



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escuela Técnica Superior de Ingeniería Informática

Universitat Politècnica de València

Simulación de redes inalámbricas de sensores móviles

Autor: Bernardo López Ferreres

Directores: Juan Vicente Capella Hernández, Salvador Climent Bayarri

Julio 2012

Tabla de Contenidos

1. Introducción.....	3
1.1 Motivación.....	3
1.2 Objetivos.....	4
1.3 Estructura de la memoria.....	4
2. Underwater Sensor Networks.....	5
2.1 Estado del arte.....	5
3. Herramientas.....	12
3.1 Descripción del simulador NS-3.....	12
4. Implementación.....	12
4.1 Descripción del protocolo VBF (Vector-Based Forwarding).....	12
4.1.2 El algoritmo “Self-Adaptation”.....	13
4.2 Desarrollo del protocolo.....	15
4.3 Descripción de la cabecera.....	16
5. Experimentación.....	17
5.1 Distribución espacial.....	17
5.2 Parámetros utilizados.....	17
5.3 Resultados.....	18
6. Conclusiones y trabajo futuro.....	37
7. Bibliografía.....	38

1.Introducción

1.1 Motivación

En este proyecto se va a proceder a estudiar la viabilidad y la capacidad de un algoritmo para redes de sensores subacuáticos. Aunque ya existen un gran número de algoritmos de enrutamiento para redes de sensores móviles, éstos están diseñados para funcionar con una arquitectura física completamente diferente, basada en ondas electromagnéticas. Es necesaria, por tanto, la adaptación o el desarrollo desde cero de un método de encaminamiento para el caso de las redes subacuáticas, teniendo en cuenta las peculiaridades de este medio. Destacan especialmente las siguientes: una velocidad de propagación enormemente inferior(en torno a 5 órdenes de magnitud), un rango de transmisión generalmente menor al que ofrecen las ondas electromagnéticas en el aire y una velocidad de transmisión también reducida. Estas diferencias vienen dadas por el uso de ondas acústicas, como los sonars, en vez de ondas electromagnéticas, que sufren una atenuación muy fuerte en el agua. Las frecuencias usadas en este tipo de sistemas van desde los 20 Hz hasta los 20 kHz, el rango del oído humano, si bien es cierto que existen otros de alta frecuencia, de 85 a 350 kHz, aunque son de un alcance más reducido.

En este campo, uno de los aspectos más fundamentales a tener en cuenta, y el cuál es uno al que mayor interés se le está prestando actualmente, es el gasto energético, ya que en la mayoría de los casos los sensores o nodos son de difícil recuperación o esta es muy costosa, por lo que nos interesa que estén activos el mayor tiempo posible.

Hay diversos campos de investigación donde las redes de sensores subacuáticos pueden aportar datos de gran importancia. Pueden ser usados para recoger información sobre, por ejemplo en geología, sobre el fondo marino o para prevenir desastres naturales, como las erupciones volcánicas. También se pueden usar para explorar unos yacimientos arqueológicos en zonas hundidas en el mar, y también permiten el estudio del ecosistema marino y las especies que en él habitan, pudiendo medir el nivel de contaminación en el agua, y en donde la comprensión biológica del entorno permite una mejor gestión de la capacidad pesquera de la zona. También hay una existencia de actividades militares interesadas en este campo, ya que se podrían usar las redes de sensores subacuáticos para recoger información sobre vigilancia táctica.

Como se trata de un área de investigación de reciente aparición, donde los dispositivos físicos reales son escasos y caros, y donde existen muchas dificultades para realizar un despliegue real de un banco de pruebas, se hace necesario la utilización de un medio de simulación donde se permita poner a prueba los algoritmos propuestos de una forma realista y eficiente. Estas simulaciones también nos permitirían ajustar, con un coste muchísimo menor, lo mejor posible los parámetros del protocolo estudiado en posterior despliegue real. Para que esto sea posible, es necesario que el simulador sea lo más fiel y realista posible en sus resultados.

En nuestro caso, el simulador que vamos a emplear es la suite académica ns-3 (network simulator versión 3), nueva versión del ampliamente conocido ns-2. En la versión actual ya aparecen implementados las capas físicas y MAC de la pila de protocolos para las redes subacuáticas, incluyéndose varios modelos de propagación, de cálculo de ruidos, de medios físicos así como de protocolos de acceso al medio. El simulador también incluye una serie de elementos que nos serán de gran ayuda para el desarrollo del proyecto: sistemas de posicionamiento, para situar los nodos en un espacio tridimensional, sistemas de movilidad, para simular el movimiento de los nodos o sistemas de alimentación energética, para simular el consumo energético y una posible desconexión

de los nodos al agotarse la energía.

Por lo tanto, vamos a crear un nuevo elemento sobre una capa de la pila de protocolos todavía no estudiada para el simulador ns-3: vamos a implementar un protocolo en el nivel de red para sensores inalámbricos. Entre las varias alternativas posibles se elige uno pensado para este tipo de dispositivos. Es el protocolo VBF[1], por las siglas en inglés Vector Based Forwarding.

En este protocolo, cada paquete contiene la posición de la fuente, el destino y del nodo que lo ha enviado. El “camino de reenvío” (forwarding path) está especificado por el vector definido entre la fuente y el destino. Cada vez que un nodo recibe un paquete, comprueba si está suficientemente cerca del “camino de reenvío” (tiene que estar a una menor distancia que una distancia umbral predefinida), y si lo está reenvía el paquete, en caso contrario lo descarta.

Por tanto, durante el funcionamiento de una red de estos sensores, los nodos que reenvían el paquete forman una “tubería de enrutamiento”, los nodos que se encuentran en la tubería pueden reenviar los paquetes y aquellos que no están suficientemente cerca del “vector de enrutamiento” no lo reenvían. Uno de los puntos a estudiar será el consumo energético asociado al protocolo, ya que es uno de los factores más importantes a la hora de decidir la viabilidad para aplicar el protocolo en un despliegue real.

1.2 Objetivos

Este proyecto tiene como primer objetivo la implementación del protocolo de enrutamiento VBF en el simulador ns-3. Una vez implementado, se prosigue a realizar un estudio intensivo de éste, comparándolo en primer lugar con los resultados del artículo que tomaremos como referencia para validar el protocolo, y en segundo lugar, desarrollando nuevos tests para cubrir casos no tratados.

Esto nos permitirá también el estudio en profundidad del módulo para redes subacuáticas del simulador ns-3. Actualmente hay implementados los niveles inferiores de la pila de protocolos: la física y la MAC. La primera incluye el canal, modelos de propagación, de SINR, etc... y la segunda varias propuestas distintas entre las que elegir. De esta forma, nuestro protocolo añadirá una nueva capa aún no explorada en la suite.

1.3 Estructura de la memoria

El contenido de la memoria será el siguiente:

- ♣ Un apartado donde se explicará el estado actual de desarrollo de las redes de sensores subacuáticas o UWSN(Underwater Sensor Networks).
- ♣ Un apartado donde se describirá con más detalle las herramientas utilizadas para el desarrollo del proyecto, en este caso el simulador NS-3.
- ♣ Un apartado donde se explicará la implementación del protocolo a desarrollar, en este caso el protocolo de enrutamiento VBF, y donde se describirán y explicarán las decisiones tomadas para su implementación.
- ♣ Un apartado donde se detallará los experimentos realizados y se mostrarán y analizarán los resultados obtenidos.
- ♣ Y finalmente una apartado donde se expondrán las conclusiones obtenidas y se propondrán posibles mejoras para realizar en el futuro

2.Underwater Sensor Networks

2.1 Estado del arte

Las redes de sensores subacuáticas (o UWSN, underwater sensor networks en inglés) tienen el potencial de permitir nuevas aplicaciones y de mejorar nuestra habilidad para observar e investigar el océano. Los Vehículos autónomos subacuáticos (o AUVs , Autonomous Underwater Vehicles en inglés) equipados con sensores subacuáticos también pueden tener aplicación en la exploración de recursos naturales y en la obtención de información científica en misiones colaborativas de seguimiento. Estas aplicaciones potenciales serán viables habilitando comunicaciones entre dispositivos subacuáticos. UWSN consistirán en sensores y vehículos desplegados debajo del agua e interconectados mediante enlaces acústicos para desempeñar tareas de seguimiento.

Las redes de sensores subacuáticas permiten un gran rango de aplicaciones, incluyendo:

- ⤴ **Redes de muestreo del océano.** Las redes de sensores pueden realizar un muestreo sinóptico y adaptativo del entorno costero del océano.
- ⤴ **Seguimiento ambiental.** Las USWN pueden realizar un seguimiento sobre la contaminación (química, biológica y nuclear), de las corrientes marinas y del viento, y un seguimiento biológico tal como rastrear una determinada especie de pez o un microorganismo. Además, UWSN pueden mejorar la predicción del tiempo, detectar cambios en el clima y predecir el efecto de las actividades humanas en el ecosistema marino.
- ⤴ **Exploraciones submarinas.** Las redes de sensores subacuáticas pueden ayudar a detectar reservas de petróleo submarinas, determinar rutas para cables submarinos y asistir en la búsqueda de valiosos minerales.
- ⤴ **Prevención de desastres.** Las redes de sensores también pueden medir la actividad sísmica, pueden proporcionar aviso de tsunamis a zonas costeras o estudiar el efecto de los terremotos submarinos (maremotos).
- ⤴ **Seguimiento sísmico.** Realizar un seguimiento frecuente de la actividad sísmica tiene una gran importancia en la extracción de petróleo de yacimientos submarinos. Las redes subacuáticas de sensores permitirían un mejor enfoque en la administración de los yacimientos.
- ⤴ **Navegación asistida.** Los sensores pueden ser usados para detectar peligros en el fondo marino, localizar rocas peligrosas en aguas poco profundas, posiciones de amarre y restos de naufragios sumergidos.
- ⤴ **Vigilancia táctica.** Las redes de sensores subacuáticas pueden realizar tareas de vigilancia, reconocimiento, localización y de detección de intrusos.
- ⤴ **Reconocimiento de minas.** Las simultáneas operaciones de varios AUVs con sensores acústicos y ópticos pueden ser usadas para realizar rápidas evaluaciones del entorno y detectar objetos parecidos a minas.

Las comunicaciones acústicas son la tecnología mas usada para la capa física en redes subacuáticas. De hecho, las ondas de radios solo se pueden propagar a grandes distancias en agua salada en frecuencias muy bajas (30-300Hz), lo cuál requiere antenas muy grandes y una gran potencia de transmisión. Las ondas ópticas no sufren de una atenuación tan grande pero se ven afectadas por la dispersión. Además, transmitir ondas ópticas requiere una alta precisión en apuntar los rayos láser. Por estas razones, los enlaces en las redes subacuáticas de sensores están normalmente basados en comunicaciones acústicas inalámbricas.

El enfoque tradicional para investigar el océano consiste en desplegar sensores subacuáticos que graben la información durante la misión de seguimiento y entonces recoger los instrumentos[2]. Este enfoque tiene varios inconvenientes:

- ⤴ Los datos obtenidos no pueden ser obtenidos hasta que los sensores son recuperados
- ⤴ La interacción entre un sistema de control en tierra y los sensores es imposible, por lo que se impide cualquier reconfiguración del sistema
- ⤴ Si ocurren fallos o desconfiguraciones puede no ser posible detectarlos hasta las recuperación de los sensores.
- ⤴ La cantidad de datos que puede ser guardada por cada sensor está limitado a la capacidad de su dispositivo de almacenamiento.

Estos inconvenientes pueden ser salvados conectando los dispositivos subacuáticos con enlaces inalámbricos que funcionan mediante comunicaciones acústicas. A pesar de la existencia de protocolos de red recientemente desarrollados para redes inalámbricas, las características únicas del canal de comunicación en las redes subacuáticas, tales como una capacidad limitada y un retraso de propagación alto y variable[2], hacen necesario nuevos protocolos de comunicación de datos que sean eficientes y fiables.

Los mayores desafíos en el diseño de redes subacuáticas acústicas son:

- ⤴ El ancho de banda disponible es muy limitado.
- ⤴ El retardo de propagación es de un orden de magnitud 5 veces más grande que en los canales terrestres de radiofrecuencia (RF), y variable.
- ⤴ Se pueden tener grandes tasas de error y pérdidas temporales de conexión.
- ⤴ Los sensores subacuáticos están caracterizados por su alto precio debido al pequeño número de proveedores
- ⤴ La energía de las baterías es limitada y normalmente no se pueden recargar recargables.
- ⤴ Los sensores subacuáticos son propensos a fallar debido a la corrosión y a la suciedad.

Diferencias con las redes de sensores terrestres

Las principales diferencias entre redes de sensores terrestres y subacuática son.

- ⤴ **Coste.** Mientras los nodos sensores se espera que se vuelvan muy baratos, los sensores subacuáticos son dispositivos caros. Esto es debido a unos transceptores subacuáticos más complejos y la protección necesaria para el hardware para soportar el ambiente subacuático.
- ⤴ **Despliegue.** Mientras que las redes de sensores terrestres tienen un despliegue muy denso, en las subacuáticas es generalmente más esparcido.
- ⤴ **Energía.** La cantidad de energía necesaria en sensores subacuáticos es mayor debido a la diferencias en la tecnología de la capa física (acústico vs radiofrecuencia), mayores distancias y unas técnicas de procesamiento de señal más complejas en los receptores
- ⤴ **Memoria.** Mientras que los sensores terrestres tienen una capacidad de almacenamiento muy limitada, los sensores subacuáticos pueden necesitar guardar cierta información debido al que canal subacuático puede ser intermitente.
- ⤴ **Correlación Espacial.** Mientras que las lecturas de los sensores terrestres suele ser correlacionadas, esto no suele ocurrir en las redes subacuáticas debido a la mayor distancia entre sensores.

Factores que influyen en el diseño de protocolos subacuáticos.

- ⤴ **Pérdida de transmisión.** Consiste en la atenuación y el esparcimiento geométrico. La atenuación es provocada por la absorción debida a la conversión de energía acústica en calor, e incrementa con la distancia y la frecuencia. El esparcimiento geométrico se refiere al esparcimiento de energía sónica debido a la expansión de los frentes de onda. Aumenta con la distancia de propagación y es independiente de la frecuencia.
- ⤴ **Ruido.** Puede ser clasificado como provocado por el hombre o como ambiental. El primero es normalmente generado por el ruido de máquinas (bombas, estaciones de energía) y actividad de navegación mientras que el segundo está relacionado con la hidrodinámica (movimiento del agua, lluvia) y fenómenos biológicos y sísmicos.
- ⤴ **Alto retraso y variabilidad de retraso.** La velocidad de propagación en el canal subacuático acústico es del orden de 5 veces menor que en el canal de radio. Este gran retraso de propagación (0.67s/km) y su variabilidad reducen el rendimiento del sistema.

Arquitecturas de comunicación.

Redes de sensores subacuáticas en 2D

Un grupo de nodos son desplegados en el fondo del océano. Los nodos sensores subacuáticos están interconectados con una o más “puertas subacuáticas” (underwater gateways o uw-gateways) mediante enlaces acústicos inalámbricos. Uw-gateways son dispositivos de red encargados de transportar la información desde el fondo del océano hasta estación en la superficie. En aguas poco profundas, los nodos desplegados en el fondo pueden comunicarse directamente con la estación en la superficie. En la Imagen 1[9] podemos ver la arquitectura.

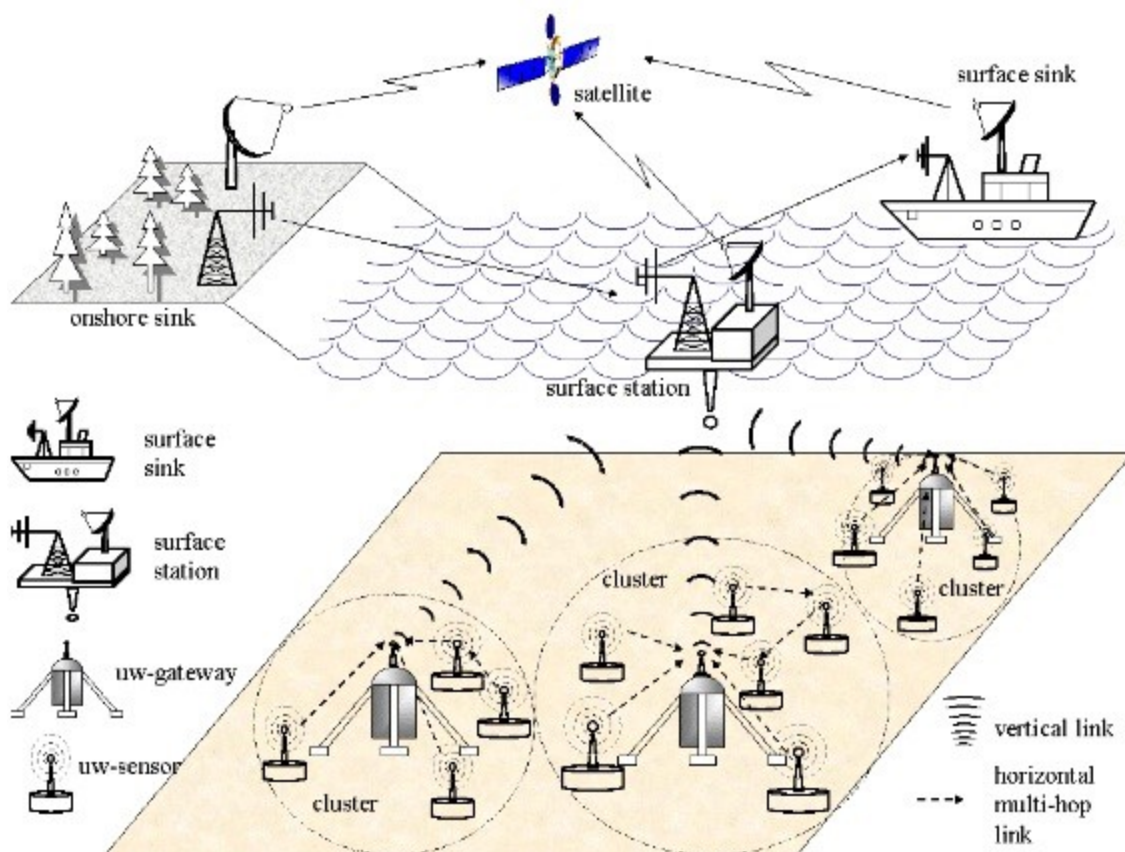


Imagen 1

Red de sensores subacuáticas 2D

Redes de sensores en 3D

Las redes subacuáticas tridimensionales son usadas para detectar y observar fenómenos que no pueden ser adecuadamente observados mediante una red con los sensores desplegados en el fondo del océano. En las redes tridimensionales, los nodos sensores flotan a diferentes profundidades para observar fenómenos. En esta arquitectura, mostrada en la Imagen 2[9], cada sensor tiene una boya flotante que puede ser inflada mediante una bomba. La boya empuja el sensor hacia la superficie marítima. La profundidad del sensor puede ser regulada.

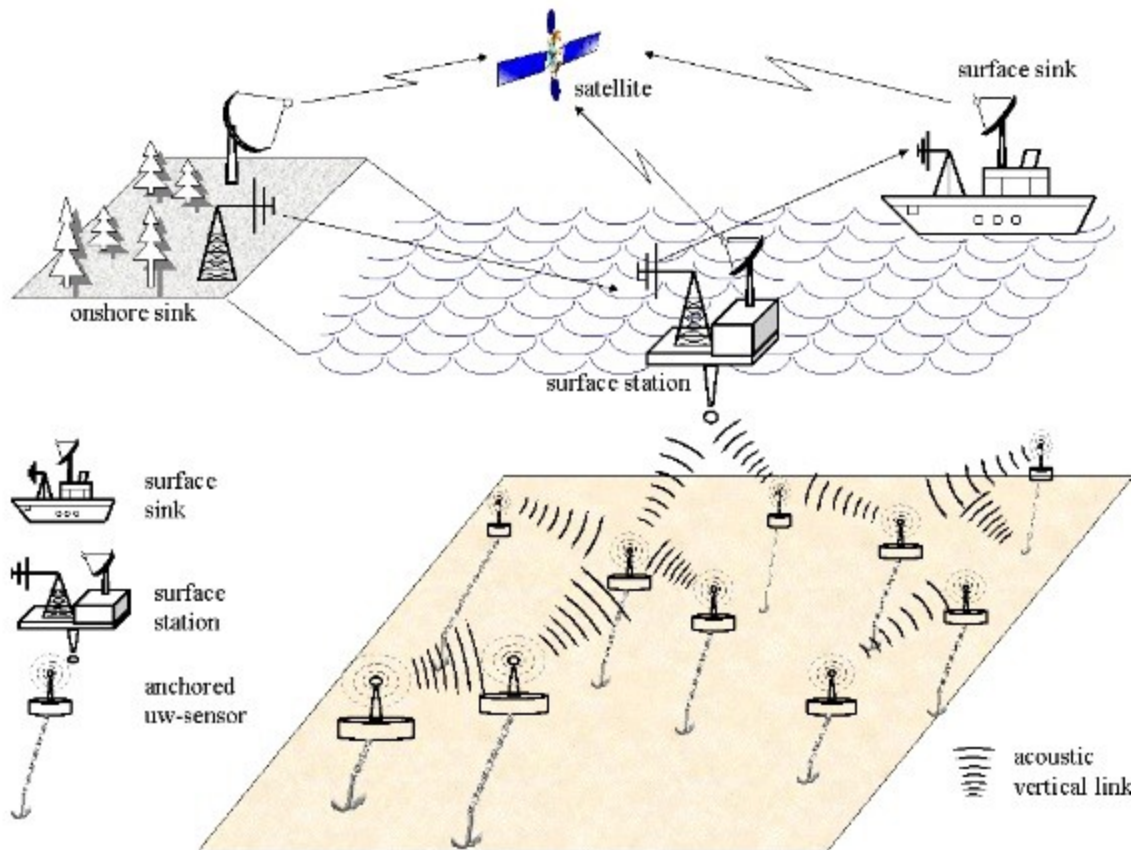


Imagen 2

Red de sensores subacuáticas 3D

Redes de sensores con vehículos autónomos subacuáticos

Los vehículos autónomos subacuáticos, o AUVs, pueden funcionar sin amarres, cables o control remoto y tienen una multitud de aplicaciones en oceanografía y estudios de recursos subacuáticos. La integración de las redes subacuáticas con los AUVs requiere nuevos algoritmos de coordinación de red como:

- ⤴ Muestreo adaptativo. Esto incluye el control de estrategias para comandar el vehículo móvil hacia los lugares donde la información será más útil.
- ⤴ Auto.configuración. Esto incluye procedimiento de control para detectar fallos de conectividad debido a problemas en el canal o al fallo de un nodo. Además, un AUV puede ser usado para el despliegue de nuevos sensores o para la instalación y mantenimiento de la red de sensores.

Uno de los objetivos de los AUVs es hacerles confiar en su inteligencia local y ser menos dependiente de las comunicaciones provenientes de la costa. En la Imagen 3[9] podemos observar la arquitectura.

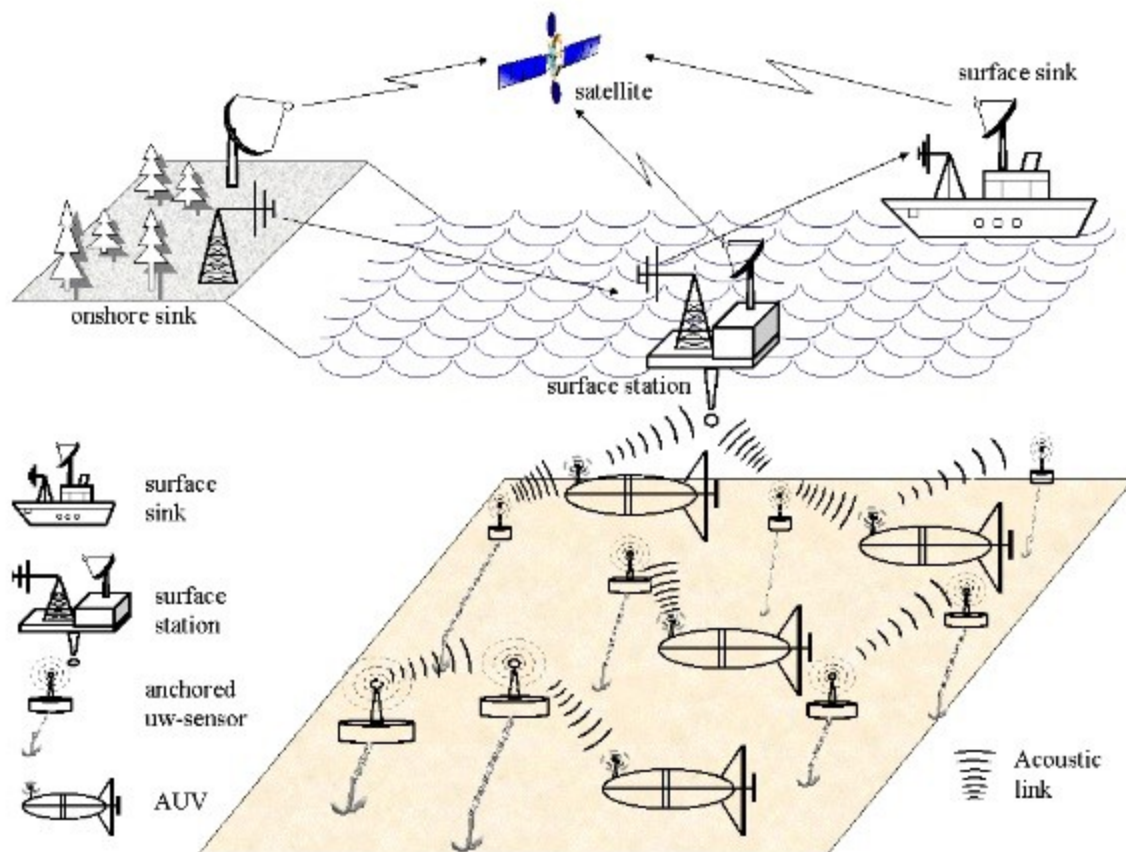


Imagen 3

Redes de sensores con vehículos autónomos subacuáticos

Capa de control de acceso al medio (MAC)

En la última década ha habido una investigación intensiva en los protocolos MAC para redes ad hoc[4] y de sensores terrestres[3]. Pero, debido a la diferente naturaleza del entorno subacuático y sus aplicaciones, los protocolos MAC para redes de sensores terrestres no son las más adecuadas. De hecho, el acceso al medio en redes de sensores subacuáticas posee desafíos adicionales debido a las características del canal subacuático, en particular su limitado ancho de banda, un retraso de propagación alto y variable, asimetría del canal y pérdidas de conexión temporales.

Las soluciones para el control de acceso al medio existentes están basadas en CSMA (Carrier Sense Multiple Access) o CDMA (Code Division Multiple Access). Esto es porque FDMA (Frequency Division Multiple Access) no es adecuada para las UWSN debido al bajo ancho de banda en los canales subacuáticos. Además, TDMA (Time Division Multiple Access) muestra una eficiencia limitada en el ancho de banda debido a las largas esperas de tiempo necesarias en el canal subacuático. Además, el retraso variable hace muy difícil de conseguir una sincronización precisa.

Protocolo basado en CSMA: Slotted FDMA

Protocolos basados en CDMA: DSSS (Direct Sequence Spread Spectrum) y FHSS (Frequency Hopping Spread Spectrum)

Capa de red

En los últimos años ha habido un gran interés en desarrollar nuevos protocolos de enrutamiento para redes de sensores terrestres. Sin embargo, existen varios inconvenientes en usarlos para las redes de sensores subacuáticos. Los protocolos existentes se dividen en 3 tipos, proactivos, reactivos y geográficos.

Los protocolos proactivos (ej. DSDV[5], OLSR) realizan una gran sobrecarga de señalización para establecer las rutas por primera vez y cada vez que la topología de red cambia debido al movimiento o al fallo de los nodos, ya que la nueva información sobre la topología debe ser transmitida a todos los nodos. De esta manera, cada dispositivo es capaz de establecer un camino a cualquier otro nodo de la red, algo que no tiene porque ser necesario en las UWSN. Es necesario que cada nodo guarde en unas tablas la información para poder establecer las rutas.

Los protocolos reactivos (ej. AODV, DSR[6]) son más apropiados para entornos dinámicos pero sufren de una alta latencia y aún necesitan un flujo de paquetes iniciado desde la fuente para establecer los caminos. Los protocolos reactivos son inadecuados para las UWSN ya que causan una gran latencia en el establecimiento de los caminos, que es fuertemente amplificada debajo del agua debido a la lenta propagación de las señales acústicas. Por otra parte, es poco probable que la topología de las UWSN varíe dinámicamente en una escala de tiempo pequeña.

Los protocolos de enrutamiento geográficos (ej. GFG[7], PTKF) son muy prometedores debido a su escalabilidad. Sin embargo, los receptores de radio GPS que se usan en los sistemas terrestres para estimar de una manera precisa la situación geométrica de los nodos sensores, no funcionan correctamente en el entorno subacuático. De hecho, el GPS usa ondas en la banda de 1.5 Ghz que no se propaga en el agua. Con todo, los dispositivos subacuáticos necesitan estimar su posición actual, con independencia del enfoque elegido para el enrutamiento. La localización subacuática puede ser conseguida mediante el aprovechamiento de la baja velocidad del sonido en el agua, que permite una sincronización exacta de las señales, y los datos de distancia entre una pareja de nodos pueden ser usados para realizar una localización 3D.

Recientemente se están proponiendo protocolos de enrutamiento específicos para redes subacuáticas acústicas, como por ejemplo VBF (Vector Based Forwarding)

Capa de transporte

Un protocolo para la capa de transporte es necesario en las UWSN para conseguir un transporte fiable y realizar un control de flujo y un control de congestión. La mayoría de las implementaciones existentes de TCP son inadecuadas para el entorno subacuático ya que el control de flujo funciona basado en un mecanismo basado en ventanas que depende en una estimación precisa del Round Trip Time (RTT). El largo RTT que caracteriza el entorno subacuático afectaría al rendimiento de la mayoría de las implementaciones de TCP.

Los protocolos de transporte “rate-based” existentes también parecen inadecuados para este entorno, ya que confían en las respuestas de control enviados por el receptor para adaptar dinámicamente la tasa de transmisión. El RTT grande y variable puede causar inestabilidad en el control del feedback. Por estas razones es necesario idear nuevas estrategias para conseguir un control de flujo y fiabilidad en las UWSN.

Un protocolo para la capa de transporte diseñado para el entorno subacuático ha sido recientemente propuesto. SDRT[8] (Segment Data Reliable Transport) consigue superar algunos de los desafíos de

las redes de sensores subacuáticas como el bajo ancho de banda, los grandes retardos de propagación..., pero no consigue solucionar el problemas de las zonas sombrías, zonas donde no hay conectividad. SDRT es un primer intento de proponer un protocolo de transporte para las UWSN, sin embargo necesita mejoras.

3. Herramientas

3.1 Descripción del simulador NS-3

Para la realización del proyecto vamos a utilizar el simulador ns-3, concretamente en su versión 3.13. Se trata de un simulador de eventos discretos, Permite simular tanto protocolos unicast como multicast y se utiliza intensamente en la investigación de redes móviles ad-hoc. Implementa una amplia gama de protocolos tanto de redes cableadas como de redes inalámbricas

Se encuentra 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 de estos y por supuesto, el de mayor importancia para nosotros: la capa de red física y MAC para redes subacuáticas.

La versión actual es un fork 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 solventar 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 actual está diseñada para soportar todo el flujo de trabajo de la simulación desde la configuración hasta la recolección y análisis de tramas[10].

4. Implementación

4.1 Descripción del protocolo VBF (Vector-Based Forwarding)

“Vector Based Forwarding” es un protocolo de enrutamiento especialmente diseñado para las redes de sensores subacuáticas y estaría englobado dentro del grupo de protocolos geográficos. En el protocolo cada paquete contiene la posición de la fuente, el destino y del nodo que lo ha enviado, por lo que es necesario añadir al paquete los campos necesarios para contener esta información. El “forwarding path” viene definido por el vector formado entre la fuente y el destino, si el nodo se encuentra lo suficientemente cerca a este “forwarding path”, el nodo reenvía el paquete, sobrescribiendo los campos de posición necesarios. Asimismo, cada paquete contiene un campo RANGE, que sirve para controlar la movilidad de los nodos. Cuando un paquete llega a la zona especificada en el campo destino, el paquete es transmitido en un área determinada por el valor de RANGE. Cada paquete también contiene un campo RADIUS, que es el umbral usado por el nodo para determinar si está suficientemente cerca del “forwarding path”.

También se creará una cola de enviados donde se guardarán los datos identificativos de los paquetes

para evitar múltiples retransmisiones y el gasto energético asociado. Algunos paquetes escuchados pero no retransmitidos también se guardarán en esta cola para evitar su posible reenvío posterior. Todo esto queda esquematizado en el diagrama 1.

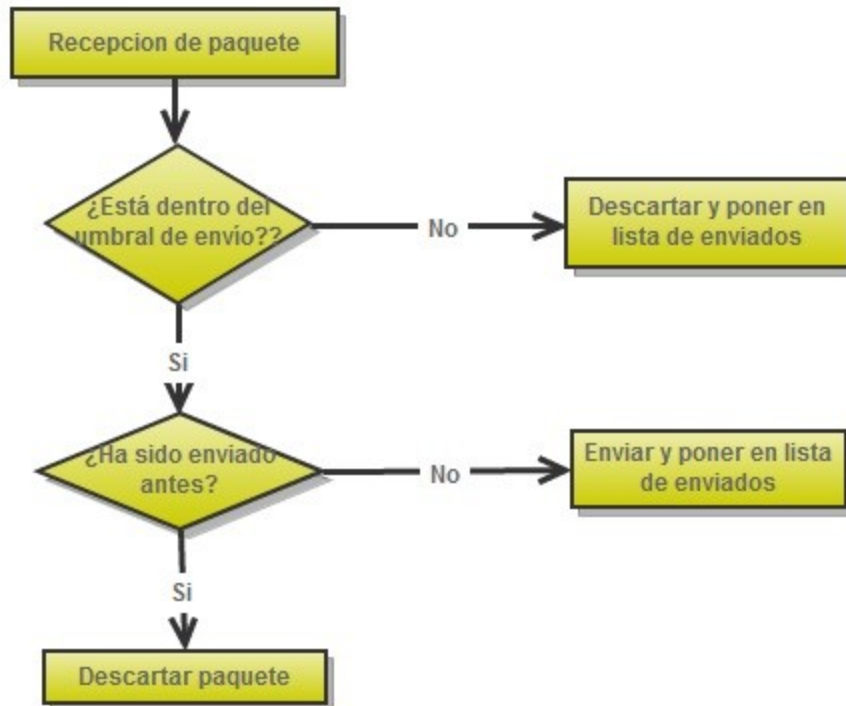


Diagrama 1 - Funcionamiento de VBF

4.1.2 El algoritmo “Self-Adaptation”

En el caso de que haya un gran número de nodos desplegados, puede que en VBF demasiados nodos se vean envueltos en la retransmisión de un paquete, ya que todos los nodos lo suficientemente cerca al “vector de enrutamiento” pueden retransmitir. Debido a la movilidad de los nodos en la red es imposible calcular la densidad global de nodos. Por otra parte, es inapropiado medir la densidad en los extremos de la transmisión (ej. en la fuente y en el destino) debido a la baja velocidad de propagación de las señales acústicas. A continuación se propone un algoritmo “self-adaptation” para VBF que permite a cualquier nodo estimar la densidad en su zona y retransmitir el paquete si es necesario.

El algoritmo está basado en el concepto de factor de deseabilidad para medir la idoneidad de un nodo para reenviar paquetes. El factor de deseabilidad se define como, dado un vector $S1S0$ donde $S1$ es la fuente y $S0$ es el sumidero, para un nodo que puede reenviar F , el factor de deseabilidad α , de un nodo A , es definido como la fórmula de la ecuación 1, donde p es la proyección de A al vector $S1S0$, d es la distancia entre el nodo A y el nodo F , y θ es el ángulo entre $FS0$ y el vector FA . R es el rango de transmisión y W es el radio de la “routing pipe” (la distancia umbral). Cuanto mas grande es este factor para un nodo significa que su proyección al vector de enrutamiento es grande o que no está muy lejos del nodo que le ha enviado el paquete, por lo tanto no es deseable para este nodo reenviar el paquete.

$$\alpha = \frac{p}{W} + \frac{(R-d \times \cos\theta)}{R},$$

Ecuación 1 - Factor de deseabilidad

El algoritmo “self-adaptation” tiene como objetivo seleccionar los nodos más deseables como nodos que reenvíen. En el algoritmo, cuando un nodo recibe un paquete, primero determina si está suficientemente cerca al vector de enrutamiento. Si lo está, entonces el nodo mantiene el paquete por un tiempo relativo a su factor de deseabilidad. En otras palabras, cada nodo cualificado retrasa el reenvío del paquete un intervalo de tiempo llamado $T_{adaptation}$ cuya manera de ser calculado se ve en la ecuación 2, y donde T_{delay} es un retardo máximo predefinido, v_0 es la velocidad de propagación del sonido en el agua (1500 m/s) y d es la distancia entre el nodo y el que le ha enviado el mensaje.

$$T_{adaptation} = \sqrt{\alpha} \times T_{delay} + \frac{R - d}{v_0},$$

Ecuación 2

Durante el periodo de tiempo $T_{adaptation}$, si un nodo recibe el mismo paquete desde otros n nodos, entonces este nodo tiene que calcular el factor de deseabilidad relativo a estos n nodos, $\alpha_1, \dots, \alpha_n$, y el del primer nodo desde el cuál recibió el paquete, α_0 . Si $\min(\alpha_0, \alpha_1, \dots, \alpha_n) < \alpha_c/2n$. Donde α_c es un valor inicial predefinido entre 0 y 3, entonces el nodo reenvía el paquete, sino, lo descarta.

Para que esto funcione, un nodo ha de ser capaz de identificar de forma única cada paquete, para no retransmitirlo de nuevo si lo escucha de otro nodo. Para ello, en la cabecera añadiremos 2 campos más, un ID de nodo de origen y un número de secuencia, para diferenciar los distintos paquetes provenientes del mismo nodo. Asimismo también será necesaria la creación de una cola de paquetes pendientes de envío. Todo esto queda esquematizado en el diagrama 2.

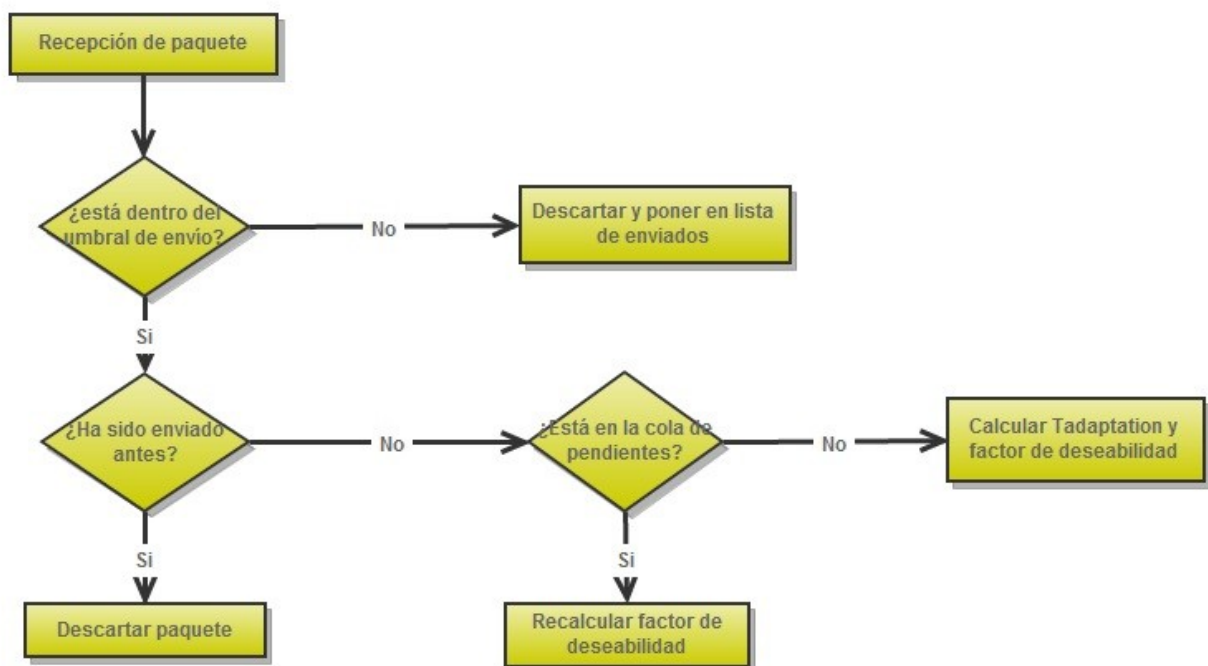


Diagrama 2 - Funcionamiento de VBF con Self-Adapted

4.2 Desarrollo del protocolo

Para el desarrollo del protocolo se han implementado dos clases principales. Por un lado la clase VBFHeader() y por otro la clase VBFNode(). El diagrama 3 muestra un esquema de las relaciones y elementos más importantes de estas clases.

La clase VBFHeader contiene los métodos necesarios para la creación de las cabeceras que se añadirán a los paquetes. A cada paquete enviado se le debe añadir dicha cabecera para ser manejada por el protocolo y que éste pueda tomar las decisiones correspondientes. Esta clase se encargará de serializar y deserializar los campos de la cabecera. Para la codificación de estos se usará un tipo float para los campos que marcarán las posiciones de la fuente, destino y el último nodo que lo ha reenviado. También un tipo float indicar los atributos RADIUS y RANGE, un tipo uint16_t para el ID del nodo origen, un tipo uint16_t para el número de secuencia y un tipo uint8_t para indicar el tipo de mensaje. Todo ellos suma un tamaño total de cabecera de 49 bytes. Podría utilizarse un tipo de menor tamaño para el número de secuencia, dependiendo de la cantidad de paquetes que vayan a ser enviados. No se utiliza un tipo uint8_t para la dirección de origen de los paquetes (que es el tamaño con que se almacena dentro de la clase UanAddress ya implementada en el ns-3) pues limita el número de nodos a 255, lo cual provocaría problema a la hora de realizar simulaciones con un número de nodos superior.

Por otro lado se ha creado la clase VBFNode que se instalará en los nodos que deban tratar con este protocolo. Tiene dos funciones principales: Receive() y Send().

Receive() se encargará de recibir los paquetes que lleguen hasta el nodo, y será invocado por las capas inferiores de la pila de protocolos. Tomará una primera decisión sobre descartarlo o prepararlo para ser retransmitido.

La función de Send() se encargará de llamar a las capas inferiores con el paquete que haya de ser enviado, ya con su cabecera correspondiente debidamente insertada.

Otro método de gran importancia es Forward(), que se encargará de recibir los paquetes aceptados para el reenvío por Receive(), preparar la cabecera y reenviarlo. En caso de estar utilizando el algoritmo “self-adapted”, también se encargará de calcular el tiempo de espera Tadapt, y el método AdaptEnvio() se encargará de decidir si el paquete debe ser reenviado o no.

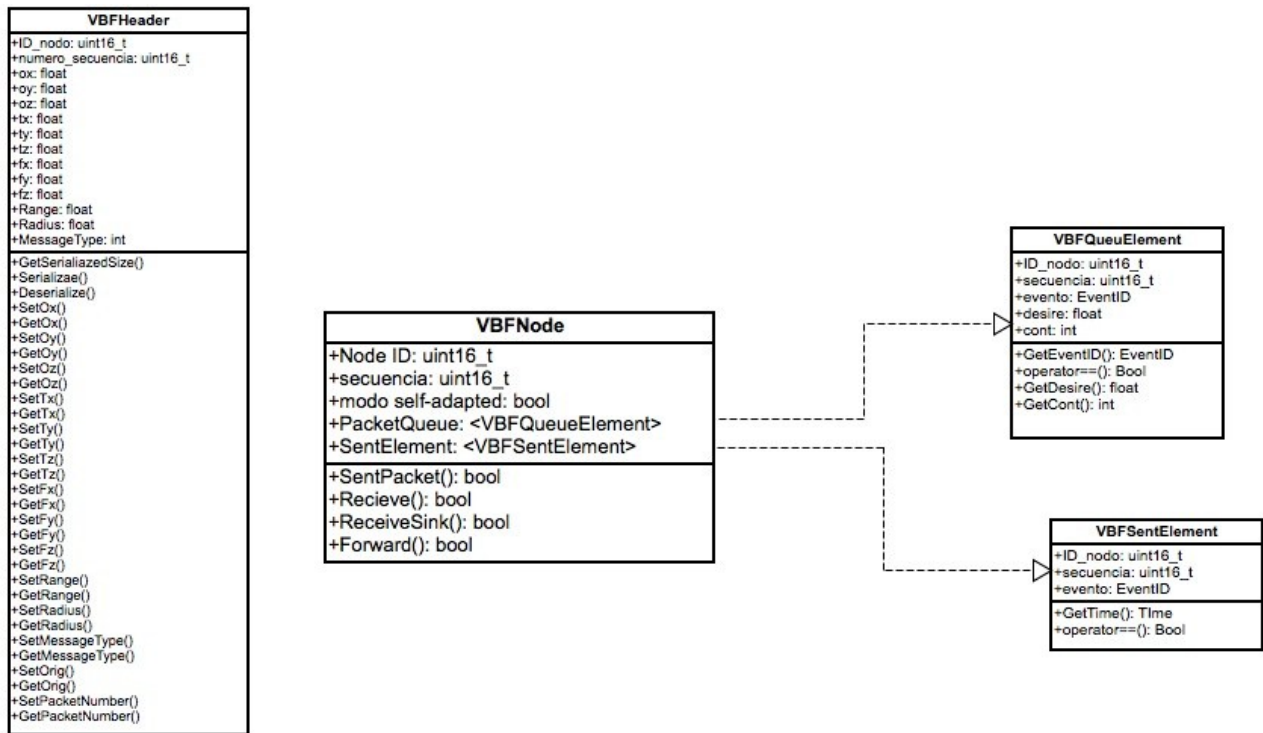


Diagrama 3 - Diagrama de clases

4.3 Descripción de la cabecera

La cabecera que se añade a cada paquete es de vital importancia para el correcto funcionamiento del protocolo. Ha sido implementada mediante la clase VBFHeader, y consta de catorce campos:

- ⤴ ox,oy,oz,tx,ty,tz,fx,fy,fz: Son los campos que indican las coordenadas de el nodo origen, el nodo destino y el último nodo que ha reenviado el paquete. Los campo ox,oy,oz son las coordenadas del nodo origen, tx,ty,tz son las del nodo destino y fx,fy,fz son las del último nodo que ha reenviado el paquete y se codifican mediante un float de 32 bits. Se prefiere un tipo de datos que permita almacenar número reales frente a enteros, aunque podría argumentarse que los sensores disponibles para medir la profundidad no tienen la precisión suficiente para medir distancias inferiores a un metro, por lo que un entero de 16 bits sería suficiente para dicho campo. Pero, sin embargo, se prefiere tomar la decisión de contar con esta capacidad.
- ⤴ Radius: Es el campo que indica el valor umbral para decidir si un nodo está suficientemente cerca de la “routing pipe”. Está codificado como un float de 32 bits, para que haya concordancia con los valores de las coordenadas de los nodos.
- ⤴ Range: Es el campo que indica el límite de la zona para decidir si el paquete ha llegado a su objetivo. Está codificado como un float de 32 bits, para que haya concordancia con los valores de las coordenadas de los nodos.
- ⤴ MessageType: Es el campo que indica el tipo de mensaje que hay en el paquete. Está codificado como un entero de 8 bits porque solo hay 4 tipos de mensaje diferente.
- ⤴ NodeID: Es el identificador del nodo, que se almacena como un entero sin signo de 16 bits. Sirve para identificar de forma única a cada nodo. No es la dirección UAN, aunque cumple una función similar. No se ha utilizado este último tipo, pues se codifica como un entero de 8 bits, lo que reduce el número de posibles nodos a 255, impidiéndonos realizar simulaciones con número mayores. Por ello empleamos 16 bits, lo que nos proporciona más

de 65000 posibles nodos, que estimamos un tamaño suficiente para nuestras simulaciones e hipotéticos despliegues reales. Al instalar el nodo, se le ha de asignar un valor.

- ▲ Sequence number: Es el número de secuencia, para identificar de forma única a cada paquete de un mismo nodos. Hemos utilizado igualmente un entero sin signo de 16 bits. Dependiendo del número máximo de envíos por unidad de tiempo y de la duración de la cache de paquetes enviados, podría reducirse a un entero de 8 bits, pero para evitar posibles inconvenientes elegimos uno de 16. Se inicializa a 0 al arrancar, y se incrementa en uno con cada paquete enviado.

5. Experimentación

5.1 Distribución espacial.

Los nodos se distribuyen en un espacio tridimensional con forma de cubo de 100 metros de lado. En el centro de la superficie se colocará el nodo sumidero. Por convención, la profundidad viene dada por la coordenada Z. De esta forma, Z=100 sería la superficie, 99 correspondería a un metro de profundidad, etc...

A ello se le ha añadido el esquema de movilidad RandomWayMobilityModel, para trazar trayectorias aleatorias dentro de un espacio 3D, lo que nos permite simular el movimiento de los nodos. Con este esquema, el nodo decide un destino aleatoriamente, y se mueve hacia él con la velocidad indicada, cuando llega al destino, después de un tiempo programado de espera, vuelve a decidir otro destino y se dirige hacia él. Existen varios parámetros para configurar la trayectoria que siguen los objetos, pero nosotros nos quedaremos principalmente con el que nos permite establecer el rango de velocidades con que se moverán los nodos.

Todos los nodos excepto el sumidero se distribuirán inicialmente mediante el RandomBoxPositionAllocator, y se les instalará el modelo de movilidad descrito. Cualquier nodo excepto el sumidero puede ser fuente, ya que al enviar un paquete se elige aleatoriamente que nodo lo envía. El nodo sumidero se encontrará, como ya se ha dicho, en el centro de la superficie, en la coordenada (50,50,100).

Así mismo, se realizarán también una serie de simulaciones en un escenario bidimensional de tamaño 100x100, y en este caso el nodo se situará en el centro del plano, es decir en la coordenada (50,50). El resto de nodos serán configurados de la misma manera que en el escenario anterior.

A los nodos se les instalará un módulo de energía infinita con el cuál podremos medir la energía consumida por cada nodo en la simulación. Los consumos de este dispositivo son los siguientes: 0.120 W para transmisión, 0.024 W para recepción y 0.024 W para idle.

5.2 Parámetros utilizados.

Para la configuración de la capa física de la red UAN se utiliza el método estándar proporcionado por el simulador para el cálculo del SINR (ratio señal-interferencia), así como el método estándar de cálculo de errores en los paquetes.

A ello, hay que sumar que se utiliza como modulación FSK (modulación por desplazamiento de frecuencias), con una frecuencia central de 85kHz, un ancho de banda de 1kHz y una constelación de 2 valores. Normalmente se usa a un datarate de 1000 bits/s, aunque realizaremos varias pruebas cambiando este valor para ver el rendimiento del protocolo.

Hay que añadir que la potencia de emisión se sitúa en 690 dB, lo que permitirá un radio de alcance de unos 20 metros. Todo ello se combina junto con un modelo de propagación usando la conocida

aproximación de Thorp (similar al modelo de canal implementado en ns-2) y con un canal adecuado para constituir la capa física de la red.

Para la capa MAC (de control de acceso al medio) vamos a usar Aloha ya que es el más similar al descrito en el paper. Aloha transmite nada más tiene un paquete que enviar, sin importar en que estado se encuentre el medio. El paper describe como MAC un método que escucha el medio antes de enviar, y si lo encuentra libre procede al envío. Si no, se espera un tiempo para volver a intentarlo, y el paquete se descarta si encuentra el medio ocupado un número determinado de veces.

5.3 Resultados

Habiendo ejecutado diversas simulaciones para los parámetros previamente descritos, enviando 1800 paquetes de 64 bytes, uno cada 2 segundos, usando MAC Aloha, un datarate de 1000 bps, obtenemos los siguientes resultados.

Las primeras simulaciones serán comparativas a las propuestas en el paper, para comprobar que la implementación realizada muestra valores similares.

En primer lugar estudiaremos como el movimiento de los nodos influye en el retardo de los paquetes desde el origen hasta el destino. Con un radio de “routing pipe” (el valor w) de 20 metros obtenemos los siguientes resultados:

Velocidad (m/s)	N° Nodos			
	50	100	150	200
0	INF	INF	INF	3,22487
1	1,90619	2,58074	2,92434	3,16626
3	1,69711	2,61817	3,01616	3,07663
5	1,72478	2,5431	2,91865	3,13098

Tabla 1 - Escenario 3D Relación Velocidad-Retardo Medio

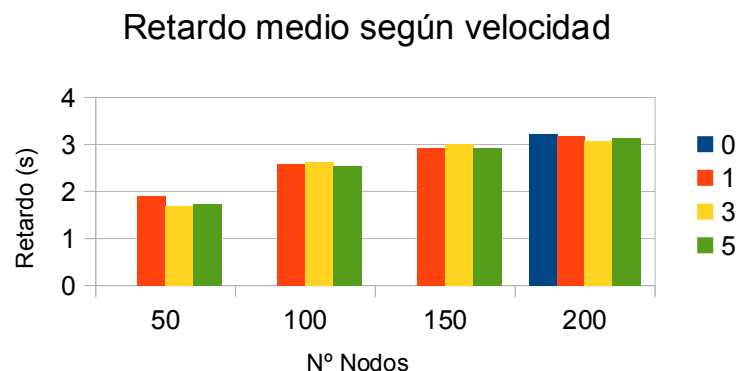


Figura 1 - Escenario 3D Relación Velocidad-Retardo Medio

Puede observarse tanto en la Figura 1 como en la Tabla 1 que el retardo medio apenas varía cuando

el valor de la velocidad de los nodos cambia. También puede observarse como el retardo medio aumenta según va aumentando el número de nodos, esto es debido principalmente al que haber más nodos hay más caminos por los que el mensaje puede ir. En los casos de 50, 100 y 150 nodos, con éstos estáticos no se produjo la entrega de ningún paquete, por lo que no hay valor para el retardo.

Para observar el consumo energético, podemos observar la siguiente gráfica

Velocidad(m/s)	Consumo Total			
	50	100	150	200
0	4527,12	9037,74	13628,7	18426,3
1	4667,15	9554,31	14555,3	19521,7
3	4636,66	9531,69	14551,6	19524,6
5	4637,57	9562,47	14533,4	19493

Tabla 2 – Escenario 3D Relación Velocidad-Consumo Total

Consumo total según velocidad

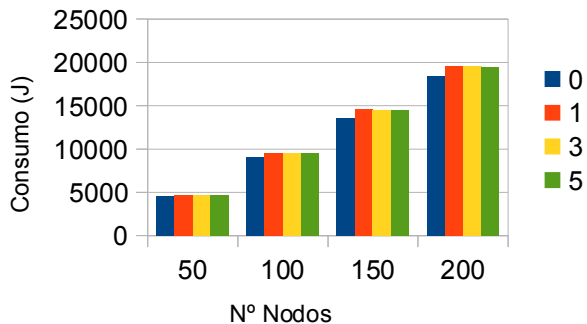


Figura 2

Consumo medio según velocidad

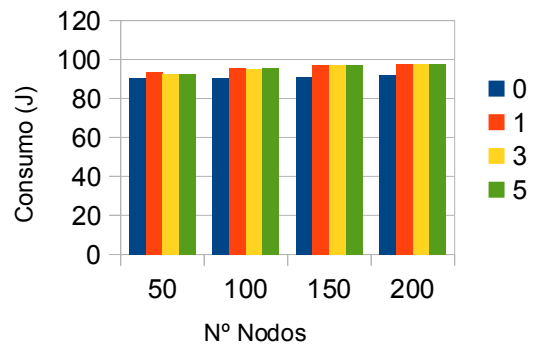


Figura 3

Escenario 3D Relación Velocidad-Consumo Total Escenario 3D Relación Velocidad-Consumo Medio

Se puede observar en la Tabla 2 y en las Figuras 2 y 3 como el consumo aumenta evidentemente con el número de nodos pero como se mantiene estable cuando cambia la velocidad de los nodos. Asimismo, el consumo medio por nodo se mantiene más o menos estable para cualquier número de nodos y para cualquier velocidad, tal como ocurre en el paper.

A continuación vamos a estudiar como influye el valor del radio de la “routing pipe”, el valor w , en la cantidad de paquetes entregados, Se estudiará para una velocidad fija de 3 m/s, con un número variable de nodos. Los resultados son los siguientes.

Nº Nodos	w(m)			
	5	10	15	20
50	32	60	80	104
100	42	120	253	348
150	59	224	490	674
200	88	327	615	784

Tabla 3 – Escenario 3D Relación Tasa de aciertos – Valor de w

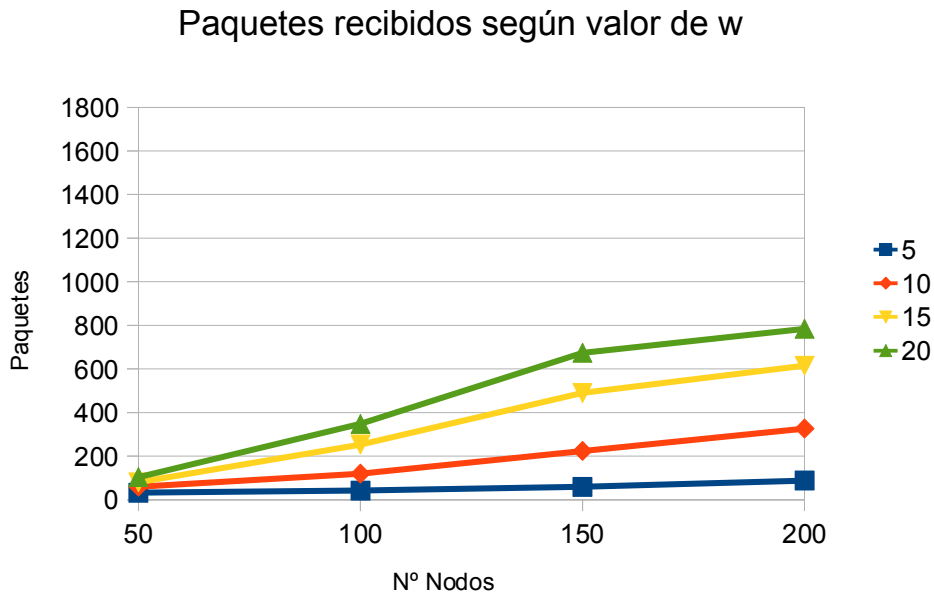


Figura 4 - Escenario 3D Relación Tasa de aciertos – Valor de w

Como se puede observar en la Figura 4 y en la Tabla 3, a mayor tamaño valor de w, más paquetes son recibidos correctamente por el sumidero, y a mayor número de nodos también aumenta la la tasa de aciertos del algoritmo. Las diferencias respecto al paper en el número de paquetes entregados con éxito es debido a la diferencia en la tasa de datos, 1000 bps contra los 500000 bps del paper, un valor completamente irreal si usamos módems de bajo consumo, y también a la diferencia en el número de nodos, las cantidades de nodos propuestas en el paper, un rango entre 500-1500 nodos, son completamente irreales.

El aumento de la tasa de aciertos que se produce al aumentar el valor w implica un aumento del consumo energético, como veremos más adelante.

Influencias de los parámetro del algoritmo.

Estudio de la velocidad de los nodos

Vamos a estudiar cómo cada parámetro configurable del protocolo influye en los resultados de las simulaciones, midiendo principalmente tres factores: el retardo, los paquetes entregados, y el

consumo de energía.

Primero vamos a estudiar como afecta la velocidad de desplazamiento de los nodos en los resultados de las simulaciones. Seguiremos una distribución de velocidades de 0,1,3,5 m/s. Con ella obtenemos:

Nº Nodos	Velocidad	Paquetes Recibidos	Retardo	Consumo	Consumo medio
50	0	0		4527,12	90,5424
100	0	0		9037,74	90,3774
150	0	0		13628,7	90,858
200	0	631	3,22487	18426,3	92,1317
50	1	113	1,90619	4667,15	93,3431
100	1	328	2,58074	9554,31	95,5431
150	1	609	2,92434	14555,3	97,0351
200	1	766	3,16626	19521,7	97,6085
50	3	104	1,69711	4636,66	92,7332
100	3	375	2,61817	9531,69	95,3169
150	3	674	3,01616	14551,6	97,0107
200	3	784	3,07663	19524,6	97,623
50	5	122	1,72478	4637,57	92,7513
100	5	348	2,5431	9562,47	95,6247
150	5	618	2,91865	14533,4	96,8895
200	5	781	3,13098	19493	97,4652

Tabla 4 – Escenario 3D Resultados de simulación variando la velocidad.

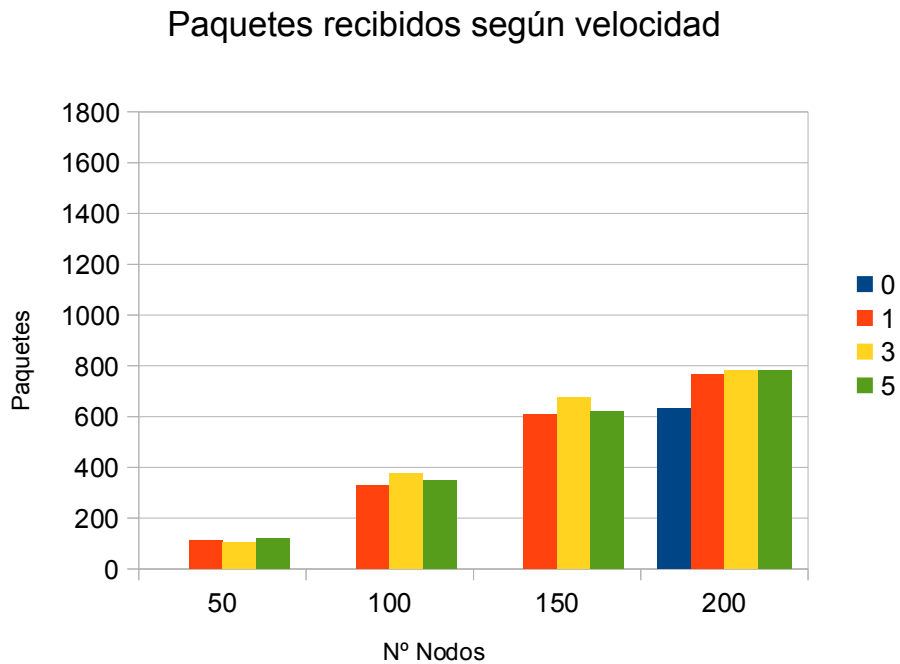


Figura 5 - Escenario 3D Relación Tasa de aciertos – Velocidad.

Retardo medio según velocidad

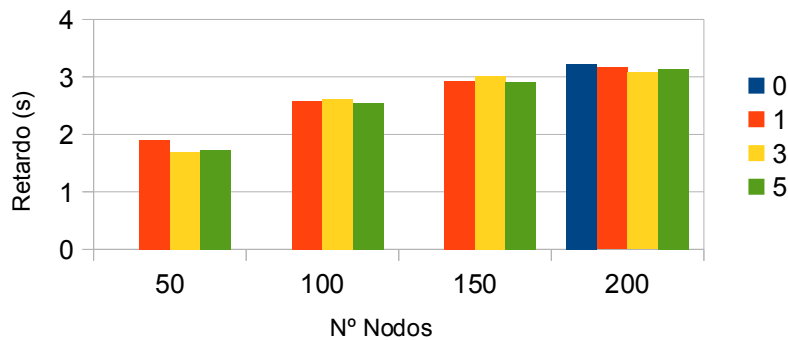


Figura 6 - Escenario 3D Velocidad – Retardo Medio

Consumo total según velocidad

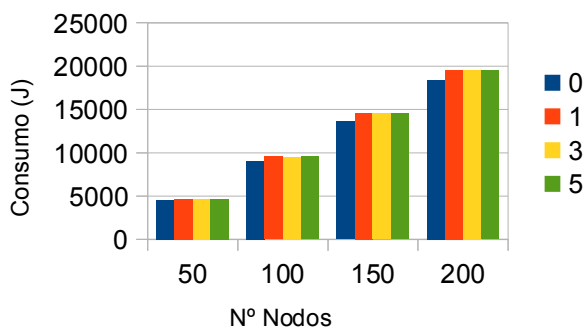


Figura 7

Escenario 3D Relación Velocidad-Consumo Total

Consumo medio según velocidad

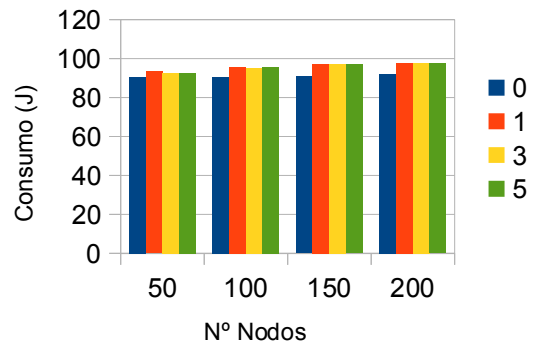


Figura 8

Escenario 3D Relación Velocidad-Consumo Medio

Como se puede observar, la velocidad de desplazamiento de los nodos tiene una influencia prácticamente nula sobre el retardo con el que los paquetes llegan al sumidero. La tasa de aciertos aumenta según aumenta la cantidad de nodos. La tasa de aciertos debería ser mayor cuando los nodos son estáticos, pero debido a la baja densidad de nodos, para los casos de 50, 100 y 150 nodos ningún paquete llega al sumidero. En el caso de cuando hay 200 nodos, la mayor tasa de aciertos se encuentra justamente cuando los nodos son estáticos. De todas maneras, la tasa de aciertos se muestra bastante estable cuando la velocidad de movimiento de los nodos varía.

En la Figura 7 y en la Figura 8 se puede observar como al incrementar el número de nodos el consumo total aumenta, debido a que hay más nodos y hay más retransmisiones totales, pero se también se puede observar como en VBF el consumo medio de cada nodo apenas varía aunque la velocidad de desplazamiento de los nodos cambie.

Los resultados de éstas simulaciones han demostrado que en VBF la velocidad de los nodos tiene poca influencia en el consumo energético y en el retardo. En el caso de la tasa de aciertos, la influencia de la velocidad de movimiento de los nodos depende de la densidad de los nodos. En caso de haber una densidad alta influye poco, para densidades bajas influye más.

Estudio de la variación del valor w

A continuación vamos a estudiar la influencia del valor w o el radio de la “routing pipe”. Los parámetros serán los mismo que en el apartado anterior, pero manteniendo una velocidad constante de 3 m/s.

Los resultados obtenidos son los siguientes:

Nº Nodos	w	Paquetes recibidos	Retardo	Consumo	Consumo medio
50	5	32	0,71642	4348,1	86,962
100	5	42	0,844803	8694,23	86,9423
150	5	59	1,10143	13049,1	86,9937
200	5	88	1,23529	17404,5	87,0227
50	10	60	1,15592	4410,69	88,2138
100	10	120	1,61094	8855,02	88,5502
150	10	224	2,02828	13346	88,9735
200	10	327	2,09896	17824,3	89,1216
50	15	80	1,37255	4516,59	90,3319
100	15	253	2,21569	9150,55	91,5055
150	15	490	2,6247	13886,2	92,5749
200	15	615	2,66436	18589,5	92,9473
50	20	104	1,69711	4636,66	92,7332
100	20	348	2,5431	9562,47	95,6247
150	20	674	3,01616	14551,6	97,0107
200	20	784	3,07663	19524,6	97,623

Tabla 5 – Escenario 3D Resultados de simulación variando el valor de w

Paquetes recibidos según valor de w

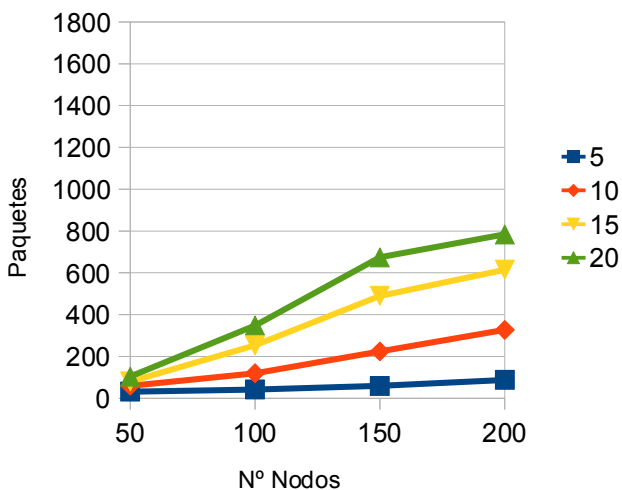


Figura 9

Escenario 3D Relación Tasa de aciertos-Valor de w

Retardo medio según valor de w

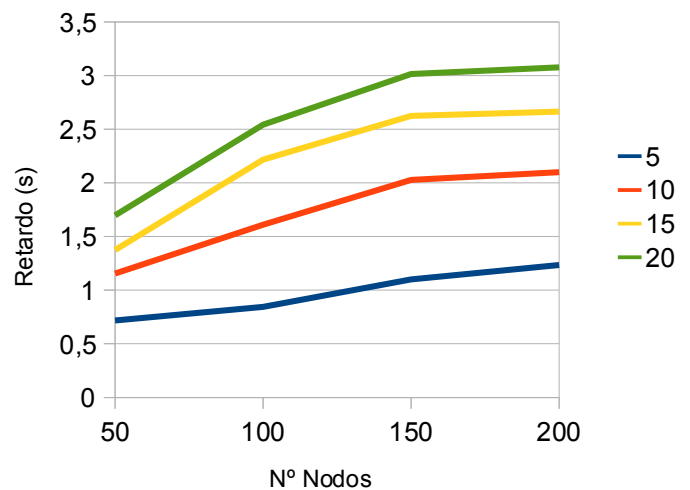


Figura 10

Escenario 3D Retardo Medio-Valor de w

Consumo total según valor de w

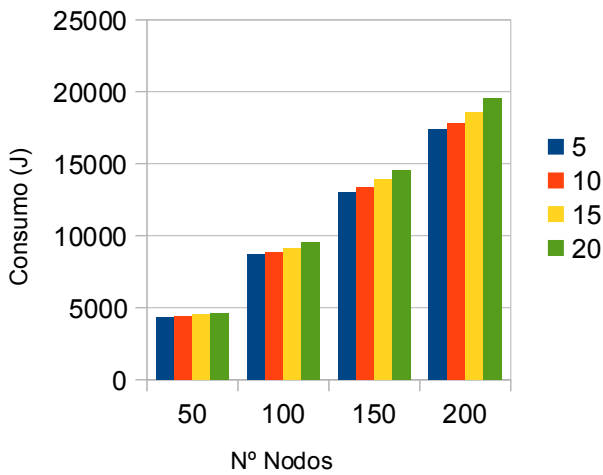


Figura 11

Escenario 3D Relación Consumo Total-Valor de w

Consumo medio según valor

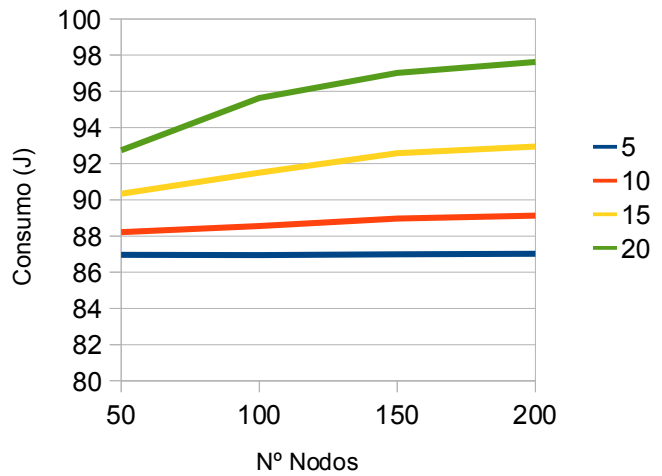


Figura 12

Escenario 3D Relación Consumo Medio-Valor de w

Como se puede observar el valor de w tiene influencia directa sobre la tasa de aciertos de paquetes recibidos, esto es porque cuanto más alto es el valor w más nodos estarán cualificados para reenviar el paquete, por lo que menos probabilidades habrá que el paquete no encuentre el camino hasta el sumidero. Al aumentar el valor de w aumenta el número de nodos implicados en el proceso de envío del paquete por lo que aumenta el retardo de tiempo que tarda el paquete en llegar de la fuente al destino, como se puede ver en la Figura 10.

Considerando el consumo energético, a mayor valor de w más nodos reenvían el paquete por lo que claramente aumenta, como se puede observar en las Figuras 11 y 12. Y también se puede observar que cuanto mayor es la densidad de nodos, el consumo es también mayor.

En conclusión, un alto valor de w nos proporciona una mayor tasa de aciertos, pero a costa de un aumento en el consumo energético, que es justamente una de las mayores prioridades en las redes subacuáticas de sensores.

Estudio de la variación de la tasa de datos.

Ahora vamos a estudiar como afecta al resultado de las simulaciones la variación de la tasa de datos o “datarate”. Los parámetros serán los mismos que en el anterior apartado, pero con un valor de w de 20 m, y la distribución de la tasa de datos será: 100 bps. 500 bps. 1000 bps. Con ello obtendremos:

Nº Nodos	Datarate	Paquetes recibidos	Retardo	Consumo	Consumo medio
50	100	51	12,4247	6308,29	126,166
100	100	102	21,6361	12789,9	127,899
150	100	121	26,2238	19266,7	128,445
200	100	127	30,5202	25720,9	128,604
50	500	99	3,35657	4945,3	98,906
100	500	276	5,08166	10230,5	102,305
150	500	456	6,04511	15633,3	104,222
200	500	536	6,5982	21058,3	105,292
50	1000	104	1,69711	4636,66	92,7332
100	1000	348	2,5431	9562,47	95,6247
150	1000	674	3,01616	14551,6	97,0107
200	1000	784	3,07663	19524,6	97,623

Tabla 6 – Escenario 3D Resultados de simulación variando la tasa de datos.

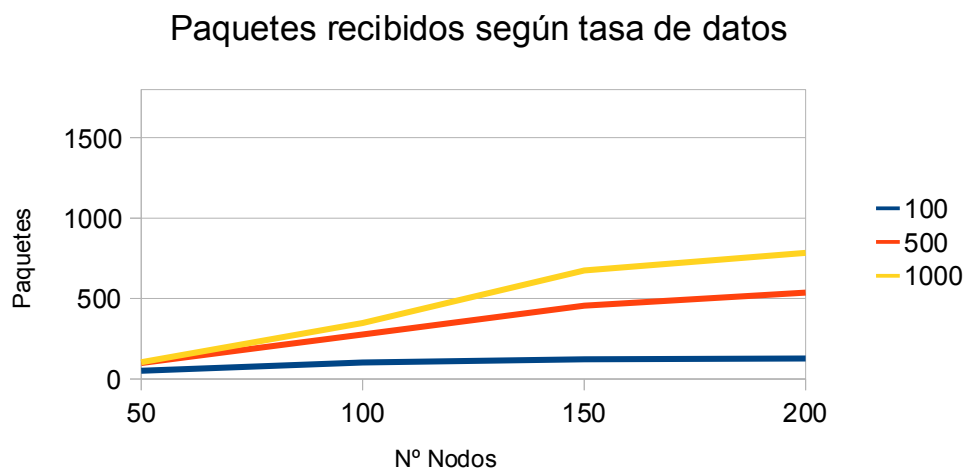


Figura 13 - Escenario 3D Relación Tasa de aciertos–Tasa de datos

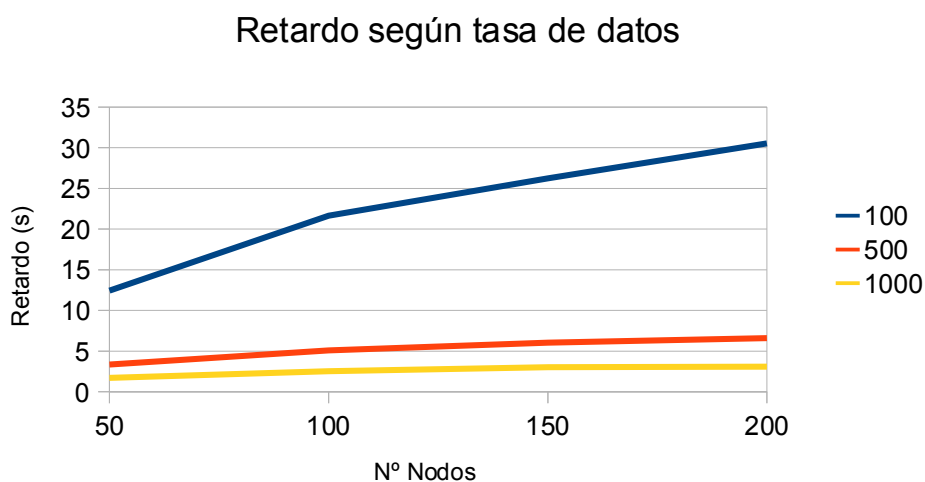


Figura 14 - Escenario 3D Relación Retardo Medio–Tasa de datos

Consumo total según tasa de datos

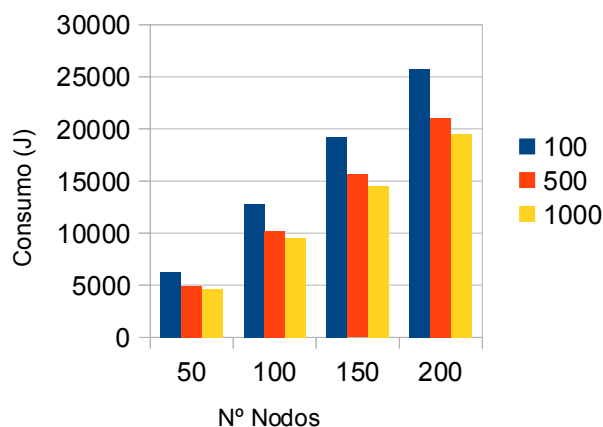


Figura 15

Consumo medio según tasa de datos

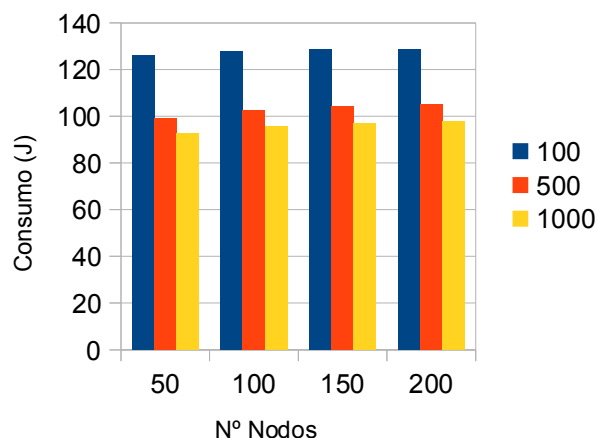


Figura 16

Escenario 3D Relación Consumo Total-Tasa de datos Escenario 3D Relación Consumo Medio-Tasa de datos

Como se puede observar en la Figura 13, y como era de esperar, a una mayor tasa de datos mayor es el índice de aciertos del algoritmo. Asimismo, cuánto menor es la tasa de datos mayor es el retardo, como se observa en la Figura 14, ya que tarda más tiempo en recibir y transmitir los paquetes desde el medio.

Respecto al consumo energético, una tasa de datos menor implica un mayor uso del canal, por lo que se producirán más reenvíos y por lo tanto un aumento del consumo energético, tal como se ve en las Figuras 15 y 16.

Estudio de la variación de la tasa de congestión de datos.

Hasta ahora todas nuestras simulaciones han tenido una tasa de envío de datos de 0.5 paq/s, o 1 paquete cada 2 segundos. A continuación realizaremos una serie de simulaciones para comprobar que efecto produce variar este parámetro. Los parámetros de la simulación son los mismos que para los casos anteriores, con una tasa de datos de 1000 bps, y con una distribución de la tasa de envío de 2,1,0.5,03,0.25 paquetes por segundo. Los resultados obtenidos son los siguientes:

Nº Nodos	Paq/s	Tasa de aciertos
50	2	0,05
100	2	0,1
150	2	0,15
200	2	0,172
50	1	0,053
100	1	0,151
150	1	0,2375
200	1	0,283
50	0,5	0,05
100	0,5	0,17
150	0,5	0,33
200	0,5	0,457
50	0,3	0,06
100	0,3	0,23
150	0,3	0,405
200	0,3	0,51
50	0,25	0,06
100	0,25	0,25
150	0,25	0,408
200	0,25	0,54

Tabla 7 – Escenario 3D Resultados simulación variando la tasa de congestión de datos.

Tasa de aciertos según paq/s

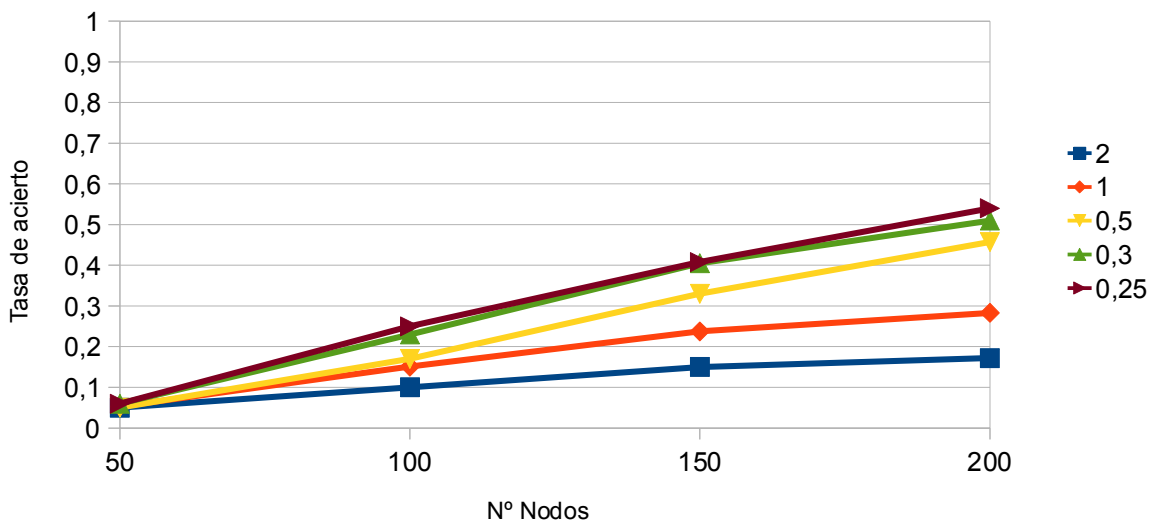


Figura 17 - Escenario 3D Relación Tasa de aciertos–Tasa de congestión de datos.

Como se puede observar en la figura 17, para una tasa de envíos de paquetes de 2 paquetes/s, la tasa de aciertos no llega al 20% en el mejor de los casos. De la misma manera, en el mejor de los casos como mucho se produce una tasa de acierto alrededor del 50 %. Esto es debido a la poca densidad de nodos y al valor de la tasa de datos, que claramente actúan como cuellos de botella y limitan el rendimiento del algoritmo.

Estudio del efecto del uso del algoritmo Self-Adapted.

A continuación vamos a estudiar el efecto que produce el usar el algoritmo Self-Adapted, que forma parte de VBF, cuya función es reducir el número de reenvíos de un paquete y por lo tanto de ahorrar energía.

Para las simulaciones se han tomado como valores un “datarate” de 1000 bps, un valor de w o de radio de la “routing pipe” de 20 m y una velocidad de movimiento de los nodos con una distribución de 0,1,3,5 m/s. Los resultados obtenidos son los siguientes:

Nº Nodos	Velocidad	Paquetes Recibidos	Retardo	Consumo	Consumo medio
50	0	0		4501,9	90,038
50	1	103	1,8327	4587,4	91,748
50	3	90	1,70196	4553,37	91,0674
50	5	97	1,65739	4541,21	90,8242
100	0	0		8971,15	89,7115
100	1	315	2,58124	9329,81	93,2981
100	3	310	2,66198	9264,79	92,6479
100	5	303	2,52217	9273,36	92,7336
150	0	0		13502,3	90,0156
150	1	568	3,03725	14151,1	94,3409
150	3	599	3,02571	14113,2	94,0883
150	5	538	2,94741	14045,2	93,6347
200	0	652	3,44422	18196	90,9799
200	1	758	3,26301	18953,4	94,767
200	3	732	3,15684	18875,8	94,3792
200	5	700	3,23414	18829,3	94,1467

Tabla 8 – Escenario 3D Resultados simulación usando algoritmo Self-Adapted

Ahora, pasaremos a comparar estos resultados con los correspondientes obtenidos sin usar el algoritmo “self-adaption”. Para esta comparación usaremos los resultados obtenidos al realizar simulaciones con un valor de la velocidad de desplazamiento de nodos de 3 m/s. Los resultados obtenidos son:

Nº Nodos	Normal Retardo	Self-Adapted Retardo
50	1,69711	1,70196
100	2,61817	2,66198
150	3,01616	3,02571
200	3,07663	3,15684

Tabla 9

Escenario 3D Tasa de aciertos Normal-Self-Adapted

Nº Nodos	Normal Paquetes	Self-Adapted Paquetes
50	104	90
100	375	310
150	674	599
200	784	732

Tabla 10

Escenario 3D Retardo Medio Normal-Self-Adapted

Nº Nodos	Normal Consumo Total	Self-Adapted Consumo Total
50	4636,66	4553,37
100	9531,69	9264,79
150	14551,6	14113,2
200	19524,6	18875,8

Tabla 11

Escenario 3D Consumo Total Normal-Self-Adapted

Nº Nodos	Normal Consumo medio	Self-Adapted Consumo medio
50	92,7332	91,0674
100	95,3169	92,6479
150	97,0107	94,0883
200	97,623	94,3792

Tabla 12

Escenario 3D Consumo Medio Normal-Self-Adapted

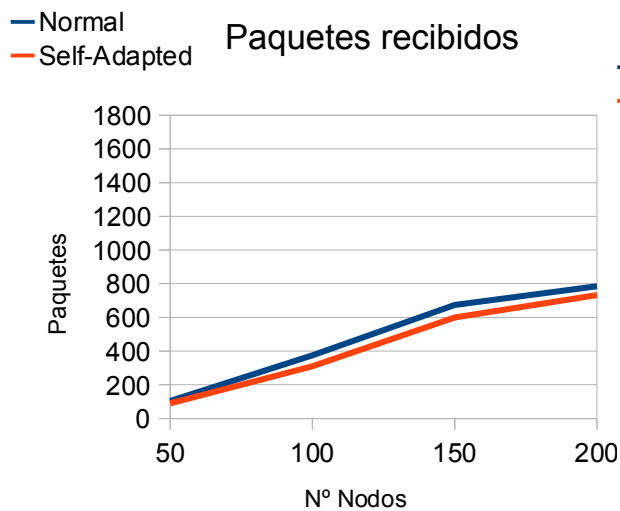


Figura 18

Escenario 3D Tasa de aciertos Normal-Self-Adapted

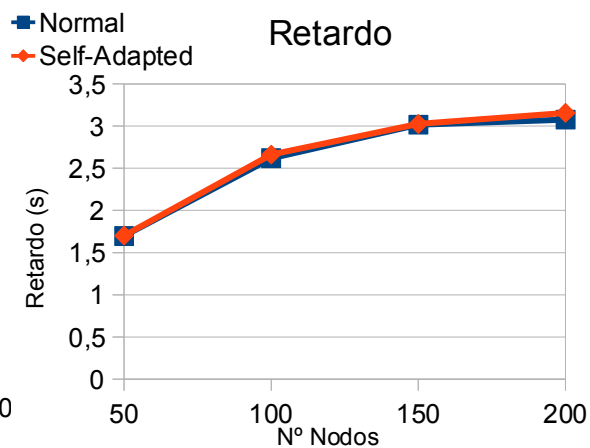


Figura 19

Escenario 3D Retardo Medio Normal-Self-Adapted

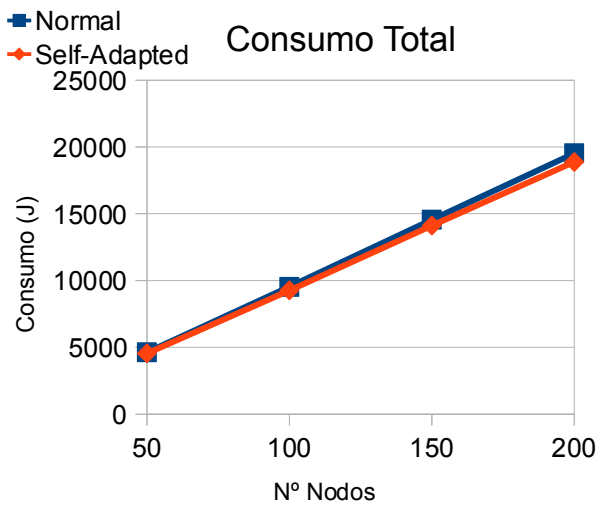


Figura 20

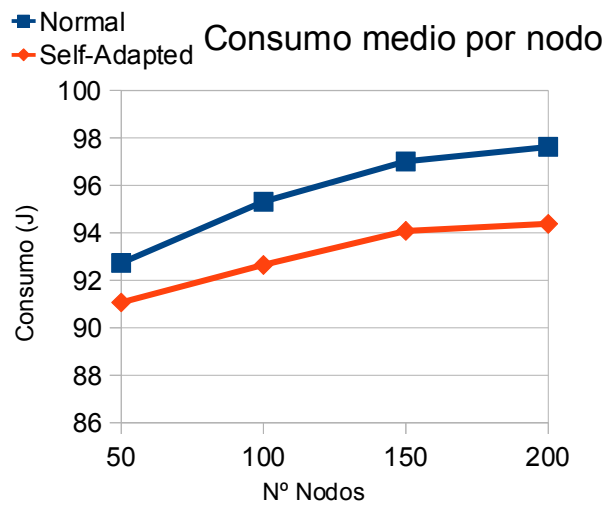


Figura 21

Escenario 3D Consumo Total Normal-Self-Adapted

Escenario 3D Consumo Medio Normal-Self-Adapted

Como se puede observar en la Figura 18, VBF con el algoritmo “self-adapted” tiene una tasa de acierto un poco menor que sin él, esto es debido a que se reenvía menos veces el paquete, pero la diferencia no es demasiado grande.

Debido a que en el algoritmo “self-adapted” cada nodo tiene que esperar un tiempo de adaptación para decidir si tiene que reenviar o no el paquete. El retardo es un poco mayor usando “self-adapted”, como se puede apreciar en la Figura 19. Debido a que el algoritmo “self-adapted” está ideado para usarse en casos de una gran densidad de nodos, el cuál no es nuestro caso, la diferencia no es demasiado grande.

Respecto al consumo energético, debido a que hay menos reenvíos el consumo energético es menor cuando usamos el algoritmo “self-adapted”, como se puede apreciar en las Figuras 20 y 21. Igual que con el retardo, la diferencia es mínima debida a la baja densidad de nodos.

Escenario bidimensional.

A continuación vamos a estudiar los resultados obtenidos al realizar una serie de simulaciones sobre un escenario bidimensional. Se procederá a observar como afectan a los resultados la variación de las mismas variables estudiadas para el escenario tridimensional.

Primero, procederemos a estudiar el impacto del valor de la velocidad de movimiento de los nodos en las simulaciones. Habiendo enviado 1800 paquetes, 1 cada 2 segundos, de 64 Bytes cada uno, usando MAC Aloha, un datarate de 1000 bps y una distribución de velocidades de 0,1,3,5 m/s, la misma configuración que para el escenario tridimensional, los resultados obtenidos son:

Paquetes Recibidos según velocidad

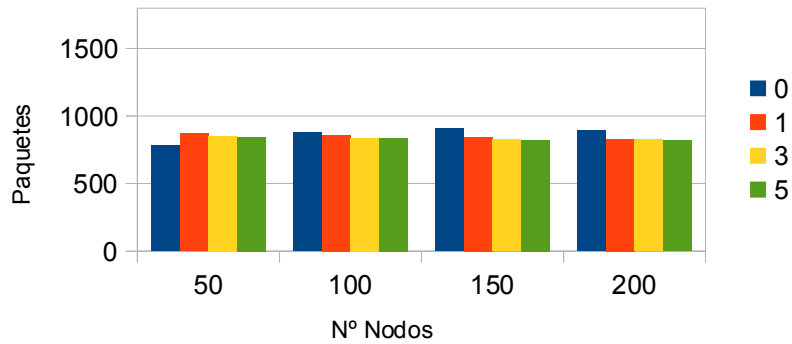


Figura 22 - Escenario 2D Velocidad – Tasa de aciertos

Retardo medio según velocidad

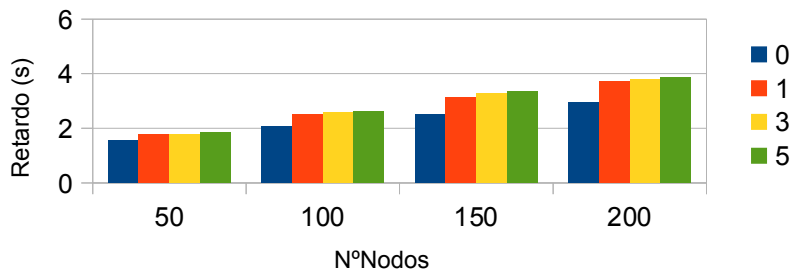


Figura 23 - Escenario 2D Velocidad – Retardo Medio

Consumo total según velocidad

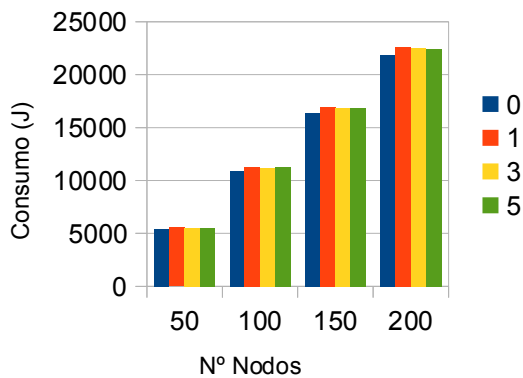


Figura 24

Consumo medio según velocidad

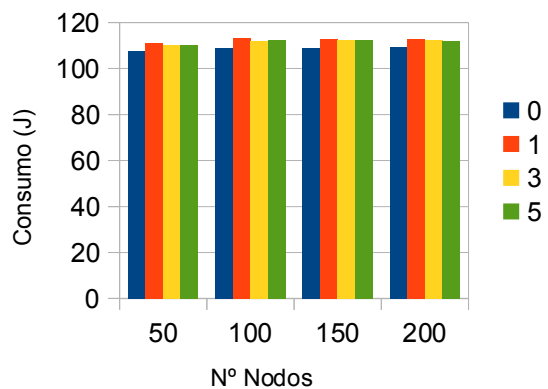


Figura 25

Escenario 2D Relación Velocidad-Consumo Total Escenario 2D Relación Velocidad-Consumo Medio

Como se puede observar en las Figuras 22, 23, 24 y 25, la velocidad de movimiento de los nodos no tiene demasiada influencia en el valor de la tasa de aciertos, tal como pasaba en el escenario tridimensional. Igualmente, la velocidad de movimiento de los nodos tiene poca influencia en el valor del retardo medio y el consumo.

A continuación procederemos a estudiar como afecta la variación del valor de w o del radio de la “routing pipe”. Para un escenario configurado de la misma manera que en el caso anterior, pero con un valor para la velocidad de los nodos de 3 m/s y una distribución de w de 5,0,15,20 m, los resultados obtenidos son:

Nº Nodos	w	Paquetes recibidos	Retardo	Consumo Total	Consumo medio
50	5	692	1,12832	4604,74	92,0948
100	5	814	1,60606	9344,51	93,4451
150	5	818	1,90609	14092,9	93,9523
200	5	809	2,19719	18801	94,0052
50	10	793	1,40496	4949	98,9799
100	10	849	2,11475	10077	100,77
150	10	832	2,50059	15213,9	101,426
200	10	830	2,97789	20308	101,54
50	15	837	1,63411	5247,87	104,957
100	15	850	2,36755	10718	107,18
150	15	836	3,02091	16128,2	107,522
200	15	830	3,51342	21501	107,505
50	20	852	1,75854	5513,65	110,273
100	20	839	2,5909	11212,1	112,121
150	20	827	3,28844	16881,9	112,546
200	20	824	3,78151	22489,8	112,449

Tabla 13 – Escenario 2D Resultados de simulación variando el valor de w

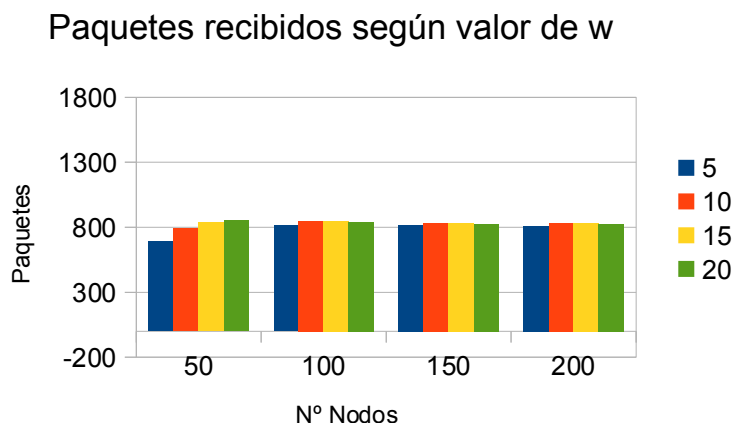


Figura 26 - Escenario 2D Tasa de aciertos–Velocidad

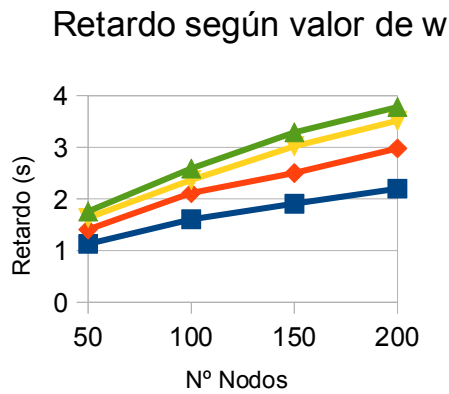


Figura 27

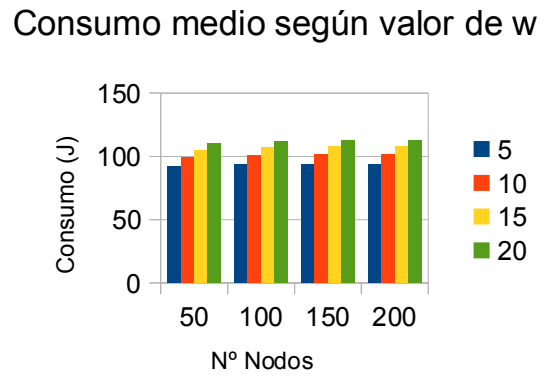


Figura 28

Escenario 2D Relación Valor de w-Retardo Medio Escenario 2D Relación Valor de w-Consumo Medio

Como se puede observar en la Figura 26, a diferencia de en el escenario tridimensional, la tasa de aciertos es similar para cualquier valor de w , en vez de ir aumentando según va creciendo el valor de w . Esto se debe a que hay el atributo w tiene un valor umbral al partir del cuál la tasa de aciertos del algoritmo VBF no mejora, y al haber una mayor densidad de nodos en este escenario este valor es más pequeño que en el escenario tridimensional.

También se puede observar, en las Figuras 27 y 28, como el retardo y el consumo energético si que aumentan según va aumentando el valor de w , igual que en el escenario tridimensional. Esto es debido a que cuanto mayor sea el valor de w más nodos forman parte del envío de un paquete.

Seguidamente vamos a estudiar como afecta la variación del datarate o tasa de datos. Para una configuración igual a la de los casos anteriores, con un valor de w de 20 m y una distribución para el datarate de 100, 500, 1000 bps, los datos obtenidos son:

Nº Nodos	Datarate	Paquetes Recibidos	Retardo	Consumo Total	Consumo medio
50	100	150	16,2046	7353,44	147,069
100	100	137	24,2389	14318,2	143,182
150	100	135	30,0369	21188,2	141,255
200	100	130	36,5536	27982	139,91
50	500	535	3,7413	5998,49	119,97
100	500	495	5,69562	12084,6	120,846
150	500	449	7,2496	17972,8	119,819
200	500	445	8,53902	23868,6	119,343
50	1000	852	1,75854	5513,65	110,273
100	1000	839	2,5909	11212,1	112,121
150	1000	827	3,28844	16881,9	112,546
200	1000	824	3,78151	22489,8	112,449

Tabla 14 - Escenario 2D Resultados de simulación variando la tasa de datos.

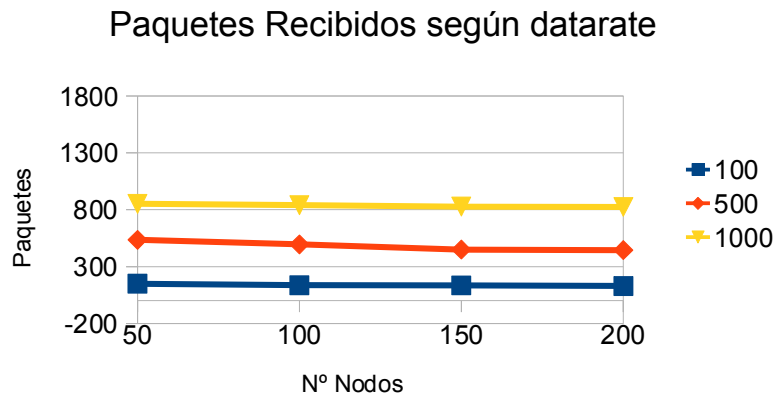


Figura 29 - Escenario 2D Relación Tasa de aciertos–Tasa de datos

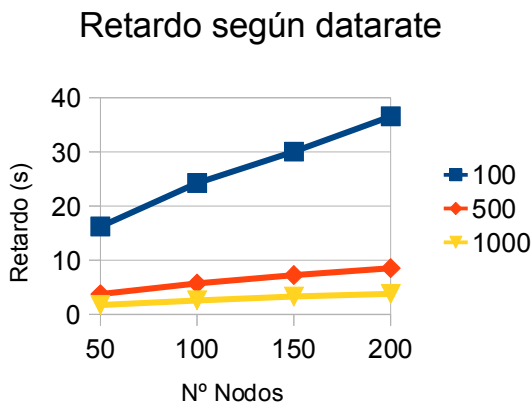


Figura 30

Escenario 2D Relación Consumo Total-Tasa de datos

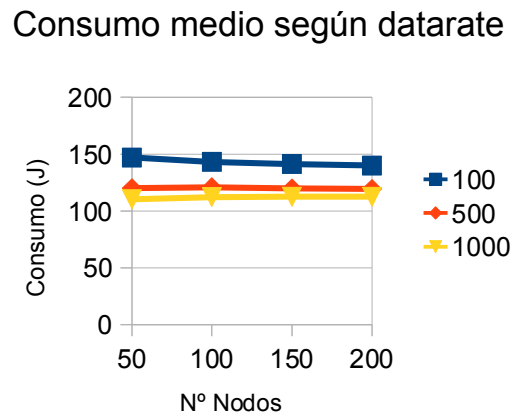


Figura 31

Escenario 2D Relación Consumo Medio-Tasa de datos

Como se puede observar en las Figuras 29, 30 y 31, los resultados obtenidos son similares a los obtenidos para el escenario tridimensional. La tasa de acierto de los paquetes recibidos es mayor cuanto mayor es el datarate, como era de esperar, y el retardo y el consumo energético es inversamente proporcional al valor de la tasa de datos.

A continuación vamos a observar como afecta al resultado de las simulaciones la variación de la tasa de envío de paquetes. Para una configuración igual a la de las simulaciones anteriores y un datarate de 1000 bps. Con una distribución de la tasa de envíos de 2,1,0.5,03,0.25 paquetes por segundo, los resultados son los siguientes:

Tasa de aciertos según tasa de envío

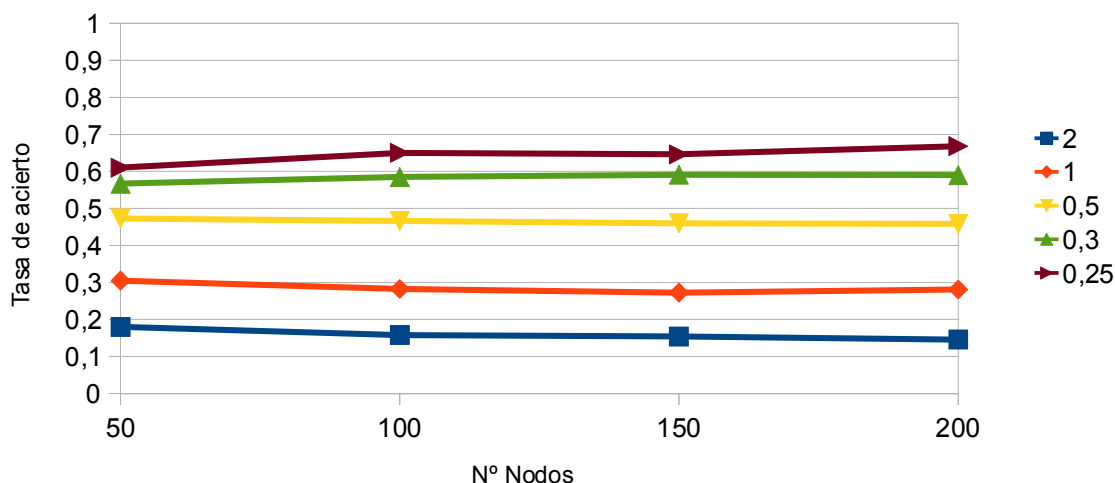


Figura 32 - Escenario 2D Relación Tasa de aciertos–Tasa de congestión de datos.

Como se puede observar en la Figura 32, en comparación con los resultados obtenidos en el escenario tridimensional, se obtienen unas tasas de acierto mejores cuando hay pocos nodos. Esto es debido a que en este escenario la densidad de nodos es mayor. Igual que en el escenario tridimensional, parece que llega un momento en el que aunque sigamos espaciando el tiempo entre envíos de paquetes la tasa de aciertos apenas crece. Como ya se comentó, esto es debido a la configuración de ciertos parámetros, como el datarate, que limitan la efectividad del algoritmo VBF.

Finalmente vamos a estudiar el efecto del algoritmo “self-adapted” tiene en este escenario, Al haber una mayor densidad de nodos el efecto debería ser mayor que en el escenario tridimensional.

Para las simulaciones se han tomado los mismos valores que los tomados en el escenario tridimensional, es decir: un “datarate” de 1000 bps, un valor de w o de radio de la “routing pipe” de 20 m y una velocidad de movimiento fija para los nodos de 3 m/s. Los resultados obtenidos, comparándolos con los equivalentes sin estar usando el algoritmo “self-adapted” son los siguientes:

Nº Nodos	Normal Paquetes	Self-Adapted Paquetes
50	852	826
100	839	826
150	827	788
200	824	734

Tabla 15

Nº Nodos	Normal Retardo	Self-Adapted Retardo
50	1,61717	1,75854
100	2,44506	2,5909
150	2,91886	3,28844
200	3,56823	3,78151

Tabla 16

Escenario 2D Tasa de aciertos Normal-Self-Adapted Escenario 2D Retardo Medio Normal-Self-Adapted

Nº Nodos	Normal Consumo Total	Self-Adapted Consumo Total
50	5513,65	5242,02
100	11212,1	10683,4
150	16881,9	16097,2
200	22489,8	21336,4

Tabla 17

Nº Nodos	Normal Consumo Medio	Self-Adapted Consumo Medio
50	110,273	104,84
100	112,121	106,834
150	112,546	107,315
200	112,449	106,682

Tabla 18

Escenario 2D Consumo Total Normal-Self-Adapted Escenario 2D Consumo Medio Normal-Self-Adapted

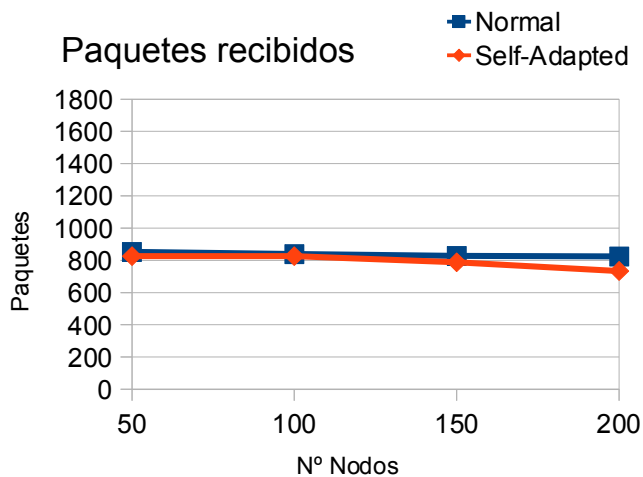


Figura 33

Escenario 2D Tasa de aciertos Normal-Self-Adapted

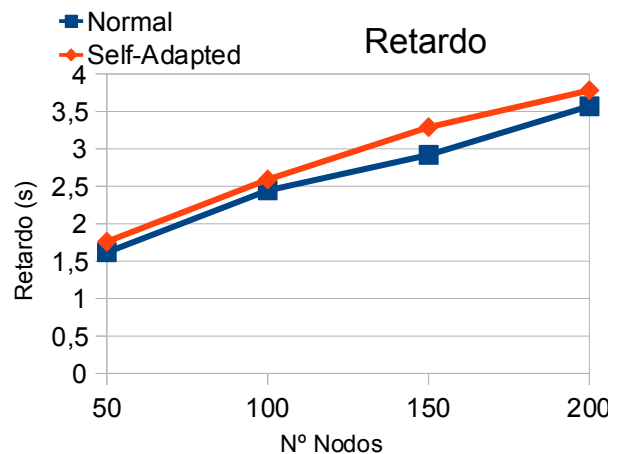


Figura 34

Escenario 2D Retardo Medio Normal-Self-Adapted

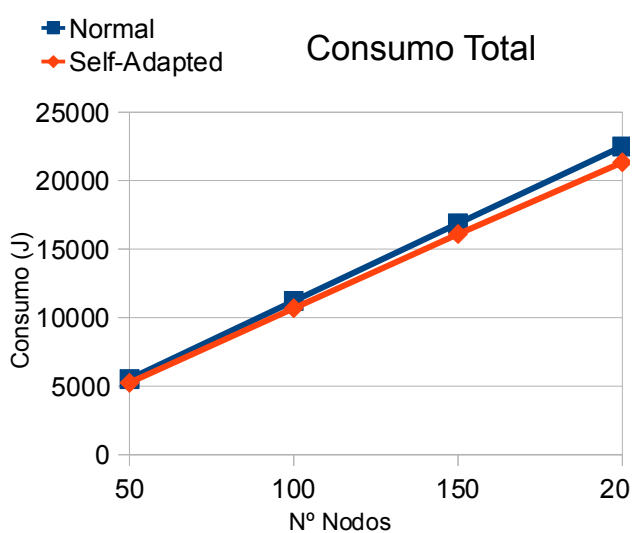


Figura 35

Escenario 2D Consumo Total Normal-Self-Adapted

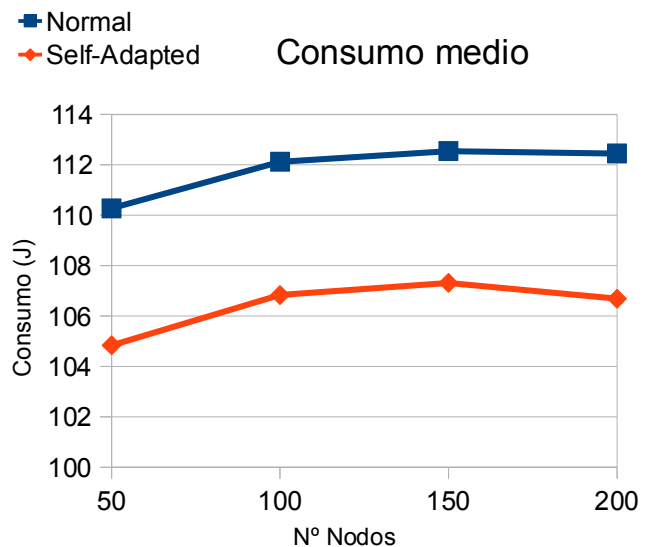


Figura 36

Escenario 2D Consumo Medio Normal-Self-Adapted

Se puede observar en las Figuras 33, 34, 35 y 36 como los resultados obtenidos son muy similares a los obtenidos en el escenario tridimensional. Como se suponía, hay una tasa de aciertos ligeramente menor y un retardo mayor, pero a cambio el consumo es menor usando el algoritmo “self adapted”. Se puede observar en la Figura 34 como hay una mayor diferencia en el consumo de cada nodo que en el escenario tridimensional.

6. Conclusiones y trabajo futuro.

Se ha realizado la implementación de un modelo del protocolo de enrutamiento VBF para redes de sensores subacuáticas para el simulador NS-3 y se ha realizado un estudio sobre este modelo sobre el cuál se han extraído las siguientes conclusiones.

En el estudio del protocolo VBF hemos llegado a confirmar los resultados presentados en el artículo de referencia.

En primer lugar, hemos comprobado que el protocolo VBF que la velocidad de movimiento de los nodos tiene poco impacto en la tasa de aciertos, el consumo energético y el retardo medio. Por lo tanto VBF puede manejar efectivamente la movilidad de los nodos.

También hemos comprobado como afecta la variación de uno de los parámetros del algoritmo, el radio de la "routing pipe" o w . Hemos demostrado que a mayor valor de w un mayor número de nodos se verán inmiscuidos en el envío de un paquete, pero a costa de un mayor retardo y un mayor consumo energético. También hemos descubierto como llega un momento en que aumentar la w no nos asegura una mejor tasa de aciertos, pero si que sigue aumentando el consumo energético. Debido a que nuestra mayor preocupación es el consumo energético, a la hora de configurar el algoritmo hay que encontrar un valor que nos de una buena tasa de aciertos con el mínimo consumo posible.

También se ha estudiado el efecto del algoritmo "self-adapted", y se ha demostrado que a costa de aumentar un poco el retardo medio se consigue mejorar el consumo energético. Debido a que el consumo energético es nuestra prioridad a la hora de seleccionar los parámetros del protocolo, se recomienda siempre usar el algoritmo "self-adapted".

Como trabajo futuro se propone la creación de un protocolo MAC eficiente para las redes de sensores subacuáticas. En las simulaciones hemos usado como protocolo MAC el Aloha, que es el más parecido al protocolo MAC definido en el artículo de referencia, pero se considera necesaria la creación de un protocolo MAC eficiente que se ajuste a las necesidades de las redes subacuáticas de sensores para dar un paso más en las investigaciones.

7. Bibliografía.

- [1] P. Xie, J.-H. Cui, and L. Lao. Vbf: Vector-based forwarding protocol for underwater sensor networks.
- [2] J. Proakis, J. Rice, E. Sozer, and M. Stojanovic. Shallow Water Acoustic Networks. In J. G. Proakis, editor, *Encyclopedia of Telecommunications*. John Wiley and Sons, 2003.
- [3] K. Kredo II and P. Mohapatra. Medium Access Control in Wireless Sensor Networks. *Computer Networks (Elsevier)*, vol. 51(4), pp. 961–994, Mar. 2007.
- [4] S. Kumar, V. S. Raghavanb, and J. Dengc. Medium Access Control Protocols for Ad Hoc Wireless Networks: A Survey. *Ad Hoc Networks (Elsevier)*, vol. 4(3), pp. 326–358, May 2006.
- [5] C. Perkins and P. Bhagwat. Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers. In *Proc. of ACM Special Interest Group on Data Communications (SIGCOMM)*. London, UK, 1994.
- [6] D. B. Johnson, D. A. Maltz, and J. Broch. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. In C. E. Perkins, editor, *Ad Hoc Networking*, pp. 139–172. Addison-Wesley, 2001.
- [7] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with Guaranteed Delivery in Ad Hoc Wireless Networks. *ACM-Kluwer Wireless Networks (Springer)*, vol. 7(6), pp. 609–616, Nov. 2001.
- [8] P. Xie and J.-H. Cui. SDRT: A Reliable Data Transport Protocol for Underwater Sensor Networks. University of Connecticut, Technical Report UbiNet-TR06-03, Feb. 2006.
- [9] I. F. Akyildiz D. Pompili, and T. Melodia State of the Art in Protocol Research for Underwater Acoustic Sensor Networks
- [10] Homepage of ns-3: <http://www.nsnam.org/>