UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica

Universitat Politècnica de València

Diseño e implementación de un modelo del protocolo AURP para redes de sensores subacuáticas

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Pablo Cardós Marqués

Tutor: Juan Vicente Capella Hernández

Julio de 2017

Resumen

Esta memoria versa sobre el estudio, diseño e implementación en el simulador especializado NS3 de un protocolo de red de sensores subacuático (AURP) definido en [1]. Durante los últimos años ha crecido el interés de esta tecnología debido a sus diversas aplicaciones tales como la minería, estudio marino, ecología, militar, etc. Se compara los resultados de la simulación con los presentados en el artículo antes citado y adicionalmente se ha examinado su comportamiento dese otras perspectivas no incluidas en ese artículo. Como consecuencia, se han identificado algunos problemas que presenta el protocolo tanto en su eficacia como en su consumo energético. Finalmente se propone posibles soluciones.

Palabras clave: red, sensores, subacuático, submarino, simulación, NS3, protocolo.

Abstract

This paper focuses on the study, design and implementation by means of the specialized software NS3 of a protocol for an underwater sensor network (AURP) as defined by [1]. In recent years this technology has received growing attention due to its different applications such as mining, marine studies, ecology, military, etc. Simulation results are compared with the corresponding ones presented in the cited paper and additionally its behavior has been examined from new perspectives. As a consequence, a number of problems have been pointed out related to its efficacy and power consumption. Finally some solutions are proposed to cope with these drawbacks.

Keywords: network, sensors, underwater, submarine, simulator, NS3, protocol.

Tabla de contenidos

1. Introducción

1.1. Introducción y Motivación del trabajo

Los estudios sobre distintos aspectos de las tecnologías de las telecomunicaciones han sido muy numerosos durante las últimas décadas. Habitualmente estas comunicaciones se realizan mediante conexión cableada, por aire o vacío mediante hilos, radiofrecuencia, ultrasonidos, etc. Sin embargo, con frecuencia las comunicaciones acuáticas quedan relegadas a un segundo plano por motivos tales como las características del medio, que se expondrán posteriormente. Además, no existía motivación para investigar puesto que hay soluciones más eficientes para largas distancias como son los medios mencionados anteriormente. En consecuencia las comunicaciones subacuáticas y sus protocolos han recibido poca atención de la comunidad científica.

Recientemente las redes de sensores han motivado un creciente interés en esta área. La red de sensores es un conjunto de computadoras con pocas capacidades, bajo consumo y sencillas que captan información del medio mediante sus sensores. Estos ordenadores también son llamados nodos o sensores. Estos nodos se comunican con una puerta de enlace o sumidero que recopila la información que los nodos generan y la trata posteriormente como detallan [2] y [3].

La red de sensores está ampliamente extendida a nivel industrial, militar y en menor medida doméstico o domótico. Pero actualmente está empezando a introducirse también en el mar y para ello se necesita una tecnología de comunicación adecuada. Algunas de las causas que permiten entender el creciente número de trabajos científicos en comunicaciones en red de sensores subacuáticas son descritas por [4] :

- Monitorización ambiental. La red de sensores es capaz de supervisar el estado de variables ambientales. Tales como productos químicos, medicamentos o insecticidas que son agentes contaminantes. Como salinidad, temperatura u oxígeno en el agua para determinan cambios en un ecosistema y cambio climático. O control de población tanto sea de peces como microbios. Tanto en entornos marinos como ríos, lagos u otros entornos.

- Cartografiado. La red de sensores es capaz de cartografiar de forma precisa los entornos subacuáticos. Un ejemplo de ello es la cartografía de parte de la costa de florida gracias a una red de sensores y un AUV (Vehículo autónomo).

- Exploración subacuática. Con los sensores subacuáticos se puede explorar nuevos yacimientos de petróleo, minerales o buscar la mejor orografía para canales. Cables, conducciones, etc.

- Alerta de catástrofes naturales. Gracias a los sensores se puede detectar terremotos submarinos que causan tsunami y prevenir las catástrofes. También se puede utilizar esos datos para estudio de los terremotos.

- Asistencia a la navegación. Los sensores pueden ser utilizados para detectar los peligros en el fondo oceánico. Peligros como rocas, arrecifes, pecios o algas.

- Vigilancia táctica distribuida. Esto incluye vigilancia, reconocimiento, detección de intrusiones y selección de objetivo en un área. Estos mecanismos suele ir acompañados de una AUV. Este método son más efectivos que los tradicionales radares y con una mejor clasificación de la amenaza utilizando diferentes sensores.

Todos ello indica el interés que despierta esta tecnología y es en la actualidad cuando se está implementando esta aproximación para responder a estas necesidades gracias al abaratamiento del coste de los nodos necesarios para la implantación de la red de sensores. Esto es importante debido a que el número de sensores necesarios en la red es alto y un coste alto de cada elemento haría económicamente inviable su implantación.

Como hemos visto, la red de sensores subacuáticos es una tecnología emergente y con un gran abanico de aplicaciones prácticas, lo cual hace de la red de sensores subacuáticos el terreno ideal para investigar y explica el creciente número de publicaciones sobre el tema. Por ese motivo este Trabajo de Fin de Grado versa sobre el diseño e implementación de un protocolo de red de sensores subacuático y más concretamente de un protocolo propuesto en el artículo [1].

Debido al alto coste de una implementación real de una red de sensores subacuáticos el estudio se realiza mediante simulaciones. Esto nos permite implementar y evaluar el protocolo de una forma asequible.

Cabe mencionar que este trabajo se inició durante la realización de una beca de colaboración con el Centro de Investigación ITACA (UPV).

1.2. Objetivos del proyecto

Los objetivos de este proyecto son:

- Estudio del campo red de sensores subacuáticos.
- Posibles aplicaciones de la tecnología.
- Problemática de la red de sensores subacuáticos.
- Estudio de la herramienta de simulación NS3.
- Estudio del protocolo AURP propuesto en el artículo [1].
- Replicación de lo resultado del artículo.
- Modelado del protocolo.
- Diseño e implementación de un modelo del protocolo AURP.
- Testeo sistemático del modelo de simulación.
- Validación del modelo mediante test y observaciones.
- Evaluación del protocolo AURP mediante simulación.
- Posibles mejoras respecto al protocolo AURP.

1.3. Estructura de la memoria

A continuación se pondrá en contexto el artículo a evaluar exponiendo los antecedentes previos a la red de sensores, problemática intrínseca que tiene y qué protocolos son los más destacados para finalizar con una breve explicación del protocolo a implementar. Posteriormente se explicará la herramienta de simulación seleccionada y la motivación de dicha selección, para después proceder a una explicación detallada del protocolo y del diseño elegido para su implementación. Una vez explicada la estructura pasaremos a ver qué procesos de validación se han seguido para saber que la implementación del protocolo es correcta. Al saber que el código está bien implementado podemos pasar a evaluar el protocolo y compararlo con otros protocolos ya descritos anteriormente. Finalmente se identificarán algunas líneas futuras de trabajo antes de exponer detalladamente las conclusiones del proyecto.

2. Estado del arte

2.1. Antecedentes

Previamente a la tecnología de red de sensores subacuáticos se utilizaban sensores que recopilaban información durante toda la misión y al terminar salían a la superficie. Al salir a la superficie emitían una señal vía aérea y el sensor se recogía. Pero este tipo de tecnología tenía ciertas desventajas como explica [4]:

- Imposibilidad de monitorización a tiempo real. La recopilación de los datos para su estudio no era posible hasta el final de la misión por lo que era necesario esperar meses para obtener los datos. Por lo que una monitorización a tiempo real era imposible. Esto hace que necesidades como la monitorización de seísmos o la detección de intrusos no sea posible.

- Imposibilidad de configuración online. Del mismo modo que no se podía obtener los datos, no se podía enviar nada al sensor para re-configurar o modificar su configuración. Los sensores estaban aislados. Esto genera menos flexibilidad a la hora de utilizar la tecnología.

- Detección de fallos. Como en los casos anteriores, al estar aislado no se puede saber si la misión está fracasando o que errores se están produciendo.

- Limitación de almacenamiento de datos. Debido a que tienes que almacenar todos los datos obtenidos hay un límite de datos a recolectar dado por la capacidad máxima del dispositivo.

Además de un uso distinto de sensores también ha cambiado el medio de comunicar la información. En la segunda guerra mundial ya se intentó comunicaciones subacuáticas pero utilizando radio frecuencia, que es lo que funciona bien en el aire. El problema de este método es que no funciona bien en medios acuáticos por motivos que explicaremos en el siguiente apartado. Por lo que se necesitaban grandes intensidades para transmitir.

Estas son las limitaciones que tenían estas tecnologías y con esto se entiende mejor la necesidad de migrar a otras tecnologías como es la red de sensores subacuáticos.

2.2. Características del medio

El medio subacuático es muy distinto al aéreo debido a sus características especiales. Para empezar, es necesario mencionar que el medio impide la utilización de cualquier mecanismo de trasmisión de la información que no sea el acústico puesto que los otros mecanismos se ven más afectados por las perturbaciones y su alcance y ancho de banda los hacen inviables. En [5] explica detenidamente las características que posee el medio y que pueden resumirse a continuación:

- Velocidad de propagación del medio. La velocidad del medio viene determinada por la densidad del medio y la presión. La salinidad del agua cambia la densidad y la

profundidad la presión. La velocidad en el agua es alrededor de 1500 m/s, superior a la del aire pero menor que la radiofrecuencia.

- Perturbaciones acústicas. Las señales acústicas tienen una gran cantidad de perturbaciones en el mar debido procesos físicos como son las olas, mareas, corrientes, cambios de presión, etc.
- Baja intensidad. La energía necesaria para trasmitir a grandes distancias es mayor que en el aire. Se necesita más energía para trasmitir y esto es muy importante puesto que los sensores tienen una batería limitada y se intenta maximizar su vida útil.
- Alta atenuación. Este punto va relacionado con el anterior. Al tener una alta atenuación en el medio, la energía necesaria para trasmitir más lejos es mayor.
- Múltiples caminos. Un mensaje puede llegar por diferentes caminos gracias a la reflexión de la onda en la superficie del agua y en el fondo marino.

Estos son los principales problemas y características del medio a los que los protocolos submarinos se tienen que adaptar.

2.3 Protocolos

En este apartado explicaremos las diferentes clasificaciones que hay de protocolos. Para ello nos apoyaremos en [6]. Como es habitual hay diferentes métodos de clasificación. Empecemos por la clasificación según su arquitectura:

- Clusters de nodos. Los sensores se organizan en agrupaciones que mandan toda su información a una puerta de enlace o nodo central. Luego ese nodo central es el encargado de enviar la información a la estación en la superficie.
- Enrutamiento entre sensores. La información se envía entre los propios sensores que van acercando la información y los nodos más cercanos envían la información a la estación. En este tipo de arquitecturas, los nodos no pueden estar alejados en profundidad de la estación. Respecto a esto hay diferentes métodos donde lo que se acerca es la estación en la superficie colocando una antena con un cable.
- Vehículos subacuáticos autónomos. La recolección de la información se realiza mediante vehículos subacuático autónomos como pueden ser submarinos. Posteriormente los submarinos envían la información a la estación en la superficie.

Otra posible clasificación es el modo en que se identifican las rutas de envío en cada uno de los nodos y que según [6] y [7] pueden ser:

- Proactivos, donde los nodos tiene una tabla de las rutas a enviar. El problema de este método es que las tablas tienen que estar actualizadas y para ello hay que enviar mensajes de actualización periódicamente.
- Reactivos, donde solo se calculan las rutas cuando son necesarias, cuando hay datos para enviar.

- Geográfico. En este tipo, los datos son enrutados basados en la posición geográfica hacia donde se quieren dirigir, de forma aproximada o con relación a los vecinos que tiene. Hay diferentes tipos, some-for-some, some-for-all, all-for-some and all-for-all. El problema de ese método es el hecho de que necesitan información sobre donde están colocados.

2.4. Protocolo AURP

El protocolo utilizado es AURP (AUV-Aided Underwater Routing Protocol) o protocolo de encaminamiento subacuático asistido por una AUV. La figura 2.1 explica el funcionamiento del mismo.

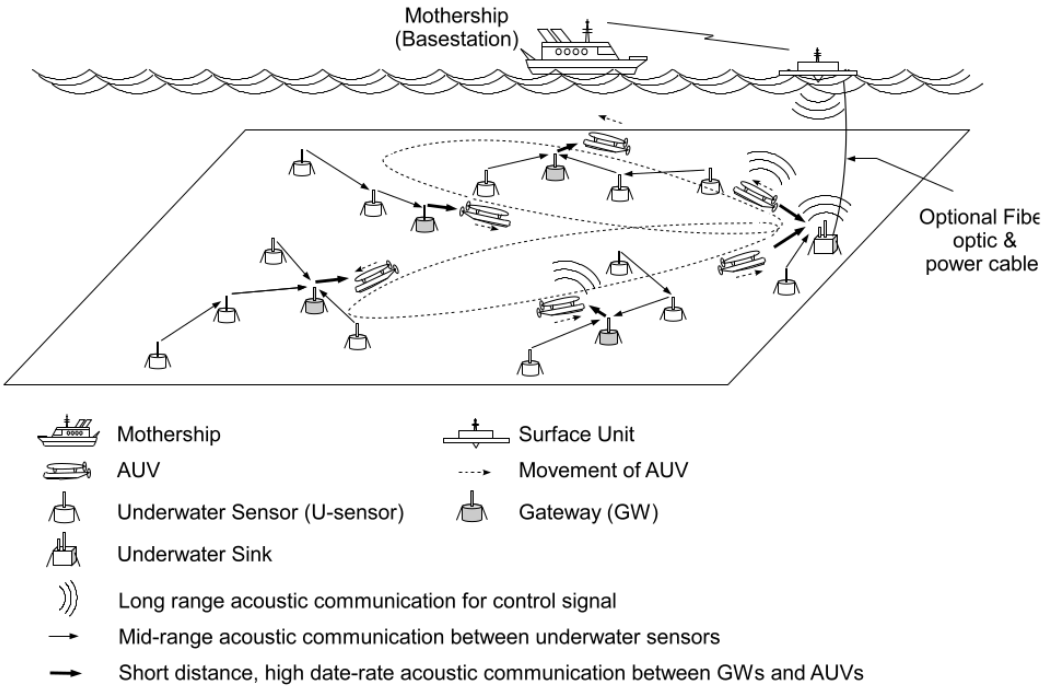


Figura 2.1: Esquema del funcionamiento de AURP [1]

Este protocolo puede clasificarse de diferentes formas. Por ejemplo, según la arquitectura tiene una estructura híbrida porque incluye diferentes elementos entre los que destacan los vehículos submarinos autónomos, utilizados para la recolección de los datos. Estos datos le son ofrecidos por unos nodos denominados puertas de enlace que centralizan la información de los sensores cercanos. Por este motivo, también se pueden clasificar como Clusters de nodos. Las AUV no envían la información directamente a la estación de la superficie sino que la envían a un sumidero sumergido que con medios como fibra óptica envían la información a la estación en superficie.

En cuanto a la forma de calcular las rutas de envío, ésta es proactiva puesto que envía periódicamente paquetes de configuración de la red para que los nodos actualicen su puerta de enlace.

Este protocolo ha recibido una gran atención de la comunidad científica, como indica que haya sido citado 35 veces en revistas indexadas por ISI.

3 Herramienta de simulación

3.1 Justificación del simulador utilizado

Un objetivo de este trabajo es la replicación del trabajo realizado en [1] para el cual utilizan un simulador NS. En consecuencia, lo más lógico es utilizar ese mismo tipo de simulador para evitar discrepancias imputables a la utilización de diferentes enfoques técnicos de la simulación.

Esto no implica que no se consideren otras opciones como son:

- Cisco Packet Tracer [8] es una herramienta muy potente y con mucha documentación aunque tiene un problema, es una herramienta propietario y no está diseñada específicamente para crear tus propios protocolos.
- GNS3 [9] es un aplicación potente y libre que sirve para la simulación de redes aunque la funcionalidad de desarrollo propio de protocolos no es muy potente.
- NetSim [10] sí que es utilizado a la hora de estudiar nuevos protocolos aunque es un sistema propietario.

Por otra parte, es un simulador ampliamente utilizado en el área de investigación de redes y gran parte de los artículos relacionados sobre el tema utilizan NS. La parte relacionada con redes subacuática tiene una gran cantidad de funcionalidades e incluso tiene modelos implementadas con AUV, así como la información que proporciona sobre la energía consumida por los nodos, la cual es muy necesaria y útil para nuestro estudio.

Por todos estos motivos hacen del NS3 el mejor simulador para el propósito perseguido.

3.2 NS-3

Como dice en su página principal [11] NS-3 es un Simulador de eventos discretos destinado principalmente a la investigación y uso educativo. Es software libre disponible para investigación, desarrollo y uso.

Está escrito en C++ y cuenta con una gran diversidad de módulos, que permiten modelar diversos aspectos interesantes a estudiar: la movilidad y distribución espacial de los nodos, su consumo energético y por supuesto, el de mayor importancia para nosotros: la capa de red física y MAC para redes subacuáticas.

La versión actual es una rama de la antigua versión NS-2, que estaba escrita en C y usaba un lenguaje de scripting basado en Tcl para definir los tests. En 2005 se empezó a desarrollar la nueva versión NS-3 desde cero y basada exclusivamente en C++, en la que se aprovechó para resolver algunas de las deficiencias de la versión anterior, especialmente las referidas al lenguaje de scripting y la falta de modularización. En 2008 se lanzó la primera versión de esta rama.

La versión utilizada para este proyecto es la 3.10 por ser una de las versiones más nuevas cuando se empezó el proyecto. Aunque había versiones más nuevas disponían de poca documentación por lo que se optó por un punto intermedio entre novedad y documentación.

3.3 Estructura y funcionamiento del NS-3

En NS-3 hay tres partes importantes como son el simulador, los módulos y la especificación de la simulación:

- Simulador. Es la parte central del simulador, el core. Es el encargado de gestionar los tiempos, gestionar los eventos, schedule, etc.
- Módulos. Son los encargados de definir los diferentes entornos y protocolos. Por poner un ejemplo, aquí es donde se define el comportamiento de los mensajes en el agua (con respecto a ciertas variables que le definas) o que es lo que tiene que hacer un sensor al recibir cierto tipo de paquete de datos en el protocolo AURP.
- Especificación de la simulación. Este apartado es el encargado de especificar cosas concretas de la simulación como son el número de nodos, la colocación geográfica, el ancho de banda se utilizará, los protocolos, etc.

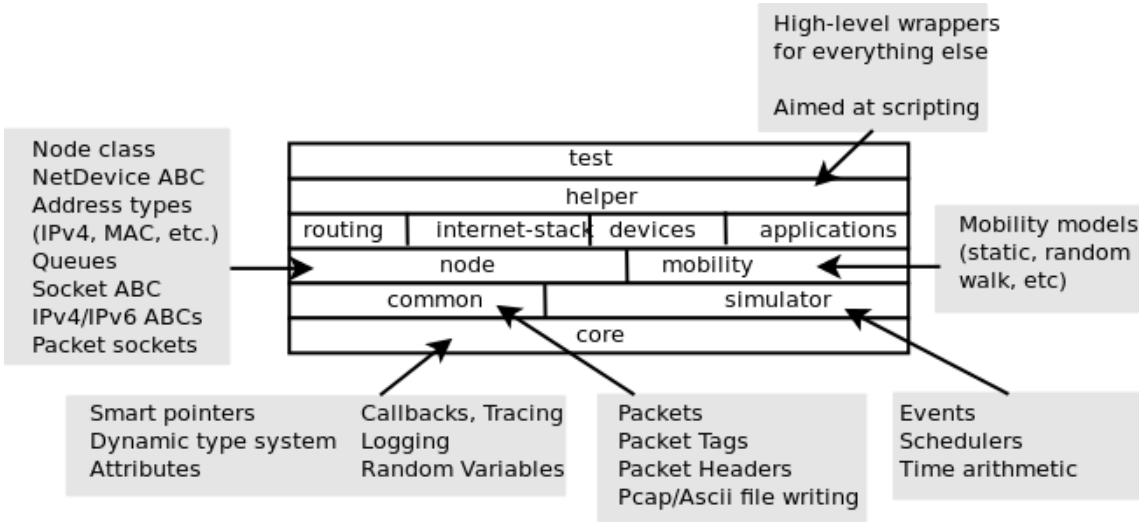


Figura 3.1: Estructura de los bloques básicos

La Figura 3.1 muestra los distintos módulos que conforman el núcleo del simulador. Aportan la funcionalidad que todo protocolo o entorno de simulación necesita. Esto es un esquema ampliado del apartado simulador. Además de esto, se le tiene que añadir los distintos modelos para los protocolos como son wifi o underwater, así como las especificaciones de distintos entornos de simulación.

4 Modelado del protocolo AURP

4.1 Especificación técnica del protocolo

En este apartado vamos a explicar con detalle el funcionamiento del protocolo descrito en [1] , Pare ello nos apoyaremos en la figura 4.1.

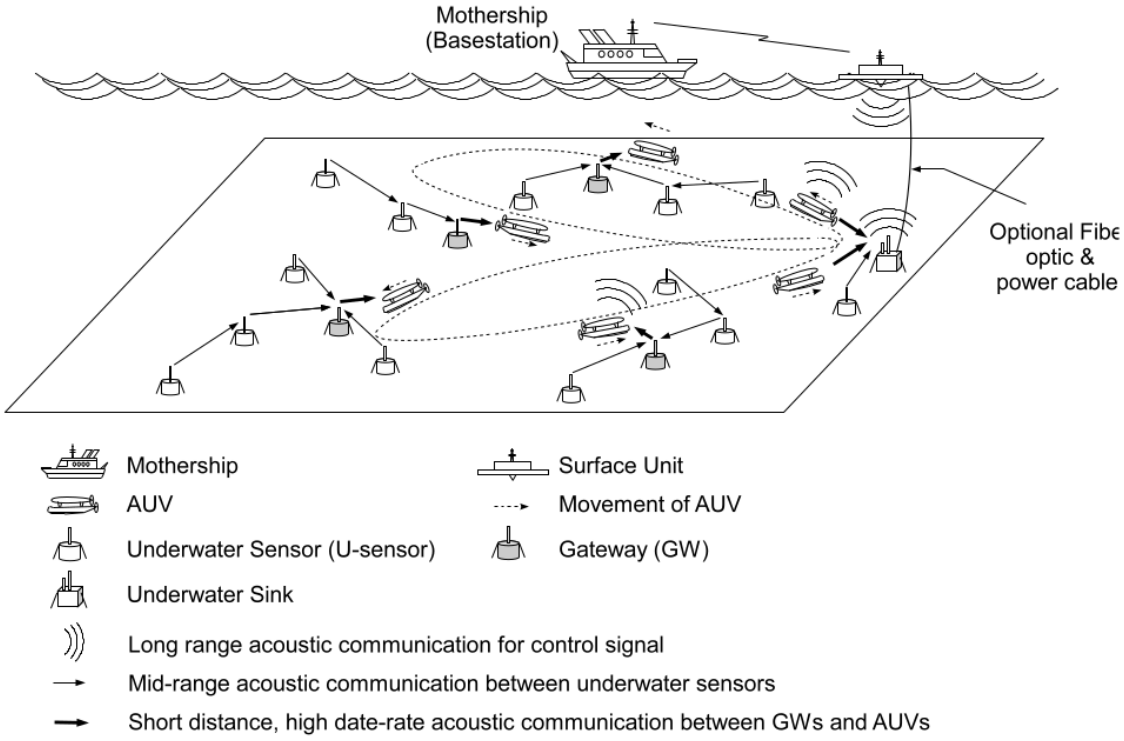


Figura 4.1: funcionamiento básico del AURP

El protocolo contempla 4 tipos de nodos en función de su funcionalidad:

- Sensores. Son los nodos que generan los datos y los envían a su puerta de enlace. También pueden hacer de puerta de enlace de otros sensores. Recogen la información de dichos sensores y la envían a su puerta de enlace.

- Puertas de enlace. Son los encargados de recolectar la información de los sensores cercanos, los que estén en su “cluster”. Cuando detectan que una AUV está cerca, negocian cómo va a ser la comunicación y le envía los datos si todo es correcto.

- AUV o submarino. Este nodo se mueve con un movimiento prefijado y va recolectando los datos de los nodos puerta de enlace. Cuando está cerca del sumidero envía toda la información al sumidero haciendo previamente una negociación.

- Sumidero. Es el nodo encargado de recoger la información. Este nodo puede obtener la información mediante la AUV o de forma directa cuando los sensores le envían directamente la información. En ese caso, el sumidero hace la función de puerta de enlace.

Una vez explicado los diferentes tipos de nodos, veamos cuáles son los caminos posibles que puede recorrer un mensaje para alcanzar el sumidero. El más sencillo es cuando el sensor envía la información directamente al sumidero. Esto solo es posible cuando el sensor está cerca del sumidero. Otro camino posible es cuando el sensor envía los datos a la puerta de enlace, de la puesta al submarino y del submarino al sumidero. Cabe mencionar que se puede añadir saltos entre sensores para estos dos tipos de comunicación. Esto quiere decir que los datos pueden hacer recorridos tales como sensor, sensor, sensor y sumidero o sensor, sensor, puerta de enlace, AUV y sumidero.

Como hemos visto anteriormente, la comunicación entre las puertas de enlace y el submarino se negocia, del mismo modo se hace entre el submarino y el sumidero. Vamos a ver más detenidamente cómo se negocia:

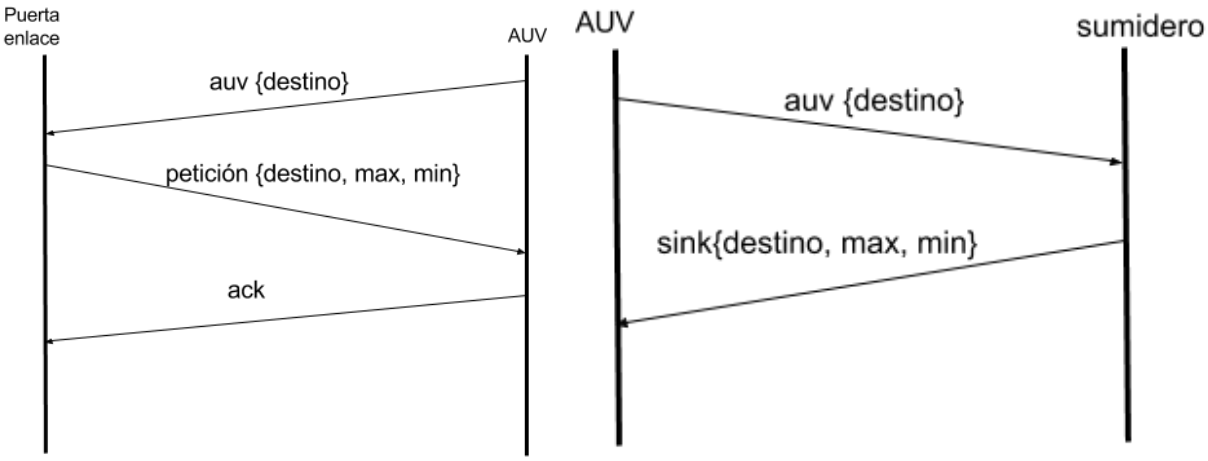


Figura 4.2: Comunicaciones con la AUV

Como se puede ver en la figura 4.2, la AUV va enviando mensajes de reconocimiento cada cierto tiempo, mensajes auv. Cuando pasa cerca de una puesta de enlace, la puerta de enlace le manda una petición de conexión con que frecuencias mínimas y máximas quiere utilizar. Si a la AUV le parece bien ese intervalo de frecuencias, le envía un mensaje de aceptación. En ese momento la puesta de enlace envía los paquetes de datos.

En el caso del sumidero, el sumidero es el encargado de enviar un mensaje de sink con el máximo y mínimo para que el submarino decida si enviarle los datos o no. Estos dos tipos de conexión se realizan a frecuencias mayores que el resto para que no interfieran y porque cubren una menor distancia a mayor velocidad. Aprovechan la proximidad entre ambos nodos.

Por ultimo cabe mencionar los mensajes de reconocimiento de red (PHE) que son utilizados para que los sensores escojan su puerta de enlace. Cada cierto tiempo las puertas de enlace y los sumideros envían un mensaje de reconocimiento PHE con numero de saltos=1. Cuando un sensor recibe un mensaje de este tipo, se espera un cierto tiempo por si llegan más mensajes PHE y pone todos los mensajes que han llegado en una lista. Cuando pasa ese tiempo escoge de la lista el mensaje con el mínimo número de saltos y actualiza su puerta de enlace a la dirección de origen de ese mensaje. Una vez hecho eso, envía un mensaje en broadcast de tipo PHE con su dirección e incrementando el salto en uno.

Esto garantiza que aunque la topología cambie los nodos puedan encontrar la mejor puerta de enlace. Además, gracias al reenvío los nodos más alejados pueden estar comunicados.

4.2 Diseño e implementación

El desarrollo del modelo de simulación se ha realizado con el sistema de espiral concéntrica donde con aproximaciones sucesivas se van desarrollando los diferentes partes del proyecto. Inicialmente se desarrolló un plan general donde se estipularon las diferentes fases del proyecto y los hitos a superar. Para pasar a la siguiente iteración del proyecto debía asegurarse que los hitos estuvieran cumplidos. Cada iteración consta de:

- Requisitos. Se especifican que requisitos u objetivos se deben cumplir en esa iteración del proyecto.
- Diseño. Se diseña como se cumplirán los requisitos. Qué estructura va a tener, cómo se va a implementar, qué cosas hay que añadir, etc.
- Implementación. A partir del diseño anterior, se procede a codificarlo e implementar el código.
- Pruebas. Una vez desarrollado el código de la nueva versión, se tiene que probar que cumple con los requerimientos establecidos en la primera fase. Además, esta fase es crucial puesto que hay que asegurar que el programa no tenga un comportamiento no deseado e inesperado que nos perjudique en las siguientes fases. Encontrar un error más tarde es más costoso de solventar.

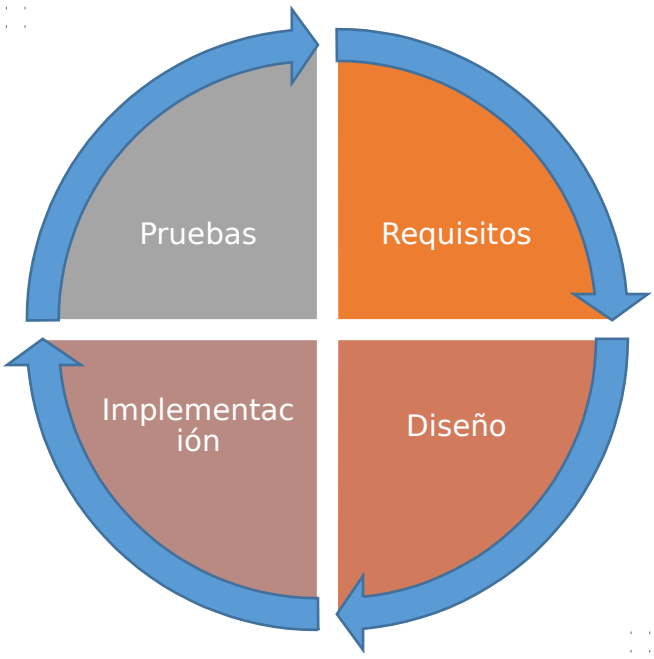


Figura 4.3: Ciclo de desarrollo.

Siguiendo este esquema, se han ido desarrollando las diferentes fases del proyecto que son las descritas a continuación:

- Búsqueda de trabajos relacionados. Al ser NS-3 un proyecto open source ampliamente utilizado en la investigación de redes subacuática es razonable pensar que otros equipos de investigación han realizado trabajos parecidos y han implementado funcionalidad que nos ayude con el proyecto. En este caso, fue añadido el funcionamiento de la AUV (que no estaba en la versión estándar del NS-3).

- Simulación sencilla. Para entender mejor el funcionamiento, el primer paso es conseguir implementar una simulación de un entorno subacuática con dos nodos que emitan y reciban.
- Introducción de la AUV. Tras conseguir que el entorno funcione con nodos simples hay que introducir el submarino. En esta fase también se genera la trayectoria elíptica que se describe en el artículo [1]
- Generación reducida del modelo. En este punto se genera el modelo con ciertas funcionalidades sin implementar. Se implementa toda la funcionalidad menos las comunicaciones relacionadas con la AUV.
- Generación completa del modelo. Esta es la última fase de desarrollo y es donde se termina de implementar el modelo introduciendo las comunicaciones relacionadas con la AUV.

En cuanto a las decisiones de diseño e implementación, han primado la legibilidad, flexibilidad, eficiencia y estilo ordenados por su orden de importancia:

- La legibilidad y la flexibilidad son importantes en un modelo de simulación donde se necesita saber que está pasando en todo momento y realizar cambios continuos que no supongan un trastorno. Además de la facilidad para que otras personas entiendan el código.
- La eficiencia es importante aunque no es lo más importante en una simulación. El tiempo de maquina o de respuesta no es crítico. No es significativo esperar una hora o esperar cinco, es tiempo de máquina.
- El estilo se refiere a realizar un código que sea elegante. Esto es importante pero no es el pilar central. Un buen estilo también redunda en una buena legibilidad.

Una vez explicado las diferentes fases del proyecto se verá el resultado final y las partes que tiene. A modo orientativo del tamaño y complejidad del modelo desarrollado, se ha codificado más de 2700 líneas de código. Explicando tanto el modelo como el lanzador de la simulación.

4.2.1 Esquema del descriptor de la simulación

Como ya hemos visto, éste es el encargado de especificar los parámetros y lanzar la simulación. Para explicar su comportamiento empezaremos explicando las funciones que implementa.

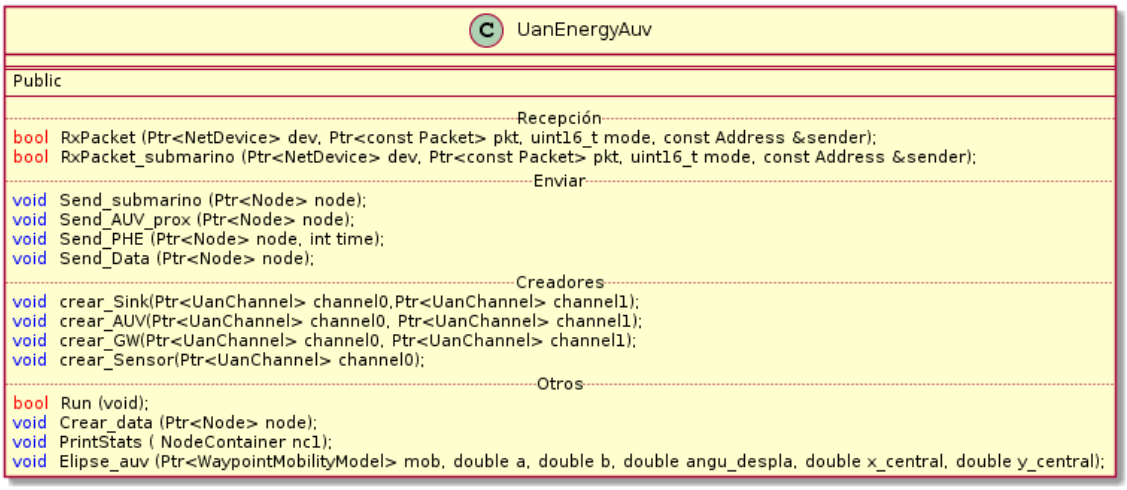


Figura 4.4: funciones del descriptor

Como se puede apreciar en la figura 4.4 las diferentes funciones se pueden agrupar en diversos tipos.

- Creadores. Son los encargados de crear los distintos tipos de nodos, como son el sumidero, la AUV, las puertas de enlace y los sensores. La estructura general de ambos es muy parecida pero con características especiales dependiendo de los nodos que generan. Por motivos de legibilidad y flexibilidad se ha optado por un diseño con diversas funciones de creación en vez de una única función.

- Enviar. Son los métodos encargados de enviar los diferentes tipos de mensajes de una forma periódica.

- Recepción. Son las funciones encargadas de gestionar las recepciones de los diferentes tipos de mensajes. Hay dos funciones porque hay dos canales de comunicación con sus respectivos tipos de mensajes: las comunicaciones relacionadas con el submarino y las comunicaciones no relacionadas con el submarino.

- Otros. Esta clasificación incluye funciones que por sí mismas no pueden formar grupo como son Run, que es el que lanza todo el programa; elipse, que genera el movimiento elíptico del submarino y Print, que es el encargado de mostrar los datos de la simulación.

Esto es una descripción de cuáles son las funciones que se implementan y utilizan en el descriptor de simulación. Ahora veamos el flujo del programa.

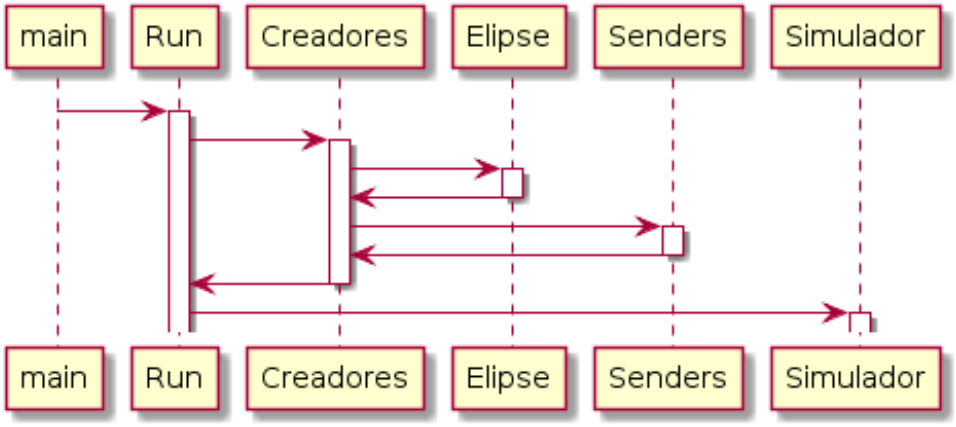


Figura 4.5: Flujo del descriptor

Como se puede observar en la figura 4.5 el flujo del programa empieza en el main el cual llama a Run, el cual llama a los distintos creadores de nodos (en el dibujo está simplificado aunando todos en una agrupación). Cada uno de los creadores llama a la función correspondiente de Sender. Adicionalmente, el creador de la AUV llama a la función que genera el movimiento elíptico que realizará el submarino.

Al terminar los creadores se empieza la simulación puesto que todos los parámetros están especificados. Cabe mencionar que run espera a que termine la simulación para llamar a Print.

Con esto se explica el funcionamiento del descriptor. Pasemos ahora a describir como está diseñado el modelo.

4.2.2 Esquema del modelo

En el modelo es donde se implementa el protocolo a simular. En este caso hay dos partes fundamentales a describir. La primera parte, que podríamos denominar como lógica de negocio, dicta qué se hace en cada momento. La segunda parte es donde se describe cómo van a ser los mensajes del protocolo y sus cabeceras. Siguiendo esta separación se han creado dos archivos distintos, el primero relacionado con la lógica (AURPNode.h) y el segundo relacionado con los mensajes (AURPHeader.h).

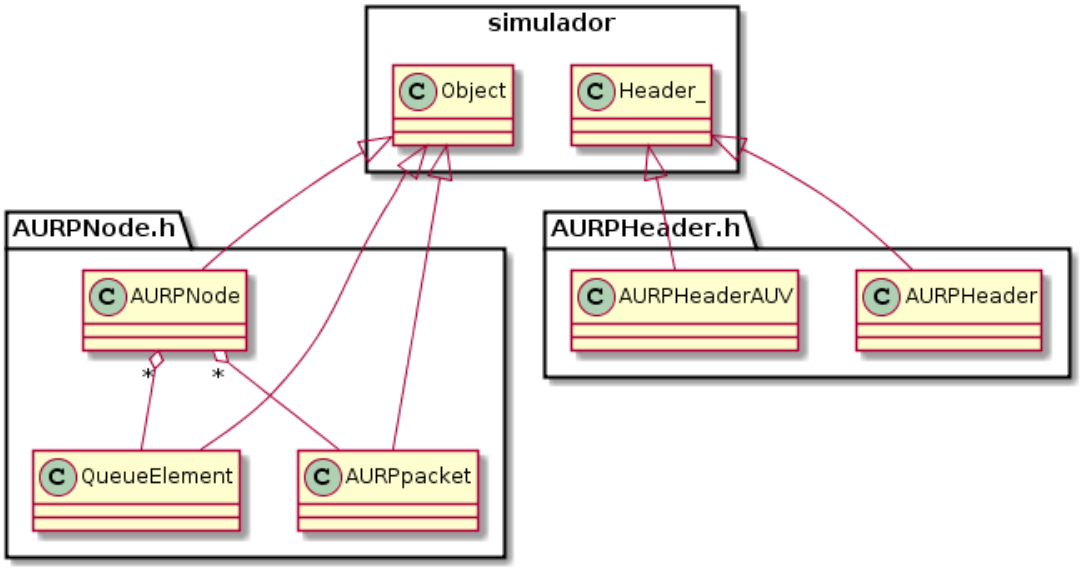


Figura 4.6: Esquema general del modelo

El esquema general del protocolo deja patente esa diferenciación en dos partes. Si se presta atención a AURPHeader se observan dos métodos que heredan de una clase genérica que proporciona el simulador. Cada una de las clases corresponde a diferentes tipos de mensajes y por lo tanto de comunicación: AURPHeader corresponde a las comunicaciones no asociadas con el submarino y AURPHeaderAUV son los tipos de mensajes relacionados con comunicaciones específicas con el submarino.

Por otra parte, está AURPNode el cual posee tres clases que todas ellas heredan de la clase genérica Object. La clase principal es AURPNode y las otras dos son clases creadas para dar más funcionalidad. Como se puede apreciar, dentro de AURPNode hay una lista con tantos elementos cono sean necesarios de QueueElement y AURPpacket.

Pasemos a explicar más detenidamente las funciones que poseen cada uno de ellos.

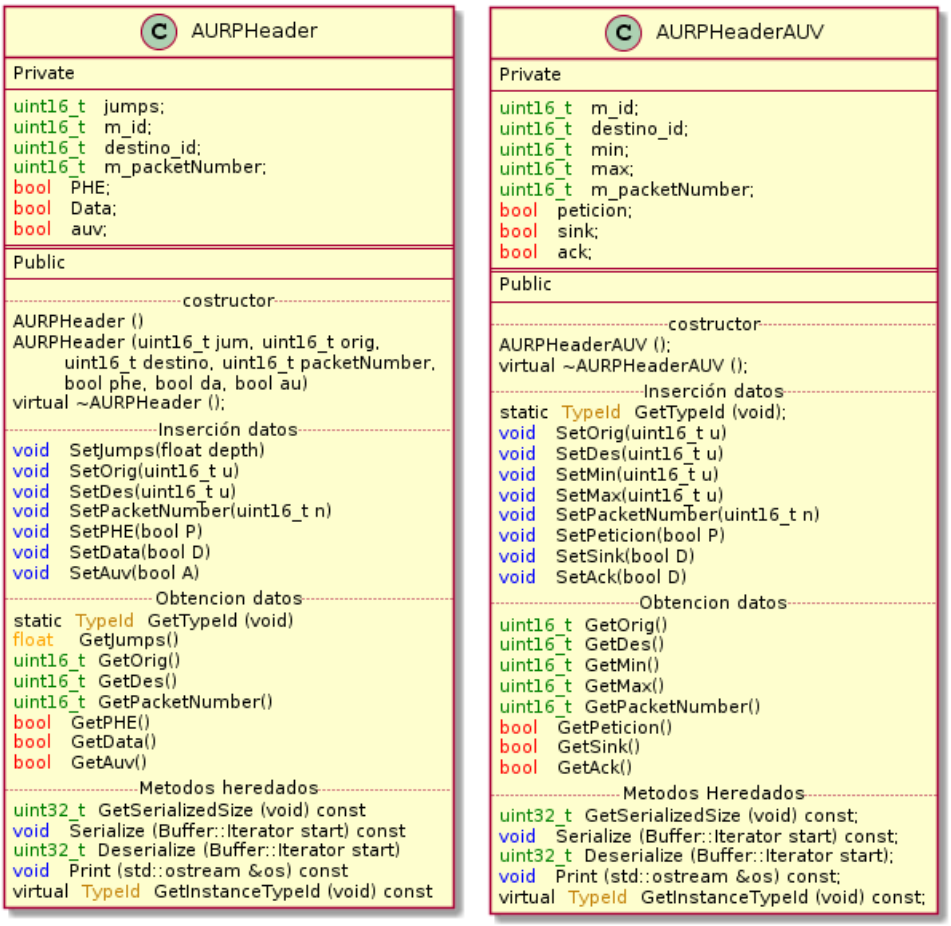


Figura 4.7: Estructura de los métodos cabecera

Ambas cabeceras poseen los campos origen, destino y número de paquete. Por otro lado, poseen otros atributos. AURPHeader posee además el número de saltos realizados, utilizado para los mensajes PHE. Además, posee flags que identifican los distintos tipos de mensajes, ya sean PHE, mensajes de datos o mensajes de proximidad de la AUV. Por otro lado están los atributos min y max de AURPHeaderAUV que son utilizados para establecer las frecuencias a las que se realizará la comunicación con el submarino. En el ámbito de los flags, existen los identificadores de petición por parte de la puerta de enlace, petición de comunicación por parte del sumidero y aceptación de la comunicación por parte del submarino.

Por otra parte están los métodos para las cabeceras, los cuales se pueden clasificar en cuatro tipos:

- Constructores. Son las funciones encargadas de crear las cabeceras.
- Inserción de datos. Son utilizadas para poder modificar los valores establecidos en las cabeceras.
- Obtención de datos. Son los métodos encargados para poder visualizar las diferentes variables.
- Métodos heredados. Son métodos que sirven para tratar adecuadamente la cabecera. Utilizados para cosas como extraer e introducir cabeceras en un mensaje o pintar en la traza los diferentes mensajes y variables que contienen.

Una vez explicadas las funciones de las cabeceras se explicará la de AURPNode.

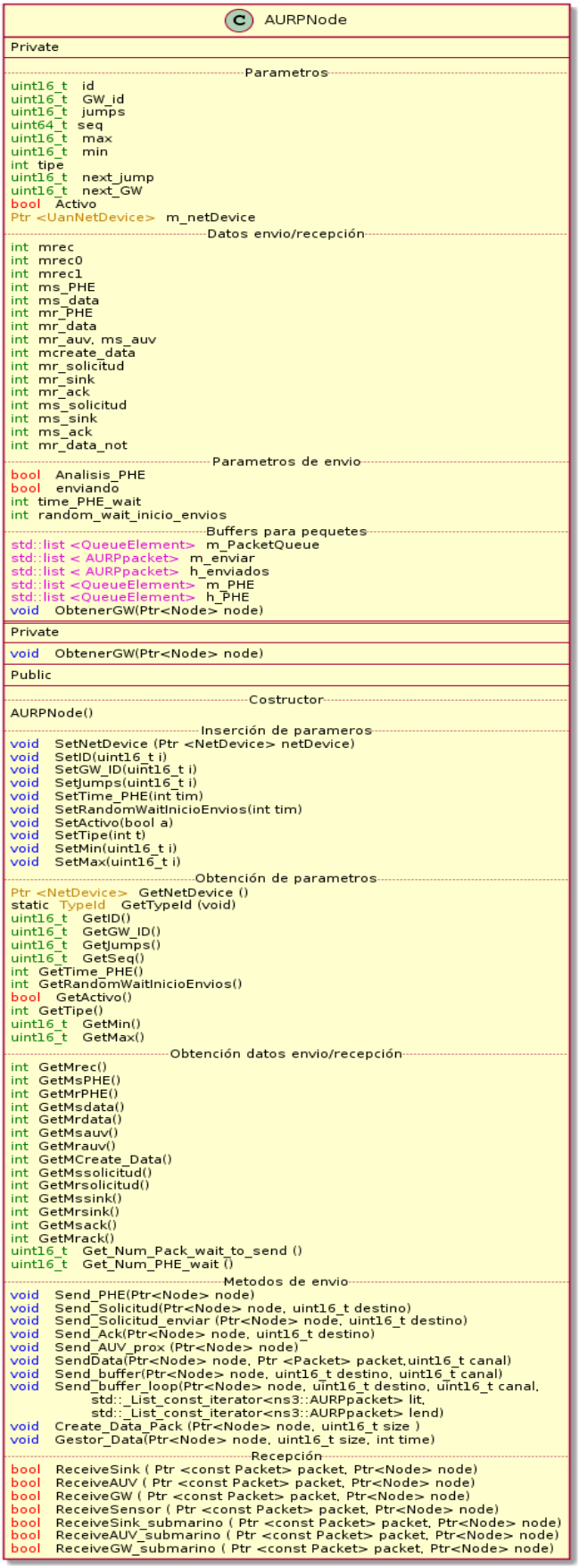


Figura 4.8: Estructura de AURPNode

Este bloque posee diferentes variables que se pueden clasificar en:

- Parámetros del nodo. Son variables que identifican al nodo o necesarias para su funcionamiento o lógica de negocio, tales como id, puerta de enlace, número de secuencia o packetnumber(seq), etc.
- Datos de envío y recepción. Son variables que contabilizan el número de eventos de diferentes tipos, todos ellos relacionados con envío y recepción de datos. Tales como números de paquetes recibidos, número de paquetes generados, número de paquetes de solicitud, datos, PHE, ack, etc se han recibido.
- Parámetros de envío. Variables relacionadas con procesos muy concretos de envío como pueden ser el tiempo de espera para procesar la cola de paquetes PHE desde que se recibe el primer mensaje PHE o variables semáforo.
- Buffers para paquetes. Lista donde se almacenan los distintos paquetes ya sean paquetes para enviar, paquetes ya enviados, cola de paquetes PHE o paquetes PHE ya procesados.

Una vez explicado las variables, el siguiente apartado son las funciones y diferentes tipos:

- Constructor. Es el método para construir el objeto y definir los valores por defecto.
- Inserción y obtención de parámetros. Funciones encargadas de modificar o entregar el dato solicitado.
- Obtención datos de envío y recepción. Muestran los datos de la simulación relacionados con las variables descritas anteriormente, las cuales tienen que ver con envío y recepción de paquetes.
- Métodos de envío. Métodos para enviar los diferentes tipos de datos. Hay que mencionar algunas funciones especiales como el gestor de datos que gestiona la creación y el envío de mensajes y se puede programar para que se ejecute cada cierto tiempo. Además Send_buffer_loop es una función que envía de forma recursiva todos los paquetes que estén en el buffer de envío. Al ser recursivo, necesita un lanzador Send_buffer.
- Recepción. Son las funciones encargadas de gestionar qué hacer con los paquetes que se reciben. Hay una función por cada tipo de nodo y por los dos tipos de comunicación.

Como se puede observar en la figura 4.8 las listas que posee AURPNode son de objetos QueueElement y AURPpacket los cuales son parte del modelo y se muestran a continuación.

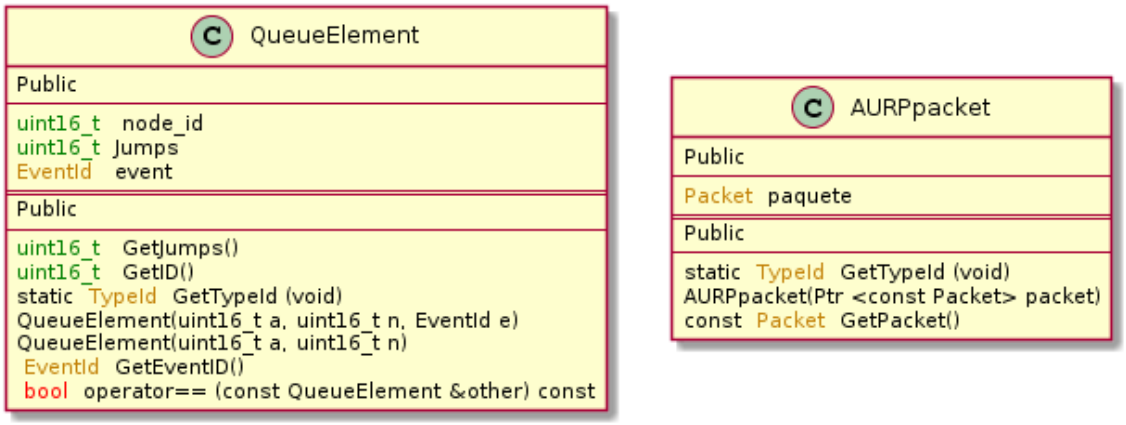


Figura 4.9: Esquema de QueueElement y AURPpacket

Como muestran las figuras, los dos métodos son simples y poseen pocas variables y funcionalidad. QueueElement es utilizada para almacenar la información relevante de los paquetes PHE y por tanto tiene la id del emisor y el número de saltos, con sus correspondientes funciones para obtener los datos y los constructores. Por su parte, el AURPpacket solo contiene paquetes porque es utilizado para almacenar los paquetes que llegan y colocarlos en la lista de salida.

5 Pruebas y validación

5.1 Pruebas realizadas

Como se ha visto en apartados anteriores, el desarrollo del proyecto se ha realizado con aproximaciones concéntricas. En una de las fases se realiza la comprobación del buen funcionamiento y si cumple con los requisitos marcados, por lo que se describirá en cada fase las pruebas realizadas.

En la simulación sencilla la única prueba realizada es la comprobación de emisión y recepción correcta. Cuando se introduce la AUV con el movimiento elíptico es necesario comprobar que describe el movimiento correctamente y que los diferentes nodos son capaces de verse. Pero es cuando se entra en la fase de creación del modelo cuando es necesario hacer una batería de pruebas adecuada.

Para afrontar las pruebas en las dos fases de diseño del modelo se ha optado por realizar pruebas de caja blanca y caja negra. Las pruebas de caja blanca se han utilizado para asegurar que las líneas de código sean correctas. Por otra parte, las técnicas de caja negra comprueban que los posibles escenarios están considerados y su comportamiento es el esperado.

Las pruebas realizadas para la fase de generación reducida del modelo se presentan en la Tabla 5.1 donde se enumera las pruebas realizadas por cada una de las combinaciones de nodos.

	Puerta de enlace	Sumidero	Sensor
Puerta de enlace	-Una puerta de enlace envía mensajes PHE. -Una puerta de enlace envía mensaje de datos con id distinta. -Una puerta de enlace envía mensaje de datos con id correcta.	-Puerta de enlace envía mensaje PHE. -Sumidero envía mensaje PHE. -Puerta de enlace envía mensaje de datos con id distinta. -Puerta de enlace envía mensaje de datos con id correcta.	-Puerta de enlace envía mensaje PHE. -Sensor envía mensaje PHE. -Sensor envía mensaje de datos con id distinta. -Sensor obtiene la id a trabes de los mensajes PHE y envía mensaje de datos. -Puerta de enlace envía mensaje de datos.
Sumidero	PRUEBAS YA GENERADAS	-Un sumidero envía mensaje de PHE.	-Sumidero envía mensaje PHE. -Sensor envía mensaje PHE.

			-Sensor envía mensaje de datos con id distinta. -Sensor obtiene la id a trabes de los mensajes PHE y envía mensaje de datos.
Sensor	PRUEBAS YA GENERADAS	PRUEBAS YA GENERADAS	-Un sensor envía mensaje PHE. -Un sensor envía mensaje de datos con id distinta. -Un sensor obtiene la id atabes de los mensajes PHE y envía mensaje de datos.

Tabla 5.1: Enumeración de pruebas realizadas.

Puede observarse que el número de pruebas realizadas es de 20. Como se ha mencionado con anterioridad, todas estas pruebas son necesarias para abarcar todos los posibles escenarios conforme a la técnica de caja negra. Las pruebas correspondientes a caja blanca son las marcadas en verde, como es lógico también sirven para caja negra.

Además de esas pruebas también se hicieron pruebas de integración con una puerta de enlace y cinco sensores donde la puerta de enlace envía los paquetes PHE y los sensores tratan de llevar los mensajes de datos a la puerta de enlace.

Estas son las pruebas realizadas para la fase de generación reducida del modelo pero en la siguiente fase también hay que realizar pruebas. En la fase de Generación completa del modelo tenemos nuevas pruebas, las cuales se pueden separar en dos tipos: las relacionadas con los tipos de mensajes anteriores que involucran a la AUV y las relacionadas con los nuevos tipos de mensajes. La tabla 5.2 enumera las pruebas realizadas.

	Sumidero	AUV	Puerta de enlace	Sensor
Mensajes anteriores con AUV	-Sumidero envía un mensaje PHE.	-Una AUV envía mensajes de AUV cercana.	-Puerta de enlace envía mensajes PHE.	-Sensor envía mensajes PHE. -Sensor envía mensajes de datos con id distinta. -Sensor envía mensajes de datos con su id.

Mensajes nuevos con la AUV	-Comunicación completa de conexión. -Sumidero intenta conectar con otra AUV.		-Comunicación completa de conexión. -La puerta de enlace intenta conectar con otra AUV.	- La AUV envía mensajes de cercanía.

Tabla 5.2: Pruebas realizadas para Generación completa del modelo.

Como se ve en a tabla 5.2 con estos experimentos se cubre toda la casuística posible. También cabe remarcar que estos experimentos son más complejos que los anteriores por dos motivos. Primero los tipos de comunicaciones son más complejas y segundo porque se hace una prueba completa de dicha comunicación y no mensaje por mensaje.

5.2 Validación

Una vez comprobado que todos los componentes se comportan adecuadamente en los diferentes escenarios se realizan las pruebas de integración para validar que la implementación del modelo está bien realizada.

Para validar el modelo se ha realizado dos escenarios, un primer escenario con un nodo de cada tipo separados 100 metros y un segundo escenario con un nodo de cada tipo y la AUV realizando un movimiento elíptico. Una vez estas pruebas funcionan adecuadamente es momento de realizar un escenario base replicando alguno de los escenarios que aparecen en el artículo.

5.2.1 Escenario base

Probablemente a causa de la complejidad de este sistema de comunicaciones y del elevado número de parámetros involucrados, los autores no explicitan los valores adoptados para todos los parámetros en beneficio de la legibilidad del artículo. Por ello que a la hora de reproducir fielmente los escenarios se ha elegido valores que sigan la línea del artículo.

El escenario que se ha utilizado para la validación es el más simple posible donde hay un sumidero, un submarino, una puerta de enlace y 20 nodos. Cada 20 segundos la AUV envía un mensaje de posición, la puerta de enlace envía mensajes PHE cada 150s, el sumidero envía PHE cada 60s y los sensores generan un mensaje de datos cada 30s. El tiempo total de la simulación es que tiempo que tarda el submarino en realizar 100 vueltas a 19km/h. En el caso base esto implica un total de 77824segundos. La disposición geográfica se especifica en la figura 5.1.

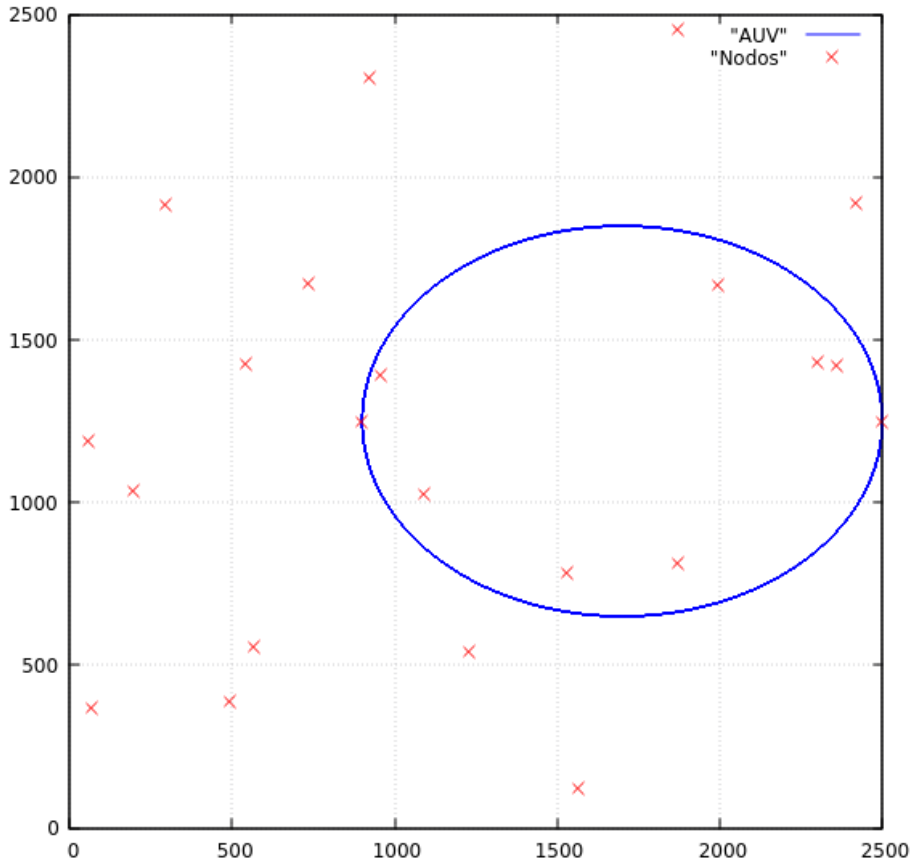


Figura 5.1: Posición geográfica escenario base.

La línea azul representa el movimiento del submarino y los puntos rojos son los diferentes nodos. El nodo situado en (2500, 1250) es el sumidero y el situado en el lado opuesto de la elipse es la puerta de enlace. El resto de nodos (sensores) están repartidos aleatoriamente por la zona especificada (un cuadrado de 2500m de lado).

Tras parametrizar utilizando los datos proporcionados por el artículo, se comprueba si los datos de salida proporcionados por la simulación concuerdan con los presentados en el artículo. El artículo especifica que para este escenario el ratio de llegada es 0,5 aproximadamente mientras que la validación obtiene que un ratio de 0,54 que se considera razonablemente similar. También se obtiene la energía consumida por nodo pero no es posible comparar los valores porque en el artículo no se especifica en qué rango de tiempo se obtiene. Sin embargo, eso datos serán útiles para observar el patrón de consumo ante diferentes factores.

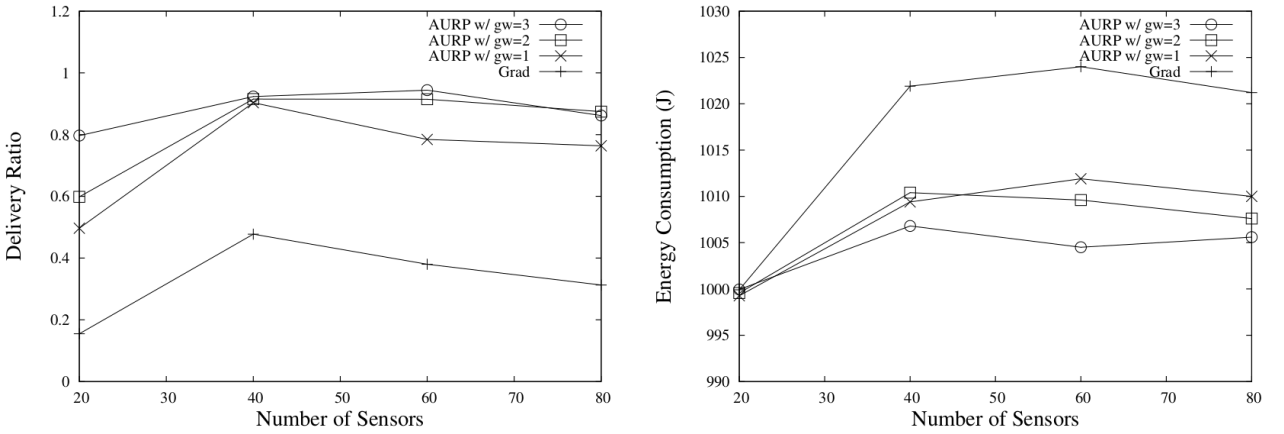
6 Estudio del artículo

6.1 Resultados obtenidos y comparativos con el artículo

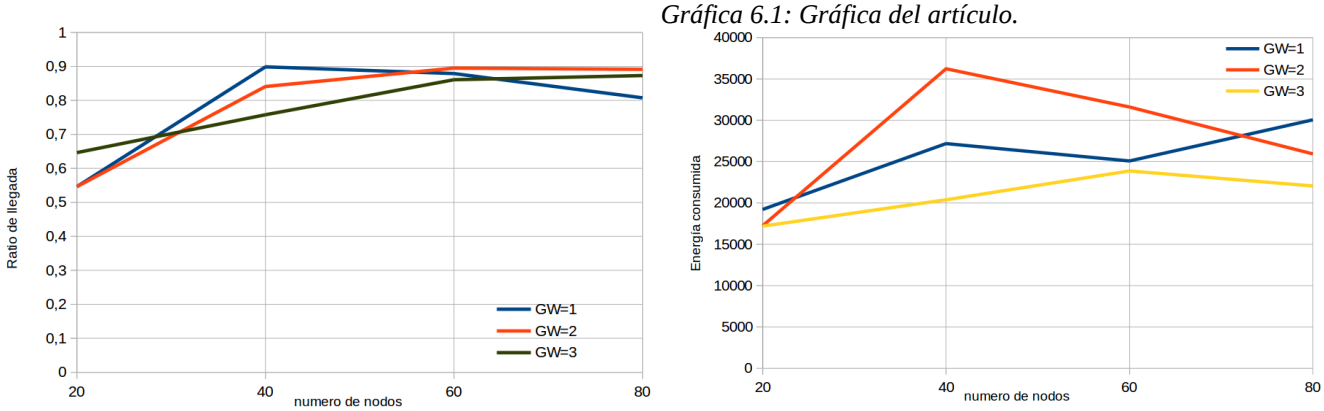
Una vez comprobado el buen funcionamiento de la implementación del protocolo, es momento de comparar si los resultados del artículo son parecidos a los obtenidos. Se empezará estudiando la primera gráfica del artículo donde se plantea el mismo escenario que el base pero aumentando el número de nodos.

6.1.1 Numero de nodos

El número de nodos contemplado es 20, 40, 60 y 80. El resultado obtenido en el artículo está en la figura 6.1 y el obtenido en el trabajo en la figura 6.2.

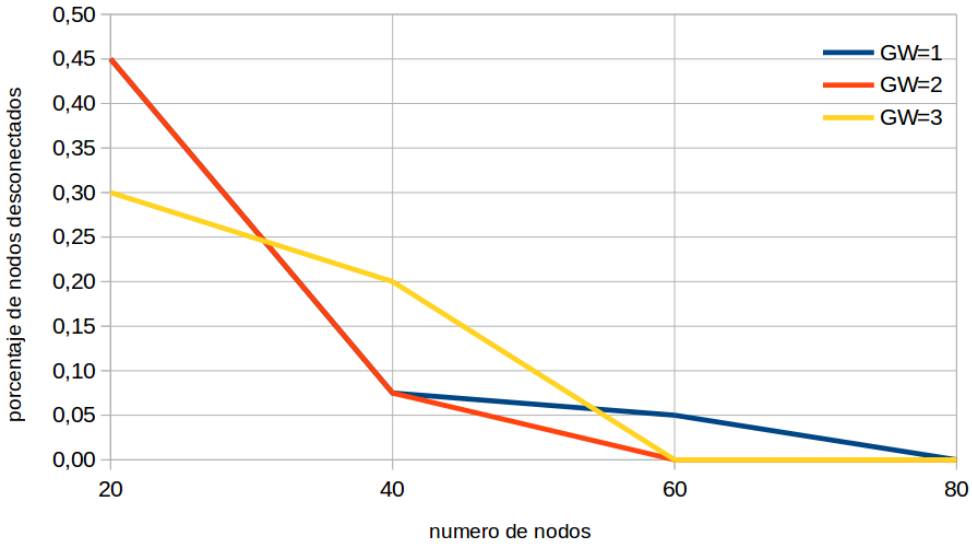


Gráfica 6.1: Gráfica del artículo.



Gráfica 6.2: Gráficas de los datos obtenidos

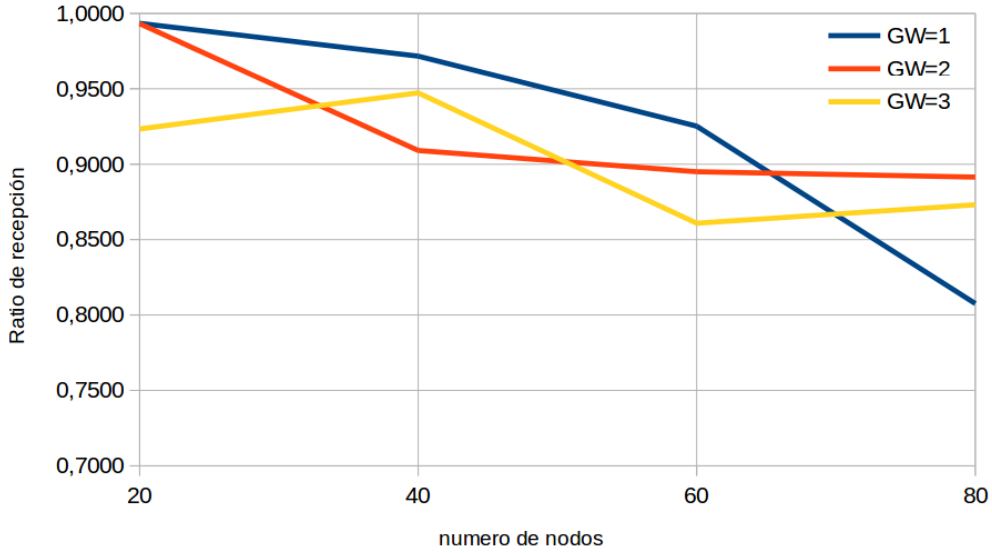
Como se puede observar en las gráficas los resultados son bastante parecidos, las tendencias se repiten. En las dos gráficas se puede apreciar como con 20 nodos el ratio de entrega es mucho menor. Afirman en el artículo que esto es comprensible puesto que con tan pocos nodos existen nodos que se quedan incomunicados, pero los autores no proporcionan datos que lo avalen. Veamos en la Gráfica 6.3 el número de nodos desconectados de la red en porcentaje de nodos totales.



Gráfica 6.3: Porcentaje de nodos desconectados

Como se puede observar en la Gráfica 6.3 es cierto que el número de nodos desconectados de la red para 20 nodos es muy superior al resto de casos por lo que la afirmación anterior parece correcta.

Veamos pues el ratio de entrega en función del número de nodos conectados a la red y no en función del número total de nodos. Esto nos dará una mejor aproximación de cómo se comporta el protocolo en función del número de elementos involucrados en la comunicación.

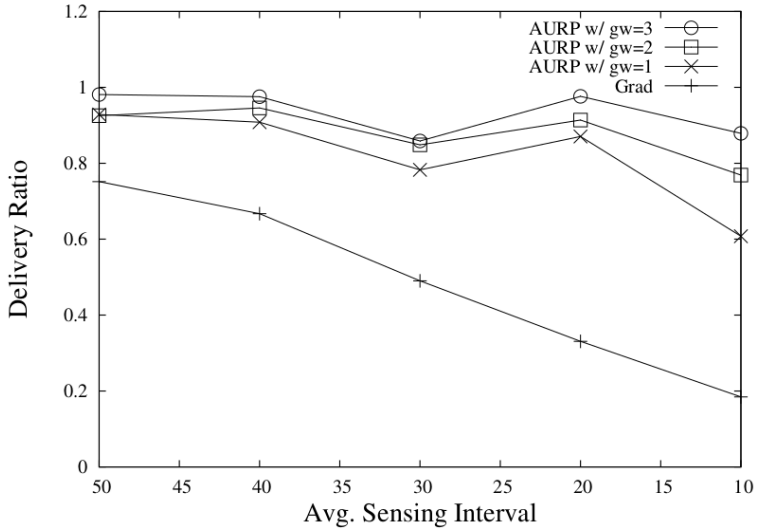


Gráfica 6.4: Ratio en función del nodos conectados a la red.

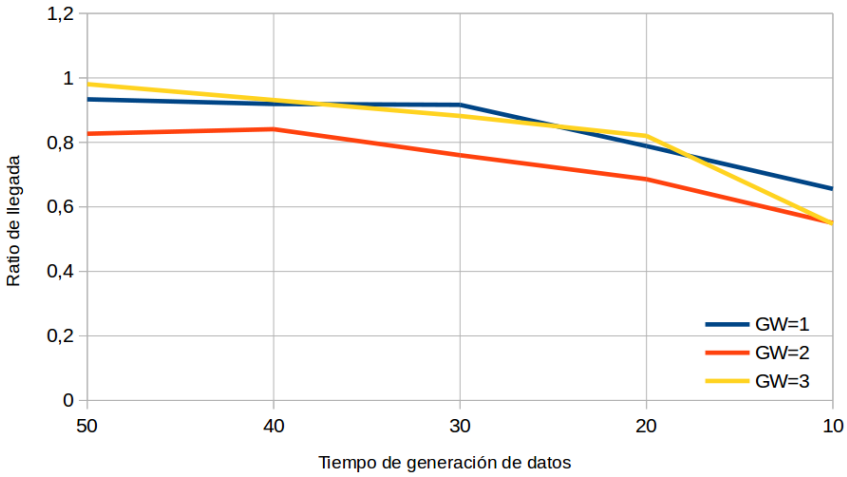
Como se observa, esa anomalía ha desaparecido y como es de esperar la tasa de entrega desciende en función del número de nodos. Además, no existe mucha diferencia entre utilizar dos o tres puertas de enlace para la comunicación pero si se utiliza una puerta de enlace y el número de elementos en la red es muy alto, la tasa de recepción desciende rápidamente. Esto es causado porque la comunicación de los sensores hacia la puerta de enlace empieza a estar colapsada. Esto se puede ver porque el porcentaje de mensajes que le llegan a la puerta de enlace con 60 nodos es del 66% y con 80 es del 55%. Además, en el caso de dos puertas de enlace el porcentaje de llegada a través de las puertas de enlace es del 80%.

6.1.2 Mensajes por segundo

Para saber cómo se comporta la red en función de la sobrecarga, el artículo realiza una serie de simulaciones donde se genera mensajes de datos cada 50s, 40s, 30s, 20s y 10s. Las gráficas 6.5 y 6.6 muestran los resultados de ambas ejecuciones con respecto al ratio de recepción.

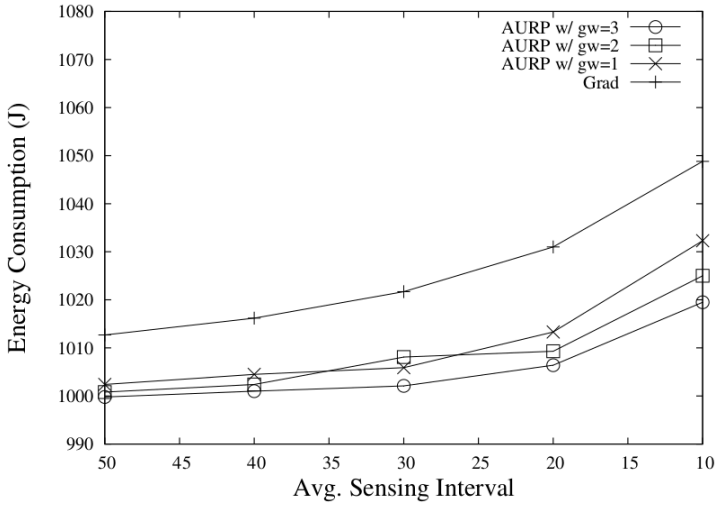


Gráfica 6.5: Datos obtenidos por el artículo.

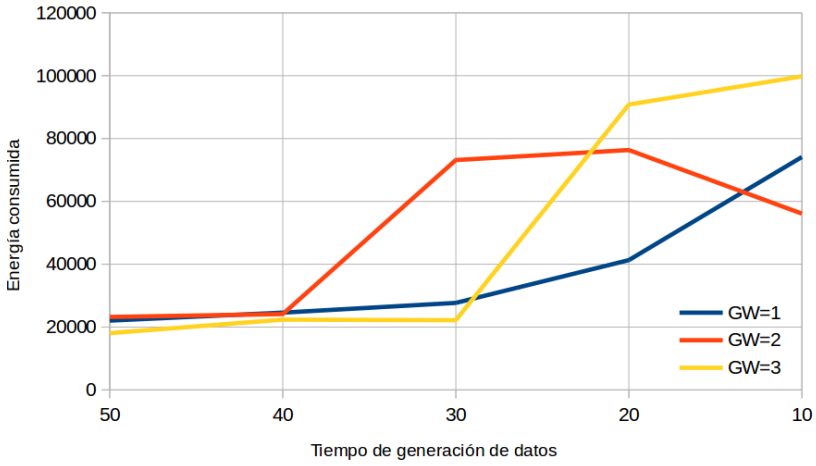


Gráfica 6.6: Tasa de llegada en la simulación.

Observando las dos gráficas se percibe que la tendencia en los dos caso es la misma pero con ligeros matices. En la del artículo se observa un mínimo local en 30 y un máximo en 20 que no aparecen en la Gráfica 6.6. Esto es significativo puesto que estos mínimos y máximos son un comportamiento anómalo que no está presente en la simulación realizada. A continuación se presentan los datos de consumo.



Gráfica 6.7: Consumo en el articulo.



Gráfica 6.8: Energía consumida por los nodos.

En este caso, los consumos del artículo tienen un comportamiento más cercano a lo esperado y los de la simulación tienen anomalías. Las gráficas para una puerta de enlace son idénticas en ambas gráficas, lo mismo ocurre con el principio de las gráficas para 2 y 3 puertas de enlace. Pero, ¿por qué suceden estas anomalías?, ¿por qué luego vuelve a descender el consumo para 2 puertas y 10 segundos?

Para responder a esas preguntas vamos a adentrarnos en alguna de las simulaciones para entender qué está sucediendo, por ejemplo 30 segundos y dos puertas de enlace. Si examinamos el consumo individualmente por cada nodo y no de forma global se aprecia que 10 nodos, el 20% de los nodos, consumen dos tercios de la energía en la simulación y los restantes 40 nodos consumen un tercio del total. Por ello, si esos 10 nodos tuvieran un consumo similar a los nodos restantes el consumo global de energía estaría en márgenes aceptables.

Analizando las posibles causas de este mayor consumo se observa que dichos nodos reciben y envían una cantidad 10 veces superior de mensajes. Se podría pensar que estos nodos reciben tanto volumen de datos porque están cerca de las puertas de enlace, pero si se representan las posiciones de los distintos nodos y la posición de estos nodos especiales se comprueba que no es así.

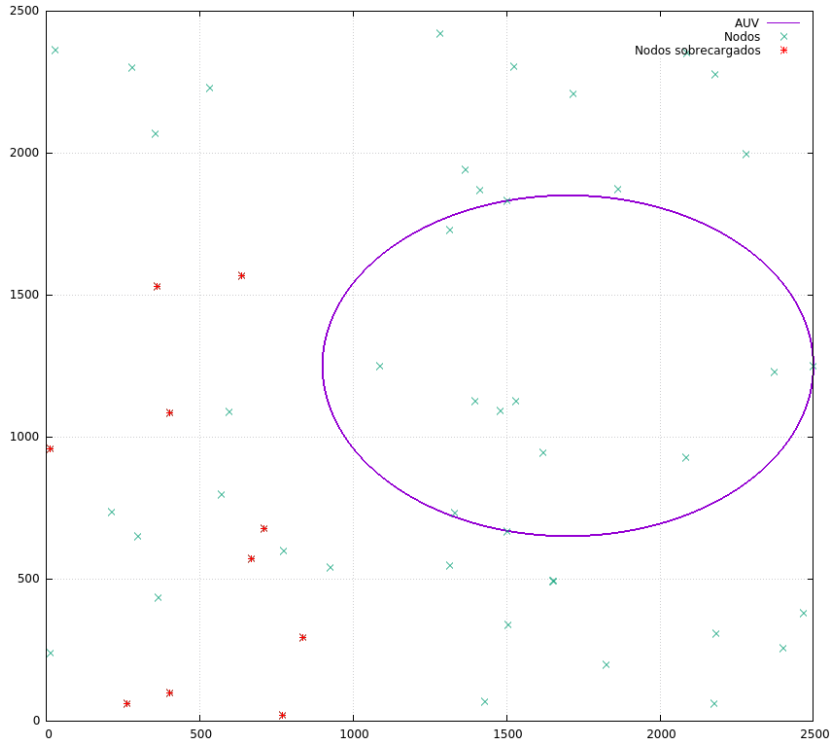


Figura 6.1: Localización de los nodos especiales

Como muestra la Figura 6.1, los nodos no están cerca de las puestas de enlace sino que están localizadas cercanos unos de otros y lejos de las puertas de enlace luego esa no es la causa.

Si se pone de nuevo la atención en el número de mensajes recibidos en un nodo con alto consumo, se obtiene que ha recibido 124.000 mensajes. Si cada nodo genera un total de 2600 mensajes y suponemos que los mensajes no se duplican, por ese nodo está pasando en promedio la información de 48 nodos y esto es imposible por la posición geográfica de dicho nodo.

Una forma de explicar este comportamiento es dudando de las suposiciones anteriores. Observando el número de mensajes generado por cada nodo se obtiene que todos los nodos generan 2600 mensajes con una variación de más menos un mensaje. Esto indica que la primera suposición es real. Si se duplican los mensajes los nodos enviarían más mensajes que la suma de los mensajes recibidos más los generados. Una vez repasados todos los nodos se observa que no es así. También es posible que el mensaje se duplique en vuelo, es decir, que lo reciban dos nodos y ambos lo traten como mensaje dedicado a ellos y por lo tanto ambos lo retransmitan. En ese caso se generarían dos mensajes iguales, se duplicarían.

Otra posible explicación reside en la generación de bucles a la hora de encaminar los paquetes, donde solo existe una ocurrencia de ese mensaje pero nunca llega al sumidero y se retransmite dentro de un bucle infinito. Si ocurre alguno de estos problemas un mensaje se enviaría más veces de lo normal. La Tabla 6.1 muestra el número de veces que ha sido enviado un mensaje tomando como identificador de paquete el origen y el número de paquete.

Identificador de paquete (origen, paquete)	Ocurrencias
16 40	386
39 41	386

8 40	386
36 40	385
45 42	383
37 42	382
36 42	381
45 43	381
5 43	381
13 43	380
39 44	380

Tabla 6.1: Paquetes más enviados en los primeros 7000s de la simulación.

La tabla indica que el problema reside en algunas de las causas citadas anteriormente puesto que un mensaje no tendría que ser enviado más de 10 veces. El siguiente paso es seguir la traza de un paquete para saber por dónde pasa. La Tabla 6.2 muestra una parte de la traza del paquete 16 40 donde las dos últimas filas se repiten indefinidamente.

tiempo	destino	Identificador de paquete (origen, paquete)
1212	10	16 40
1218,05	37	16 40
1234,15	46	16 40
1257,2	15	16 40
1273,25	46	16 40
1287,55	15	16 40
1303,6	46	16 40

Tabla 6.2: Recorrido del mensaje 16 40.

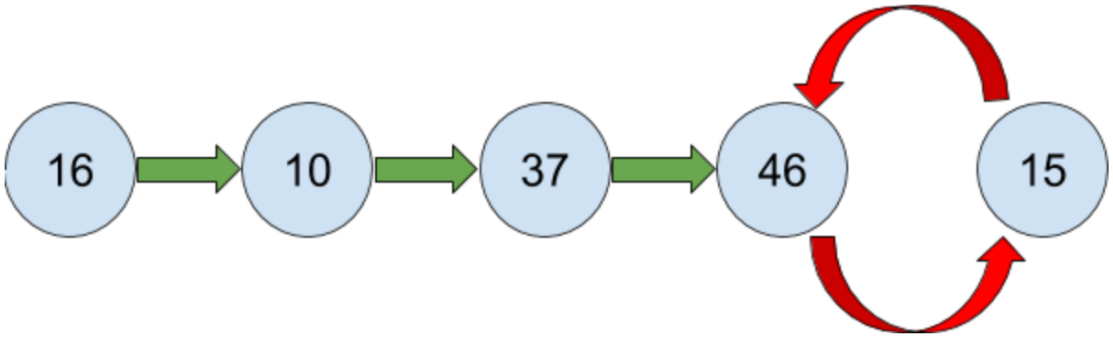


Figura 6.2: Flujo del mensaje 16 40.

Como se observa en la Tabla 6.2 y se representa en la Figura 6.2, el mensaje se ve envuelto en un bucle. Una vez que ya sabemos cuál es el problema, hay que averiguar qué lo causa y por qué permanece. Observemos todos los paquetes que recibe el nodo 15. La Tabla 6.3 refleja los últimos mensajes de reconocimiento que recibe. Previamente a la tabla el nodo 15 elegía como puerta de enlace el 9.

Tiempo	Saltos	Origen
928,129	5	54
928,207	5	10
1077,85	4	46
1077,92	4	45
1077,97	4	37
1078,03	5	9
1078,12	4	5
1078,13	5	29
1078,19	5	54
1078,28	5	10

Tabla 6.3: Mensajes PHE recibidos por el nodo 15.

Cuando el nodo 15 tiene que elegir en esta ocasión, el mensaje del nodo 9 tiene más saltos por lo que elige el nodo 46 como puerta de enlace. Tras la elección, no recibe ningún mensaje más de PHE. Esto es debido al bucle. Cuando se forma el bucle se crea una avalancha de paquetes que circulan indefinidamente por él e impide que reciban más mensajes. Toda la comunicación se ve acaparada por los mensajes de datos que se envían entre sí.

Este comportamiento es más dañino de lo que parece puesto que afecta al funcionamiento de todos los nodos que están en el radio de acción de esos nodos. Si nos fijamos en el nodo 9 (el cual hacía de puerta de enlace al 15) observaremos que tras la formación del bucle solo recibe dos mensajes más de PHE en los primeros 7000seg de simulación, tasa muy inferior a la que recibía antes de formarse el bucle. La Tabla 6.4 muestra los mensajes.

tiempo	Saltos	origen	PHE
1078.08	4	5	1
1078.1	4	37	1
5729.26	2	28	1
6929.51	2	51	1

Tabla 6.4: Mensajes PHE recibidos por el nodo 9.

Esto ejemplifica el hecho de que los bucles se forman y que no solo dejan ciegos a los nodos involucrados en el bucle sino también los que están en su radio de alcance. Este efecto genera una doble barrera que imposibilita que los bucles se rompan. Los nodos del bucle “envenenan sus abrevaderos” de PHE. La Figura 6.3 ilustra el fenómeno.

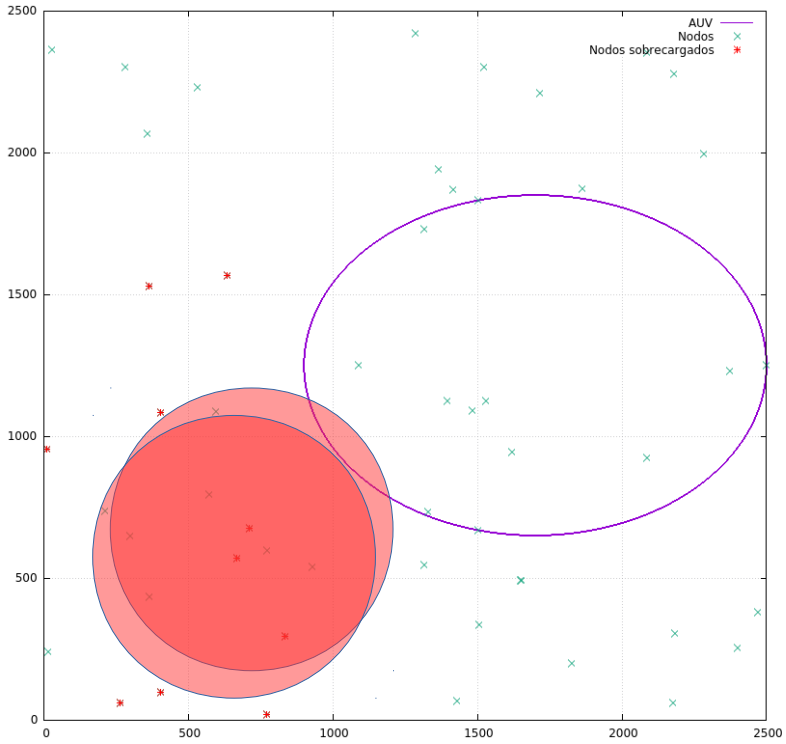


Figura 6.3: Efectos de un bucle.

Como se puede apreciar, el bucle tiene un área de interferencia que afecta a 7 nodos con lo que crea un problema muy serio en esa área.

6.1.3 Recorrido de la AUV

Para finalizar, realizaremos el último experimento que realiza el artículo donde se estudia la influencia de las dimensiones del recorrido de la AUV. En él se plantean cuatro elipses con las dimensiones del eje mayor y menor de (200,300), (300, 500), (600, 800) y (800,1100). En la Figura 6.2 se ilustra el movimiento elíptico que describe el submarino. Estos recorridos se especifican en el artículo.

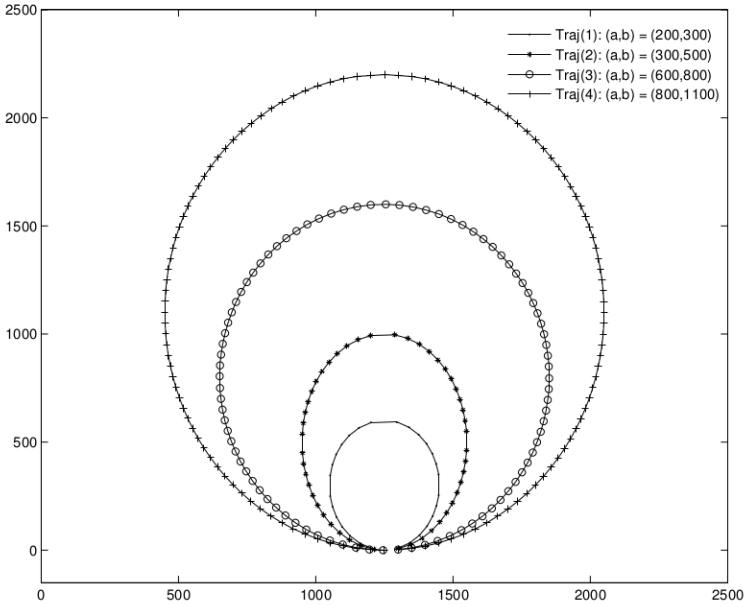
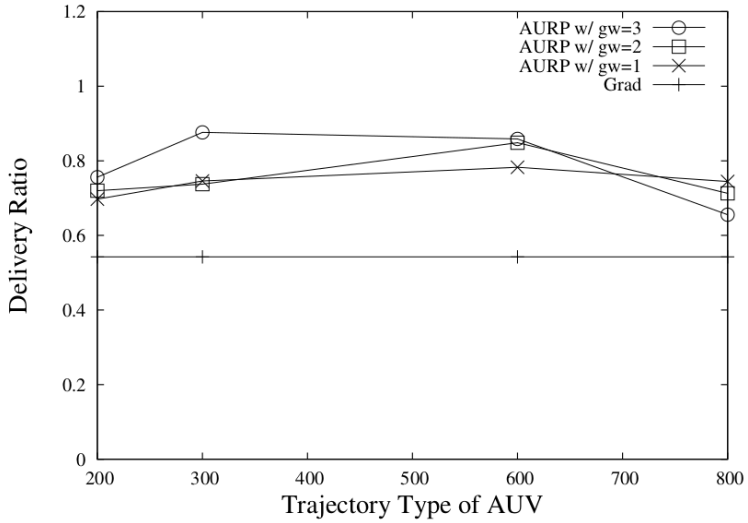
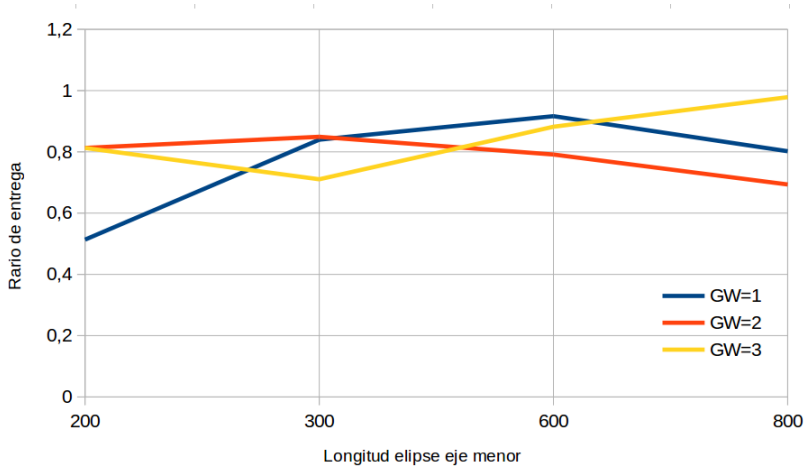


Figura 6.2: Descripción de movimientos elípticos

A continuación se ilustran los resultados obtenidos por el artículo y los obtenidos por la simulación del proyecto.



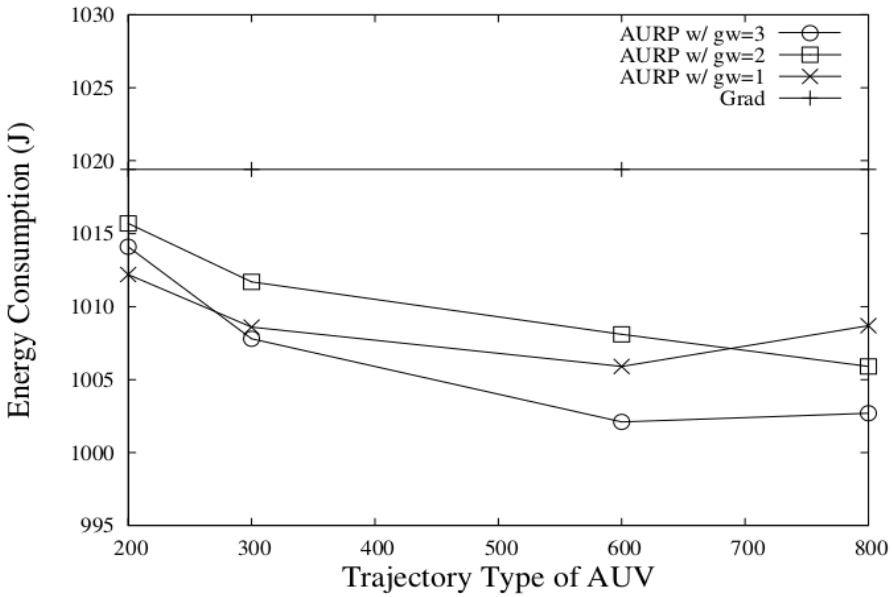
Gráfica 6.9: Resultados del articulo



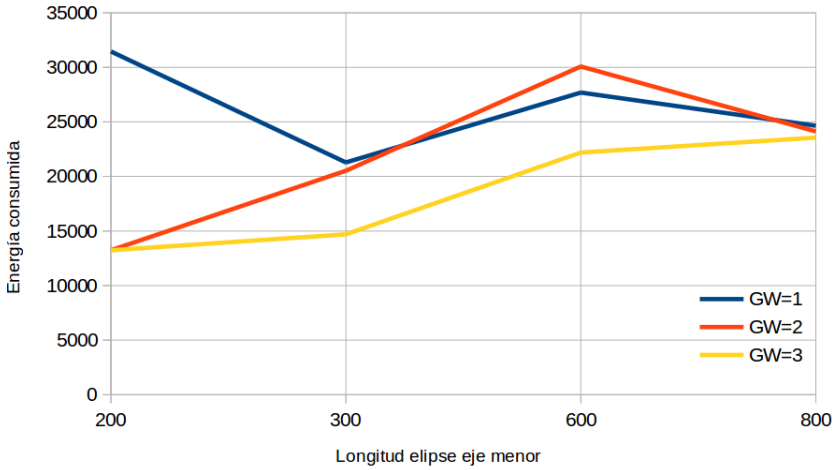
Gráfica 6.10: Ratio de entrega en la simulación.

La comparativa de las dos gráficas muestra que las tendencias son las mismas. Al inicio y al final el ratio desciende. Esto es lo normal puesto que al inicio la elipse es más pequeña y al final la elipse es demasiado grande. Pero en la simulación hay anomalías. La primera anomalía aparece al inicio de GW=1, el ratio es muy inferior a las otras dos y esto es debido a la formación de bucles de donde la información no sale. Como hemos explicado anteriormente este tipo de bucles genera un gran consumo de energía por lo que en la gráfica 6.10 se debe apreciar un mayor consumo. La segunda anomalía se encuentra en la gráfica de tres puertas de enlace y longitud 300m donde el ratio presenta un mínimo local cuya causa se explicará con la Gráfica 6.13. La última anomalía tiene lugar en el final de la gráfica, para 3 puertas de enlace, donde tiene un ratio elevado. Esto es explicable puesto que con tres puertas de enlace y una elipse grande las puertas están bien repartidas en el mapa.

Desde el punto de vista de los consumos, en las Gráficas 6.11 y 6.12 se muestran los datos obtenidos.

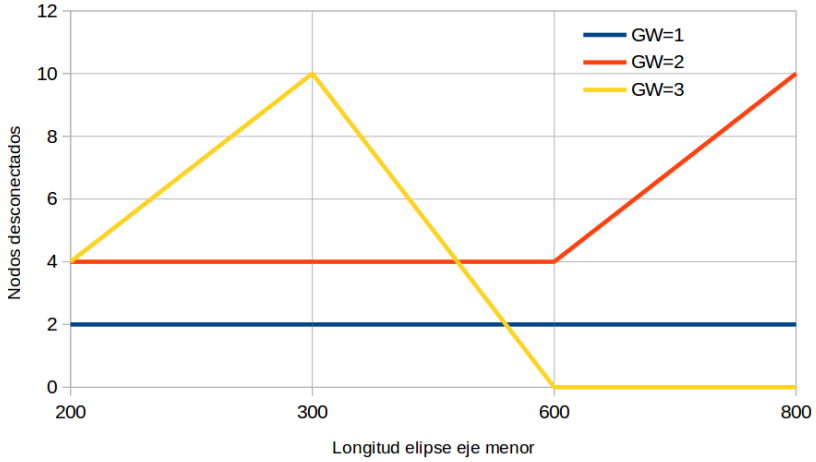


Gráfica 6.11: Consumo en el artículo.



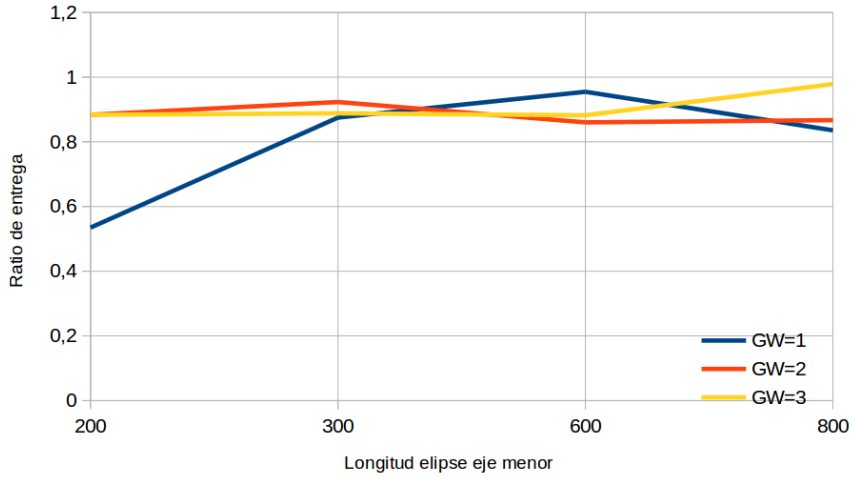
Gráfica 6.12: Consumo en la simulación.

Como se dijo anteriormente, el consumo de energía para GW=1 y longitud 200m es muy elevado afianzando más la teoría de que es debido a los bucles. Cuanto mayor es la tasa de entrega menor consumo tienen los nodos.



Gráfica 6.13: Numero de nodos desconectados.

El número de nodos desconectados de la red para tres puertas y 300m de longitud es de 10 lo que explica por qué el ratio en esa simulación es menor. Si no se ven contemplados en la simulación esos nodos el ratio es el de la Gráfica 6.14



Gráfica 6.14: Ratio en función de los nodos conectados.

Como se puede observar, la anomalía ha desaparecido.

6.2 Conclusiones

Ha sido posible replicar los resultados del artículo, aunque aparecen algunas discrepancias imputables a ciertos parámetros no especificados en el artículo.

El buen funcionamiento del protocolo depende en gran medida a una buena configuración de la red, donde se genere un grafo sin bucles y con forma de árbol. Para ello es necesario que los mensajes de reconocimiento llegan adecuadamente y periódicamente a los nodos.

7 Evaluación del protocolo

7.1 Parámetros a evaluar

En este apartado se van a evaluar dos parámetros que en el artículo no se mencionan como son:

- Posición del sumidero. Se va a estudiar la influencia de la posición del sumidero en el mapa.
- Tiempo de espera de PHE. Se va a estudiar la influencia que tiene el tiempo de espera a la hora de procesar los mensajes de PHE. Al recibir un mensaje de reconocimiento el nodo pone un contador esperando que lleguen más mensaje de reconocimiento. Cuando expira el tiempo se procesan todos esos mensajes. Se va a estudiar la influencia de modificar ese tiempo de espera.

7.2 Escenarios y motivación

7.2.1 Posición de sumidero

El experimento donde se cambia la posición del sumidero viene motivado por el hecho de que no hay motivo para colocar el sumidero en el lateral puesto que la información viaja por cable a una estación en la superficie y de ella a un barco. No hay restricciones en la colocación del sumidero por lo que ver la influencia de su posición. Se espera que cuanto más centrado se encuentre mejores resultados se obtendrán.

El escenario para este experimento consta de 5 pruebas para cada 1, 2 y 3 puertas de enlace. Donde el sumidero avanza sobre un eje tomando las posiciones de 2500m, 2313m, 2126m, 1939m y 1750m. El submarino adopta los movimientos que muestra la Figura 7.1.

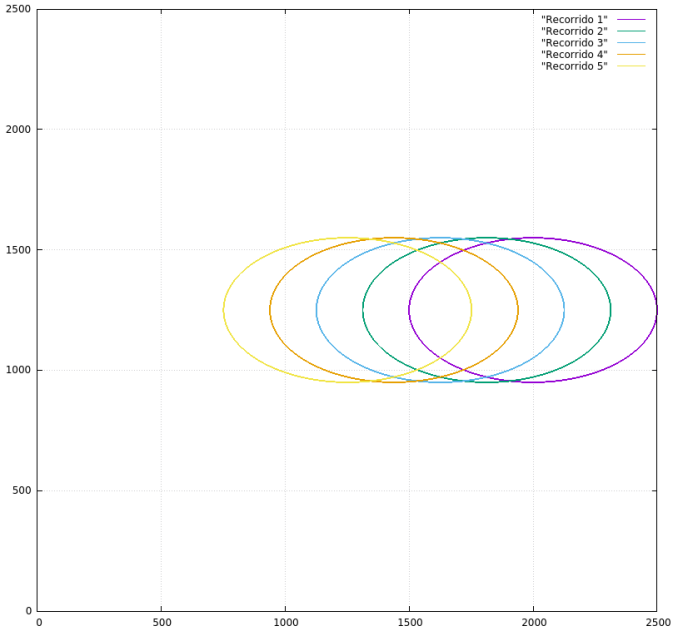


Figura 7.1: Movimientos de la AUV.

7.2.2 Tiempo de procesado

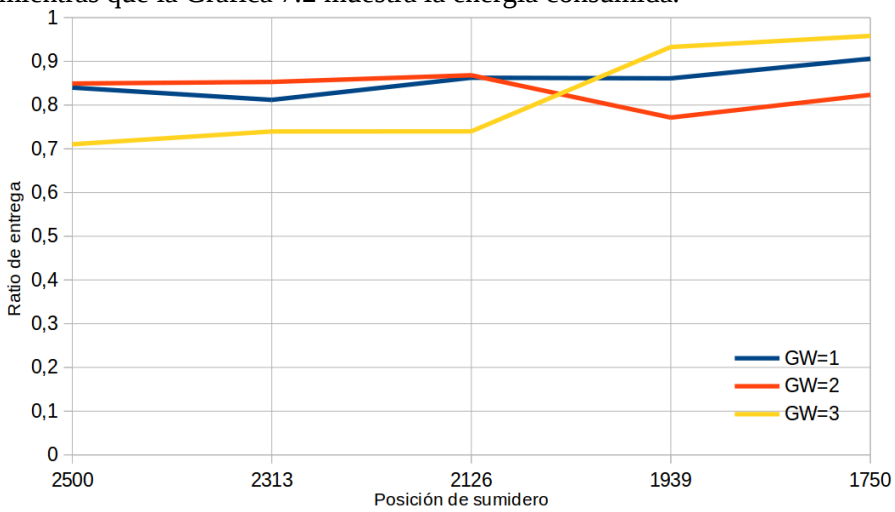
Este experimento está motivado por la necesidad de eliminar los bucles que se generan. Se estudia el efecto que tiene modificar la variable que regula cuanto tiempo hay que esperar antes de procesar la cola de mensajes de reconocimiento, evaluando su eficacia para resolver el problema de creación de bucles.

La serie de experimentos consta de 20 experimentos que combinan 1, 2 y 3 puertas de enlace con diferentes tiempo, 50s, 100s, 150s, 200s, 250s, 300s y 350s.

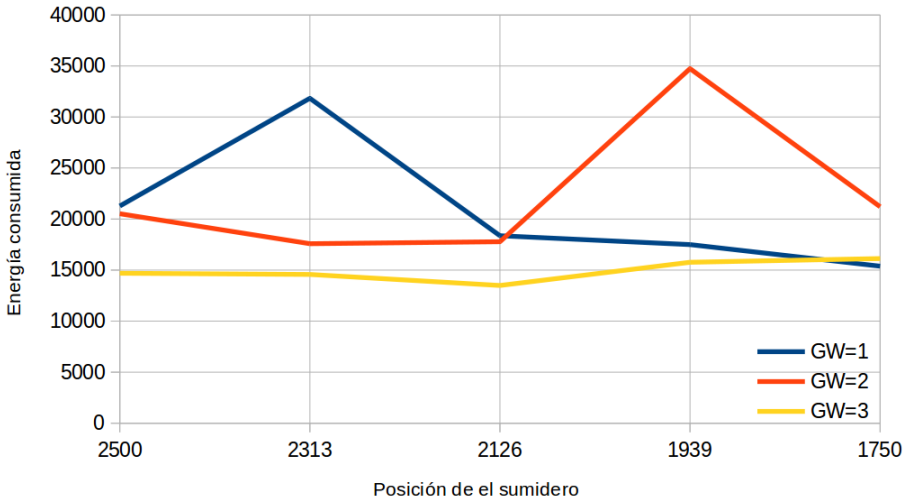
7.3 Resultado

7.3.1 Posición de los sumideros

La Gráfica 7.1 muestra el efecto que tiene el cambio de posición del sumidero con respecto a la eficiencia, mientras que la Gráfica 7.2 muestra la energía consumida.



Gráfica 7.1: Ratio según la posición del sumidero.



Gráfica 7.2: Energía consumida.

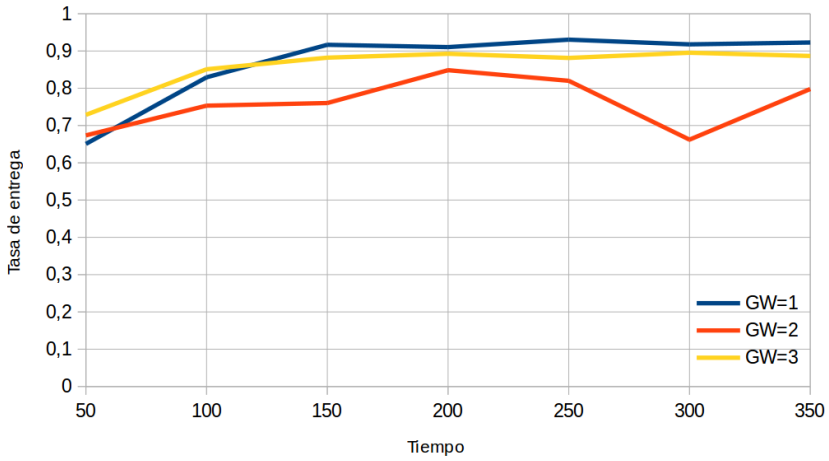
Como se puede observar en las gráficas, con forme el sumidero se va acercando al centro la tasa de entrega aumenta y el consumo de energía se mantiene o cae ligeramente. Existen dos anomalías causadas por bucles, dos picos de consumo de energía. Se ha comprobado que los nodos 15 y 46 consumen mucho más, indicador claro de que es un bucle como ya se vio anteriormente. El incremento tan drástico de la tasa de entrega para 3 puestas se debe al hecho

de que pasa de tener 10 nodos desconectados, un 20%, a tener todos conectados. Tanto con 1 y 2 puertas de enlace están desconectados 2 y 4 nodos respectivamente.

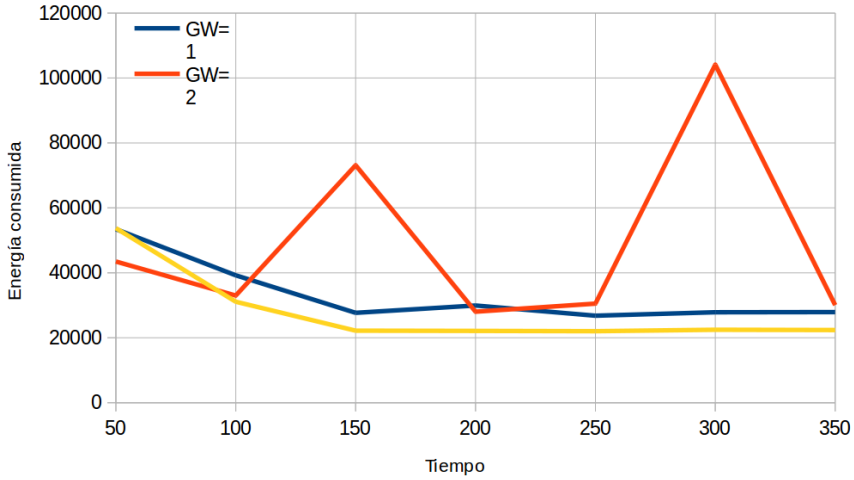
Lo que esto indica es que centrar el sumidero mejora la recepción, el consumo y el número de nodos conectados aunque sólo ligeramente.

7.3.2 Tiempo de procesado

La Figura 7.3 representa la tasa de entrega para la serie de experimentos que se han definido anteriormente y la Figura 7.4 muestra la energía consumida.



Gráfica 7.3: Ratio de entrega para los diferentes tiempos.



Gráfica 7.4: Energía consumida

Las gráficas demuestran la importancia que tiene el tiempo de procesamiento a la hora de la eficiencia. Si el tiempo es mayor que el tiempo entre la creación de mensajes PHE, la eficiencia y el consumo mejoran. También es importante elegir un tiempo adecuado porque como se ve en la Gráfica 7.4 si se elige un tiempo que sea múltiplo del tiempo de creación de mensajes PHE se generan bucles que afectan en la eficacia y eficiencia.

La línea que representa cuando hay dos puertas de enlace es ligeramente inferior puesto que el número de nodos conectados a la red es menor, hay 4 nodos desconectados. Al colocar los nodos aleatoriamente pueden generarse cúmulos donde los mensajes PHE no alcanzan. Los mensajes de PHE no alcanzan por dos motivos, primero por estar muy alejados y un segundo por inanición. La inanición se produce porque la señal de los mensajes PHE se va degradando debido a que los sensores solo envían un mensaje PHE si han recibido uno. Por lo que cuanto más alejado estén menos mensajes de PHE recibirán.

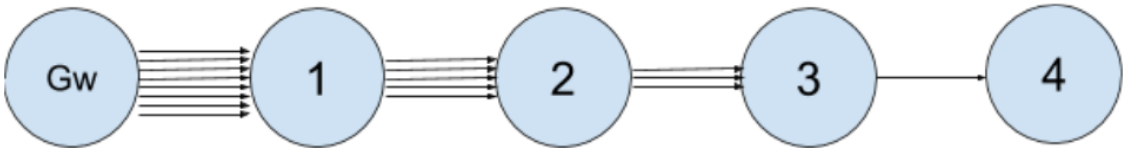
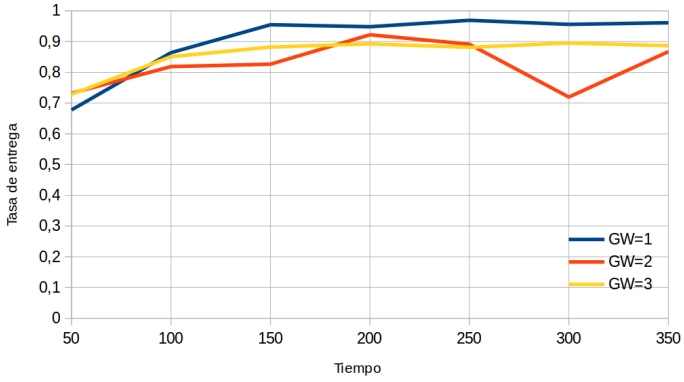


Figura 7.2: Flujo de mensajes PHE.

La Figura 7.2 ejemplifica el flujo normal de los mensajes PHE donde a los nodos más alejados les llegan menos paquetes de reconocimiento.

Puesto que la colocación de los nodos es aleatoria y el número de nodos conectados a la red varía por motivos discutibles, además del explicado anteriormente, la Gráfica 7.5 muestra la tasa de entrega en función de los nodos conectados. En ella se aprecia que todas las líneas se comportan de forma similar, no se aprecia una línea sensiblemente inferior.



Gráfica 7.5: Tasa de recepción en función de nodos conectados

También destacar que es la única experimentación en la que se han obtenido resultados cercanos a los esperados porque hay menos perturbaciones de las habituales. Esto indica que es esta variable la que ayuda a generar más inestabilidades. Además destacar que una buena parametrización de esta variable no soluciona el problema, aunque lo puede mitigar notablemente.

8 Conclusiones y trabajos futuros

Para concluir decir que en el trabajo se ha cumplido todos los objetivos marcados. Se ha estudiado el protocolo AURP que se explica en el artículo [1] estudiando el contexto en el que se encuentra, con los distintos tipos de protocolos de comunicación subacuática en redes de sensores y las dificultades con la que se encuentran las comunicaciones subacuáticas debido al medio. Además se ha visto las posibles aplicaciones de esta tecnología.

Para realizar la simulación se ha realizado un estudio de las mejores opciones para simular el modelo y se ha decidido que el mejor simulador para el trabajo es el NS-3. Posteriormente se ha visto la estructura general del mismo, se ha diseñado e implementado el modelo del protocolo utilizando una serie de pruebas para comprobar su buen funcionamiento y pudiendo asegurar que la implementación del protocolo se ajusta a lo especificado en el artículo.

Una vez todo diseñado se ha realizado una prueba de concepto donde se validaba que el modelo tenía un comportamiento parecido al del artículo. Una vez validado se ha evaluado el funcionamiento del mismo observando ciertas anomalías en el que hay que solventar para que el protocolo funcione adecuadamente. Problemas como los bucles o el deterioro del paso de mensajes PHE han sido identificados.

Con todo esto se puede afirmar que se han cumplido los objetivos marcados pero la investigación en este protocolo puede profundizarse más con trabajos futuros. Por ejemplo, el protocolo necesita un mecanismo de verificación de entrega de mensaje. Esto es implementable con métodos como envío de mensajes ack de vuelta para validar la entrega, ya sean ack por paquete o por bloque de paquetes.

Se ha visto que la creación de bucles es un problema importante que se puede mitigar utilizando un tiempo de espera para procesar los mensajes PHE mayor que el tiempo que tarda las puertas en crearlos pero no debe ser múltiplo del mismo. Esto no soluciona el problema, solo lo mitiga. Hay diferentes métodos que pueden solucionar este problema. Uno de ellos es que los nodos comprueben qué mensajes han enviado y si hay mensajes repetidos dejar de enviarlos y poner como puerta de enlace a sí mismos. Otro método puede ser establecer un número máximo de saltos que puede hacer un mensaje, si los nodos se descongestionan podrán oír los mensajes PHE. Este es un trabajo destacable y donde convendría estudiar de las diferentes opciones.

Como trabajo futuro hay que solucionar el problema de la degradación de mensajes PHE. Una posible solución es que los sensores envíen mensajes de reconocimiento periódicamente sin necesidad de recibir mensajes de PHE. Adicionalmente a esto, los nodos podrían reiniciar el valor de puerta de enlace al mismo tiempo. Estas opciones podrían solucionaría los dos problemas.

9 Bibliografía

[1] S. Yoon, A. K. Azad, H. Oh, and S. Kim, “AURP: An AUV-aided underwater routing protocol for underwater acoustic sensor networks,” *Sensors*, vol. 12, no. 2, pp. 1827–1845, 2012.

[2] “Red de sensores,” 2017. [Online]. Available: https://es.wikipedia.org/wiki/Red_de_sensores.

[3] A. Gkikopouli, G. Nikolakopoulos, and S. Manesis, “A survey on Underwater Wireless Sensor Networks and applications,” *2012 20th Mediterr. Conf. Control Autom.*, pp. 1147–1154, 2012.

[4] I. F. Akyildiz, D. Pompili, and T. Melodia, “Underwater acoustic sensor networks: research challenges,” *Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, May 2005.

[5] X. Lurton, *An Introduction to Underwater Acoustics*, vol. 97. 2010.

[6] S. EL-Rabaie, D. Nabil, R. Mahmoud, and M. A. Alsharqawy, “Underwater Wireless Sensor Networks (UWSN), Architecture, Routing Protocols, Simulation and Modeling Tools, Localization, Security Issues and Some Novel Trends,” *Netw. Commun. Eng.*, vol. 7, no. 8, pp. 335–354, 2015.

[7] D. Mazón Menéndez, “Implementación de algoritmos de enrutamiento para redes inalámbricas de sensores móviles.” pp. 1–29, 2011.

[8] “Packet Tracer | Cisco NetAcad.” [Online]. Available: https://www.netacad.com/es/about-networking-academy/packet-tracer/. [Accessed: 26-Jun-2017].

[9] “GNS3 | The software that empowers network professionals.” [Online]. Available: https://www.gns3.com/. [Accessed: 26-Jun-2017].

[10] “NetSim-Network Simulator & Emulator | Home.” [Online]. Available: http://www.tetcos.com/#. [Accessed: 26-Jun-2017].

[11] “ns-3.” [Online]. Available: https://www.nsnam.org/. [Accessed: 26-Jun-2017].