- `int num_targets_for_location`: Contiene el número de targets que utilizan un anchor determinado para la localización.
- `int num_ranging_available`: Indica el número de solicitudes de ranging recibidas, lo que es necesario para implementar la mejora de respuesta multicast.
- `int hops_to_location_manager[MAX_NUMBER_OF_LOCATION MANAGERS]`: Almacena el número de saltos que hay desde el anchor hasta cada location manager.
- `CPicocell_manager* picocell_manager`: Apunta al elemento que actúa como coordinador de la picocelda.

A continuación, se explican las funciones implementadas en esta clase.

- `CAnchor_node(int ID)`: Esta función es la encargada de crear el anchor node.
- `location_info_packet Create_position_report(int type, int target_ID, double pos_x, double pos_y, double speed_x, double speed_y)`: Esta función crea un paquete de datos para la transmisión de la posición actualizada.
- `location_info_packet Create_measurement_report(int type, int num, int *targets)`: Esta función crea un paquete de datos para la transmisión de una o varias distancias estimadas, en el caso de que sean los anchors los encargados de iniciar el intercambio de ranging.
- `ranging_request Create_ranging_request(int type, int num, int *targets)`: Esta función crea un paquete de solicitud de ranging, en el caso de que los anchors sean los encargados de iniciar el intercambio de ranging.
- `void Start_ranging()`: En el caso de que el intercambio de ranging sea iniciado por los anchors, esta función llama a `create_ranging_request` para crear los paquetes de solicitud de ranging en función del modo de adquisición seleccionado.

A.4.6 CTarget_node

La clase `CTarget_node` representa los nodos móviles a localizar. Dispone de una serie de variables que almacenan sus datos e información relevante para poder simular su comportamiento.

- `int target_ID`: Representa el identificador del target.
- `double real_position_x`, `double real_position_y`: Variables que almacenan la posición real de los targets, utilizadas para calcular el error en la estimación.
- `double estimated_position_x`, `double estimated_position_y`: En estas variables se almacena la posición estimada del target calculada por el location controller.
- `double speed`, `double direction`, `double direction_change_rate`: En estas variables se almacenan la velocidad y la dirección (entre $-\pi$ y π) actuales del móvil.
- `double max_speed`, `double min_speed`, `double direction_change_rate`: En estos parámetros guardamos la información introducida por el fichero de entrada de parámetros acerca del modelo dinámico. En `max_speed` y `min_speed` almacenamos las velocidades máxima y mínima y `direction_change_rate` indica el tiempo que debe transcurrir para que el móvil escoja una nueva dirección y velocidad.
- `double estimated_speed_x`, `double estimated_speed_y`: En estas variables se almacenan la velocidad estimada de cada target.
- `double position_update_rate`, `double refresh_rate`: En `position_update_rate` tenemos la tasa de actualización de la posición y en `refresh_rate` la tasa para las actualizaciones broadcast periódicas para los métodos de selección que las implementan.
- `double time_since_last_update`: En esta variable se va almacenando el tiempo que ha pasado desde la última actualización de la posición, cuando es igual ó mayor al `position_update_rate`, el target necesita una actualización de la posición, y cuando se realiza su valor se pone a cero.
- `double time_since_last_change`: Similar a la variable anterior pero almacena el tiempo desde su último cambio de dirección y velocidad.
- `double time_since_last_refresh`: Esta variable contiene el tiempo transcurrido desde la última actualización broadcast periódica y se pone a cero cuando se realiza.
- `int num_anchors_for_location`: Almacena el número de anchors utilizado para calcular la posición
- `int anchors_for_location[MAX_NUMBER_OF_ANCHORS]`: Almacena el identificador de los anchors utilizados para calcular la posición.
- `int num_of_neighbour_anchors`: Contiene el número de anchors en cobertura del target.

- `CAnchor_node* neighbour_anchors[MAX_NUMBER_OF_ANCHORS]`: Este vector permite a cada target acceder a sus anchors vecinos.
- `int number_of_hops_to_location_manager`: Almacena el número de saltos necesarios para llegar al location manager.
- `int update_status`: Indica el estado en el que se encuentra el target, 0:Estado inicial, se fuerza una actualización inicial 1:La actualización está pedida 2: Se está recibiendo información de ranging 3:Actualizado.
- `int num_ranging_available`: Indica el número de distancias estimadas, necesario para implementar las mejoras de respuesta multicast y agregación de datos.
- `double positioning_error, double sum_of_positioning_error, double sum_of_squared_positioning_error, double average_positioning_error, double variance_positioning_error, int number_of_positioning_proceses, int histogram_count[MAX_NUMBER_OF_CATEGORIES]`: En estas variables se almacenan los datos relativos al error de posicionamiento para cada target, se calculan mediante la función `Update_stats`.
- `int number_of_location_samples, double sum_of_location_error, double sum_of_squared_location_error, double average_location_error, double variance_location_error`: En estas variables se calculan los datos relativos al error de seguimiento, mediante la función `Update_location_error`.
- `int number_of_histogram_categories, int max_histogram_value`: En estas variables se almacenan los datos relativos a los histogramas, se introducen mediante el fichero de entrada `parameters`.
- `double estimated_distances[MAX_NUMBER_OF_ANCHORS]`: Este vector almacena la distancia estimada a cada anchor vecino.
- `CLocation_manager* associated_location_manager`: Apunta al location manager asociado a este target.
- `CPicocell_manager* picocell_manager`: Apunta al elemento que actúa como coordinador de la picocelda.

Una vez conocemos las variables de esta clase, se analizan las funciones que realiza.

- `CTarget_node(int ID, double maximum_speed, double minimum_speed, double dir_change_rate, double pos_update_rate, double v_refresh_rate, int hist_categories, int max_hist_value)`: Constructor del nodo target
- `double Estimate_distance(CAnchor_node* anchor)`: Calcula la distancia estimada al anchor introducido en base a la distancia real mediante la aplicación del modelo probabilístico de ranging.

- `void Update_real_position(int n_slots)`: En esta función se actualizan los contadores de tiempo (`time_since_last_update`, `time_since_last_change`, `time_since_last_refresh`) y calcula la nueva posición real. En caso de que se haya superado el valor de `direction_change_rate`, se determinarán una nueva dirección y velocidad.
- `CAnchor_node* Pick_anchor_for_location_manager()`: Cuando se utiliza una arquitectura distribuida, esta función selecciona el anchor que actuará como location manager.
- `ranging_request Create_ranging_request(int type, int num, int *anchors)`: Esta función crea el paquete de solicitud de ranging, en caso de que el intercambio de ranging sea iniciado por los nodos target.
- `location_info_packet Create_measurement_report(int type, int num, int *anchors)`: Esta función crea un paquete de datos para la transmisión de una o varias distancias estimadas.
- `void sort_neighbours_by_distance(int* anchor_list)`: Ordena la lista de anchors vecinos de menor a mayor distancia respecto al target.
- `void Select_anchors()`: Selecciona los anchors más cercanos a un target, de acuerdo al método de selección escogido.
- `void Check_position_update()`: Comprueba si le toca actualizar la posición (`time_since_last_update`) y si lo necesita llama a `start_update`.
- `void Start_update()`: Inicia el proceso de actualización de la posición. Determina cuantos anchors se utilizarán para calcular la posición, lo que depende del número de anchors para posicionamiento y del número anchors en cobertura, comprobando además la necesidad de actualización broadcast periódica en el caso de los métodos de selección que la implementan. Se ejecuta la función `select_anchors` para que los elija y se llama a `Start_ranging` para crear las solicitudes de ranging correspondientes.
- `void Start_ranging()`: Esta función llama a `create_ranging_request` para crear los paquetes de solicitud de ranging en función del modo de adquisición seleccionado.
- `void Update_target_info()`: Cuando se asigna un target a un determinado location controller, mediante esta función se proporciona al location controller toda la información relativa al target.
- `void Update_stats()`: Calcula los datos relativos al error de posicionamiento.
- `void Update_location_error()`: Calcula los datos relativos al error de seguimiento.

A.4.7 Fichero de entrada de parámetros

En el fichero de entrada (Parameters.txt) se seleccionan los distintos parámetros a utilizar en la simulación, que se exponen a continuación.

- Parámetros generales
 - Tipo de simulador: ideal (ranging y transmisión de datos instantánea e ilimitada) o adaptada a la MAC de EUWB (el número de slots está limitado y la temporización se adapta a la estructura de la supertrama MAC).
 - Tiempo de simulación: Se introduce la duración de la simulación en segundos.
 - Paso de simulación: Se puede elegir este parámetro, por defecto determinado por la duración de los slots de la trama MAC (3.6864 ms).
 - Estrategia de localización: Este parámetro permite seleccionar entre las distintas arquitecturas explicadas anteriormente: centralizada en la red, distribuida, centralizada en el móvil.
 - Iniciador de la actualización de la posición: Permite seleccionar quien inicia el proceso de actualización de la posición, la red o los móviles.
 - Selección de anchors: Indica quien realiza la selección de anchors, la red o los móviles.
 - Modos de selección de anchors: Mediante este parámetro seleccionamos el método de selección de anchors, 1: Ideal, 2: Distancias estimadas con broadcast periódico, 3: Posición estimada y 4: Posición estimada con broadcast periódico.
 - Iniciador del ranging: Permite seleccionar quien inicia los intercambios de ranging, los nodos anchor o los nodos target.
 - Tipo de ranging: Mediante este parámetro se selecciona el modo de funcionamiento del ranging, 2-way ranging o 3-way ranging.
 - Agregación de datos: Con este dato seleccionamos si queremos utilizar agregación de datos o no.
 - Ranging Request: En este caso podemos seleccionar tres modos de funcionamiento, Single request, Multicast request y Broadcast request.
 - Ranging Response: Se elige entre Single response ó Multicast response.
- Parámetros supertrama MAC

Mediante los siguientes parámetros seleccionamos las características de la supertrama, por defecto se utilizan los valores propuestos en el proyecto EUWB.

- Duración de la supertrama: El valor utilizado para este proyecto es de 200 ms
- Duración del slot: Se fija en 3.6864 ms
- Tamaño de paquete (bytes): El tamaño de los paquetes se fija en 160 bytes.
- Número total de slots en la supertrama: Se considera 53.
- Número de slots para el beacon period: Como indica el proyecto se utilizan 12 slots.
- Número de slots para el topology management period: Se le da un valor de 3 slots.
- Número de slots para comunicaciones de datos: Se utiliza un valor de 8 slots.
- Número de slots para ranging: Esta fijado en 12 slots.
- Número de slots para GTS request period: Se fija en 6 slots.
- Número de slots en CAP period: Tiene un valor de 12 slots.

- Parámetros del escenario

A continuación se seleccionan los parámetros que definen las características de la red.

- Distancia entre anchors: La distancia entre los nodos anchor determina el número de anchors que cubrirán el escenario.
- Número de targets nodes: Selecciona el número de dispositivos a seguir.
- Distancia entre location managers: Permite seleccionar la distancia (en número de nodos anchor) que separa a los controladores de localización, lo que determina también su número.
- Tamaño del área x.
- Tamaño del área y.

- Nodos móviles

En este apartado se definen las características de los dispositivos a seguir.

- Velocidad máxima (m/s): Mediante la velocidad máxima y mínima se define el rango de la velocidad del móvil.

- Velocidad mínima (m/s).
- Tasa de cambio de dirección: Marca el tiempo en que la velocidad y la dirección permanece constante, transcurrido ese tiempo selecciona otra dirección y velocidad.
- Tasa de actualización de la posición (ms): Define el tiempo entre actualizaciones de la posición.
- Refresh rate (ms): Para los modos de selección con broadcast periódico, define el tiempo entre las actualizaciones broadcast periódicas.
- Refresh threshold (m): Para el modo de selección basado en distancias estimadas con broadcast periódico, este parámetro se refiere al umbral en metros para realizar una actualización por pérdida de cobertura.

- Algoritmo de seguimiento

Mediante estos parámetros seleccionamos las características del algoritmo de seguimiento.

- Algoritmo de seguimiento: Permite seleccionar entre la trilateración ó el filtro de Kalman.
 - Número de anchors utilizados para la localización.
- Estadísticas
- Número de categorías de histograma.
 - Valor máximo para el histograma.
- Parámetros del modelo de ranging
- Modelo de ranging: Permite seleccionar el modelo de ranging a utilizar.
 - Nivel de error residual de ranging: Este parámetro hace referencia a la varianza del error residual en la estimación de las distancias.

A.4.8 Ficheros de salida: error y resources

El simulador tiene como salida dos ficheros donde se guardan los resultados de la simulación realizada.

En el fichero error.txt, se almacenan las estadísticas relativas al error en el posicionamiento, tanto a nivel global como para cada target. Contiene los siguientes datos:

- Número de procesos de posicionamiento.

- Error medio de posicionamiento: Este es el error de la posición cada vez que se actualiza y por tanto hace referencia al error del procedimiento de cálculo de la posición.
- Varianza del error de posicionamiento.
- Histograma del error de posicionamiento: En base al valor máximo y número de categorías definido en el fichero de parámetros.
- Número de muestras de error de seguimiento.
- Error medio de seguimiento: Este es el error en la posición disponible en el móvil a cada paso de simulación y por tanto tiene en cuenta tanto el error de posicionamiento como el debido a que la posición no esté actualizada.
- Varianza del error medio de seguimiento.

En el fichero resources.txt, se guardan las estadísticas relacionadas con los recursos consumidos para cada simulación, los datos son los siguientes:

- Número de slots utilizados para la transmisión de paquetes de datos relativos a localización: Es la suma de los dos siguientes
- Número de slots utilizados para la transmisión de las distancias estimadas.
- Número de slots utilizados para transmisión de la posición actualizada.
- Número de slots utilizados para respuestas de ranging.
- Número de slots utilizados para respuestas de ranging.
- Número total de slots usados para la localización.
- Número de bytes usados para la localización.
- Tiempo total (s).
- Tasa de datos usada para localización.
- Tiempo total utilizado para localización (s).
- Porcentaje de tiempo usado para la localización.

ANEXO B Resultados completos

B.1 Introducción

Las prestaciones del sistema de localización y seguimiento UWB se han evaluado en términos de error de posicionamiento (media, varianza, error cuadrático medio, distribución del error...) y de la cantidad de recursos utilizados para la localización (número de timeslots utilizados en la adquisición y distribución de la información de la localización).

Con ese propósito se han realizado diversas series de simulaciones con las diferentes arquitecturas y estrategias para la adquisición y distribución de la información de localización propuestas. La influencia del número de anchors utilizados para la localización también se ha analizado, evaluando los distintos métodos de selección de anchors. Finalmente, también se ha analizado el impacto de la movilidad de los targets y la tasa de actualización de la posición.

Para estas simulaciones se han definido un conjunto de parámetros. La duración de las simulaciones se ha fijado en 10000 segundos y el número de targets en 10, excepto en las simulaciones en función del número de targets. El tamaño del área se ha fijado en 50 m x 50 m y la distancia entre anchors en 10 m, salvo que se especifique lo contrario, lo que implica que el área se cubre con 25 anchors. Respecto al modelo dinámico de los targets, las velocidades máxima y mínima se han fijado en 3 m/s y 0.1 m/s, con cambios de dirección cada 20 segundos. El alcance de los nodos UWB se considera de 15 m y la tasa de actualización se ha fijado en 1 segundo salvo en las simulaciones en función de este parámetro.

B.2 Impacto de la arquitectura del sistema de localización

Como se ha explicado anteriormente, la función de controlador de localización (LC) puede localizarse físicamente en uno o varios nodos anchor o en los nodos móviles, en función de la arquitectura del sistema de localización definida. En la arquitectura centralizada en la red, la función LC se implementa en uno o varios nodos anchor definidos previamente. En la arquitectura distribuida, la función LC está implementada en todos los nodos anchor, y el nodo target elige dinámicamente qué nodo anchor la ejecuta. Finalmente, en la arquitectura centralizada en los móviles la función LC se implementa en los nodos target, que calculan su propia posición en base a las distancias estimadas.

En este apartado se evalúa el impacto de cada una de las arquitecturas propuestas sobre la cantidad de recursos necesarios para la localización. Los recursos se miden en términos de timeslots utilizados por segundo. Los slots

se clasifican según su uso en solicitudes de ranging (RRq), respuestas de ranging (RRp), paquetes de datos con informe de medidas (DP-MR) y paquetes de datos con actualización de posición (DP-PU). También se muestra el número total de slots utilizados para la localización.

La Figura B. 1 muestra los resultados de las simulaciones obtenidos para una arquitectura centralizada en la red con 1 controlador de localización implementado en el anchor con coordenadas (25,25), en función del número de targets. Como puede observarse, para las solicitudes de ranging se necesitan aproximadamente 4 slots por segundo y por target, ya que el número de anchors utilizados para la localización se ha fijado en 4. El número de slots necesarios para las respuestas de ranging es el doble que el de solicitudes, ya que estamos utilizando Three-Way Ranging en el que se envían dos respuestas por cada solicitud. Respecto al número de paquetes de datos de informe de medidas, habrá uno por cada intercambio de ranging, pero su transmisión puede necesitar más de un slot, ya que debe retransmitirse hasta el controlador de localización que puede no estar en cobertura del target. Finalmente, únicamente se necesita un paquete de datos de actualización de posición por cada proceso de actualización, pero nuevamente puede requerir varios slots para retransmitirse, necesitándose aproximadamente 2 slots por segundo y por target. En total se necesitan aproximadamente 20 slots por segundo y por target.

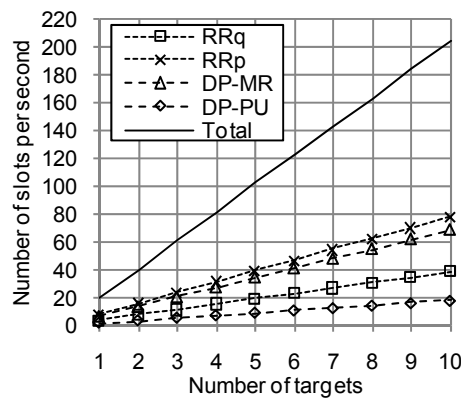


Figura B. 1. Recursos utilizados para la localización para la arquitectura centralizada en la red con 1 controlador de localización

Una posibilidad de cara a reducir el número de veces que deben retransmitirse los paquetes de datos es incrementar el número de controladores de localización. La Figura B. 2 muestra el número de slots utilizados para una arquitectura centralizada en la red con 4 controladores de localización implementados en los anchors con coordenadas (15,15), (15,35), (35,15) y (35,35). Como puede observarse, el número de informes de medidas es prácticamente igual que el de solicitudes de ranging, ya que ahora el target estará la mayor parte del tiempo en cobertura de uno de los controladores de localización y pocas veces será necesario retransmitirlos. De la misma forma, el número de slots necesarios para los paquetes de datos de actualización de posición se reducen a aproximadamente un slot por segundo

y por target. Así, el número total de slots necesarios es de aproximadamente 17 slots por segundo y por target.

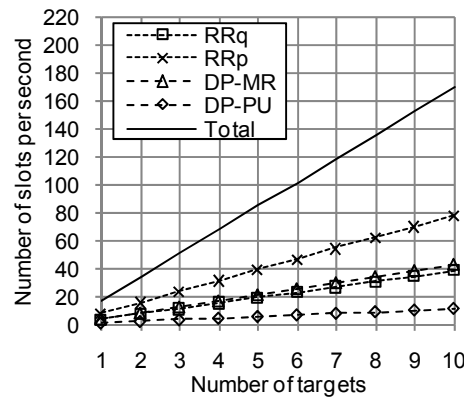


Figura B. 2. Recursos utilizados para la localización para la arquitectura centralizada en la red con 4 controladores de localización

Otra opción para reducir el número de slots necesarios para la localización es utilizar una arquitectura distribuida. En este caso la funcionalidad de controlador de localización siempre es ejecutada por uno de los anchors vecinos del target, por lo que no serán necesarias las retransmisiones. Como se puede observar en la Figura B. 3, el número de slots utilizados para los informes de medidas es exactamente el mismo que el número de solicitudes de ranging, y el número de slots utilizados para los paquetes de actualización de posición es exactamente 1 slot por segundo y por target.

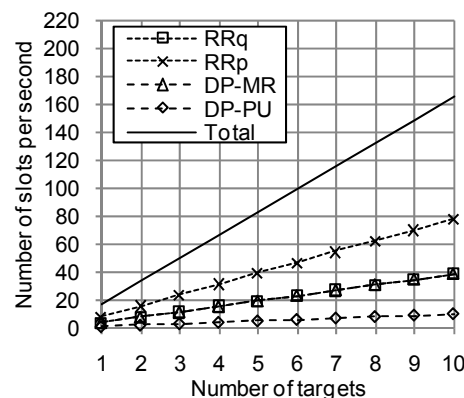


Figura B. 3. Recursos utilizados para la localización para la arquitectura distribuida

Finalmente, otra posibilidad es utilizar una arquitectura centralizada en los móviles. Como puede observarse en la Figura B. 4, la cantidad de slots utilizados para las solicitudes y respuestas de ranging es similar al caso de la arquitectura centralizada en la red, pero en este caso no serán necesarios los paquetes de datos de informe de medidas y de actualización de posición ya que la posición la calcula el propio target. Esto permite que el número total de slots necesarios para la localización se reduzca a aproximadamente 12 slots por segundo y por target.

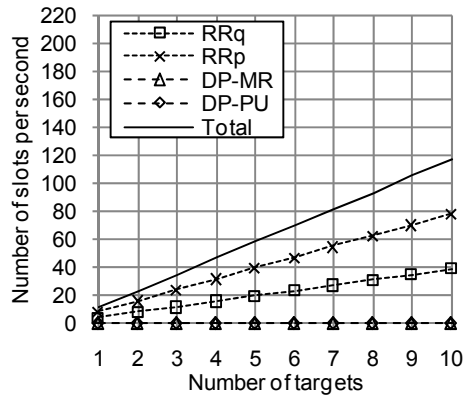


Figura B. 4. Recursos utilizados para la localización para la arquitectura centralizada en los móviles

La Figura B. 5 muestra los resultados obtenidos para cada una de las arquitecturas en términos del número de slots utilizados para la localización. Por lo que respecta a la arquitectura centralizada en la red, se han considerado los casos de 1 y 4 controladores de localización. La arquitectura centralizada en los móviles es la óptima en términos de consumo de recursos, dado que no es necesario transmitir las distancias estimadas ni la posición calculada, lográndose una reducción del 40% en el número de slots con respecto a la arquitectura centralizada en la red con 1 LC. Sin embargo, la implementación de la funcionalidad LC en los móviles incrementaría su complejidad, lo que puede no ser deseable, por lo que la viabilidad de esta arquitectura depende de la aplicación y los dispositivos móviles específicos.

La arquitectura distribuida presenta una reducción de casi un 20% en el número de slots en comparación con la centralizada en la red con 1 LC, ya que la funcionalidad LC siempre se ejecuta en un anchor vecino del target móvil, con lo únicamente se necesita un slot para transmitir las tramas entre el target y el LC. La arquitectura centralizada en la red con 4 LCs presenta unos resultados muy parecidos a la distribuida, con la ventaja de que únicamente es necesario implementar la funcionalidad de LC en 4 de los 25 nodos anchor.

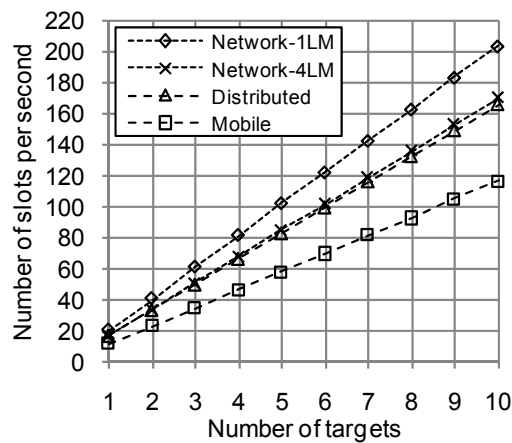


Figura B. 5. Recursos utilizados para la localización para las diferentes arquitecturas

B.3 Mejoras en la adquisición y distribución de la información de localización

De cara a reducir la cantidad de recursos utilizados para la adquisición y distribución de la información de localización, se han implementado diversas mejoras, que se presentaron en detalle en el apartado 2.4.

- Solicitud de ranging multicast/broadcast: Se puede enviar una única solicitud de ranging a todos los nodos vecinos (broadcast), o a un grupo de ellos (multicast).
- Respuesta de ranging multicast: Después de recibir varias solicitudes de ranging de varios iniciadores, se puede enviar una única respuesta multicast.
- Agregación de datos: Se pueden agrupar todas las distancias estimadas y enviarse al LC en un único paquete de datos.

Cabe destacar que no todas las mejoras requieren la misma complejidad. En el caso de que el intercambio de ranging sea iniciado por el nodo target, la implementación de la respuesta multicast requeriría la actualización simultánea de todos los targets para que los anchors pudieran agregar todas las respuestas en una respuesta multicast. Lo mismo ocurriría para la solicitud de ranging multicast y para la agregación de datos en caso de que el intercambio de ranging sea iniciado por los nodos anchor.

Para cada una de las mejoras propuestas se ha evaluado la cantidad de recursos necesarios para la localización considerando una arquitectura centralizada en la red con un controlador de localización, 10 nodos target, una distancia entre anchors de 10 metros, lo que resulta en 25 anchors, una tasa de una actualización por segundo y Three Way Ranging.

En la Figura B. 6 se muestra la cantidad de recursos necesarios en función del número de anchors utilizados para la localización cuando los intercambios de ranging son iniciados por los nodos target y no se aplica ninguna de las mejoras. Como puede observarse, el número de slots necesarios aumenta conforme se incrementa el número de anchors utilizados, aunque permanece constante para más de 7 anchors, ya que con esta configuración con 10 metros entre anchors no es probable que los targets estén en cobertura de más de 7 anchors. El número de solicitudes de ranging es igual al número efectivo de anchors utilizados para la localización, que será igual o menor que el número de anchors para la localización seleccionado dependiendo del número de anchors en cobertura del target. El número de respuestas de ranging es el doble que el de solicitudes al utilizarse Three Way Ranging. El número de paquetes de datos con informes de medidas es igual al número de intercambios de ranging, pero cada paquete puede necesitar varios slots para retransmitirse hasta el controlador de localización. Finalmente, se

necesita un paquete de datos de actualización de posición por segundo y por target independientemente del número de anchors utilizados para la localización, aunque puede requerir varios slots para retransmitirse hasta el target.

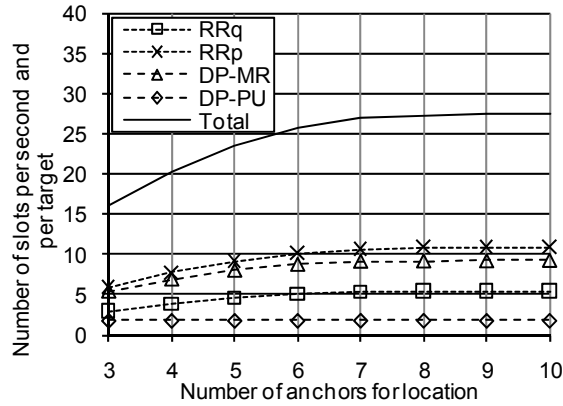


Figura B. 6. Recursos utilizados para la localización sin mejoras (ranging iniciado por targets)

La primera mejora consiste en enviar una solicitud de ranging multicast en lugar de solicitudes individuales a cada anchor. Esto implica que únicamente se enviará una solicitud de ranging en cada actualización independientemente del número de anchors para localización, minimizando así el número de slots utilizados para las solicitudes de ranging a 1 slot por segundo y por target, como se muestra en la Figura B. 7.

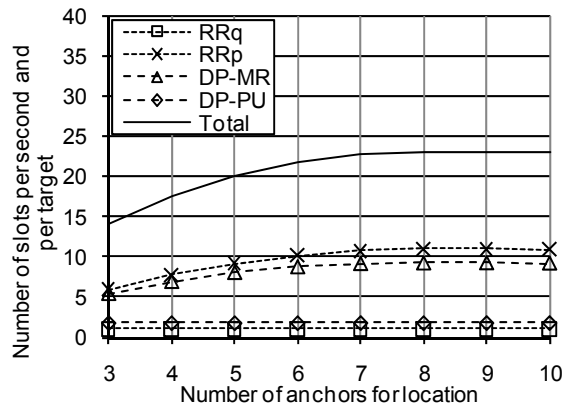


Figura B. 7. Recursos utilizados para la localización con solicitud de ranging multicast (ranging iniciado por targets)

El uso de solicitudes de ranging broadcast tiene el mismo efecto en el número de slots utilizados para las solicitudes de ranging, que se reduce al mínimo de 1 slot por segundo y por target, como se muestra en la Figura B. 8. Además, es una solución más simple que la de la solicitud multicast, ya que no es necesario seleccionar los anchors a utilizar. Sin embargo, tiene un gran inconveniente, ya que todos los anchors en cobertura del target responderán, independientemente del número de anchors para localización seleccionado. En consecuencia, el número de respuestas de ranging y de paquetes de informe de medidas aumenta y se vuelve independiente del número de anchors para la

localización. Esto no tiene un efecto apreciable cuando el número de anchors para la localización es elevado, pero conlleva un importante incremento del número de slots utilizados cuando el número de anchors utilizados para la localización es inferior a 6.

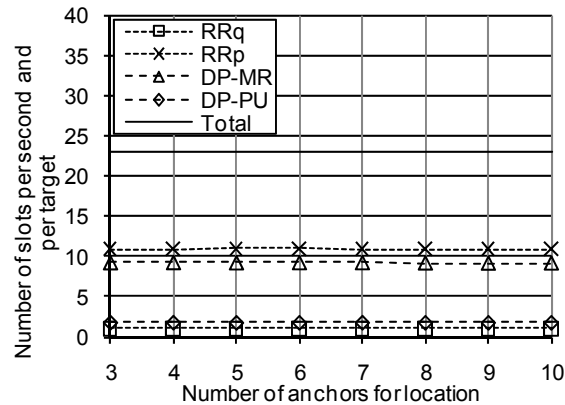


Figura B. 8. Recursos utilizados para la localización con solicitud de ranging broadcast (ranging iniciado por targets)

Cuando se utiliza respuesta de ranging multicast, cada anchor envía únicamente un par de respuestas para todos los targets que le han enviado solicitudes de ranging. Por lo tanto, el número de respuestas de ranging se reduce, en el caso de esta configuración con 10 targets y 10 metros entre anchors en un factor 2.5 aproximadamente, como se puede observar en la Figura B. 9. Este factor está relacionado con el número medio de targets que están en cobertura de cada nodo anchor.

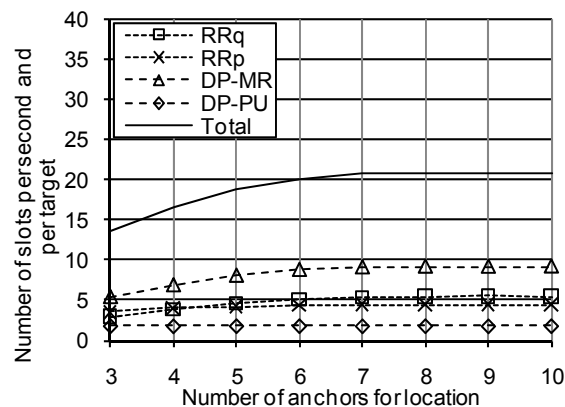


Figura B. 9. Recursos utilizados para la localización con respuesta de ranging multicast (ranging iniciado por targets)

Finalmente, la Figura B. 10 muestra los resultados obtenidos para la mejora de agregación de datos. Dado que todos los informes de medidas se agrupan en un único paquete de datos, el número de slots utilizados para los paquetes de informes de medidas se reduce y pasa a ser independiente del número de anchors utilizados para la localización. Únicamente se enviará un paquete por actualización, por lo que el número de slots necesarios será igual que el utilizado para los paquetes de actualización de posición,

aproximadamente 2 slots por segundo y por target debido a la retransmisión hasta el controlador de localización.

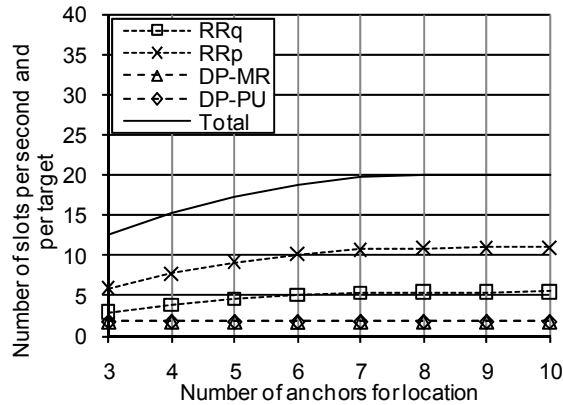


Figura B. 10. Recursos utilizados para la localización con agregación de datos (ranging iniciado por targets)

Por lo que respecta al iniciador del intercambio de ranging, la Figura B. 11 muestra los resultados obtenidos cuando las solicitudes de ranging son enviadas por los anchors en lugar de por los targets. En comparación con el caso en el que las solicitudes de ranging son enviadas por los nodos target, hay una pequeña reducción en el número de slots necesarios para enviar los informes de medidas al controlador de localización. Esto es debido a la reducción del número de retransmisiones que son necesarias para enviar estos paquetes al controlador de localización cuando son los anchors los que estiman las distancias.

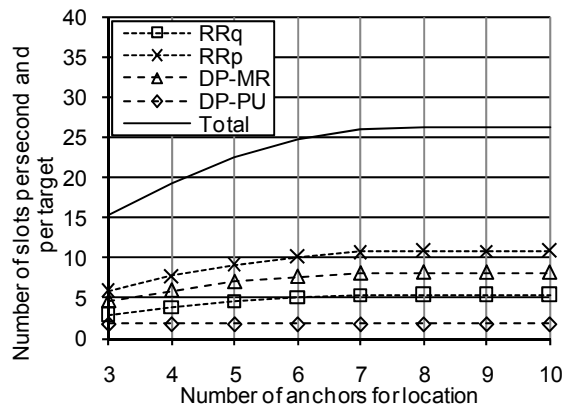


Figura B. 11. Recursos utilizados para la localización sin mejoras (ranging iniciado por anchors)

La Figura B. 12 resume los resultados obtenidos para cada una de las mejoras cuando los intercambios de ranging son iniciados por los targets. Como puede observarse, el número de slots se incrementa conforme aumenta el número de anchors utilizados para el posicionamiento, pero se estabiliza a partir de los 7 anchors, ya que con la configuración utilizada (25 anchors), no es probable que un target esté en cobertura de más de 7 anchors. La agregación de datos (DA) y la respuesta multicast (MRp) permiten reducir en aproximadamente un 25% el número de slots necesarios. Por lo que respecta

a las solicitudes de ranging multicast (MRq) y broadcast (BRq), la primera de ellas muestra una reducción de aproximadamente un 15% en el número de slots para todos los casos, mientras que BRq es independiente del número de anchors, ya que se realiza el ranging con todos los vecinos, presentando una reducción similar para 7 ó más anchors, pero peores resultados para 6 anchors o menos.

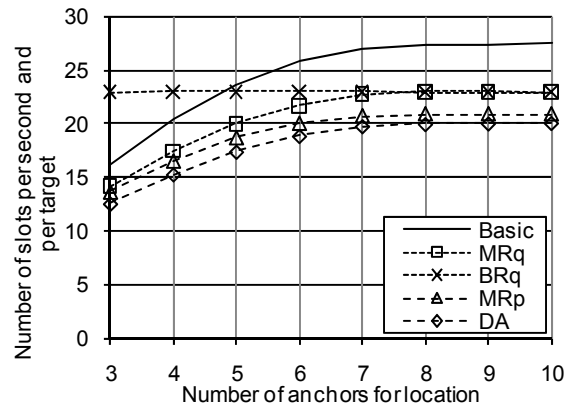


Figura B. 12. Recursos utilizados para la localización para las diferentes mejoras (ranging iniciado por target)

La Figura B. 13 compara los resultados obtenidos para las distintas mejoras cuando los intercambios de ranging son iniciados por los nodos anchor. Utilizando la solicitud multicast (MRq) se logra una reducción de aproximadamente el 12%, que es menor que el 15% que se lograba cuando el intercambio de ranging era iniciado por los targets. La razón es que, para el número de nodos anchor y target utilizado, el número medio de targets con los que un anchor realiza el ranging es menor que el número de anchors utilizados para localizar un target. Por la misma razón, la ganancia de utilizar agregación de datos (DA) se reduce a un 17%. Por contra, la reducción debida al uso de respuesta multicast (MRp) se incrementa hasta aproximadamente un 34%. Finalmente, el uso de solicitudes broadcast (BRq) no se va a tener en cuenta en adelante ya que sus resultados son iguales (para 7 ó mas anchors) o peores (para 6 ó menos anchors) que los de la solicitud multicast.

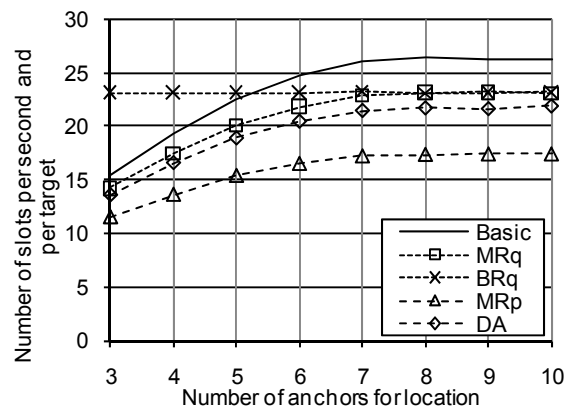


Figura B. 13. Recursos utilizados para la localización para las diferentes mejoras (ranging iniciado por anchors)

En la Figura B. 14 se establece una comparativa entre los modos básicos, los modos mejorados que no requieren la actualización simultánea de las posiciones de todos los targets y los modos que implementan todas las mejoras y que por tanto requieren la actualización simultánea. Cuando no se utiliza ninguna mejora, el número de slots necesarios cuando los anchor inician los intercambios de ranging es ligeramente inferior a cuando lo inician los targets, debido a que la transmisión de las distancias desde los anchors al controlador de localización requiere por lo general los mismos o menos slots que desde el target.

Respecto a los modos mejorados sin actualización simultánea, cuando los targets son los iniciadores del intercambio de ranging pueden aplicarse la solicitud multicast y la agregación de datos (MRq&DA), lográndose una reducción en el número de slots necesarios del 40%. Por contra, cuando los intercambios de ranging son iniciados por los anchors únicamente se puede aplicar la respuesta multicast, con lo que la reducción es menor. Finalmente, cuando se sincronizan las actualizaciones de los targets y se aplican todas las mejoras, los resultados obtenidos tanto si los intercambios de ranging son iniciados por los targets como por los anchors son similares, siendo el número de slots necesarios prácticamente independiente del número de anchors utilizados para localizar un target, variando entre 8 y 9 slots por actualización, lo que implica una reducción del 50%-65% respecto a los modos básicos.

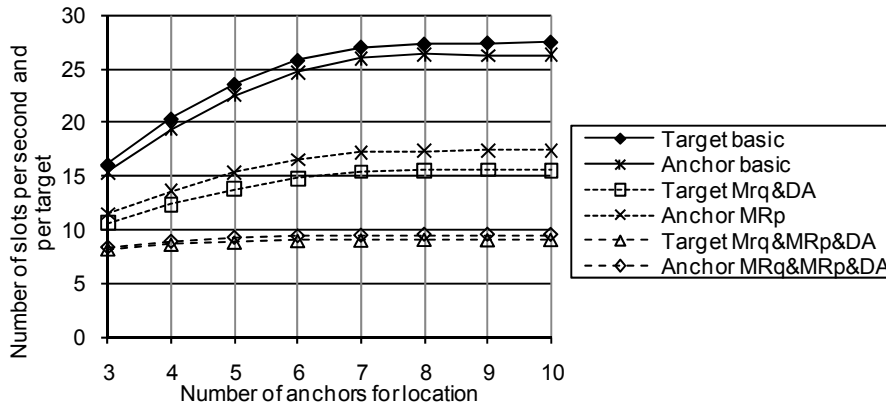


Figura B. 14. Recursos utilizados para la localización en función del iniciador del ranging y de las mejoras aplicadas

Finalmente, se van a evaluar nuevamente las distintas arquitecturas del sistema de LT en el caso de que se apliquen las mejoras de agregación de datos y solicitud multicast, siendo los targets los iniciadores del intercambio de ranging. No se tiene en cuenta la respuesta multicast ya que requiere la actualización simultánea de todos los targets, lo que en las arquitecturas distribuida y centralizada en los targets resultaría complejo al no existir un controlador centralizado en la red que pueda coordinar las actualizaciones de todos los targets. Los resultados se muestran en la Figura B. 15. Nuevamente, la arquitectura centralizada en los móviles muestra los mejores resultados, con una reducción de un 30% en el número de slots necesarios con respecto a

la arquitectura centralizada en la red con 1 LC. La arquitectura distribuida muestra resultados similares a la centralizada en la red con 4 controladores de localización, con una reducción del 12% respecto a la arquitectura centralizada en la red con 1 LC.

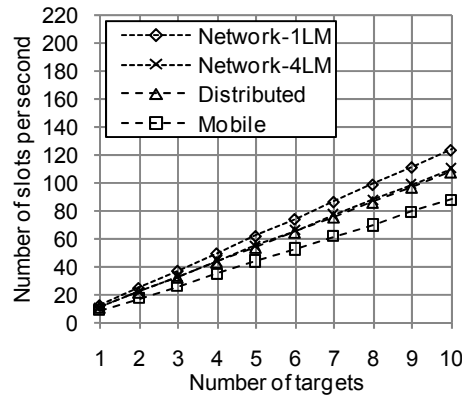


Figura B. 15. Recursos utilizados para la localización para las diferentes arquitecturas con agregación de datos y solicitud de ranging multicast

B.4 Número de anchors utilizados en la localización y distancia entre anchors

Otra posible estrategia de cara a mantener la cantidad de recursos temporales utilizados en la localización en un nivel razonable es limitar el número de anchors que se utilizan para actualizar la posición de un target. De esta forma, el ranging no se realiza con todos los anchors vecino, sino únicamente con un determinado número de ellos. Esto permite reducir el número de slots utilizados para las solicitudes y respuestas de ranging y para la transmisión de las distancias.

Como ya se ha mostrado en la Figura B. 6, para una distancia entre anchors de 10 metros, lo que resulta en 25 anchors, la cantidad de timeslots utilizados se incrementa conforme aumenta el número de anchors utilizados para la localización, y permanece constante para más de 7 anchors, ya que con esta configuración no es probable que el target este en cobertura de más de 7 anchors. Si se limita el número de anchors utilizados para actualizar la posición del target a 4, los recursos utilizados se reducirían de 27 a 20 slots por segundo y por target para esta configuración.

Obviamente, esta reducción también tendrá un efecto sobre el error de posicionamiento, ya que un mayor número de anchors permite una mayor precisión y una menor variabilidad. Sin embargo, dado que el error en la distancia estimada entre anchor y target depende en gran medida de la distancia entre ellos, el error para los anchors cercanos será por lo general menor que para los anchors más alejados, por lo que utilizar un mayor número de anchors no siempre será positivo. Por lo general hay un número de anchors óptimo que minimiza el error de posicionamiento. Cabe recordar que

el modelo de ranging utilizado tiene dos componentes: un sesgo positivo que es debido al efecto del canal y que por tanto es dependiente de la distancia, y un error residual centrado que modela el error en la medida del TOA y que es independiente de la distancia.

La Figura B. 16 muestra el error RMSE (Root Mean Square Error) en función del número de anchors utilizados en la localización para distintos algoritmos (trilateración y filtro de Kalman) y niveles de error residual (0.7 m y 0.3 m). Idealmente se escogen siempre los anchors más cercanos al target. Cabe destacar que el filtro de Kalman muestra un nivel de error RMSE mucho menor que la trilateración ya que, a pesar de que el error medio para ambos algoritmos es comparable, el filtro de Kalman presenta una varianza mucho menor al utilizar también la información de la posición anterior, por lo que el error RMSE es menor para el filtro de Kalman incluso en la situación de error residual alto.

En el caso de la trilateración, la posición se calcula utilizando las tres menores distancias de entre todas las estimadas, a menos que correspondan a tres nodos anchor alineados, en cuyo caso se sustituye la mayor de ellas por la cuarta menor. Por ello sus prestaciones son prácticamente independientes del número de anchors utilizados excepto en el caso de 3 anchors que presenta un importante incremento del error, ya que en ese caso no será posible sustituir uno de ellos en el caso de que estén alineados y habrá dos soluciones posibles (efecto espejo). Por lo que respecta al filtro de Kalman, como se ha comentado anteriormente existe un número óptimo de anchors que minimiza el error, que para esta configuración (10 m entre anchors) es de 4, incrementándose ligeramente el error conforme se utilizan mas anchors. Esta degradación es más marcada en el caso de error residual bajo, ya que en ese caso la importancia relativa del sesgo, que es dependiente de la distancia, es mayor.

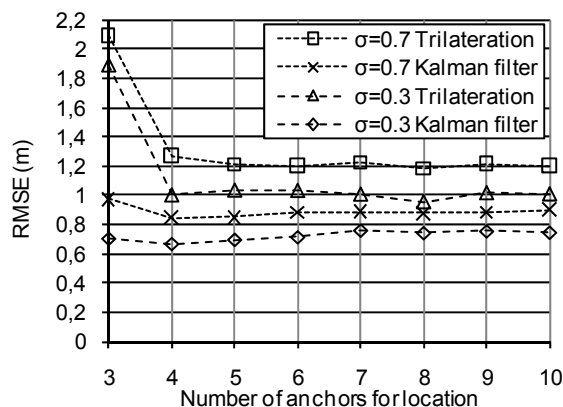


Figura B. 16. Error de posicionamiento en función del número de anchors para la localización con $N_a=25$ anchors

Por tanto, si únicamente se utilizan los 4 anchors más cercanos, tanto el error de posicionamiento como los recursos necesarios se reducen. Lógicamente, el número óptimo de anchors que minimiza el error de

posicionamiento depende de la distancia entre los anchors y del error residual de ranging.

La Figura B. 17 y la Figura B. 18 muestran el error RMSE para el filtro de Kalman con nivel de error residual de ranging alto y bajo respectivamente para tres configuraciones diferentes: 25 anchors (10 m entre anchors), 49 anchors (7.15 m entre anchors) y 100 anchors (5 m entre anchors). Para error residual de ranging alto, el RMSE mínimo se reduce de 0.85 m para 25 anchors a 0.56 m para 49 anchors y 0.42 m para 100 anchors. Conforme se reduce la distancia entre anchors, el número óptimo de anchors aumenta, pasando de 4 para 25 anchors a 6 para 49 anchors y 10 para 100 anchors. Cuando el error residual de ranging es bajo, las distancias estimadas son más precisas y el error se reduce, con un error RMSE mínimo de 0.66 m para 25 anchors, 0.32 m para 49 anchors y 0.25 m para 100 anchors. Al reducir el error residual de ranging, el sesgo tendrá más importancia y el número óptimo de anchors se reduce a 4 para 25 y 49 anchors y 7 para 100 anchors.

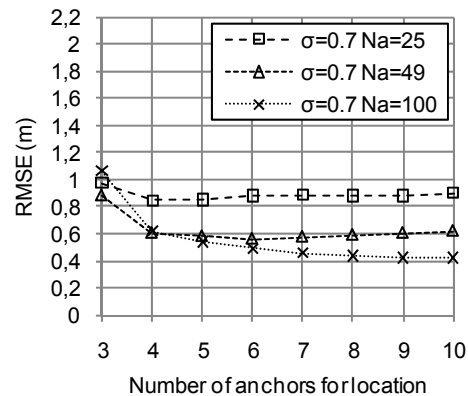


Figura B. 17. Error de posicionamiento en función del número de anchors para la localización con filtro de Kalman y $\sigma_n=0.7m$

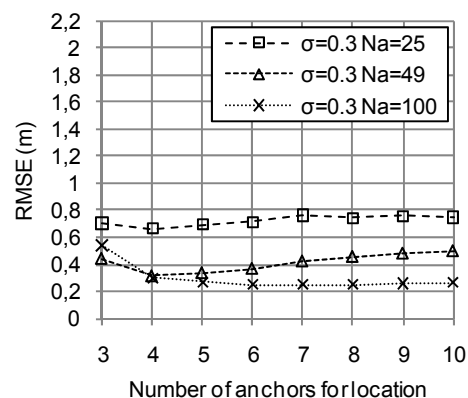


Figura B. 18. Error de posicionamiento en función del número de anchors para la localización con filtro de Kalman y $\sigma_n=0.3m$

La Figura B. 19 muestra la cantidad de recursos utilizados para la localización en función del número de anchors utilizados para localizar a los targets. Como puede observarse, para 100 anchors el número de timeslots necesarios se incrementa linealmente de acuerdo al número de anchors utilizados, ya que el target siempre estará en cobertura de más de 10 anchors.

Para 49 anchors, el número de timeslots se incrementa linealmente para menos de 8 anchors, con un incremento menor para más de 8 anchors. Para 25 anchors, el incremento no es lineal ya que es posible que el target no esté en cobertura del número seleccionado de anchors, permaneciendo constante para más de 7 anchors.

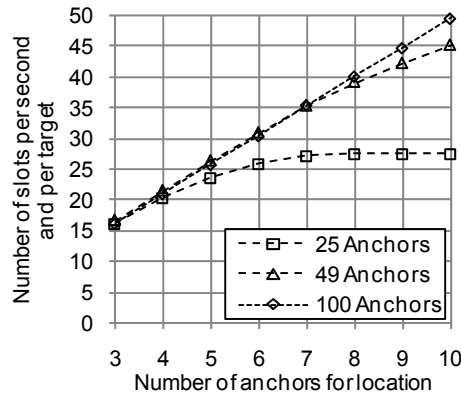


Figura B. 19. Recursos utilizados para la localización para diferentes valores del número total de anchors

Conforme la distancia entre los anchors se reduce, el target estará en cobertura de más anchors y la necesidad de limitar el número de anchors utilizados para la localización es más importante de cara a evitar un uso excesivo de los recursos. Hasta este momento se ha considerado un enfoque ideal en el que siempre se seleccionaban los anchors más cercanos. En el caso real, se deberá diseñar un método de selección de anchors en base a la información disponible en los nodos target o en los controladores de localización (posición estimada del móvil, posiciones conocidas de los anchors, distancias estimadas previamente...). Los distintos métodos de selección de anchors diseñados se evalúan en el siguiente apartado.

B.5 Método de selección de anchors

Tal y como se concluyó en el apartado anterior, limitar el número de anchors utilizados para la localización puede reducir tanto el error de posicionamiento como la necesidad de recursos, aunque para lo primero es necesario escoger los anchors más cercanos al target. Por ello es necesario implementar un método de selección de anchors basado en la información disponible tanto en la red como en los nodos target. Los distintos métodos propuestos se presentaron en el apartado 2.7:

- Selección por mínima distancia en base a conocimiento ideal (Ideal)
- Selección por mínima distancia en base a las distancias estimadas con broadcast periódico (PB)
- Selección por mínima distancia en base a la posición estimada del target (EP)

- Selección por mínima distancia en base a la posición estimada del target con broadcast periódico (EP-PB)

Las prestaciones de cada uno de los métodos se han evaluado en función del número de anchors utilizados para el posicionamiento. La distancia entre anchors se ha fijado en 10 metros y, en el caso de los métodos PB y EP-PB, el tiempo entre actualizaciones broadcast periódicas en 10 segundos. Se ha utilizado una arquitectura centralizada en la red sin mejoras. La Figura B. 20 y la Figura B. 21 muestran el error RMSE y el número de slots utilizados para la localización respectivamente. Cuando se utilizan 4 anchors para el posicionamiento, los mejores resultados se obtienen con el método EP-PB, acercándose a los resultados de la selección ideal, mientras que el método PB obtiene los peores resultados. Para 5 anchors, los tres métodos obtienen resultados similares y ligeramente superiores a la selección ideal. Finalmente, para 6 o más anchors los resultados son similares independientemente del método utilizado, ya que en la mayoría de las ocasiones el target no estará en cobertura de más de 6 anchors.

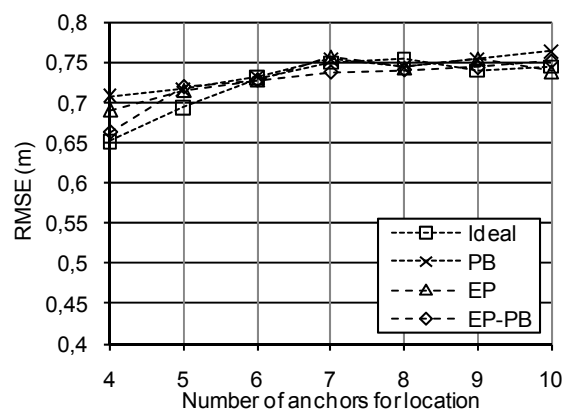


Figura B. 20. Error de posicionamiento para los distintos métodos de selección (distancia entre anchors = 10m)

Por lo que respecta a la necesidad de recursos, dado que las actualizaciones broadcast periódicas se realizan con todos los anchors en cobertura y no con el número seleccionado, los métodos que presentan esta característica (PB y EP-PB) requieren un mayor número de slots. El número de slots para el método PB es superior al de EP-PB ya que también realiza actualizaciones broadcast cuando se pierde la cobertura con uno de los anchors seleccionados. Para 7 ó más anchors todos los métodos utilizan la misma cantidad de slots, ya que no es probable que un target se encuentre en cobertura de más de 7 anchors.

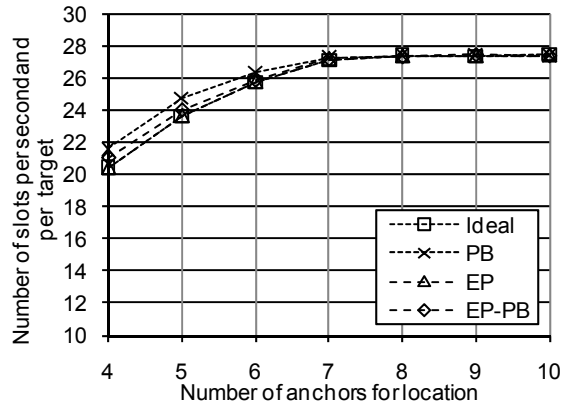


Figura B. 21. Recursos utilizados para la localización para los distintos métodos de selección (distancia entre anchors = 10m)

El mismo análisis se ha realizado para una configuración con distancia entre anchors igual a 7.15 metros, mostrándose los resultados en la Figura B. 22 y la Figura B. 23. Con esta configuración, el error para los métodos no ideales ya no es tan cercano al caso ideal, ya que el target tiene más anchors en cobertura y la probabilidad de que la selección no sea óptima se incrementa. Los resultados para EP y EP-PB son ligeramente superiores a los de la selección ideal, aunque en el caso de EP hay un importante incremento en el caso de que se utilicen 4 anchors. Esto es debido a la realimentación del error en la estimación de la posición a través de una selección incorrecta de anchors. Por otro lado, el método PB presenta un error RMSE mayor que el resto de métodos, ya que los anchors seleccionados en la actualización broadcast periódica ya no serán los óptimos al cabo de unos pocos segundos.

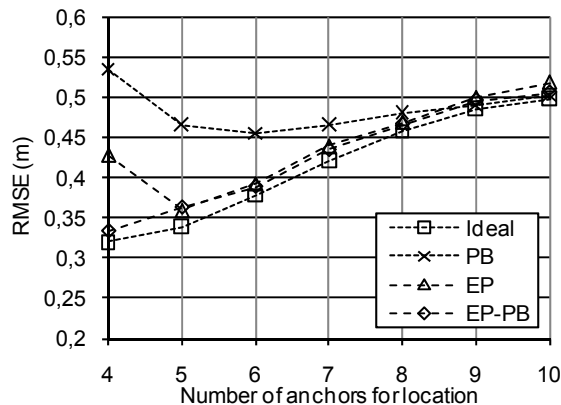


Figura B. 22. Error de posicionamiento para los distintos métodos de selección (distancia entre anchors = 7.15m)

Por lo que respecta a la necesidad de recursos, los métodos con actualizaciones broadcast periódicas requieren un mayor número de slots, especialmente en el caso de PB ya que también realiza actualizaciones broadcast cuando se pierde la cobertura con un anchor seleccionado. Conforme se reduce la distancia entre anchors, el target está en cobertura de más anchors, las actualizaciones broadcast requieren más slots y la diferencia entre los métodos con y sin actualizaciones broadcast periódicas aumenta.

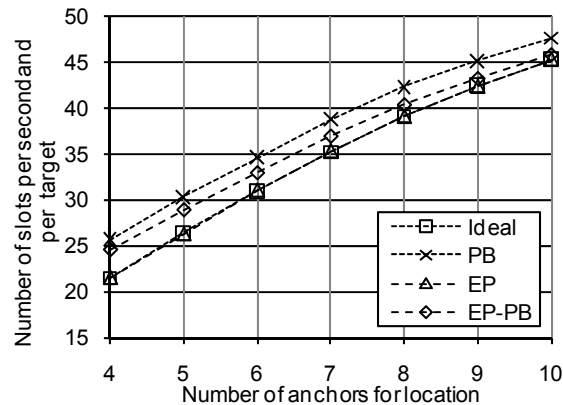


Figura B. 23. Recursos utilizados para la localización para los distintos métodos de selección (distancia entre anchors = 7.15m)

En la Figura B. 24 y en la Figura B. 25 se analiza el impacto del tiempo entre actualizaciones broadcast periódicas en el error de posicionamiento y en la necesidad de recursos. El número de anchors utilizados para el posicionamiento se ha fijado en 4 y la distancia entre anchors en 10 metros. Como cabe esperar, los métodos que no implementan actualizaciones broadcast periódicas (ideal y EP) son independientes del tiempo entre actualizaciones broadcast periódicas, aunque se muestran en la figura como referencia. El método PB logra resultados incluso mejores que la selección ideal cuando el tiempo entre actualizaciones periódicas es igual o menor que 2 segundos. Esto es debido a que realiza el ranging con todos los anchors que tiene en cobertura, aunque luego únicamente se utilizan los cuatro con menor distancia estimada para el cálculo. De esta forma, si la distancia estimada para uno de los anchors más cercanos presenta un error elevado (por ejemplo debido a la existencia de un obstáculo), otro anchor con un menor error podría presentar una distancia estimada menor y por tanto ser seleccionado, logrando un mejor resultado que escogiendo únicamente los cuatro anchors más cercanos como en el método ideal. Conforme se incrementa el tiempo entre las actualizaciones broadcast periódicas, el error del método PB aumenta, ya que los anchors seleccionados dejarán de ser los óptimos al cabo de unos pocos segundos. El método EP-PB presenta un resultado similar al PB para el caso de 1 segundo, ya que en ese caso todas las actualizaciones serán tipo broadcast, mientras que para valores superiores su comportamiento es muy similar al del método ideal.

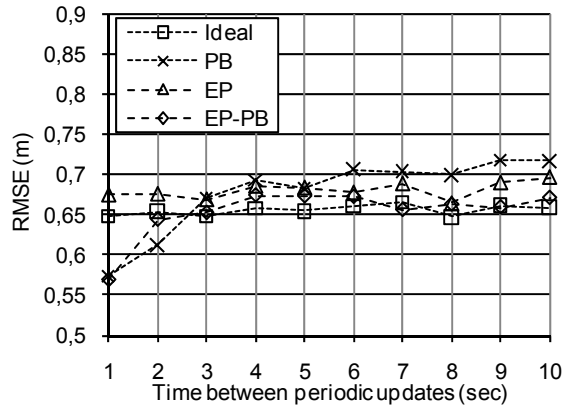


Figura B. 24. Error de posicionamiento en función del tiempo entre actualizaciones broadcast periódicas (distancia entre anchors = 10m)

Respecto a la cantidad de recursos necesarios para la localización, los métodos PB y EP-PB utilizan una mayor cantidad de slots debido a las actualizaciones broadcast periódicas, que se reduce conforme se incrementa el tiempo entre actualizaciones broadcast periódicas. Para valores superiores a 7 segundos, el número de slots utilizados con el método PB permanece constante ya que se realizarán actualizaciones broadcast por pérdida de cobertura con uno de los anchors seleccionados antes de que ocurran las actualizaciones broadcast periódicas.

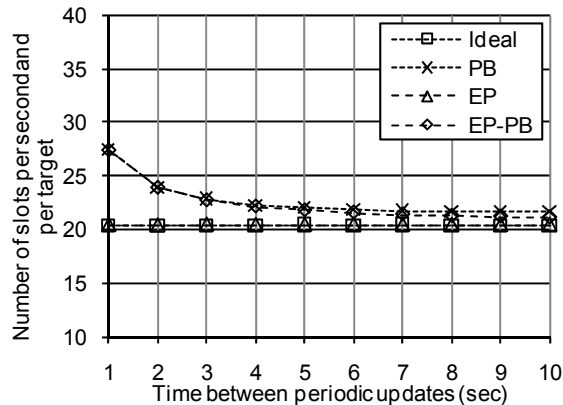


Figura B. 25. Recursos utilizados para la localización en función del tiempo entre actualizaciones broadcast periódicas (distancia entre anchors = 10m)

Un análisis similar se muestra en la Figura B. 26 y en la Figura B. 27 para una configuración con 7.15 metros entre los anchors. Por lo que respecta al error RMSE, cabe destacar que el método EP presenta una variabilidad muy alta, aunque en promedio es independiente del tiempo entre actualizaciones periódicas. Esto es debido a la realimentación del error en la estimación de la posición a través de una selección incorrecta, que hace que en ocasiones el error pueda alcanzar puntualmente valores muy elevados, lo que provoca la alta variabilidad en función del número de veces que esto ocurra. Por ello se propone combinar el método EP con actualizaciones broadcast periódicas. Como puede observarse, el comportamiento del método EP-PB es muy cercano al de la selección ideal. Por lo que respecta al método PB, sus prestaciones se degradan de forma importante conforme se incrementa el tiempo entre

actualizaciones broadcast periódicas para valores por encima de los 3 segundos, ya que los anchors escogidos ya no sean los óptimos al cabo de 3-4 segundos.

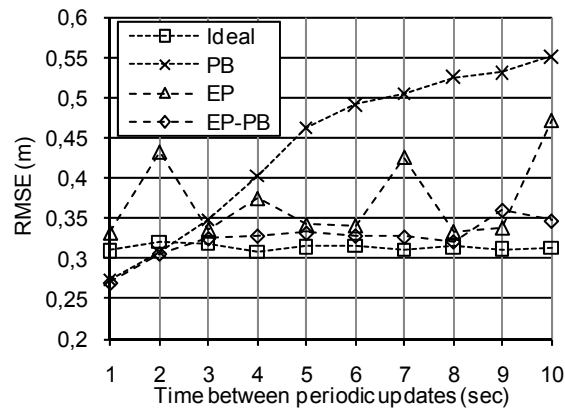


Figura B. 26. Error de posicionamiento en función del tiempo entre actualizaciones broadcast periódicas (distancia entre anchors = 7.15 m.)

Por lo que respecta a la cantidad de recursos, los resultados para los métodos ideal y EP son iguales que para la configuración de 10 metros, pero el número de slots utilizados para los métodos PB y EP-PB se incrementa debido a que el target tendrá más anchors en cobertura y por tanto se necesitarán más slots en las actualizaciones broadcast.

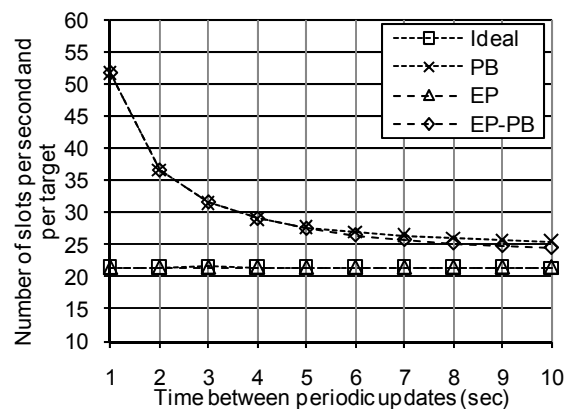


Figura B. 27. Recursos utilizados para la localización en función del tiempo entre actualizaciones broadcast periódicas (distancia entre anchors = 7.15 m.)

Finalmente, en la Figura B. 28 y en la Figura B. 29 se analiza el comportamiento de los distintos métodos de selección en función de la distancia entre anchors. El número de anchors utilizados para el posicionamiento se ha fijado en 4 y el tiempo entre actualizaciones broadcast periódicas en 10 segundos. Cuando la distancia entre anchors es de 10 metros, el comportamiento es similar para todos los métodos, ya que la mayor parte del tiempo el target sólo estará en cobertura de 5 ó 6 anchors y será muy probable que se seleccionen los mismos anchors independientemente del método. La necesidad de recursos es ligeramente superior para los métodos con actualizaciones broadcast periódicas.

Conforme la distancia entre anchors se reduce, la diferencia entre los distintos métodos se hace más evidente ya que la selección de los anchors se hace más crítica. Para una distancia de 7.15 metros el error RMSE se reduce para todos los métodos, ya que los anchors seleccionados serán más cercanos que en el caso de 10 metros. Sin embargo, ahora el target tendrá más anchors en cobertura, por lo que aumenta la probabilidad de que la selección con los métodos no ideales no sea la óptima, y sólo con EP-PB se consigue un resultado similar al de la selección ideal. Por lo que respecta a la necesidad de recursos, el coste de realizar actualizaciones broadcast periódicas en los métodos PB y EP-PB se incrementa al aumentar el número de anchors que el target tiene en cobertura.

Para la configuración con 5 metros entre anchors, el error RMSE únicamente se reduce para la selección ideal y se incrementa para el resto. Esto es debido principalmente al incremento en la varianza del error, ya que se incrementa la probabilidad de una selección de anchors no óptima. El incremento del error RMSE es especialmente destacable en el caso del método EP debido a la realimentación del error en la posición a través de una selección incorrecta. Cabe destacar que el método EP presenta mejores resultados que el PB en cuanto a error medio (26 cm. frente a 34 cm.) y el % de veces que el error está por debajo de 20 cm (80% frente a 60%), pero el importante incremento de la varianza debido a estos errores ocasionalmente elevados hace que el error RMSE sea mayor. Finalmente, la necesidad de recursos de los métodos con actualización broadcast periódica (PB y EP-PB) también se incrementa ya que el target tendrá más anchors en cobertura, por lo que las actualizaciones broadcast requerirán más slots.

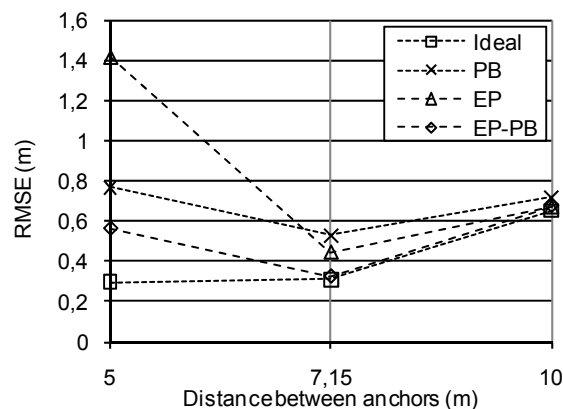


Figura B. 28. Error de posicionamiento para los distintos métodos de selección en función de la distancia entre anchors

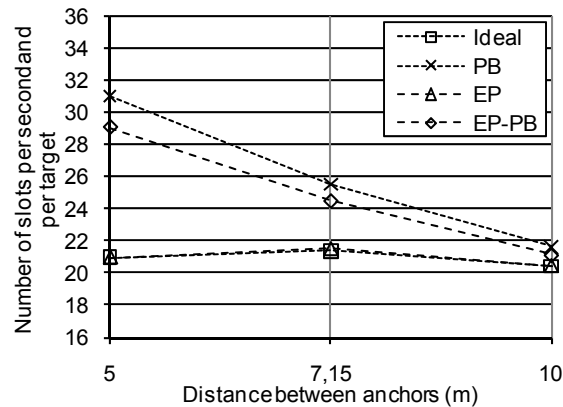


Figura B. 29. Recursos utilizados para la localización para los distintos métodos de selección en función de la distancia entre anchors

En resumen, la importancia del método de selección de anchors se incrementa conforme se reduce la distancia entre anchors y el target está en cobertura de un mayor número de anchors. PB requiere que se realicen actualizaciones broadcast periódicas cada 3-4 segundos para proporcionar buenos resultados, pero esto implica una mayor necesidad de recursos. EP proporciona buenos resultados por lo general, aunque en algunos casos el error puede incrementarse de manera importante si se realimenta a través de una selección incorrecta de los anchors. De cara evitar esto pueden proponerse diversas soluciones, siendo una de ellas la combinación de EP con actualizaciones broadcast periódicas (EP-PB), que proporciona unas prestaciones cercanas a la selección ideal con un uso razonable de recursos.

B.6 Efecto de la movilidad y de la tasa de actualización de la posición

Finalmente, otro parámetro importante relacionado tanto con el error de posicionamiento como con la cantidad de recursos utilizados para la localización es la tasa de actualización de la posición, que define la frecuencia con la que se actualizan las posiciones de los nodos móviles. La tasa de actualización de la posición está íntimamente relacionada con la movilidad de los nodos target, ya que una mayor velocidad de los móviles requiere de una mayor frecuencia de actualización de la posición de cara a realizar un seguimiento preciso de los targets.

De cara a evaluar el efecto de la movilidad de los targets y de la tasa de actualización de la posición se han llevado a cabo una serie de simulaciones. Se ha considerado una arquitectura centralizada en la red con 1 LC, sin mejoras de adquisición y distribución, una distancia de 10 metros entre los nodos anchor y 4 anchors utilizados para el posicionamiento.

En primer lugar se analiza el efecto de la movilidad de los targets sobre el error en la localización. La movilidad de los targets viene definida por su velocidad y por la tasa de cambio de dirección. De cara a medir el error se

definen dos parámetros. El error en la estimación de la posición (PEE) mide el error de la posición estimada por el algoritmo en cada actualización. El error de seguimiento (TE) mide el error en la posición disponible en el móvil a cada paso de simulación, no sólo en las actualizaciones. Este error no se debe únicamente al error en la estimación, sino también al movimiento de los nodos target desde la última actualización.

La Figura B. 30 muestra el error de posicionamiento y el error de tracking en función de la velocidad de los targets. El tiempo entre actualizaciones, que es el inverso de la tasa de actualización de la posición, se ha fijado en 1 segundo. Como puede observarse, el error de posicionamiento es prácticamente independiente de la velocidad de los targets en el caso del algoritmo de trilateración, ya que la posición se calcula de manera independiente en cada actualización. Por otro lado, en el filtro de Kalman la posición actual se calcula en base a la posición anterior, lo que le permite lograr una mejor precisión para velocidades bajas, aunque sus prestaciones se degradan conforme se incrementa el error, aunque incluso para 3 m/s el error de posicionamiento es inferior al de la trilateración. Por lo que respecta al error de seguimiento, en ambos casos aumenta conforme se incrementa la velocidad, ya que la distancia recorrida por los targets entre cada actualización de la posición será mayor.

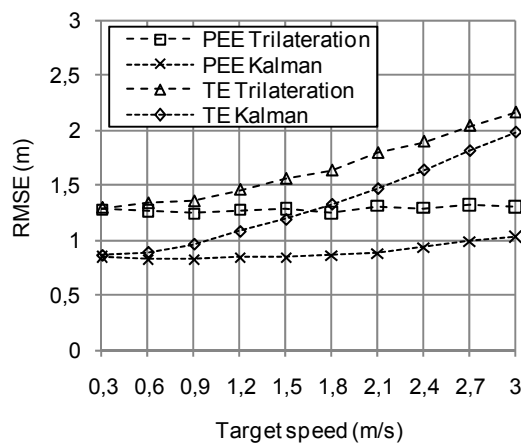


Figura B. 30. Error de posicionamiento y error de seguimiento en función de la velocidad de los targets

Por lo que respecta a la tasa de cambio de dirección, tanto el error de posicionamiento como el error de seguimiento es prácticamente independiente del tiempo entre cambios de dirección, como se observa en la Figura B. 31. Únicamente para tiempos menores de 4 segundos hay un ligero incremento del error.

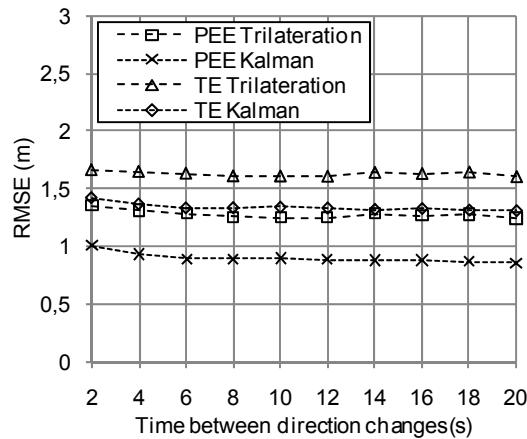


Figura B. 31. Error de posicionamiento y error de seguimiento en función del tiempo entre cambios de dirección

La Figura B. 32 muestra el error en la estimación de la posición y el error de seguimiento en función del tiempo entre actualizaciones, considerando una velocidad de los targets aleatoria y uniformemente distribuida entre 0.1 m/s y 3 m/s y una tasa de cambio de dirección de 20 segundos. Por lo que respecta al PEE, la precisión de la trilateración es independiente del tiempo entre actualizaciones, ya que la posición se calcula de manera independiente en cada actualización. Por otro lado, la estimación en el filtro de Kalman se basa en la posición anterior, lo que resulta en una mayor precisión cuando el tiempo entre actualizaciones es inferior a 1.6 segundos, aunque la precisión se degrada conforme se incrementa el tiempo entre actualizaciones. Por lo que respecta al TE, la precisión se degrada conforme aumenta el tiempo entre actualizaciones para ambos casos, ya que la distancia recorrida por los móviles entre actualizaciones consecutivas será mayor. Esta degradación es mayor en el caso del filtro de Kalman, ya que al error por la falta de actualización se suma la mayor degradación en el error de estimación.

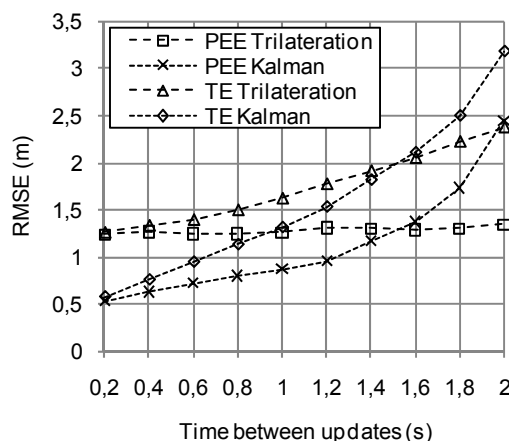


Figura B. 32. Error de posicionamiento y error de seguimiento en función del tiempo entre actualizaciones

Finalmente, en la Figura B. 33 se muestra el impacto de la tasa de actualización de la posición respecto a la cantidad de recursos dedicados a la

localización. Como puede observarse, el número de timeslots utilizados en la localización es inversamente proporcional al tiempo entre actualizaciones. Por tanto, de cara a minimizar la cantidad de recursos necesarios, sería deseable que el tiempo entre actualizaciones sea elevado.

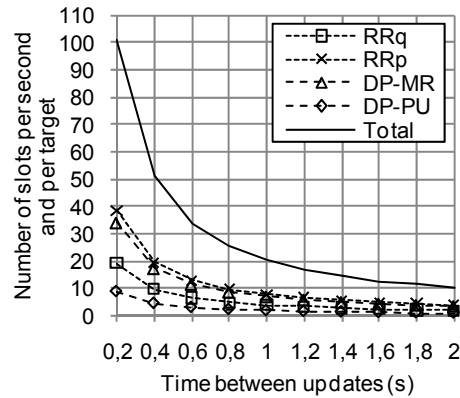


Figura B. 33. Recursos utilizados para la localización en función del tiempo entre actualizaciones

Tal y como puede observarse, no existe un valor óptimo para la tasa de actualización de la posición, ya que hay un compromiso entre el error de seguimiento y la cantidad de recursos utilizados para la localización. La tasa de actualización de la posición debe escogerse de acuerdo a la precisión de seguimiento requerida y al modelo dinámico de los móviles. Para seguir a personas desplazándose a pie, 1 actualización por segundo es un valor adecuado para mantener el error de seguimiento alrededor de 1 metro con un uso razonable de recursos.