



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

MODELAMIENTO DE SERVICIOS DE GEO-LOCALIZACIÓN  
EN REDES CONVERGENTES

MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL ELECTRICISTA

NICOLÁS CARRASCO STEVENSON

PROFESOR GUÍA:  
ALBERTO CASTRO ROJAS

MIEMBROS DE LA COMISIÓN:  
DIEGO SAAVEDRA RENDIC  
CLAUDIO ESTÉVEZ MONTERO

SANTIAGO DE CHILE  
MARZO 2013



# Resumen

La aparición de una serie de servicios basados en localización (LBS) ha hecho necesaria la creación de técnicas de geo-localización más precisas que los métodos basados en parámetros de red que se utilizan habitualmente. Así, se hace necesario contar con metodologías y parámetros que permitan evaluar y posteriormente escoger aquél que resulte más apropiado para la aplicación LBS que se desea implementar.

En la actualidad existe una serie de métodos de localización basados en parámetros de la señal como: *Radio Signal Strength* (RSS), *Time of Arrival* o *Angle of Arrival*, por nombrar algunos, que se utilizan por ejemplo en aplicaciones de asignación de recursos de red, control de flota o determinación del origen de una llamada de emergencia. En general, se conoce el desempeño de cada método para las distintas tecnologías de transmisión, pero no existe un criterio unificado que permita evaluar y comparar *a priori* dos modelos de características similares, usando los conceptos de *precision* & *accuracy* como estimadores del desempeño de un modelo de geo-localización.

En el desarrollo de este trabajo se simularon una serie de métodos de geo-localización, aplicados a un mismo modelo que incluye un terminal móvil. La simulación se realizó mediante el software MiXiM, construido sobre el *framework* de código abierto Omnet++. El tratamiento posterior de los datos obtenidos de la simulación se realizó mediante funciones programadas en el software de cálculo científico Matlab.

Los resultados del experimento se centran en comparar distintos modelos de localización, mostrando ser superiores aquellos basados en intensidad de señal. También se comparan técnicas de resolución de la localización: aquí el método de mínimos cuadrados no-lineales se alza como el mejor a la hora de minimizar el error de localización. Por otra parte, el uso del algoritmo de Levenberg-Marquardt permite aumentar el número de localizaciones favorables al costo de disminuir el desempeño del modelo.

Como conclusión del trabajo, se puede mencionar que la falta de una herramienta idónea para la simulación hace difícil medir el desempeño de los métodos basados en tiempo de vuelo de la señal, ya que se requiere de precisión del orden de microsegundos para obtener resultados significativos. No obstante, la simplicidad del método basado en RSS permite generar simulaciones con errores del orden de decenas de metros, aún cuando el modelo implementado sea bastante rudimentario.



*In memoriam Luz María y Juan Pablo*

# Agradecimientos

# Tabla de contenido

<b>1. Introducción</b>	<b>1</b>
1.1. Alcances y Objetivo General . . . . .	1
1.2. Objetivos Específicos . . . . .	2
1.3. Descripción del documento . . . . .	2
<b>2. Antecedentes</b>	<b>4</b>
2.1. Locación, localización y localizabilidad . . . . .	4
2.2. Servicios basados en locación (LBS) . . . . .	4
2.3. Localización basada en redes y aplicaciones . . . . .	5
2.3.1. Localización mediante servidores DHCP . . . . .	5
2.3.2. Localización mediante servidores de información de locación (LIS) . . . . .	5
2.4. Técnicas de medición para localización en redes inalámbricas . . . . .	6
2.4.1. Técnicas basadas en alcance . . . . .	6
2.4.2. Técnicas basadas en proximidad . . . . .	8
2.5. Métodos matemáticos para localización . . . . .	10
2.5.1. Multilateración . . . . .	10
2.5.2. Triangulación . . . . .	12
2.5.3. Perfil RSS . . . . .	13
2.6. Algoritmos de Localización . . . . .	13
2.6.1. Algoritmos basados en distancia . . . . .	13
2.6.2. Algoritmos basados en conectividad . . . . .	15
2.6.3. Precisión y Exactitud . . . . .	15
2.7. Errores de localización . . . . .	15
2.7.1. Cota inferior de Cramér-Rao . . . . .	15
2.7.2. Errores intrínsecos . . . . .	16
2.7.3. Errores extrínsecos . . . . .	16
2.7.4. Impacto de errores en algoritmos de localización . . . . .	16
2.8. Técnicas estadísticas para el manejo de errores . . . . .	17
2.8.1. Métodos para corrección de errores intrínsecos . . . . .	17
2.8.2. Métodos para corrección de errores extrínsecos . . . . .	19
<b>3. Implementación</b>	<b>21</b>
3.1. Descripción del <i>hardware</i> disponible . . . . .	21
3.2. Descripción de simuladores de red revisados . . . . .	21
3.2.1. Opnet IT Guru Academic Edition . . . . .	21
3.2.2. Omnet++ . . . . .	22

3.2.3.	NCTUns 6.0 . . . . .	22
3.3.	Proceso de selección del simulador definitivo . . . . .	23
3.4.	Visión general de MiXiM/Omnet++ . . . . .	24
3.4.1.	Descripción funcional de Omnet++ . . . . .	24
3.4.2.	Simulación en Omnet++ . . . . .	24
3.4.3.	Instalación de Omnet++ . . . . .	25
3.4.4.	MiXiM . . . . .	25
3.4.5.	Descripción de los módulos de MiXiM . . . . .	26
3.4.6.	Instalación de MiXiM . . . . .	27
3.5.	Modelo de pruebas en MiXiM/Omnet++ . . . . .	28
3.5.1.	Definición . . . . .	28
3.5.2.	Implementación . . . . .	28
3.6.	Análisis de sensibilidad de los distintos parámetros . . . . .	31
3.6.1.	Método . . . . .	31
3.6.2.	Analogue Model . . . . .	32
3.6.3.	Potencia de transmisión . . . . .	32
3.6.4.	Ruido térmico . . . . .	32
3.6.5.	Coefficiente de atenuación . . . . .	32
3.6.6.	Posición de los nodos . . . . .	33
3.6.7.	Definición del caso base . . . . .	33
3.7.	Implementación de una rutina de localización en MATLAB . . . . .	34
3.7.1.	Esquema de entrada/salida del código en MATLAB . . . . .	34
3.7.2.	Descripción detallada de los bloques operativos . . . . .	34
<b>4.</b>	<b>Resultados</b>	<b>40</b>
4.1.	Resultados del análisis de sensibilidad . . . . .	40
4.1.1.	Analogue Model . . . . .	41
4.1.2.	Coefficiente de atenuación . . . . .	43
4.1.3.	Potencia de transmisión . . . . .	44
4.1.4.	Ruido térmico . . . . .	44
4.1.5.	Distribución de los nodos . . . . .	45
4.2.	Discusión de resultados . . . . .	45
4.2.1.	Correcciones al modelo de pruebas e implementación del modelo definitivo	47
4.2.2.	Resultados del modelo definitivo . . . . .	47
<b>5.</b>	<b>Conclusiones</b>	<b>56</b>
5.1.	Resultados y discusión . . . . .	56
5.2.	Trabajos futuros . . . . .	57
	<b>Bibliografía</b>	<b>58</b>
<b>A.</b>		<b>i</b>
A.1.	Código de MiXiM . . . . .	i
A.1.1.	omnet.ini . . . . .	i
A.1.2.	MyMacLayer.cc . . . . .	iv
A.1.3.	BurstApplLayer.cc . . . . .	v
A.1.4.	MyDecider80211.cc . . . . .	vi

A.1.5. MyDeciderResult80211.h . . . . .	vii
A.2. Código de Matlab . . . . .	viii
A.2.1. freeSpaceDistance.m . . . . .	viii
A.2.2. Order.m . . . . .	viii
A.2.3. linearLS.m . . . . .	xi
<b>B.</b>	<b>xiii</b>
B.1. Redes de telecomunicaciones . . . . .	xiii
B.1.1. Redes inalámbricas . . . . .	xiii
B.2. Estándares para comunicación inalámbrica . . . . .	xiii
B.2.1. IEEE 802.11 . . . . .	xiii
B.2.2. 3G . . . . .	xiv
<b>C.</b>	<b>xv</b>
C.1. Gráficos . . . . .	xv
C.1.1. Gráficos de resultados del modelo de prueba . . . . .	xv

# Índice de tablas

3.1.	Resumen de características técnicas del <i>hardware</i> . . . . .	21
3.2.	Módulos de una simulación en MiXiM . . . . .	26
3.3.	Componentes del módulo <i>Node</i> en MiXiM . . . . .	27
3.4.	Componentes del módulo <i>Nic</i> en MiXiM . . . . .	27
3.5.	Composición por módulos del nodo simulado . . . . .	30
3.6.	Estructura de <i>group</i> . . . . .	37
3.7.	Estructura de <i>data</i> . . . . .	38
3.8.	Estructura de <i>pos</i> . . . . .	38
3.9.	Estructura de <i>rsi</i> . . . . .	39
4.1.	Resultados variación del modelo de atenuación . . . . .	41
4.2.	Resultados variación coeficiente de atenuación . . . . .	43
4.3.	Resultados variación de potencia de transmisión . . . . .	44
4.4.	Resultados variación ruido térmico . . . . .	44
4.5.	Resultados desplazamiento de un nodo . . . . .	45
4.6.	Definición de los escenarios . . . . .	48

# Índice de figuras

2.1.	Efecto del retardo en la medición del tiempo de vuelta . . . . .	7
2.2.	Ubicación mediante medición de áreas . . . . .	9
2.3.	Ejemplo de dos terminales ubicados a dos saltos de distancia . . . . .	10
2.4.	Trilateración en un plano . . . . .	10
2.5.	Multilateración utilizando dos pares de referencias . . . . .	11
2.6.	Triangulación del punto O en base a los puntos conocidos A y B . . . . .	12
2.7.	Soluciones posibles en una bilateración . . . . .	14
2.8.	Ejemplo de una ambigüedad del tipo <i>flip</i> donde los puntos D y D' se encuentran a la misma distancia de las referencias A,B y C . . . . .	17
3.1.	Topología de red simple en NCTUns corriendo de Fedora 12 . . . . .	23
3.2.	Esquema de la jerarquía de los módulos de MiXiM . . . . .	26
3.3.	Proceso de instalación de MiXiM . . . . .	27
3.4.	<i>Wizard</i> para la creación de un nuevo proyecto . . . . .	28
3.5.	Elección de plantilla . . . . .	29
3.6.	Configuración del proyecto . . . . .	29
3.7.	Simulación del modelo de pruebas . . . . .	31
3.8.	Diagrama de operación de código en MATLAB . . . . .	34
4.1.	Caso base . . . . .	40
4.2.	Jakes' fading . . . . .	42
4.3.	Log-normal shadowing . . . . .	42
4.4.	$\alpha = 2.7$ . . . . .	43
4.5.	Cambio de posición del nodo . . . . .	45
4.6.	Desglose de los datos del experimento . . . . .	48
4.7.	Comportamiento de la cobertura en función de las variaciones del coeficiente de atenuación . . . . .	49
4.8.	Exactitud de cada escenario para la localización mediante RSSI . . . . .	51
4.9.	Exactitud de cada escenario para la localización mediante ToA . . . . .	52
4.10.	Exactitud de cada escenario para la localización mediante TDoA . . . . .	52
4.11.	Error máximo del método RSSI, resuelto mediante NLLS para cada simulación . . . . .	53
4.12.	Error mínimo del método RSSI, resuelto mediante NLLS para cada simulación . . . . .	54
4.13.	Error medio del método RSSI, resuelto mediante NLLS para cada simulación . . . . .	54
4.14.	Accuracy del método RSSI para una precisión del 95 %, resuelto mediante NLLS para cada simulación . . . . .	55
C.1.	$\alpha = 2.0$ . . . . .	xv

C.2. $\alpha = 3.3$ . . . . .	xvi
C.3. Potencia al 90 % . . . . .	xvi
C.4. Potencia al 110 % . . . . .	xvii
C.5. Ruido térmico al 90 % . . . . .	xvii
C.6. Ruido térmico al 110 % . . . . .	xviii

## Lista de Acrónimos

<b>RSSI</b>	Radio Signal Strength Indicator
<b>RSS</b>	Radio Signal Strength
<b>ToA</b>	Time of Arrival
<b>TDoA</b>	Time Difference of Arrival
<b>LLS</b>	Linear Least Squares
<b>NLLS</b>	Non-Linear Least Squares
<b>NLLS-LM</b>	Algoritmo de Levenberg-Marquardt
<b>LBS</b>	Location Based Services
<b>RTT</b>	Round-Trip Time
<b>AoA</b>	Angle of Arrival
<b>RLS</b>	Robust Least Squares



# Capítulo 1

## Introducción

Dentro de los últimos años se ha observado un explosivo aumento de los equipos de comunicaciones móviles en nuestro país y el mundo. La disminución en los costos de estos equipos, así como los avances tecnológicos que han elevado su poder de cómputo a un nivel comparable al de un ordenador de escritorio, han fomentado su penetración entre los usuarios nacionales. Si a lo anterior añadimos una creciente necesidad de la gente de contar con una conexión permanente a información en red y a las redes sociales que hoy pueblan Internet, se perfila un escenario donde cuesta imaginar un declive de las tecnologías inalámbricas.

Sin embargo, uno de los problemas actuales es que los servicios inalámbricos cuentan con una calidad de servicio inferior a sus contrapartes que utilizan cable: el hand-over de un terminal móvil entre dos celdas adyacentes no siempre es idóneo, produciendo interrupciones en las llamadas o sesiones de internet; un servicio de emergencia, como carabineros, no puede detectar con facilidad el origen de una llamada de alerta cuando esta se origina desde un teléfono IP; etc.

A esta falencia se suma además un área de servicios que en la actualidad no se explota plenamente: Son los llamados LBS o Location-Based Services, aplicaciones que conociendo la localización de un usuario pueden entregar información especializada, como la ubicación de sucursales de bancos a su alrededor. Además, un LBS puede hacerse cargo de los problemas planteados en el párrafo anterior: la asignación de recursos de red basados en la posición del usuario pueden mejorar significativamente su experiencia de uso; y la localización de una llamada de emergencia por parte de un servicio de socorro permitiría responder de manera más eficaz ante catástrofes.

### 1.1. Alcances y Objetivo General

Para el correcto funcionamiento de un LBS, es necesario que esté sostenido por un sistema de geo-localización que se adecue a las necesidades específicas del servicio. En la literatura existe una gama muy grande de soluciones implementadas en distintas tecnologías y con diferentes métodos de determinación de posición, siendo algunas más idóneas que otras para

ciertos escenarios. Ante esto se vuelve necesario contar con modelos que permitan evaluar el desempeño de distintas alternativas y así escoger aquella que cubra de mejor manera los requerimientos del LBS que se desea implementar.

El objetivo general del trabajo de tesis que se expone a continuación es la elección y validación de un modelo de geo-localización, mediante simulación, que constituya la base para el desarrollo de aplicaciones LBS y evaluación de los errores de localización.

## 1.2. Objetivos Específicos

Los hitos a través de los cuales se medirá el progreso del trabajo quedan determinados por los siguientes objetivos:

- Implementar la simulación de modelos de geo-localización en simuladores de red.
- Determinar la validez y pertinencia de los modelos a aplicaciones reales mediante análisis estadístico de sus características tanto de red como físicas.
- Definir o seleccionar, si existen, métricas de desempeño de modelos de geo-localización.
- Establecer la incidencia de los parámetros físicos relacionados con el modelo en el comportamiento del error de localización.

## 1.3. Descripción del documento

La estructura utilizada en este documento para exponer el trabajo realizado es la siguiente:

- **Capítulo 1. Introducción:** Corresponde a la descripción del tema, la motivación de éste y los alcances y objetivos del trabajo realizado.
- **Capítulo 2. Antecedentes:** Corresponde a la revisión bibliográfica o antecedentes. En este capítulo se explican brevemente los temas que se abordarán en las etapas de diseño e implementación.
- **Capítulo 3. Implementación:** Aquí se describen los pasos que se siguieron para obtener un modelo de simulación estable, así como la creación de la herramienta de postproceso para los datos. Se parte desde la evaluación del software utilizado y se concluye con la simulación de un modelo de pruebas y la obtención de datos a partir de él.
- **Capítulo 4. Resultados:** Es el capítulo donde se realiza una discusión de los resultados obtenidos en las etapas anteriores.
- **Capítulo 5. Conclusiones:** Se enumeran las conclusiones del trabajo realizado y se

proponen trabajos a realizar en el futuro.

- **Anexos del proyecto:** En ellos se incluyen códigos fuentes de las rutinas implementadas en MATLAB así como algunos de los módulos propios implementados en el simulador MiXiM.

# Capítulo 2

## Antecedentes

### 2.1. Locación, localización y localizabilidad

Una locación corresponde a la ubicación física de un objeto determinado. En telecomunicaciones resulta de interés el conocer la locación de distintos elementos de red, especialmente los terminales de usuario, con el fin de poder optimizar la intercomunicación entre nodos del sistema.

La localización corresponde al proceso mediante el cual se computa la locación de uno o más nodos de una red a partir de mediciones de distintos parámetros: físicos; de red; geográficos; etc.

La localizabilidad corresponde a la propiedad de una red de ser localizada de manera única e inequívoca. En general, una red con una localizabilidad alta permite aplicar métodos optimizados de localización, aumentando así su desempeño o disminuyendo sus costos [21].

### 2.2. Servicios basados en locación (LBS)

Un LBS corresponde a una aplicación conciente del entorno. En particular estas aplicaciones entregan resultados o cumplen funciones basándose en el conocimiento a priori de la locación del terminal que accede a ellas.

Se considera que el origen de los LBS ocurre en 1996, con la aparición del mandato E911 de la Comisión Federal de Comunicaciones (FCC) de los Estados Unidos [10]. El mandato ordena a los operadores de telefonía móvil poder ubicar con precisión el origen de las llamadas recibidas por el número de emergencia 911. El cumplimiento de dicho mandato resultó en una inversión muy elevada, por lo que los operadores introdujeron los primeros LBS comerciales con el fin de poder recuperar parte de su capital [7].

En la actualidad los LBS se han visto potenciados debido a la alta penetración de

terminales móviles entre la población, así como la tendencia de los usuarios de estar permanentemente conectados e informados. [4] Ejemplos de estos servicios son: la ubicación de llamadas de emergencia, como el ya mencionado E911; la asignación de recursos de red basados en proximidad, con el objetivo de hacer la comunicación más eficiente [23] [16]; y brindar soporte a la logística mediante control de inventarios y bodegas [21].

## **2.3. Localización basada en redes y aplicaciones**

Es posible explotar la naturaleza de las redes fijas de telecomunicaciones para poder localizar equipos conectadas a ellas mediante métodos que no involucran la medición de parámetros físicos. Estas técnicas se basan en el conocimiento a priori de la distribución física de la red.

### **2.3.1. Localización mediante servidores DHCP**

Dentro de una red de computadores, el servidor DHCP es el encargado de ajustar los parámetros de red de un equipo para permitir su comunicación con otros nodos. Una serie de estándares se han propuesto para que junto con esta información se envíe además la ubicación del equipo que se está conectando, ya sea en formato de coordenadas físicas [26] o bien como dirección civil [31]. Ambos estándares definen el formato de un mensaje que es enviado por el servidor al terminal de usuario, donde queda almacenada la información de la ubicación para su posterior uso por parte de alguna aplicación.

Para una localización eficiente, es necesario que el servidor DHCP tenga la capacidad de asociar puntos de acceso de la red con posiciones o direcciones. Esto se puede lograr mediante el uso de bases de datos construidas a priori y almacenadas en el servidor. Para redes de gran tamaño este trabajo puede resultar demasiado complejo, quedando la opción de entregar la ubicación del servidor DHCP más cercano en lugar del terminal de interés, perdiendo así precisión.

### **2.3.2. Localización mediante servidores de información de locación (LIS)**

Un LIS es un nodo dentro de una red que utiliza técnicas de mapeo de dispositivos y bases de datos para proveer información de locación a todos los terminales conectados a la misma red de acceso. La localización mediante estos aparatos se logra mediante el uso del protocolo HELD [6] que comunica al terminal con el LIS.

Respecto a la localización utilizando servidores DHCP, este método presenta diferencias únicamente en los equipos utilizados. El LIS es un dispositivo especializado que se encarga exclusivamente de proveer información de locación la cual puede ser solicitada en cualquier

momento por un terminal, a diferencia del uso de DHCP donde la información es enviada desde el servidor en el momento en el cual el equipo se conecta a la red.

## 2.4. Técnicas de medición para localización en redes inalámbricas

Dentro del proceso de localización es necesario destacar por lo menos dos categorías de nodos de red: se tienen los terminales de usuario, cuya ubicación se desea conocer; y nodos faro o ancla, que comunmente corresponden a nodos de la infraestructura de la red en estudio y cuya posición se conoce a priori.

Para la determinación de la locación de un terminal es necesario un conocimiento previo del entorno físico en el cual se encuentra. [21] Por esto todo proceso de localización se inicia con la medición de alguna variable que permita estimar la posición del elemento de red deseado. Esta estimación en ningún caso da una ubicación absoluta, si no más bien relativa a algún nodo ancla involucrado en la medición.

Dentro de las técnicas de medición existe una división ampliamente aceptada [27]: métodos basados en alcance, en los cuales la medición física se traduce en una distancia o una dirección respecto a un nodo de referencia; y métodos basados en proximidad, donde la localización se basa en la proximidad o conectividad entre el nodo a localizar y el punto ancla.

### 2.4.1. Técnicas basadas en alcance

#### Intensidad de señal de radio (RSS)

Este método se basa en el conocimiento de que la intensidad de señal disminuye a medida que esta se propaga hacia el receptor. En una primera aproximación la distancia se obtiene a partir de la medición de la intensidad de señal en el terminal del usuario, y con el uso de la ecuación de propagación en espacio libre:

$$P_r(d) = \left( \frac{\lambda}{4\pi d} \right)^2 P_t G_t G_r \quad (2.1)$$

Donde  $P_r$  es la potencia recibida,  $G_t$  y  $G_r$  corresponden a las ganancias de las antenas y  $\lambda$  es la longitud de onda de la señal transmitida.

Al aplicar este modelo en escenarios reales se obtienen errores de hasta un 50% [27] debido a factores tales como la multiplicidad de trayectorias, ausencia de línea vista y ruido. Sin embargo, mediante la inclusión de nuevos términos en la ecuación 2.1 es posible mejorar significativamente los resultados de este método. [8] [33]

## Tiempo de llegada (ToA)

Para una señal de velocidad conocida, la determinación del tiempo de propagación permite calcular la distancia entre emisor mediante una simple ecuación de movimiento:

$$d = v(t_i - t_o) \quad (2.2)$$

Donde  $t_i - t_o$  corresponde al tiempo de propagación.

Considerando la alta velocidad de propagación de las ondas de radiofrecuencia y con el objetivo de obtener una buena estimación de la distancia, es necesario que los relojes de todos los equipos de la red se encuentren sincronizados al nivel de nanosegundos [4]. Dicho ajuste puede resultar muy costoso para el operador de red, por lo que existe otro enfoque independiente de la sincronización.

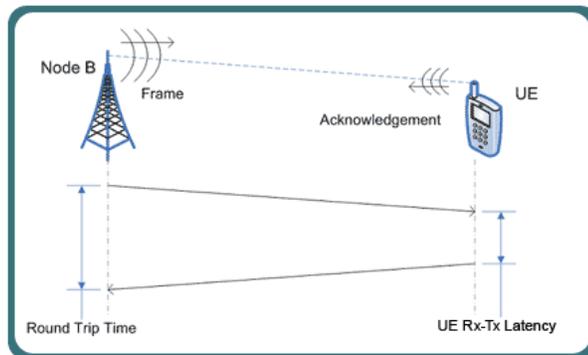
En lugar de tomar el tiempo de vuelo en una dirección es posible realizar el cálculo con el tiempo de vuelta o *round-trip time* de la señal, es decir el tiempo que transcurre entre la emisión de la señal y la recepción por parte del emisor de una señal de confirmación del receptor. En este caso la ecuación que permite determinar la distancia es:

$$d = v \frac{(t_{RT} - t_d)}{2} \quad (2.3)$$

Donde  $t_{RT}$  es el tiempo de vuelta y  $t_d$  es un retardo introducido por el procesamiento de la señal en el equipo receptor.

El uso de este método en conjunto con tecnologías de comunicación celular como GSM o UTMS permite la localización con errores desde las decenas a las centenas de metros [9]; mientras con el uso de Wi-Fi, los resultados se ven severamente perjudicados por la baja precisión de reloj del *hardware* típico de esta tecnología [22].

Figura 2.1: Efecto del retardo en la medición del tiempo de vuelta



## Diferencia de tiempo de llegada (TDoA)

Para aquellos casos en los que se cuenta con más de un nodo de referencia, se abre la posibilidad de una nueva técnica basada en el tiempo de vuelo. En la aplicación de TDoA el nodo a ser ubicado actúa como transmisor, enviando una señal que es captada por 2 o más nodos de referencia. Luego el tiempo de llegada a estos puntos que se denominarán  $i$  y  $j$  se relaciona con la distancia mediante la ecuación:

$$\Delta t_{ij} = (t_i - t_o) - (t_j - t_o) = t_i - t_j = \frac{1}{v} (\|r_i - r_t\| - \|r_j - r_t\|) \quad (2.4)$$

Donde  $t_o$  es el tiempo de emisión de la señal y  $r_t$  es la posición del transmisor. A diferencia de lo que ocurre en ToA, aquí sólo es necesario que los receptores se encuentren sincronizados. Sin embargo se requiere que estos mismos posean una gran capacidad de cómputo para poder determinar correctamente la locación del transmisor. Por estos motivos, este método es muy útil en redes con infraestructuras poderosas, tales como las de telefonía celular [21].

## Ángulo de llegada (AoA)

La técnica de ángulo de llegada se basa en el principio geométrico de triangulación: conocida la distancia entre dos vértices y sus ángulos respecto al tercero de ellos es posible determinar un único triángulo. Como en el caso de TDoA, se requiere contar con al menos dos nodos ancla.

La determinación del ángulo de llegada se puede realizar mediante la ecuación 2.1, asumiendo que la ganancia tanto de emisor como receptor son conocidas  $G = G(\Theta)$ ; o bien mediante un arreglo de antenas ubicadas en cada nodo que permitan determinar la dirección de llegada en base a los desfases percibidos por cada una [21] [4].

Debido a la necesidad de contar con gran número de artefactos en cada terminal de la red, la implementación de AoA resulta más costosa que la de sus contrapartes, a la vez que se dificulta mucho cuando el tamaño de los receptores es pequeño.

### 2.4.2. Técnicas basadas en proximidad

En estos métodos no existe un cálculo explícito de distancia, si no más bien condiciones de conectividad o proximidad de nodos. Si bien estas técnicas no alcanzan el nivel de precisión de sus contrapartes basadas en alcance, su bajo costo y fácil implementación las han convertido en una opción para aplicaciones donde no se requiere de una localización demasiado exacta [4].

Una aplicación real de las técnicas que se describen a continuación puede encontrarse en [11].

## Medición de Áreas

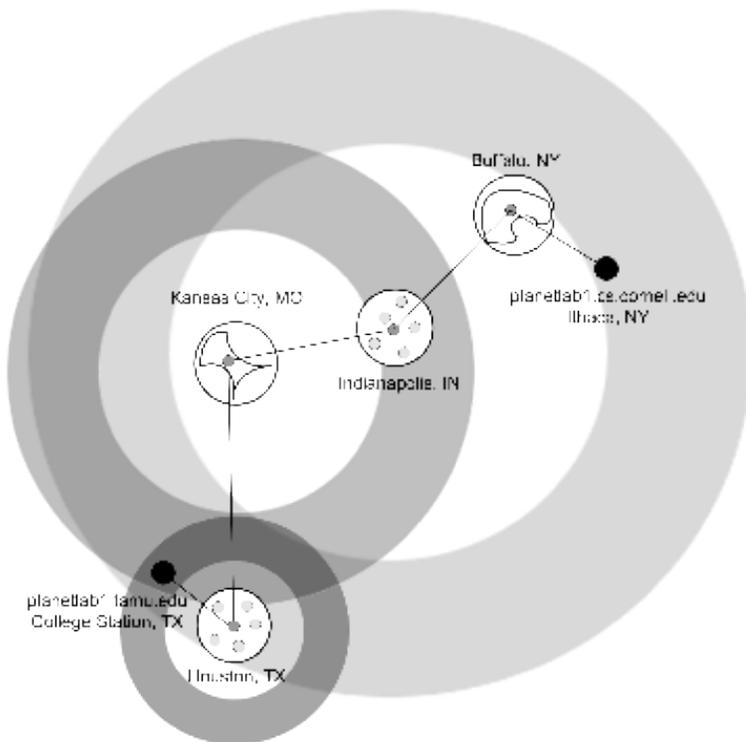
En los casos en que la señal de radio u otra variable pueda ser confinada a un área geométrica, dicha área puede utilizarse para dar una estimación gruesa de la ubicación de un terminal. En general, el método consiste en computar la intersección de diversas áreas de cobertura y tomar su centroide como la posición del nodo requerido. [21]

Un ejemplo clásico es suponer que se conocen el radio máximo y mínimo de cobertura de una serie de antenas,  $R_{i_{max}}$  y  $R_{i_{min}}$  respectivamente, cuya ubicación es conocida con anterioridad. Luego si el nodo buscado posee conectividad con la antena  $i$ , su distancia respecto a dicha antena  $d_i$  será tal que cumplirá con la relación:

$$R_{i_{min}} \leq d_i \leq R_{i_{max}} \quad (2.5)$$

Al considerar estas restricciones para un número creciente de antenas, se obtendrá un área cada vez más reducida que contendrá el nodo buscado.

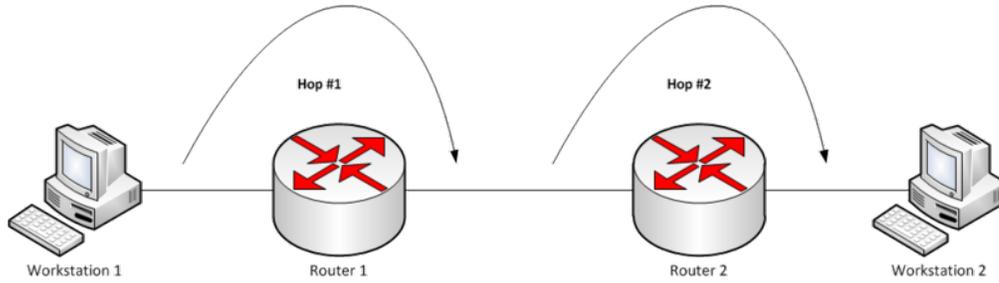
Figura 2.2: Ubicación mediante medición de áreas



## Conteo de Saltos (Hop-Count)

Al poseer la información de conectividad de una gran número de elementos, es posible construir un grafo cuyos vértices son nodos dentro de la red y cuyos arcos corresponden a la presencia de conectividad entre un par de puntos del sistema.

Figura 2.3: Ejemplo de dos terminales ubicados a dos saltos de distancia



La distancia en saltos  $h_{ij}$  entre dos nodos  $n_i$  y  $n_j$  será la distancia del camino de arcos más corto que une ambos nodos. Si se posee una red isotrópica, entonces la distancia real entre ambos nodos  $d_{ij}$  se aproximará bastante a la distancia en saltos. [21]

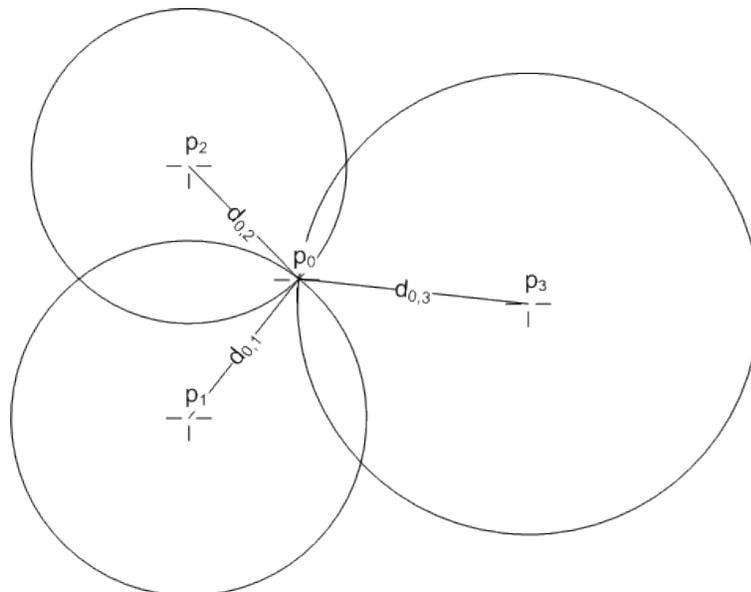
$$d_{ij} \approx h_{ij} \quad (2.6)$$

## 2.5. Métodos matemáticos para localización

### 2.5.1. Multilateración

La multilateración es el proceso de ubicar la intersección de un número arbitrario de superficies esféricas de las que se conocen sus centros y radios. Dicha intersección será única mientras los centros no sean colineales [21]. En el caso específico que se trate de la intersección de 3 esferas, el proceso recibirá el nombre de trilateración.

Figura 2.4: Trilateración en un plano



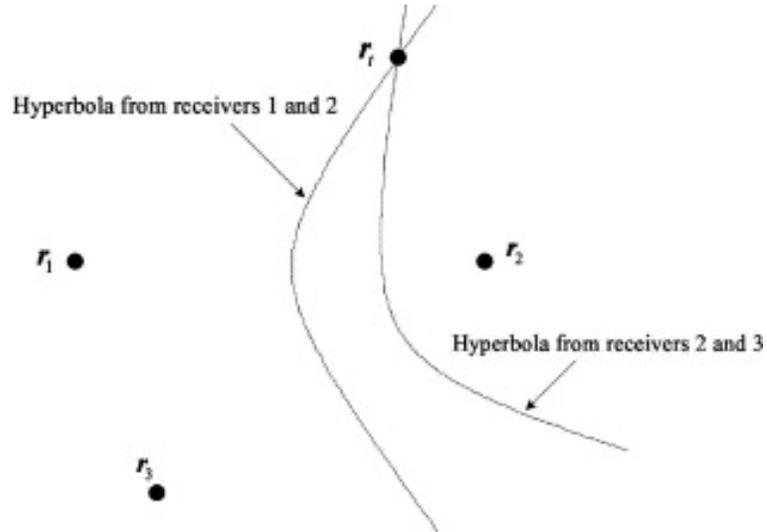
La trilateración se utiliza generalmente con mediciones de RSS o ToA. Definiendo  $r_1, r_2$  y  $r_3$  como la distancia desde un nodo que se desea ubicar a tres referencias conocidas, la locación del terminal en cuestión puede obtenerse resolviendo el sistema de ecuaciones:

$$\begin{aligned} r_1^2 &= (x_o - x_1)^2 + (y_o - y_1)^2 + (z_o - z_1)^2 \\ r_2^2 &= (x_o - x_2)^2 + (y_o - y_2)^2 + (z_o - z_2)^2 \\ r_3^2 &= (x_o - x_3)^2 + (y_o - y_3)^2 + (z_o - z_3)^2 \end{aligned} \quad (2.7)$$

Donde  $x_i, y_i, z_i$  representan las coordenadas de la referencia  $i$ , y  $x_o, y_o, z_o$  las coordenadas del punto que se desea ubicar.

Cuando se cuenta con mediciones de TDoA, resulta más conveniente utilizar multilateración con al menos cuatro nodos de referencia. En este caso, se fija el sistema de referencia en uno de los nodos  $(x_4, y_4, z_4) = (0, 0, 0)$  y se utilizan las mismas ecuaciones planteadas en el sistema 2.7. Para el desarrollo matemático se considerará que la distancia  $d_i = vt_i$ ; donde  $v$  es la velocidad de la señal transmitida.

Figura 2.5: Multilateración utilizando dos pares de referencias



Ahora, restando una de las ecuaciones a todas las demás se puede obtener un nuevo sistema:

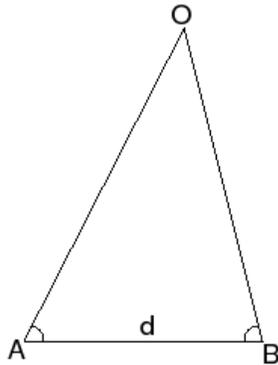
$$\begin{aligned} t_{14} = t_1 - t - 4 &= \frac{\sqrt{(x_1 - x_o)^2 + (y_1 - y_o)^2 + (z_1 - z_o)^2} - \sqrt{(x_o)^2 + (y_o)^2 + (z_o)^2}}{v} \\ t_{24} = t_2 - t - 4 &= \frac{\sqrt{(x_2 - x_o)^2 + (y_2 - y_o)^2 + (z_2 - z_o)^2} - \sqrt{(x_o)^2 + (y_o)^2 + (z_o)^2}}{v} \\ t_{34} = t_3 - t - 4 &= \frac{\sqrt{(x_3 - x_o)^2 + (y_3 - y_o)^2 + (z_3 - z_o)^2} - \sqrt{(x_o)^2 + (y_o)^2 + (z_o)^2}}{v} \end{aligned} \quad (2.8)$$

Cada una de estas ecuaciones representa una hipérbola, y la intersección de todas ellas corresponde a la posición del terminal a ubicar. Debido a esto, la multilateración también recibe el nombre de posicionamiento hiperbólico [4]. En general, aumentar el número de nodos ancla aumenta la precisión del método [22].

### 2.5.2. Triangulación

La triangulación se basa en el principio de la geometría plana que establece que conociendo dos ángulos y un lado de un triángulo, este queda definido de manera única. A diferencia de la multilateración y la trilateración, en este método no existen mediciones de distancias, sino de ángulos. Por esto es la opción lógica para trabajar con mediciones de AoA. Una comparación más detallada entre la multilateración y la triangulación se encuentra en [4].

Figura 2.6: Triangulación del punto O en base a los puntos conocidos A y B



Las ecuaciones que permiten obtener la locación desconocida corresponden a la conocida ley del seno:

$$\frac{A}{\sin a} = \frac{B}{\sin b} = \frac{C}{\sin c} \quad (2.9)$$

Y a la ley del coseno [27]:

$$\begin{aligned} C^2 &= A^2 + B^2 + 2AB\cos c \\ B^2 &= A^2 + C^2 - 2AC\cos b \\ A^2 &= B^2 + C^2 + 2BC\cos a \end{aligned} \quad (2.10)$$

Donde  $A, B, C$  corresponden a los lados del triángulo y  $a, b, c$  corresponden a los ángulos opuestos a dichos lados.

### 2.5.3. Perfil RSS

Este método consiste en dar otro uso a las mediciones de RSS: en lugar de estimar distancias mediante la ecuación de espacio libre (2.1); se comparan los valores RSS con un perfil de intensidad de señal de la zona donde se ubica la red [8]. Dependiendo de la forma en la cual se crea este perfil se dirá que se trata de un mapa estático; o un mapa en línea: un perfil estático ocupará datos de intensidad fijos, obtenidos una única vez de manera previa a la operación de la red; mientras uno en línea utilizará los datos entregados por una serie de nodos de referencia de manera simultánea a la operación de la red. Este último enfoque tiene la virtud de permitir una localización efectiva ante cambios dinámicos de RSS. [21].

En ambos casos, un sistema central almacenará un vector  $(x_i, y_i, r_{i1}, r_{i2}, \dots, r_{in})$  donde  $r_{ij}$  corresponde a la intensidad de señal recibida del nodo  $j$  en la posición definida por  $(x_i, y_i)$ . Luego el nodo a ser localizado elegirá su posición a partir de aquel vector de mediciones RSS que más se asemeje al suyo. Esta similitud se puede obtener a partir de métodos determinísticos [5] o probabilísticos [15].

## 2.6. Algoritmos de Localización

Un algoritmo de localización es el proceso iterativo mediante el cual se busca localizar una red completa, en lugar de un único terminal. Estos procesos, al igual que las técnicas en las que se basan se podrán categorizar en algoritmos basados en alcance; y basados en conectividad.

### 2.6.1. Algoritmos basados en distancia

Dentro de esta categoría se pueden diferenciar dos enfoques distintos para resolver el problema de la localización: el centralizado, en el cual las mediciones físicas son enviadas por los nodos a una estación base para su análisis; y el distribuido, donde cada nodo calcula su ubicación de manera paralela a otros terminales. La primera metodología permite utilizar nodos con baja capacidad de cálculo, pero carga la red con un gran flujo de datos que viaja desde y hacia la estación base; mientras, un algoritmo distribuido utiliza la comunicación internodos para suplir su escaso poder de procesamiento, utilizando una gran cantidad de nodos en cálculos simples [21].

#### Algoritmos centralizados

La literatura presenta una serie de soluciones al problema de localización de una red mediante el uso de algoritmos centralizados. La gran mayoría de ellos corresponde a una mejora respecto a dos métodos básicos: MDS [32] y SDP [14].

El MDS o *Multidimensional scaling* es un método basado en el conocimiento de la distancia

de cada par posible dentro de un conjunto de  $n$  puntos, cuyas ubicaciones son desconocidas. En base a este conocimiento y mediante el uso de álgebra lineal y la ley de los cosenos es posible crear un mapa multidimensional de las posiciones relativas de todos los terminales. Posteriormente este mapa es pasado a coordenadas absolutas basandose en las coordenadas de un conjunto de nodos de referencia.

SDP o *Semidefinite programming* corresponde a una generalización de la programación lineal. Se utiliza una serie de restricciones geométricas entre nodos, representadas como inecuaciones de matrices lineales (LMI). El conjunto de LMI forma un único problema de programación semidefinida que es resuelto para encontrar una región acotada en la cual puede encontrarse cada nodo.

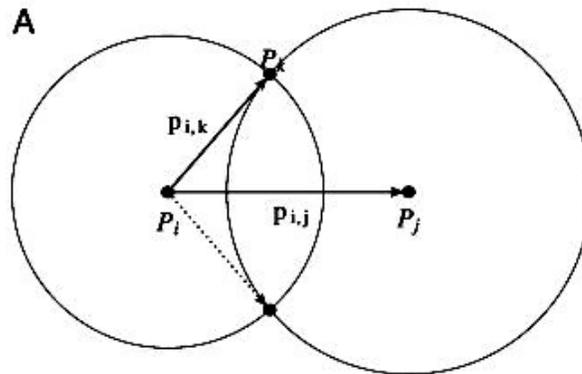
### Algoritmos distribuidos

Para el caso de los algoritmos distribuidos la mayoría de los métodos cae dentro de tres grandes categorías: Trilateración sucesiva, bilateración y unión de sistemas de coordenadas.

La trilateración sucesiva corresponde a la ubicación de nodos mediante la técnica ya descrita en la sección 2.5. Se comienza mediante la ubicación de terminales que poseen información sobre tres o más nodos de referencia. Una vez que estos nodos son localizados, entran a una nueva iteración del método como nodos ancla, permitiendo la localización de nodos cada vez más alejados de las referencias originales. Este algoritmo es extremadamente susceptible a la propagación de errores [21].

La bilateración corresponde a un caso particular de multilateración donde solo se usa información de dos nodos de referencia. Así, el método entrega dos posibles soluciones para la posición de cada terminal. Esto lleva a la construcción de un árbol con un total de  $2^n$ , donde  $n$  es el número de nodos a ubicar. Las soluciones son descartadas en la medida que los datos de distintos pares de referencias resultan incompatibles [21].

Figura 2.7: Soluciones posibles en una bilateración



La unión de sistemas de coordenadas es un enfoque de localización *bottom-up* donde primero se determina la posición relativa de una pequeña región de la red. Posteriormente, se

utilizan nodos comunes entre distintas regiones para poder unirlos en segmentos más grandes, hasta tener una solución única de toda la red [21].

## 2.6.2. Algoritmos basados en conectividad

Si bien para una red isotrópica el método basado en conteo de saltos entrega buenos resultados [24]; su desempeño baja considerablemente para sistemas que no cumplen con esta condición. Por esto, existe una serie de algoritmos especializados para lograr la ubicación de nodos al interior de redes anisotrópicas como por ejemplo: PDM y REP [21].

## 2.6.3. Precisión y Exactitud

Ante la gran variedad de algoritmos presentes para localización, se hace necesario contar con un indicador de desempeño para los mismos. En [21] se definen dos conceptos que permiten comparar métodos: precisión y exactitud (*accuracy*).

Se entenderá por exactitud la distancia euclidiana esperada entre la posición real y la estimada de un nodo al interior de un sistema. Por su parte, la precisión corresponde al porcentaje sobre el total de terminales que cumple con la exactitud esperada.

En general, el método utilizado para comparar dos algoritmos consiste en fijar una precisión y luego calcular la exactitud necesaria de cada método para cumplir con el requisito deseado de precisión.

## 2.7. Errores de localización

### 2.7.1. Cota inferior de Cramér-Rao

Esta cota se trata de un resultado clásico de la estadística que establece un valor mínimo para la varianza de un estimador [12]. En su forma más simple, la cota establece que la varianza de un estimador insesgado es al menos tan alta como el inverso de la información de Fisher. Un estimador que alcanza esta cota se define como un estimador eficiente, es decir, tiene el menor error cuadrático medio de entre todos los estimadores.

Una propiedad útil de esta cota es que su cálculo no depende de ningún método de estimación. Solo hace falta contar con la distribución estadística de la variable aleatoria, es decir  $f(X|\theta)$  donde  $X$  son las observaciones de la variable aleatoria y  $\theta$  es el parámetro a estimar. Así cualquier estimador  $\hat{\theta}$  debe cumplir que [21]:

$$Cov(\hat{\theta}) \geq \{-E[\nabla_{\theta}(\nabla_{\theta} \ln f(X|\theta))^T]\}^{-1} \quad (2.11)$$

Donde  $Cov(\hat{\theta})$  es la covarianza del error del estimador,  $E[...]$  es la esperanza y  $\nabla_{\theta}$  es el gradiente respecto a  $\theta$ .

### 2.7.2. Errores intrínsecos

Dentro de la clasificación de errores de medición, se llamará intrínseco a todo aquel que provenga directamente de limitaciones técnicas del equipo o del programa de adquisición de datos. Este tipo de error se encuentra tipificado para las tecnologías de localización más comunes [21], y si bien suele ser bajo resulta un problema serio cuando se utilizan algoritmos iterativos de localización.

### 2.7.3. Errores extrínsecos

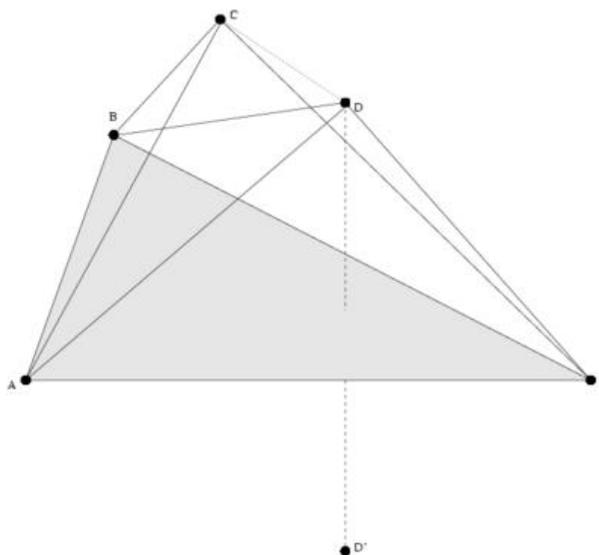
Estos errores corresponden a todos aquellos causados por factores físicos que afecten el canal de propagación o la velocidad de transmisión de la señal. Además caerán en esta categoría las fallas imprevistas del equipo involucrado así como los ataques externos al sistema de localización. [21]

### 2.7.4. Impacto de errores en algoritmos de localización

Los errores de medición dificultan la tarea de localizar una red en alguna de las siguientes cuatro maneras:

1. Incertidumbre: Representa el caso en que no existen un punto geométrico que satisfaga todas las condiciones impuestas por las mediciones de distancia.
2. Inconsistencia: En los casos en que un nodo se encuentra en el rango de múltiples nodos ancla, es posible determinar su posición utilizando algún subconjunto de ellos, sin embargo cuando distintos grupos entregan distintas ubicaciones se está ante un problema de inconsistencia.
3. Ambigüedad: Es producida por ambigüedades discontinuas en los grafos que representan el sistema de comunicación. Estas corresponden a casos de *flip* y *flex*.

Figura 2.8: Ejemplo de una ambigüedad del tipo *flip* donde los puntos D y D' se encuentran a la misma distancia de las referencias A, B y C



4. Propagación de errores: en los casos en los que la localización viene de una iteración de multilateraciones, el error del primer paso se acumula y es transmitido a los nodos ubicados durante los pasos posteriores.

## 2.8. Técnicas estadísticas para el manejo de errores

### 2.8.1. Métodos para corrección de errores intrínsecos

Debido a que la localización de un sistema generalmente se hace de manera iterativa, un nodo mal ubicado propaga su error a otros terminales cuya posición se obtiene a partir de él. Un método efectivo para solucionar este problema es el refinamiento de la locación [13].

#### Método de mínimos cuadrados

Corresponde al algoritmo clásico de implementación de refinamiento de locación [21]. El proceso consiste en que los nodos una vez ubicados comienzan a actualizar su posición en diversas rondas. Todos los nodos transmiten su ubicación a sus vecinos y luego con estos datos proceden a realizar una multilateración basada en minimizar el error cuadrático medio. Una vez que las actualizaciones son menores a cierto umbral se detiene el proceso.

El mayor problema con la implementación de este método es que no existe claridad sobre las condiciones necesarias para asegurar convergencia hacia la solución real ni qué tan precisa será la solución obtenida. Es por esto que [21] define este enfoque como refinamiento sin control de error. En la literatura se proponen soluciones que sí poseen control de error.

## Método del valor de confianza

Propuesto en [30], el método consiste en agregar un ponderador al método clásico de refinamiento que represente un valor de certeza en la solución obtenida. La multilateración simple consiste en la resolución de un sistema de ecuaciones lineales  $Ax = b$ , mientras que en esta versión el problema se transforma en:

$$\omega Ax = \omega b; \omega \in (0, 1) \quad (2.12)$$

Donde  $\omega$  es un vector con valores de confianza. Este valor será 1 para los nodos de referencia y muy cercano a cero para los terminales con locaciones indefinidas. En cada proceso de multilateración el nodo obtendrá una confianza igual al promedio de confianza de los terminales que fueron utilizados para ubicarlo, y solo actualizará su posición si el resultado entrega un valor de confianza superior al que este ya poseía.

## Método de mínimos cuadrados robustos (Robust-LS)

Propuesto en [20], el método reemplaza el estimador de mínimos cuadrados  $\hat{x}_t = \arg \min_x |(Ax - b)|^2$  por un estimador que incluye la incertidumbre de localización, representada como una perturbación:

$$\hat{x}_t = \arg \min_x |(A + \Delta A)x - (b + \Delta b)|^2 \quad (2.13)$$

Este enfoque aumenta la estabilidad del algoritmo básico para los casos en que la matriz  $A$  sea singular o mal condicionada.

## Calidad de Trilateración (QoT)

Como se presenta en [35], este método guarda similitudes con el valor de confianza. En este caso los autores sugieren una forma de determinar la calidad de una trilateración mediante métodos estadísticos. Esta calidad tiene directa relación con la precisión de la estimación que a su vez está ligada con la distribución espacial de los nodos ancla.

Además asignan un valor de confianza para cada nodo, pero a diferencia de [30] este valor no corresponde a un promedio sino que es entregado por la expresión:

$$C_t(s) = Q(t) \prod_{i=1}^3 C(s_i) \quad (2.14)$$

Donde  $Q(t)$  es la calidad de la trilateración y  $C(s_i)$  es la confianza de los nodos usados en

dicho proceso. Así el método no solo usa la confiabilidad de un nodo para ubicar, sino también su distribución geográfica evitando el uso de configuraciones que empeoren los resultados.

## 2.8.2. Métodos para corrección de errores extrínsecos

Comparativamente con los errores intrínsecos, este tipo de errores son más impredecibles y causados por factores externos de manera no sistemática. Las principales causas de estos errores son: fallas en los equipos de medición o una calibración deficiente, por ejemplo la falta de sincronía entre relojes de dos terminales para mediciones de ToA; factores ambientales, como las reflexiones e interferencia que modifican la amplitud de la señal para mediciones de RSS, o cambios del medio de transmisión que afecten la velocidad de la señal y no permitan una correcta medición de TDoA; y los ataques de usuarios maliciosos, los que pueden volverse más frecuentes en la medida en que aumente la popularidad de los LBS. [21].

Estos errores pueden considerarse como mediciones anómalas que escapan a la distribución de las demás mediciones. Los métodos existentes para combatir estos problemas se separan en dos grandes grupos: depuración explícita y desestimación implícita.

### Depuración explícita

Los métodos agrupados bajo esta categoría se basan en la intuición de que las mediciones erradas no tendrán similitud con el gran grueso de mediciones correctas. Así mediante una revisión de las inconsistencias es posible identificar y destacar las mediciones con errores.

Como se muestra en [21] es posible utilizar un estimador de mínimos cuadrados para identificar las inconsistencias. Suponiendo un modelo del error de acuerdo a la siguiente ecuación:

$$\xi^2 = \frac{1}{m} \sum_{i=1}^m (\delta_i - |\bar{p}_o - p_i|)^2 \quad (2.15)$$

Donde  $\delta_i$  es la distancia medida a la referencia  $i$ ,  $\bar{p}_o$  es la posición estimada del nodo objetivo y  $p_i$  es la posición de la referencia  $i$ . Se considerará que un set de posiciones es consistente cuando el error cuadrático es menor a cierto umbral que debe ser escogido en función del modelo de error [21].

### Desestimación implícita

Este enfoque se basa fuertemente en principios de estadística robusta. Los métodos de la estadística clásica se basan en una serie de suposiciones que muchas veces no se cumplen en sets de datos reales: por ejemplo asumir una distribución normal de datos. En general, estos métodos empeoran su desempeño ante la presencia de mediciones anómalas. La estadística

robusta es un campo teórico que se ocupa del problema de mediciones anómalas mediante el uso de estimadores con una baja sensibilidad ante las variaciones de los datos.

Uno de los enfoques más utilizados es el de punto de quiebre de un estimador. Este concepto hace referencia al porcentaje de los datos que pueden tomar valores arbitrariamente grandes sin que el estimador tome un valor arbitrariamente grande. Por ejemplo el punto de quiebre del estimador promedio  $\frac{1}{n}(x_1 + x_2 + \dots + x_n)$  es cero, pues basta que un único dato tienda a infinito para que el valor del estimador tienda a infinito. En este contexto, es necesario contar con estimadores con un alto punto de quiebre, que permita disminuir el efecto de las mediciones anómalas dentro de la estimación.

En [29] se propone el uso del estimador de mínimos cuadrados medianos:

$$\bar{p}_o = \arg \min_{p_o} \text{med}_i(\delta_i - ||p_o - p_i||)^2 \quad (2.16)$$

# Capítulo 3

## Implementación

### 3.1. Descripción del *hardware* disponible

El equipo disponible y que será utilizado para el desarrollo de este trabajo consiste en dos computadores personales de propiedad del autor, cuyas especificaciones técnicas más relevantes se encuentran descritas en la tabla.

	Computador 1	Computador 2
Modelo	Sony Vaio VGN-NR150FE	Samsung RF411
Procesador	Intel Core 2 Duo T5250 @1.5 Ghz	Intel Core i5-2430M @2.4 GHz
RAM	2,00 GB	6,00 GB
Sistema Operativo	Ubuntu Linux 11.10 (32 bits)	Windows 7 Home Basic, Service Pack 1 (64 bits)

Tabla 3.1: Resumen de características técnicas del *hardware*

### 3.2. Descripción de simuladores de red revisados

#### 3.2.1. Opnet IT Guru Academic Edition

- *Distribuidor*: Opnet.
- *Sitio web*: [https://www.opnet.com/university\\_program/itguru\\_academic\\_edition/](https://www.opnet.com/university_program/itguru_academic_edition/)
- *Método de adquisición*: Obtención de licencia gratuita y descarga del programa desde su página web.
- *Sistema operativo*: Windows NT 4.0, 2000, XP y Vista.
- *Capacidad de expansión*: Programa de código cerrado. Debido a esto no es posible

modificar la funcionalidad original del programa o añadir nuevas características. La única posibilidad de obtener más prestaciones es mediante la compra de una licencia comercial de Opnet IT Guru.

- *Apreciación:* Posee un enfoque muy simple, ideal para la simulación de problemas básicos de redes y como complemento a la docencia de cursos universitarios. El hecho de que su código no pueda ser modificado lo pone por detrás de los demás simuladores estudiados a la hora de buscar una herramienta versátil para la investigación o el desarrollo de proyectos. Es necesario acotar que a pesar de que el distribuidor no lo señala en su página, el programa es compatible con Windows 7 en su versión de 64 bits.

### 3.2.2. Omnet++

- *Distribuidor:* The Omnet++ Community (desarrollado originalmente por Andrés Varga).
- *Sitio web:* <http://www.omnetpp.org/>
- *Método de adquisición:* Descarga gratuita desde sitio web oficial.
- *Sistema operativo:* Linux, Mac OS X y Windows (para el uso del IDE se requiere al menos Linux 32/64, Mac OS X 10.5 o Windows XP).
- *Capacidad de expansión:* Omnet++ no es un simulador de red por sí mismo, sino más bien un *framework* para el desarrollo de simuladores. En consecuencia es completamente modificable y expandible, teniendo como lenguaje base para su programación C++.
- *Apreciación:* Dentro del *framework* es posible encontrar una gran serie de proyectos mantenidos por diferentes grupos y enfocados a distintos tipos de redes de comunicaciones tales como redes TCP/IP, móviles, de sensores inalámbricos, de vehículos, etc. Cuenta con un amplio repositorio de información, entre los que destacan una wiki y un mailing list donde los desarrolladores participan activamente. Los puntos bajos que presenta frente a los otros programas revisados son dos, por un lado su carácter de software gratuito lleva a que la corrección de *bugs* y errores resulte un poco lenta, y por otra parte no es tan mencionado en publicaciones científicas como sus competidores.

### 3.2.3. NCTUns 6.0

- *Distribuidor:* SimReal Inc. (Network and System Laboratory, Department of Computer Science, National Chiao Tung University)
- *Sitio web:* <http://nsl10.csie.nctu.edu.tw/>
- *Método de adquisición:* El sitio web oficial ya no ofrece la descarga de NCTUns, en

lugar de eso es posible adquirir la versión pagada del programa llamada EstiNet. La versión gratuita se distribuye entre usuarios que descargaron la versión gratuita antes de que fuese eliminada del repositorio o a través de sitios de descarga alternativos.

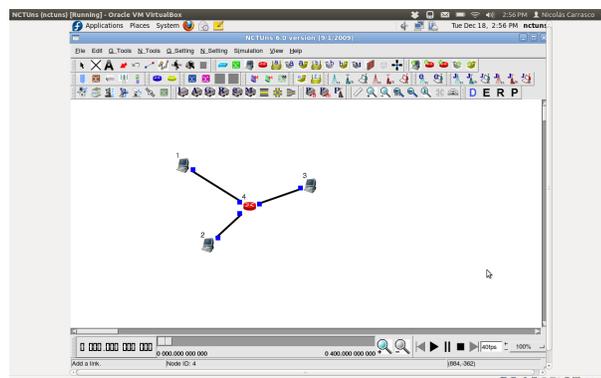
- *Sistema operativo:* Fedora 12. Es posible modificar el programa de modo de instalarlo en otras distribuciones de Linux.
- *Capacidad de expansión:* Es posible incluir nuevos módulos al programa. Sin embargo se requiere recompilar el mismo para completar esta tarea. El lenguaje de programación de los módulos es C++, al igual que en el caso de Omnet++.
- *Apreciación:* Se trata de un programa con una muy variada gama de prestaciones y cuyo principal atributo es el uso de la pila TCP/IP real de la máquina en la cual se esta ejecutando para dar mayor verosimilitud a sus resultados. Además, un gran número de publicaciones avalan su efectividad y resaltan sus características. Su principal defecto respecto a sus competidores es la dificultad que se presenta al tratar de obtener una licencia gratuita del programa. Adicionalmente, es de esperar que con el paso a una versión comercial, el soporte que recibe la versión gratuita se vea disminuido con el pasar del tiempo.

### 3.3. Proceso de selección del simulador definitivo

Luego de una simple inspección de la versión gratuita de Opnet IT Guru, se observó que si bien el programa cuenta con bastantes opciones interesantes, no se acerca a los requerimientos necesarios para el desarrollo del trabajo realizado en esta tesis, siendo descartada esta opción rápidamente.

A continuación se configuró un equipo para la instalación y uso de NCTUns 6.0 en el mismo. Luego de la instalación, se realizaron algunas pruebas exitosas con el simulador (ver fig. 3.1) pero sin embargo rápidamente se llegó a una piedra de tope: las variables de interés para la localización, tales como RSS o RTT, no pueden medirse de manera directa, siendo necesario modificar inmediatamente los módulos originales del programa para alcanzar esta funcionalidad.

Figura 3.1: Topología de red simple en NCTUns corriendo de Fedora 12



Finalmente se realizó una revisión de la factibilidad de realizar el proyecto en Omnet++. Dentro de la serie de proyectos construidos sobre este *framework* existe uno denominado MiXiM que se enfoca precisamente a redes móviles e inalámbricas. El software presenta desde ya la ventaja de ser multiplataforma, permitiendo realizar una instalación mucho más simple que la de NCTU<sub>ns</sub>, además de poder conseguir el instalador de manera sencilla en la página del proyecto. Respecto a las variables de interés para el trabajo, es posible su medición de manera directa simplemente utilizando el módulo correcto para la tarea.

En definitiva se decidió utilizar MiXiM/Omnet++ debido a las razones señaladas en el párrafo anterior, y por poseer un *mailing list* mucho más activo donde es posible obtener solución a muchos de los problemas que se pueden presentar, a la vez que el trabajo con el simulador permite entregar *feedback* a sus desarrolladores que generen mejoras futuras en la herramienta.

### 3.4. Visión general de MiXiM/Omnet++

Los siguientes párrafos contienen una revisión somera de las características principales de Omnet++ y MiXiM de modo de dar al lector una comprensión básica del trabajo a realizar. En caso de estar interesado en el uso de estas plataformas se recomienda encarecidamente expandir esta lectura con [3] y [2].

#### 3.4.1. Descripción funcional de Omnet++

Se vuelve a recalcar que Omnet++ por sí solo no es un simulador, sino un *framework* que facilita la construcción de simulaciones. Los modelos a simular son fundamentalmente una arquitectura de bloques llamados módulos simples, que pueden interactuar entre ellos mediante el envío de mensajes entre distintas compuertas ubicadas en los módulos. Estos módulos pueden combinarse en la creación de módulos compuestos. Todos los módulos simples son escritos en el lenguaje de programación C++ y hacen uso de las librerías de simulación de Omnet++.

Los módulos simples pueden tener parámetros, cuyos usos pueden variar desde personalizar el módulo a fijar la topología de la red. La topología de la red a simular, así como la estructura jerárquica de los módulos simples, se realiza mediante la escritura de otros archivos con extensión .NED (de su nombre en inglés *Network Description*). Estos archivos contienen instrucciones en un lenguaje denominado NED, al igual que la extensión.

#### 3.4.2. Simulación en Omnet++

La simulación en Omnet se construye a partir de tres tipos de elementos fundamentales: los archivos de topología, con extensión .NED; los archivos de descripción de los módulos

programados en C++ y con extensiones .h o .cc; y las definiciones de los mensajes, con extensión .msg

El primer paso que realiza el programa, es traducir los archivos .msg a clases de C++, luego toma estas clases junto a aquellas que definen los módulos simples y las compila junto al kernel de simulación. Finalmente, los archivos .NED son cargados de manera dinámica una vez que la simulación comienza. Un último componente que se integra a la simulación es un archivo de configuración (cuyo nombre por defecto es omnetpp.ini) que permite fijar los valores de algunos parámetros, así como determinar el número de veces que se ejecuta una misma simulación.

La salida de la simulación se escribe en distintos tipos de archivo de texto, dependiendo de la naturaleza de la variable estudiada: escalar o vectorial. Estos archivos de texto pueden ser procesados por las herramientas propias de Omnet++ o entregarse a un *software* externo, como MATLAB o GNU R, para realizar cualquier tipo de operación con el mismo.

### 3.4.3. Instalación de Omnet++

El proceso de instalación típica para distintos sistemas operativos se encuentra descrito de manera detallada en [34], por lo que en esta sección solo se informará de ciertos detalles puntuales de la instalación.

La instalación se realizó en el computador 1, descrito en la sección 3.1. Para esto se siguieron exactamente los pasos descritos en los capítulos 4 y 5 de [34], optando por realizar la instalación mediante un terminal de sistema en lugar de hacer uso de *synaptic*. No se hizo uso de ninguna opción o *flag* adicional durante la instalación.

Se instaló el paquete opcional Akaroa, de acuerdo a las instrucciones de [34]. Sin embargo es necesario destacar un punto que fue dejado de lado en el manual: es necesario llenar un pequeño formulario de inscripción para la descarga de esta librería. Este formulario genera una respuesta vía *e-mail* con las instrucciones para realizar la descarga, el cual puede tardar en llegar. En la realización de este trabajo fue necesario contactar a un encargado del proyecto para tener acceso a la descarga ya que el correo automático tardó más de lo estipulado en llegar.

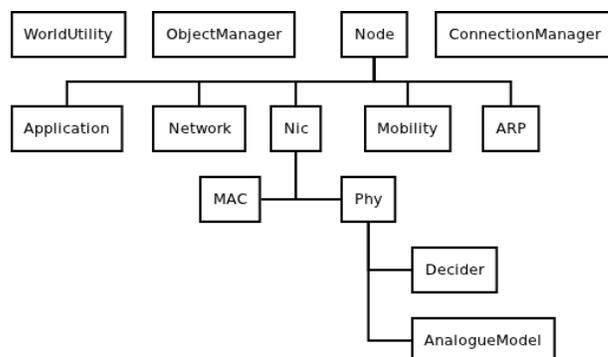
### 3.4.4. MiXiM

MiXiM (o *Mixed Simulator*) es un simulador construido sobre Omnet++ para la modelación de redes inalámbricas, tanto móviles como fijas. Posee modelos detallados de propagación, interferencia, protocolos MAC y consumo energético de los equipos de radio. Es la unión de una serie de proyectos originalmente iniciados por las universidades de Berlín y Delft.

### 3.4.5. Descripción de los módulos de MiXiM

Como se señaló en la sección 3.4.1, las simulaciones en Omnet++ se construyen en base a módulos, por lo que el valor que pueda aportar MiXiM yace en los módulos que este implemente. La figura 3.2 representa la estructura jerarquizada de los nodos que son comunes a todas las simulaciones. Las tablas 3.2,3.3 y 3.4 contienen una breve descripción de las funciones de cada uno de los módulos.

Figura 3.2: Esquema de la jerarquía de los módulos de MiXiM



ConnectionManager	Se encarga de determinar las conexiones entre distintos nodos, principalmente la calidad del enlace basado en interferencia y movilidad.
WorldUtility	Mantiene información acerca de las dimensiones físicas del espacio simulado.
ObjectManager	Maneja la interferencia, tanto en la señal como en la movilidad, que producen objetos físicos tales como muros o casas en la simulación
Node	Módulo compuesto que representa un terminal. En él destacan los módulos de aplicación, red, NIC y movilidad

Tabla 3.2: Módulos de una simulación en MiXiM

Application	Define el comportamiento de los nodos frente a distintos eventos
Network	Simula la operación de la capa de red del modelo OSI. Se centra principalmente en el ruteo dentro de la red.
Nic	Es un módulo compuesto que agrupa los módulos MAC y Phy
Mobility	Se encarga de almacenar información tal como la posición y la velocidad del nodo
ARP	En él se ejecuta el protocolo de resolución de direcciones

Tabla 3.3: Componentes del módulo *Node* en MiXiM

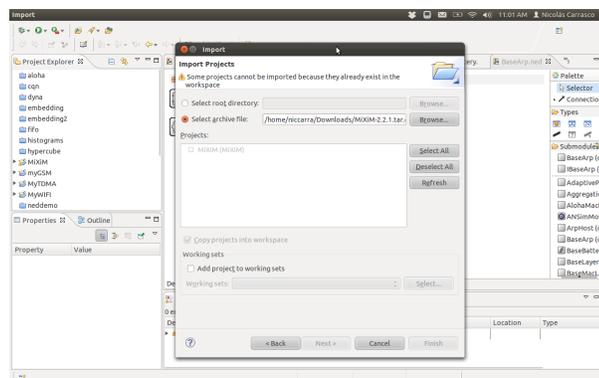
MAC	Realiza las tareas de crear señales, sensar el canal de transmisión y controlar los equipos de radio.
Phy	Es un módulo compuesto por el Decider, que se encarga de decidir si una señal fue recibida correctamente; y por el AnalogueModel, que simula los efectos del canal sobre la señal

Tabla 3.4: Componentes del módulo *Nic* en MiXiM

### 3.4.6. Instalación de MiXiM

El proceso de la instalación en detalle puede encontrarse en [2]. Para este trabajo se optó por realizar la instalación mediante Omnet++ indicada en el documento.

Figura 3.3: Proceso de instalación de MiXiM



## 3.5. Modelo de pruebas en MiXiM/Omnet++

### 3.5.1. Definición

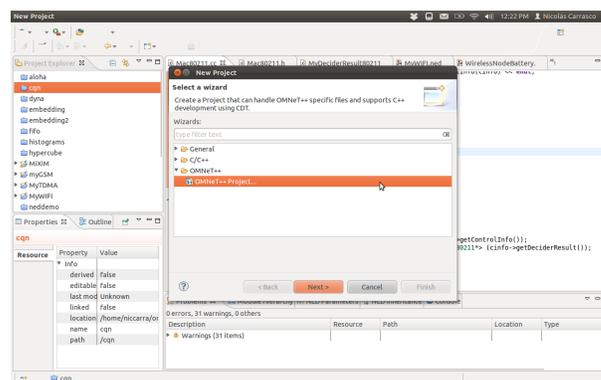
Una vez que ambos programas fueron instalados satisfactoriamente, se procedió a definir un escenario de prueba que permitiese obtener una visión más acertada tanto de *software* a utilizar como del problema de localización. Se busca que este escenario sea lo más simple posible, con el fin de enfocarse en posibles errores de implementación así como en la obtención de resultados para su posterior manipulación. A continuación se hace una enumeración de las características del modelo.

- Dimensiones físicas: todos los nodos serán ubicados sobre un plano, de este modo el problema se reduce a una localización bidimensional.
- Distribución de los nodos: se utilizarán 5 nodos; cuatro de ellos serán estáticos y el nodo a ubicar corresponderá a uno móvil. Los nodos estáticos serán equidistantes al punto de partida del nodo móvil, buscando así disminuir el error debido a la distribución física.
- Método de localización: se realizará una trilateración basada en los valores de RSS. Para esto será necesario almacenar los valores de posición y RSS de los nodos durante la simulación.
- Tecnología de radio: de acuerdo a lo planteado en los objetivos de este trabajo, se requiere que los equipos simulados cuenten con tecnología Wi-Fi o GSM.

### 3.5.2. Implementación

El primer paso de la implementación es la creación de un nuevo proyecto en Omnet++. Para esto se hace uso del *wizard* provisto por la aplicación. Accediendo al menú *File->New->Project* se despliega el panel que se muestra en la figura 3.4.

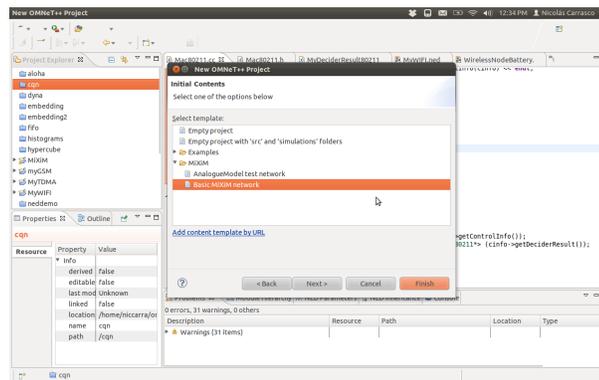
Figura 3.4: *Wizard* para la creación de un nuevo proyecto



Posteriormente, haciendo click sobre la opción *Omnet++ Project* se llega a una nueva

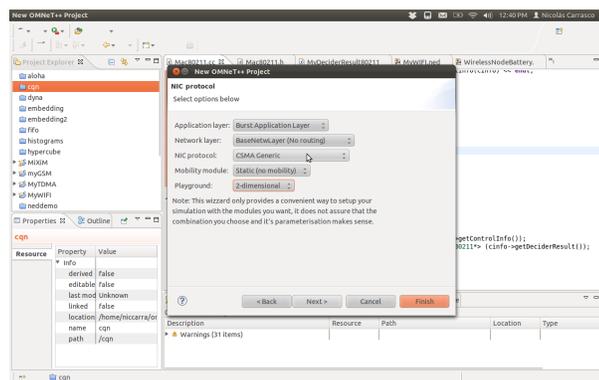
pantalla donde se introduce el nombre del proyecto. En este caso se optó por nombrarlo "MyGSM", ya que se trata de la simulación de una red que utiliza esta tecnología de radio. En este punto es posible terminar con el setup del proyecto, sin embargo al hacer click en *Next* se llega al menú que se observa en la figura 3.5, desde donde se elige la opción *Basic MiXiM network* como plantilla para la red.

Figura 3.5: Elección de plantilla



Esto conduce al último paso de la creación del proyecto, donde un menú con ventanas desplegadas, que puede verse en la figura 3.6, permite fijar una serie de opciones del proyecto. Se fijó la opción *Mobility Module* como *Constant Speed*, dejando todas las demás en su alternativa por defecto. Se debe señalar que estas opciones pueden modificarse en etapas posteriores del proyecto.

Figura 3.6: Configuración del proyecto



El siguiente paso de la implementación correspondió a traspasar las características planteadas en la sección 3.5.1 a variables y parámetros dentro de MiXiM. Esto se tradujo en dos grandes tareas: la creación del archivo de configuración, y la elección de los módulos a utilizar en la simulación.

## Elección de módulos

De los módulos mencionados en la sección 3.4, solo aquellos que componen a *Node* pueden ser elegidos de manera independiente. Sin embargo muchos módulos *Node* ya vienen contruidos dentro de MiXiM. Luego de una revisión somera de los nodos disponibles, se identificó que no existen nodos que implementen tecnología GSM u alguna otra tecnología de redes celulares; sin embargo existen nodos que implementan una sub-capa MAC denominada L-MAC [17], diseñada como una MAC para redes de sensores, que al igual que GSM se basan en la técnica de TDMA. Se decidió tomar este nodo como punto de partida, y renombrar el proyecto a "MyTDMA" para reflejar la nueva naturaleza del mismo.

La tabla 3.5 describe los distintos módulos que conforman el nodo que se utilizará en esta simulación.

Application	BurstApplLayer (modificado)	Corresponde al módulo implementado en MiXiM, el cual se ha modificado para enviar ráfagas periódicas de mensajes en lugar de solo una.
Network	BaseNetwLayer	Módulo por defecto ya implementado en MiXiM
Nic	NicLMac	Módulo por defecto ya implementado en MiXiM
Mac	LMacLayer (modificado)	Módulo por defecto, al cual se le agrega la funcionalidad de registrar valores de RSSI al decapsular mensajes.
Phy	MyPhyLayer	Módulo con instrucciones para la inicialización de MyDecider80211
Decider	MyDecider80211	Extensión del módulo Decider80211 de MiXiM al cual se le ha agregado la capacidad de almacenar información sobre RSSI
AnalogueModels	SimplePathlossModel	Módulo por defecto ya implementado en MiXiM
Mobility	ConstSpeedMobility	Módulo por defecto ya implementado en MiXiM
ARP	ArpHost	Módulo por defecto ya implementado en MiXiM
Battery	SimpleBattery	Módulo por defecto ya implementado en MiXiM

Tabla 3.5: Composición por módulos del nodo simulado

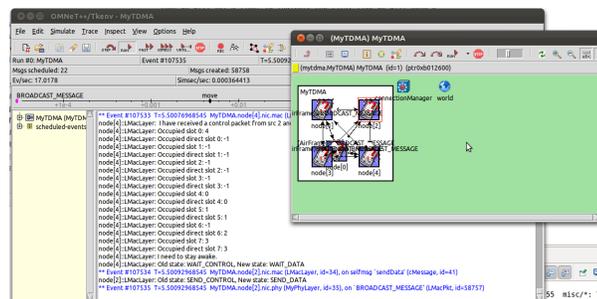
## Archivo de configuración

El archivo de configuración de MiXiM entega una serie de parámetros a los distintos módulos de la simulación. Estos parámetros se encuentran ordenados de acuerdo al módulo

al que pertenecen. La gran mayoría de los parámetros no fueron alterados, quedando sus valores por defecto; pero sí se manipularon todos aquellos que guardaban relación con la posición y velocidad de los nodos, así como algunas propiedades del medio de transmisión y el tráfico generado por los nodos.

El tiempo de simulación se fijó en 60 segundos, y las dimensiones del área a simular corresponden a un cuadrado de 180 metros de lado. Los nodos fijos quedaron ubicados formando un cuadrado, con centro en el centro del área de simulación y con una distancia entre sí de 90 metros. El nodo móvil por su parte se encuentra en el centro del área simulada y se le ha asignado una rapidez constante de 7 [m/s] con trayectoria aleatoria. La potencia de transmisión de los nodos queda fija en 110.11 [mW], mientras la sensibilidad de recepción toma el valor de -84 [dBm]

Figura 3.7: Simulación del modelo de pruebas



## 3.6. Análisis de sensibilidad de los distintos parámetros

Una vez que se hubo completado el modelo de prueba, se definió un experimento inicial para probar la influencia de los distintos parámetros físicos sobre la simulación. A continuación se entrega una descripción del método, las variables a manipular y la definición del caso base.

### 3.6.1. Método

Cada caso será simulado por un tiempo de 600[s], un total de 10 veces. Cada una de estas iteraciones utilizará una semilla de números aleatorios diferentes. Con el uso de MATLAB se procederá a calcular la posición y el error de localización de cada simulación mediante dos técnicas: *Linear least squares* y *Non-linear least squares*, ambas aplicadas a mediciones de RSS. El paso de intensidad de señal a distancia se hará mediante el uso de la ecuación de transmisión de Friis. De los resultados, se descartarán todos los casos en los que existen menos de 3 nodos, comprendiendo que estos pueden no tener solución única y que en general generarán una mala idea del desempeño del modelo; el porcentaje de casos favorables dentro del total será expresado en un porcentaje al que se le llamará cobertura. Sobre los resultados restantes se obtendrán los siguientes estadísticos: errores mínimo, máximo y promedio,

desviación estándar y percentil 95%, que corresponde al valor de *precision* & *accuracy*; además de entregar un histograma y un gráfico de distribución percentil.

### 3.6.2. Analogue Model

El *Analogue model* se trata de un filtro que aplica el efecto del canal sobre la señal simulada. Para esto se estima un mapa de atenuación evaluado en ciertos valores clave y se procede a interpolar entre estos valores para tener una función continua. La potencia recibida corresponde a una multiplicación de la potencia transmitida por el mapa de atenuación. Para determinar el modelo a utilizar, se modifica el archivo *config.xml* donde además se le asignan ciertos parámetros a cada modelo.

Existen al menos 3 modelos de de atenuación conocidos ya implementados en MiXiM:

- *SimplePathLossModel*: corresponde a una implementación de la ecuación de transmisión de Friis.
- *JakesFading*: se trata de una implementación del modelo de *fading* de Rayleigh, modificado por Jakes.
- *LogNormalShadowing*: implementación en la cual la atenuación sigue una distribución log-normal. Se aplica generalmente para áreas con muchas edificaciones.

### 3.6.3. Potencia de transmisión

El elemento *maxTxPow* permite fijar el valor de potencia, en [mW], a la cual transmiten los equipos. Se asigna en el archivo *omnetpp.ini*. Este parámetro puede modificarse en cualquiera de los modelos análogos ya presentados.

```
** .phy.maxTXPower = 110.11mW
```

### 3.6.4. Ruido térmico

El parámetro *thermalNoise* modela a la intensidad, en [dB], del ruido que llega al receptor junto a la señal. Se puede modificar de igual manera que el valor del punto anterior.

```
** .phy.thermalNoise = -110dBm
```

### 3.6.5. Coeficiente de atenuación

Se trata, tal como lo dice su nombre, del coeficiente de atenuación del medio. Se entrega como parámetro al modelo análogo en el archivo *config.xml* y la sintaxis de la instrucción es

la que se muestra a continuación:

```
<parameter name="alpha" type="double" value="3"/>
```

Este parámetro es adimensional y solo puede modificarse en el modelo *SimplePathLossModel*.

### 3.6.6. Posición de los nodos

Si bien no se trata de un parámetro físico de la señal, la posición relativa de los nodos afecta directamente la precisión del método de localización. Es deseable contar con distribuciones equiespaciadas de nodos, mientras que aquellas donde los nodos utilizados para la localización son más bien colineales deben evitarse. La posición de un nodo, en [m], se fija en el archivo *omnetpp.ini* mediante una serie de instrucciones.

```
**node[0].mobility.initialX = 90m  
**node[0].mobility.initialY = 90m
```

### 3.6.7. Definición del caso base

1. *Analogue Model*: Se utilizará el modelo *SimplePathLossModel*, con una frecuencia de transmisión de 2.4[GHz] y un coeficiente de atenuación igual a 3. La instrucción en el .xml para esto se muestra a continuación:

```
<AnalogueModels>  
  <AnalogueModel type="SimplePathlossModel">  
    <parameter name="alpha" type="double" value="3"/>  
    <parameter name="carrierFrequency" type="double" value="2.412e+9"/>  
  </AnalogueModel>  
</AnalogueModels>
```

2. *Potencia de transmisión y ruido térmico*: se fijaron en los valores que pueden verse en las secciones 3.6.3 y 3.6.4 respectivamente.
3. *Descripción del espacio*: la simulación se realizará en un espacio cuadrado de  $180 \times 180 [m^2]$  con 5 nodos, uno de ellos móvil. Los nodos fijos se ubican a medio camino entre el centro del area y las aristas de la misma, quedando el nodo móvil en el centro. Se fija la velocidad del nodo móvil en 7 [m/s] dentro del archivo *omnetpp.ini* mediante las líneas

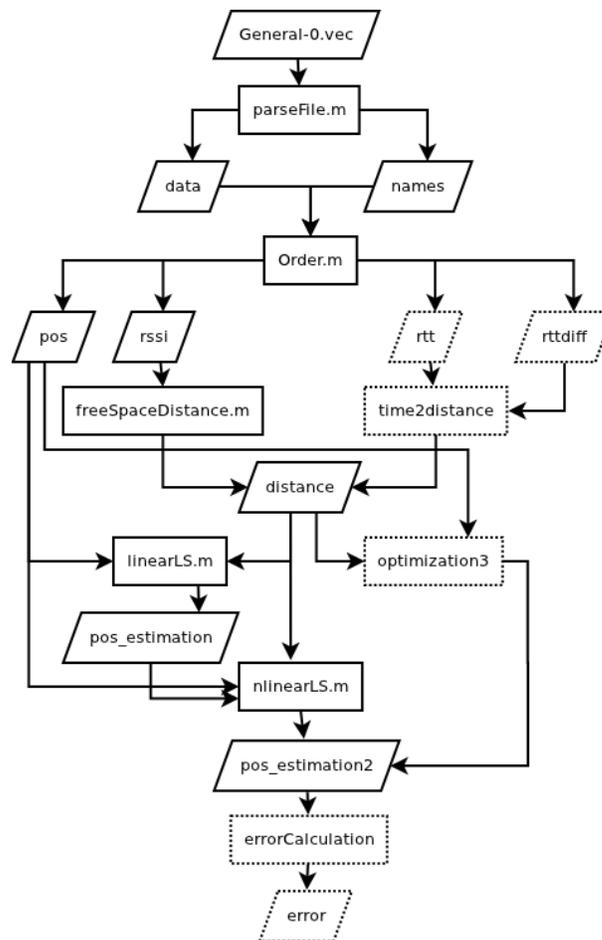
```
**node[*].mobilityType = "ConstSpeedMobility"  
**node[0].mobility.speed = 7mps
```

## 3.7. Implementación de una rutina de localización en MATLAB

### 3.7.1. Esquema de entrada/salida del código en MATLAB

El siguiente esquema (Fig. 3.8) corresponde a una visión global del código en MATLAB utilizado durante la fase de prueba del método de geo-localización mediante RSSI.

Figura 3.8: Diagrama de operación de código en MATLAB



### 3.7.2. Descripción detallada de los bloques operativos

#### Funciones principales

##### 1. parseFile.m

- *Entrada(s)*: fileName, header, vecNum
- *Salida(s)*: data, names

- *Funciones*: Ninguna

Esta función se encarga de traspasar los datos ubicados en el archivo *fileName* a dos matrices preeliminares: *data*, con datos ordenados de manera cronológica, y *names*, con la relación entre el número identificador del dato y el nombre del mismo. Para esto se hace uso de las funciones de MATLAB *textscan* y *fscanf* y del conocimiento a priori que se tiene de la estructura del archivo de salida del simulador. La variable *header* se utiliza para descartar las líneas irrelevantes del archivo mientras que *vecNum* fija el número de variables medidas por el simulador. En la configuración actual se utilizan los valores 17 y 13 respectivamente.

## 2. Order.m

- *Entrada(s)*: data, names
- *Salida(s)*: pos, rssi, rtt, rtt\_diff
- *Funciones*: index2name, groupData

Esta función separa los datos dentro de la matriz *data* en 4 diferentes matrices: *pos*, *rssi*, *rtt* y *rtt\_diff*. La construcción de estas matrices se realiza mediante la agrupación de todos los datos de distinto tipo, medidos en un mismo instante de tiempo, en una estructura auxiliar llamada *group* la cual es construida por la función auxiliar *groupData.m*. Conocida la posición de cada dato dentro de esta estructura es posible copiarlos en la posición correcta de cada una de las matrices de salida. Para determinar la unicidad de los datos del grupo, así como su posición, se utiliza la función *index2name.m* que simplemente recibe un índice y devuelve el tipo del dato, así como el nodo al que corresponde la medida.

## 3. freeSpaceDistance.m

- *Entrada(s)*: Pr
- *Salida(s)*: distance
- *Funciones*: Ninguna

Esta función implementa la ecuación de transmisión de Friis. Se hace uso de los valores de ganancia, potencia y frecuencia utilizada dentro del simulador. Además, se utiliza el valor  $299.792.458[m/s]$  como velocidad de la luz, y un coeficiente de atenuación  $\alpha = 3$ . El valor de la potencia recibida corresponde a los valores de RSS almacenados en la matriz *rssi* obtenida en el paso anterior.

## 4. time2distance

- *Entrada(s)*: time
- *Salida(s)*: distance

- *Funciones*: Ninguna

Se obtiene la distancia a partir de la velocidad de la señal simulada, y el tiempo que esta tarda en llegar de un nodo a otro. El tiempo de vuelo se obtiene desde las matrices *rtt* para el cálculo mediante ToA y *rtt\_diff* para el cálculo mediante TDoA.

#### 5. linearLS.m

- *Entrada(s)*: pos, distance
- *Salida(s)*: pos\_estimation
- *Funciones*: Ninguna

En este bloque se cuenta con  $n$  ecuaciones no-lineales de trilateración, una de las cuales se utiliza para linealizar las restantes, quedando así un sistema de  $n-1$  ecuaciones lineales que se resuelve mediante el método de mínimos cuadrados lineales.

#### 6. nlinearLS.m

- *Entrada(s)*: pos, distance
- *Salida(s)*: pos\_estimation
- *Funciones*: linearLS.m

Esta rutina se encarga de encontrar una solución al sistema de ecuaciones no-lineal de la trilateración. Se utiliza como punto de partida la solución obtenida mediante *linearLS.m*, y luego se aplica un algoritmo basado en la región de confianza (*trust-region-reflective*) para encontrar la solución no lineal.

#### 7. optimization3

- *Entrada(s)*: pos, distance
- *Salida(s)*: pos\_estimation
- *Funciones*: Ninguna

Corresponde a un tercer método de optimización el cual no se encontraba determinado durante la fase de prueba. Posteriormente se escogería usar el algoritmo de Levenberg-Marquardt.

#### 8. errorCalculation

- *Entrada(s)*: pos, pos\_estimation
- *Salida(s)*: error
- *Funciones*: Ninguna

Realiza el cálculo de al menos un indicador de desempeño. Estos indicadores permanecieron indefinidos durante la fase de prueba.

## Funciones auxiliares

### 1. index2name.m

- *Entrada(s)*: names, ind
- *Salida(s)*: name
- *Funciones*: Ninguna

La función realiza el trabajo de buscar el índice *ind* dentro de la primera columna de la celda *names*. A continuación entrega un nombre, que corresponde a la concatenación de la segunda (nombre del nodo) y tercera (nombre del vector) columna de *names*.

### 2. groupData.m

- *Entrada(s)*: data, names, time
- *Salida(s)*: jump, group
- *Funciones*: Ninguna

En esta función se crea un grupo, estructura que almacena todos los datos ubicados en *data*, medidos en el instante de tiempo *time*. Dado que tanto *groupData.m* como *Order.m* recorren *data*, la salida *jump*, permite a *Order.m* saltar un número de filas que ya fueron reconocidas como integrantes del grupo, evitando así aumentar el tiempo de ejecución del algoritmo. La variable *group* tiene la estructura señalada en el cuadro 3.6:

time	
rss	
source	
destination	
node0 X	node0 Y
node1 X	node1 Y
node2 X	node2 Y
node3 X	node3 Y
node4 X	node4 Y

Tabla 3.6: Estructura de *group*

## Entradas/Salidas

### 1. General-0.vec

Corresponde a un archivo de texto plano, resultado de la simulación en Omnet++. Cuenta con un header que no resulta relevante, el cual es descartado, y dos tipos de líneas distintas que se utilizan para la construcción de *names* y *data*. A continuación del header, se encuentran líneas con el formato:

```
vector vectorid projectname.node[nodenummer].module Variable ETV
```

En estas líneas se encuentra la información que correlaciona el identificador del vector con su nombre, y se guardan en la variable *names*. El segundo tipo de líneas, ubicado al final del archivo posee la estructura mostrada a continuación:

```
vectorid simulationtime time value
```

Desde estas líneas, se obtienen los valores de los distintos vectores en distintos instantes de tiempo, esta información se almacena en *data*.

### 2. data

Se trata de una matriz del tipo double y de dimensión  $nx3$ . Una fila cualquiera de la matriz tiene la estructura indicada en el cuadro 3.7.

vectorid	time	value
----------	------	-------

Tabla 3.7: Estructura de *data*

### 3. names

Se trata de una matriz, del tipo cell y dimensiones  $3x1$ . Su primera fila corresponde a una matriz del tipo double y dimensiones  $13x1$  donde se almacenan los identificadores de los vectores. Sus dos filas restantes son ambas cells de dimensiones  $13x1$ , en la primera de ellas se almacena el número del nodo mientras en la otra se almacena el nombre de la variable.

### 4. pos

Corresponde a una matriz de  $nx11$ , con datos del tipo double y filas que almacenan los datos de acuerdo a la distribución del cuadro 3.8:

time	node0 X	node0 Y	node1 X	node1 Y	...	...	noden X	noden Y
------	---------	---------	---------	---------	-----	-----	---------	---------

Tabla 3.8: Estructura de *pos*

### 5. rssi

En esta matriz de  $nx5$  se almacenan variables del tipo double que cuantifican las

mediciones de RSS desde el nodo móvil a los nodos restantes. La fila tipo de esta estructura corresponde a la mostrada en el cuadro 3.9

time	node1 RSSI	node2 RSSI	...	noden RSSI
------	------------	------------	-----	------------

Tabla 3.9: Estructura de *rss\_i*

6. rtt

Se trata de la matriz que almacena las mediciones de tiempo entre la salida de la señal desde el nodo móvil hasta la llegada de la respuesta de un nodo fijo.

7. rtt\_diff

Como en el caso anterior es una medida de tiempo, pero en esta ocasión se trata de la diferencia de tiempo entre la llegada de la señal desde el nodo móvil al nodo fijo 1, y la llegada de la misma señal a cualquier otro nodo.

8. distance

Es una matriz de iguales dimensiones a *rss\_i*, sus valores corresponden a distancias entre los nodos en lugar de valores de potencia.

9. pos\_estimation

Es una matriz de  $nx2$ , con datos del tipo double cuyas filas almacenan las posiciones X e Y del nodo a ubicar.

10. error

Es una matriz que almacena el error de localización, medido como la norma euclidiana entre la ubicación real y la ubicación estimada. Se almacenan 9 valores por fila correspondientes a cada método de localización y a cada técnica de resolución de la trilateración

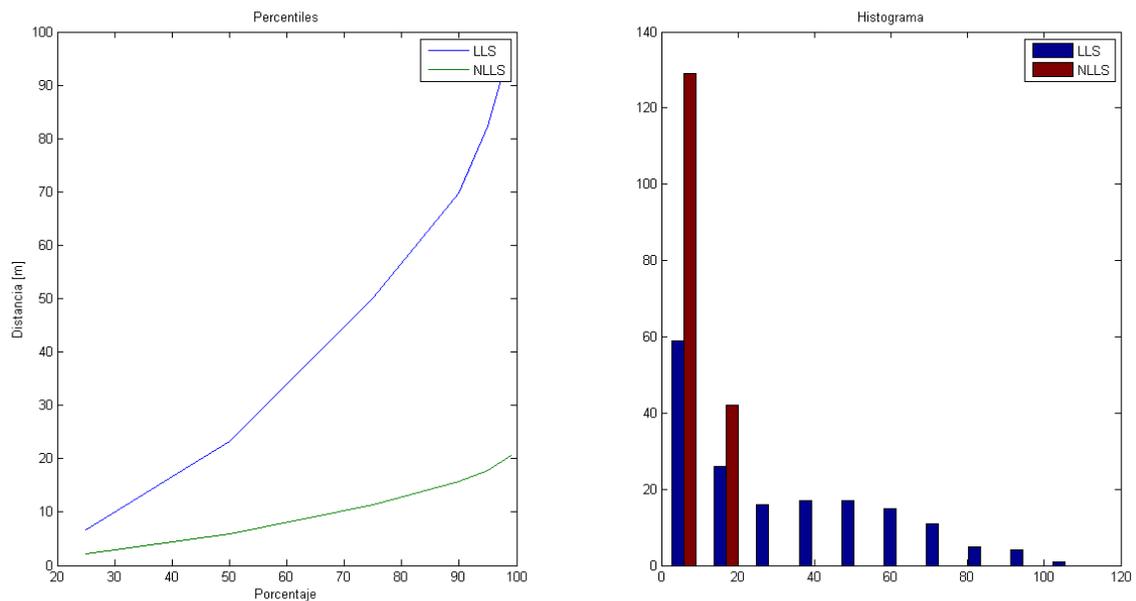
# Capítulo 4

## Resultados

### 4.1. Resultados del análisis de sensibilidad

Los resultados del caso base se incluyen en cada tabla para realizar la comparación, mientras que los gráficos correspondientes se muestran en la figura 4.1. Tanto en los gráficos como en las tablas se utilizan los acrónimos RSSI, LLS y NLLS para referirse a Intensidad de señal de radio, mínimos cuadrados lineales y mínimos cuadrados no-lineales respectivamente.

Figura 4.1: Caso base



### 4.1.1. Analogue Model

Se cambió el modelo análogo por los ya descritos en la sección 3.6.2. Se reemplaza la descripción del modelo en el archivo XML por las siguientes descripciones.

- `<AnalogueModel type="JakesFading">`  
`<parameter name="carrierFrequency" type="double" value="2.412e+9"/>`  
`<parameter name="fadingPaths" type="long" value="3"/>`  
`<parameter name="delayRMS" type="double" value="0.0001"/>`  
`<parameter name="interval" type="double" value="0.001"/>`  
`</AnalogueModel>`

El parámetro *carrierFrequency* se entrega en [Hz], mientras que los parámetros *delayRMS* *interval* tienen sus valores en [s] y *fadingPaths* se trata de una variable adimensional. La descripción detallada de estos parámetros se pueden encontrar en el archivo *JakesFading.h*.

- `<AnalogueModel type="LogNormalShadowing">`  
`<parameter name="mean" type="double" value="0.5"/>`  
`<parameter name="stdDev" type="double" value="0.25"/>`  
`<parameter name="interval" type="double" value="0.001"/>`  
`</AnalogueModel>`

Aquí los parámetros *mean* y *stdDev* se entregan en [dB] mientras que *interval* es una variable medida en [s].

	SimplePathLoss		JakesFading		LogNormal Shadowing	
	LLS	NLLS	LLS	NLLS	LLS	NLLS
Cobertura [%]	56,07		58,67		58,14	
Error máximo [m]	111,03	21,05	117,75	96,53	96,52	80,71
Error mínimo [m]	0,98	0,44	2,67	2,67	5,14	5,14
Error medio [m]	30,68	7,36	55,42	51,34	49,49	48,32
Desviación estándar [m]	26,7	5,41	24,1	19,64	19,49	17,95
percentil 95 % [m]	82,48	17,74	95,39	79,08	78,87	74,76

Tabla 4.1: Resultados variación del modelo de atenuación

Figura 4.2: Jakes' fading

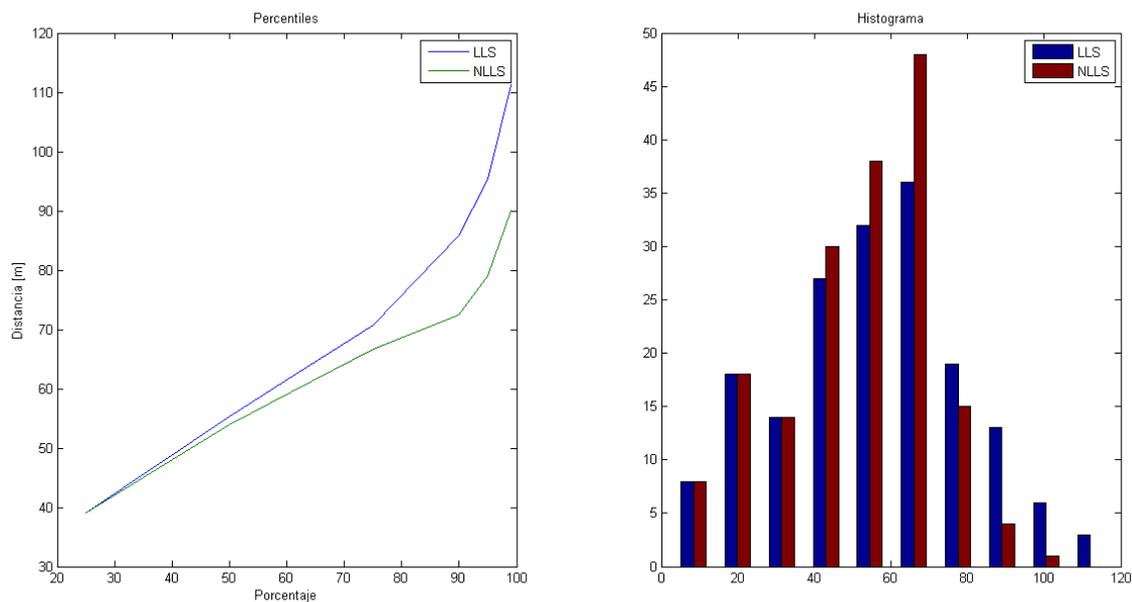
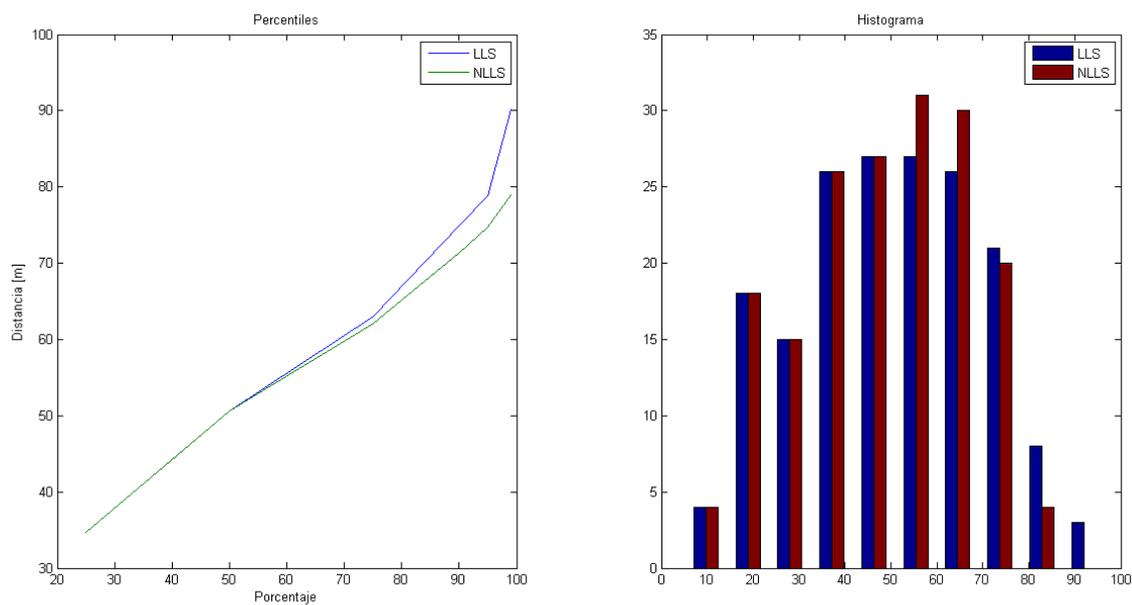


Figura 4.3: Log-normal shadowing



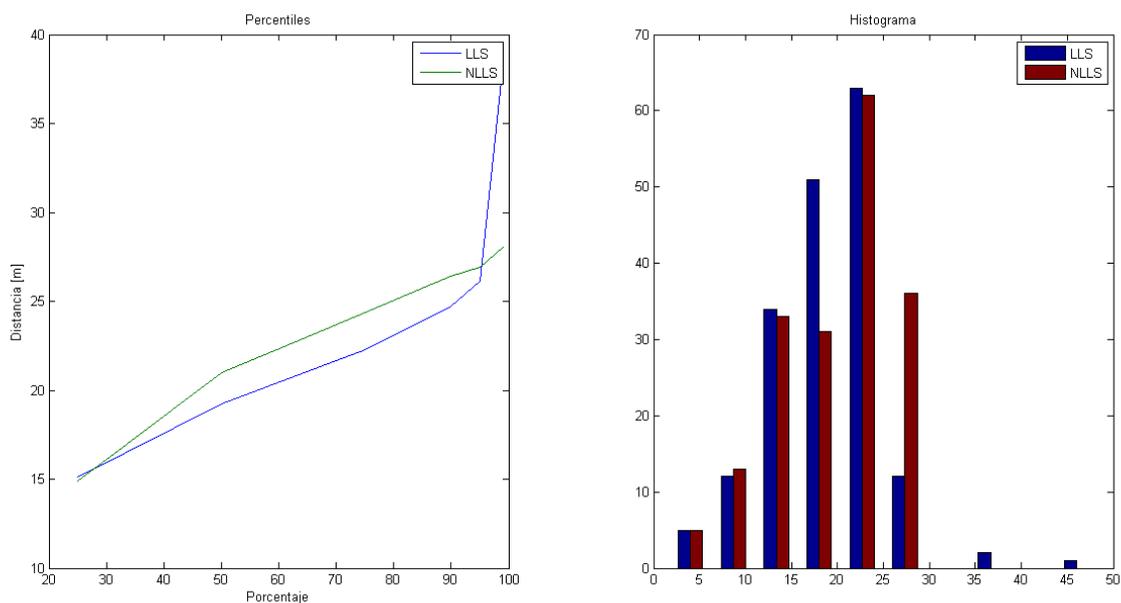
### 4.1.2. Coeficiente de atenuación

Se realizaron simulaciones para distintos valores de alfa: 2, 2.7, 3.3 y 4. Sin embargo para este último caso se obtuvo una cobertura de 0 %, es decir jamás se cumplieron las condiciones mínimas para encontrar al móvil.

	alfa = 3		alfa = 2		alfa = 2,7		alfa = 3,3	
	LLS	NLLS	LLS	NLLS	LLS	NLLS	LLS	NLLS
Cobertura [%]	56,07		60,81		60,81		22,71	
Error máximo [m]	111,03	21,05	106,19	101,54	48,32	28,41	41,32	32,73
Error mínimo [m]	0,98	0,44	4,16	3,95	1,98	1,69	1,42	1,51
Error medio [m]	30,68	7,36	51,7	48,8	18,76	19,5	21,01	18,85
Desviación estándar [m]	26,7	5,41	23,2	22,1	6,02	6,12	8,81	6,57
percentil 95 % [m]	82,48	17,74	88,63	83,97	26,11	26,94	37,56	28,45

Tabla 4.2: Resultados variación coeficiente de atenuación

Figura 4.4: alfa = 2.7



### 4.1.3. Potencia de transmisión

Se realizó una variación en la potencia de transmisión dentro de un rango de 10 %

Potencia de transmisión	100 %		90 %		110 %	
	LLS	NLLS	LLS	NLLS	LLS	NLLS
Cobertura [%]	56,07		53,55		56,07	
Error máximo [m]	111,03	21,05	88,57	13,43	137,48	28,21
Error mínimo [m]	0,98	0,44	0,19	0,06	0,43	0,68
Error medio [m]	30,68	7,36	20,68	4,11	41,19	11,77
Desviación estándar [m]	26,7	5,41	20,56	3,05	32,92	7,15
percentil 95 % [m]	82,48	17,74	65,45	9,64	102,71	24,41

Tabla 4.3: Resultados variación de potencia de transmisión

### 4.1.4. Ruido térmico

Se realizó una variación en el ruido térmico dentro de un rango de 10 %

Ruido térmico	100 %		90 %		110 %	
	LLS	NLLS	LLS	NLLS	LLS	NLLS
Cobertura [%]	56,07		56,07		56,07	
Error máximo [m]	111,03	21,05	103,71	19,48	111,63	21,18
Error mínimo [m]	0,98	0,44	1,04	0,52	0,97	0,44
Error medio [m]	30,68	7,36	28,75	6,75	30,84	7,41
Desviación estándar [m]	26,7	5,41	25,06	4,93	26,83	5,45
percentil 95 % [m]	82,48	17,74	77,49	16,39	82,89	17,84

Tabla 4.4: Resultados variación ruido térmico

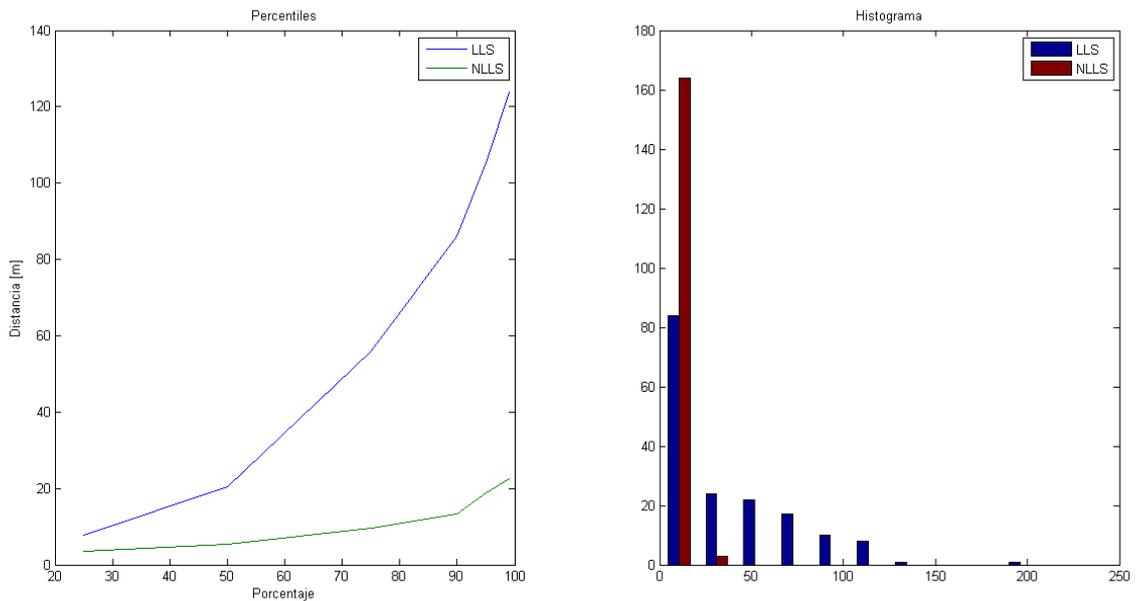
### 4.1.5. Distribución de los nodos

Se llevó el nodo de la posición (45,45) a la posición (68,68) con el objetivo de tener una línea recta más pronunciada entre 3 de los nodos.

	Cuadrado perfecto		Asimétrico	
	LLS	NLLS	LLS	NLLS
Cobertura [%]	56,07		54,05	
Error máximo [m]	111,03	21,05	206,26	26,22
Error mínimo [m]	0,98	0,44	0,94	0,24
Error medio [m]	30,68	7,36	35,41	6,96
Desviación estándar [m]	26,7	5,41	34,48	5,00
percentil 95 % [m]	82,48	17,74	105,35	18,76

Tabla 4.5: Resultados desplazamiento de un nodo

Figura 4.5: Cambio de posición del nodo



## 4.2. Discusión de resultados

El sencillo ejercicio que se ha mostrado en estas secciones permite realizar algunas observaciones respecto al impacto de las variables en el desempeño del modelo y a los estadísticos que mejor permiten cuantificar dicho desempeño. Una primera e importante observación es que el modelo de localización basado en NLLS es siempre superior a su contraparte lineal, por lo que todo el análisis posterior se centrará principalmente en este método.

Ya tenemos como una medida de calidad del modelo el concepto de *precision & accuracy*, el cual se refleja en las tablas dentro de la fila titulada percentil 95%. Respecto a los otros estadísticos, podemos ver que aquellos que señalan fenómenos puntuales como el error mínimo y máximo no guardan correlación con las variaciones del percentil 95, a la vez que no permiten emitir un juicio respecto a la utilidad del modelo, salvo en el caso puntual donde se requiere un error máximo acotado. La desviación estándar es otro de los parámetros que no guarda relación con las variaciones ni de los parámetros físicos ni de los otros estadísticos. El error medio muestra tener una correlación positiva con el percentil 95 para todos los casos, dejándolo como candidato para ser otro indicador de desempeño. Finalmente la cobertura, que no varía mayormente en los distintos casos, juega un rol protagónico al momento de variar el coeficiente de atenuación y la interpretación de los demás indicadores no podría entenderse sin su presencia.

En resumen, si bien el percentil 95 da una buena idea general del desempeño de un modelo, es necesario considerar los demás estadísticos para poder generar *tie-breakers*. La elección natural para un *tie-breaker* de carácter general es el error medio, mientras que el error máximo puede jugar igual papel en los casos donde sea crítico contar con una cota razonable del mismo. La cobertura por su parte permite realizar un análisis de la cantidad contra la calidad de la localización, pudiendo así diferenciar modelos capaces de gran precisión para un bajo número de nodos, o bien modelos con tasas de error mayores pero capaces de localizar una gran cantidad de terminales.

Respecto a los parámetros físicos, es posible apreciar que el ruido térmico genera una muy leve variación en los estadísticos mencionadas en los párrafos anteriores, teniéndose que ninguno aumenta o disminuye más allá de un 10%. De los parámetros restantes, el coeficiente de atenuación es la variable que tiene mayor impacto en la cobertura; mientras que el mismo coeficiente junto a la potencia de transmisión afectan en medidas similares al resto de los estadísticos, con la diferencia de que esta última tiene la capacidad de mejorar el desempeño del modelo mientras que la primera sólo lo empeora. El uso de otros modelos análogos en ningún caso contribuye a mejorar el desempeño de la simulación, observación que es congruente con el hecho de que el método para obtener las distancias implementado en MATLAB se basa en la ecuación de Friis al igual que *SimplePathLossModel*. Finalmente, el efecto de mover los nodos tuvo el efecto esperado de empeorar los resultados de la simulación, sin embargo es posible que el movimiento no haya sido suficientemente drástico como para causar efectos mayores.

Resumiendo, las variables más interesantes para influir en el desempeño del modelo son el coeficiente de atenuación y la potencia de transmisión, teniendo el primero un efecto mucho más importante sobre la cobertura. En consecuencia, y dado que el método de cálculo de las distancias se basa en la ecuación de Friis se considera que lo mejor es utilizar el *SimplePathLossModel* para los experimentos definitivos.

### 4.2.1. Correcciones al modelo de pruebas e implementación del modelo definitivo

Luego de ejecutar la simulación de prueba se pudo apreciar una serie de problemas con el modelo, ante los cuales fue necesario modificar los módulos de simulación para obtener mejores resultados.

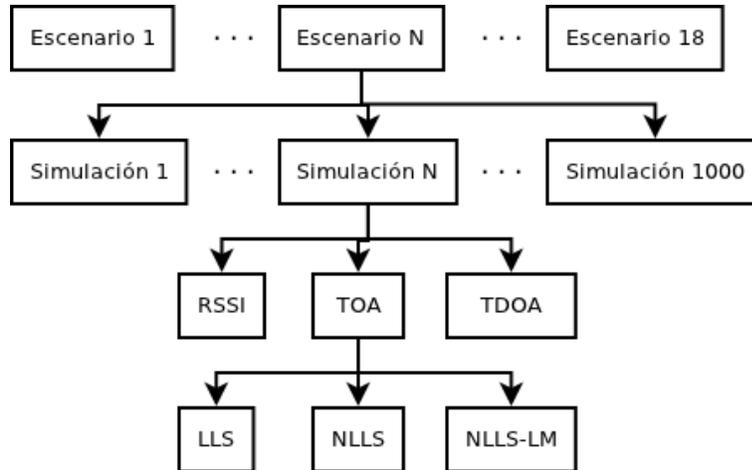
1. Registro de variables: la interacción actual entre *LMacLayer* y *BurstApplLayer* impide que las posiciones de los nodos anclas se registren en todas las mediciones. Si bien al tratarse de nodos fijos no es necesario registrar su posición constantemente o esta puede agregarse a posteriori durante el procesamiento de los datos en MATLAB, se considera valioso tener un escenario de simulación lo más generalizado posible, por lo que se modifican estos módulos para evitar este problema.
2. Pérdida del enlace entre 2 nodos: al moverse el nodo a localizar, existe la posibilidad de que este quede fuera de la cobertura de los 3 nodos necesarios para lograr realizar la trilateración. Es por esto que es necesario ajustar la distribución física de los nodos de modo de minimizar estos casos, o por otra parte utilizar la distancia máxima admisible como un parámetro que indique la validez del modelo.
3. Filosofía de programación: durante el desarrollo del modelo de prueba se modificó gran parte del código fuente de MiXiM, debido a la falta de conocimiento que se tenía sobre la herramienta. Esta práctica atenta directamente contra el sentido del simulador, por lo que se desea unificar todos los cambios en nuevos módulos que hereden de los módulos originales del simulador, evitando así pérdida de la funcionalidad del *software*.

### 4.2.2. Resultados del modelo definitivo

#### Estructura del experimento definitivo

El experimento final consiste en la elaboración de 18 escenarios que solo se diferencian por el valor del coeficiente de atenuación que utilizan. Para cada escenario se ejecutaron un total de 1000 simulaciones diferentes, las cuales utilizan cada una un generador de números aleatorio (RNG por sus siglas en inglés) diferente. El proceso de localización a partir de los datos arrojados por estas simulaciones se realizó con 3 métodos diferentes: *Radio Signal Strength*, *Time of Arrival* y *Time Difference of Arrival*; con las distancias calculadas por estos métodos se realizaron trilateraciones, las cuales se resolvieron mediante 3 métodos matemáticos distintos: Mínimos cuadrados lineales (LLS), Mínimos cuadrados no-lineales basados en región de confianza (NLLS) y Mínimos cuadrados no-lineales basados en el algoritmo de Levenberg-Marquardt (NLLS-LM). Esta información se expone de manera gráfica en la figura 4.6 y la tabla 4.6:

Figura 4.6: Desglose de los datos del experimento



Escenario	Coefficiente de atenuación (alfa)
1	2.0
2	2.1
3	2.2
4	2.3
5	2.4
6	2.5
7	2.6
8	2.7
9	2.8
10	3.0
11	3.1
12	3.2
13	3.3
14	3.4
15	3.5
16	3.6
17	3.7
18	3.8

Tabla 4.6: Definición de los escenarios

De la tabla 4.6, es posible notar que entre los escenarios 9 y 10 hay una variación del coeficiente de atenuación mayor a la que existe entre cualquier otro par de escenarios contiguos. La causa de esto es que los datos adquiridos durante las simulaciones del escenario 10, originalmente simulado con un valor de alfa igual a 2.9 fueron corrompidos durante su procesamiento. El motivo de este fenómeno se desconoce, pero gran parte de los archivos de texto no podían ser ejecutados por un procesador de texto.

Los métodos de LLS y NLLS requieren de a lo menos 3 nodos ancla para construir un sistema de ecuaciones determinado al que se le pueda aplicar el método de mínimos cuadrados.

Por su parte, el algoritmo de Levenberg-Marquardt es aplicable a sistemas indeterminados, es decir con un número de incógnitas menor al número de ecuaciones.

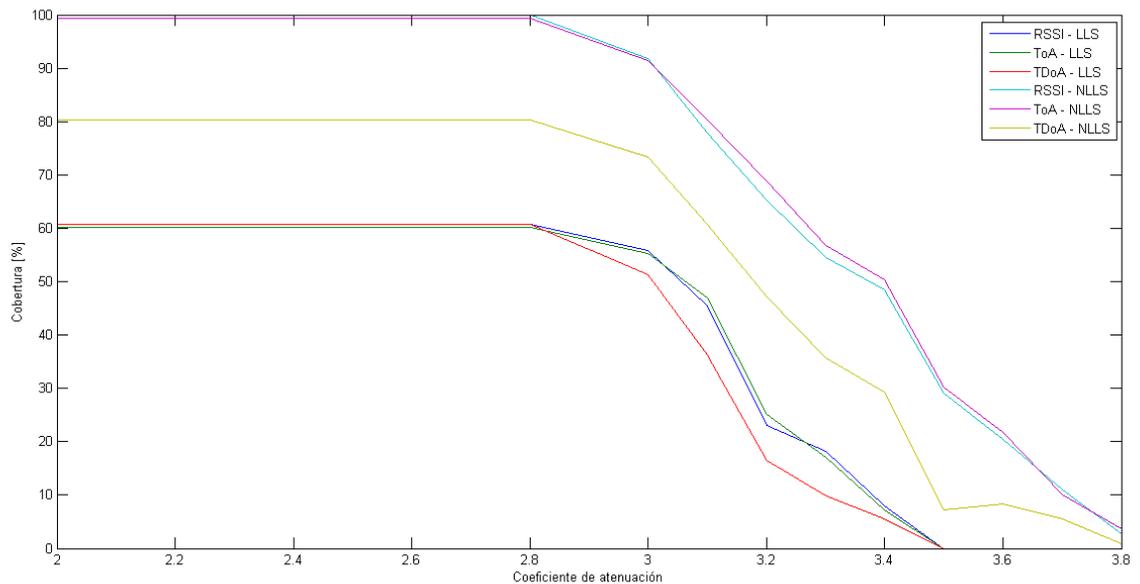
Dado que la implementación de NLLS y NLLS-LM en MATLAB requieren de un punto de partida sobre el cual iterar el método, lo que se realizó primero fue el cálculo de una solución inicial al problema de localización mediante el método de LLS, la cual luego se utilizó como punto de partida de los otros dos métodos. Esto asegura que los métodos no-lineales cuenten con un punto de partida que ya se encuentra cerca de la solución, evitando así que el algoritmo converja hacia mínimos locales.

## Cobertura de los distintos modelos de localización

Como ya se mencionó anteriormente, los distintos métodos requieren de cierto número de nodos ancla para poder realizar la localización de manera adecuada. Debido a esto se definió una *cobertura* de cada modelo, que corresponde al total de casos en los que es posible realizar la localización respecto al total de casos simulados. Para los métodos de LLS y NLLS se entenderá como caso favorable, aquél donde se reciben señales de al menos 3 nodos ancla diferentes. Mientras para el caso del método NLLS-LM se definió como caso favorable todo aquél en el cual se reciben señales de a lo menos 2 nodos distintos.

La figura 4.7 muestra el valor que posee la cobertura de los distintos métodos en cada uno de los escenarios. Cabe destacar que las curvas identificadas como “LLS” corresponden a la cobertura de los métodos que utilizan a lo menos 3 nodos, mientras que las curvas identificadas como “NLLS” corresponden a las del método que utiliza 2 nodos.

Figura 4.7: Comportamiento de la cobertura en función de las variaciones del coeficiente de atenuación



Como es esperable, todas las curvas presentan una pendiente negativa a medida que aumenta el coeficiente de atenuación. Esto es congruente con la ecuación de Friis, donde es posible apreciar que un aumento en el coeficiente de atenuación disminuye la potencia recibida, disminuyendo así el área de cobertura de cada nodo ancla y haciendo más difícil que el nodo móvil pueda encontrarse en la intersección de las áreas de cobertura de múltiples nodos a la vez.

Otro resultado esperado es que las curvas de NLLS se encuentren todas por encima de las curvas de LLS, ya que los casos en que el nodo móvil es cubierto por a lo menos 3 nodos ancla es un subconjunto de los casos en que el terminal a localizar se encuentra cubierto por al menos 2 nodos. Bajo esta misma lógica, el método de TDoA se encuentra siempre por debajo de las curvas de los otros métodos, debido a que para generar  $n$  distancias se requiere de  $n$  diferencias de tiempo, o lo que es equivalente  $(n+1)$  nodos que reciban las señales.

Un resultado novedoso es la saturación de la cobertura para valores del coeficiente de atenuación inferiores a 2.8. No se cuenta con una explicación exacta de este fenómeno, pero es posible que la respuesta este nuevamente en la ecuación de Friis. Es posible que la ecuación este optimizada para dar buenos resultados solo en torno al valor 3, o por otra parte puede que para valores tan pequeños de alfa, la potencia de transmisión tome un rol mucho más importante en la determinación de la potencia de recepción, siendo este el factor que limita la cobertura máxima en los modelos simulados.

## Desempeño de los distintos métodos

Como ya se mencionó en la sección 4.2, se utilizó el concepto de precisión & exactitud para comparar el desempeño de modelos distintos. Se realizó una medición de la exactitud, para una precisión del 95 %, de cada método de localización bajo las distintas técnicas matemáticas usadas para resolver el problema de trilateración. Los resultados de estas mediciones se expresan de manera gráfica en las figuras 4.8, 4.9, 4.10.

Figura 4.8: Exactitud de cada escenario para la localización mediante RSSI

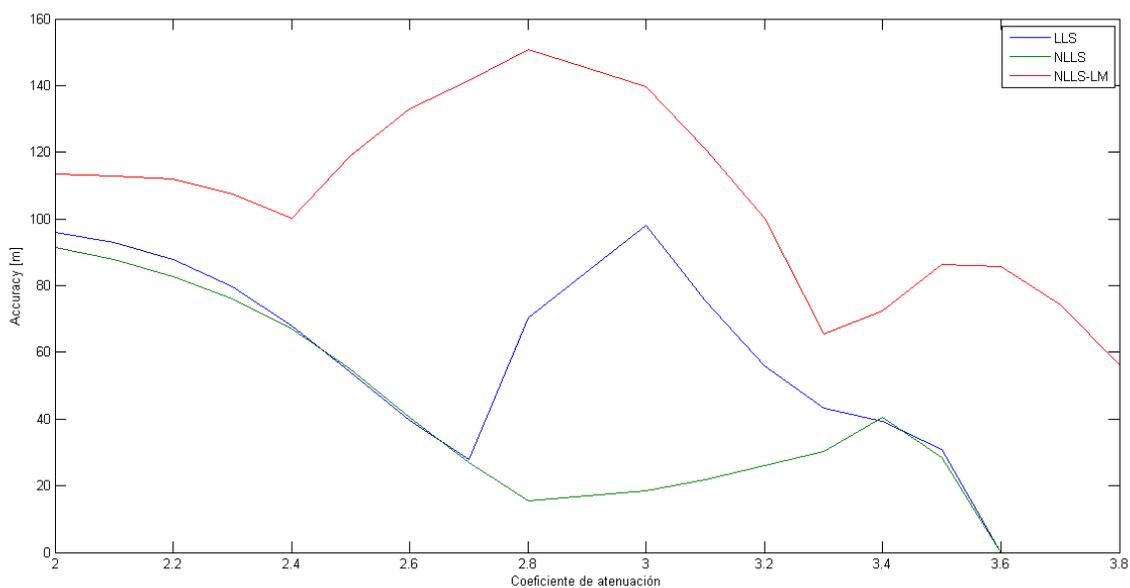


Figura 4.9: Exactitud de cada escenario para la localización mediante ToA

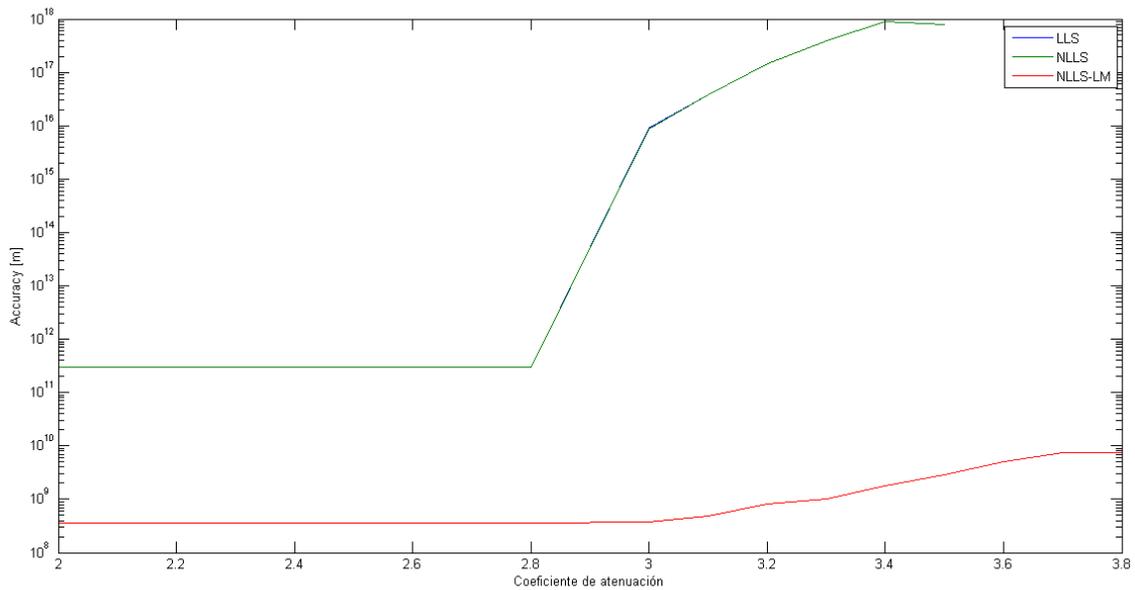
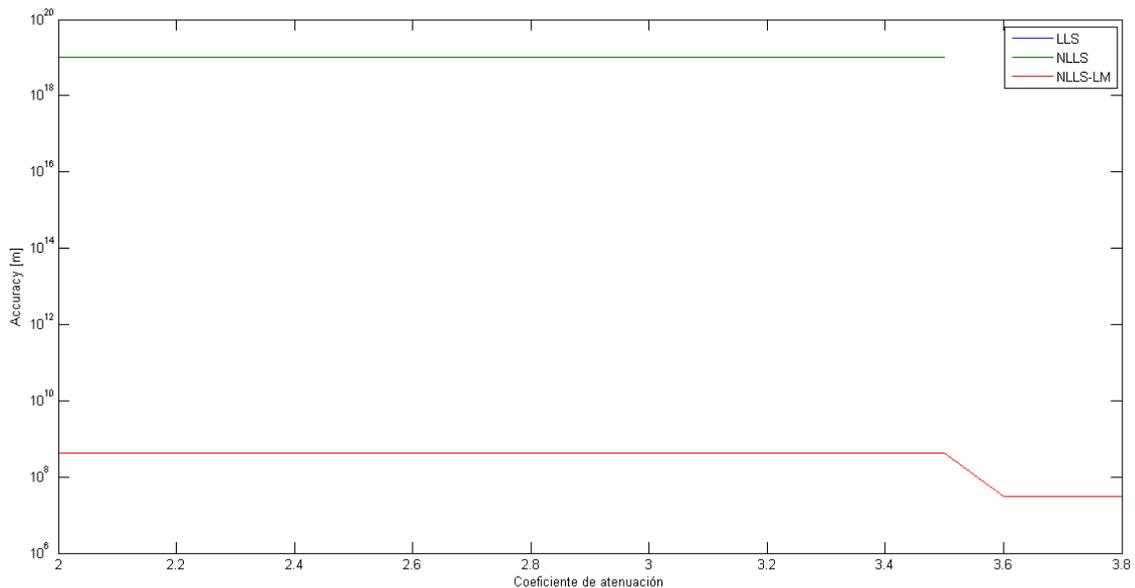


Figura 4.10: Exactitud de cada escenario para la localización mediante TDoA



La primera gran observación que se desprende de las figuras 4.9 y 4.10, es que los métodos basados en tiempo de viaje de la señal no pueden ser simulados satisfactoriamente. El principal problema para simular correctamente estos métodos yace en la ecuación planteada en la sección 2.4.1, de la cual se desprende que es necesario conocer el retardo introducido por el procesamiento de la señal para lograr estimar el tiempo de vuelo real de la misma. Considerando por un momento que las señales electromagnéticas en el aire se mueven a la

velocidad de la luz, se tiene que para errores en el orden de las decenas de metros se requiere estimar este retardo con una precisión de 0.1 microsegundos. Las simulaciones en MiXiM no cuentan con este nivel de precisión, por lo que resulta imposible que el modelo funcione en base a los datos proporcionados por el simulador. En consecuencia, que el método de NLLS-LM muestre un desempeño muy superior a las demás opciones no pasa de una observación anecdótica.

Respecto a lo que se puede ver en la figura 4.8, primero se debe destacar que el método de NLLS ofrece siempre el mejor desempeño de entre todos los métodos, cuando la cobertura no es lo más relevante. Por su parte, si bien el método de NLLS-LM tiene un desempeño peor a los otros dos métodos cabe recordar que ofrece además una cobertura mucho mayor. Para un valor del coeficiente de atenuación cercano a 3.3 se puede observar que la brecha de desempeño entre todos los modelos disminuye bastante, lo que eventualmente podría darle una ligera ventaja a NLLS-LM en aplicaciones donde la cobertura resulta más importante que el error de localización.

### Curvas de variables como función del coeficiente de atenuación

En esta sección se presentan los gráficos de dispersión para las variables medidas en cada escenario. Dados los resultados mostrados en la sección 4.2.2, se incluyen en cada gráfico solo las variables del modelo RSSI resuelto mediante la técnica de NLLS; esto por tratarse del método que exhibe el mejor desempeño. En cada uno de los gráficos se muestra la medida obtenida de cierta variable para cada escenario, en cada una de las 1000 simulaciones que conforman dicho escenario. Además de estas mediciones en el gráfico se incluye una línea de tendencia, obtenida mediante el uso de mínimos cuadrados, de modo de orientar y facilitar la visualización de la información.

Figura 4.11: Error máximo del método RSSI, resuelto mediante NLLS para cada simulación

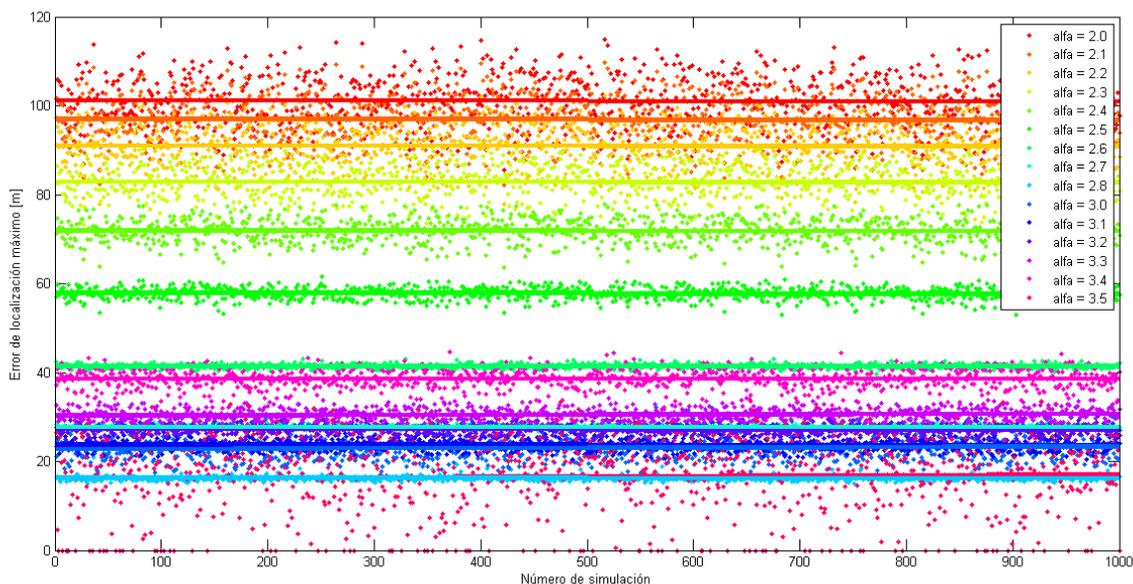


Figura 4.12: Error mínimo del método RSSI, resuelto mediante NLLS para cada simulación

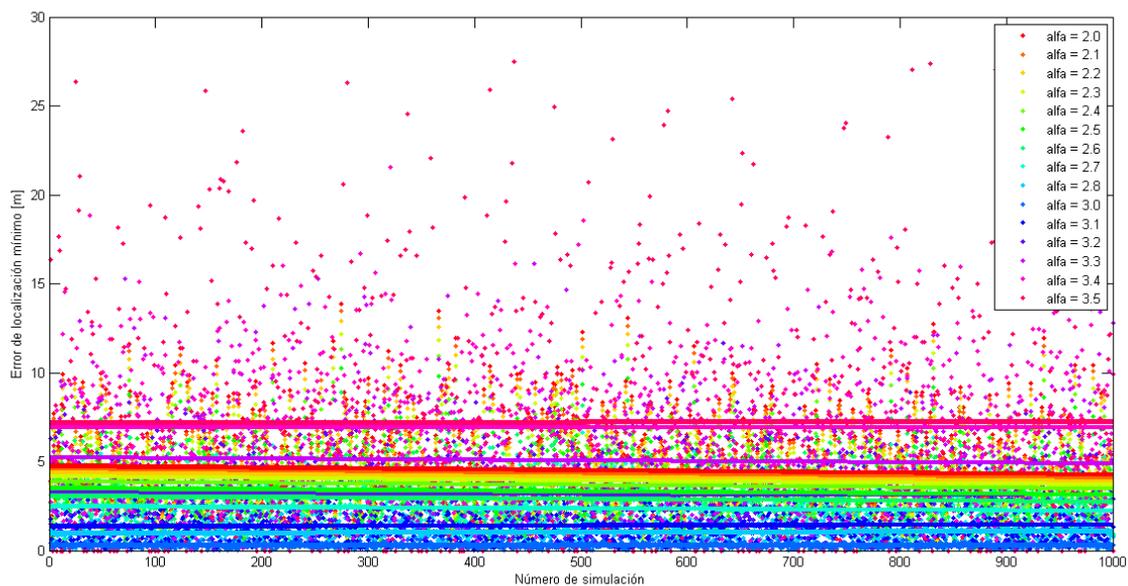


Figura 4.13: Error medio del método RSSI, resuelto mediante NLLS para cada simulación

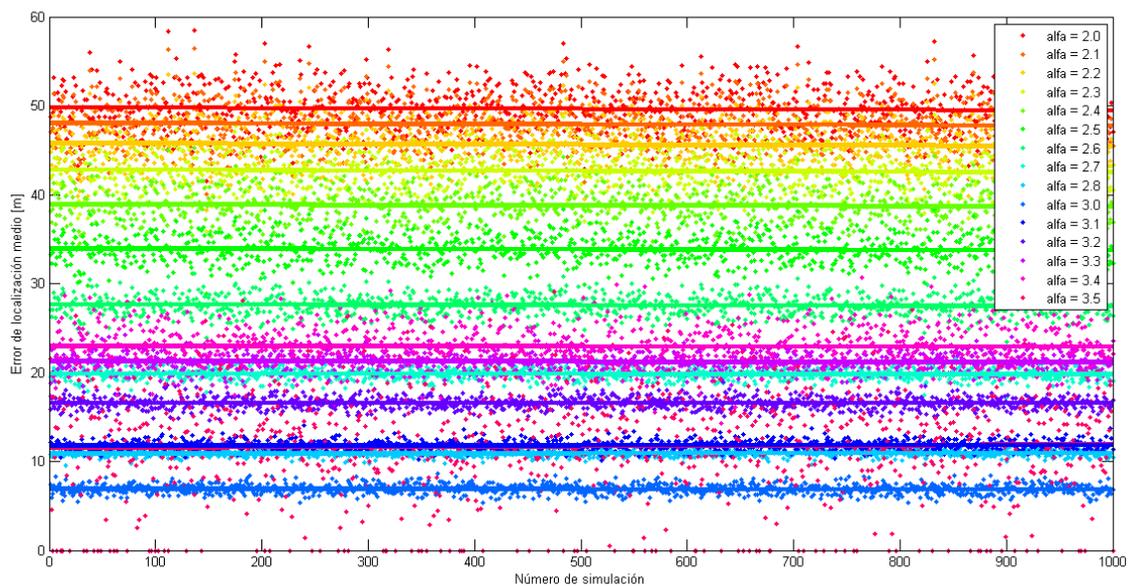
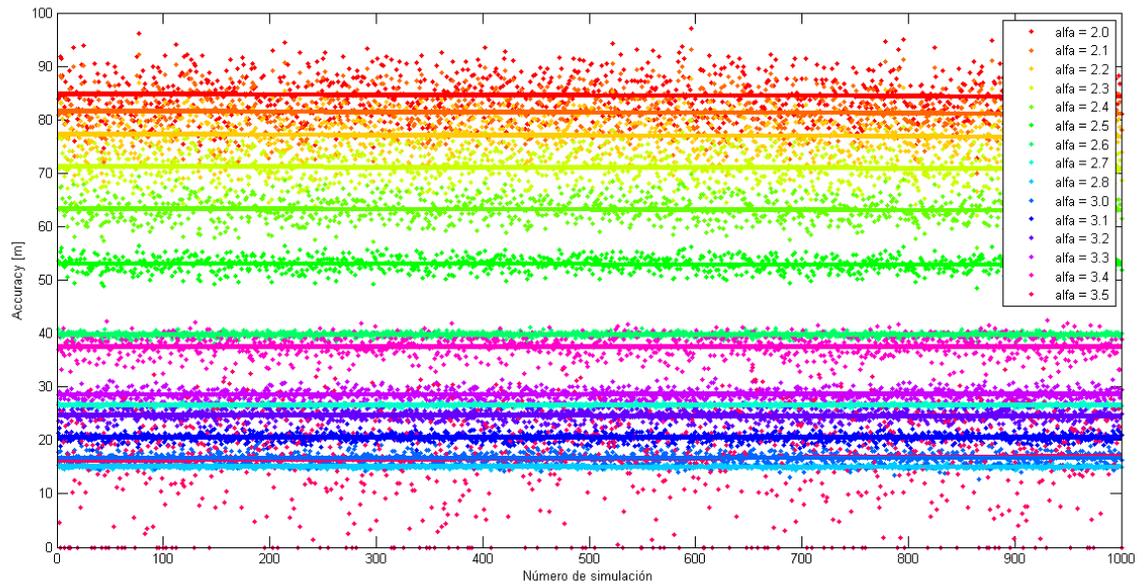


Figura 4.14: Accuracy del método RSSI para una precisión del 95 %, resuelto mediante NLLS para cada simulación



La comparación de las escalas cromáticas de los gráficos 4.11 y 4.13 con la figura 4.14 permiten rápidamente visualizar que la tendencia de estas dos variables se asemeja bastante a la presentada por la exactitud, lo que aumenta aún más la convicción de que se trata de un buen indicador del desempeño general del modelo, quedando otros criterios como el error máximo o la cobertura para diferenciar entre dos modelos similares.

Por su parte, la revisión de la figura 4.12 muestra que el error mínimo de un modelo, si bien puede ser un criterio de diseño relevante, no guarda mayor conexión con el resto de las variables o el desempeño general del modelo, por lo que generalmente es un parámetro en el cual no debiesen realizarse mayores reparos.

# Capítulo 5

## Conclusiones

### 5.1. Resultados y discusión

Con la simulación y posterior revisión de los resultados obtenidos del modelo de localización basado en RSS y mediante la resolución basada en mínimos cuadrados no-lineales, donde se obtuvieron valores de exactitud de 20 metros (que corresponden al 11,11 % de la distancia entre nodos), se logró cumplir con el objetivo de validar un modelo de localización para su posterior uso en la implementación de LBS. Lo que resulta más interesante es que los resultados son muy prometedores aún para la simpleza de la función de estimación de la distancia que se trata de la ecuación de Friis sin modificación alguna. La bibliografía sugiere que en caso de usar una función que tome en cuenta otros fenómenos como la dispersión por multiplicidad de caminos, o ampliando el método para incluir perfiles de RSS; los resultados del modelo no debiesen más que mejorar.

La implementación de los modelos de localización permite la obtención de los datos relevantes desde el simulador y su posterior procesamiento. Sin embargo, los resultados obtenidos para los modelos de ToA y TDoA son menos que satisfactorios; esto debido a que en la forma en que el simulador trata las señales, el tiempo de procesamiento no es constante y tampoco se puede predecir con la precisión necesaria para que los resultados de la estimación de posición sean útiles. La mejor solución a este problema es sin duda, utilizar un simulador diferente, ya sea que exista o implementado por el investigador, que le dé un tratamiento adecuado a los tiempos de desplazamiento y procesamiento de las señales; o por otro lado realizar la simulación con señales de menor velocidad, donde el error en la estimación del tiempo de vuelo no produzca errores no manejables en la estimación de la posición.

Lo señalado en el último párrafo nos lleva a otro punto planteado en los objetivos específicos: la factibilidad de utilizar un modelo basado en ToA o TDoA en aplicaciones reales está directamente limitado por la capacidad de quien lo implementa de medir correctamente el tiempo de procesamiento de la señal al interior de los terminales. El no poder realizar este proceso excluye estos métodos de cualquier posible solución para implementar la geolocalización. Por su parte, el valor de RSS de una señal es mucho más simple de obtener,

existiendo en la actualidad incluso aplicaciones para dispositivos móviles que realizan este trabajo.

Los supuestos iniciales acerca de la precisión y la exactitud como un par de parámetros adecuados para medir el desempeño de un modelo de geo-localización resultaron correctos porque existe una relación directa entre estos parámetros y los errores máximo y medio de un modelo. Sin embargo, algo que escapa a toda esta línea de pensamiento es el concepto de cobertura: debiese ser siempre importante considerar este parámetro en casos donde se sabe que la distribución de nodos es tal que no es posible asegurar que el terminal esté bajo la cobertura de 3 nodos ancla en todo momento. Se mostró también que el algoritmo de Levenberg-Marquardt permite hacer un intercambio de desempeño por cobertura, lo que sugiere que el balance de error y cobertura es algo que debiese estar presente en todo momento en que se diseñe un método de geo-localización.

Respecto al último de los objetivos, la cobertura es el único parámetro que parece tener una relación clara con el coeficiente de atenuación del medio. Las distintas mediciones de error resultaron ser mucho más erráticas en su comportamiento, salvo por el caso en el que se usó NLLS, en el cual se podría hablar de una relación inversa entre error de localización y la atenuación del medio. Otro punto importante a considerar es desde luego la potencia de transmisión, se mostró que para la potencia simulada en estos casos, es imposible realizar localización para valores de alfa mayores a 3.5, lo que corresponde a áreas urbanas densamente pobladas y con edificaciones en altura.

## 5.2. Trabajos futuros

El primer gran punto a tratar es la falta de herramientas adecuadas para la simulación de esta clase de modelos, los que pueden resultar muy útiles en el futuro. Las libertades que entrega el programar módulos en *Omnnet++* lo hacen una herramienta útil para la elaboración de un simulador que contribuya en el ámbito de las redes de comunicaciones móviles, y a su vez puede ser liberado como un simulador de uso público como los muchos proyectos llevados a cabo por distintas universidades a nivel mundial. De esto se desprende además, que contando con las herramientas de simulación se pueden evaluar modelos basados en tiempo de vuelo, ya que el intento realizado en este trabajo no fue positivo.

El desarrollo de un trabajo que se centre en un único modelo, permitiría además realizar mejoras sobre la función de localización planteada, y la modificación de otros parámetros para entender su incidencia en el desempeño del mismo. También la posibilidad de expandir los escenarios simulados para trabajar, por ejemplo, la localización iterativa de terminales que se describe en el capítulo 2 de este trabajo.

# Bibliografía

- [1] *IEEE 802.11 Wireless Local Area Networks*. <http://www.ieee802.org/11>.
- [2] *MiXiM Wiki*. <http://sourceforge.net/apps/trac/mixim/wiki>.
- [3] *User Manual Omnet++ version 4.2*. <http://omnetpp.org/doc/omnetpp/manual/usman.html>.
- [4] Ahson, Syed A. y Mohammad Ilyas (editores): *Location-Based Services Handbook: Applications, Technologies, and Security*. CRC Press, spanish1 edición, Julio 2010, ISBN 9781420071962. <http://amazon.com/o/ASIN/1420071963/>.
- [5] Bahl, P. y V.N. Padmanabhan: *RADAR: an in-building RF-based user location and tracking system*. En *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volumen 2, páginas 775–784 vol.2, 2000.
- [6] Barnes, M.: *HTTP-Enabled Location Delivery (HELD)*. IETF, RFC 5985, September 2010. [tools.ietf.org/html/rfc5985](http://tools.ietf.org/html/rfc5985).
- [7] Bellavista, P., A. Kupper y S. Helal: *Location-Based Services: Back to the Future*. *Pervasive Computing, IEEE*, 7(2):85–89, april-june 2008, ISSN 1536-1268.
- [8] Brida, P., P.Cepel y J. Duha: *Geometric Algorithm for Received Signal Strength Based Mobile Positioning*. En *Proc. of Czech and Slovak Technical Universities and URSI Committees*, volumen 14, 2005.
- [9] Capkun, S., K.B. Rasmussen, M. Cagalj y M. Srivastava: *Secure Location Verification with Hidden and Mobile Base Stations*. *Mobile Computing, IEEE Transactions on*, 7(4):470–483, april 2008, ISSN 1536-1233.
- [10] Commission, Federal Communications: *Wireless Enhanced 9-1-1 (E9-1-1) Coordination Initiative*. <http://transition.fcc.gov/pshs/services/911-services/enhanced911/coordination.html>.
- [11] Connolly, Gene Michael: *Distributed Traceroute Approach to Geographically Locating IP Devices*. Tesis de Licenciatura, University of Maine, May 2003.
- [12] Cramér, Harald: *Mathematical Methods of Statistics. (PMS-9)*. Princeton University

- Press, Marzo 1999, ISBN 9780691005478. <http://amazon.com/o/ASIN/0691005478/>.
- [13] Deasy, T.P. y W.G. Scanlon: *Stepwise refinement algorithms for prediction of user location using receive signal strength indication in infrastructure WLANs*. En *High Frequency Postgraduate Student Colloquium, 2003*, páginas 116 – 119, sept. 2003.
- [14] Doherty, L., K.S.J. pister y L. El Ghaoui: *Convex position estimation in wireless sensor networks*. En *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volumen 3, páginas 1655 –1663 vol.3, 2001.
- [15] Haeberlen, A., E. Flannery, A. Ladd, A. Rudys, D. Wallach y L. Karvaki: *Practical Robust Localization over Large-scale 802.11 Wireless Networks*. En *ACM MOBICOM*, páginas 70–84. ACM Press, September 2004.
- [16] Hernandez, Edwin y Sumi Helal: *Predictive Mobile IP for rapid mobility*. En *Local Computer Networks, 29th Annual IEEE International Conference on*, páginas 684–689, 2004.
- [17] Hoesel, L. van y P. Havinga: *A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks*. En *1st Int. Workshop on Networked Sensing Systems (INSS 2004)*, 2004.
- [18] ITU: *What really is a Third Generation (3G) Mobile Techonology?*, June 2009. [http://www.itu.int/ITU-D/tech/FORMER\\_PAGE\\_IMT2000/Documents/IMT2000/What\\_really\\_3G.pdf](http://www.itu.int/ITU-D/tech/FORMER_PAGE_IMT2000/Documents/IMT2000/What_really_3G.pdf).
- [19] ITU-R: *ITU global standard for international mobile telecommunications 'IMT-Advanced'*, March 2008. <http://www.itu.int/ITU-R/information/promotion/e-flash/2/article4.html>.
- [20] Liu, J., Y. Zhang y F. Zhao: *Robust distributed node localization with error management*. En *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, páginas 250–261. ACM, 2006.
- [21] Liu, Yunhao y Zheng Yang: *Location, Localization, and Localizability: Location-awareness Technology for Wireless Networks*. Springer, spanish1st edition. edición, Noviembre 2010, ISBN 9781441973702. <http://amazon.com/o/ASIN/1441973702/>.
- [22] Loschmidt, P., G. Gaderer y T. Sauter: *Clock Synchronization for Wireless Positioning of COTS Mobile Nodes*. En *Precision Clock Synchronization for Measurement, Control and Communication, 2007. ISPCS 2007. IEEE International Symposium on*, páginas 64 –69, oct. 2007.
- [23] Mota-García, Edmar y Rogelio Hasimoto-Beltran: *Fast Geographic Internet Mapping System for Location-Based Services*. IMACST, 1(1):37–43, December 2010.
- [24] Niculescu, D. y B. Nath: *DV based positioning in ad hoc networks*. *Telecommunication Systems*, 22(1):267–280, 2003.

- [25] Patil, Basavaraj, Yousuf Saifullah, Stefano Faccin, Srinivas Sreemanthula, Lachu Aravamudhan, Sarvesh Sharma y Risto Mononen: *IP in Wireless Networks*. Prentice Hall, Febrero 2003, ISBN 9780130666482. <http://amazon.com/o/ASIN/0130666483/>.
- [26] Polk, J., M. Linsner, M. Thomson y B. Aboba: *Dynamic Host Configuration Protocol Options for Coordinate-Based Location Configuration Information*. IETF, RFC 6225, July 2011. [tools.ietf.org/html/rfc6225](http://tools.ietf.org/html/rfc6225).
- [27] Poovendran, Radha, Cliff Wang y Sumit Roy (editores): *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks (Advances in Information Security)*. Springer, spanishsoftcover reprint of hardcover 1st ed. 2007 edición, Noviembre 2010, ISBN 9781441940964. <http://amazon.com/o/ASIN/1441940960/>.
- [28] PRQC, ATIS Committee: *ATIS Telecom Glossary 2012*, 2012. <http://www.atis.org/glossary/definition.aspx?id=6470>.
- [29] Rousseeuw, P.J., A.M. Leroy y J. Wiley: *Robust regression and outlier detection*, volumen 3. Wiley Online Library, 1987.
- [30] Savarese, C., J. Rabaey y K. Langendoen: *Robust positioning algorithms for distributed ad-hoc wireless sensor networks*. En *USENIX technical annual conference*, 2002.
- [31] Schulzrinne, H.: *Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information*. IETF, RFC 4776, November 2006. [tools.ietf.org/html/rfc4776](http://tools.ietf.org/html/rfc4776).
- [32] Shang, Yi y W. Ruml: *Improved MDS-based localization*. En *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volumen 4, páginas 2640 – 2651 vol.4, march 2004.
- [33] Singh, R., M. Gandetto, M. Guainazzo, D. Angiatti y C. Ragazzoni: *A novel positioning system for static location estimation employing WLAN in indoor environments*. IEEE PIMRC, 3:1762–66, 2004.
- [34] Varga, András: *Omnet++ Installation Guide version 4.2.2*, 2011. <http://omnetpp.org/doc/omnetpp/InstallGuide.pdf>.
- [35] Yang, Zheng y Yunhao Liu: *Quality of Trilateration: Confidence-Based Iterative Localization*. Parallel and Distributed Systems, IEEE Transactions on, 21(5):631 –640, may 2010, ISSN 1045-9219.

# Apéndice A

## A.1. Código de MiXiM

### A.1.1. omnet.ini

```
[General]
cmdenv-express-mode = true
network = MyTDMA

#####
# Simulation parameters                                     #
#####
sim-time-limit = 600s
**.**.coreDebug = false
**.playgroundSizeX = 180m
**.playgroundSizeY = 180m
**.playgroundSizeZ = 180m
**.numNodes = 5

*.node[0].nic.mac*.vector-recording = true
**.vector-recording = false

#####
# WorldUtility parameters                                   #
#####
**.world.useTorus = false
**.world.use2D = true

#####
# Parameters for the ConnectionManager                     #
#####
**.connectionManager.carrierFrequency = 2.412e9Hz
# max transmission power [mW]
**.connectionManager.pMax = 110.11mW
```

```

# signal attenuation threshold [dBm]
**.connectionManager.sat = -84dBm
# path loss coefficient alpha
**.connectionManager.alpha = 3
**.connectionManager.sendDirect = false

#####
#           Parameters for the Host           #
#####
**.nic.sleepCurrent = 0.02mA
**.nic.rxCurrent = 16.4mA
**.nic.decodingCurrentDelta = 0mA
**.nic.txCurrent = 17mA
**.nic.setupRxCurrent = 8.2mA
**.nic.setupTxCurrent = 8.2mA
**.nic.rxTxCurrent = 17mA
**.nic.txRxCurrent = 17mA

##### Phy parameters #####

**.nic.phyType = "org.mixim.modules.phy.MyPhyLayer"
**.phy.usePropagationDelay = false
**.phy.thermalNoise = -110dBm
**.phy.useThermalNoise = true
**.phy.analogueModels = xmldoc("config.xml")
**.phy.decider = xmldoc("config.xml")
**.phy.maxTXPower = 110.11mW
**.phy.sensitivity = -84dBm

##### MAC layer parameters #####
**.mac.headerLength = 24bit
# the length of the queue - if exceeded, new packets are dropped
**.mac.queueLength = 50
**.mac.bitrate = 100000bps
**.mac.txPower = 100
**.mac.defaultChannel = 0
# duration of one slot
**.mac.slotDuration = 0.1s
# number of slots
**.mac.numSlots = 8
**.mac.reservedMobileSlots=0
**.mac.debug = true
###.mac.animation = false

##### Battery statistic parameters #####
**.batteryStats.debug = false
**.batteryStats.detail = true

```

```

**.batteryStats.timeSeries = false

##### Battery parameters #####
**.battery.debug = false
**.battery.nominal = 1000mAh
**.battery.capacity = 1000mAh
**.battery.voltage = 3.3V
**.battery.resolution = 0.1s
**.battery.publishDelta = 0
**.battery.publishTime = 0.5s
**.battery.numDevices = 1

##### Application layer parameters #####
**.node[*].applicationType = "SensorApplLayer"
**.appl.trafficType = "periodic"
**.appl.trafficParam = 1s #in seconds
**.appl.broadcastPackets = true
**.appl.nbPackets = 3
**.appl.initializationTime = 10s
**.node[*].applicationType = "BurstApplLayer"
**.node[*].appl.debug = false
**.node[*].appl.headerLength = 512bit
**.node[*].appl.burstSize = 1
##### NETW layer parameters #####
**.node[*].networkType = "BaseNetwLayer"
**.node[*].netwl.debug = false
**.node[*].netwl.stats = false
**.node[*].netwl.headerLength = 32bit
##### Mobility parameters #####
**.node[*].mobilityType = "ConstSpeedMobility"
**.node[*].mobility.debug = false
**.node[*].mobility.initFromDisplayString = false
**.node[0].mobility.speed = 7mps
**.node[*].mobility.updateInterval = 0.1s

**.node[0].mobility.initialX = 90m
**.node[0].mobility.initialY = 90m

**.node[1].mobility.initialX = 45m
**.node[1].mobility.initialY = 45m

**.node[2].mobility.initialX = 135m
**.node[2].mobility.initialY = 45m

**.node[3].mobility.initialX = 45m
**.node[3].mobility.initialY = 135m

```

```

**.node[4].mobility.initialX = 135m
**.node[4].mobility.initialY = 135m

**.node[*].mobility.constraintAreaMinX = 0m
**.node[*].mobility.constraintAreaMinY = 0m
**.node[*].mobility.constraintAreaMaxX = 180m
**.node[*].mobility.constraintAreaMaxY = 180m

```

## A.1.2. MyMacLayer.cc

```

void MyMacLayer::initialize(int stage)
{
BaseMacLayer::initialize(stage);

    if (stage == 0) {

        queueLength = par("queueLength");
        slotDuration = par("slotDuration");
        bitrate = par("bitrate");
        headerLength = par("headerLength");
        coreEV << "headerLength is: " << headerLength << endl;
        numSlots = par("numSlots");
        // the first N slots are reserved for mobile nodes to be able to function normally
        reservedMobileSlots = par("reservedMobileSlots");
        txPower = par("txPower");

        droppedPacket.setReason(DroppedPacket::NONE);
        nicId = getNic()->getId();
        debugEV << "My Mac address is" << myMacAddr << " and my Id is " << myId << endl;

        macState = INIT;

        slotChange = new cOutVector("slotChange");

        // how long does it take to send/receive a control packet
        controlDuration = ((double)headerLength + (double)numSlots + 16) / (double)bitrate
        coreEV << "Control packets take : " << controlDuration << " seconds to transmit\n"

        pktRSSI.setName("RSSI");
        pktSrc.setName("Origen");
        pktDst.setName("Destino");
        coordX.setName("CoordenadaX");
        coordY.setName("CoordenadaY");

```

```

}

Coord* MyMacLayer::getPosition(){
    Coord *back=NULL;
    ChannelAccess *const pChanAccess = dynamic_cast<ChannelAccess *const>(this->getPar
    if(pChanAccess != NULL){
        ChannelMobilityPtrType pMobType = pChanAccess->getMobilityModule();
        if(pMobType != NULL){
            back = new Coord(pMobType->getCurrentPosition());
        }
    }
    return back;
}

cPacket *MyMacLayer::decapsMsg(MacPkt *frame) {
    cPacket *m = frame->decapsulate();
    setUpControlInfo(m, frame->getSrcAddr());
    debugEV << " message decapsulated " << endl;
    PhyToMacControlInfo* cinfo = static_cast<PhyToMacControlInfo*> (frame->getControlI
    const MyDeciderResult80211* result = static_cast<const MyDeciderResult80211*> (cin
    Coord* currentPos = getPosition();
    coordX.record(currentPos->getX());
    coordY.record(currentPos->getY());
    pktRSSI.record(result->getRSSI());
    pktSrc.record(frame->getSrcAddr());
    pktDst.record(frame->getDestAddr());
    return m;
}

```

### A.1.3. BurstApplLayer.cc

```

void BurstApplLayer::handleSelfMsg(cMessage *msg)
{
    switch(msg->getKind())
    {
        case SEND_BROADCAST_TIMER:
            for(int i=0; i<burstSize; i++) {
                sendBroadcast();
            }
            scheduleAt(simTime()+5,msg);
            break;
        default:
            EV <<" Unkown selfmessage! -> delete, kind: "<<msg->getKind()<<endl;
            break;
    }
}

```

```
}
```

#### A.1.4. MyDecider80211.cc

```
DeciderResult* MyDecider80211::checkIfSignalOk(AirFrame* frame)
{
// check if the snrMapping is above the Decider's specific threshold,
// i.e. the Decider has received it correctly

// first collect all necessary information
Mapping* snrMap = calculateSnrMapping(frame);
assert(snrMap);

Signal& s = frame->getSignal();
simtime_t start = s.getReceptionStart();
simtime_t end = s.getReceptionEnd();

start = start + RED_PHY_HEADER_DURATION; //its ok if the phy header is received only
//partly - TODO: maybe solve this nicer
Argument min(DimensionSet::timeFreqDomain);
min.setTime(start);
min.setArgValue(Dimension::frequency, centerFrequency - 11e6);
Argument max(DimensionSet::timeFreqDomain);
max.setTime(end);
max.setArgValue(Dimension::frequency, centerFrequency + 11e6);

Mapping::argument_value_t snirMin = MappingUtils::findMin(*snrMap, min, max, Argument::Map

deciderEV << " snrMin: " << snirMin << endl;

ConstMappingIterator* bitrateIt = s.getBitrate()->createConstIterator();
bitrateIt->next(); //iterate to payload bitrate indicator
double payloadBitrate = bitrateIt->getValue();
delete bitrateIt;
double rssi = calcChannelSenseRSSI(start,end);

MyDeciderResult80211* result = 0;

if (snirMin > snrThreshold) {
if(packetOk(snirMin, frame->getBitLength() - (int)PHY_HEADER_LENGTH, payloadBitrate)) {
result = new MyDeciderResult80211(true, payloadBitrate, snirMin, rssi);
} else {
deciderEV << "Packet has BIT ERRORS! It is lost!\n";
result = new MyDeciderResult80211(false, payloadBitrate, snirMin, rssi);
}
}
```

```

} else {
deciderEV << "Packet has ERRORS! It is lost!\n";
result = new MyDeciderResult80211(false, payloadBitrate, snirMin, rssi);
}

delete snrMap;
snrMap = 0;

return result;
}

```

### A.1.5. MyDeciderResult80211.h

```

class MIXIM_API MyDeciderResult80211 : public DeciderResult{
protected:
    /** @brief Stores the bit-rate of the transmission of the packet */
    double bitrate;

    /** @brief Stores the signal to noise ratio of the transmission */
    double snr;

    /** @brief Stores the received signal strength */
    double rssi;
public:

    /**
     * @brief Initialises with the passed values.
     *
     * "bitrate" defines the bit-rate of the transmission of the packet.
     */
    MyDeciderResult80211(bool isCorrect, double bitrate, double snr, double rssi):
        DeciderResult(isCorrect), bitrate(bitrate),snr(snr),rssi(rssi)
    {}

    /**
     * @brief Returns the bit-rate of the transmission of the packet.
     */
    double getBitrate() const { return bitrate; }

    /**
     * @brief Returns the signal to noise ratio of the transmission.
     */
    double getSnr() const { return snr; }

    /**

```

```

    * @brief Returns the received signal strength of the transmission.
    */
    double getRSSI() const { return rssi; }

};

```

## A.2. Código de Matlab

### A.2.1. freeSpaceDistance.m

```

function distance = freeSpaceDistance(Pr, alpha)
%freeSpaceDistance(Pr,Pt,Gr,Gt,lambda) returns the distance between two
%
%                               nodes based on the Friis
%                               transmission equation.
% Current version uses fixed parameters obtained from the OMNET++
% simulation

%Constant Variables
Gt=1;
Gr=1;                               %Gain in times, not dB
SoL=299792458;                       %Speed of light in m/s
frecuency=2.412*10^9;                 %Carrier frecuency in Hz
lambda=SoL/frecuency;
%alpha=2;                             %Attenuation coefficient
Pt=110.11*10^(-3);                   %Tx Power in watts
maxDistance=140;

%handle nodes out of coverage area
if Pr==0
    distance=maxDistance;
else
distance = (lambda/(4*pi()))^(4/alpha)*((Pt/Pr)*Gt*Gr)^(2/alpha);
end
end

```

### A.2.2. Order.m

```

function [pos, rssi, tdoaDistance] = Order(data, names, anchors)
%ORDER(DATA,NAMES) returns a series of matrices containing data from
%
% 'data'sorted by its data type. 'Names' is used to
% identify data types.
% ORDER currently returns a 'pos' vector with a structure compatible with

```

```

% the localization functions, and a 'rssi' matrix that groups up values
% with a similar time into a single event (row).
timei=0;          %Initial time is used to keep track of last iteration time
                  %value
delta=0.2;        %Maximum time difference between any 2 signals in the same
                  %event
numBeacon=5;     %Number of beacons
pos = zeros(0,11);
rssi = zeros(0,5);
tdoaDistance = zeros(0,5);
ind=1;           %index used to move through 'data'
posInd=0;        %index used to move through 'pos'
rssiInd=0;       %index used to move through 'rssi'
tdoaInd=0;
SoL = 299792458; %Speed of light in m/s
refrac_index = 1.000293;

%Move through the data, sorting into pos
while(true)
    %get the next RSSI measurement
    auxName=index2name(names,data(ind,1));
    if ~isempty(strfind(auxName,'RSSI'))
        %read the time of the RSSI value
        time = data(ind,2);
        %call groupData with the given time
        [jump,auxGroup]=groupData(data,names,time);

        %build rssi
        %check time distance between current and last iteration. Save data
        %in a new row (event) if distance is too large.
        if time-timei>delta
            rssiInd=rssiInd+1;
            posInd=posInd+1;
            tdoaInd=tdoaInd+1;
            rssi(rssiInd,1)=time;
            pos(posInd,1)=time;
            tdoa_time = time;
        end
        %check source (BROADCAST) or destination (UNICAST) to sort data in
        %a given row (event)
        if auxGroup(3,1)==1 || auxGroup(4,1)==1
            %rssi(rssiInd,2)=freeSpaceDistance(auxGroup(2,1));
            rssi(rssiInd,2)= auxGroup(2,1);
            tdoaDistance(tdoaInd,2) = SoL/refrac_index*(time - tdoa_time);
        elseif auxGroup(3,1)==2 || auxGroup(4,1)==2
            rssi(rssiInd,3)=auxGroup(2,1);
            tdoaDistance(tdoaInd,3) = SoL/refrac_index*(time - tdoa_time);
        end
    end
    ind=ind+1;
end

```

```

elseif auxGroup(3,1)==3 || auxGroup(4,1)==3
    rssi(rssiInd,4)=auxGroup(2,1);
    tdoaDistance(tdoaInd,4) = SoL/refrac_index*(time - tdoa_time);
elseif auxGroup(3,1)==4 || auxGroup(4,1)==4
    rssi(rssiInd,5)=auxGroup(2,1);
    tdoaDistance(tdoaInd,5) = SoL/refrac_index*(time - tdoa_time);
else
    display('Exceeded number of beacons!');
end

%build pos

pos(posInd,4:5)=anchors(1,:);
pos(posInd,6:7)=anchors(2,:);
pos(posInd,8:9)=anchors(3,:);
pos(posInd,10:11)=anchors(4,:);
%pos(posInd+1,:)=auxGroup(5,:);
%pos(posInd+2,:)=auxGroup(6,:);
%pos(posInd+3,:)=auxGroup(7,:);
%pos(posInd+4,:)=auxGroup(8,:);
%pos(posInd+5,:)=auxGroup(9,:);

if auxGroup(5,:)~= [0,0]
pos(posInd,2:3)=auxGroup(5,:);
end
%if auxGroup(6,:)~= [0,0]
%pos(posInd,4:5)=auxGroup(6,:);
%end
%if auxGroup(7,:)~= [0,0]
%pos(posInd,6:7)=auxGroup(7,:);
%end
%if auxGroup(8,:)~= [0,0]
%pos(posInd,8:9)=auxGroup(8,:);
%end
%if auxGroup(9,:)~= [0,0]
%pos(posInd,10:11)=auxGroup(9,:);
%end

%set index values for next iteration
%posInd=posInd+numBeacon;
ind=jump;
timei=time;
else
    ind=ind+1;
end
if ind>size(data,1)
    break

```

```

    end
end

end

```

### A.2.3. linearLS.m

```

function lls = linearLS(pos,distance)
%linearLS(pos,distance)    returns the position of a single node
%                          calculated by trilateration, using linear least
%                          squares as its optimization algorithm
%
% This method does a trilateration-based localization. It begins by
% linearizing the equation system described by the spherical constraints
% and then applying a linear least squares criteria to compute the
% position.
%
% In order for the function to work properly, pos must have at least the
% position of 4 unique anchor nodes.

%constant variables
x=1;    %column in pos with the x position
y=2;    %column in pos with the y position
lin=2;  %row in pos with the linearizing beacon (can be anyone, but the
        %program uses the first beacon after the mobile one).

A=zeros(0,2);
b=zeros(0,1); %Matrices of the linear system Ax=b
d2=zeros(0,1);%Array to store inter-anchor distances

%remember to skip first beacon to use it as linearizing tool
for i=lin+1:size(pos,1)
    %Build A
    A(i-lin,x)= pos(i,x)-pos(lin,x);
    A(i-lin,y)= pos(i,y)-pos(lin,y);
    %calculate dij
    d2(i-lin)= (pos(i,x)-pos(lin,x))^2+(pos(i,y)-pos(lin,y))^2;
    %Build b
    b(i-lin,1)=(distance(lin-1)^2-distance(i-1)^2+d2(i-lin))/2;
end
%Apply Linear Least squares
lls=(A\b)'+[pos(lin,x),pos(lin,y)]; %the solution has to be referred to the
                                     %linearizing node's position

end

```



# Apéndice B

## B.1. Redes de telecomunicaciones

Una red de telecomunicaciones corresponde a una organización de estaciones capaces de comunicarse entre si [28]. Estas redes pueden clasificarse en base a una serie de parámetros tales como: medio de transporte de la información; protocolo de comunicación; cobertura; y topología.

Algunos ejemplos clásicos de redes de telecomunicaciones son: las redes de computadores; el Internet; y la red de telefonía fija.

### B.1.1. Redes inalámbricas

Dentro de la clasificación de redes en base a la tecnología de transmisión de datos es posible distinguir dos grandes grupos: tecnologías alámbricas, como par trenzado y fibra óptica; y tecnologías inalámbricas, como las ondas de radio e infrarrojas.

Una red inalámbrica es aquella que utilice una tecnología de transmisión de la segunda categoría mencionada. La implementación de estas tecnologías ocurre en la capa física del modelo OSI. [25]

## B.2. Estándares para comunicación inalámbrica

### B.2.1. IEEE 802.11

El IEEE 802.11 corresponde a una familia de estándares para la implementación de comunicación entre computadores mediante el uso de redes de área local inalámbricas (WLAN). Este estándar fue creado en 1997 por el grupo de trabajo IEEE P802.11, parte del IEEE LAN MAN Standards Committee (LMSC) y es además la base de los productos distribuidos bajo la marca Wi-Fi. Su mantención y posteriores enmiendas son también responsabilidad

del mismo grupo de trabajo [1].

Dentro de la familia, el primer estándar de uso masivo fue el 802.11b, seguido de los 802.11g y 802.11n. Todos ellos utilizan el mismo protocolo básico; variando la técnica de modulación utilizada. Estas técnicas siempre corresponden a casos de modulación half-duplex para señales transmitidas por el aire. Gracias a esto, en la actualidad es común contar con equipos que pueden operar bajo 2 o 3 de estos estándares.

Los servicios 802.11b y 802.11g operan en la banda de 2.4 GHz, sufriendo interferencia de otros equipos electrónicos como los hornos microondas y telefonos inalámbricos. En el caso del estándar 802.11n opera tanto en la banda de 2.4 GHz como en la de 5 GHz. Actualmente se encuentra en desarrollo un nueva versión del IEEE 802.11, que utilizará exclusivamente la banda de 5 GHz dejando así los problemas de interferencia actuales.

### **B.2.2. 3G**

3G o tercera generación es el nombre comercial de una serie de tecnologías utilizadas en telefonía móvil, que cumplen con la especificación IMT-2000 definida por la Unión Internacion de Telecomunicaciones (ITU)

El concepto de generación dentro de la telefonía móvil va asociado con la idea de que cada una de ellas tiene mejoras significativas en el desempeño y en las capacidades. Así se tiene una primera generación donde se ofrecían servicios celulares basados en redes de radio analógicas; para posteriormente dar paso a una segunda generación con una infraestructura de radio digital.

Debido al amplio desarrollo de redes 2G, la tercera generación no tiene una identificación tan clara como las anteriores, por el contrario se trata de una familia de tecnologías que bien revoluciona o evoluciona a partir de su versión anterior.

El rol de la ITU en la definición del estándar se enfocó principalmente en fijar cotas mínimas de desempeño que marcaran una diferencia significativa respecto a la generación anterior. Además se determino contar con una única banda de frecuencias en la cual operarían las tecnologías 3G; esta idea sería posteriormente descartada en pos del modelo multibanda. [18]

Actualmente distintas agrupaciones participan en el desarrollo de una nueva generación para la telefonía móvil, cuya principal característica es el reemplazo de las actuales redes de conmutación de circuitos por redes de conmutación de paquetes basadas en el protocolo IP [19]

# Apéndice C

## C.1. Gráficos

### C.1.1. Gráficos de resultados del modelo de prueba

Figura C.1: alfa = 2.0

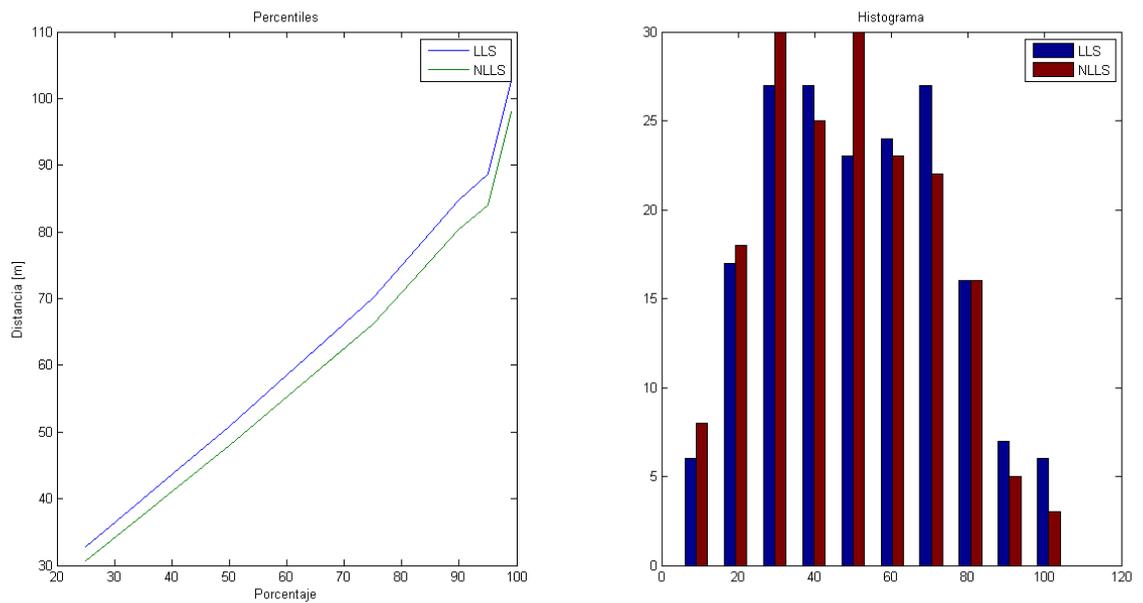


Figura C.2:  $\alpha = 3.3$

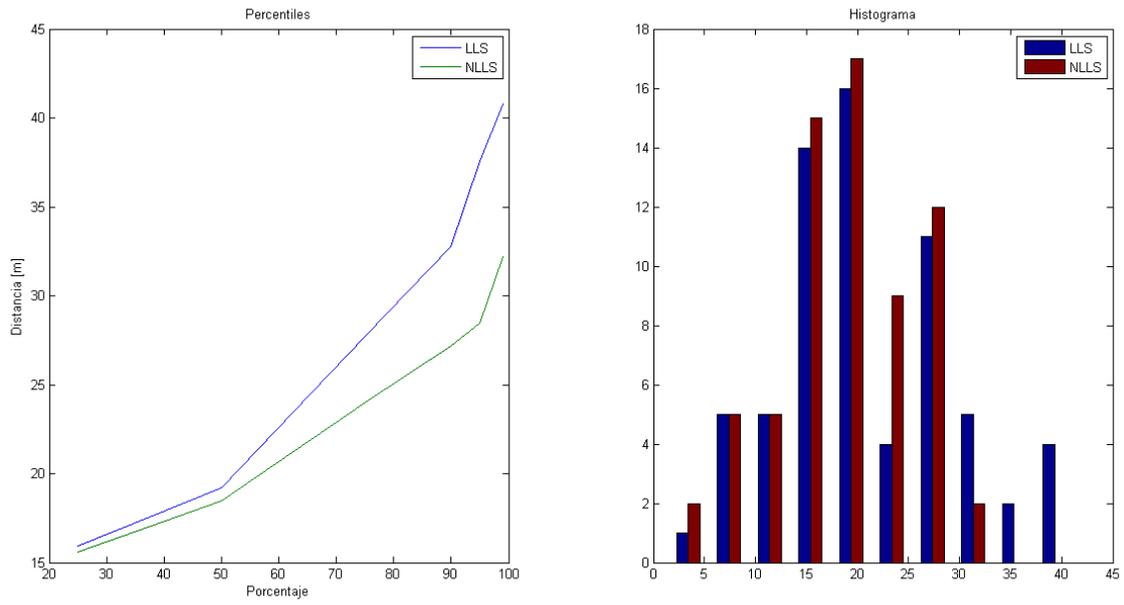


Figura C.3: Potencia al 90 %

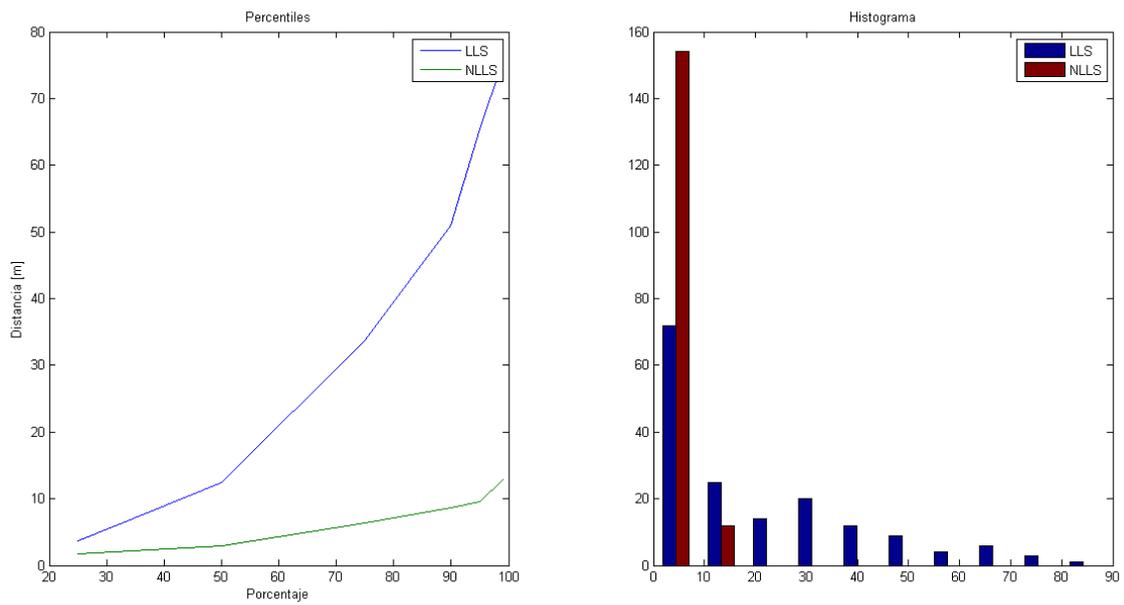


Figura C.4: Potencia al 110 %

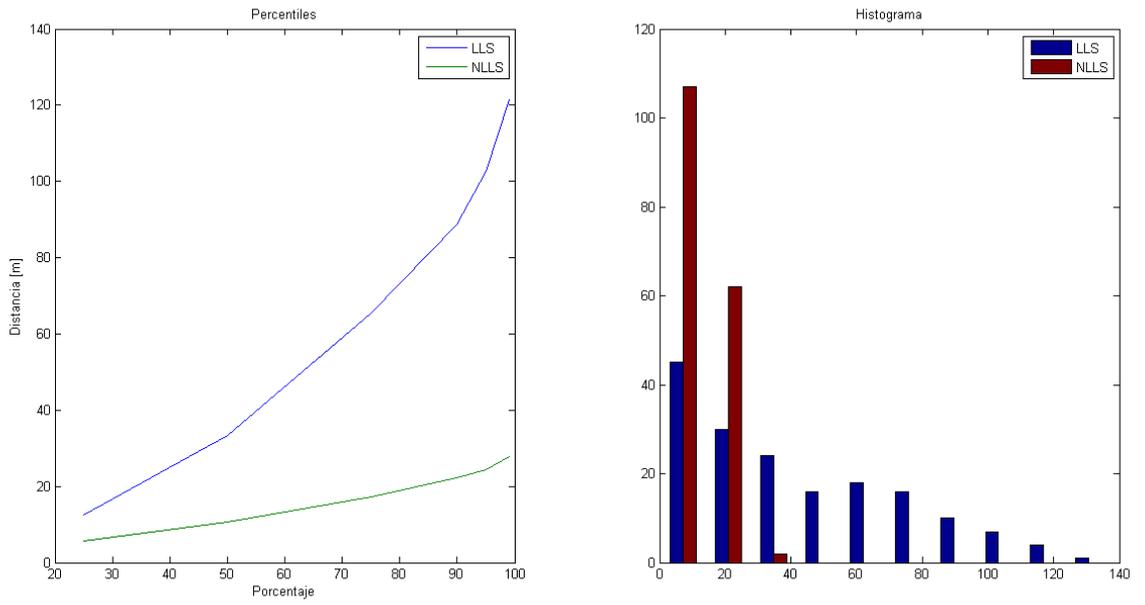


Figura C.5: Ruido térmico al 90 %

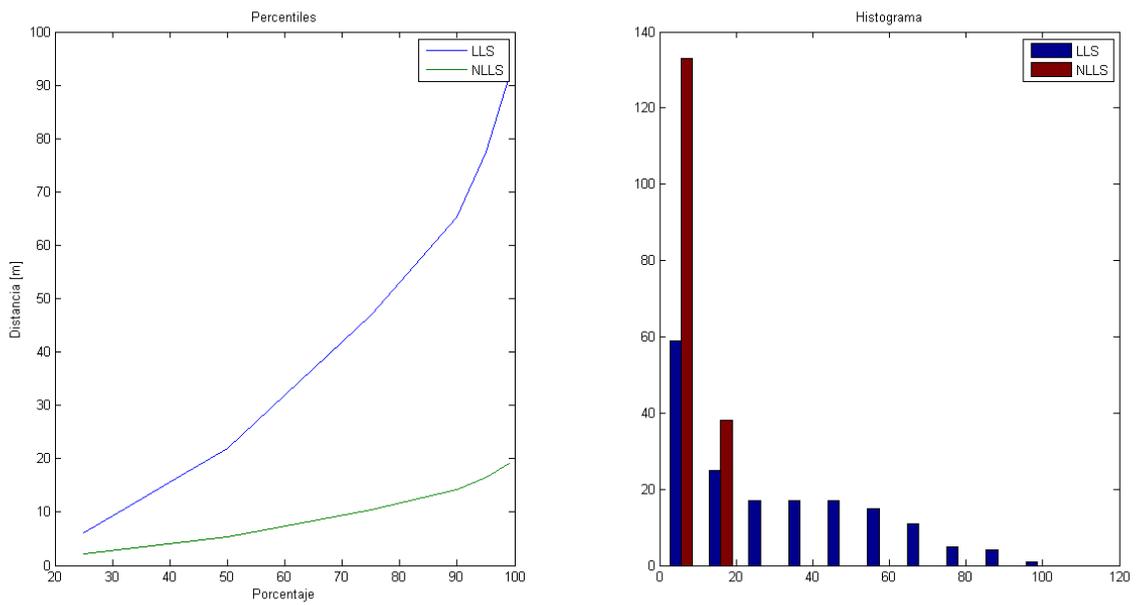


Figura C.6: Ruido térmico al 110 %

