

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

MEJORAMIENTO DE UN PROTOCOLO DE SINCRONIZACIÓN EN REDES WSN

**TESIS DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE MAGÍSTER EN
CONECTIVIDAD Y REDES DE TELECOMUNICACIONES**

ALBÁN LEÓN FERNANDO XAVIER

DIRECTOR: GONZÁLEZ GONZÁLEZ FABIO MATÍAS

Quito, febrero 2019

AVAL

Certifico que el presente trabajo fue desarrollado por Fernando Xavier Albán León, bajo mi supervisión.

FABIO MATÍAS GONZÁLEZ GONZÁLEZ

DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Fernando Xavier Albán León, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración, cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

FERNANDO XAVIER ALBÁN LEÓN

DEDICATORIA

A mi amada Madre y amiga María Virginia León Abendaño, por su amor, paciencia, cariño y ternura, por sus sueños y desvelos, por su dedicación, apoyo y entrega en todos los instantes de mi vida.

AGRADECIMIENTO

A Dios por su amor, compañía, bondad e innumerables bendiciones.

A mi Director MSc. Fabio Matías González González, por su valiosa colaboración en el planteamiento, desarrollo y culminación de esta tesis de grado

A los Doctores Jorge Carvajal y Robín Álvarez por sus contribuciones, ideas y tiempo.

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO	V
ÍNDICE DE TABLAS	VII
ÍNDICE DE ECUACIONES	VII
ÍNDICE DE FIGURAS	VII
RESUMEN	IX
ABSTRACT	X
1 INTRODUCCIÓN	1
1.1 Pregunta de investigación	2
1.2 Objetivo General	2
1.3 Objetivos Específicos	2
1.4 Alcance	2
1.5 Marco Teórico	3
1.5.1 Tipos de sincronización	3
1.5.2 Protocolos basados en la sincronización emisor /receptor	8
1.5.3 Protocolos basados en la sincronización receptor /receptor	13
1.5.4 Análisis del funcionamiento del protocolo HRTS	20
1.5.5 Pull: ITR Protocolo Individual basado en la solicitud de tiempo	25
1.5.6 Protocolo HRTS Modificado	26
1.5.7 Métodos para la Optimización de Algoritmos de Telecomunicaciones	27
2 METODOLOGÍA	38

Implementación de la Metodología	39
2.1 Simulación del Protocolo HRTS original.....	39
2.1.1 Análisis	39
2.1.2 Diseño	40
2.1.3 Pruebas	40
2.2 Simulación del Protocolo HRTS mejorado	40
2.2.1 Análisis	40
2.3 Programas, funciones y scripts utilizados	41
3 RESULTADOS Y DISCUSIÓN	56
4 CONCLUSIONES	71
5 REFERENCIAS BIBLIOGRÁFICAS [1].....	73
6 ANEXOS.....	75
6.1 ANEXO I.....	76
7 ORDEN DE EMPASTADO	100

ÍNDICE DE TABLAS

Tabla 1.1 Etapas de funcionamiento TPSN	12
Tabla 1.2 Condiciones de sincronización y asignación de nivel	19
Tabla 2.1 Posición (numeración) espacial de los nodos y ubicación ascendente en función de la suma de las distancias	42
Tabla 2.2 Distribución Espectral de niveles	47
Tabla 3.1 <i>HRTS Distribución Volumétrica</i>	58
Tabla 3.2 <i>HRTS Distribución Espectral</i>	61
Tabla 3.3 Comparación del desempeño del Protocolo Original con los Protocolos mejorados de Distribución Volumétrica y la Distribución Espectral	64
Tabla 3.4 Distribución volumétrica (Nodo 79) y Distribución espectral (Nodo 46)	66
Tabla 3.5 Camino de acuerdo con la ubicación de los nodos por el que recorren los mensajes de cada uno de los nodos hacia el nodo de referencia	68

ÍNDICE DE ECUACIONES

Ecuación 1.1 Relación entre el tiempo real y el tiempo del reloj local i	3
Ecuación 1.2 Condición para sincronización externa	3
Ecuación 1.3 Condición para sincronización interna	3
Ecuación 1.4 Condición para tasa de deriva máxima	5
Ecuación 1.5 Marca de tiempo en t_4	11
Ecuación 1.6 <i>Marca de tiempo en t_7</i>	12
Ecuación 1.7 Diferencias de marcas de tiempo	12
Ecuaciones 1.8 Relación entre tiempos, retrasos de propagación y compensación de tiempo de los relojes locales	21
Ecuación 1.9 Transformación proyectiva	29
Ecuación 1.10 Secuencia de problema con la técnica de barrera logarítmica	29
Ecuación 1.11 Función Lagrangiana	32

ÍNDICE DE FIGURAS

Figura 1.1 Funcionamiento de LTS	9
Figura 1.2 Cómo funciona la sincronización entre dos nodos	13
Figura 1.3 LTS Sincronización por pares	17
Figura 1.4 Funcionamiento de la sincronización entre dos nodos en HRTS	18
Figura 1.5 Sincronización HRTS	21

Figura 1.6 Camino seguido para alcanzar el punto óptimo x^* desde el punto inicial x_0 por a) el método simple; b) un método de punto interior	28
Figura 1.7 Algoritmo de la ruta más corta.....	37
Figura 2.1 Diagrama de flujo de la función distancia	45
Figura 2.2 Diagrama de Flujo de la Simulación de la Distribución Volumétrica	46
Figura 2.3 Toda la red, con todos los nodos	50
Figura 2.4 Distintos nodos de referencia con su respectiva Distribución Espectral	50
Figura 2.5 Distintos nodos de referencia con su respectiva Distribución Espectral	51
Figura 2.6 Diagrama de flujo de la Distribución Espectral	52
Figura 2.7 Diagrama de Flujo de la Simulación del Protocolo HRTS.....	54
Figura 3.1 Distribución volumétrica	56
Figura 3.2 Distribución Espectral.....	57
Figura 3.3 Los tres niveles: nodo de referencia (negro) 1) blanco, 2) azul y 3) celeste ...	64
Figura 3.4 Cálculo del número de saltos, camino recorrido y distancia recorrida	65

RESUMEN

Esta tesis de grado es un aporte para comprender el funcionamiento de las redes inalámbricas de sensores y en especial de los protocolos de sincronización que permiten mantener el orden y la coherencia en la comunicación de la red.

Para simular una red inalámbrica de sensores, lo primero que se hace es definir el número, el tipo y la ubicación de los nodos en la red, mediante un archivo o en forma aleatoria, luego se determinan las distancias y la suma de ellas a cada nodo, para definir la densidad volumétrica de la nube de nodos, la densidad espectral casi constante que origina el ruido aditivo gaussiano blanco, la distancia recorrida y la frecuencia determina las pérdidas en el espacio libre; el número de saltos define la demora en el procesamiento de cada nodo, hasta el nodo de referencia, el área de barrido será lo que establece los niveles y las profundidades del protocolo HRTS.

Se optimiza el proceso con la densidad volumétrica, espectral, suma de distancia, pérdidas y principalmente el tiempo de corrección para la sincronización, como el criterio del más cercano.

Se utiliza MatLab para procesar datos, lograr mejores resultados y optimizar los procesos.

Al final se realiza una comparación entre el protocolo original y el optimizado y se evalúan las semejanzas y diferencias.

PALABRAS CLAVE: HRTS, WSN, protocolo, sincronización, tiempo

ABSTRACT

This degree thesis is a contribution to understanding the operation of wireless sensors networks, especially the synchronization protocols that allow maintaining order and coherence in the communication of the network.

To simulate a wireless sensor network, the first thing we do is define the number, type and location of the nodes in red, by means of a file or in a random way, then determine the distances and the sum of them in each node, to define the volumetric density of the cloud of nodes, the almost constant spectral density that causes the additive noise, the distance traveled and the frequency determined in the free space, the number of jumps defines the delay in the processing of each node up to the point of reference, the sweep area will be what establishes the levels and depths of the HRTS protocol. The process is optimized with volumetric and spectral density, sum of distance, and correction time for synchronization, as the closest criterion.

MatLab is used to process data, achieve best results and optimize the processes.

In the end, a comparison is made between the original and the optimized protocol and the similarities and differences are evaluated.

KEYWORDS: HRTS, WSN, protocol, synchronization, time

1 INTRODUCCIÓN

El tiempo es fundamental en la operación de los sistemas distribuidos en general y para gran cantidad de aplicaciones, algoritmos y protocolos que se utilizan en las Redes Inalámbricas de Sensores (WSN, por sus siglas en inglés) particularmente en el monitoreo e interacción, con diversos fenómenos físicos.

Estas redes están compuestas por nodos que pueden medir el tiempo independientemente, utilizando relojes locales excitados por sus propios osciladores, lo que provoca lecturas diferentes, debido a los cambios de fase y tasas de variación de dichos osciladores, posición de los nodos y de la distancia de la fuente del fenómeno físico y la baja precisión de los relojes que provocarían errores aparentemente pequeños en la sincronización del tiempo que daría estimaciones significativamente sesgadas.

En sistemas distribuidos la sincronización del tiempo es un problema, más aún en Redes Inalámbricas de Sensores con restricciones en el consumo de energía de los algoritmos, gran cantidad de nodos a sincronizar y requerimientos de precisión [1, 2].

Para solucionar estos problemas existen dos maneras de conseguir estimaciones más confiables [3, 4, 5]:

- Mantener los relojes de los sensores tan sincronizados como sea posible, mediante el uso de algoritmos de sincronización de tiempo.
- Combinar lecturas de múltiples sensores y promediar los errores de sincronización.

Para evitar que las transmisiones se superpongan y colisionen, muchos protocolos de red, entre ellos los protocolos MAC (basados en TDMA o Coordinados con Wake up) necesitan tener el tiempo exacto y acordar los límites de los intervalos de tiempo de los nodos sensores. [6]

La duración de un segundo en los diversos nodos a sincronizar debe ser lo más cercana posible al tiempo físico de un segundo del Tiempo Universal Coordinado (UTC, por sus siglas en inglés). Es decir que el tiempo necesario en las WSN debe ser lo más aproximado al tiempo físico, diferenciándose por supuesto del concepto de tiempo lógico que permita distinguirlos con la información necesaria suministrada por otros sensores.

Un segundo dura como definición y precisión en términos del número de transiciones entre los dos niveles hiper finos del estado fundamental del Cesio 133 [7] y esta información es proporcionada por una serie de relojes atómicos.

El orden en los que producen los eventos en un sistema distribuido no corresponde necesariamente con el tiempo real [8, 9].

Me enfocaré en los algoritmos más notables en las Redes Inalámbricas de Sensores, que se priorizan el consumo de energía y operan en redes de gran escala. [10, 11, 12]

El problema de la precisión en los relojes de nodo

La mayoría de los mecanismos de reloj de nodos sensores y computadoras comparten una estructura similar: el nodo está conformado con un oscilador con una frecuencia específica y un registro contador, que va aumentando en hardware luego que transcurren cierto número de pulsos de oscilador. El software del nodo accede únicamente al valor de este registro y el intervalo de tiempo entre dos incrementos establece la resolución de tiempo conseguida: si existen acontecimientos que ocurren en dicho lapso no es factible definir sus respectivas marcas.

1.1 Pregunta de investigación

¿Cómo mejorar el Protocolo de Sincronización HRTS en las redes WSN, verificando su adecuado funcionamiento en una simulación?

1.2 Objetivo General

Mejorar un protocolo de sincronización en Redes WSN.

1.3 Objetivos Específicos

- 1.3.1 Describir los protocolos de sincronización.
- 1.3.2 Analizar el funcionamiento del protocolo HRTS y programarlo en MatLab o Lenguaje C/C++.
- 1.3.3 Modificar el funcionamiento del protocolo HRTS, para mejorar su desempeño en Redes WSN y programar el protocolo modificado en MatLab.
- 1.3.4 Verificar los comportamientos del protocolo mejorado y el original, observando su desempeño en una simulación.
- 1.3.5 Demostrar las ventajas de la mejora en una simulación.

1.4 Alcance

La presente tesis de grado tiene las siguientes características:

- 1.4.1 Metodología de analítica de datos en donde se detallan el estudio y análisis de los protocolos de sincronización.

1.4.2 La metodología propuesta analizará el trabajo del protocolo de sincronización HRTS, además, modificará su funcionamiento para mejorar su desempeño y se elaborarán sus respectivos programas en MatLab o Lenguaje C/C++.

1.4.3 Al final, la metodología evaluará los resultados mediante una simulación, para comprobar, si realmente se ha conseguido el objetivo planteado en el proyecto.

1.5 Marco Teórico

Sincronización “Es la acción y efecto de sincronizar (que viene del fr. *synchroniser*, y este de *synchrone* 'simultáneo' e *-iser* '-izar'), mientras que Sincronizar “Es hacer que coincidan en el tiempo dos o más movimientos o fenómenos” definición de la Real Academia Española de la Lengua (RAE)

1.5.1 Tipos de sincronización

Se tiene que diferenciar entre el valor del reloj de hardware del nodo i en tiempo real t que se puede representar como $H_i(t)$ con el reloj de software local $L_i(t)$. Pueden estar ligados mediante una relación lineal de la siguiente forma en la Ecuación 1.1:

$$L_i(t) = \theta_i * H_i + \varphi_i$$

Ecuación 1.1 Relación entre el tiempo real y el tiempo del reloj local i

En donde θ_i se llama tasa de deriva y φ_i es el desplazamiento de fase, que pueden ser modificados adecuadamente para realizar el ajuste del reloj, sin afectar el oscilador o el registro contador.

Ahora se puede definir la noción de precisión de los relojes dentro de una red, en donde se distingue dos tipos de sincronización:

- La Ecuación 1.2 representa la Sincronización externa en donde los nodos 1, 2, 3, . . . , n se dice que son precisos en el tiempo t dentro de un obligado δ si

$$L_i(t) - t < \delta$$

Ecuación 1.2 Condición para sincronización externa

contiene todos los nodos $i \in \{1, 2, \dots, n\}$.

- La Ecuación 1.3 muestra la Sincronización interna en donde los nodos 1, 2, 3, . . . , n se dice que acuerdan el tiempo con un límite de δ si

$$|L_i(t) - L_j(t)| < \delta$$

Ecuación 1.3 Condición para sincronización interna

se mantiene para todo $i, j \in \{1, 2, 3, \dots, n\}$.

Es fundamental, para conseguir la sincronización externa, poseer una fuente confiable de tiempo real / hora del Tiempo Universal Coordinado (UTC, por sus siglas en inglés) por ejemplo, un receptor de Sistema de Posicionamiento Global (GPS, por sus siglas en inglés) [13]. Con seguridad si los nodos se sincronizan externamente con un límite δ , también están sincronizados internamente con el límite de 2δ .

Se tendrían tres problemas que no permiten la sincronización:

- El encendido de los nodos se realiza en forma aleatoria a tiempos diferentes, por lo tanto, no existe corrección; sus fases iniciales ϕ_i también son aleatorias. En este documento los tiempos reales van con letras minúsculas y los tiempos de los relojes locales con letras mayúsculas. Para facilitar, no se toma en cuenta los excesos del registro contador.
- Los osciladores frecuentemente poseen una leve desviación aleatoria a priori de su frecuencia nominal, conocida como deriva y , a veces como reloj sesgado; probablemente producida por sus cristales impuros, además los osciladores se ven afectados por diversas condiciones ambientales como presión, temperatura, altura, humedad etc., que cuando funciona WSN, diferirían de las especificaciones del laboratorio. Esta deriva del reloj se mide en partes por millón (ppm) y da el número de oscilaciones adicionales o faltantes que un reloj hace en la cantidad de tiempo necesaria para un millón de oscilaciones a la velocidad nominal. Generalmente, los osciladores más económicos utilizados en diseños de nodos de sensores baratos tienen altas derivas. En varias publicaciones del campo de las redes inalámbricas de sensores, se supone derivas del reloj en el rango entre 1 y 100 ppm [14, 15]. Para las motas de Berkeley, se especifica una velocidad de deriva máxima de 40 ppm. Una desviación de 1 ppm equivale a 1 s (segundo) de error cada $\approx 11,574$ días, una desviación de 100 ppm a 1 s cada $\approx 2,777$ h.
- La frecuencia del oscilador cambia en el tiempo, si varía a corto plazo, es debido a factores físicos como: la presión de aire, los cambios de temperatura, las variaciones de voltaje de suministro eléctrico, entre otras causas; si tiene variaciones a largo plazo son causadas por la vejez del oscilador. [16] Comúnmente se acepta que la frecuencia del oscilador es estable en intervalos de minutos a decenas de minutos; lo que implica la resincronización permanente de los algoritmos de sincronización del tiempo cada cierto intervalo de minutos para realizar un adecuado seguimiento de las frecuencias cambiantes.

Cuando dos nodos tienen el mismo tipo de oscilador e inician su operación simultáneamente usando relojes lógicos idénticos, la diferencia $|L_i(t) - L_j(t)|$ puede volverse arbitrariamente grande conforme t aumenta; por lo tanto, se requiere un protocolo de sincronización del tiempo.

¿Con qué frecuencia debe ejecutarse dicho protocolo de sincronización del tiempo?

Si se supone que un nodo únicamente ajusta su cambio de fase ϕ_i en una ronda de sincronización, asumimos que la velocidad de deriva del oscilador del nodo es fija y se conoce que la deriva es x ppm y si la precisión deseada es δ s. Por ejemplo, si se tiene como máximo una precisión de 10^{-6} s, la restricción es para $x = 20$ ppm y una precisión de 1 ms, necesita una resincronización cada 50 s. Los diseños más desarrollados intentan estimar y corregir el cambio de fase ϕ_i y la deriva actual θ_i , con la esperanza de prolongar los períodos anteriores a la resincronización. [13, 15] Se necesitan varias sincronizaciones para obtener resultados alentadores; debido a que la tasa de deriva no es constante en el tiempo.

La Ecuación 1.4 indica que generalmente es factible enlazar la tasa de deriva máxima, sí, hay un $\rho_i > 0$ tal que:

$$\frac{1}{1 + \rho_i} \leq \frac{d}{dt} H_i(t) \leq 1 + \rho_i$$

Ecuación 1.4 Condición para tasa de deriva máxima

Con lo que podríamos hallar una frecuencia de resincronización conservadora.

1.5.1.1 Propiedades y estructura de los algoritmos de sincronización del tiempo

Los protocolos de sincronización del tiempo se pueden clasificar según ciertos criterios[13, 14, 15]:

1. Tiempo físico versus tiempo lógico

Las redes inalámbricas de sensores, las aplicaciones y los protocolos necesitan principalmente del tiempo físico.

2. Los algoritmos de sincronización externa versus interna

Pueden o no emplear la sincronización del tiempo con patrones de externos, como UTC.

3. Algoritmos globales versus locales

El algoritmo global trata de sincronizar todos los nodos de una WSN, mientras que los algoritmos locales, restringen su sincronización al barrido de un evento de interés. En algoritmos globales, los nodos se mantienen sincronizados con sus nodos vecinos de un solo salto y con los nodos distantes (multihop). Un algoritmo que ofrece sincronización global siempre provee sincronización local.

4. Tiempo absoluto versus tiempo relativo

Numerosas aplicaciones requieren solo de las diferencias de tiempo precisas, para estimar la deriva, en lugar del desplazamiento de fase. La sincronización absoluta abarca todas las sincronizaciones, incluida la sincronización relativa como un caso especial.

5. Algoritmos basados en hardware versus software

Ciertos algoritmos necesitan un hardware y equipos de comunicación dedicados, como Receptores GPS; mientras otros algoritmos basados en software usan los mismos canales que para la transmisión de los paquetes de datos normales y de mensajes sin formato.

6. Sincronización a priori versus a posteriori

En algoritmos A Priori, la sincronización del tiempo del protocolo se ejecuta permanentemente, sin importar la existencia o no de eventos externos. En sincronizaciones A Posteriori o post-facto, la sincronización, se activa solo si existe un evento externo [17].

7. Límites de precisión determinística frente a estocástica

Algunos algoritmos, bajo ciertas circunstancias, garantizan límites superiores absolutos en el error de sincronización, internamente entre nodos, o con respecto al tiempo de un reloj externo; mientras otros algoritmos, exclusivamente proporcionan límites estocásticos en los cuales el error de sincronización debe ser menor que un límite establecido.

Se debe tener una disciplina para la actualización del reloj local. ¿Cómo debe actualizar un nodo los parámetros del reloj local ϕ_i y θ_i ?

Dos restricciones deben cumplirse, la primera es evitar los saltos hacia atrás en el tiempo, es decir, para $t < t_-$, no debe suceder que $L_i(t) > L_i(t_-)$ después de un ajuste. Y la segunda es evitar saltos repentinos, la diferencia $L_i(t_-) - L_i(t)$, para los tiempos t inmediatamente antes y t_- inmediatamente después del reajuste, debe ser mínima.

Los parámetros más importantes de rendimiento de los algoritmos de sincronización del tiempo son las siguientes: precisión, costos de energía y requisitos de memoria.

Precisión La precisión del error de sincronización máximo entre un nodo y el tiempo real o entre dos nodos, es interesante en algoritmos determinísticos; mientras que el error medio, la varianza de error y ciertos cuartiles, son relevantes para algoritmos estocásticos.

Costos de energía Los costos de energía de un protocolo de sincronización de tiempo están en función del número de paquetes intercambiados por el algoritmo, cada vez que se ejecuta, además de la cantidad de procesamiento de cómputo necesario para manejar los paquetes y la frecuencia de resincronización requerida.

Requisitos de memoria Se requiere de un historial de tiempos previos de los paquetes de sincronización, para estimar las tasas de deriva, mientras más información exista, se consiguen estimaciones con mayor precisión, pero necesita más espacio de memoria.

Los protocolos de sincronización del tiempo para WSN constan de cuatro bloques de construcción conceptuales [13, 15, 16, 17]:

1. **Bloque de detección de eventos de resincronización**

Determina los puntos en el tiempo, donde la resincronización se dispara. La mayor parte de los protocolos, realizan las resincronizaciones cada cierto período en función de la tasa máxima de deriva. Un solo proceso de resincronización se conoce como ronda. Si existen superposición de rondas en el tiempo, es fácil identificarlas, asignando números de secuencia. Un nodo solo utiliza las rondas de resincronización más recientes.

2. **Bloque de estimación de reloj remoto**

Adquiere valores de reloj desde nodos o relojes remotos.

- **Técnica de transmisión de tiempo.** -Un nodo i envía su reloj local $L_i(t)$ en el tiempo t a un nodo vecino j , que lo recibe a la hora local $L_j(t_*)$. Básicamente, el nodo j asume $t \approx t_*$ y usa $L_i(t)$ como estimación para el tiempo $L_j(t_*)$. Esta estimación puede ser más precisa, eliminando factores conocidos de la diferencia $t_* - t$
- **Técnica de lectura de reloj remoto.** - Un nodo j transmite un mensaje de solicitud al nodo i , que envía un paquete de respuesta. El nodo j considera como conocidos los valores del reloj del nodo i como son: el tiempo de ida y vuelta del mensaje; y los tiempos de transmisión de paquetes. Por último, el nodo j informa al nodo i sobre el fin de la comunicación.

3. **Bloque de corrección del reloj**

Establece los valores de los ajustes del reloj local, de acuerdo con los resultados del bloque de estimación de reloj remoto.

4. Bloque de configuración de malla de sincronización

En una red multihop fija qué nodos se sincronizan entre sí. Si las redes están completamente conectadas, este bloque es intrascendente.

Sincronización del tiempo en redes inalámbricas de sensores

Los algoritmos de sincronización del tiempo en las redes inalámbricas de sensores necesitan cumplir ciertas características básicas en su diseño y son las siguientes:

1. Debe escalar o avanzar a redes grandes de multisaltos poco confiables y rigurosamente restringidas de energía. La escalabilidad se refiere tanto a la cantidad de nodos como a la densidad de estos.
2. La precisión puede ir desde microsegundos hasta segundos y depende del tiempo que se demora entre la recepción y el momento en que se marca el arribo del último bit de un paquete.
3. Se evita un hardware adicional para mejorar la sincronización del tiempo, porque incrementan los gastos y el consumo de energía por el uso de circuitos dedicados.
4. El grado de movilidad para redes WSN, es bajo, porque el beacom de un nodo puede llegar a sus vecinos en cualquier momento; mientras que en redes MANETs (Mobile ad hoc network) existe un alto grado de movilidad, conectividad intermitente y comunicación esporádica.
5. No existen restricciones en el protocolo MAC, para el retraso en la entrega de paquetes y retransmisiones.
6. El retardo de propagación es minúsculo entre nodos vecinos, en el orden de $0,1 \mu s$ para una distancia de 30 metros y para otra distancia de 60 metros sería de $0,2 \mu s$.
7. El algoritmo debe programar automáticamente los nodos, nunca en forma manual, como sucede en el protocolo NTP. [18, 19, 20]

1.5.2 Protocolos basados en la sincronización emisor /receptor

1.5.2.1 Lightweight time synchronization (LTS)

Este protocolo fue creado por Van Greunen y Rabaey [15]. Sincroniza los relojes de una red de sensores con los relojes de ciertos nodos que se toma de referencia, por ejemplo, los receptores GPS [21]. El protocolo permite controlar y ajustar el consumo de energía y alcanzar una precisión adecuada, lo que da límites de precisión de un modelo estocástico. No existen restricciones cuando se actualiza el reloj local y trata de alcanzar la tasa de deriva. En la Figura 1.1 se indica el Funcionamiento de LTS:

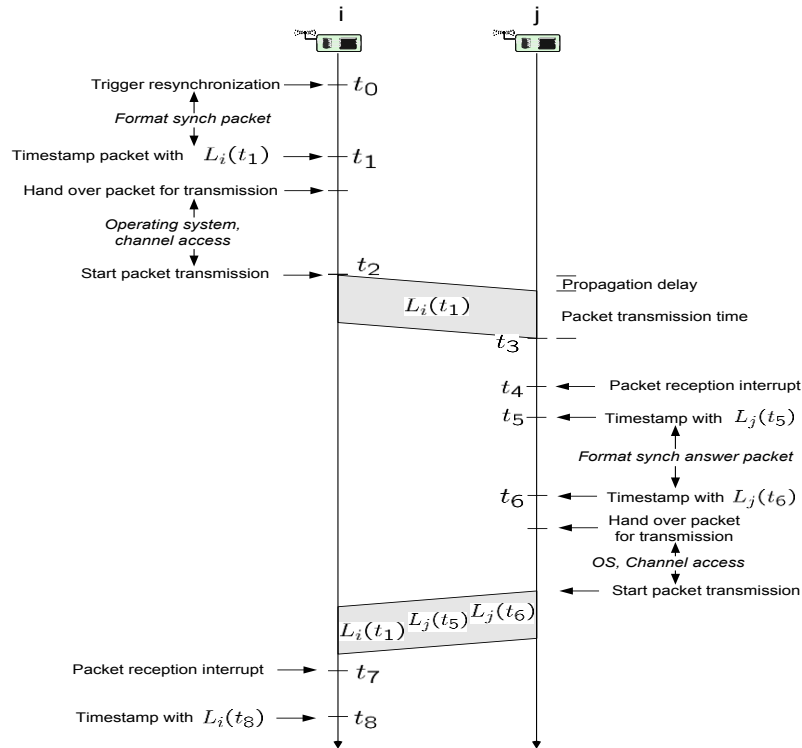


Figura 1.1 Funcionamiento de LTS

LTS está constituido por dos unidades:

- Protocolo que sincroniza los nodos vecinos por pares.
- Construcción de un árbol de expansión para sincronizar a todos los nodos o a un grupo de nodos de interés, con un nodo de referencia común; esto permite anular el error promedio, causado por los errores de sincronización de un solo salto que son libres e igualmente distribuidos.

Sería fantástico tener un error de sincronización igual a cero; pero no es posible, se busca un árbol de expansión de altura mínima, reduciendo la varianza total, desde el nodo de referencia hasta el nodo *i*.

El protocolo de sincronización por pares es una técnica de umbral que lee el reloj en forma remota.

Los tiempos τ y T_p se conocen con alta precisión. [22]

El error viene dado por las interrupciones de latencia y el tiempo entre la recepción y el marcado del paquete.

El receptor registrará los paquetes tan rápido como sea posible. Para muchos receptores que marcan el mismo paquete en sus rutinas de interrupción, las diferencias de tiempo de

acceso tienen una distribución similar con media cero, y su respectiva desviación estándar oscilaría entre $2,3\sigma$ a $4,0\sigma$ como máximo [14, 15].

LTS tiene como meta construir un árbol de expansión de altura mínima y sincronizar pares de nodos al borde del árbol; si se demora mucho tiempo, se introduciría errores adicionales.

1.5.2.1.1 LTS Multihop Centralizado

En primer lugar, se escoge un nodo con receptor GPS o un reloj de alta calidad y precisión para utilizar como nodo de referencia que se sincroniza de forma frecuente y secuencial, [9, 15, 23] con los siguientes fines:

1. Compensar la deriva, conociendo su máximo valor; la altura máxima de expansión del árbol y la desviación estándar σ de un salto simple.
2. Obtener errores de sincronización menores a la precisión δ deseada, con un 99% de probabilidad, consiguiendo de esta manera poca tolerancia a fallas; optimizando esta característica con múltiples nodos de referencia que pueden alternarse y tener el control de acuerdo con el administrador, en caso de que uno falle.
3. Iniciar el árbol de expansión que va cubriendo toda la descendencia debidamente conocida por el nodo de referencia.

El costo y la energía necesaria para la comunicación son factores críticos. Cuando se trata de sincronizar un par de nodos simples, su valor alcanza los 3 paquetes. Si se necesita hacer la sincronización con n pares de nodos, el costo se incrementa a $3n$ paquetes; excluyendo los fallos en el canal y las colisiones.

1.5.2.1.2 LTS Multihop Distribuido

Es más práctico que el LTS centralizado; pues no utiliza un árbol de expansión; pero cada nodo conoce las identidades de los demás integrantes de la red y las rutas apropiadas y asociadas a cada nodo de referencia; esto permite encontrar los mejores caminos. Los nodos cada cierto tiempo reinician la sincronización como parte de sus obligaciones.

El nodo i solicita sincronización al nodo de referencia R más cercano, con el fin de minimizar el error de sincronización; como lo hacen los nodos más cercanos al nodo R , pues tienen períodos de sincronización más cortos; de esta manera actúan secuencial y sucesivamente los nodos intermedios y alejados al sincronizarse con el nodo R , al que acceden directamente, siempre y cuando no exista un nodo de referencia más cercano, al que deberían sincronizarse.

Un nodo j , puede optimizar su sincronización aprovechando las solicitudes, las mismas que se encuentran en curso de los nodos más cercanos; espera cierto tiempo, y trata de sincronizar con el respondedor, o diversifica sus rutas a través de otros nodos con saltos intermedios, como un subproducto.

Un protocolo LTS multipunto distribuido sincroniza todos sus nodos e incrementa el costo por el número de paquetes intercambiados y por la sobrecarga del algoritmo central, entre el 40% y el 100%. Cuando existe una diversificación de rutas o sincronizaciones continuas, el costo se reduce al 15% y el 60%, pero, si solo se tiene un grupo de nodos de interés, se limita la sobrecarga con algoritmos distribuidos y ahorra energía de todos los otros nodos, con la ventaja de sincronizar post facto, de esta manera, el nodo decide sincronizarse con los Nodos de Referencia más próximos.

Al utilizar algoritmos centralizados, que manejan todos los nodos, con un precio base independiente de las solicitudes de sincronización de tiempo, [23, 24, 25] se consigue:

1. Aumentar la precisión al asumir que la deriva entre los nodos i y el Nodo de referencia R , es insignificante por un cierto tiempo. Si el nodo i quiere conocer su desplazamiento de fase, se puede aumentar la precisión al repetir el intercambio de paquetes, para conseguir múltiples estimaciones.
2. Estimar la deriva no es factible con una sola resincronización, por lo tanto, se propone una deriva constante a través del intervalo. Cuando el nodo x con una sola observación, no se reajusta automáticamente. Existe un desplazamiento exacto y datos con señales independientes de ruido gaussiano blanco con promedio cero y una varianza 2σ , sin considerar el desplazamiento inicial.

1.5.2.2 Timing Synchronization Protocol for Sensor Network (TPSN)

Es un protocolo de sincronización por pares, basado en receptor, similar al LTS, funciona en modo asimétrico, sincroniza el nodo i con otro nodo j en forma unidireccional. [22, 26]

$$L_j(t_4) = L_i(t_2) + \sigma + \tau + t_p + \delta_{s,i} + \delta_{r,j}$$

Ecuación 1.5 Marca de tiempo en t_4

En la Ecuación 1.5 se observa la marca de tiempo en t_4 : σ , es la desviación estándar; τ , es el retardo de propagación; t_p , es el tiempo de transmisión del paquete; $\delta_{s,i}$, es una incertidumbre pequeña en el lado del transmisor entre la marca de tiempo y el inicio real de la transmisión; $\delta_{r,j}$, es una incertidumbre pequeña en el lado del receptor en la otra dirección, del paquete de reconocimiento,

En la Tabla 1.1 se visualiza las etapas que cumple para su funcionamiento el protocolo TPSN.

Tabla 1.1 Etapas de funcionamiento TPSN

Acción	Lugar	Tiempo/Marca
Inicio de sincronización	Emisor Nodo i	T_0
Formatea el paquete del pulso de sincronización	Sistema operativo y pila de redes	t_1
Ejecuta el protocolo MAC de la pila de redes, previo a la marca del paquete y, elimina la incertidumbre	Transmisión del paquete, Sistema operativo, pila de redes y retraso de acceso al medio	$t_2 / L_i(t_2)$
Llegada al Receptor	Receptor Nodo j	t_3
Interrupción de recepción de paquete	Receptor Nodo j	$t_4 / L_j(t_4)$
Formateo del acuse recibo del Nodo j	Sistema operativo y pila de redes	t_5
Ejecuta el protocolo MAC de la pila de redes, previo a la marca del paquete y, elimina la incertidumbre	Transmisión del paquete, Sistema operativo, pila de redes y retraso de acceso al medio	$t_6 / L_j(t_6)$ incluidos $L_i(t_2)$ y $L_j(t_4)$
Formateo de acuse de recibo del Nodo i	Sistema operativo y pila de redes	$t_7 / L_j(t_7)$

Mientras en las Ecuaciones 1.6 y 1.7 se observa la marca de tiempo en t_7 y su diferencia.

$$L_i(t_7) = L_i(t_2) + \sigma + \tau + t_p + \delta_{s,i} + \delta_{r,j}$$

Ecuación 1.6 Marca de tiempo en t_7

$$[L_j(t_4) - L_i(t_2)] - [L_i(t_7) - L_j(t_0)] = 2\sigma + (-\delta_{s,i} + \delta_{r,j})$$

Ecuación 1.7 Diferencias de marcas de tiempo

Lo más sobresaliente de esta visión, es que el nodo i marca el tiempo del paquete saliente, lo más tarde posible, y el nodo j marca el tiempo del paquete entrante, lo antes posible, con el indispensable apoyo de la capa MAC; lo que se consigue con tarjetas de red IEEE 802.11.

Con el levantamiento de un árbol de expansión, el nodo raíz que tiene un nivel 0, sincroniza toda la red con TPSN, y, cada uno de los nodos, conoce su nivel en el árbol y la identidad de su padre.

La construcción del árbol se consigue a partir del nodo raíz al que se le asigna el nivel 0, luego los vecinos de este nodo, con un solo salto, se los sitúa en el nivel 1, y así

sucesivamente. Si se tiene un nodo i , se sincronizará siempre con su padre el nodo $i-1$, y asignará el nivel $i+1$ a los nodos que están a un solo salto.

Si el nodo j desconoce su nivel, envía a sus nodos cercanos, de un salto del árbol, una solicitud de nivel y escoge al vecino, con el menor nivel como su padre, si no lo encuentra, es porque su padre se ha ido o ha fallecido, escoge al nivel 1. En la Figura 1.2 se grafica cómo funciona la sincronización entre dos nodos;

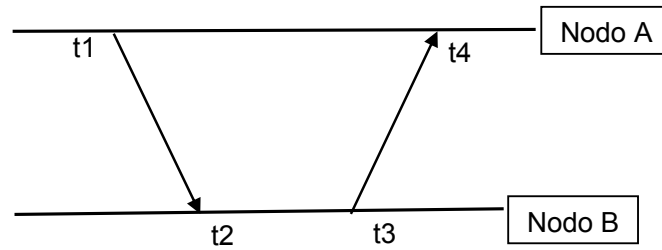


Figura 1.2 Cómo funciona la sincronización entre dos nodos

Se procura tener árboles de expansión mínimos para compensar la falta de precisión en la sincronización. Los errores promedio de sincronización se deben: a las rutas más largas, a las sobrecargas por colisiones y a las demoras por pérdidas en el uso de rutas aleatorias. Con el uso de algoritmos que realizan sincronizaciones periódicas, se optimiza la construcción de los árboles de expansión, de manera que todos los caminos y todos los nodos con sus respectivos hijos, son tomados en cuenta, para sincronizarlos siempre que sean del mismo nivel. El error de sincronización sube en un promedio del 7% por cada salto.

No es necesario levantar un árbol de expansión cuando es factible sincronizar post facto. Se puede definir la edad de un evento, conociendo el número de saltos, cada salto representa un nodo que tiene que pasar dicho evento.

1.5.3 Protocolos basados en la sincronización receptor /receptor

En estos protocolos, el receptor de un paquete se sincroniza con el remitente de este, con una marca de tiempo. Si hay múltiples receptores que reciben el mismo paquete, con una marca de tiempo, se sincronizan entre sí, pero no con el remitente.

1.5.3.1 Reference Broadcast Synchronization (RBS)

Este protocolo [14] se compone de dos partes:

- En la primera, existe un grupo de nodos dentro de un solo dominio de difusión que se escuchan entre sí y aprecian los relojes de sus compañeros.

- La segunda parte, facilita la relación de las marcas de tiempo entre nodos distantes con diferentes dominios de difusión entre ellos.

1.5.3.1.1 Sincronización en un dominio de difusión

El objetivo de esta sincronización se fundamenta en el envío periódico de un paquete por parte del remitente, con o sin marca de tiempo, por un canal de transmisión que marca el tiempo en cada uno de los nodos receptores. Estos datos, permiten conocer los relojes de los vecinos. Cuando se repite este procedimiento, se aprende sobre desfases de fase mutuos, y las tasas de deriva. Se elabora una estadística para cada nodo vecino con la información suficiente para convertir los valores de reloj de los nodos, de los cuales se escoge uno, sin ajustar los nodos a los relojes locales.

Cuando se tiene dos nodos i y j , a distancias distintas, de un nodo de referencia R , que transmite un paquete de impulsos en un tiempo t_i , con su respectiva identificación y secuencia, se producen retardos de propagación diferentes e interrupciones de recepción de paquetes para cada uno respectivamente, luego se marcan el tiempo e intercambian las identidades, uno después del otro, facilitando de esta manera, el cálculo de cambios de fase relativos, como la compensación de fase en una tabla local, sin necesidad de reajustar su reloj, lo más rápido posible, en la interrupción de la rutina.

En TPSN se conoce como incertidumbre del receptor al lapso entre la llegada del último bit y el tiempo marcado por el paquete en cada uno de los nodos i y j .

Basta enviar paquetes de datos normales, con secuencia de números o de pulsos, para sincronizar, no es necesario transmitir un paquete específico de sincronización.

Causas de los errores de sincronización

Si se idealiza y supone que las marcas de tiempo de los paquetes son iguales para los nodos i y j , se puede describir las probables causas de los errores de sincronización:

1. El retraso de propagación en redes WSN, el dominio de difusión es generalmente pequeño y el retraso de propagación del paquete, es irrisorio.
2. El tiempo entre la recepción del último bit y la generación de la interrupción de recepción del paquete es dilatado; debido a retardos en el procesamiento del hardware por cálculos de comprobación, suspensiones y bloqueos de corto y largo plazo por la existencia de secciones críticas o servicio a usuarios de alta prioridad.
3. La tardanza entre la recepción de la interrupción y la marca de tiempo de paquete, cuando se producen simultáneamente, es pequeña.

4. La deriva entre la marca de tiempo t_0 y el intercambio de las marcas de tiempo t_i observadas, generan errores directamente proporcionales al tiempo transcurrido.
5. Los tiempos requeridos para que el nodo R formatee su paquete, lo lleve por el sistema operativo usando el software de la red, y la demora para acceder al medio, son totalmente insignificantes, ya que el punto de referencia común para los nodos i y j es el instante t_0 , cuando el paquete aparece en el medio.

1.5.3.1.2 Comparación de RBS y TPSN

En comparación con otros protocolos basados en emisor/receptor, TPSN elimina gran parte de la incertidumbre en el emisor (sistema operativo y pila de red, retraso de acceso al medio) marcando un paquete antes de la transmisión. [22, 26, 27]

Con RBS se tiene dos receptores i y j , donde el nodo i quiere estimar el desplazamiento al nodo j , se supone insignificante la diferencia en el retardo de propagación y mínima la incertidumbre de transmisión. El error del desplazamiento depende por completo de los términos $(\delta_{r,i} - \delta_{r,j})$ para RBS y $(\delta_{r,j} - \delta_{r,i})$ para TPSN.

Si se utilizan múltiples paquetes de pulsos se produce intercambio de observaciones locales, ignoradas para un cierto umbral. Se acomodarían las tasas de deriva variables en el tiempo.

Para un dominio de difusión se tiene más de 2 nodos, el pulso de paquetes de sincronización es acogido por todos los nodos, que intercambian observaciones. Cualquier nodo r puede estimar desplazamientos de deriva y fase para todos los otros nodos, y estadísticamente tabulados se pueden convertir sus marcas de tiempo locales a cualquiera de sus nodos pares. Los nodos remitentes de pulso tienen las siguientes opciones:

- Con los nodos de infraestructura estáticos se envían paquetes, que recolectan observaciones de los nodos sensores, emparejan desplazamientos, y las tasas de deriva, con el fin de remitir los resultados de regreso.
- Cuando los nodos actúan como receptores y transmisores de paquetes de impulsos, se requiere procesarlos conociendo su identidad y número de secuencia.

La dispersión de grupo comprende los errores de fase que existen entre todos los pares de nodos, los cuales deben minimizarse, como un objetivo de precisión prioritario, en un dominio de difusión. Esta dispersión, es directamente proporcional al tamaño del grupo, e inversamente proporcional al número de observaciones. [24, 26]

RBS tiene el problema que sobrecarga la red, si tiene n nodos, se necesita transmitir $2n$ paquetes. Estos nodos que pueden estar en uno o varios dominios de difusión aumentan

la carga de trabajo, encarece su funcionamiento; esto se reduce, con una sincronización post facto, que se activa cuando existe un evento externo de interés, ya que los nodos no se sincronizan mutuamente.

Todo empieza en un nodo i , que marca el inicio de un evento, activando la sincronización entre los nodos que aprenden sobre su funcionamiento, el desplazamiento de fase y la tasa de deriva, desde el nodo inicial hasta el nodo de destino, con una marca de tiempo en la escala UTC. Cada nodo, mantiene su escala de tiempo local, y su reenvío en un nodo vecino j u otro, que puede ser escuchado por los anteriores.

No se tiene información vinculada con la sincronización de tiempo, la misma que debería estar disponible, sin restricciones, en una tabla con sus respectivos parámetros de conversión.

Si se transmite paquetes entre nodos que están en distintos dominios de difusión, deben existir nodos vecinos que conecten al menos un nodo para ir del origen al destino, con la suficiente superposición de dominios y varios nodos puerta de enlace, para conseguir un determinado equilibrio de carga de reenvío.

Los grandes dominios de difusión intercambian un gran número de paquetes y consumen mayores potencias de transmisión, agotan las baterías velozmente, ayudan a evitar pérdidas de precisión con un reducido número de conversiones indispensables entre un nodo fuente y un nodo receptor. [24, 25]

Los pequeños dominios de difusión solo tienen nodos vecinos de un solo salto, para intercambiar observaciones de pulso, se utilizan para sincronizar todos los nodos, cuando no existen nodos dedicados. A continuación, la Figura 1.3 muestra la Sincronización por pares en LTS:

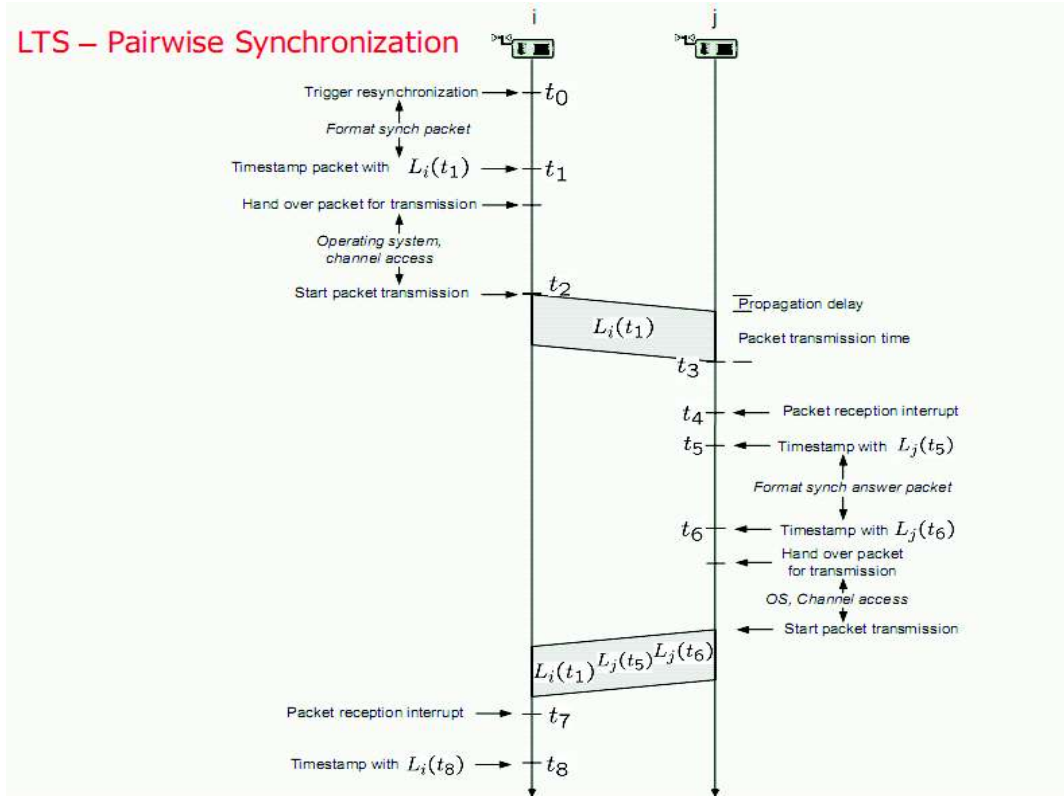


Figura 1.3 LTS Sincronización por pares

1.5.3.2 Hierarchy Referencing Time Synchronization (HRTS)

Este interesante protocolo presentado por Dai y Han, [28] combina los beneficios de dos tipos de protocolos que pueden actuar independientemente, el uno basado en receptor/receptor (RBS) y el otro, basado en emisor/receptor (LTS), sincronizando la mayoría de las redes WSN en forma ágil, de esta manera soluciona la sincronización de los nodos en forma jerárquica. La Figura 1.4 exhibe el funcionamiento de la Sincronización entre dos nodos en HRTS.

En el protocolo basado en receptor/receptor RBS, sincroniza todos los nodos en un dominio de difusión y construye un árbol de sincronización de toda la red, periódicamente. Se requiere tener otro transmisor de impulsos, en el mismo rango del emisor del impulso original y sus nodos asociados. Este protocolo ignora el retraso de transmisión, lo que permite que un paquete sincronice, en un solo salto, a cada nodo. [26, 28]

En el protocolo basado en emisor/receptor LTS sincroniza todos los nodos que pertenecen a un subconjunto de nodos con un reloj de referencia, solo considera los cambios de fase y no intenta corregir las distintas tasas de deriva.

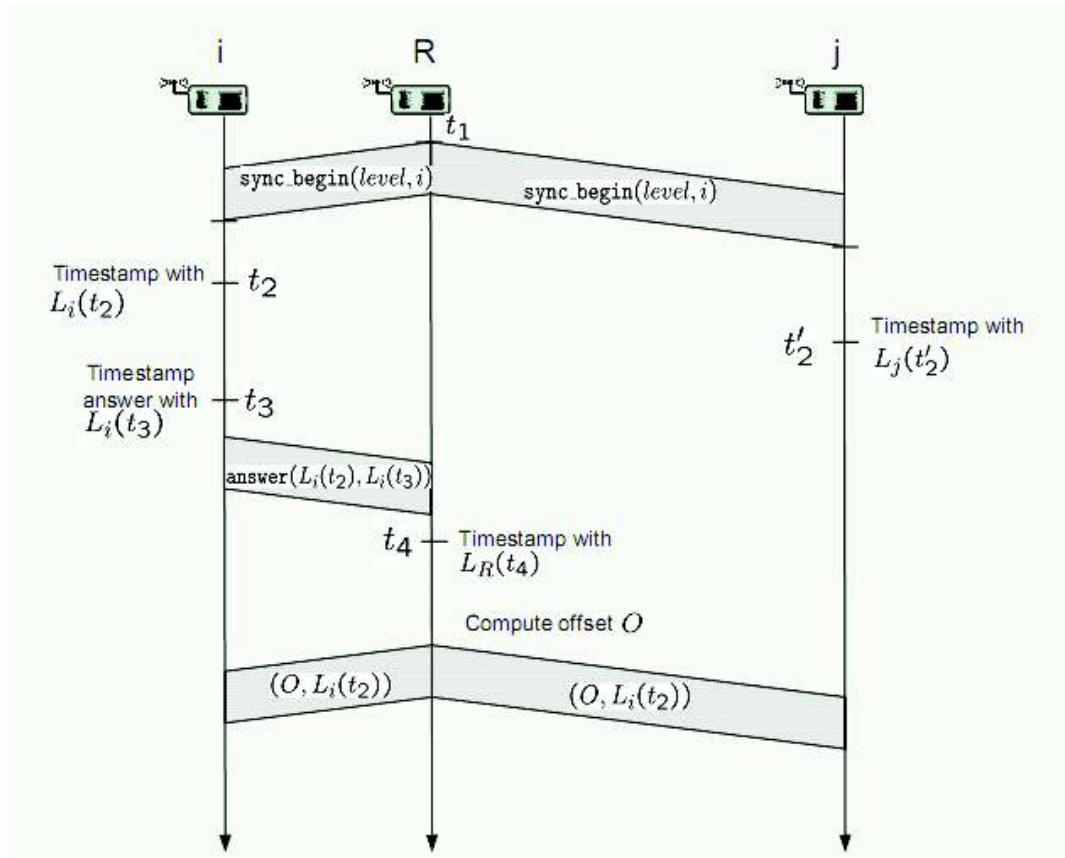


Figura 1.4 Funcionamiento de la sincronización entre dos nodos en HRTS

1.5.3.2.1 Sincronización en un dominio de difusión

Un nodo R, dedicado o raíz, llamado también estación base, a través de la transmisión de un paquete de anuncio de inicio de sincronización, activa la sincronización de tiempo, en el tiempo en t_1 , con marca $L_R(t_1)$, este paquete tiene dos parámetros, un valor de nivel y la identificación de uno de los nodos R y sus vecinos (nodos i y j). El nodo i recibe el paquete en el tiempo t_2 , con marca $L_i(t_2)$, y el nodo j, acepta el mismo paquete en el tiempo t'_2 , con marca $L_j(t'_2)$, muy parecido a RBS; t'_2 y t_2 , podrían diferir ligeramente, debido a las incertidumbres del receptor aleatorio. El nodo i, encuentra la identificación del paquete, formatea la respuesta en tiempo t_3 , con marca $L_i(t_3)$, que junto a $L_i(t_2)$, pertenecen al paquete de respuesta. El nodo raíz R recibe el paquete en tiempo t_4 , con marca $L_R(t_4)$. [28]

El nodo R debe conocer su vecindad para una identificación explícita del nodo; en caso de omitirla, todos los vecinos comienzan a emitir una señal temporizadora con un valor aleatorio, de la cual se escoge la primera respuesta.

Ahora el nodo raíz R puede estimar el desplazamiento entre su propio reloj R y el reloj del nodo local i, como lo hace el protocolo LTS $\sigma_{R,i} = (L_i(t_2) - L_i(t_2)) - (L_R(t_4) - L_i(t_3))$

Lo relevante, es que cualquier nodo i, puede calcular el desplazamiento de fase del nodo R desde su propia observación.

El nodo raíz R, difunde los valores del desplazamiento y de $L_i(t_2)$ a todos los nodos. La característica de este modelo es que, con solo tres paquetes, se sincroniza todos los nodos R vecinos al reloj de R, independientemente del número, y si están en rango mutuo o no.

Cuando en los receptores del paquete comienza la sincronización, no necesitan intercambiar sus observaciones en forma de pares, a diferencia de RBS, por lo que no importa la densidad de la red.

La distribución de errores de sincronización es similar en los protocolos HRTS y RBS, cuando se tiene un solo dominio de difusión y múltiples saltos, en repeticiones de observaciones experimentales. Se disminuyen las incertidumbres de recepción y transmisión, marcando el tiempo de los paquetes salientes lo más tarde posible, y los paquetes entrantes, tan pronto como sea posible. [27, 28]

Dai y Han proponen ejecutar este protocolo sobre un canal MAC dedicado, con la meta de disminuir las posibilidades de colisión y las demoras de acceso al medio (cuando se independiza el canal de reloj entre el nodo i y el nodo raíz R) se separa el tráfico de datos del usuario y el relacionado con la sincronización.

Se presume que toda la red es estática y permanece conectada con varios nodos de referencia R, cada cierto periodo se activa la resincronización. A los nodos de referencia R se les asigna el nivel 0 (incluido en el paquete de inicio de sincronización), y por lo menos uno de estos nodos R posee acceso a una fuente de tiempo externa (hora UTC). Los nodos de referencia R se convierten en nodos raíz para todos sus nodos vecinos, de un solo salto. La Tabla 1.2 nos enseña las condiciones de sincronización y asignación de nivel. [28, 29]

Tabla 1.2 Condiciones de sincronización y asignación de nivel

Variable de nivel local inicializada con una sincronización con un valor	Recibe un paquete con un inicio de sincronización con un valor de nivel	El Nodo acepta el paquete y cambia de nivel al recibido más uno
Suficientemente grande	Realmente más pequeño	Si
Valor pequeño	Mucho mayor	No

La incertidumbre de sincronización proviene del error introducido por R, al estimar el desplazamiento de fase y al asumir que $t_2 = t_{_2}$.

Esto hace que HRTS solo sea factible para dominios de difusión geográficamente pequeños.

1.5.4 Análisis del funcionamiento del protocolo HRTS

1.5.4.1 Definiciones

Es un protocolo de sincronización ligero, bidireccional, flexible y auto organizado, que tiene como objetivo, permitir en forma ágil la sincronización de la mayoría de las redes WSN por autoridades centrales. No requiere una topología fija, ni garantiza una determinada latencia, para implementar el servicio de sincronización de tiempo en una WSN. [28, 30]

Se necesita medios y equipos físicos, que automáticamente satisfaga el medio inalámbrico, como de una red conectada, para extender toda la onda de sincronización, a través de una amplia red de nodos (normales y de referencia).

Este protocolo mejora la precisión, reduce la saturación de la comunicación y baja el consumo de energía, explotando el uso de radios multicanal. El canal de control común que se utiliza para el tráfico de datos en general también es compartido por todos los nodos sensores; entrega mensajes beacon.

Uno o varios nodos de referencia simultáneamente pueden operar en el sistema y transmitir en forma periódica, mensajes beacon a sus vecinos, iniciando así las ondas de sincronización, que dinámicamente seleccionará el nodo de referencia más cercano, como su referencia para la sincronización del reloj. A cada nodo sensor se le asigna un único canal de reloj dedicado diferente a todos los canales de reloj de todos sus vecinos, lo que permite minimizar la variación en el retraso de la propagación debido a las retransmisiones y a la colisión de paquetes mejorando así la precisión en la estimación del reloj, caso contrario si solo se utiliza un solo canal de reloj la precisión, se afectará grandemente el funcionamiento del protocolo. En la Figura 1,5 se ve que, si se tiene un par de nodos A y B que desean sincronizar uno con el otro, en el tiempo t_1 el nodo A envía un mensaje al nodo B y éste lo recibe en el tiempo t_2 , que a su vez envía una réplica en el tiempo t_3 , que es recibida por el nodo A en el tiempo t_4 . Los tiempos registrados por el reloj local del nodo A son t_1 y t_4 , mientras que por el nodo B son t_2 y t_3 . [26, 28]

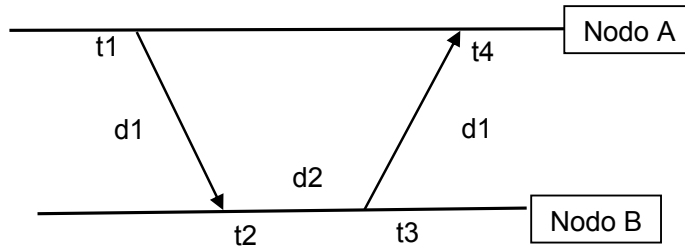


Figura 1.5 Sincronización HRTS

En las Ecuaciones 1.8 se definen los lazos entre los tiempos t_1 , t_2 , t_3 y t_4 , sobre la base de un retraso de propagación del mensaje d_1 y de la compensación del tiempo de los relojes locales entre A y B d_2 , se los toma como constantes en todo el proceso de sincronización, [29] entonces se tiene:

$$t_2 = t_1 + d_1 + d_2 \quad \text{y} \quad t_4 = t_3 + d_1 - d_2 \quad \text{de donde fácilmente se deduce:}$$

$$d_1 = \frac{(t_2 - t_1) + (t_4 - t_3)}{2}$$

$$d_2 = \frac{(t_2 - t_1) - (t_4 - t_3)}{2}$$

Ecuaciones 1.8 Relación entre tiempos, retrasos de propagación y compensación de tiempo de los relojes locales

Como consecuencia, se puede sincronizar el reloj local en el nodo A con el reloj local del nodo B, añadiendo la compensación d_2 y al revés también.

Si se asume que el retardo d_1 es similar en una y otra dirección, se introducen errores que reducen la precisión de ciertos algoritmos de sincronización, Estos cuatro tiempos categorizados producen el mayor retraso en la sincronización de tiempo: tiempo de envío, tiempo de acceso, tiempo de propagación y tiempo de recepción.

El tiempo de envío se modifica cuando existe sobrecarga en la operación del sistema, en el ámbito de la conmutación y en la asignación de recursos, durante la construcción del mensaje. El error en la sincronización se minimiza, procurando que la marca de tiempo sea tan pequeña, como sea posible. [28, 29]

El tiempo de acceso es la demora que se produce cuando el nodo intenta varias veces acceder al medio y está determinado por el protocolo de la capa MAC.

Los tiempos de envío y de acceso contribuyen a los errores de estimación entre t_1 y t_2 .

El tiempo de propagación es el tiempo necesario para que el mensaje sea entregado entre el transmisor y receptor. Cuando existen múltiples saltos es la mayor fuente de error en la

sincronización del sistema. El retardo de la propagación es casi constante en los dominios de difusión y directamente vinculado con la velocidad de transmisión del mensaje en el medio. Sin embargo, el retardo de propagación, varía notablemente en redes de área extendida multi salto, debido a factores aleatorios, como el tiempo de pausa después de colisiones, retransmisiones y retrasos por hacer cola, lo que provoca, la variación de d_1 , que contradice la suposición ideal planteada al inicio que d_1 tiene un valor constante. [29]

El tiempo de recepción, es el retardo entre el tiempo en que el paquete llega a la interfaz del radio receptor y el momento en que el sistema es notificado de dicho arribo, este tiempo influenciará la precisión al valorar t_2 y t_4 .

Primero se analizará un solo nodo de referencia y luego se lo generalizará para múltiples nodos de referencia.

1.5.4.2 Un solo nodo de referencia

El protocolo HRTS repite en cada nivel de jerarquía pasos simples que se detallan a continuación: [28, 29]

1. Para iniciar la sincronización un nodo de referencia que se toma o conoce también como estación base difunde un beacon en el canal de control en el tiempo t_1 .
2. Entonces todos los nodos vecinos secundarios especificados por el nodo de referencia saltarían al canal de reloj especificado almacenando el tiempo del anuncio t_2 y envían una réplica en el canal de reloj en el tiempo t_3 , estos tiempos se almacenan en el nodo de referencia junto con la marca de tiempo t_2 (t_2').
3. La estación base con todos los tiempos de t_1 a t_4 puede calcular d_2 y difundir en el canal de control.
4. Todos los nodos cercanos de interés n_i comparan el tiempo t_2 con su tiempo de marca recibido t_2' y calculan la diferencia d' como $d' = t_2 - t_2'$. Para concluir se corrige el tiempo en el nodo n_i con la siguiente fórmula: $T = t + d_2 + d'$ en donde t es la lectura del reloj local.
5. Entonces los nodos n_i inician el comienzo de la sincronización con sus nodos vecinos descendentes conocidos.

Cuando el proceso de sincronización comienza se supone que cada nodo sensor n_i conoce a sus vecinos, solo el nodo n_i salta a un canal de reloj especificado por el nodo de referencia o estación base, distinto a los de sus nodos vecinos y sin afectar a los otros nodos que esperan la segunda actualización de la estación base.

Un temporizador se reinicializa cada vez que la estación base transmite un anuncio con una alarma cuando se cumple un determinado tiempo de expiración evitando una espera indefinida, o cuando en el camino se pierde el anuncio desde la estación base hasta el nodo n_i especificado.

Se asignan niveles a cada nodo de acuerdo con la distancia o el número de saltos al nodo de referencia o estación base, dinámicamente se va creando un sistema jerárquico multinivel mientras la onda de sincronización se propaga desde la estación base al resto de componentes de la red.

Los pasos anteriormente descritos se repiten en cada nodo de referencia en cada nivel desde la base hasta todas las hojas del árbol.

1.5.4.3 Parametrización de HRTS

El protocolo HRTS admite que las solicitudes sean dimensionadas con dos variables: *nivel* y *profundidad* con lo que se evita actualizaciones debido a solicitudes de sincronización redundantes de nodos pares o descendientes. [29]

El Nivel indica el número de saltos para cada punto o nodo de sincronización implícito en cada paquete de inicio de sincronización o la distancia a su estación base que posee cada nodo con un nivel asociado. Al inicio de cada onda de sincronización, los nodos de referencia establecen el nivel 0 para el paquete de inicio de sincronización, Si se tiene un nodo P actualizado por un paquete de inicio de sincronización marcado por un nivel r , se configuraría el nivel $r+1$, difundiendo un mensaje de inicio de sincronización a todos los nodos vecinos y cercanos del nodo P. Si en el mismo periodo de sincronización, el nodo P recibe posteriormente otro paquete de inicio de sincronización, el nodo P averiguará el nivel que posee este paquete, si es mayor o igual a r , sencillamente no tomará en cuenta la solicitud de actualización, caso contrario enviará una respuesta al paquete de sincronización último e inmediatamente actualizará su nivel basado en el remitente, para luego proceder a construir dinámicamente un nuevo árbol con los nodos de referencia determinados en la estación base. La estación base o nodo de referencia tiene nivel 0, sus nodos vecinos sincronizados de un solo salto tienen nivel 1, luego estos nodos de nivel 1 inician el proceso de paquete de sincronización con sus nodos descendientes o hijos, si dos nodos pertenecen a otros dominios de difusión, pero pertenecen al mismo nivel, solicitudes mutuas simplemente se ignoran.

La profundidad además del nivel HRTS indica el radio de ondulación de sincronización de los nodos de la red, con lo que los nodos se dimensionan como un segundo campo en el mensaje de inicio de la sincronización. En los paquetes de sincronización se inicializan los

nodos con un valor de profundidad, que disminuye en una unidad en cada nivel, hasta que la profundidad se vuelve cero lo que detiene la expansión de la onda de sincronización. Los nodos ulteriores flexibilizan el valor de la profundidad según sus necesidades, lo que permite al punto de referencia, con un flexible artificio, controlar el rango de los nodos que se actualizan.

1.5.4.4 Estudio de HRTS

El protocolo HRTS determina un único punto de referencia común con las facilidades que brindan las transmisiones inalámbricas, lo que consigue igualando el tiempo de llegada t_2 del mensaje de difusión en todos los nodos vecinos, para que se produzcan en el mismo instante, permitiendo la sincronización simultánea de todos los nodos compañeros vecinos, que individualmente calculan sus propios desplazamientos al canal de referencia del reloj de la estación base utilizando solo tres mensajes (dos transmisiones de reloj y una respuesta del canal de reloj) . [28, 29]

HRTS es altamente escalable y ligero, porque solo utiliza por salto un intercambio inalámbrico ligero entre el nodo de referencia y todos sus vecinos, lo que no produce sobrecargas ni la tormenta de difusión, como sucede en RBS, provocada por el incremento del número de pares para la verificación de la sincronización. HRTS tiene la ventaja de difundir por radio un único mensaje de transmisión de inicio de sincronización desde un punto de referencia común a todos sus vecinos.

El nivel y la profundidad de cada nodo se determinan en forma ágil y dinámica, durante la difusión de la onda de propagación, sin importar la densidad de nodos en la red ni conocer protocolos de enrutamiento adicionales, con solo tres mensajes constantes en el dominio de difusión definen su nivel de jerarquía.

En HRTS, el error del remitente se elimina comparando el tiempo recibido en cada nodo, las más importantes fuentes de error son: [29]

1. Varianza en el retardo de propagación HRTS, debido a que el tiempo de arribo de los mensajes de inicio de sincronización es distinto para cada nodo receptor, pero como son señales electromagnéticas viajan a la velocidad de la luz, y la distancia de nodos es corta, se ignora la diferencia de los tiempos de propagación que están en el rango de nano segundos, porque HRTS asume que las velocidades de propagación son las mismas en diferentes canales.
2. Latencia del receptor debido a la inestabilidad del sistema operativo que funciona en el mismo receptor, lo que se controla mejorando el diseño y el procesamiento del sistema

operativo para disminuir la inestabilidad y hacerla determinística con una lectura inmediata apenas el paquete de sincronización es recibido.

De forma aleatoria HRTS selecciona el nodo vecino para enviar el mensaje de inicio de sincronización, sin embargo se puede tener información histórica de sincronizaciones anteriores para seleccionar dicho nodo, combinándolas adecuadamente para simplificar el proceso e incluir las diferencias de retardo de propagación, cuando se trabaja en un dominio de difusión local pequeño, pero son mucho más apreciables cuando en WSN altamente distribuida y las distancias entre los nodos vecinos es muy grande.

1.5.4.5 Múltiples nodos de referencia

El protocolo HRTS admite la presencia de múltiples nodos de referencia o estaciones base en una red WSN para proveer lecturas de reloj más precisas. Cuando comienza la sincronización del paquete se establece el nivel del nodo lo que permite eliminar actualizaciones redundantes de otros nodos de referencia, determinar si el mensaje es de un nodo de referencia más cercano o no. Si existen dos o más nodos de referencia siempre se actualiza a la estación base de nivel más bajo y va construyendo un árbol de ruta más corto para cada nodo de referencia con lo que cada nodo sensor hijo seleccione el nodo de referencia más cercano para corregir su reloj local con lo cual se reduce la inestabilidad. Si un nodo sensor está exactamente a igual distancia de varios nodos de referencia se sincroniza con el primero que llega. [28, 29]

1.5.5 Pull: ITR Protocolo Individual basado en la solicitud de tiempo

El protocolo individual basado en la solicitud de tiempo complementa la sincronización de tiempo bidireccional TSync, facilita a cada nodo sensor obtener de forma independiente el tiempo o sincronizar solo aquellos nodos vecinos a pedido, HRTS sincroniza demasiados nodos mientras que ITR solo sincroniza los nodos que lo necesitan bajo demanda.

ITR funciona de la siguiente forma:

Cuando un nodo n_i envía una consulta ITR al canal de control, enseguida el canal de reloj transmite ITR QUERY.

El nodo de referencia padre o estación base de n_i oye la solicitud y notifica a su estación base con un ITR ACK y así sucesivamente con los padres ascendentes hasta que alcanzan un nodo de referencia.

Si el nodo n_i oye un ITR ACK del nodo n_j , cambia al nuevo canal de reloj especificado como todos los nodos a lo largo de esta ruta, siempre utilizan diferentes canales de reloj, pero si es del mismo nodo n_j el canal de reloj se mantiene.

El procedimiento se repite para sincronizar n_i de acuerdo con los comentarios de la estación base.

Un nodo sensor n_i se puede sincronizar con un nodo de referencia vecino cercano o remoto de muchos saltos de acuerdo con la profundidad que tiene diversos valores, hasta que el campo profundidad o el tiempo hayan expirado o caducado respectivamente, cuando se llegue a los límites de la red, distribuyendo los relojes de manera escalable y jerárquica solo consultando a un nodo vecino superior sin regresar al nodo de referencia.

ITR también es débil a las demoras en la propagación bidireccional que contribuyen al error, pero esto se soluciona con un enfoque multicanal de ITR, que utiliza diversos canales y evita atascos, tráfico cruzado y colisiones.

La sincronización ITR bajo demanda será utilizada con poca frecuencia y no revierte interrupción en el resto de la red.

1.5.6 Protocolo HRTS Modificado

El protocolo HRTS modificado utilizará las ventajas de HRTS original y del ITR, el primero construye un árbol de jerarquía dinámico tomando en cuenta todos los nodos de la red, mientras que el segundo solo toma en cuenta los nodos sensores activos, pero mejorará su desempeño de la siguiente manera: [29]

1. A cada nodo sensor activo se le sincronizará con el nodo de referencia más cercano, es decir que tenga el menor tiempo o el menor número de saltos, si existen dos iguales con el que envíe primero que se sincronice, almacenando dos nodos para escoger cualquiera de los dos si existen diferentes, en un único canal de reloj con un enfoque multicanal.
2. Entonces todos los nodos vecinos secundarios asociados a un nodo de referencia común formarán una subred de dicha estación base.
3. Si una estación base tiene una excesiva sobrecarga o un excesivo número de nodos asignados, los nodos sensores más lejanos se sincronizarán al segundo nodo de referencia en la lista hasta ir balanceando la carga.
4. Si existe una saturación o congestión en la red encontrará el lugar para colocar una nueva estación base para mejorar el desempeño de WSN y del protocolo HRTS.
5. Además de estos pasos descritos anteriormente se utilizará un algoritmo de optimización para mejorar el protocolo HRTS.

1.5.7 Métodos para la Optimización de Algoritmos de Telecomunicaciones

1.5.7.1 Método de punto interior dual primario para programación lineal de gran escala

En las últimas décadas el desarrollo de complejos métodos de punto interior mejoró el rendimiento, proporcionó grandes ventajas y alentó investigaciones exitosas para implementar eficientemente el algoritmo simple. Para cada tamaño y estructura de problema, existe un método de punto interior y un algoritmo apropiado respectivamente, siendo la principal alternativa, a medida que el tamaño del problema se incrementa. [30]

Los métodos de punto interior constituyen una familia de técnicas básicas no-símples para programación lineal de gran escala, con el fin de resolver los problemas numéricos que se presentan en las telecomunicaciones y encontrar soluciones en el interior de la región factible, y no, en los extremos o vértices de esta área, mediante un algoritmo simple que busca el mejor valor para la función objetivo, trasladándose de un vértice a otro.

Si existen problemas con estructuras bien definidas, se resolverá con éxito, perfeccionando algoritmos. El algoritmo simple y los métodos de punto interior permiten eficazmente soluciones factibles básicas y refinadas en los problemas de flujo de red. Los problemas de programación lineal pueden resolverse con el método de ramificación y corte, como subproblemas de otros problemas.

El algoritmo efectúa una serie de transformaciones de forma iterativa en el espacio original, de esta manera, se encuentra el politopo, y al final regresa a su valor original. Se basa en la aplicación a problemas lineales de métodos que clásicamente habían sido considerados como de programación no lineal. La mayoría de las obras recientes de programación no lineal, incluyen este tipo de técnicas.

Se debe tomar en cuenta ciertas consideraciones computacionales, en tres categorías:

- Inicialización
- Búsqueda y escalado del vector de direcciones newtonianas, y
- Terminación.

La inicialización es una etapa de pre-solución y la búsqueda de un punto inicial.

Hay dos enfoques para resolver el vector de las direcciones de Newton: el de sistemas aumentados y el de ecuaciones normales. La elección del enfoque se basa en la densidad relativa. Si la primera es más densa en comparación con la última; la formulación y el uso de las ecuaciones normales es costosa, y amplía la búsqueda de correcciones múltiples, dando métodos de orden superior, que reduce el número de iteraciones requeridas para resolver el problema.

Sobre los criterios de Terminación, se monitorea la brecha de dualidad y viabilidad en cada iteración, hasta que sean lo suficientemente pequeños, como para cumplir con la precisión requerida de 8 dígitos.

La última de estas condiciones, es la más estricta. Algunas implementaciones pueden presentar una fase de terminación, en la que se recupera un vértice / base óptimo.

Existe una clara diferencia con relación al método simple, el cual avanza por la frontera de la región, moviéndose de un punto extremo a otro. La Figura 1.6 muestra los dos comportamientos. El método de punto interior necesita que el punto inicial x_0 , sea parte de la región factible.

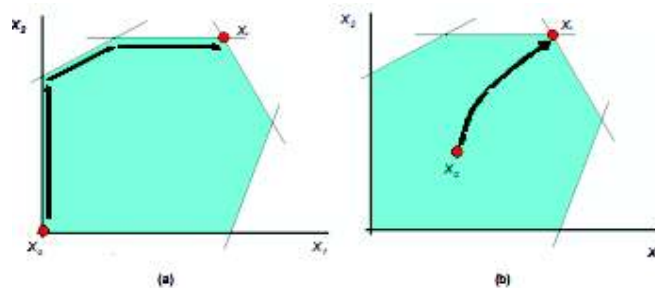


Figura 1.6 Camino seguido para alcanzar el punto óptimo x^* desde el punto inicial x_0 por a) el método simple; b) un método de punto interior

Existen algoritmos de punto interior polinómicos diferentes al método simple que llega al punto óptimo en forma iterativa, con una función polinómica, de acuerdo con el tamaño del problema.

Los métodos de punto interior son los siguientes:

- Métodos de escalado afín

Son los más sencillos de comprender, porque transforman el problema original por medio de una fácil implementación, con muy buenos resultados, una variante del método de gradiente proyectado.

- Métodos basados en transformaciones proyectivas

La Ecuación 1.9 indica cómo se transforma el problema original, pero de una manera más sofisticada, con una transformación proyectiva. [30]

$$\tilde{x} = \frac{X^{-1}x}{e^T X^{-1}x}$$

Ecuación 1.9 Transformación proyectiva

En donde $x \in R^n$ es el punto actual; \tilde{x} es la imagen de x en el nuevo espacio de variables; X es una matriz diagonal cuyos elementos diagonales los componentes de x , e^T es el vector n-dimensional del algoritmo desarrollado por Karmarkar (1984).

- Método de seguimiento de la ruta o path-following

La Ecuación 1.10 revela cómo se emplea la técnica de barrera logarítmica para problemas no lineales, no directamente, sino mediante una secuencia de problemas de la forma:

$$\min_{x \in C} c^T x - \mu \sum_{j=1}^n \ln(x_j) \quad x \text{ en donde Sujeto a } Ax = b$$

Ecuación 1.10 Secuencia de problema con la técnica de barrera logarítmica

Si $\mu \rightarrow 0$ se consigue demostrar que la secuencia de soluciones de (P_m) se acercan a la solución del problema (P) original, el conjunto de soluciones de (P_m) para diferentes m proporciona el denominado camino central (central path). Los métodos path-following siguen una trayectoria cercana a este camino central en su búsqueda del óptimo. El algoritmo primal-dual de punto interior pertenece a esta categoría. Este algoritmo ha mostrado ser, en general, el método de punto interior más eficiente. Hoy en día es considerado como la mejor opción para solucionar problemas lineales de gran dimensión.

Programación no lineal en telecomunicaciones

Las telecomunicaciones han sido y son objeto de aplicaciones para técnicas matemáticas avanzadas. Se revisa enfoques clásicos de programación no lineal para modelar y resolver ciertos problemas. Se resalta los aspectos comunes de las telecomunicaciones y las redes

viales. Existen varias lecciones en el campo de la ciencia del transporte similares, con enfoques teóricos y de equilibrio de juegos, estudiados por más de cuarenta años. [31]

La mayoría de los análisis de problemas de diseño, en sistemas de telecomunicaciones, pueden establecerse en términos de optimización, es decir, minimizar alguna función de rendimiento o utilidad en varias variables; cuyos valores deben satisfacer un conjunto de requisitos específicos. Estos problemas de optimización suelen ser de naturaleza combinatoria en el contexto de sistemas de comunicación; incluidas las áreas de redes inalámbricas, Internet de alta velocidad, ecualización, acceso de banda ancha, topología de red, enrutamiento, conservación de energía en redes inalámbricas ad-hoc como en la extensión de la vida útil de las mismas. Garantiza la optimización global de la solución, cumpliendo con los requisitos de convexidad de la función objetivo y se encuentran dentro de la región factible.[30, 31]

Los problemas de equilibrio del tráfico, a menudo, se dividen en dos casos de modelos: demanda fija y demanda elástica.[31]

En el caso de demanda fija, hay una matriz de demanda de origen-destino. En el caso de demanda elástica, se modela como una función del menor costo de viaje entre los puntos finales del enésimo par origen-destino. El usuario tiene una serie de opciones de viaje disponibles y de acuerdo con su valor económico, toma la decisión de hacer o no el viaje. En la implementación secuencial del algoritmo de Frank-Wolfe, los subproblemas deben ser resueltos secuencialmente, bajo un orden preespecificado, mediante la incorporación de información. Es obvio, que al implementar el algoritmo de Frank-Wolfe, se genera nueva información durante el ciclo de resolución de subproblemas, sin poder aprovecharla durante la duración del ciclo. Por lo tanto, se sugiere en Migdalas (2004), bajo el nombre de linealización cíclica, que la nueva información generada durante el bucle de resolución de subproblemas Frank-Wolfe debería utilizarse durante la duración del ciclo.

Los modelos de demanda elástica proporcionan un instrumento adecuado para motivar o desalentar a los usuarios en sus elecciones de servicios a través de mecanismos de asignación y / o asignación de precios apropiados.

1.5.7.2 Programación entera en telecomunicaciones

Los programas enteros, y los enteros mixtos (IPs y MIPs, por sus siglas en inglés), brindan una forma flexible con precisión matemática para formular muchos problemas del mundo

real, pero la solución optimizada, puede ser computacionalmente exigente. Si existen dos formulaciones válidas distintas de IP del mismo problema, una puede resolverse fácilmente con métodos existentes, mientras que la otra, podría necesitar de un gran esfuerzo computacional.[30,31]

Los modelos de programación enteros surgen prácticamente en todas las áreas de aplicación matemática y desempeñan un papel importante en las decisiones gerenciales. Si se tiene subestructuras en una formulación del método IP, ofrecen modelos básicos y sugieren posibles aplicaciones metaheurísticas para resolver y optimizar los problemas de telecomunicaciones, de manera más eficiente.

El método clásico para resolver programas enteros es la bifurcación (Land y Doig, NW). El método **branch-and-bound (ramificación y rastreo)** se basa en la idea de analizar iterativamente la bifurcación (Conjunto S) con el fin de dividir al programa entero original en subproblemas. Cada subproblema se resuelve de forma exacta o aproximada, para obtener un límite superior en el objetivo propuesto, por medio de la relajación del problema.

La solución de un problema de relajación o facilitamiento proporciona un **buscador o rastreador** superior sobre el valor objetivo del problema subyacente.

Se ha demostrado que el preprocesamiento y reformulación de problemas es una forma muy efectiva de mejorar las formulaciones de programación de enteros antes y durante las ramificaciones.

Los problemas de telecomunicaciones a los que se ha aplicado la programación entera se pueden dividir aproximadamente en dos tipos:[31]

- Problemas de estructura de red. - El objetivo de este problema, es encontrar una asignación óptima de la capacidad en los bordes de la red, de modo que satisfagan todas las demandas de transmisión y otros requisitos como, por ejemplo:
 - Especificar una topología particular de los bordes de la red, con capacidad distinta de cero,
 - Hay que asegurar que se cumplan las demandas de transmisión, incluso, en el caso de cualquier nodo o borde del gráfico o red,
 - Evitar transmisiones de datos de longitudes de onda similares para que no se produzca diafonía en los enlaces de la red

- Problemas de telecomunicaciones. - Consiste en una amplia variedad de asignaciones:
 - Para agrupar nodos en un gráfico y determinar un diseño óptimo de una red óptica (SONET),
 - De frecuencias a estaciones base para minimizar la diafonía,
 - De clientes a las instalaciones cercanas.
 - De canales
 - De células a conmutadores o áreas de registro de ubicación
 - De tráfico de redes satelitales e intervalos de tiempo en un cronograma
 - Para ubicar instalaciones de forma óptima
 - Para agregar / soltar multiplexores, y / o estaciones base con el fin de crear una red con la arquitectura topológica deseada.

Los procedimientos heurísticos proporcionan soluciones rápidas y factibles para programaciones enteras de telecomunicaciones, pueden usarse repetidamente dentro del árbol de búsqueda de ramas y límites, fijan un límite superior para definir un costo reducido. Un buen rastreador superior aumenta la probabilidad de localizar nodos activos, lo cual es extremadamente importante, cuando se resuelve un programa entero a gran escala, ya que tienden a crear muchos nodos activos que conducen a la saturación de la memoria.

La heurística en el sentido más estricto se refiere a ciertos procedimientos basados en LP que permiten construir soluciones factibles integrales, a partir de puntos, que, en cierto sentido, son buenos, porque satisfacen la integralidad. Típicamente, estos puntos no integrales se obtienen como soluciones óptimas de relajaciones de LP. Los procedimientos heurísticos primarios implican sucesivos ajustes variables de mezcla y redondeo. [30,31]

Dado un problema primal (P), en la Ecuación 1.11 se define como Lagrangiano a la función:

$$L(x, \mu, \lambda) = f(x) + \sum_{i=1}^m \mu_i g_i(x) + \sum_{j=1}^p \lambda_j h_j(x)$$

Ecuación 1.11 Función Lagrangiana

Las variables duales o multiplicadores de Lagrange deben cumplir con las siguientes condiciones: $\mu_i \geq 0$ y $\lambda_j \in \mathbb{R}$

1.5.7.3 Algoritmos de corte y relajación Lagrangianos

En la relajación de programación lineal, se usa ampliamente las implementaciones de ramificación y unión, sin embargo, se puede utilizar otras relajaciones de mejor calidad con un rastreador superior del valor objetivo óptimo para el problema (IP). En 1974, Geoffrion, puso el nombre de "relajación lagrangiana" para este enfoque. La relajación lagrangiana funciona dividiendo un conjunto de restricciones en dos ("se dializan"). [32]

La facilidad Lagrangiana no es conveniente para todos los casos de problemas de programación entera, es esencialmente adecuada para el problema de asignación generalizada.

La técnica puede usarse o combinarse con algoritmos de solución basados en LP en un enfoque de solución híbrida, parece ser muy atractivo para el desarrollo de la heurística lagrangiana.

La definición de una mejor topología de red determina qué pares de nodos deben estar conectados mediante enlaces de transmisión y fija la configuración e interconexión de las fuentes de tráfico y sumideros.

La definición de capacidades (o dimensionamiento) del enlace: determina el tipo y la capacidad del equipo de transmisión que se instalará en cada enlace para construir la red de acuerdo con la topología elegida.

1.5.7.4 Algoritmos y modelos de flujo de red de múltiples variables en telecomunicaciones

Se presenta una descripción general de los modelos de optimización matemática y los algoritmos de solución relacionados con el diseño y el dimensionamiento de redes óptimos en telecomunicaciones. Todos los modelos discutidos se expresan en grandes cantidades de flujos de red de múltiples componentes de costo mínimo con la elección apropiada de funciones de costo de enlace (o nodo). Se examinan varios casos especiales relacionados con aplicaciones prácticas, incluyendo el caso lineal, el caso lineal con costo fijo, y el caso de las funciones generales de costos no decrecientes. [31, 32] También se muestra cómo los modelos genéricos presentados pueden acomodar una variedad de restricciones encontradas frecuentemente en las aplicaciones, tales como, entre otras, limitaciones en la elección de rutas, robustez y restricciones de supervivencia. Finalmente se proporciona una visión general de los desarrollos recientes en algoritmos de solución, enfatizando aquellos enfoques capaces de encontrar

soluciones probables exactas, que se basan esencialmente en técnicas de programación de enteros mixtos generales, relajación, técnicas de generación de plano de corte y bifurcación y corte. [33]

Aquí interesa el desarrollo de modelos y métodos de solución exactos para un diseño y dimensionamiento óptimos de las redes de telecomunicaciones.

Expresado en términos generales, el problema básico abordado aquí puede ser subsumido como sigue. Dada una lista de nodos de tráfico (fuentes o sumideros), (tráfico telefónico, datos, tráfico multimedia) junto con los valores anticipados conocidos para el volumen de tráfico a intercambiarse entre nodos, se requiere construir una red que conecte fuentes y sea capaz de manejar todos los flujos de tráfico solicitados simultáneamente. Logrando esto implica la resolución conjunta de dos tipos de problemas:

- Definición de una mejor topología de red: determina qué pares de nodos deberían ser conectados por enlaces de transmisión. Por lo tanto, la topología de red no es otra cosa que el gráfico que especifica cómo las fuentes de tráfico y los sumideros están interconectados.[33]
- Definición de capacidades de enlace (o dimensionamiento): determinar el tipo y la capacidad del equipo de transmisión que se instalará en cada enlace para construir la red según la topología elegida.

Una solución al problema de diseño de red es la elección de adecuadas topología y capacidades de enlace, competentes para que simultáneamente cubran los requisitos del tráfico entre todos los pares fuente-sumidero. Se verifica y asegura que cada nodo sea fuente o sumidero, se puede encontrar una o varias rutas para que fluya el tráfico correspondiente solucionando el llamado problema de enrutamiento de tráfico.

1.5.7.5 Algoritmos de ruta más corta

Los problemas de ruta más corta son primordiales para la optimización de una red, se encuentran en muchos campos, con una amplia gama de aplicaciones, incluidas la programación dinámica, administración de proyectos, enrutamiento en redes y transporte. Son sencillos de resolver y llevan a una gran cantidad de algoritmos adaptados para obtener una solución óptima en diversas condiciones especiales y/o formulaciones restringidas. Son el inicio del estudio de problemas de red más complejos. [34]

Una ruta simple, es una caminata sin repetición de nodos, es decir, no tiene bordes hacia atrás. La longitud de cualquier trayectoria P, se define como la suma de las longitudes de los bordes que conectan nodos consecutivos en la ruta. Existe al menos una ruta directa de i a cada nodo.

El algoritmo genérico de ruta más corta.

Existen muchas implementaciones del algoritmo genérico con enfoques iterativos que en cada paso establecen etiquetas de distancias de trayecto más cortas a los nodos. Difieren solo en el criterio de selección del próximo nodo, que se eliminará del conjunto Q. [33,34] Los problemas de ruta más corta se dividen en dos métodos:

- **Métodos de configuración de etiquetas:** Los algoritmos que pertenecen a este grupo eliminan de Q (conjunto de los nodos del trayecto); el nodo r correspondiente a la etiqueta mínima y designan como óptima una etiqueta permanente en cada paso. En cada iteración, debe calcularse la etiqueta mínima sobre Q
- **Métodos de corrección de etiquetas:** Estos algoritmos consideran a todas las etiquetas como temporales, hasta la última iteración, cuando se vuelven permanentes. La elección del nodo i que se eliminará de Q requiere menos cálculos, pero un nodo i puede estar incluido más de una vez.

1.5.7.5.1 Algoritmos de establecimiento de etiquetas

El primer algoritmo de establecimiento de etiquetas pertenece a Dijkstra (1959). [33, 34] Propuso un método para resolver el problema del árbol de la ruta más corta.[33, 34]

Algoritmo de Dijkstra. - Mantiene y ajusta un vector de etiquetas de distancia de nodo i . En el proceso, el conjunto de nodos V se divide en dos grupos: los nodos etiquetados permanentemente y temporalmente, que se almacenan en la lista de candidatos Q.

El algoritmo de Dijkstra realiza dos tipos de operaciones: selecciones de nodos de la lista de candidatos Q y actualizaciones de etiquetas de distancia.

El algoritmo de Dijkstra Reverso. - Mantiene y actualiza los nodos con una etiqueta de distancia temporal o permanente; y los coloca en la lista de candidatos Q. Una etiqueta

permanente asociada con un nodo i , representa la distancia más corta desde i a d , mientras que una etiqueta temporal, expresa un límite superior. Iterativamente, busca la mínima distancia y se detiene cuando la encuentra, hasta que todas las etiquetas sean permanentes.

Algoritmo combinado de Adelante / Reverso de Dijkstra.

El algoritmo Reverso Dijkstra es útil si se combina con su homólogo directo, es más eficiente, en la selección de nodos y actualizaciones de etiquetas de distancia. El espacio de memoria requerido puede ser grande, pero puede reducirse de 4 a 1. [34]

Implementaciones de Heap (de montón). Otras implementaciones proponen un montón para representar el conjunto Q . Un montón es una colección de objetos, cada uno con un número asociado real llamado clave. Es compatible con las siguientes cinco operaciones,

Create -heap: crear un montón vacío; **Find-min:** encuentra el objeto con la clave mínima y lo almacena en I ; **Insert:** inserta el objeto I con la tecla (i); tecla de disminución (valor, i , Q): disminuye la clave del objeto i por valor. Tener en cuenta que el valor debe ser menor que la clave que está modificando; **Delete-min** elimina un objeto i con la clave mínima; **Decrease** disminuye un objeto i con la clave mínima.

1.5.7.5.2 Algoritmos de corrección de etiquetas

Los algoritmos de corrección de etiquetas resuelven complejos problemas, de ruta más corta, con longitudes de borde arbitrario, identifican un ciclo negativo, cuando existe, o resuelven un problema de ruta más corto, si el gráfico subyacente, no contiene un ciclo negativo, proporciona una condición de terminación y brinda sugerencias útiles para resolver problemas de manera óptima. Todos estos métodos, en algún tipo de cola, almacenan el conjunto de etiquetas de distancia. [33, 34]

Los Algoritmos de detección de ciclo negativo, surge directamente como un subproblema en algoritmos para resolver otros problemas de red; si se supone que los algoritmos no contienen ciclos negativos, sería ideal. Con la presencia de ciclos negativos, ningún conjunto de etiquetas de distancia cumpliría las condiciones de optimización y cualquier algoritmo de corrección seguirá reduciendo las etiquetas de distancia indefinidamente, sin terminar nunca. Sin embargo, los algoritmos pueden modificarse fácilmente para gestionar

la presencia de ciclos negativos. En la Figura 1.7 se grafica el Algoritmo de la ruta más corta.

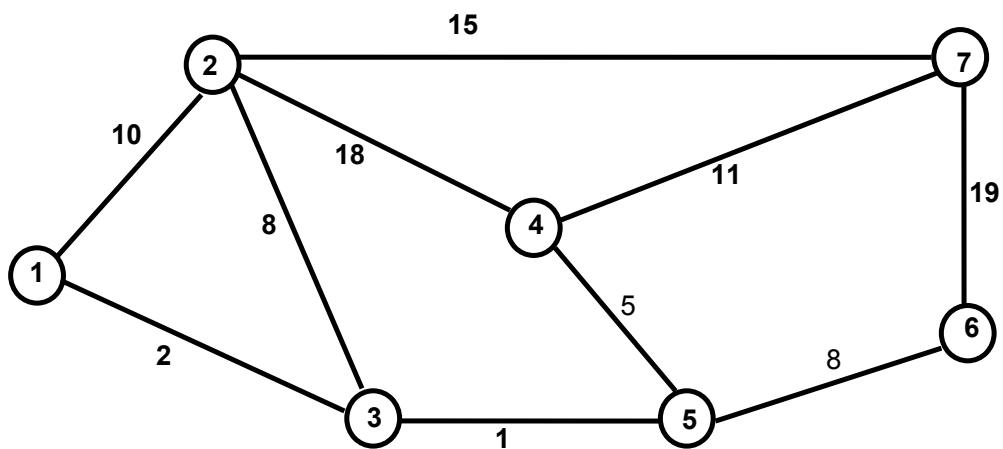


Figura 1.7 Algoritmo de la ruta más corta

2 METODOLOGÍA

En primer lugar se simula una Red Inalámbrica de Sensores compuesta por unos dispositivos electrónicos elementales inalámbricos, conocidos como nodos sensores que receptan, transmiten y cumplen tareas de monitoreo (de diversos parámetros físicos como: temperatura, presión, humedad relativa, radiación, velocidad del viento, caudal, sismicidad entre otros) con una reducida o amplia capacidad de cómputo (según el caso pueden ser simples sensores, estaciones base o nodos de referencia y elementos de acción, reacción y corrección) y una limitada fuente de energía, que cumplen sus tareas de monitorear satisfactoriamente.

Una WSN puede tener cientos o miles de sensores en una determinada área geográfica, con un alcance limitado por lo que los mensajes requieren realizar varios saltos hasta llegar a informar al nodo de referencia, central o estación base que mantiene una comunicación interactiva para concentrar los datos de la red en una computadora o dispositivo portátil.

De los métodos de optimización que se ha estudiado se establece una combinación de:

- Método de punto interior dual primario encuentra una solución que pertenece al conjunto de la red, en su núcleo y cubre a todos los nodos de esta.
- Programación lineal entera en telecomunicaciones, este proyecto dividió su solución en tres partes, la primera distribución volumétrica, la segunda distribución espectral y la tercera el protocolo propiamente dicho, con lo cual se pudo mejorar el funcionamiento de este.
- Algoritmo de la ruta más corta; se establece la ruta más corta utilizando los alcances mínimos requeridos para cada nodo y el menor número de saltos.

Existen algunos simuladores de WSN los más conocidos: Cooja, Castalia y GloMoSin, desarrollados en ambientes Java, OMNet++ y Lenguaje C (y Parse) respectivamente.

Este proyecto de simulación realizado en MatLab, ayuda a comprender el funcionamiento de la red WSN y del protocolo de sincronización HRTS, la arquitectura y el funcionamiento de las comunicaciones y el sistema en general.

Es ventajoso y conveniente desarrollar la simulación del protocolo HRTS para comprender su funcionamiento. Para esto se desarrolló el programa FALGON-HRTS versión 1.0 que permite mejorar la sincronización, realizar pruebas y valorar los algoritmos desarrollados, además de ser un aporte para futuras investigaciones, probar nuevos modelos y modificar ciertos parámetros de diseño y probar diversas combinaciones para mejorar el funcionamiento del protocolo.

Implementación de la Metodología

A través de MatLab (**Matrix Laboratorio**) se compila los algoritmos para posteriormente programarlos y ejecutarlos.

MATLAB es el nombre abreviado de “MATriz LABoratorio”. Es un programa con un lenguaje de alto nivel, estructurado y orientado a objetos, para realizar cálculos numéricos con vectores y matrices, en forma rápida y con alta precisión, manejo simbólico y por tanto se trabaja también con números escalares (tanto reales como complejos), con cadenas de caracteres y con otras estructuras de información más complejas. Matlab es un lenguaje de alto rendimiento para cálculos técnicos, es al mismo tiempo un entorno y un lenguaje de programación. Uno de sus puntos fuertes es que permite construir nuestras propias herramientas reutilizables. Se crea fácilmente nuestras propias funciones y programas especiales (conocidos como m-archivos) en código Matlab, lo que se agrupa en Volvox (también llamadas librerías): colección especializada de M-archivos para trabajar en clases particulares de problemas, especializadas en algunas ramas de la ciencia e ingeniería. Matlab, a parte del cálculo matricial y álgebra lineal, también puede manejar polinomios, funciones, ecuaciones diferenciales ordinarias e integrales, gráficos y visualizaciones avanzadas, con una extensa cantidad de funciones. Integración con aplicaciones externas y soporte básico para diseño de interfaz básica.

2.1 Simulación del Protocolo HRTS original

2.1.1 Análisis

Se determinan las principales propiedades que debe poseer el software:

- Facilitar al sistema establecer un lugar para la ubicación de los nodos sensores en el espacio tridimensional de forma aleatoria o a través de archivo creado anteriormente.
- Determinar el número total de nodos normales que el usuario desee y el de referencia.
- Calcular la distancia que existe entre los distintos sensores y la suma de las distancias de cada uno de los nodos a los demás y guardarlas.
- Se calcula el número de saltos de todos los nodos hasta el nodo de referencia.
- Almacenar las características técnicas de cada sensor y qué tipo de parámetros monitorea.
- Variar el número de sensores y su ubicación en la red existente.
- Calcular los tiempos de transmisión, recepción, retardos de propagación, ruido blanco gaussiano aditivo y procesamiento dentro del nodo para deducir el tiempo de corrección de sincronización.

- Encontrar el Nivel y la Profundidad (1.5.4.3) de cada nodo descendiente y junto con el tiempo de corrección de sincronización, mostrarlo en una tabla.
- Un gráfico que indique cada nivel

2.1.2 Diseño

- Lo primero que se hace es definir la ubicación física de los nodos en el espacio tridimensional.
- Se tiene tres parámetros para definir el árbol jerárquico de HRTS:
 - Distancia entre el nodo de referencia y los otros nodos y su suma (se analiza tomando a cada nodo como si fuera de referencia) y se establece una tabla en orden ascendente.
 - Se analiza la densidad espectral-

2.1.3 Pruebas

Se determinan las principales características de la red WSN y se realizan cambios para ver su funcionamiento.

2.2 Simulación del Protocolo HRTS mejorado

Esta comprende el ciclo de vida clásico de un software: Análisis, Diseño, Implementación y prueba del Sistema.

2.2.1 Análisis

Además de las características fundamentales enunciadas en la Simulación del Protocolo HRTS original se añaden las siguientes:

1. Probar a cada nodo de la red WSN, como si fuese una estación base y ver cuáles son los que tienen los menores tiempos de corrección de sincronización y distancia (sumados) y colocarlos en orden ascendente.
2. Analizar que nodo hijo es más cercano a la estación base e ir generando una Nueva Tabla de referencia jerárquica con las mejores combinaciones y tiempos de corrección.
3. Ver la sobrecarga de la estación base y colocar otra de ayuda para que se equilibre la carga con el fin de mejorar la red.
4. Utilizar un mecanismo de optimización del capítulo 3 para mejorar la respuesta y la simulación.
5. Una figura que indique el original y el optimizado.

2.3 Programas, funciones y scripts utilizados

Para la implementación de la simulación del protocolo HRTS se utilizan los siguientes procesos:

1. Generación de un archivo aleatorio o utilización de un archivo previamente almacenado.
2. Determinación de distancias entre cada nodo y su suma.
3. Establecimiento de los niveles de cada uno de los nodos de acuerdo con su densidad espectral, utilizando una esfera de un tercio de la distancia máxima y volúmenes semiesféricos que cubren los dos tercios y la distancia máxima.
4. Desarrollo del protocolo HRTS con sus niveles y profundidades de acuerdo con el número de saltos.
5. Establecer la comparación de los modelos anteriores con los modelos optimizados para establecer las ventajas y desventajas de estos.

2.3.1 Determinación de distancias entre cada nodo y su suma.

Para establecer las distancias se creó la función distancia, que permite definir la distancia de cada uno de los nodos con los demás. Al nodo que se establece como nodo de referencia se le va asignando una ubicación en una tabla de resultados ordenada ascendentemente de acuerdo con la suma total de las distancias entre los nodos hijos y este nodo de referencia, lo que permite tener una posición final de cada nodo de acuerdo con la distancia total para llegar a la estación base. Esta función posee variables globales (posición nodos y posición final) y variables locales (distancias y suma de distancias), como se puede ver en la Tabla 2.1

:

Tabla 2.1 Posición (numeración) espacial de los nodos y ubicación ascendente en función de la suma de las distancias

Nodo	x (m)	y (m)	z (m)	Nodo	Suma de distancias (m)
1	379	952	248	46	48436,2067
2	319	543	386	58	49596,4541
3	986	251	431	89	49782,2639
4	718	578	830	35	50281,4054
5	413	915	824	2	50566,4737
6	98	895	452	25	51135,1886
7	734	482	380	84	51583,3214
8	637	442	925	19	52313,6474
9	73	311	740	72	52700,2053
10	120	55	737	91	52931,2741
11	981	753	946	79	53222,3261
12	496	131	510	44	53320,6557
13	22	355	791	26	53589,9594
14	53	395	452	64	54634,2909
15	140	885	849	7	55356,5737
16	893	21	390	51	55511,4377
17	465	844	738	66	55518,68
18	560	288	976	37	56212,7576
19	494	250	523	29	56593,9289
20	67	488	429	83	57550,1376
21	897	729	207	12	57808,0036
22	288	202	323	97	58310,6136
23	269	216	110	32	58400,1328
24	594	976	375	88	59156,4704
25	475	593	329	73	59488,7858
26	368	304	342	42	59517,0615
27	655	967	817	22	59548,6717
28	938	895	531	17	59717,7269
29	620	190	521	4	59981,2724
30	282	1	774	20	60457,8628
31	205	711	120	61	60993,6471
32	439	867	625	48	61010,3411
33	27	118	346	82	61124,1586
34	876	39	334	69	61576,5237
35	610	598	574	14	61822,8325
36	203	604	863	87	61991,7508
37	519	516	198	41	62156,3628

Nodo	x (m)	y (m)	z (m)	Nodo	Suma de distancias (m)
38	53	7	672	8	62158,0278
39	862	688	901	62	62371,114
40	442	946	199	78	62547,475
41	548	873	298	36	62647,7012
42	566	113	496	43	62661,7561
43	680	354	889	70	63017,6715
44	371	241	501	75	63339,9832
45	78	560	276	45	63661,7535
46	456	612	533	47	64186,1035
47	47	300	574	92	64300,0822
48	738	798	412	94	64706,7407
49	38	795	14	99	64747,0883
50	954	781	702	60	65241,5509
51	742	351	506	74	65265,9541
52	937	54	381	55	65416,5236
53	513	708	64	76	65702,9194
54	240	992	358	9	65712,207
55	259	162	234	65	65857,0912
56	758	113	203	81	66259,5075
57	993	912	813	77	66298,4774
58	356	481	393	53	66779,5097
59	752	851	53	5	66780,1056
60	110	809	375	24	66915,3551
61	597	186	774	18	66936,7118
62	430	247	165	85	67371,8924
63	730	54	912	31	67485,4814
64	261	608	319	1	67982,5046
65	94	777	329	95	68817,542
66	450	511	204	6	68866,6738
67	640	27	767	71	69073,4177
68	132	990	69	23	69082,9894
69	452	500	950	40	69230,4355
70	652	331	158	13	69490,1622
71	826	173	286	54	70178,8253
72	308	625	687	67	70206,2917
73	402	575	141	39	71365,4538
74	884	750	512	27	71671,3891
75	700	153	721	86	71781,6417
76	241	356	928	56	72042,2394
77	759	143	732	50	72168,3319

Nodo	x (m)	y (m)	z (m)	Nodo	Suma de distancias (m)
78	290	850	749	3	72444,0957
79	277	337	407	80	72535,3927
80	6	275	239	21	72669,1905
81	374	6	520	30	72794,1351
82	436	801	219	15	73474,8967
83	304	497	842	10	73905,3614
84	290	537	662	33	74160,2357
85	242	870	816	28	74280,8006
86	936	722	793	96	76446,7616
87	860	668	469	34	76608,137
88	397	178	309	63	77086,0387
89	479	550	687	16	77368,096
90	564	959	986	90	77748,7271
91	489	596	769	59	77955,7439
92	269	808	829	52	78127,2139
93	989	984	706	38	78356,6014
94	183	885	595	100	79500,8349
95	861	213	752	11	81784,5083
96	32	34	496	57	82900,5534
97	331	451	865	49	83304,3586
98	748	13	68	93	83551,7428
99	644	473	968	98	83616,7433
100	169	951	98	68	84486,8112

En la figura 2.1 se presenta el diagrama de flujo de la función distancia y en el Anexo 1 se encuentra el código fuente de esta.

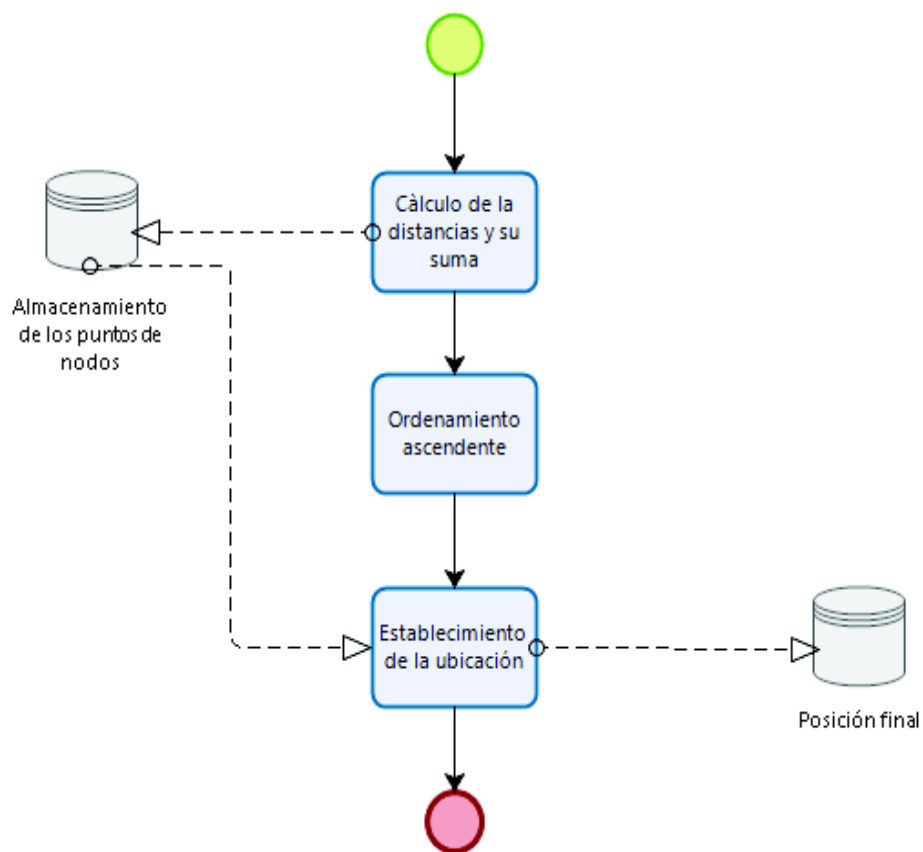


Figura 2.1 Diagrama de flujo de la función distancia

En la figura 2.2 se presenta el diagrama de flujo de la Simulación de la Distribución Volumétrica y en el Anexo 1 se encuentra el código fuente de esta.

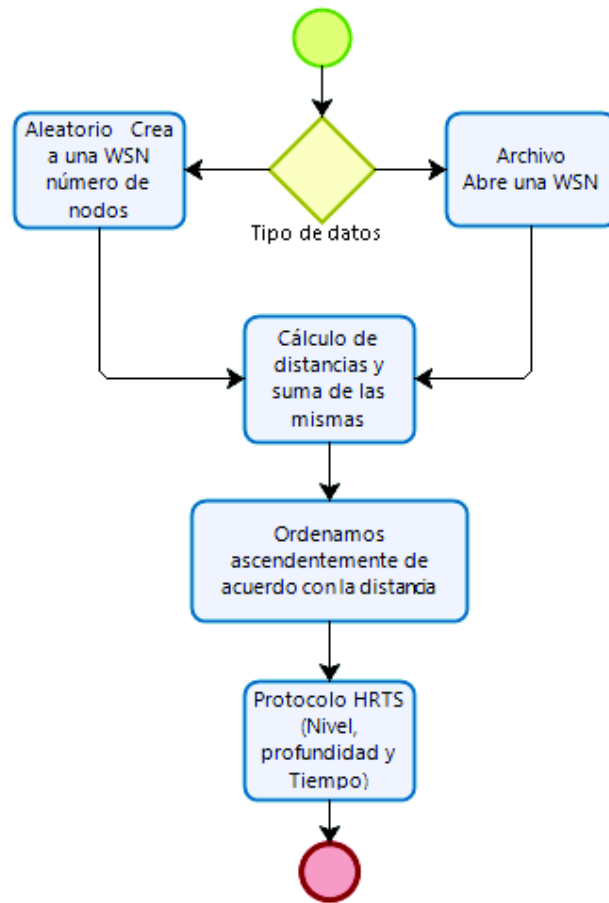


Figura 2.2 Diagrama de Flujo de la Simulación de la Distribución Volumétrica

2.3.2 Establecimiento de los niveles de cada uno de los nodos de acuerdo con su densidad espectral, utilizando una esfera de un tercio de la distancia máxima y volúmenes semiesféricos que cubren los dos tercios y la distancia máxima.

Para establecer los niveles se utilizó esferas y superficies semiesféricas cada un tercio de la distancia máxima que existe entre el nodo de referencia y el nodo más alejado. Al hablar de densidad espectral en esta tesis de grado, se refiere al mayor número de nodos cercanos a la estación base o nodo de referencia. Se establecieron tres niveles:

- Nivel 1: $0 < d \leq \frac{d_{máx}}{3}$
- Nivel 2: $\frac{d_{máx}}{3} < d \leq \frac{2*d_{máx}}{3}$

- Nivel 3: $\frac{2*d_{m\acute{a}x}}{3} < d \leq d_{m\acute{a}x}$

El número de nodos más cercano al nodo de referencia se calcula con el criterio anterior y se establece nuevamente una comparación estadística descendente del mayor al menor número de nodos y se va ordenando sucesivamente en cada columna, que corresponde a cada nivel, para establecer una nueva posición final de los nodos, como se indica en la Tabla 2.2:

Tabla 2.2 Distribución Espectral de niveles

Nodo	Suma distancias	Nivel 1	Nivel 2	Nivel 3
79	53222,3261	21	53	26
22	59548,6717	20	50	30
61	60993,6471	20	55	25
26	53589,9594	18	55	27
78	62547,475	18	54	28
31	67485,4814	18	43	39
72	52700,2053	17	55	28
29	56593,9289	17	52	31
42	59517,0615	17	48	35
14	61822,8325	17	54	29
43	62661,7561	17	58	25
92	64300,0822	17	56	27
55	65416,5236	17	48	35
81	66259,5075	17	47	36
77	66298,4774	17	56	27
18	66936,7118	17	56	27
95	68817,542	17	55	28
33	74160,2357	17	51	32
91	52931,2741	16	55	29
88	59156,4704	16	52	32
17	59717,7269	16	55	29
36	62647,7012	16	57	27
80	72535,3927	16	54	30
96	76446,7616	16	53	31
100	79500,8349	16	45	39
64	54634,2909	15	50	35
32	58400,1328	15	56	29
82	61124,1586	15	45	40
8	62158,0278	15	57	28

Nodo	Suma distancias	Nivel 1	Nivel 2	Nivel 3
62	62371,114	15	50	35
60	65241,5509	15	46	39
65	65857,0912	15	45	40
85	67371,8924	15	55	30
1	67982,5046	15	48	37
23	69082,9894	15	55	30
54	70178,8253	15	43	42
67	70206,2917	15	55	30
19	52313,6474	14	52	34
12	57808,0036	14	51	35
97	58310,6136	14	48	38
69	61576,5237	14	48	38
75	63339,9832	14	60	26
45	63661,7535	14	53	33
47	64186,1035	14	56	30
94	64706,7407	14	49	37
9	65712,207	14	52	34
5	66780,1056	14	57	29
6	68866,6738	14	49	37
58	49596,4541	13	53	34
2	50566,4737	13	53	34
44	53320,6557	13	59	28
37	56212,7576	13	41	46
4	59981,2724	13	52	35
20	60457,8628	13	56	31
40	69230,4355	13	47	40
56	72042,2394	13	40	47
15	73474,8967	13	59	28
16	77368,096	13	41	46
66	55518,68	12	47	41
83	57550,1376	12	52	36
48	61010,3411	12	59	29
99	64747,0883	12	56	32
71	69073,4177	12	44	44
39	71365,4538	12	46	42
27	71671,3891	12	47	41
50	72168,3319	12	49	39
63	77086,0387	12	62	26
52	78127,2139	12	41	47
49	83304,3586	12	47	41
51	55511,4377	11	56	33

Nodo	Suma distancias	Nivel 1	Nivel 2	Nivel 3
73	59488,7858	11	48	41
41	62156,3628	11	50	39
74	65265,9541	11	58	31
53	66779,5097	11	44	45
13	69490,1622	11	50	39
86	71781,6417	11	47	42
3	72444,0957	11	49	40
30	72794,1351	11	51	38
10	73905,3614	11	56	33
28	74280,8006	11	48	41
34	76608,137	11	39	50
90	77748,7271	11	49	40
38	78356,6014	11	57	32
57	82900,5534	11	45	44
93	83551,7428	11	48	41
68	84486,8112	11	47	42
89	49782,2639	10	57	33
84	51583,3214	10	61	29
87	61991,7508	10	59	31
70	63017,6715	10	47	43
76	65702,9194	10	55	35
11	81784,5083	10	42	48
46	48436,2067	9	55	36
25	51135,1886	9	45	46
24	66915,3551	9	53	38
21	72669,1905	8	52	40
59	77955,7439	8	50	42
98	83616,7433	8	44	48
7	55356,5737	7	49	44
35	50281,4054	5	56	39

También se puede observar cómo se distribuye cada uno de los niveles en forma global e independiente en las Figuras 2.3, 2.4 y 2.5:

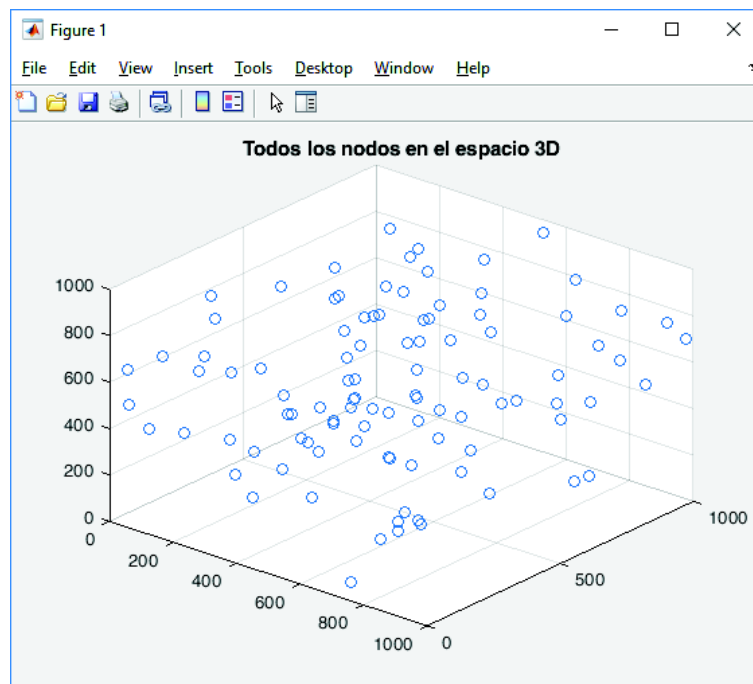


Figura 2.3 Toda la red, con todos los nodos

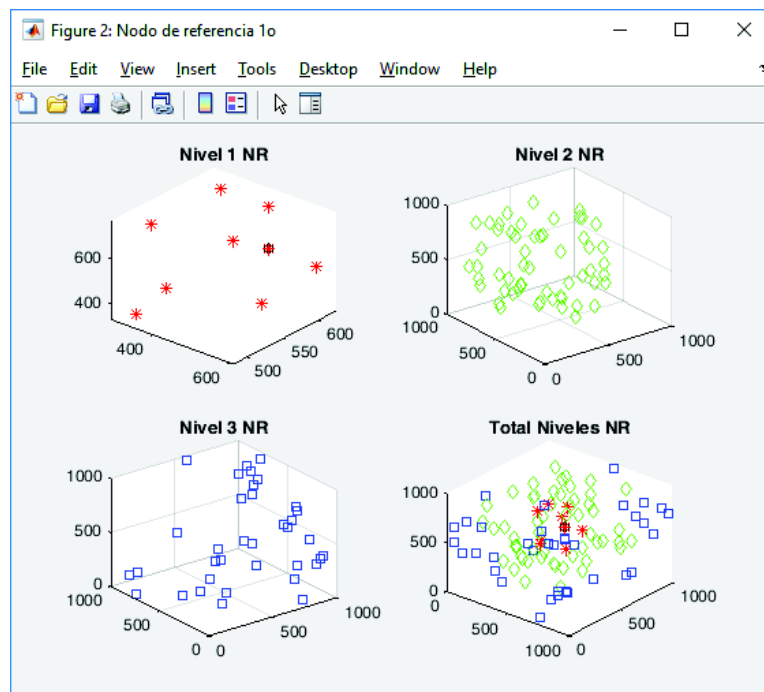


Figura 2.4 Distintos nodos de referencia con su respectiva Distribución Espectral

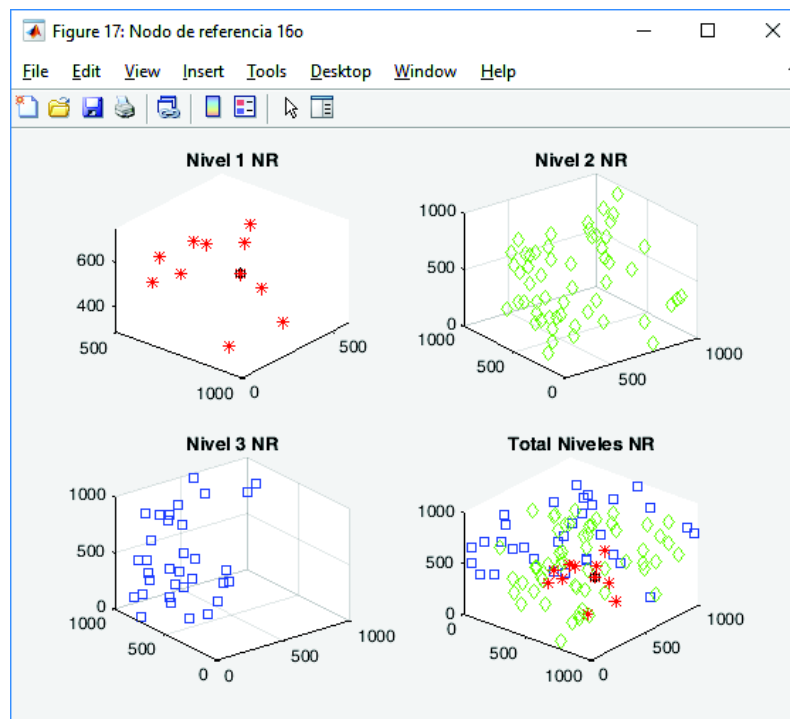
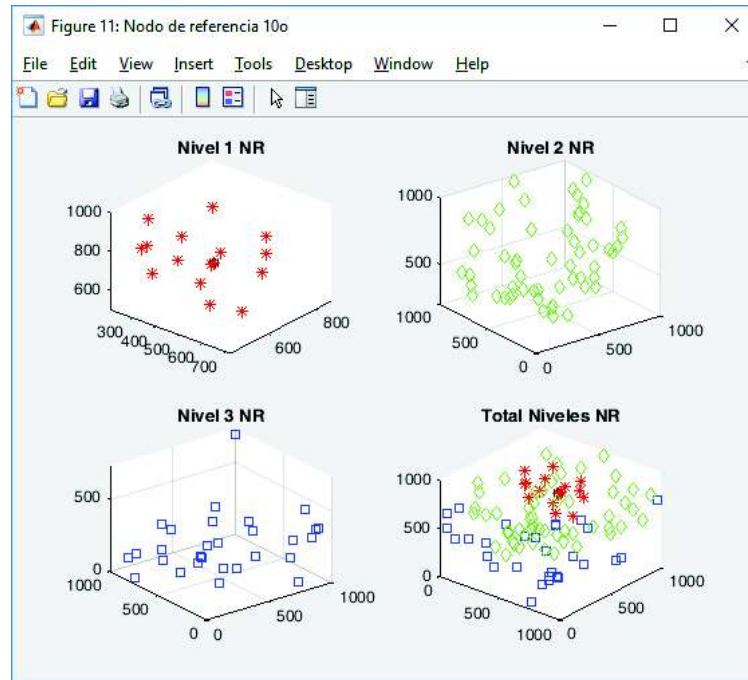


Figura 2.5 Distintos nodos de referencia con su respectiva Distribución Espectral

Esta función posee variables globales (posición nodos, niveles y posición final) y variables locales (distancias y suma de distancias). A continuación, se presenta el diagrama de flujo de dicha función en la Figura 2.6 y en el Anexo 1 se encuentra el código fuente de esta. Además, se almacenan la posición final y los datos totales de los nodos con sus respectivos niveles y tiempos de sincronización.

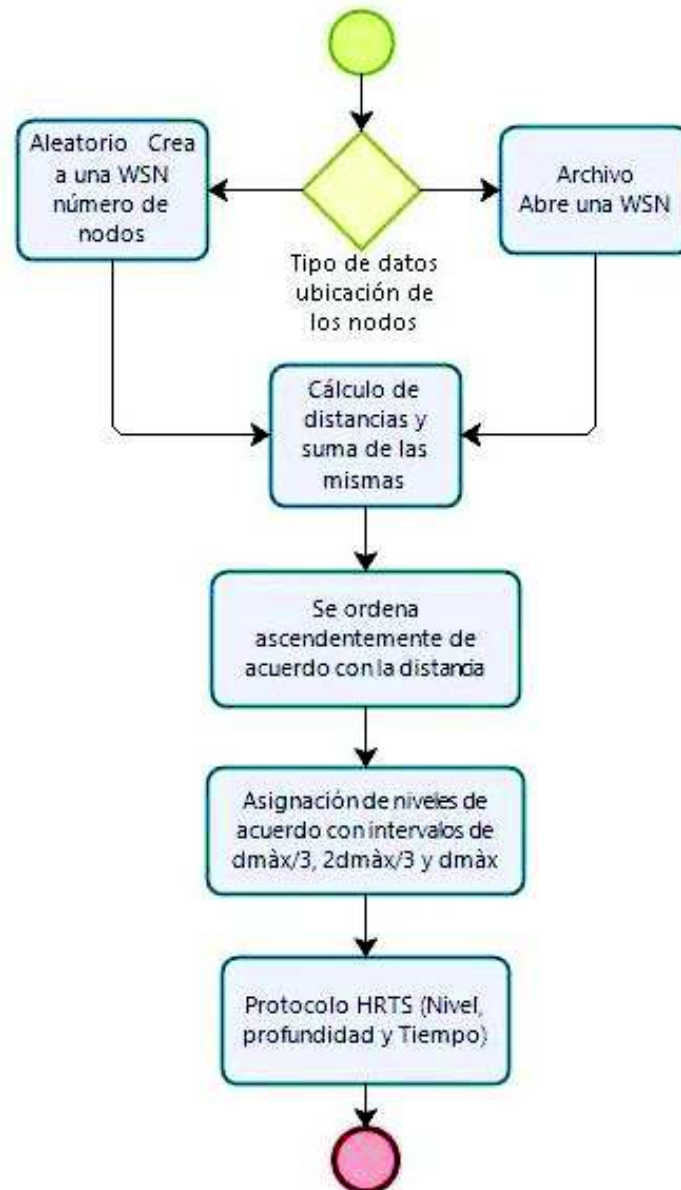


Figura 2.6 Diagrama de flujo de la Distribución Espectral

2.3.3 Desarrollo del protocolo HRTS con sus niveles y profundidades de acuerdo con el número de saltos.

El protocolo HRTS, establece niveles y profundidades de cada uno de los nodos con respecto al nodo de referencia (o a los nodos de referencia). El nivel del nodo es igual al número de saltos que necesita dicho nodo, para llegar a la estación base, esta característica es una función del alcance que tiene cada nodo y determina si requiere uno o más saltos hasta llegar al nodo de referencia. Esta tesis de grado estableció un alcance típico de 100 metros mínimo, que en un lazo se va incrementando en un 30% hasta alcanzar, un nodo vecino más cercano al nodo de referencia y así sucesivamente hasta cumplir el objetivo de comunicarse y sincronizarse con el reloj del nodo de referencia. Al realizar este trabajo, el programa va calculando el número de saltos necesarios y la distancia recorrida, parámetros fundamentales para establecer la sincronización del nodo hijo con el nodo padre; se toma muy en cuenta el tiempo de retardo que se produce en cada nodo para establecer el tiempo de sincronización con la siguiente fórmula:

$$t_r = \frac{d_{total}}{c} + t_{retardo} * n_{saltos}$$

Este script nos proporciona el alcance mínimo de cada nodo, el número de saltos, la distancia total recorrida, y el camino por el que viajan los mensajes desde cada uno de los nodos hasta llegar al nodo de referencia; lo que permite establecer el nivel y la profundidad de cada uno de los elementos de la red como se presenta en la siguiente tabla:

2.3.4 Establecer la comparación de los modelos anteriores con los modelos optimizados para establecer las ventajas y desventajas de estos.

Al empezar esta tesis de grado, las inquietudes sobre los alcances del mismo debían ser adecuada y eficientemente obtenidas para lograr mejorar el tiempo de sincronización, lo que se consiguió con valores muy aceptables que van desde el 40% al 50% de optimización.

Al ser una simulación se ha tomado ciertas consideraciones de carácter ideal al asumir: la velocidad de la luz, pérdidas y tiempo de propagación constantes y ciertos artilugios para

A continuación, se presenta el diagrama de flujo de dicha función en la Figura 2.7 y en el Anexo 1 se encuentra el código fuente de esta.

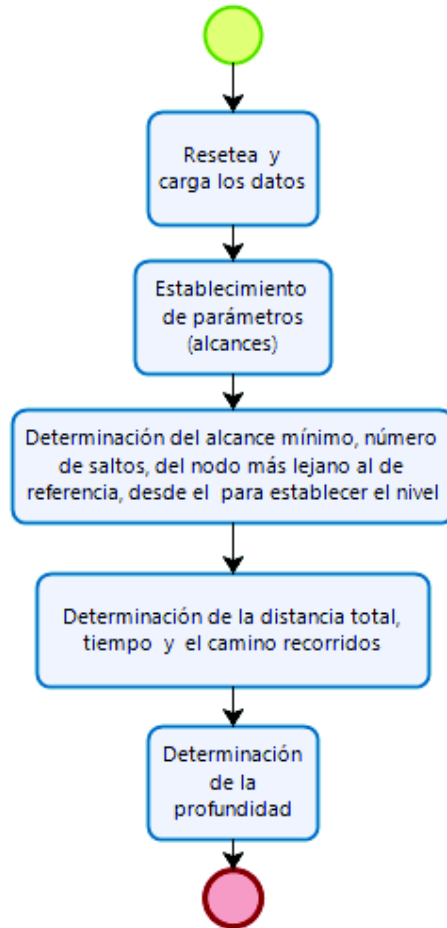


Figura 2.7 Diagrama de Flujo de la Simulación del Protocolo HRTS

conseguir nuestro objetivo. Para futuros proyectos esta simulación provee datos que pueden ser la base de nuevas investigaciones al simular una red inalámbrica de sensores en tres dimensiones (que no existía anteriormente), solo se requiere la ubicación de los nodos y el programa entrega datos importantes como cuales serían los nodos de referencia más apropiados para establecer las estaciones base, se podría utilizar las herramientas de cloud computing para agrupar y crear subredes que cubran toda la red para reducir el consumo de energía, la vida útil de los sensores remotos y de la red en general.

Al establecer canales de sincronización independientes se consigue que los datos y la transmisión de señales de control no se vean afectados en el caso de existir un excesivo tráfico o colisiones en la red.

3 RESULTADOS Y DISCUSIÓN

Los resultados de la investigación son presentados de manera lógica y objetiva, utilizando cuadros comparativos de datos obtenidos en la simulación del protocolo original y de los tres tipos de mejoras analizadas en esta investigación por distribución volumétrica, distribución espectral y optimización del número de saltos.

En primer lugar, se analiza la distribución volumétrica y espectral de la red WSN, calculando la distancia, la suma de las distancias de cada uno de los nodos, para ordenarlos en forma descendente y ver cuáles nodos son los que se toman como estaciones base. Además, la distribución espectral de acuerdo con la concentración de nodos en los tres niveles establecidos y estudiados como se observa en las Figuras 3.1 y 3.2:

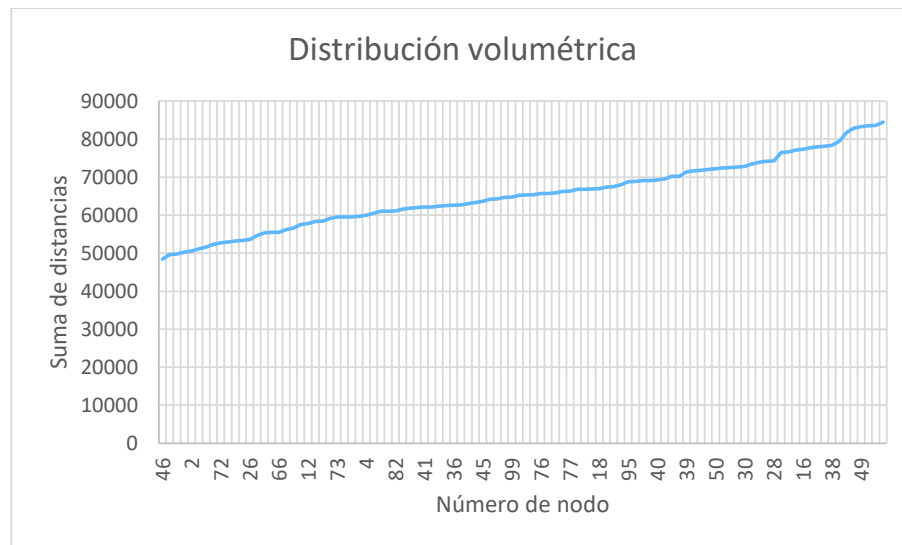


Figura 3.1 Distribución volumétrica

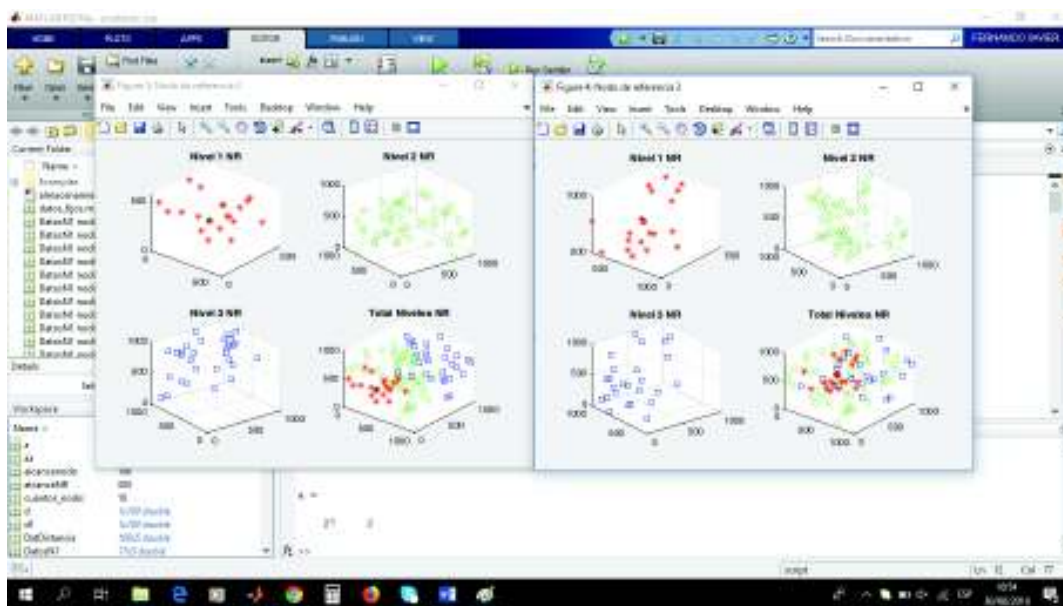
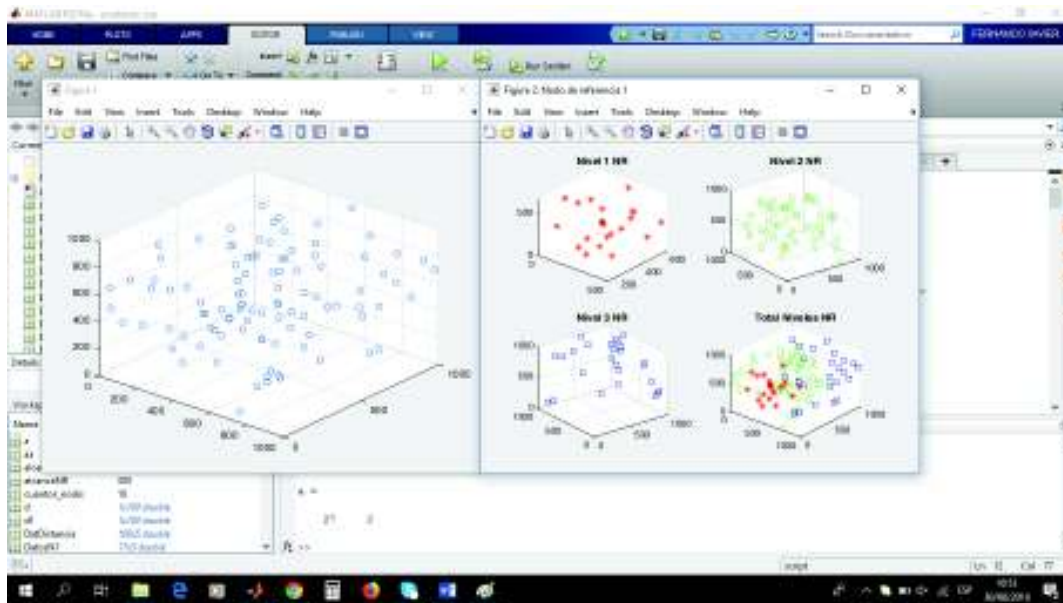


Figura 3.2 Distribución Espectral

Con la información de la posición de los nodos y los alcances, se hicieron dos programas principales, uno con la distribución volumétrica, y otro con la espectral, que calculan el tiempo como si el nodo de referencia tuviera un alcance para toda la red como se observa en la Tabla 3.1 y en la Tabla 3.2 respectivamente:

Tabla 3.1 HRTS Distribución Volumétrica

DISTRIBUCIÓN VOLUMÉTRICA nodis4					ORIGINAL nodis16					DISTRIBUCIÓN VOLUMÉTRICA nodis1				
NODO 26	Distancia recorrida (m)	Nivel	Profundidad	Tiempo (s)	NODO 18	Distancia recorrida (m)	Nivel	Profundidad	Tiempo (s)	NODO 79	Distancia recorrida (m)	Nivel	Profundidad	Tiempo (s)
26	0,000	0	0	0,0000E+00	18	0,000	0	0	0,0000E+00	79	0,000	0	0	0,0000E+00
79	116,598	1	0	1,0039E-04	43	162,250	1	0	1,0054E-04	26	116,598	1	0	1,0039E-04
22	131,015	1	-1	1,0044E-04	8	266,600	2	0	2,0089E-04	22	159,380	1	-1	1,0053E-04
44	171,053	1	-2	1,0057E-04	61	382,109	2	-1	2,0127E-04	44	163,976	1	-2	1,0055E-04
58	184,591	1	-3	1,0062E-04	99	320,070	3	0	3,0107E-04	58	164,842	1	-3	1,0055E-04
55	232,809	2	0	2,0078E-04	69	462,084	3	-1	3,0154E-04	14	235,722	1	-4	1,0079E-04
88	243,501	2	-1	2,0081E-04	75	502,554	3	-2	3,0168E-04	2	237,382	2	0	2,0079E-04
2	257,131	2	-2	2,0086E-04	63	614,825	3	-3	3,0205E-04	55	261,174	2	-1	2,0087E-04
19	296,328	2	-3	2,0099E-04	19	660,867	3	-4	3,0220E-04	88	271,866	2	-2	2,0091E-04
14	352,320	2	-4	2,0118E-04	30	747,417	3	-5	3,0249E-04	19	289,251	2	-3	2,0096E-04
66	397,798	2	-5	2,0133E-04	4	508,604	4	0	4,0170E-04	20	332,542	2	-4	2,0111E-04
46	400,834	2	-6	2,0134E-04	77	563,398	4	-1	4,0188E-04	66	378,049	2	-5	2,0126E-04
64	367,031	3	0	3,0122E-04	97	617,863	4	-2	4,0206E-04	46	381,085	2	-6	2,0127E-04
23	368,426	3	-1	3,0123E-04	67	649,496	4	-3	4,0217E-04	47	390,464	2	-7	2,0130E-04
62	406,553	3	-2	3,0136E-04	91	670,281	4	-4	4,0224E-04	80	484,676	2	-8	2,0162E-04
12	416,053	3	-3	3,0139E-04	12	780,591	4	-5	4,0260E-04	64	347,282	3	0	3,0116E-04
20	449,139	3	-4	3,0150E-04	44	786,143	4	-6	4,0262E-04	23	396,791	3	-1	3,0132E-04
37	467,239	3	-5	3,0156E-04	29	800,438	4	-7	4,0267E-04	12	408,976	3	-2	3,0136E-04
73	499,626	3	-6	3,0167E-04	10	922,142	4	-8	4,0308E-04	62	434,918	3	-3	3,0145E-04
47	507,062	3	-7	3,0169E-04	83	675,949	5	0	5,0225E-04	37	447,489	3	-4	3,0149E-04
35	560,812	3	-8	3,0187E-04	95	688,714	5	-1	5,0230E-04	73	479,877	3	-5	3,0160E-04
89	568,432	3	-9	3,0190E-04	39	703,224	5	-2	5,0235E-04	45	501,994	3	-6	3,0167E-04
61	575,086	3	-10	3,0192E-04	76	763,101	5	-3	5,0255E-04	84	514,966	3	-7	3,0172E-04
80	601,274	3	-11	3,0201E-04	89	764,833	5	-4	5,0255E-04	25	529,621	3	-8	3,0177E-04
84	624,043	3	-12	3,0208E-04	42	897,752	5	-5	5,0299E-04	35	541,063	3	-9	3,0180E-04
32	672,455	3	-13	3,0224E-04	79	950,118	5	-6	5,0317E-04	9	558,848	3	-10	3,0186E-04
42	489,674	4	0	4,0163E-04	81	955,546	5	-7	5,0319E-04	61	568,009	3	-11	3,0189E-04
81	591,007	4	-1	4,0197E-04	51	1002,996	5	-8	5,0335E-04	32	652,706	3	-12	3,0218E-04
45	618,591	4	-2	4,0206E-04	38	1027,109	5	-9	5,0343E-04	33	675,828	3	-13	3,0225E-04
25	625,435	4	-3	4,0209E-04	36	824,579	6	0	6,0275E-04	42	482,597	4	0	4,0161E-04

70	644,017	4	-4	4,0215E-04
91	662,984	4	-5	4,0221E-04
9	675,445	4	-6	4,0225E-04
53	689,202	4	-7	4,0230E-04
75	695,531	4	-8	4,0232E-04
72	717,279	4	-9	4,0239E-04
31	739,930	4	-10	4,0247E-04
17	790,667	4	-11	4,0264E-04
33	792,425	4	-12	4,0264E-04
43	794,945	4	-13	4,0265E-04
83	808,964	4	-14	4,0270E-04
29	586,988	5	0	5,0196E-04
77	756,375	5	-1	5,0252E-04
13	759,932	5	-2	5,0253E-04
67	842,473	5	-3	5,0281E-04
97	867,051	5	-4	5,0289E-04
82	885,679	5	-5	5,0295E-04
56	890,563	5	-6	5,0297E-04
8	899,296	5	-7	5,0300E-04
7	911,796	5	-8	5,0304E-04
5	913,716	5	-9	5,0305E-04
10	935,741	5	-10	5,0312E-04
49	954,828	5	-11	5,0318E-04
36	957,595	5	-12	5,0319E-04
96	964,417	5	-13	5,0322E-04
78	966,115	5	-14	5,0322E-04
100	983,610	5	-15	5,0328E-04
65	985,609	5	-16	5,0329E-04
95	881,691	6	0	6,0294E-04
99	952,765	6	-1	6,0318E-04
76	1012,288	6	-2	6,0338E-04
71	1013,498	6	-3	6,0338E-04
63	1015,257	6	-4	6,0339E-04
69	1022,829	6	-5	6,0341E-04
41	1040,498	6	-6	6,0347E-04
38	1040,708	6	-7	6,0347E-04

86	838,486	6	-1	6,0280E-04
11	846,091	6	-2	6,0282E-04
84	860,871	6	-3	6,0287E-04
35	944,371	6	-4	6,0315E-04
9	1019,212	6	-5	6,0340E-04
26	1066,716	6	-6	6,0356E-04
58	1114,960	6	-7	6,0372E-04
7	1184,933	6	-8	6,0395E-04
96	1206,402	6	-9	6,0402E-04
3	1277,151	6	-10	6,0426E-04
50	948,423	7	0	7,0316E-04
72	954,107	7	-1	7,0318E-04
13	1103,699	7	-2	7,0368E-04
46	1104,349	7	-3	7,0368E-04
2	1187,500	7	-4	7,0396E-04
47	1187,595	7	-5	7,0396E-04
88	1200,155	7	-6	7,0400E-04
25	1290,463	7	-7	7,0430E-04
33	1378,393	7	-8	7,0460E-04
87	1426,579	7	-9	7,0476E-04
16	1528,607	7	-10	7,0510E-04
57	1124,499	8	0	8,0375E-04
17	1228,353	8	-1	8,0410E-04
64	1297,400	8	-2	8,0433E-04
22	1312,640	8	-3	8,0438E-04
14	1342,337	8	-4	8,0448E-04
62	1363,207	8	-5	8,0455E-04
66	1442,034	8	-6	8,0481E-04
74	1522,230	8	-7	8,0508E-04
80	1569,545	8	-8	8,0524E-04
52	1584,339	8	-9	8,0528E-04
48	1613,750	8	-10	8,0538E-04
21	1698,120	8	-11	8,0566E-04
93	1253,531	9	0	9,0418E-04
32	1346,565	9	-1	9,0449E-04
5	1351,402	9	-2	9,0451E-04

81	583,930	4	-1	4,0195E-04
72	608,202	4	-2	4,0203E-04
13	643,334	4	-3	4,0215E-04
53	669,453	4	-4	4,0223E-04
70	672,381	4	-5	4,0224E-04
75	688,454	4	-6	4,0230E-04
83	699,888	4	-7	4,0233E-04
31	720,181	4	-8	4,0240E-04
82	721,364	4	-9	4,0241E-04
17	770,918	4	-10	4,0257E-04
43	787,868	4	-11	4,0263E-04
10	819,143	4	-12	4,0273E-04
78	847,298	4	-13	4,0283E-04
96	847,819	4	-14	4,0283E-04
29	579,911	5	0	5,0193E-04
77	749,298	5	-1	5,0250E-04
97	757,974	5	-2	5,0253E-04
89	794,927	5	-3	5,0265E-04
67	835,396	5	-4	5,0279E-04
36	848,518	5	-5	5,0283E-04
41	876,183	5	-6	5,0292E-04
1	885,349	5	-7	5,0295E-04
8	892,219	5	-8	5,0298E-04
5	893,967	5	-9	5,0298E-04
56	918,928	5	-10	5,0307E-04
38	924,110	5	-11	5,0308E-04
85	932,110	5	-12	5,0311E-04
49	935,079	5	-13	5,0312E-04
92	940,061	5	-14	5,0314E-04
18	950,118	5	-15	5,0317E-04
100	963,861	5	-16	5,0322E-04
65	965,860	5	-17	5,0322E-04
30	993,869	5	-18	5,0332E-04
51	782,469	6	0	6,0261E-04
95	874,614	6	-1	6,0292E-04
91	889,478	6	-2	6,0297E-04

60	1043,884	6	-8	6,0348E-04
68	1044,692	6	-9	6,0348E-04
1	1049,663	6	-10	6,0350E-04
85	1050,927	6	-11	6,0351E-04
98	1058,863	6	-12	6,0353E-04
92	1058,878	6	-13	6,0353E-04
18	1078,867	6	-14	6,0360E-04
4	1083,909	6	-15	6,0362E-04
51	1093,733	6	-16	6,0365E-04
30	1110,467	6	-17	6,0370E-04
27	1161,339	6	-18	6,0387E-04
40	1129,701	7	0	7,0377E-04
15	1159,176	7	-1	7,0387E-04
6	1159,940	7	-2	7,0387E-04
34	1164,362	7	-3	7,0388E-04
24	1177,077	7	-4	7,0393E-04
54	1231,380	7	-5	7,0411E-04
48	1274,423	7	-6	7,0425E-04
90	1353,448	7	-7	7,0451E-04
16	1225,591	8	0	8,0409E-04
52	1242,816	8	-1	8,0415E-04
94	1326,595	8	-2	8,0443E-04
87	1461,594	8	-3	8,0488E-04
3	1451,885	9	0	9,0484E-04
74	1557,244	9	-1	9,0519E-04
21	1733,134	9	-2	9,0578E-04
28	1713,135	10	0	1,0057E-03
50	1762,088	10	-1	1,0059E-03
59	1977,316	10	-2	1,0066E-03
86	1872,025	11	0	1,1062E-03
93	1968,122	11	-1	1,1066E-03
39	2007,287	12	0	1,2067E-03
57	2097,153	12	-1	1,2070E-03
11	2150,154	13	0	1,3072E-03
	90275,49			5,2601E-02

55	1414,434	9	-3	9,0472E-04
20	1439,157	9	-4	9,0480E-04
92	1447,426	9	-5	9,0483E-04
37	1511,475	9	-6	9,0504E-04
73	1543,863	9	-7	9,0515E-04
34	1662,792	9	-8	9,0555E-04
28	1678,121	9	-9	9,0560E-04
71	1772,772	9	-10	9,0591E-04
41	1847,676	9	-11	9,0616E-04
59	1942,302	9	-12	9,0648E-04
78	1540,189	10	0	1,0051E-03
23	1550,051	10	-1	1,0052E-03
90	1577,192	10	-2	1,0053E-03
45	1608,609	10	-3	1,0054E-03
53	1733,438	10	-4	1,0058E-03
56	1895,706	10	-5	1,0063E-03
24	1984,255	10	-6	1,0066E-03
82	2002,495	10	-7	1,0067E-03
85	1625,001	11	0	1,1054E-03
94	1730,951	11	-1	1,1058E-03
27	1769,301	11	-2	1,1059E-03
98	2064,007	11	-3	1,1069E-03
70	2142,253	11	-4	1,1071E-03
15	1733,251	12	0	1,2058E-03
60	1974,887	12	-1	1,2066E-03
65	2033,163	13	0	1,3068E-03
6	2090,943	13	-1	1,3070E-03
31	2278,841	14	0	1,4076E-03
54	2286,926	14	-1	1,4076E-03
1	2468,642	15	0	1,5082E-03
49	2493,739	15	-1	1,5083E-03
100	2522,521	15	-2	1,5084E-03
40	2548,680	16	0	1,6085E-03
68	2583,603	16	-1	1,6086E-03
	128889,33			7,5930E-02

76	903,212	6	-3	6,0301E-04
69	913,753	6	-4	6,0305E-04
99	945,688	6	-5	6,0315E-04
40	965,386	6	-6	6,0322E-04
63	1008,180	6	-7	6,0336E-04
24	1012,763	6	-8	6,0338E-04
60	1024,135	6	-9	6,0342E-04
68	1024,943	6	-10	6,0342E-04
15	1040,359	6	-11	6,0347E-04
71	1041,863	6	-12	6,0348E-04
54	1067,066	6	-13	6,0356E-04
4	1076,832	6	-14	6,0359E-04
98	1087,228	6	-15	6,0363E-04
48	1110,109	6	-16	6,0370E-04
27	1141,589	6	-17	6,0381E-04
7	964,406	7	0	7,0322E-04
6	1140,191	7	-1	7,0380E-04
34	1192,727	7	-2	7,0398E-04
94	1268,071	7	-3	7,0423E-04
3	1271,447	7	-4	7,0424E-04
39	1271,452	7	-5	7,0424E-04
87	1297,280	7	-6	7,0433E-04
90	1333,699	7	-7	7,0445E-04
16	1253,956	8	0	8,0418E-04
52	1271,180	8	-1	8,0424E-04
74	1392,930	8	-2	8,0465E-04
86	1406,715	8	-3	8,0469E-04
11	1414,319	8	-4	8,0472E-04
21	1568,820	8	-5	8,0523E-04
50	1516,651	9	0	9,0506E-04
28	1548,821	9	-1	9,0517E-04
59	1813,002	9	-2	9,0605E-04
57	1692,728	10	0	1,0056E-03
93	1821,759	11	0	1,1061E-03
	82475,76			4,8275E-02

Tabla 3.2 HRTS Distribución Espectral

DISTRIBUCIÓN ESPECTRAL nodis10					ORIGINAL nodis16					DISTRIBUCIÓN ESPECTRAL nodis40				
NODO 46	Distancia recorrida (m)	Nivel	Profundidad	Tiempo (s)	NODO 18	Distancia recorrida (m)	Nivel	Profundidad	Tiempo (s)	NODO 35	Distancia recorrida (m)	Nivel	Profundidad	Tiempo (s)
46	0,000	0	0	0,0000E+00	18	0,000	0	0	0,0000E+00	35	0,000	0	0	0,0000E+00
35	159,978	1	0	1,0053E-04	43	162,250	1	0	1,0054E-04	46	159,978	1	0	1,0053E-04
89	167,598	1	-1	1,0056E-04	8	266,600	2	0	2,0089E-04	89	179,538	1	-1	1,0060E-04
25	205,762	1	-2	1,0069E-04	61	382,109	2	-1	2,0127E-04	7	257,814	1	-2	1,0086E-04
32	271,621	1	-3	1,0091E-04	99	320,070	3	0	3,0107E-04	91	274,090	2	0	2,0091E-04
91	262,149	2	0	2,0087E-04	69	462,084	3	-1	3,0154E-04	25	365,740	2	-1	2,0122E-04
72	354,322	2	-1	2,0118E-04	75	502,554	3	-2	3,0168E-04	72	366,263	2	-2	2,0122E-04
66	357,334	2	-2	2,0119E-04	63	614,825	3	-3	3,0205E-04	32	431,599	2	-3	2,0144E-04
2	379,212	2	-3	2,0126E-04	19	660,867	3	-4	3,0220E-04	51	439,751	2	-4	2,0147E-04
17	389,833	2	-4	2,0130E-04	30	747,417	3	-5	3,0249E-04	87	499,460	2	-5	2,0167E-04
7	417,792	2	-5	2,0139E-04	4	508,604	4	0	4,0170E-04	84	459,499	3	0	3,0153E-04
48	447,429	2	-6	2,0149E-04	77	563,398	4	-1	4,0188E-04	4	511,758	3	-1	3,0171E-04
37	426,774	3	0	3,0142E-04	97	617,863	4	-2	4,0206E-04	37	523,936	3	-2	3,0175E-04
84	447,558	3	0	3,0149E-04	67	649,496	4	-3	4,0217E-04	58	541,242	3	-3	3,0181E-04
58	451,752	3	0	3,0151E-04	91	670,281	4	-4	4,0224E-04	17	549,811	3	-4	3,0183E-04
73	459,162	3	0	3,0153E-04	12	780,591	4	-5	4,0260E-04	74	595,111	3	-5	3,0199E-04
64	489,112	3	0	3,0163E-04	44	786,143	4	-6	4,0262E-04	48	686,631	3	-6	3,0229E-04
4	499,817	3	0	3,0167E-04	29	800,438	4	-7	4,0267E-04	19	708,068	3	-7	3,0236E-04
5	512,882	3	0	3,0171E-04	10	922,142	4	-8	4,0308E-04	3	713,906	3	-8	3,0238E-04
78	565,281	3	0	3,0189E-04	83	675,949	5	0	5,0225E-04	21	771,000	3	-9	3,0257E-04
51	599,729	3	0	3,0200E-04	95	688,714	5	-1	5,0230E-04	66	593,377	4	0	4,0198E-04
87	634,600	3	0	3,0212E-04	39	703,224	5	-2	5,0235E-04	2	613,782	4	-1	4,0205E-04
20	640,704	3	0	3,0214E-04	76	763,101	5	-3	5,0255E-04	83	644,421	4	-2	4,0215E-04
41	681,354	3	0	3,0227E-04	89	764,833	5	-4	5,0255E-04	5	672,860	4	-3	4,0224E-04
79	616,594	4	0	4,0206E-04	42	897,752	5	-5	5,0299E-04	8	696,371	4	-4	4,0232E-04
83	632,480	4	-1	4,0211E-04	79	950,118	5	-6	5,0317E-04	78	725,259	4	-5	4,0242E-04
53	648,738	4	-2	4,0216E-04	81	955,546	5	-7	5,0319E-04	26	725,834	4	-6	4,0242E-04
85	650,092	4	-3	4,0217E-04	51	1002,996	5	-8	5,0335E-04	28	751,002	4	-7	4,0251E-04
92	658,044	4	-4	4,0219E-04	38	1027,109	5	-9	5,0343E-04	70	755,267	4	-8	4,0252E-04
70	658,105	4	-5	4,0220E-04	36	824,579	6	0	6,0275E-04	27	789,540	4	-9	4,0263E-04
8	684,430	4	-6	4,0228E-04	86	838,486	6	-1	6,0280E-04	50	799,955	4	-10	4,0267E-04
39	694,437	4	-7	4,0232E-04	11	846,091	6	-2	6,0282E-04	12	827,792	4	-11	4,0276E-04

74	730,250	4	-8	4,0244E-04
14	737,524	4	-9	4,0246E-04
94	756,042	4	-10	4,0252E-04
27	760,504	4	-11	4,0254E-04
45	810,156	4	-12	4,0270E-04
24	817,934	4	-13	4,0273E-04
82	836,173	4	-14	4,0279E-04
21	906,140	4	-15	4,0302E-04
97	690,566	5	0	5,0230E-04
26	733,192	5	-1	5,0245E-04
99	737,900	5	-2	5,0246E-04
15	758,342	5	-3	5,0253E-04
22	775,974	5	-4	5,0259E-04
36	781,110	5	-5	5,0261E-04
43	788,781	5	-6	5,0263E-04
11	837,304	5	-7	5,0279E-04
18	864,002	5	-8	5,0288E-04
19	877,581	5	-9	5,0293E-04
28	886,141	5	-10	5,0296E-04
47	892,265	5	-11	5,0298E-04
71	925,732	5	-12	5,0309E-04
50	935,094	5	-13	5,0312E-04
90	952,614	5	-14	5,0318E-04
80	986,477	5	-15	5,0329E-04
60	999,979	5	-16	5,0334E-04
1	1000,158	5	-17	5,0334E-04
59	1150,322	5	-18	5,0384E-04
76	835,804	6	0	6,0279E-04
69	846,345	6	-1	6,0282E-04
55	877,768	6	-2	6,0293E-04
88	888,460	6	-3	6,0296E-04
12	997,306	6	-4	6,0333E-04
44	1002,857	6	-5	6,0335E-04
61	1008,640	6	-6	6,0336E-04
29	1017,152	6	-7	6,0339E-04
86	1045,031	6	-8	6,0349E-04
56	1048,667	6	-9	6,0350E-04
65	1058,254	6	-10	6,0353E-04

84	860,871	6	-3	6,0287E-04
35	944,371	6	-4	6,0315E-04
9	1019,212	6	-5	6,0340E-04
26	1066,716	6	-6	6,0356E-04
58	1114,960	6	-7	6,0372E-04
7	1184,933	6	-8	6,0395E-04
96	1206,402	6	-9	6,0402E-04
3	1277,151	6	-10	6,0426E-04
50	948,423	7	0	7,0316E-04
72	954,107	7	-1	7,0318E-04
13	1103,699	7	-2	7,0368E-04
46	1104,349	7	-3	7,0368E-04
2	1187,500	7	-4	7,0396E-04
47	1187,595	7	-5	7,0396E-04
88	1200,155	7	-6	7,0400E-04
25	1290,463	7	-7	7,0430E-04
33	1378,393	7	-8	7,0460E-04
87	1426,579	7	-9	7,0476E-04
16	1528,607	7	-10	7,0510E-04
57	1124,499	8	0	8,0375E-04
17	1228,353	8	-1	8,0410E-04
64	1297,400	8	-2	8,0433E-04
22	1312,640	8	-3	8,0438E-04
14	1342,337	8	-4	8,0448E-04
62	1363,207	8	-5	8,0455E-04
66	1442,034	8	-6	8,0481E-04
74	1522,230	8	-7	8,0508E-04
80	1569,545	8	-8	8,0524E-04
52	1584,339	8	-9	8,0528E-04
48	1613,750	8	-10	8,0538E-04
21	1698,120	8	-11	8,0566E-04
93	1253,531	9	0	9,0418E-04
32	1346,565	9	-1	9,0449E-04
5	1351,402	9	-2	9,0451E-04
55	1414,434	9	-3	9,0472E-04
20	1439,157	9	-4	9,0480E-04
92	1447,426	9	-5	9,0483E-04
37	1511,475	9	-6	9,0504E-04

44	833,343	4	-12	4,0278E-04
29	847,638	4	-13	4,0283E-04
41	920,557	4	-14	4,0307E-04
71	943,490	4	-15	4,0315E-04
59	1015,183	4	-16	4,0339E-04
73	695,205	5	0	5,0232E-04
97	702,507	5	-1	5,0234E-04
64	723,682	5	-2	5,0241E-04
99	749,840	5	-3	5,0250E-04
36	793,051	5	-4	5,0265E-04
43	800,721	5	-5	5,0267E-04
85	810,070	5	-6	5,0270E-04
92	818,022	5	-7	5,0273E-04
69	827,661	5	-8	5,0276E-04
79	842,431	5	-9	5,0281E-04
88	859,273	5	-10	5,0287E-04
20	875,274	5	-11	5,0292E-04
18	875,943	5	-12	5,0292E-04
86	909,891	5	-13	5,0304E-04
94	916,020	5	-14	5,0306E-04
42	944,952	5	-15	5,0315E-04
57	976,031	5	-16	5,0326E-04
90	981,649	5	-17	5,0327E-04
81	1002,747	5	-18	5,0334E-04
24	1057,136	5	-19	5,0353E-04
56	1066,425	5	-20	5,0356E-04
82	1075,376	5	-21	5,0359E-04
52	1131,923	5	-22	5,0378E-04
76	847,744	6	0	6,0283E-04
53	884,781	6	-1	6,0295E-04
15	918,320	6	-2	6,0306E-04
22	971,758	6	-3	6,0324E-04
14	972,093	6	-4	6,0324E-04
61	1020,580	6	-5	6,0340E-04
62	1022,325	6	-6	6,0341E-04
45	1044,726	6	-7	6,0348E-04
39	1045,154	6	-8	6,0349E-04
93	1105,062	6	-9	6,0369E-04

9	1060,649	6	-11	6,0354E-04
34	1076,596	6	-12	6,0359E-04
40	1080,196	6	-13	6,0360E-04
6	1116,035	6	-14	6,0372E-04
93	1141,128	6	-15	6,0381E-04
3	1155,316	6	-16	6,0385E-04
33	1177,629	6	-17	6,0393E-04
54	1181,875	6	-18	6,0394E-04
23	1013,385	7	0	7,0338E-04
62	1051,512	7	-1	7,0351E-04
42	1114,466	7	-2	7,0372E-04
75	1129,085	7	-3	7,0377E-04
16	1137,825	7	-4	7,0380E-04
13	1145,135	7	-5	7,0382E-04
52	1155,050	7	-6	7,0385E-04
81	1172,260	7	-7	7,0391E-04
98	1216,967	7	-8	7,0406E-04
57	1270,159	7	-9	7,0424E-04
31	1303,933	7	-10	7,0435E-04
96	1349,620	7	-11	7,0450E-04
77	1189,929	8	0	8,0397E-04
67	1276,027	8	-1	8,0426E-04
30	1442,454	8	-2	8,0481E-04
49	1518,830	8	-3	8,0507E-04
100	1547,613	8	-4	8,0516E-04
95	1315,244	9	0	9,0439E-04
63	1448,810	9	-1	9,0483E-04
68	1608,695	9	-2	9,0537E-04
10	1617,180	9	-3	9,0539E-04
38	1722,146	10	0	1,0057E-03
	86354,97			4,9988E-02

73	1543,863	9	-7	9,0515E-04
34	1662,792	9	-8	9,0555E-04
28	1678,121	9	-9	9,0560E-04
71	1772,772	9	-10	9,0591E-04
41	1847,676	9	-11	9,0616E-04
59	1942,302	9	-12	9,0648E-04
78	1540,189	10	0	1,0051E-03
23	1550,051	10	-1	1,0052E-03
90	1577,192	10	-2	1,0053E-03
45	1608,609	10	-3	1,0054E-03
53	1733,438	10	-4	1,0058E-03
56	1895,706	10	-5	1,0063E-03
24	1984,255	10	-6	1,0066E-03
82	2002,495	10	-7	1,0067E-03
85	1625,001	11	0	1,1054E-03
94	1730,951	11	-1	1,1058E-03
27	1769,301	11	-2	1,1059E-03
98	2064,007	11	-3	1,1069E-03
70	2142,253	11	-4	1,1071E-03
15	1733,251	12	0	1,2058E-03
60	1974,887	12	-1	1,2066E-03
65	2033,163	13	0	1,3068E-03
6	2090,943	13	-1	1,3070E-03
31	2278,841	14	0	1,4076E-03
54	2286,926	14	-1	1,4076E-03
1	2468,642	15	0	1,5082E-03
49	2493,739	15	-1	1,5083E-03
100	2522,521	15	-2	1,5084E-03
40	2548,680	16	0	1,6085E-03
68	2583,603	16	-1	1,6086E-03
	128889,330			7,5930E-02

60	1159,957	6	-10	6,0387E-04
34	1210,377	6	-11	6,0404E-04
98	1234,725	6	-12	6,0412E-04
1	1239,361	6	-13	6,0413E-04
30	1272,941	6	-14	6,0425E-04
55	1073,552	7	0	7,0358E-04
9	1103,856	7	-1	7,0368E-04
47	1126,835	7	-2	7,0376E-04
75	1141,025	7	-3	7,0381E-04
11	1188,021	7	-4	7,0396E-04
65	1218,232	7	-5	7,0406E-04
80	1221,047	7	-6	7,0407E-04
16	1271,606	7	-7	7,0424E-04
6	1276,013	7	-8	7,0426E-04
40	1319,398	7	-9	7,0440E-04
54	1421,077	7	-10	7,0474E-04
10	1447,667	7	-11	7,0483E-04
13	1188,342	8	0	8,0396E-04
77	1201,869	8	-1	8,0401E-04
23	1209,169	8	-2	8,0403E-04
67	1287,967	8	-3	8,0430E-04
95	1315,616	8	-4	8,0439E-04
33	1412,199	8	-5	8,0471E-04
31	1463,911	8	-6	8,0488E-04
38	1552,633	8	-7	8,0518E-04
63	1460,751	9	0	9,0487E-04
96	1584,190	9	-1	9,0528E-04
49	1678,808	9	-2	9,0560E-04
100	1707,591	9	-3	9,0570E-04
68	1768,673	10	0	1,0059E-03
	92216,65			5,1508E-02

Las consideraciones expuestas en los cuadros anteriores son para tres niveles (cada nivel abarca un tercio de la distancia máxima del nodo más alejado al nodo de referencia en una esfera y dos sectores esféricos respectivamente, se muestran en la Figura 3.3:

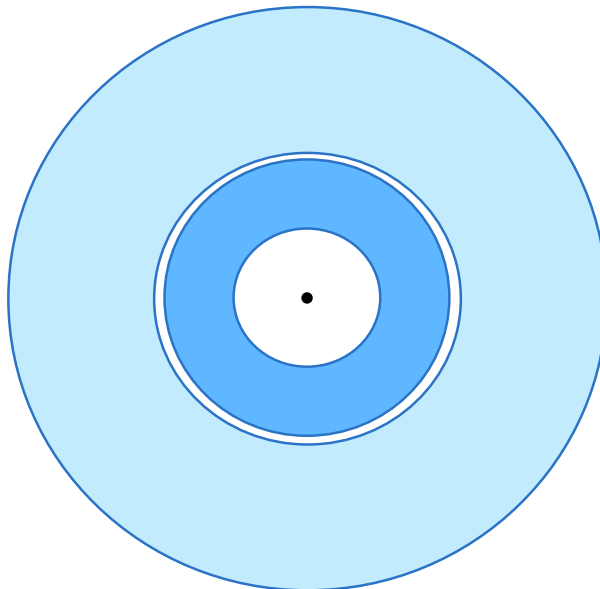


Figura 3.3 Los tres niveles: nodo de referencia (negro) 1) blanco, 2) azul y 3) celeste

Como podemos ver la Distribución Volumétrica y la Distribución Espectral, mejoran ostensiblemente el desempeño de la sincronización Original del protocolo HRTS en un rango considerable como se detalla en la Tabla 3.3:

Tabla 3.3 Comparación del desempeño del Protocolo Original con los Protocolos mejorados de Distribución Volumétrica y la Distribución Espectral

Nodis1				Original nodis16				Nodis4			
	Distancia		Suma tiempos		Distancia		Suma tiempo		Distancia		Suma tiempos
DV	82475,76		4,8275E-02		128889,330		7,5930E-02		90275,49		5,2601E-02
DE	86354,97		4,9988E-02		128889,330		7,5930E-02		92216,65		5,1508E-02

Para la Distribución Volumétrica y Espectral para el primer nodo mejora entre el 36,42% y el 34,17% respectivamente, mientras que, para el cuarto nodo, entre el 30,72% y el 32,16% respectivamente.

Las ventajas de las mejoras se reflejarían en el mejor desempeño del protocolo.

Se calcula el número de saltos que viajan los mensajes desde cada nodo, por ejemplo, del nodo m pasa por el nodo n, luego salta al nodo o y finalmente para llegar hasta su destino final (nodo de referencia, nr), además se calcula la distancia total recorrida. Se hace lo anterior, asumiendo un alcance mínimo de 100 metros, que se incrementa en un 30% paulatinamente, con el fin de encontrar el alcance mínimo para que cada uno de los nodos se enlace con el vecino más cercano a la estación base, como se grafica en la Figura 3.4:

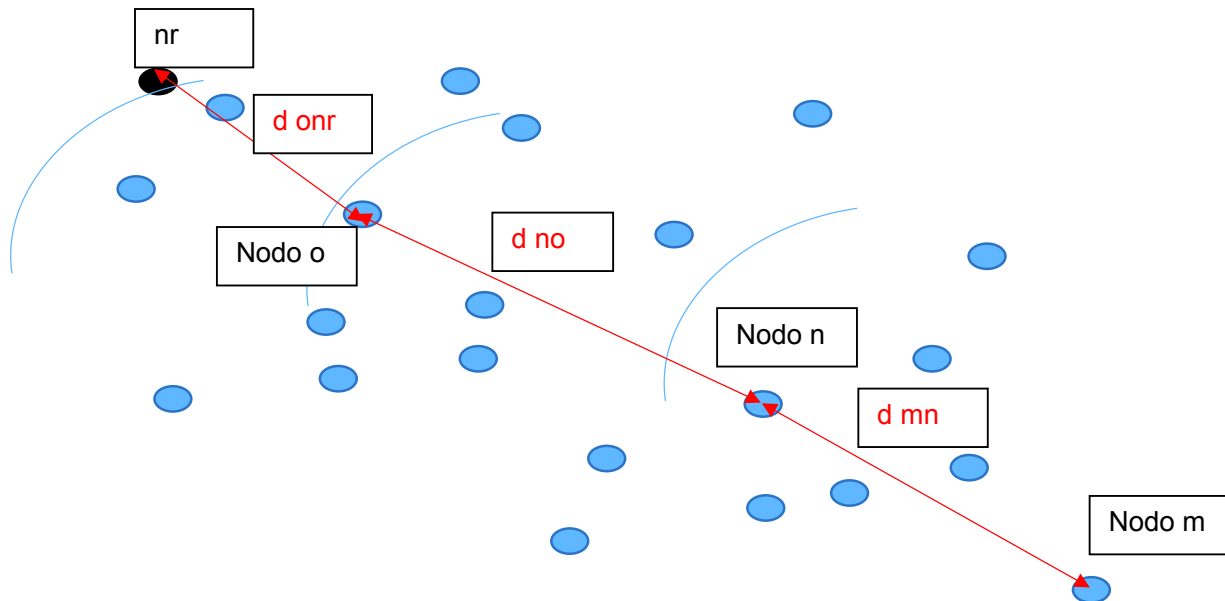


Figura 3.4 Cálculo del número de saltos, camino recorrido y distancia recorrida

La Tabla 3.4 nos indica como el programa de simulación asigna los alcances y el determina el número de saltos en la Distribución Volumétrica y la Distribución Espectral

Tabla 3.4 Distribución volumétrica (Nodo 79) y Distribución espectral (Nodo 46)

NODO 79	ALCANCE	Número de Saltos	NODO 46	ALCANCE	Número de saltos
79	0	0	46	0	0
26	130	1	35	160	1
22	160	1	89	190	1
44	190	1	25	220	1
58	190	1	32	190	2
14	100	2	91	190	2
2	130	2	72	100	3
55	250	1	66	100	3
88	130	2	2	100	2
19	100	2	17	280	1
20	130	2	7	130	3
66	130	3	48	130	2
46	160	2	37	280	2
47	190	3	84	160	2
80	130	3	58	190	4
64	220	2	73	100	3
23	160	3	64	310	2
12	280	3	4	190	3
62	250	2	5	250	3
37	190	3	78	190	4
73	160	3	51	280	5
45	220	3	87	160	4
84	220	2	20	130	5
25	190	4	41	130	6
35	100	3	79	190	3
9	100	4	83	100	5
61	130	3	53	220	4
32	100	5	85	130	3
33	190	3	92	250	3
42	190	4	70	100	4
81	100	4	8	190	3
72	190	5	39	160	5
13	250	4	74	160	4
53	160	3	14	280	3
70	100	4	94	130	3
75	280	4	27	250	5
83	190	4	45	160	6
31	100	5	24	100	4
82	220	6	82	100	6
17	250	4	21	100	4

43	130	5	97	190	5
10	190	7	26	160	6
78	250	5	99	190	4
96	100	6	15	130	6
29	190	5	22	190	4
77	100	6	36	100	4
97	280	3	43	190	6
89	160	6	11	100	6
67	250	4	18	190	4
36	160	5	19	130	6
41	190	4	28	160	5
1	130	4	47	130	6
8	250	5	71	250	7
5	130	6	50	130	5
56	250	7	90	100	5
38	130	7	80	250	4
85	160	6	60	250	4
49	280	3	1	220	6
92	160	5	59	100	7
18	160	5	76	160	5
100	100	5	69	160	6
65	220	4	55	190	7
30	220	4	88	130	5
51	100	5	12	190	6
95	130	4	44	220	5
91	190	5	61	130	7
76	130	5	29	220	4
69	190	5	86	130	6
99	220	5	56	190	5
40	250	6	65	160	5
63	190	6	9	280	4
24	190	6	34	100	7
60	100	6	40	130	6
68	190	6	6	220	5
15	100	5	93	100	8
71	160	7	3	160	7
54	190	7	33	130	9
4	100	6	54	190	7
98	130	6	23	250	5
48	100	8	62	250	8
27	250	5	42	280	5
7	160	6	75	250	5
6	250	7	16	250	6
34	130	6	13	160	8

94	100	8	52	130	6
3	130	5	81	220	6
39	190	6	98	130	7
87	100	8	57	280	8
90	100	6	31	220	6
16	280	8	96	100	9
52	250	9	77	190	9
74	250	6	67	160	5
86	220	7	30	220	8
11	160	8	49	190	7
21	130	9	100	190	9
50	160	9	95	160	6
28	220	7	63	130	10
59	160	8	68	100	7
57	190	10	10	100	7
93	130	11	38	190	7
Promedio	170,700	4,800	Promedio	172,800	4,970
Desviación Estándar	57,424	2,170	Desviación Estándar	59,102	2,052
Máximo	280	11	Máximo	310	10
Mínimo	100	1	Mínimo	100	1

La Tabla 3.5 nos indica el camino por el que recorren los mensajes e información de cada nodo al nodo de referencia o estación base

Tabla 3.5 Camino de acuerdo con la ubicación de los nodos por el que recorren los mensajes de cada uno de los nodos hacia el nodo de referencia

Ubicación del nodo	NODO 79	CAMINO																
1	79	1																
2	26	2	1															
3	22	3	1															
4	44	4	1															
5	58	5	1															
6	14	6	5	1														
7	2	7	3	1														
8	55	8	1															
9	88	9	3	1														
10	19	10	8	1														
11	20	11	4	1														
12	66	12	6	5	1													
13	46	13	8	1														
14	47	14	7	3	1													

15	80	15	11	4	1								
16	64	16	5	1									
17	23	17	9	3	1								
18	12	18	6	5	1								
19	62	19	8	1									
20	37	20	10	8	1								
21	73	21	16	5	1								
22	45	22	19	8	1								
23	84	23	5	1									
24	25	24	15	11	4	1							
25	35	25	16	5	1								
26	9	26	15	11	4	1							
27	61	27	16	5	1								
28	32	28	26	15	11	4	1						
29	33	29	13	8	1								
30	42	30	22	19	8	1							
31	81	31	18	6	5	1							
32	72	32	31	18	6	5	1						
33	13	33	14	7	3	1							
34	53	34	23	5	1								
35	70	35	29	13	8	1							
36	75	36	29	13	8	1							
37	83	37	18	6	5	1							
38	31	38	37	18	6	5	1						
39	82	39	28	26	15	11	4	1					
40	17	40	27	16	5	1							
41	43	41	36	29	13	8	1						
42	10	42	39	28	26	15	11	4	1				
43	78	43	40	27	16	5	1						
44	96	44	32	31	18	6	5	1					
45	29	45	36	29	13	8	1						
46	77	46	43	40	27	16	5	1					
47	97	47	11	4	1								
48	89	48	38	37	18	6	5	1					
49	67	49	27	16	5	1							
50	36	50	37	18	6	5	1						
51	41	51	27	16	5	1							
52	1	52	47	11	4	1							
53	8	53	33	14	7	3	1						
54	5	54	53	33	14	7	3	1					
55	56	55	46	43	40	27	16	5	1				
56	38	56	46	43	40	27	16	5	1				
57	85	57	38	37	18	6	5	1					
58	49	58	23	5	1								

59	92	59	52	47	11	4	1						
60	18	60	49	27	16	5	1						
61	100	61	52	47	11	4	1						
62	65	62	58	23	5	1							
63	30	63	47	11	4	1							
64	51	64	62	58	23	5	1						
65	95	65	58	23	5	1							
66	91	66	63	47	11	4	1						
67	76	67	63	47	11	4	1						
68	69	68	49	27	16	5	1						
69	99	69	40	27	16	5	1						
70	40	70	60	49	27	16	5	1					
71	63	71	67	63	47	11	4	1					
72	24	72	68	49	27	16	5	1					
73	60	73	68	49	27	16	5	1					
74	68	74	53	33	14	7	3	1					
75	15	75	62	58	23	5	1						
76	71	76	54	53	33	14	7	3	1				
77	54	77	70	60	49	27	16	5	1				
78	4	78	67	63	47	11	4	1					
79	98	79	61	52	47	11	4	1					
80	48	80	76	54	53	33	14	7	3	1			
81	27	81	40	27	16	5	1						
82	7	82	60	49	27	16	5	1					
83	6	83	54	53	33	14	7	3	1				
84	34	84	75	62	58	23	5	1					
85	94	85	76	54	53	33	14	7	3	1			
86	3	86	65	58	23	5	1						
87	39	87	59	52	47	11	4	1					
88	87	88	77	70	60	49	27	16	5	1			
89	90	89	81	40	27	16	5	1					
90	16	90	77	70	60	49	27	16	5	1			
91	52	91	90	77	70	6	49	27	16	5	1		
92	74	92	86	65	58	23	5	1					
93	86	93	71	67	63	47	11	4	1				
94	11	94	93	71	67	63	47	11	4	1			
95	21	95	94	93	71	67	63	47	11	4	1		
96	50	96	88	77	70	6	49	27	16	5	1		
97	28	97	92	86	65	58	23	5	1				
98	59	98	93	71	67	63	47	11	4	1			
99	57	99	95	94	93	71	67	63	47	11	4	1	
100	93	100	99	95	94	93	71	67	63	47	11	4	1

4 CONCLUSIONES

- 1.** Es importante recalcar el aporte de esta tesis de grado como un elemento didáctico para el estudio de redes WSN, ya que permite introducir datos, como la posición en tres dimensiones y características de los nodos, y devuelve el alcance, número de saltos, tiempos de demora y sincronización.
- 2.** Tomando en cuenta las densidades volumétricas y espectrales se obtuvo una reducción del 50% en el tiempo de resincronización del protocolo HRTS, lo que mejora ostensiblemente su desempeño.
- 3.** En la simulación, pueden variar diversos parámetros como: la posición y el alcance mínimo de los nodos, el tipo de distribución, el nodo de referencia, con el fin de realizar estudios de varias configuraciones de redes WSN y tener diversos casos y resultados.
- 4.** El mejoramiento del protocolo acelera el proceso de sincronización y su rendimiento.
- 5.** La valoración de este trabajo se realizó con el programa MatLab, que brinda herramientas para la elaboración y desarrollo del simulador de Redes WSN.
- 6.** El programa permite evaluar sistemas reales o virtuales de redes WSN, para determinar la ubicación de las estaciones base, el alcance mínimo de cada nodo, el número de saltos y los niveles jerárquicos del protocolo HRTS.
- 7.** Si se realiza un estudio de la nube de nodos, se podría estimar la localización de nodos de referencia para mejorar el desempeño y rendimiento de la red WSN.
- 8.** Para futuros estudios, cuando la red es muy grande, se podría para disminuir el tiempo de sincronización y optimizar el funcionamiento de la WSN mediante la división de la red en subredes, estableciendo en cada subred su respectivo nodo de referencia, además se podría mejorar las características de las estaciones base con el fin de aumentar el sincronismo y desempeño de la red.
- 9.** Podría hacerse más amigable el programa utilizando las herramientas de Simulink para que el usuario vea todo de forma más atractiva y divertida.
- 10.** Matlab resultó un programa adecuado para programar el Protocolo y sus mejoras, es un poderoso instrumento y permite ver paso a paso el desenvolvimiento de las diferentes etapas de desarrollo de la tesis de grado.

- 11.** Dividir al problema en partes ayudó a eliminar nodos que aumentan el tiempo de sincronización y encontrar las mejores opciones por tiempo, distancia, niveles y profundidad.
- 12.** El protocolo HRTS es muy valioso porque permite sincronizar redes inalámbricas de sensores de gran escala.
- 13.** Puede mejorarse cualquier protocolo de sincronización si se utilizan los algoritmos de optimización adecuados para este propósito.
- 14.** Los estudiantes de la FIEE podrían utilizar esta tesis para desarrollar trabajos futuros en tres dimensiones y con datos reales, lo que no sucede con Cooja o Castalia que solo admite valores en dos dimensiones y limita su aplicación a verificar conectividad.
- 15.** Las ventajas se reflejan en la determinación de alcance mínimo de cada nodo, el camino por el que viajarían los mensajes desde cada nodo al nodo de referencia.
- 16.** Los resultados son lógicos y permiten ver que la metodología aplicada es la ideal para mejorar el protocolo de sincronización HRTS.

5 REFERENCIAS BIBLIOGRÁFICAS [1]

- [1] J. Elson and K. Römer. Wireless Sensor Networks: A New Regime for Time Synchronization. In Proceedings of the First Workshop on Hot Topics In Networks (Hot Nets-I), Princeton, NJ, October 2002.
- [2] J. E. Elson. Time Synchronization in Wireless Sensor Networks. PhD dissertation, University of California, Los Angeles, CA, Department of Computer Science, 2003.
- [3] J. Elson, L. Girod, and D. Estrin. Short Paper: A Wireless Time-Synchronized COTS Sensor Platform, Part I: System Architecture. In Proceedings of the IEEE CAS Workshop on Wireless Communications and Networking, Pasadena, CA, September 2002.
- [4] H. Wang, L. Yip, D. Maniezzo, J. C. Chen, R. E. Hudson, J. Elson, and K. Yao. A Wireless Time-Synchronized COTS Sensor Platform: Applications to Beamforming. In Proceedings of IEEE CAS Workshop on Wireless Communications and Networking, Pasadena, CA, September 2002.
- [5] K. Yao, R. E. Hudson, C. W. Reed, D. Chen, and F. Lorenzelli. Blind Beamforming on a Randomly Distributed Sensor Array System. IEEE Journal on Selected Areas in Communications, 16(8): 1555-1567, 1998.
- [6] LAN/MAN Standards Committee of the IEEE Computer Society. IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs), October 2003.
- [7] Physikalisch Technische Bundesanstalt, Braunschweig/Berlin, Germany. Die gesetzlichen Einheiten in Deutschland, 2002.
- [8] L. Lamport. Time, clocks and the ordering of events in a distributed system. Communications of the ACM, 21(7): 558-565, 1978.
- [9] N. A. Lynch. Distributed Algorithms. Morgan Kaufmann Publishers, San Francisco, CA, 1996.
- [10] E. Anceaume and I. Puaut. A Taxonomy of Clock Synchronization Algorithms. IRISA Research Report No. PI 1103, IRISA, 1997.
- [11] G. Coulouris, J. Dollimore, and T. Kindberg. Distributed Systems – Concepts and Design. Addison-Wesley, Harlow, England, third edition, 2001.
- [12] P. Ramanathan, K. G. Shin, and R. W. Butler. Fault-Tolerant Clock Synchronization in Distributed Systems. IEEE Computer, 23(10): 33-42, 1990.
- [13] E. Kaplan, editor. Understanding GPS: Principles and Applications. Artech House, Boston, MA, 1996.
- [14] J. Elson, L. Girod, and D. Estrin. Fine-Grained Network Time Synchronization using Reference Broadcasts. In Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002), Boston, MA, December 2002.
- [15] J. v. Greunen and J. Rabaey. Lightweight Time Synchronization for Sensor Networks. In Proceedings of the 2nd ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), San Diego, CA, September 2003.
- [16] J. R. Vig. Introduction to Quartz Frequency Standards. Technical Report SLCET-TR-92-1 (Rev. 1), Army Research Laboratory, October 1992.
- [17] J. Elson and D. Estrin. Time Synchronization for Wireless Sensor Networks. In Proceedings of the 2001 International Parallel and Distributed Processing Symposium (IPDPS), Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, pages 1965-1970, San Francisco, CA, April 2001.
- [18] D. L. Mills. Network Time Protocol (Version 3) Specification, Implementation and Analysis. RFC 1305, 1992.

- [19] D. L. Mills. Improved Algorithms for Synchronizing Computer Network Clocks. *IEEE/ACM Transactions on Networking*, 3(3): 245-254, 1995.
- [20] D. L. Mills. Adaptive Hybrid Clock Discipline Algorithm for the Network Time Protocol. *IEEE/ACM Transactions on Networking*, 6(5): 505-514, 1998.
- [21] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein. Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. In *Proceedings of the 10th International Conference of Architectural Support for Programming Languages and Operating Systems*, San Jose, CA, October 2002.
- [22] S. Ganeriwal, R. Kumar, and M.B. Srivastava. Timing-Sync Protocol for Sensor Networks. In *Proceedings of the 1st ACM International Conference on Embedded Networked Sensor System (SenSys)*, pages 138-149, Los Angeles, CA, November 2003.
- [23] B. Awerbuch. Optimal Distributed Algorithms for Minimum Weight Spanning tree, Counting, Leader Election and Related Problems. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, New York, May 1987.
- [24] S.M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, Upper Saddle River, NJ, 1993.
- [25] F. Cristian. Probabilistic Clock Synchronization. *Distributed Computing*, 3: 146-158, 1989.
- [26] S. Ganeriwal, R. Kumar, S. Adlakha, and M. Srivastava. Network-Wide Time Synchronization in Sensor Networks. Technical Report NESL 01-01-2003, Networked and Embedded System Lab (NESL), University of California, Los Angeles (UCLA), 2003.
- [27] S. Mitra and J. Rabek. Power Efficient Clustering for Clock Synchronization in Dynamic Multi-Hop Sensor Networks. http://theory.lcs.mit.edu/~mitras/courses/6829/project/project_main.htm%1, 2003.
- [28] H. Dai and R. Han. TSynC: A Lightweight Bidirectional Time Synchronization Service for Wireless Sensor Networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 8(1): 125-139, 2004.
- [29] X. Chaonong, Z. Lei, X. Yongjun and L. Xiaowei, Broadcast Time Synchronization Algorithm for Wireless Sensor Network, *Institute of Computing Technology* September 2006.
- [30] A. Hu and S. D. Servetto. Asymptotically Optimal Time Synchronization in Dense Sensor Networks. In *Proceedings of the Second ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'03)*, San Diego, CA, September 2003.
- [31] E. D. Andersen and R. D: Andersen. Presolving in linear programming. *Mathematical Programming*, 71:221-245, 1996.
- [32] F. Barahona. Network design using cut inequalities. *SIAM Journal on Optimization*, 6:823-837.1996
- [33] E. Berggren. Power control and adaptive resource allocation in DS-CDMA systems PHD thesis, Royal Institute of Technology. 2003.
- [34] J. Kang, K. Park and S. Park. ATM VP based network design. *European Journal of Operational Research*: 158(3):555-569, 2004

6 ANEXOS

6.1 ANEXO I

Los programas

6.1.1 Programa original

```
clc
clear all
close all

% Simulación de Nodos

numero_nodos=100; % Ingreso de Número de
nodos
factor=1000; % factor de
multiplicación número ramdomico
alcancenodo=100; %alcance del nodo
descendiente
alcanceNR=300; %alcance del nodo de
referencia o estación base
% posicion_nodos=floor(rand(numero_nodos,3)*factor);% Calculo de la
posición
% save ('datos_fijos.mat','posicion_nodos');
load('datos_fijos.mat')
cuantos_nodis=16; %Máximo 16
nrl=5; % Nodos de
Referencia
nr=1; % Nodo Inicial
m=7;

% posición nodos eje x y z
x=posicion_nodos(:,1);
y=posicion_nodos(:,2);
z=posicion_nodos(:,3);
figure; %figura de todos los nodos
scatter3(x,y,z)
title('Todos los nodos en el espacio 3D')
view(40,35)

for nrl=1:1:cuantos_nodis
nr=1; % Nodo Inicial
m=7;

%funcion Distancia al nodo de refrencia

[d,sumd]=distancia(x,y,z,nr);

for nr=1:1:length(x)
[d1,sumd1]=distancia(x,y,z,nr);
aa(nr,1)=nr;
aa(nr,2)=sumd1;
end

% Nodo de menor distancia
nodis=sortrows(aa,2);
```

```

nr=nodis(1,1);
[d,sumd]=distancia(x,y,z,nr);

% Niveles del nodo

[dn1,dn2,dn3]=niveles(d);

% nodos de referencia

nrr=0;
for i=1:1:nr1;
    nrr(i)=nodis(i,1);
end

matriz_nivel=(zeros(numero_nodos,nr1));
matriz_distancia=(zeros(numero_nodos,nr1));

for i=1:1:length(nrr);
    nr=nrr(1,i);
    [d,sumd]=distancia(x,y,z,nr);

% Niveles del nodo

[dn1,dn2,dn3]=niveles(d);
a=size(dn1);
for j=1:1:a(1,1)
    matriz_nivel(dn1(j,1),i)=1;
    matriz_distancia(dn1(j,1),i)=dn1(j,2);
end
a=size(dn2);
for j=1:1:a(1,1);
    matriz_nivel(dn2(j,1),i)=2;
    matriz_distancia(dn2(j,1),i)=dn2(j,2);
end
a=size(dn3);
for j=1:1:a(1,1)
    matriz_nivel(dn3(j,1),i)=3;
    matriz_distancia(dn3(j,1),i)=dn3(j,2);
end
nrpo=-length(dn1)+find(dn1==0)
xdnrpo=x((dn1(nrpo))');
ydnrpo=y((dn1(nrpo))');
zdnrpo=z((dn1(nrpo))');
xdn1o=x((dn1(:,1))');
ydn1o=y((dn1(:,1))');
zdn1o=z((dn1(:,1))');
xdn2o=x((dn2(:,1))');
ydn2o=y((dn2(:,1))');
zdn2o=z((dn2(:,1))');
xdn3o=x((dn3(:,1))');
ydn3o=y((dn3(:,1))');
zdn3o=z((dn3(:,1))');
end

DatDistancia(:,1)= x;
DatDistancia(:,2)= y;
DatDistancia(:,3)= z;
DatDistancia(:,4)= d;

```



```

for i=1:1:length(DatDistancia)
    DatDistancia(i,5)= i;
end

DatosN1 = sortrows(DatDistancia,4)

dn1
dn2
dn3
DatosN1= sortrows(dn1,2)
DatosN2= sortrows(dn2,2)
DatosN3= sortrows(dn3,2)

% ingreso de los niveles de cada nodo

DatosN1(:,3)=1;
DatosN2(:,3)=2;
DatosN3(:,3)=3;

% Ingreso de la profundidad de cada nodo
j=1;
for i=1:1:length(DatosN1)
    j=j-1;
    DatosN1(i,4)=j;
    DatosN1(i,5)=tiempor(1,DatosN1(i,2));
end

j=1;
for i=1:1:length(DatosN2)
    j=j-1
    DatosN2(i,4)=j;
    DatosN2(i,5)=tiempor(1,DatosN2(i,2));
end

j=1;
for i=1:1:length(DatosN3)
    j=j-1;
    DatosN3(i,4)=j;
    DatosN3(i,5)=tiempor(1,DatosN3(i,2));
end

DATOTAL=[DatosN1; DatosN2; DatosN3];
DATOSN1N2=[DatosN1; DatosN2];
posfinN1=DatosN1(:,1);
posfinN1N2=DATOSN1N2(:,1);
posfin=DATOTAL(:,1);
format longEng
dimDTR=length(DATOTAL)+1;
DATOTAL(dimDTR,:)=sum(DATOTAL,1);

%load('datos_fijos.mat')
pos_total=posicion_nodos(posfin);
posicion=posicion_nodos;

pos_total(:,1)=posfin;
pos_total(:,2)=posicion(posfin);
pos_total(:,3)=posicion(posfin+numero_nodos);

```

```

pos_total(:,4)=posicion(posfin+(2*numero_nodos));

xtotal=posicion(posfin);
yttotal=posicion(posfin+numero_nodos);
zttotal=posicion(posfin+(2*numero_nodos));
[pfinal, dnodo,sumadnodo]=distancianodo (posicion_nodos,100,posfin);
[pfinala, dnodoa,sumadnodoa]=distancianodo (posicion_nodos,50,posfin);
[pfinalaa, dnodoaa,sumadnodoaa]=distancianodo (posicion_nodos,50,posfin);
%[numsalp,distrecorridap, caminop,tiempoip]= saltosnecam (posicion_nodos,
alcancenodo, alcanceNR, posfin)

almacenamientorig

end

```

6.2 Principal 7

```

clc
clear all
close all

% Simulacion de Nodos

numero_nodos=100; % Ingreso de Número de
nodos
factor=1000; % factor de
multiplicación número ramdomico
alcancenodo=100; %alcance del nodo
descendiente
alcanceNR=300; %alcance del nodo de
referencia o estación base
% posicion_nodos=floor(rand(numero_nodos,3)*factor);% Calculo de la
posición
% save ('datos_fijos.mat','posicion_nodos');
load('datos_fijos.mat');
cuantos_nodis=16; %MAXIMO 16
%nr1=2; % Nodos de
Referencia
% posición nodos eje x y z
x=posicion_nodos(:,1);
y=posicion_nodos(:,2);
z=posicion_nodos(:,3);

figure;
scatter3(x,y,z)
title('Todos los nodos en el espacio 3D')
view(40,35)

for nr1=1:1:cuantos_nodis
nr=1; % Nodo Inicial
m=7;

%funcion Distancia al nodo de referencia

for nr=1:1:length(x)
[d1,sumd1]=distancia(x,y,z,nr);
[dn1,dn2,dn3]=niveles(d1);

```

```

aa(nr,1)=nr;
aa(nr,2)=sumd1;
aa(nr,3)=length(dn1);
aa(nr,4)=length(dn2);
aa(nr,5)=length(dn3);

end

% Nodo optimizado
nodis=sortrows(aa,[3 2 4],{'descend' 'ascend' 'ascend'});

% nodos de referencia

nrr=0;
for i=1:1:nr1;
    nrr(i)=nodis(i,1);
end

matriz_nivel=(zeros(numero_nodos,nr1));
matriz_distancia=(zeros(numero_nodos,nr1));

for i=1:1:length(nrr);
    nr=nrr(1,i);
    [d,sumd]=distancia(x,y,z,nr);

% Niveles del nodo

    [dn1,dn2,dn3]=niveles(d);
    a=size(dn1);
    for j=1:1:a(1,1)
        matriz_nivel(dn1(j,1),i)=1;
        matriz_distancia(dn1(j,1),i)=dn1(j,2);
    end
    a=size(dn2)
    for j=1:1:a(1,1);
        matriz_nivel(dn2(j,1),i)=2;
        matriz_distancia(dn2(j,1),i)=dn2(j,2);
    end
    a=size(dn3)
    for j=1:1:a(1,1)
        matriz_nivel(dn3(j,1),i)=3;
        matriz_distancia(dn3(j,1),i)=dn3(j,2);
    end

    nrp=-length(dn1)+find(dn1==0);
    xdnrp=x((dn1(nrp))');
    ydnrp=y((dn1(nrp))');
    zdnrp=z((dn1(nrp))');
    xdn1=x((dn1(:,1))');
    ydn1=y((dn1(:,1))');
    zdn1=z((dn1(:,1))');
    xdn2=x((dn2(:,1))');
    ydn2=y((dn2(:,1))');
    zdn2=z((dn2(:,1))');
    xdn3=x((dn3(:,1))');
    ydn3=y((dn3(:,1))');
    zdn3=z((dn3(:,1))');

```

```

%end
end

DatDistancia(:,1)= x;
DatDistancia(:,2)= y;
DatDistancia(:,3)= z;
DatDistancia(:,4)= d;

for i=1:length(DatDistancia)
    DatDistancia(i,5)= i;
end

DatosN1 = sortrows(DatDistancia,4);

dn1;
dn2;
dn3;
DatosN1= sortrows(dn1,2);
DatosN2= sortrows(dn2,2);
DatosN3= sortrows(dn3,2);

% ingreso de los niveles de cada nodo

DatosN1(:,3)=1;
DatosN2(:,3)=2;
DatosN3(:,3)=3;
    DATOTAL=[DatosN1; DatosN2; DatosN3];
    DATOSN1N2=[DatosN1; DatosN2];
    posfinN1=DatosN1(:,1);
    posfinN1N2=DATOSN1N2(:,1);
    posfin=DATOTAL(:,1);
% alcancenodo=70;
% alcanceNR=300;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SALTOS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dimtotal=length(posfin);
%dimtotal=100;
ee0=DATOTAL(:,2);
ee0f=ee0<alcanceNR;
coberturaNR=sum(ee0f);
for j=1:coberturaNR
    ne(j,1)=0;
    numsal(j,1)=1;
    camino(j,1)=0;
%    tiempoi(j,1)=DATOTAL(j,5);
%    distmax(j,1)=DATOTAL(j,2);
end
k=1;

%%for ii=dimtotal:-1:(coberturaNR+1)
for nei=dimtotal:-1:1
%ne es el nodo de enlace
k=1;
alcancenodo=70
alcanceNR=300

```

```

while nei>1
    %k=1;
    % load('datos_fijos.mat');
    % load('DATOTAL_nodis1.mat');
    % load('POSFINnodis1.mat');

    posinicial=posicion_nodos;
    [dposfinalei,distninodoi,sumdistnodoi] =
    distancianodo(posinicial,nei,posfin);
    e=distninodoi;
    dime=length(e);
    % dposfine=dposfinalei;
    % sumadiste=sumdistnodoi
    %alcancenodo=70
    for jj=1.0:0.3:5.5
        alcancenodoi=alcancenodo*jj;
        f=e<alcancenodoi;
        g=e>0;
        h=e.*f.*g;
        if sum(h)==0
            disp('fuera del alcance, se busca otro alcance')
        else
            alcancenodoideal(nei,k)=alcancenodoi;
            disp('dentro del alcance')
            num2str(alcancenodoi)
            break
        end
    end
    dmax=max(h);
    nes=find(e==dmax);
    if e(1)<dmax
        nes=1;
        dmax=e(1);
        distmax(nei,k)=dmax;
        alcance(nei,k)=alcancenodoi;
        nodoenlacecam(nei,k)=nes;
    else
        if dimtotal==dime
            nodoenlacecam(nei,k)=nes;
            distmax(nei,k)=dmax;
        % else
        %     nodoenlacecam(ii,k)=length(e)-nes+1;
        %     distmax(ii,k)=dmax;
        end
        %continue
        %break
    end
    %while k~=length(posfin)
    k=k+1;
    nei=nes;
    %end
%
end
%k=k+1;
%nei=nes;
end
distrecorrida=sum(distmax,2);
camino= nodoenlacecam;
numprevio=camino>=1;
numsal=sum(numprevio,2);

```

```

tiempoconsaltos=tiemporeal(1,numsal,distrecorrida)
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% [numsalp,distrecorridap, caminop]= saltosnecam22 (posicion_nodos,
alcancenodo, alcanceNR, posfin, DATOTAL)
% % Ingreso de la profundidad de cada nodo
j=1;
for i=1:length(DatosN1)
    j=j-1;
    DatosN1(i,4)=j;
    DatosN1(i,5)=tiempor(1,DatosN1(i,2));
end

j=1;
for i=1:length(DatosN2)
    j=j-1;
    DatosN2(i,4)=j;
    DatosN2(i,5)=tiempor(1,DatosN2(i,2));
end
j=1;
for i=1:length(DatosN3)
    j=j-1;
    DatosN3(i,4)=j;
    DatosN3(i,5)=tiempor(1,DatosN3(i,2));
end
DATOTAL=[DatosN1; DatosN2; DatosN3];
DATOSN1N2=[DatosN1; DatosN2];
posfinN1=DatosN1(:,1);
posfinN1N2=DATOSN1N2(:,1);
posfin=DATOTAL(:,1);
format longEng
dimDTR=length(DATOTAL)+1;
DATOTAL(dimDTR,:)=sum(DATOTAL,1);

format longEng
dimDTR=length(DATOTAL)+1;
DATOTAL(dimDTR,:)=sum(DATOTAL,1);
for i=1:length(DatosN3)
    DatosTIEMPO(i,4)=j;
    DatosN3(i,5)=tiempor(1,DatosN3(i,2));
end

%load('datos_fijos.mat')
pos_total=posicion_nodos(posfin);
posicion=posicion_nodos;

pos_total(:,1)=posfin;
pos_total(:,2)=posicion(posfin);
pos_total(:,3)=posicion(posfin+numero_nodos);
pos_total(:,4)=posicion(posfin+(2*numero_nodos));

xtotal=posicion(posfin);
ytotal=posicion(posfin+numero_nodos);
ztotal=posicion(posfin+(2*numero_nodos));

```

```

[pfinal, dnodo,sumadnodo]=distancianodo (posicion_nodos,100,posfin);
[pfinala, dnodoa,sumadnodoa]=distancianodo (posicion_nodos,50,posfin);
[pfinalaa, dnodoaa,sumadnodoaa]=distancianodo (posicion_nodos,50,posfin);
%[numsalp,distrecorridap, caminop,tiempoip]= saltosnecam (posicion_nodos,
alcancenodo, alcanceNR, posfin)

```

6.3 Función niveles

```

function [dn1,dn2,dn3] = niveles(d)

n=0;
for i=1:length(d)
    c1=max(d)
    a=c1/3;
    b=2*a
    if d(i)<=a
        n(i)=1;
    end
    if d(i)>a && d(i)<=b
        n(i)=2;
    end
    if d(i)>b && d(i)<=c1
        n(i)=3;
    end
end
i=0;
k=0;
m=0;

for j=1:length(n)

    if n(j)==1
        i=i+1;
        n1(i)=j;
    end
    if n(j)==2
        k=k+1;
        n2(k)=j;
    end
    if n(j)==3
        m=m+1;
        n3(m)=j;
    end
end

for j=1:length(n1)

    dn1(j,1)=n1(j);
    dn1(j,2)=d(n1(j));

end
for j=1:length(n2)

    dn2(j,1)=n2(j);

```

```

        dn2(j,2)=d(n2(j));

    end
    for j=1:1:length(n3)

        dn3(j,1)=n3(j);
        dn3(j,2)=d(n3(j));

    end

end

end

```

6.4 Función Distancia

```

function [d,sumd] = distancia(x,y,z,nr)

x1=x(1,1);
y1=y(1,1);
z1=z(1,1);

aux1=x1;
aux2=y1;
aux3=z1;

x(1,1)=x(nr,1);
y(1,1)=y(nr,1);
z(1,1)=z(nr,1);

x(nr,1)=aux1;
y(nr,1)=aux2;
z(nr,1)=aux3;

c=[x(1,1),y(1,1),z(1,1)];

for i=2:1:length(x)
    dx(i)=x(i,1)-c(1,1);
    dy(i)=y(i,1)-c(1,2);
    dz(i)=z(i,1)-c(1,3);
    d(i)=sqrt(dx(i)^2+dy(i)^2+dz(i)^2);
    l=num2str(i);

end

aux=d(nr);
d(nr)=d(1);
d(1)=aux;
sumd=sum(d);

end

```

6.5 Saltos


```

%function [numsal,distrecorrida, camino,tiempoi]= saltosnecam4
(posicion_nodos, alcancenodo, alcanceNR, posfin, DATOTAL)
%esta función encuentra el número de saltos, distancia recorrida y el
camino de cada nodo i al nodo de referencia
%[dmax, nodoenlace,numsaltos]= saltomaxascendiente (e, alcance, posfin)
clear all
close all
clc
load('datos_fijos.mat');
load('DATOTAL nodisl.mat');
load('POSFINnodel.mat');
alcancenodo=100
alcanceNR=300

dimtotal=length(posfin);
% ee0=DATOTAL(:,2);
% ee0f=ee0<alcanceNR;
% coberturaNR=sum(ee0f);
% for j=1:1:coberturaNR
%     ne(j,1)=0;
%     numsal(j,1)=1;
%     camino(j,1)=0;
%     tiempoi(j,1)=DATOTAL(j,5);
%     distmax(j,1)=DATOTAL(j,2);
% end
k=1;

%dimtotal=100
%for ii=dimtotal:-1:(coberturaNR+1)
for ii=dimtotal:-1:1
    nei=ii                %ne es el nodo de enlace
    k=1;
    ik=1
    while nei>1
        posinicial=posicion_nodos;
        [dposfinalei,distninodoi,sumdistnodoi] =
distancianodo(posinicial,nei,posfin);
        e=distninodoi;
        dime=length(e);
        %     dposfine=dposfinalei;
        %     sumadiste=sumdistnodoi
        for jj=1.0:0.3:5.5
            alcancenodoi=alcancenodo*jj;
            f=e<alcancenodoi;
            g=e>0;
            h=e.*f.*g;
            if sum(h)==0
                disp('fuera del alcance, se busca otro alcance')
            else
                alcancenodoideal(nei,ik)=alcancenodoi;
                disp('dentro del alcance')
                num2str(alcancenodoi)
                ik=ik+1
                break
            end
        end
    end
    dmax=max(h);
    nes=find(e==dmax);
    if e(1)<dmax
        nes=1;
    end
end

```

```

        dmax=e(1);
        distmax(ii,k)=dmax;
        nodoenlacecam(ii,k)=nes;
    else
        if dimtotal==dime
            nodoenlacecam(ii,k)=nes;
            distmax(ii,k)=dmax;
        else
            nodoenlacecam(ii,k)=length(e)-nes+1;
            nodoenlacecam(ii,k)=nes;
            distmax(ii,k)=dmax;
        end
        %continue
        %break
    end
    %while k~=length(posfin)
    k=k+1;
    nei=nes;
    %end
end
%
end
%
end
distrecorrida=sum(distmax,2);
camino= nodoenlacecam;
numprevio=camino>=1;
numsal=sum(numprevio,2);
[tr]= tiemporeal(1,numsal, distrecorrida);
alcancemaxnodo=max(alcancenodoideal,[],2);
hrts1(:,1)=posfin
hrts1(:,2)=distrecorrida
hrts1(:,3)=numsal
hrts1f=sortrows(hrts1,[3 2]);
[thrts1f]= tiemporeal(1,hrts1f(:,3), hrts1f(:,2))
hrts1f(:,5)=thrts1f
profundidad
%end

```

6.6 Profundidad

```

din1=find(hrts1f(:,3)==1)
din2=find(hrts1f(:,3)==2)
din3=find(hrts1f(:,3)==3)
din4=find(hrts1f(:,3)==4)
din5=find(hrts1f(:,3)==5)
din6=find(hrts1f(:,3)==6)
din7=find(hrts1f(:,3)==7)
din8=find(hrts1f(:,3)==8)
din9=find(hrts1f(:,3)==9)
din10=find(hrts1f(:,3)==10)
din11=find(hrts1f(:,3)==11)
din12=find(hrts1f(:,3)==12)
din13=find(hrts1f(:,3)==13)
din14=find(hrts1f(:,3)==14)
din15=find(hrts1f(:,3)==15)
din16=find(hrts1f(:,3)==16)
din17=find(hrts1f(:,3)==17)
din18=find(hrts1f(:,3)==18)
din19=find(hrts1f(:,3)==19)

```

```

din20=find(hrts1f(:,3)==20)
din21=find(hrts1f(:,3)==21)
din22=find(hrts1f(:,3)==22)
din23=find(hrts1f(:,3)==23)
din24=find(hrts1f(:,3)==24)
din25=find(hrts1f(:,3)==25)
din26=find(hrts1f(:,3)==26)
din27=find(hrts1f(:,3)==27)
din28=find(hrts1f(:,3)==28)
din29=find(hrts1f(:,3)==29)
din30=find(hrts1f(:,3)==30)
din31=find(hrts1f(:,3)==31)
din32=find(hrts1f(:,3)==32)
din33=find(hrts1f(:,3)==33)
hrts1f(1,4)=0
j=1;
for i=2:1:(length(din1)+1)
    j=j-1;
    hrts1f(i,4)=j;

end

j=1;
for i=(length(din1)+2):1:(length(din1)+length(din2)+1)
    j=j-1;
    hrts1f(i,4)=j;

end
i2=(length(din1)+length(din2));
j=1;
for i=(i2+2):1:(i2+length(din3)+1)
    j=j-1;
    hrts1f(i,4)=j;

end
i2=i2+length(din3);
j=1;
for i=(i2+2):1:(i2+length(din4)+1)
    j=j-1;
    hrts1f(i,4)=j;

end
i2=i2+length(din4);
j=1;
for i=(i2+2):1:(i2+length(din5)+1)
    j=j-1;
    hrts1f(i,4)=j;

end
i2=i2+length(din5);
j=1;
for i=(i2+2):1:(i2+length(din6)+1)
    j=j-1;
    hrts1f(i,4)=j;

end
i2=i2+length(din6);
j=1;
for i=(i2+2):1:(i2+length(din7)+1)

```

```

        j=j-1;
        hrts1f(i,4)=j;

    end
    i2=i2+length(din7);
    j=1;
    for i=(i2+2):1:(i2+length(din8)+1)
        j=j-1
        hrts1f(i,4)=j;

    end
    i2=i2+length(din8);
    j=1;
    for i=(i2+2):1:(i2+length(din9)+1)
        j=j-1;
        hrts1f(i,4)=j;

    end
    i2=i2+length(din9);
    j=1;
    for i=(i2+2):1:(i2+length(din10)+1)
        j=j-1;
        hrts1f(i,4)=j;

    end
    i2=i2+length(din10);
    j=1;
    for i=(i2+2):1:(i2+length(din11)+1)
        j=j-1
        hrts1f(i,4)=j;

    end
    i2=i2+length(din11);
    j=1;
    for i=(i2+2):1:(i2+length(din12)+1)
        j=j-1;
        hrts1f(i,4)=j;

    end
    i2=i2+length(din12);
    j=1;
    for i=(i2+2):1:(i2+length(din13)+1)
        j=j-1;
        hrts1f(i,4)=j;

    end
    i2=i2+length(din13);
    j=1;
    for i=(i2+2):1:(i2+length(din14)+1)
        j=j-1;
        hrts1f(i,4)=j;

    end
    i2=i2+length(din14);
    j=1;
    for i=(i2+2):1:(i2+length(din15)+1)
        j=j-1
        hrts1f(i,4)=j;

```

```

end
i2=i2+length(din15);
j=1;
for i=(i2+2):1:(i2+length(din16)+1)
    j=j-1;
    hrts1f(i,4)=j;

end
i2=i2+length(din16);
j=1;
for i=(i2+2):1:(i2+length(din17)+1)
    j=j-1;
    hrts1f(i,4)=j;

end
i2=i2+length(din17);
j=1;
for i=(i2+2):1:(i2+length(din18)+1)
    j=j-1;
    hrts1f(i,4)=j;

end
i2=i2+length(din18);
j=1;
for i=(i2+2):1:(i2+length(din19)+1)
    j=j-1;
    hrts1f(i,4)=j;

end
i2=i2+length(din19);
j=1;
for i=(i2+2):1:(i2+length(din20)+1)
    j=j-1;
    hrts1f(i,4)=j;

end
i2=i2+length(din20);
j=1;
for i=(i2+2):1:(i2+length(din21)+1)
    j=j-1;
    hrts1f(i,4)=j;

end
i2=i2+length(din21);
j=1;
for i=(i2+2):1:(i2+length(din22)+1)
    j=j-1;
    hrts1f(i,4)=j;

end
i2=i2+length(din22);
j=1;
for i=(i2+2):1:(i2+length(din23)+1)
    j=j-1;
    hrts1f(i,4)=j;

end

```

```

i2=i2+length(din23);
j=1;
for i=(i2+2):1:(i2+length(din24)+1)
    j=j-1;
    hrts1f(i,4)=j;

end
i2=i2+length(din24);
j=1;
for i=(i2+2):1:(i2+length(din25)+1)
    j=j-1
    hrts1f(i,4)=j;

end
i2=i2+length(din25);
j=1;
for i=(i2+2):1:(i2+length(din26)+1)
    j=j-1;
    hrts1f(i,4)=j;

end
i2=i2+length(din26);
j=1;
for i=(i2+2):1:(i2+length(din27)+1)
    j=j-1;
    hrts1f(i,4)=j;

end
i2=i2+length(din27);
j=1;
for i=(i2+2):1:(i2+length(din28)+1)
    j=j-1
    hrts1f(i,4)=j;

end
i2=i2+length(din28);
j=1;
for i=(i2+2):1:(i2+length(din29)+1)
    j=j-1;
    hrts1f(i,4)=j;

end
i2=i2+length(din29);
j=1;
for i=(i2+2):1:(i2+length(din30)+1)
    j=j-1;
    hrts1f(i,4)=j;

end
i2=i2+length(din30);
j=1;
for i=(i2+2):1:(i2+length(din31)+1)
    j=j-1
    hrts1f(i,4)=j;

end
i2=i2+length(din31);
j=1;

```

```

for i=(i2+2):1:(i2+length(din32)+1)
    j=j-1;
    hrtslf(i,4)=j;

end
i2=i2+length(din32);
j=1;
for i=(i2+2):1:(i2+length(din33)+1)
    j=j-1;
    hrtslf(i,4)=j;

end

```

6.7 DISTANCIA NODO

```

function [dposfinale,distninode,sumdistnodo] =
distancianodo(posinicial,nodoi,posfin)
posicione_nodos=posinicial;
dposfinum=posfin;
numero_nodos=length(posicione_nodos);

dxx=posicione_nodos(dposfinum);
dyy=posicione_nodos(dposfinum+numero_nodos);
dzz=posicione_nodos(dposfinum+(2*numero_nodos));

dposfinale(:,1)= dxx;
dposfinale(:,2)= dyy;
dposfinale(:,3)= dzz;
dposfinale(:,4)= posfin;

c=[dxx(nodoi,1), dyy(nodoi,1), dzz(nodoi,1)];

for i=nodoi:-1:1
    dnodox(i)=dxx(i,1)-c(1,1);
    dnodoy(i)=dyy(i,1)-c(1,2);
    dnodoz(i)=dzz(i,1)-c(1,3);
    distninode(i,1)=sqrt(dnodox(i)^2+dnodoy(i)^2+dnodoz(i)^2);
    %l=num2str(i);
end

sumdistnodo=sum(distninode);

end

```

6.8 ALCANCE NODO MÍNIMO

```
function [alcancenodo] = alcancenodomin(posicion_nodos)

n=0;
alcancenodo=70;
posinicial=posicion_nodos;
[dposfinalei,distninodoi,sumdistnodoi] =
distancianodo(posinicial,nei,posfin);
e=distninodoi;
dime=length(e);

for jj=1.0:0.3:4.0
    alcancenodoi=alcancenodo*jj
    f=e<alcancenodoi;
    g=e>0;
    h=e.*f.*g;
    if sum(h)==0
        disp('fuera del alcance, se busca otro alcance')
    else
        %       alcancenodoideal(ii,k)=alcancenodoi;
        aaepito=alcancenodoi;
        disp('dentro del alcance')
        num2str(alcancenodoi)

        break
    end

end

end
```

6.9 ALMACENAMIENTO ORIGINAL

```
if nr1==1
    save('DatosN1 nodis1o','DatosN1');
    save('DatosN2 nodis1o','DatosN2');
    save('DatosN3 nodis1o','DatosN3');
    save('DATOTAL nodis1o','DATOTAL');
    save('POSN1nodis1o','posfinN1');
    save('POSFINnodis1o','posfin');
    figure('Name','Nodo de referencia 1o');
    graficoso
end
if nr1==2
    save('DatosN1 nodis2o','DatosN1');
    save('DatosN2 nodis2o','DatosN2');
    save('DatosN3 nodis2o','DatosN3');
    save('DATOTAL nodis2o','DATOTAL');
    save('POSN1nodis2o','posfinN1');
    save('POSFINnodis2o','posfin');
    figure('Name','Nodo de referencia 2o');
    graficoso
end
if nr1==3
    save('DatosN1 nodis3o','DatosN1');
    save('DatosN2 nodis3o','DatosN2');
```



```

        save('DatosN3 nodis3o','DatosN3');
        save('DATOTAL nodis3o','DATOTAL');
        save('POSN1nodis3o','posfinN1');
        save('POSFINnodis3o','posfin');
        figure('Name','Nodo de referencia 3o');
        graficoso
    end
    if nrl==4
        save('DatosN1 nodis4o','DatosN1');
        save('DatosN2 nodis4o','DatosN2');
        save('DatosN3 nodis4o','DatosN3');
        save('DATOTAL nodis4o','DATOTAL');
        save('POSN1nodis4o','posfinN1');
        save('POSFINnodis4o','posfin');
        figure('Name','Nodo de referencia 4o');
        graficoso
    end
    if nrl==5
        save('DatosN1 nodis5o','DatosN1');
        save('DatosN2 nodis5o','DatosN2');
        save('DatosN3 nodis5o','DatosN3');
        save('DATOTAL nodis5o','DATOTAL');
        save('POSN1nodis5o','posfinN1');
        save('POSFINnodis5o','posfin');
        figure('Name','Nodo de referencia 5o');
        graficoso
    end
    if nrl==6
        save('DatosN1 nodis6o','DatosN1');
        save('DatosN2 nodis6o','DatosN2');
        save('DatosN3 nodis6o','DatosN3');
        save('DATOTAL nodis6o','DATOTAL');
        save('POSN1nodis6o','posfinN1');
        save('POSFINnodis6o','posfin');
        figure('Name','Nodo de referencia 6o');
        graficoso
    end
    if nrl==7
        save('DatosN1 nodis7o','DatosN1');
        save('DatosN2 nodis7o','DatosN2');
        save('DatosN3 nodis7o','DatosN3');
        save('DATOTAL nodis7o','DATOTAL');
        save('POSN1nodis7o','posfinN1');
        save('POSFINnodis7o','posfin');
        figure('Name','Nodo de referencia 7o');
        graficoso
    end
    if nrl==8
        save('DatosN1 nodis8o','DatosN1');
        save('DatosN2 nodis8o','DatosN2');
        save('DatosN3 nodis8o','DatosN3');
        save('DATOTAL nodis8o','DATOTAL');
        save('POSN1nodis8o','posfinN1');
        save('POSFINnodis8o','posfin');
        figure('Name','Nodo de referencia 8o');
        graficoso
    end
    if nrl==9
        save('DatosN1 nodis9o','DatosN1');
        save('DatosN2 nodis9o','DatosN2');

```

```

        save('DatosN3 nodis9o','DatosN3');
        save('DATOTAL nodis9o','DATOTAL');
        save('POSN1nodis9o','posfinN1');
        save('POSFINnodis9o','posfin');
        figure('Name','Nodo de referencia 9o');
        graficoso
    end
    if nrl==10
        save('DatosN1 nodis10o','DatosN1');
        save('DatosN2 nodis10o','DatosN2');
        save('DatosN3 nodis10o','DatosN3');
        save('DATOTAL nodis10o','DATOTAL');
        save('POSN1nodis10o','posfinN1');
        save('POSFINnodis10o','posfin');
        figure('Name','Nodo de referencia 10o');
        graficoso
    end
    if nrl==11
        save('DatosN1 nodis11o','DatosN1');
        save('DatosN2 nodis11o','DatosN2');
        save('DatosN3 nodis11o','DatosN3');
        save('DATOTAL nodis11o','DATOTAL');
        save('POSN1nodis11o','posfinN1');
        save('POSFINnodis11o','posfin');
        figure('Name','Nodo de referencia 11o');
        graficoso
    end
    if nrl==12
        save('DatosN1 nodis12o','DatosN1');
        save('DatosN2 nodis12o','DatosN2');
        save('DatosN3 nodis12o','DatosN3');
        save('DATOTAL nodis12o','DATOTAL');
        save('POSN1nodis12o','posfinN1');
        save('POSFINnodis12o','posfin');
        figure('Name','Nodo de referencia 12o');
        graficoso
    end
    if nrl==13
        save('DatosN1 nodis13o','DatosN1');
        save('DatosN2 nodis13o','DatosN2');
        save('DatosN3 nodis13o','DatosN3');
        save('DATOTAL nodis13o','DATOTAL');
        save('POSN1nodis13o','posfinN1');
        save('POSFINnodis13o','posfin');
        figure('Name','Nodo de referencia 13o');
        graficoso
    end
    if nrl==14
        save('DatosN1 nodis14o','DatosN1');
        save('DatosN2 nodis14o','DatosN2');
        save('DatosN3 nodis14o','DatosN3');
        save('DATOTAL nodis14o','DATOTAL');
        save('POSN1nodis14o','posfinN1');
        save('POSFINnodis14o','posfin');
        figure('Name','Nodo de referencia 14o');
        graficoso
    end
    if nrl==15
        save('DatosN1 nodis15o','DatosN1');
        save('DatosN2 nodis15o','DatosN2');

```

```

        save('DatosN3 nodis15o','DatosN3');
        save('DATOTAL nodis15o','DATOTAL');
        save('POSN1nodis15o','posfinN1');
        save('POSFINnodis15o','posfin');
        figure('Name','Nodo de referencia 15o');
        graficoso
    end
    if nrl==16
        save('DatosN1 nodis16o','DatosN1');
        save('DatosN2 nodis16o','DatosN2');
        save('DatosN3 nodis16o','DatosN3');
        save('DATOTAL nodis16o','DATOTAL');
        save('POSN1nodis16o','posfinN1');
        save('POSFINnodis16o','posfin');
        figure('Name','Nodo de referencia 16o');
        graficoso
    end
end

```

6.10 ALMACENAMIENTO

```

if nrl==1
    save('DatosN1 nodis1','DatosN1');
    save('DatosN2 nodis1','DatosN2');
    save('DatosN3 nodis1','DatosN3');
    save('DATOTAL nodis1','DATOTAL');
    save('POSN1nodis1','posfinN1');
    save('POSFINnodis1','posfin');
    figure('Name','Nodo de referencia 1');
    graficos
end
if nrl==2
    save('DatosN1 nodis2','DatosN1');
    save('DatosN2 nodis2','DatosN2');
    save('DatosN3 nodis2','DatosN3');
    save('DATOTAL nodis2','DATOTAL');
    save('POSN1nodis2','posfinN1');
    save('POSFINnodis2','posfin');
    figure('Name','Nodo de referencia 2');
    graficos
end
if nrl==3
    save('DatosN1 nodis3','DatosN1');
    save('DatosN2 nodis3','DatosN2');
    save('DatosN3 nodis3','DatosN3');
    save('DATOTAL nodis3','DATOTAL');
    save('POSN1nodis3','posfinN1');
    save('POSFINnodis3','posfin');
    figure('Name','Nodo de referencia 3');
    graficos
end
if nrl==4
    save('DatosN1 nodis4','DatosN1');
    save('DatosN2 nodis4','DatosN2');
    save('DatosN3 nodis4','DatosN3');
    save('DATOTAL nodis4','DATOTAL');
    save('POSN1nodis4','posfinN1');
    save('POSFINnodis4','posfin');
    figure('Name','Nodo de referencia 4');
    graficos
end

```

```

end
if nrl==5
    save('DatosN1 nodis5','DatosN1');
    save('DatosN2 nodis5','DatosN2');
    save('DatosN3 nodis5','DatosN3');
    save('DATOTAL nodis5','DATOTAL');
    save('POSN1nodis5','posfinN1');
    save('POSFINnodis5','posfin');
    figure('Name','Nodo de referencia 5');
    graficos
end
if nrl==6
    save('DatosN1 nodis6','DatosN1');
    save('DatosN2 nodis6','DatosN2');
    save('DatosN3 nodis6','DatosN3');
    save('DATOTAL nodis6','DATOTAL');
    save('POSN1nodis6','posfinN1');
    save('POSFINnodis6','posfin');
    figure('Name','Nodo de referencia 6');
    graficos
end
if nrl==7
    save('DatosN1 nodis7','DatosN1');
    save('DatosN2 nodis7','DatosN2');
    save('DatosN3 nodis7','DatosN3');
    save('DATOTAL nodis7','DATOTAL');
    save('POSN1nodis7','posfinN1');
    save('POSFINnodis7','posfin');
    figure('Name','Nodo de referencia 7');
    graficos
end
if nrl==8
    save('DatosN1 nodis8','DatosN1');
    save('DatosN2 nodis8','DatosN2');
    save('DatosN3 nodis8','DatosN3');
    save('DATOTAL nodis8','DATOTAL');
    save('POSN1nodis8','posfinN1');
    save('POSFINnodis8','posfin');
    figure('Name','Nodo de referencia 8');
    graficos
end
if nrl==9
    save('DatosN1 nodis9','DatosN1');
    save('DatosN2 nodis9','DatosN2');
    save('DatosN3 nodis9','DatosN3');
    save('DATOTAL nodis9','DATOTAL');
    save('POSN1nodis9','posfinN1');
    save('POSFINnodis9','posfin');
    figure('Name','Nodo de referencia 9');
    graficos
end
if nrl==10
    save('DatosN1 nodis10','DatosN1');
    save('DatosN2 nodis10','DatosN2');
    save('DatosN3 nodis10','DatosN3');
    save('DATOTAL nodis10','DATOTAL');
    save('POSN1nodis10','posfinN1');
    save('POSFINnodis10','posfin');
    figure('Name','Nodo de referencia 10');
    graficos
end

```

```

end
if nr1==11
    save('DatosN1 nodis11','DatosN1');
    save('DatosN2 nodis11','DatosN2');
    save('DatosN3 nodis11','DatosN3');
    save('DATOTAL nodis11','DATOTAL');
    save('POSN1nodis11','posfinN1');
    save('POSFINnodis11','posfin');
    figure('Name','Nodo de referencia 11');
    graficos
end
if nr1==12
    save('DatosN1 nodis12','DatosN1');
    save('DatosN2 nodis12','DatosN2');
    save('DatosN3 nodis12','DatosN3');
    save('DATOTAL nodis12','DATOTAL');
    save('POSN1nodis12','posfinN1');
    save('POSFINnodis12','posfin');
    figure('Name','Nodo de referencia 12');
    graficos
end
if nr1==13
    save('DatosN1 nodis13','DatosN1');
    save('DatosN2 nodis13','DatosN2');
    save('DatosN3 nodis13','DatosN3');
    save('DATOTAL nodis13','DATOTAL');
    save('POSN1nodis13','posfinN1');
    save('POSFINnodis13','posfin');
    figure('Name','Nodo de referencia 13');
    graficos
end
if nr1==14
    save('DatosN1 nodis14','DatosN1');
    save('DatosN2 nodis14','DatosN2');
    save('DatosN3 nodis14','DatosN3');
    save('DATOTAL nodis14','DATOTAL');
    save('POSN1nodis14','posfinN1');
    save('POSFINnodis14','posfin');
    figure('Name','Nodo de referencia 14');
    graficos
end
if nr1==15
    save('DatosN1 nodis15','DatosN1');
    save('DatosN2 nodis15','DatosN2');
    save('DatosN3 nodis15','DatosN3');
    save('DATOTAL nodis15','DATOTAL');
    save('POSN1nodis15','posfinN1');
    save('POSFINnodis15','posfin');
    figure('Name','Nodo de referencia 15');
    graficos
end
if nr1==16
    save('DatosN1 nodis16','DatosN1');
    save('DatosN2 nodis16','DatosN2');
    save('DatosN3 nodis16','DatosN3');
    save('DATOTAL nodis16','DATOTAL');
    save('POSN1nodis16','posfinN1');
    save('POSFINnodis16','posfin');
    figure('Name','Nodo de referencia 16');
    graficos

```

```
end
```

6.11 TIEMPO REAL

```
function [tr]= tiemporeal(n,numsal, distancia)
%     n=1
%     distancia=1000
c=299792458;
v1=c/1.0;
v2=c/1.0;
d21=0.1e-3;
d22=0.2e-3;
d11=distancia/v1;
d12=distancia/v2;
if n==1 ;
    tr=d11+(d21*numsal);
else
    tr=d12+(d22*numsal);
end
end
```

7 ORDEN DE EMPASTADO