

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

**EVALUACIÓN DE LOS RETARDOS EN REDES LORAWAN
MULTISALTO CON TOPOLOGÍA LINEAL**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
TELECOMUNICACIONES**

CARLOS SEBASTIÁN GUERRERO POZO

carlos.guerrero01@epn.edu.ec

DIRECTOR: CARLOS ROBERTO EGAS ACOSTA

carlos.egas@epn.edu.ec

DMQ, febrero 2022

CERTIFICACIONES

Yo, CARLOS SEBASTIÁN GUERRERO POZO declaro que el trabajo de integración curricular 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.

CARLOS SEBASTIÁN GUERRERO POZO

Certifico que el presente trabajo de integración curricular fue desarrollado por CARLOS SEBASTIÁN GUERRERO POZO, bajo mi supervisión.

CARLOS ROBERTO EGAS ACOSTA
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

CARLOS SEBASTIÁN GUERRERO POZO

CARLOS ROBERTO EGAS ACOSTA

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
RESUMEN	V
ABSTRACT	VI
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 OBJETIVO GENERAL	2
1.2 OBJETIVOS ESPECÍFICOS	2
1.3 ALCANCE	2
1.4 MARCO TEÓRICO.....	3
1.4.1 RETARDOS EN NODOS.....	3
1.4.2 INTERNET DE LAS COSAS (IOT)	5
1.4.3 REDES LPWAN	5
1.4.4 LoRa.....	6
1.4.5 LoRaWAN.....	10
1.4.6 TOPOLOGÍAS LINEALES EN IoT	11
2 METODOLOGÍA	15
2.1 INTRODUCCIÓN	15
2.2 ELEMENTOS A UTILIZAR EN LA IMPLEMENTACION DEL PROTOTIPO 15	
2.2.1 TRANSCÉPTORES LoRa SX1276.....	15
2.2.2 NodeMCU ESP8266	17
2.2.3 MÓDULO MT3608	18
2.2.4 MÓDULO TP4056	19
2.2.5 BATERÍA 18650.....	20
2.3 EVALUACION DEL RETARDO.....	21
2.3.1 DISTRIBUCIÓN DE LOS EQUIPOS	21
2.3.2 INTERVALO DE MEDIDA.....	22
2.3.3 IMPLEMENTACION DE LOS NODOS.....	22
2.3.4 PROGRAMACIÓN DE LOS NODOS CON IDE ARDUINO Y LIBRERÍA RADIOHEAD.....	26
2.3.5 CARACTERÍSTICAS DEL ENVÍO DE PAQUETES	29
2.3.6 CARACTERÍSTICAS DE LA RECEPCIÓN DE PAQUETES.....	30
2.3.7 EVALUACIÓN MATEMÁTICA DEL RETARDO	31
2.3.8 COMPARATIVA DE DATOS ENVIADOS Y RECIBIDOS.....	33
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	36

3.1	ESCENARIO DE PRUEBAS	36
3.1.1	ENTORNO URBANO – Av. GRAL. IGNACIO DE VEINTIMILLA	36
3.1.2	ENTORNO RURAL – PARQUE BICENTENARIO	37
3.2	RESULTADOS	40
3.2.1	MEDICIONES – AMBIENTE RURAL 1	40
3.2.2	MEDICIONES – AMBIENTE RURAL 2	42
3.2.3	MEDICIONES – AMBIENTE URBANO	45
3.2.4	EXPRESIÓN MATEMÁTICA DEL RETARDO MULTISALTO.....	46
3.3	CONCLUSIONES.....	49
3.4	RECOMENDACIONES	50
	ANEXOS	53
	Diagrama de Pines del Transceptor SX1276 modelo HPD13A V1.1	54
	Esquematación y diseños en PCB.....	56
	Montaje final del nodo	57
	Código Nodo Origen.....	58
	Código Nodo Repetidor / Destino.....	61

RESUMEN

El objetivo principal de este trabajo es evaluar los retardos producidos en redes multisalto con una topología lineal cuando se emplea tecnología LoRa; además, de resaltar limitaciones que podrían afectar a las implementaciones que utilizan dicha red. La evaluación de los retardos se apoyará en experimentos efectuados en un prototipo de red con topología lineal, adicionalmente se realizó mediciones del RSSI y la SNR. El prototipo se lo construyó y programó con el módulo NodeMCU ESP8266 y el transceptor SX1276, empleando Arduino IDE y la librería de RadioHead para diversidad de radio transmisores.

Finalmente se evaluó la expresión general para el retardo extremo a extremo con tamaño fijo de paquetes, para ver si se ajusta al modelo empleado.

PALABRAS CLAVE: Retardo, SX1276, LoRa, RSSI, SNR, RadioHead.

ABSTRACT

The main objective of this work is to evaluate the delays produced in multi-hop networks with a linear topology when LoRa technology is used; in addition, to highlight limitations that could affect the implementations that use this network. The evaluation of the delays will be based on experiments carried out on a prototype network with linear topology, in addition to measurements of RSSI and SNR. The prototype was built and programmed with the ESP8266 NodeMCU module and the SX1276 transceiver, using Arduino IDE and the RadioHead library for radio transmitter diversity.

Finally, the general expression for the end-to-end delay with fixed packet size was evaluated to see if it fits the model used.

KEYWORDS: Delay, SX1276, LoRa, RSSI, SNR, RadioHead.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

La ubicuidad de Internet, los nuevos e innovadores protocolos de comunicación y la miniaturización de dispositivos computacionales dieron lugar a un nuevo paradigma llamado Internet de las Cosas (IoT).

Se proyecta que la cantidad de dispositivos IoT crecerá constantemente en los próximos años, especialmente gracias a sus posibles aplicaciones en una amplia y diversa gama de campos. Las soluciones de IoT se están utilizando en la industria, en la agricultura, en ciudades inteligentes y en muchos otros sectores. En muchos casos, los dispositivos IoT funcionan con baterías y están sujetos a restricciones de potencia muy estrictas. Por esta razón, se ha desarrollado y estandarizado una nueva gama de protocolos de comunicación inalámbrica de baja potencia para admitir la operación de redes de área amplia de baja potencia (LPWAN). Estas redes generalmente están formadas por dispositivos simples y económicos que necesitan comunicarse con poca frecuencia a largas distancias con bajas velocidades de bits.

La red de área amplia (WAN) de largo alcance (LoRa) es una de las tecnologías más prometedoras de las LPWAN. LoRaWAN aprovecha el espectro extendido (Spread Spectrum, SS), la ortogonalidad del chirp y las buenas características de propagación del espectro sub-GHz para proporcionar una comunicación confiable a largas distancias. Esto viene a expensas de la velocidad de bits, las tasas de información que se pueden enviar y del intervalo de tiempo máximo entre transmisiones consecutivas debido a limitaciones del ciclo de trabajo en las bandas utilizadas por el protocolo.

Actualmente la investigación sobre LoRa es muy extensa y se centra principalmente en cómo ciertos factores como la temperatura, la humedad, ruido y las precipitaciones afectan al alcance máximo y a la duración de la batería, mientras que la investigación sobre cómo funciona realmente una red LoRa y cuáles son sus limitaciones se limita principalmente a simulaciones y teorías. En este trabajo espera proporcionar más información sobre cómo afectan dichas limitaciones para un uso en concreto, mostrando la calidad de la señal, la relación señal-ruido y cuál es el tiempo transcurrido en el envío y recepción de información.

Con hecho en el presente trabajo se evalúa el retardo que puede producirse al utilizar la tecnología LoRa en redes multisalto empleando una topología lineal en varias zonas ambientales (entornos urbanos y rurales), presentación de una ecuación que represente dicho retardo, también el diseño/construcción de nodos finales con el módulo SX1276 de Semtech y la plataforma NodeMCU ESP8266 como medio para cumplir los objetivos posteriormente planteados.

1.1 OBJETIVO GENERAL

Evaluar los retardos en redes LoRaWAN multisalto con topología lineal.

1.2 OBJETIVOS ESPECÍFICOS

- Llevar a cabo un estudio del estado del arte para la tecnología LoRa.
- Diseñar e implementar un prototipo de red con tecnología LoRa utilizando la plataforma NodeMCU - ESP8266 en convergencia con el módulo SX1276 para el monitoreo de la comunicación.
- Evaluar los retardos en el prototipo de red y parámetros adicionales como son el RSSI y la SNR.
- Presentar una expresión matemática para evaluar el retardo para topologías lineales multisalto usando tecnología LoRa.

1.3 ALCANCE

Describir el alcance del componente en referencia a lo establecido en el Plan.

La contribución del presente trabajo se centra en la evaluación del retardo en redes LoRaWAN con topología lineal multisalto, utilizando su capa física. Para esto, se realizarán mediciones del Indicador de Nivel de Señal Recibida (RSSI, por las siglas del inglés Received Signal Strength Indicator), de la Relación Señal a Ruido (SNR, Signal to Noise), los tiempos de procesamiento, transmisión y propagación para la conexión extremo a extremo del enlace, que conforman el retardo del sistema.

El módulo se empleará en la capa física de la tecnología LoRa, para los nodos, es el SX1276 que es un chip de la compañía Semtech, este trabaja en la banda de 915 MHz (perteneciente a las ISM), permitiendo la modificación de diferentes variables como el Ancho de Banda (BW), el Factor de Ensanchamiento (SF), la Tasa de Codificación (CR), la Longitud del Preámbulo (PL) y la Potencia de Transmisión, esto conforme a lo establecido en la normativa de LoRa Alliance [1]. Mediante dichas modificaciones se pueden conseguir distintos alcances, consiguiendo así evaluar el desempeño de la transmisión, así como los retrasos.

Se efectuarán pruebas en campo y las mediciones correspondientes en dos escenarios principales de la provincia de Pichincha, un ambiente rural y otro urbano que serán elegidos en el análisis de la metodología. Para dichas pruebas, varios nodos se desplegarán en una

topología lineal, transmitiendo uno hacia otro, para finalmente culminar en un nodo central manteniendo condiciones (en su mayoría) de línea de vista.

1.4 MARCO TEÓRICO

Las soluciones elaboradas para implementaciones de IoT, actualmente son capaces de obtener una performance satisfactoria en ambientes adversos para comunicaciones de radio frecuencia. Y esto se da gracias al continuo avance e investigación que se dan en redes de sensores inalámbricos, aportando así con mejoras en confiabilidad, robustez y escalabilidad.

LoRaWAN y LoRa, como protocolo de red y tecnología de capa física respectivamente, sobresalen como una de las principales elecciones para aplicaciones de redes de sensores inalámbricos, diferenciándose de otras por su vasto rango de transmisión, ínfimo costo y frecuencias de trabajo.

1.4.1 RETARDOS EN NODOS

En términos de redes, un paquete puede definirse como una unidad de datos que se traslada desde un origen hasta un destino. Los datos transmitidos por redes informáticas como Internet se seccionan en paquetes. El dispositivo que los recibe reacopla estos paquetes.

El retardo o latencia puede definirse como el tiempo que tarda el mensaje/paquete completo en llegar a su destino, después de que el bit inicial se envíe desde el origen. Es una métrica importante para cuantificar el rendimiento de la red. El retardo de un paquete se calcula sumando los siguientes cuatro componentes: retardo de propagación, retardo de transmisión, retardo de cola y retardo de procesamiento [2] [3].

- **Retardo de Propagación:** es el tiempo que tarda un bit en recorrer desde un extremo a otro el enlace, los bits se desplazan en forma de señales electromagnéticas. La velocidad de propagación de dichas señales viene dada por el medio que atraviesen. La fórmula del retardo de propagación se presenta en Ecuación 1.1.

$$\text{Retardo de Propagación} = \frac{D}{S}$$

Ecuación 1.1 Retardo de Propagación

Donde

- D es la distancia entre la fuente y la recepción en un enlace
- S es la velocidad de transmisión.

- **Retardo de transmisión:** es el tiempo empleado para propulsar todos los bits del paquete en el enlace de transmisión. Este depende principalmente del tamaño de los datos y del ancho de banda del canal (en bps). En Ecuación 1.2 se expresa la fórmula del retardo de transmisión:

$$\text{Retardo de Transmisión} = \frac{L}{R}$$

Ecuación 1.2 Retardo de Transmisión

Donde

- L es la longitud del paquete
- R es la velocidad de transmisión.
- **Retardo de la cola:** es el tiempo que el paquete permanece en una cola a la espera de ser transmitido al enlace. En general, no se puede calcular el retardo de cola porque no se tiene ninguna fórmula para ello.

Este retraso depende de los siguientes factores:

- Si el tamaño de la cola es grande, el retardo de la cola será desmesurado. Si la cola está vacía, el retardo será menor o nulo.
- Si llegan más paquetes en un intervalo de tiempo corto o nulo, el retardo de la cola será grande.
- Cuanto menor sea el número de servidores/enlaces, mayor será el retardo de la cola.
- **Retardo de procesamiento:** es el tiempo que necesitan los dispositivos intermedios de la red/topología para decidir a dónde se reenvía el paquete y por cual interface se lo retransmite. Su valor depende de la velocidad del procesador del dispositivo y el protocolo que se está utilizando.

Este retraso depende de los siguientes factores:

- La velocidad del procesador de un equipo varía según el hardware del equipo. Así como factores externos como la temperatura o polvo.
- Las arquitecturas informáticas y los protocolos pueden optimizarse para minimizar estos retrasos.

El componente principalmente del retardo en un nodo se da por el retardo de la cola y por el de procesamiento.

1.4.2 INTERNET DE LAS COSAS (IOT)

El Internet de las cosas (IoT) describe la red de objetos físicos (cosas) la cual está integrada con sensores, software y otras tecnologías para efectuar el intercambio de datos con otros dispositivos y sistemas mediante Internet [4].

Gran parte de las implementaciones en IoT, incorporan protocolos cableados de modo que hay limitaciones para ofrecer movilidad (6TiSCH, HART), además los costes de instalación son elevados, o bien de ser inalámbricos (IEEE 802.15.4), la operación se da en bandas que de sobremana se hayan saturadas (2.4 GHz z Industrial, Scientific and Medical (ISM)) [5].

1.4.3 REDES LPWAN

Las redes de área amplia de baja potencia (Low-Power Wide Area Network, en inglés) no solo facilitan cubrir una extensa área, además operan a un costo diminuto y una mayoría en el consumo energético en contraste con las redes clásicas.

Dicha tecnología aparece con el propósito de solventar dificultades presentes en las tecnologías inalámbricas tradicionales, como Bluetooth, Zigbee, Wi-Fi, etc., no pueden ser empleadas en situaciones donde se demanden transferencias de largo alcance. Si bien, redes celulares como 2G, 3G y 4G toleran áreas extensas, el consumo energético desmesurado es su inconveniente esencial [6].

1.4.3.1 Principales características de una red LPWAN

Las siguientes características transforman a LPWAN en una de las preferidas de muchos para solventar las necesidades presentes en las aplicaciones IoT [6].

- **Baja potencia:** los transmisores-receptores de LPWAN pueden obrar con baterías que permiten (y con diferencia) tener una “longevidad” de incluso 10 años, con lo cual disminuye en gran medida los costes vinculados al mantenimiento.
- **Bajo costo:** el empleo de los protocolos LPWAN aminora la complejidad al momento de diseñar el hardware, lo que, a su vez disminuye el coste de manufactura. Desde otra perspectiva, utilizar bandas licenciadas y sin licencia, permite acortar el coste de red.

- **Largo alcance:** la distancia eficaz teórica para estas redes varía desde 1 a 10 km en entornos urbanos incluso 15 km en zonas rústicas. En lugares subterráneos y en ambientes indoor, permite una comunicación de forma efectiva.
- **Baja tasa de datos:** aceptan transmisiones pequeñas de datos 10 a 10 KB de información a una tasa de hasta 1 Mbps, perfecta en el uso de sensores remotos u otras aplicaciones con peticiones ínfimas de datos.

1.4.3.2 Tecnologías LPWAN

Las tecnologías LPWAN, son usadas para la transmisión de volúmenes reducidos de datos a grandes distancias, con la particularidad que el consumo de energía es bajo en contraste con otras tecnologías aplicadas en la actualidad, como es el caso de tecnologías telefónicas con la cual se da la posibilidad de cubrir extensas superficies [7].

Estas tecnologías no pueden ser empleadas para el uso de voz, video o servicios multimedia que contengan grandes volúmenes de datos; pero su utilidad se da en permitir conectar una gran cantidad de dispositivos con un ancho de banda menor. En Tabla 1.1 se encuentra una comparativa de las algunas tecnologías LPWAN.

Tabla 1.1 Comparación técnica entre diferentes tecnologías LPWAN [8]

Tecnología	Sigfox	NB-IoT	Ingenu	D7AP	HARE	LoRaWAN
Topología	Estrella	Estrella	Árbol Estrella	Árbol Estrella	Árbol Estrella	Estrella de estrellas
Frecuencia de operación	868 MHz 902 MHz	Frecuencias LTE 700 MHz 800 MHz 900 MHz	2.4 GHz	433 MHz 868 MHz 915 MHz	868 MHz 2.4 GHz	433 MHz 780 MHz 800 MHz 900 MHz
Alcance de transmisión	10 km	10 km	1 a 3 km	10 km	Hasta varios km	15 km
Potencia de Transmisión	≤ 20 dBm	≤ 20 dBm	≤ 20 dBm	FCC/ETSI regulaciones	≤ 24 dBm	≤ 14 dBm (EU) ≤ 27 dBm (US)
Velocidad de transmisión de enlace ascendente	100 bps	200 kbps	624 kbps por sector	9.6 kbps 55.55 kbps	50 kbps 250 kbps	0.3 a 50 kbps
Tamaño de carga útil	12 bytes	1600 bytes	6 bytes a 10 KB	≤ 256 bytes	≤ 127 bytes	Definido por el usuario (242 bytes máximo)

1.4.4 LoRa

LoRa, es una clase de modulación basada en Chirp Spread Spectrum (CSS), que utiliza solo la capa física [9], mediante el uso de una señal denominada chirp. El término Chirp

(Compressed High Intensity Radar Pulse, del inglés) hace referencia a la variación en tiempo de la frecuencia de la luz (de frecuencias menores a mayores o viceversa), que puede insertarse en el interior de un pulso [10]. Durante todo un periodo de tiempo, la frecuencia de una señal senoidal se la puede aumentar (upchirp) o disminuir (downshirp), empleando todo el ancho de banda (BW) atribuido al canal; con ello se favorece una mejora en la resistencia a la interferencia de otras señales, y a su vez, una reducción en el fading de la señal, a causa del multitrayecto [11].

Con LoRa se da una mejoría en la recepción a expensas de una merma para un ancho de banda dado la velocidad de datos; esta tecnología aplica velocidades de datos de tamaño no necesariamente fijo, que se da con la manipulación del Spreading Factor (SF), con lo que se puede llevar “convenio” entre el alcance recorrido y velocidad de datos enviada, al igual que primar el rendimiento de la red con un ancho de banda constante [12].

1.4.4.1 Criterios para transmisión

A un dispositivo LoRa, se lo puede configurar a fin de usar distinta Potencia de Transmisión (TP), Frecuencia de Portadora (CF), Factor de Ensanchamiento (SF), Ancho de banda (BW) y Tasa de Codificación (CR) con el propósito de obtener el mayor rendimiento posible en la conexión y un menor gasto de energía. Con estas variables se podrían obtener en torno a 6720 configuraciones posibles, lo que concede al cliente amoldar plenamente LoRa a su interés [14].

En seguida, se expone una breve descripción de las variables antes mencionadas [13]:

- **Potencia de Transmisión (TP):** Varía entre -4 dBm y 20 dBm, no obstante, a causa de la aplicación, se la puede regular desde 2 dBm a 20 dBm. Solo el 1% del ciclo de trabajo puede ser empleado si se sobrepasa los 17 dBm.
- **Frecuencia de Portadora (CF):** Dependiendo del chip empleado o de la región de uso, la CF se puede variar con saltos de 61 Hz entre 137 MHz y 1020 MHz.
- **Factor de Ensanchamiento (SF):** Es el producto de la relación entre la velocidad de símbolos y la tasa de chip. Un SF más grande acrecienta la SNR, la sensibilidad y el alcance, asimismo el tiempo aire del paquete. Con el aumento del SF, por añadidura, la tasa de transmisión decrece a la mitad, duplica el lapso de la transmisión y consumo energético. Este puede alterarse entre 6 y 12.
- **Ancho de banda (BW):** Rango de frecuencias en la banda de transmisión. Un BW grande, producirá un menor tiempo aire (una mayor tasa de datos), sin embargo, se consigue menor sensibilidad por añadidura de ruido. Un ancho de banda menor

demanda cristales/relojes más exactos, esto implica una disminución de ppm. El envío de datos se da a una velocidad de chip correspondiente al ancho de banda; una tasa de chip de 250 kcps es homologa a un BW de 250 KHz. La red LoRa tradicional funciona a: 500 KHz, 250 KHz o 125 KHz.

- **Tasa de Codificación (CR):** CR es la tasa FEC (corrección de errores hacia adelante) utilizada en LoRa frente a interferencias y se la puede configurar con: 4/5, 4/6, 4/7 y 4/8. Ante el ruido una tasa de codificación más grande presenta una protección superior, aunque el tiempo aire se ve ampliado. Si se tiene transmisores con distinto CR estos consiguen establecer comunicación en vista de que esta tasa se halla en una cabecera del paquete que regularmente se codifica con 4/8.

1.4.4.2 Espectro Ensanchado LoRa

Como ya se explicó con anterioridad en LoRa, se obtiene el spread spectrum originando una señal chirp que cambia en frecuencia sin intermisión. Figura 1.1 muestra la ejemplificación del espectro ensanchado LoRa en el envío de un mensaje. La complejidad del receptor se ve reducida en virtud de que las variabilidades de sincronización y frecuencia entre los transceptores son equiparables, lo cual supone una ventaja al emplear este método. El BW de este chirp es semejante al BW espectral de la señal. La señal anhelada se la extiende con un chip y se la modula sobre un chirp [12].

En Ecuación 1.3 y 1.4, se establece la velocidad de bits de la modulación R_b y el Código de tasa, respectivamente.

$$R_b = SF * \frac{\text{Código de tasa}}{\frac{2^{SF}}{BW}} \left[\frac{\text{bits}}{\text{sec}} \right]$$

Ecuación 1.3 Tasa de bits de la modulación LoRa

$$\text{Código de tasa} = \frac{4}{4 + CR}$$

Ecuación 1.4 Código de tasa de la modulación LoRa

Donde:

- SF = Spreading Factor, varía entre 7 y 12
- BW = Ancho de Banda (Hz)
- CR = Coding Rate, varía entre 1 y 4

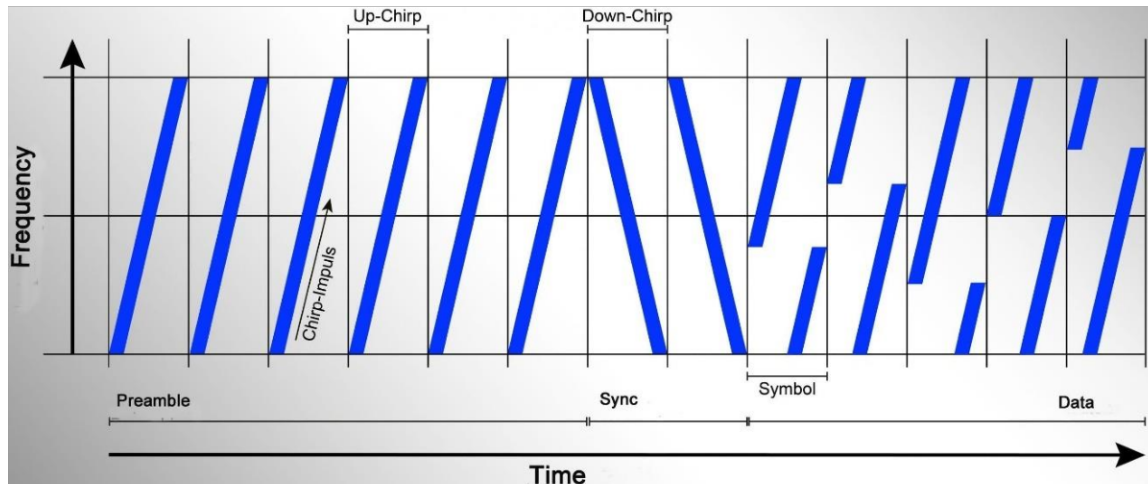


Figura 1.1 Espectrograma de un mensaje enviado [15]

1.4.4.3 Estructura de los paquetes LoRa

Dicha estructura se presenta en Figura 1.2. La sección entre el preámbulo y la carga útil se codifica a una tasa de código fija de 4/8, mientras que la tasa de código de la carga útil y el CRC de la carga útil (comprobación de redundancia cíclica, un código de detección de errores) es opcional dentro de un cierto rango. Por lo tanto, la longitud de la carga útil codificada y del CRC de la carga útil es variable y depende de la tasa de código seleccionada [16].

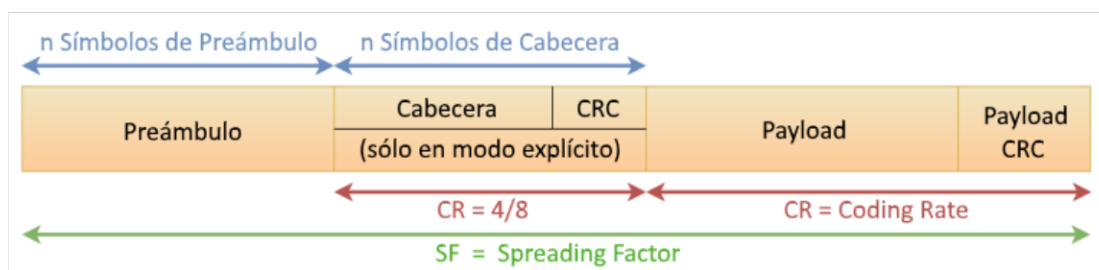


Figura 1.2 Estructura del Paquete LoRa

1.4.4.4 Propiedades importantes de LoRa

En seguida, se indican algunas particularidades significativas que destacan a Lora y la convierten en el mejor aspirante para el desarrollo de aplicaciones IoT [12] [17]:

- **Ancho de Banda Escalable:** Es posible ajustar el ancho de banda, utilizarse en banda estrecha y en banda ancha si se emplean aplicaciones de secuencia directa.
- **Robustez:** En virtud de su naturaleza asincrónica, la modulación LoRa es altamente resistente a procesos de interferencia en el interior como fuera del canal.

- **Inmunidad contra el efecto Doppler:** Un pequeño desplazamiento generado por el efecto Doppler, es mínimo en el eje X (tiempo) de la señal que está en banda base en cierta medida por la modulación Lora.
- **Resistente al desvanecimiento:** LoRa es ideal para ambientes suburbanos y urbanos ya que presenta una gran resistencia al desvanecimiento y la multitrayectoria.
- **Capacidad de Amplia Cobertura:** Si se la compara con FSK, LoRa presenta un mayor presupuesto de enlace manteniendo la misma potencia de transmisión.
- **Bajo consumo energético:** Se puede sostener o mejorar el presupuesto de enlace si se reduce la potencia de salida en comparación con FSK.

1.4.5 LoRaWAN

La especificación LoRaWAN es un protocolo LPWA desarrollado para implementar una conexión de forma inalámbrica de “cosas” que operan con baterías a Internet en redes regionales, nacionales o mundiales. Esta permite movilidad, comunicación bidireccional, servicios de localización y seguridad extremo a extremo, que forman parte de los requisitos clave para el desarrollo de aplicaciones relacionadas al IoT [18].

Una topología clásica LoRaWAN se la despliega en forma de Estrella, tiene una composición dada por estos principales elementos: dispositivos finales también denominados nodos, gateways/pasarelas, un servidor de red y por último un servidor de aplicación. Los nodos en general, se hallan constituidos por sensores/actuadores, los cuales transmiten mediante la capa física de LoRa los datos recopilados al Gateway. Este último se encarga de recibir dichos datos y la comparte con el servidor por intermedio de una conexión IP, como redes celulares o redes WiFi [19].

1.4.5.1 Arquitectura de red

La arquitectura LoRaWAN se basa en una topología en Estrella misma que permite asegurar un enlace eficiente entre ambos dispositivos, a más de un gasto de energía inferior para los nodos.

Como se describió con anterioridad los nodos son los responsables de coleccionar datos correspondiente a diferentes tipos de sensores, para más adelante enviar los datos a los gateways, valiéndose de las pautas del protocolo LoRaWAN. En Figura 1.3, se aprecia cada uno de los elementos que conforman de la arquitectura antes descrita.

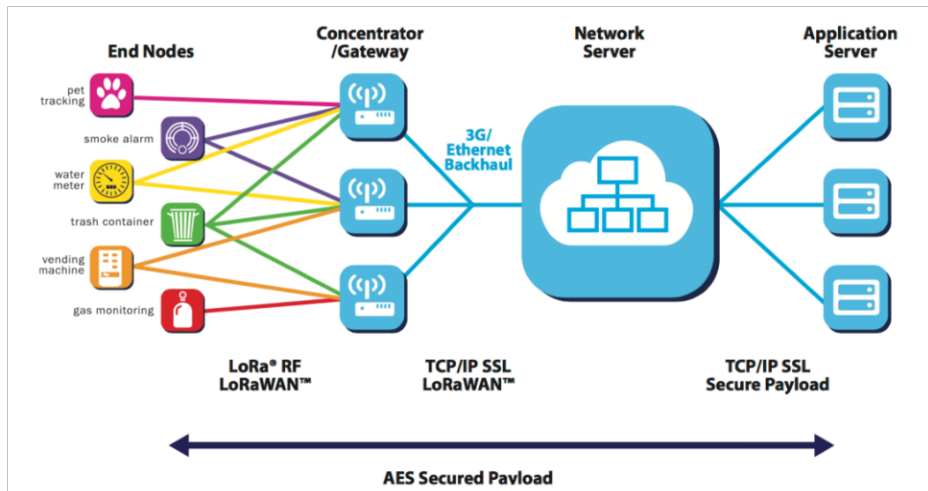


Figura 1.3 Arquitectura de red LoRaWAN [20]

1.4.5.2 Clases de Dispositivos LoRaWAN

En [21], se precisa tres clases para dispositivos finales LoRaWAN. Dichas clases poseen distintas características, con las cuales, el usuario podrá efectuar diversas aplicaciones basándose en el gasto de energía. Seguidamente, se detallan de las clases determinadas para LoRaWAN.

- **A:** por lo regular destinada a una gran parte de dispositivos finales o nodos. Permite una comunicación bilateral, la conexión de Downlink es establecida en secciones de tiempo, con lo que se obtiene un menor consumo de energía. Dispositivos de basen su funcionamiento con baterías son los idóneos para esta clase.
- **B:** Análoga a la clase A. En la clase B también se admite una conexión bilateral, pero con una sutil diferencia, el Gateway programa los slots de recepción. Esto incrementa el consumo energético del dispositivo final [21].
- **C:** Se encuentra ininterrumpidamente en periodo de escucha, a condición que no exista una transmisión. Esto ocasiona un elevado consumo energético, no obstante, a comparación de las clases anteriores la latencia de Downlink es reducida [21] [22].

1.4.6 TOPOLOGÍAS LINEALES

Las diversas soluciones LPWAN como Sigfox, NB-IoT, LTE-M, etc., para implementar redes inalámbricas de sensores (WSN, Wireless Sensor Network) han tenido una gran aceptación ya sea por su despliegue en zonas sin infraestructura invasiva, bajo coste de implementación u operación. Estas redes de sensores se presentan por lo regular en

topologías tipo árbol (cluster-tree), malla y estrella cuyos protocolos de enrutamiento son demasiado complejos e ineficientes si se desea hacer una monitorización de tuberías, oleoductos, fronteras, trenes y de próximas aplicaciones con infraestructuras lineales [23].

Para reducir el riesgo que supone el despliegue de las WSN, se aplican varias técnicas. Una de ellas es la manipulación de la topología. La topología de una red se refiere a la forma en que los nodos están conectados entre sí en una red. Este trabajo se centra en la topología lineal, la arquitectura de una topología lineal multisalto se clasifica en dos estructuras principales: la topología plana de un nivel y la topología jerárquica de varios niveles, como se ilustra en las figuras 1.4 y 1.5.

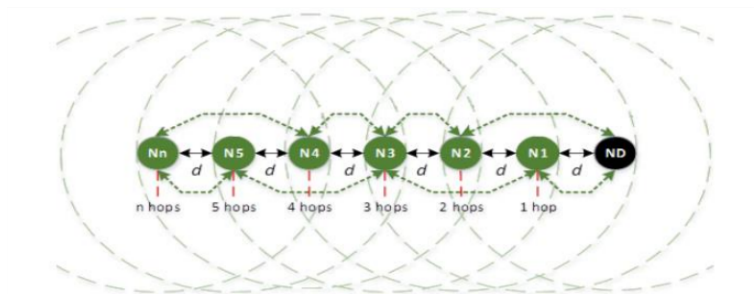


Figura 1.4 Topología Lineal Multisalto WSN [24]

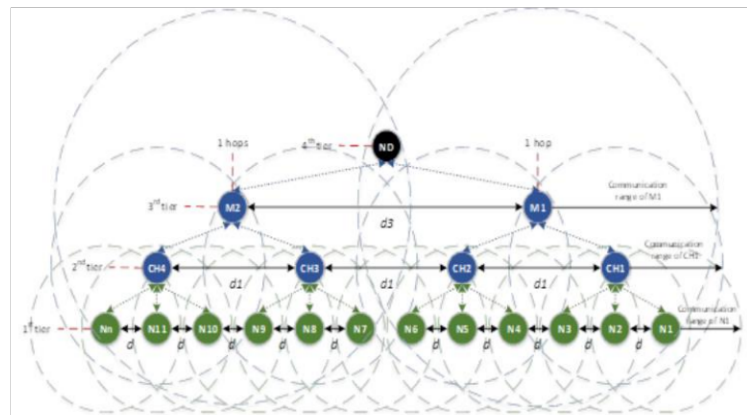


Figura 1.5 Jerarquía WSN [24]

Este tipo de redes multisalto de un solo nivel se distingue por el reducido número de nodos que se hallan dentro del área de cobertura de un nodo sensor, por los centenares de nodos que constituyen la red y por la inmensa cifra de nodos retransmisores por los que tiene que transitar el mensaje para que llegar al nodo final (Nn) [24].

1.4.7 ESTADO DEL ARTE

En este subcapítulo se realiza un breve resumen de los artículos técnicos revisados en el estudio de la materia antes vista.

En [5], se presenta un enfoque que aborda gastos generales (escalabilidad/confiabilidad, baja latencia, baja potencia y bajo costo operativo) al permitir que los dispositivos se autoorganicen y determinen sus posiciones de ranura en un marco de forma autónoma. Además de eso, se utilizó una ranura dedicada en cada trama para garantizar la sincronización global y manejar los reconocimientos. En sus resultados experimentales con 25 nodos, muestran que TS-LoRa puede lograr una relación de entrega de paquetes de más del 99% incluso para los nodos más distantes.

En [6], se ofrece un estudio completo y comparativo de las tecnologías LWAN, que sirven como soluciones eficientes para conectar dispositivos inteligentes, autónomos y heterogéneos. Se demuestra que Sigfox y LoRa son ventajosas en términos de duración de la batería, capacidad y coste. Por su parte, NB-IoT ofrece ventajas en términos de latencia y calidad de servicio.

En [9], se analiza el impacto de los ajustes de comunicación de LoRa en cuatro casos de uso de IoT, medición inteligente, aparcamiento inteligente, alumbrado público inteligente y seguimiento de flotas de vehículos. Los resultados demuestran que la configuración correspondiente a la velocidad de datos más rápida consigue un ratio de entrega de paquetes hasta un 380% mayor de entrega de paquetes y utiliza 0,004 veces la energía en comparación con otras configuraciones evaluadas, siendo adecuada para soportar los cuatro ejemplos de casos de uso del IoT.

En [14], se presenta un análisis de rendimiento y capacidad de un transceptor LoRa disponible actualmente. Describimos sus características y demostramos cómo dicho transceptor puede utilizarse de forma eficiente en un escenario de aplicación de área amplia. En particular, demostramos cómo se pueden emplear características únicas como las transmisiones concurrentes no destructivas y la detección de portadoras.

En [23], se define que previo a un análisis de las funciones que tiene la capa de red, esta no es requerida en aplicaciones (redes de sensores) que utilicen una topología lineal.

En [24], los autores presentan un análisis detallado del rendimiento de los protocolos de ruteo: AODV (Ad hoc On Demand Distance Vector), DSDV (Destination Sequenced Distance Vector) y FIXRT (Fix-Route Routing), con un tamaño de paquete de 32 y 128

utilizando una topología lineal (industria del petróleo y el gas) de acuerdo con los estándares IEEE 802.11 mediante el simulador ns2.35.

En [34], se presenta el procedimiento para obtener la potencia promedio de una señal de radio móvil. Se determina que la longitud medida de una señal de radio móvil necesaria para obtener la potencia promedio está en el rango de 20 a 40 longitudes de onda. El número suficiente de muestras para estimar los valores de potencia promedio local es de aproximadamente 36. Se basa en un intervalo de confianza del 90 por ciento y un error estimado de menos de 1 dB. Este procedimiento de estimación puede tratarse como un procedimiento estándar.

2 METODOLOGÍA

La metodología a emplear se compone de cuatro fases: inicialmente la investigación revisada en el marco teórico, como segunda el diseño funcional, la tercera fase comprende la construcción y configuración del prototipo, y en última se realizó las pruebas pertinentes para alcanzar los objetivos planteados.

2.1 INTRODUCCIÓN

La reproducibilidad es la propiedad por la que se obtienen resultados consistentes utilizando los mismos datos, pasos computacionales, métodos, código y condiciones de análisis [25]. Para conseguirla, es esencial que el procedimiento aplicado coincida a posteriori con las acciones de usuario, por ende, en este capítulo se trata de brindar una clara información de aspectos técnicos y parámetros de configuración en los dispositivos usados.

Un importante aspecto a considerar también, es realizar un estudio de campo con el cual se logra detallar el entorno a profundidad con el fin de precisar el mayor número de las variables que podrían repercutir en la obtención de datos.

2.2 ELEMENTOS A UTILIZAR EN LA IMPLEMENTACION DEL PROTOTIPO

En esta sección realiza una exposición del módulo LoRa SX1276 y la tarjeta de desarrollo NodeMCU ESP8266. La unión de ambos dispuestos tanto en el nodo principal (origen) como los retransmisores (repetidores/destino). Se describe también, módulos de alimentación o fuentes de conversión como el TP4056 y MT3608 respectivamente. A su vez el empleo de baterías 18650 para dotar a los nodos movilidad.

2.2.1 TRANSCEPTORES LoRa SX1276

Los transceptores SX1276 provee una conexión de espectro amplio de alcance ultra largo e inmunidad al ruido, a la vez que merma el uso de corriente. Por intermedio de la técnica de modulación exclusiva LoRa de Semtech, el SX1276 puede alcanzar una sensibilidad superior a -148 dBm usando un cristal y materiales cuyo costo es bajo. Este transceptor

ofrece un gran presupuesto de enlace debido a la combinación de una alta sensibilidad con un amplificador integrado de +20 dBm. El SX1276 presenta selectividad, linealidad de receptor y IIP3 para consumo de corriente notablemente inferior al ofertado por dispositivos de la competencia [26].

El modelo elegido, como se muestra en Figura 2.1, para este trabajo es el HPD13A V1.1 de la empresa HPDTeK. Cabe mencionar que la elección de este transceptor se dio, debido a su fácil adquisición en el mercado nacional. La tabla 2.1, presenta un resumen de las principales características técnicas que presenta este transceptor [26].

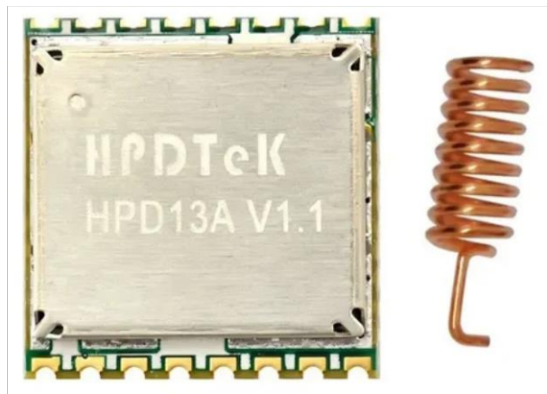


Figura 2.1 Transceptor LoRa SX1276 [27]

Tabla 2.1 Características técnicas del transceptor SX1276 [27]

Recurso	Descripción
Frecuencia de Trabajo	915 MHz
Modulaciones	FSK, GFSK, MSK, GMSK, LoRa
Potencia de Transmisión	20 dBm
Sensibilidad	-148 dBm mA si se trabaja con FSK
Velocidad de Transmisión	200 kbps
Corriente de Emisión	120 mA si se trabaja a 20 dBm
Corriente de Recepción	10.3 mA
Corriente de Espera/Reposo	0.2 uA
Interfaz de Datos	SPI
Distancia de conexión	8000 m si se trabaja con LoRa
Temperatura de Operación	-20 a 70°C
Alimentación	1.8 – 3.7 V DC
Dimensiones	16 mm x 16 mm
Impedancia Antena	50 ohm

En el Anexo 1, se muestra el diagrama de pines, mismo que han sido extraídos de su página oficial.

2.2.2 NodeMCU ESP8266

NodeMCU es un entorno que se basa en ESP8266 de código abierto en cual posibilita conectar enlazar y admitir la transmisión de información por medio del protocolo Wi-Fi. Además, de proporcionar las características más importantes de los microcontroladores como GPIO, PWM, ADC, etc., con los cuales se puede solucionar muchas de las exigencias de un proyecto [28]. En Figura 2.2 desarrollada por el expositor, se visualizan los elementos que conforman la placa y en la Tabla 2.2 algunas de sus características técnicas más importantes.

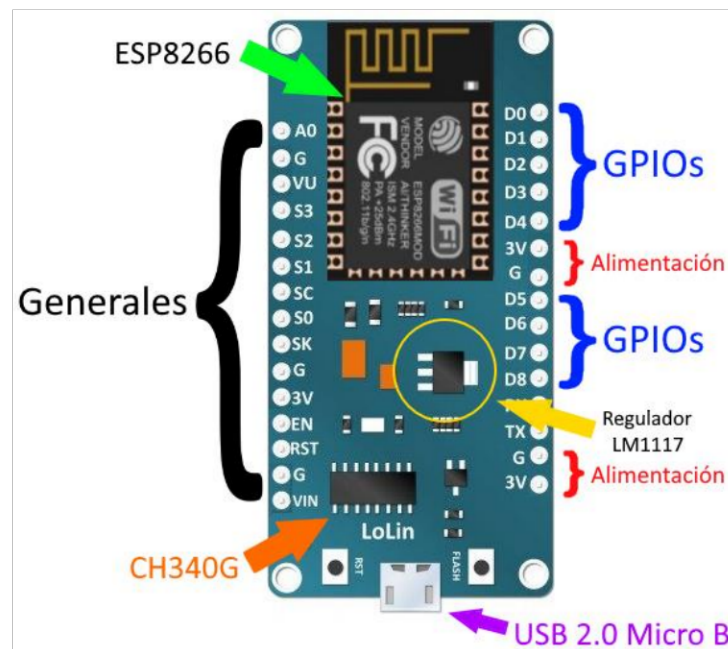


Figura 2.2 Elementos que conforman la placa NodeMCU ESP8266

Tabla 2.2 Características técnicas del NodeMCU ESP8266 [28]

Recurso	Descripción
SoM	ESP-12E
SoC	ESP8266
CPU	Tensilica Xtensa LX3 (32 bit)
Frecuencia de Trabajo	80 MHz / 160 MHz
Convertor USB a Serial	CH340G

Voltaje de Operación	3.3 V DC
Flash Memory/SRAM	4 MB / 64 KB
Pines I/O Digitales	11
Pines I/O Analógicos	1
WiFi integrado	802.11 b/g/n
Potencia de Salida	+19.5 dBm
Corriente de Fuga	< 10 uA
Interfaz de Datos	UART/SPI/I2C
Distancia de conexión	8000 m si se trabaja con LoRa
Temperatura de Operación	-40 a 125°C
Alimentación	4.5 V - 10 V DC
Dimensiones	58 mm x 32 mm
Antena	En PCB

En el Anexo 2, se muestra la distribución de pines del NodeMCU ESP8266

2.2.3 MÓDULO MT3608

Este módulo cuenta con el convertidor/elevador MT3608, el cual puede tomar voltajes de dos voltios a su entrada y elevarlos hasta veintiocho voltios. El MT3608 incluye limitación de corriente, protección por bajo voltaje, y contra sobrecarga térmica. Este módulo que se expone en Figura 2.3 tiene un potenciómetro de varias vueltas que puede utilizar para ajustar el voltaje de salida. Dado que el potenciómetro de exactitud tiene veinticinco vueltas de ajuste, puede regular con facilidad la salida del módulo exactamente al voltaje que se necesite (en el rango dado) [29].

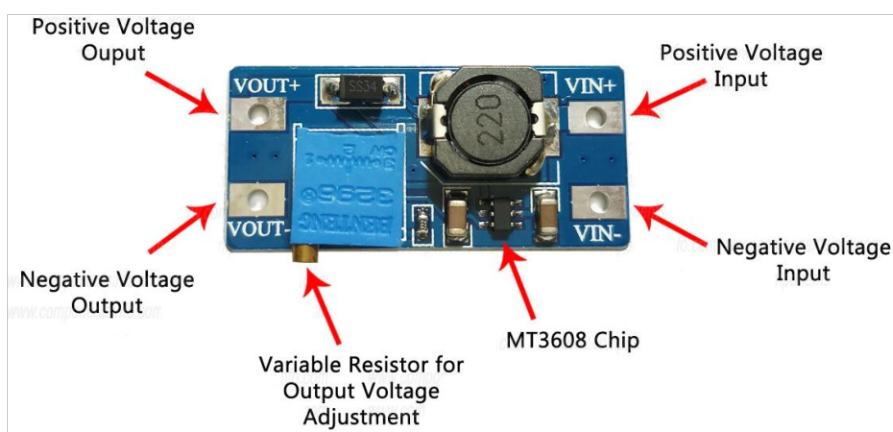


Figura 2.3 Elementos que conforman el módulo MT3608 [30]

Los convertidores de impulso son una excelente manera de aumentar fácilmente un voltaje dado, pero ese impulso tiene el costo de una menor corriente de salida dada en Ecuación 2.1, en comparación con la corriente de entrada dado que la potencia máxima de salida es de 6 vatios.

$$\text{Corriente de salida} = \frac{6 \text{ Watts}}{\text{Voltaje a la Salida}}$$

Ecuación 2.1 Corriente de Salida para el módulo MT3608

La tabla 2.3, elaborada por mi persona presenta un resumen de las principales características técnicas del módulo convertidor/elevador.

Tabla 2.3 Características técnicas del módulo MT3608

Recurso	Descripción
Voltaje de entrada	2V a 24V DC
Voltaje de salida	5V a 28V DC (Regulable por trimmer)
Corriente de salida máxima	2A
Frecuencia de Trabajo	1.2 MHz
Potencia de salida	6W
Eficiencia de conversión	93% máx
Protección de sobre-temperatura	SI
Protección de corto circuito	NO
Protección limitadora de corriente	SI (4A)
WiFi integrado	802.11 b/g/n
Dimensiones	36 mm x 17 mm

2.2.4 MÓDULO TP4056

El TP4056 es un cargador lineal de corriente/tensión constante usado para recargar baterías de polímero de litio y iones de litio. La tensión y corriente de carga está fijadas en 4,2 V y 1 A, repetitivamente, por una resistencia externa. El ciclo de carga termina automáticamente cuando la corriente de carga cae a 1/10 del valor programado después de alcanzar la tensión de flotación final. Incluye otras características como monitor de corriente, bloqueo por bajo voltaje, recarga automática y dos pines de estado para indicar la terminación de la carga y la presencia de una tensión de entrada En Figura 2.4 se puede visualizar los componentes que conforman este módulo [31].

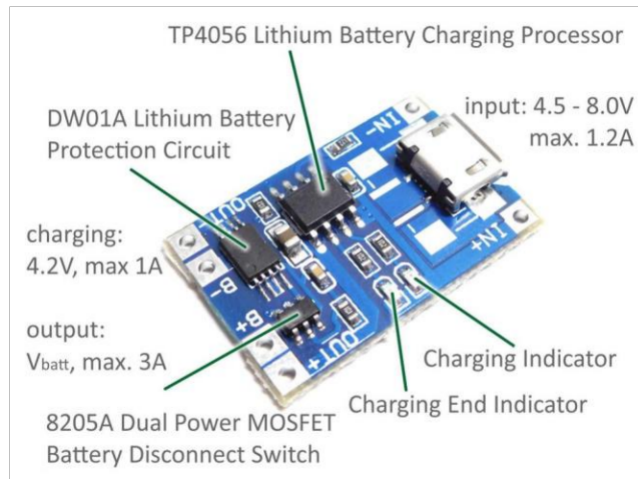


Figura 2.4 Elementos que conforman el módulo TP4056

2.2.5 BATERÍA 18650

Una batería 18650 es una batería de iones de litio. La batería 18650 tiene un voltaje de 3.6v y tiene entre 2600mAh y 3500mAh (mili-amperios-hora). Estas baterías se utilizan en computadoras portátiles, productos electrónicos e incluso algunos automóviles eléctricos debido a su confiabilidad, largos tiempos de funcionamiento y capacidad para recargarse cientos de veces. Son lo que se consideraría una "batería de alto consumo". Esto significa que la batería está diseñada para generar un alto voltaje y corriente de salida para satisfacer las demandas de energía del dispositivo portátil en el que se está utilizando [32]. En Figura 2.5 se muestra la composición interna de una batería 18650.

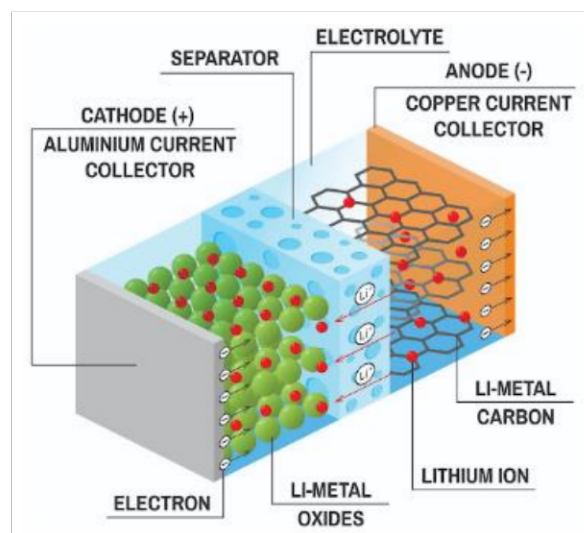


Figura 2.5 Composición interna de una batería 18650 [33]

La tabla 2.4, elaborada por mi persona presenta un resumen de las principales características técnicas de la batería 18650.

Tabla 2.4 Características técnicas de la batería 18650

Recurso	Descripción
Voltaje nominal	3.6 V DC
Capacidad Nominal	2.850 mAh
Voltaje de descarga mínimo	3 V
Voltaje de carga	4,2 V (máximo)
Tiempo de carga	3 horas (aprox.)
Método de carga	CC y CV
Peso de la celda	48 g (aprox.)
Dimensiones	1.4 mm x 65 mm

2.3 EVALUACION DEL RETARDO

Para llevar a cabo el proceso de medición, se envió diversos paquetes desde un nodo origen para que los nodos subsiguientes (repetidores) los replicaran al nodo destino en una estructura lineal. Tanto el nodo origen, destino, como los repetidores fueron acomodados a una altura de uno y medio metros de manera fija sobre el nivel del terreno, con el fin de evitar obstáculos propios en los ambientes exigidos.

2.3.1 DISTRIBUCIÓN DE LOS EQUIPOS

La forma de la disposición en la que se encuentran los equipos se detalla en Figura 2.6, la cual fue elaborada por mi persona. El nodo origen será monitoreado mediante la interfaz USB por un computador, los nodos subsiguientes estarán equidistantes entre ellos de manera sucesiva desde el nodo principal.

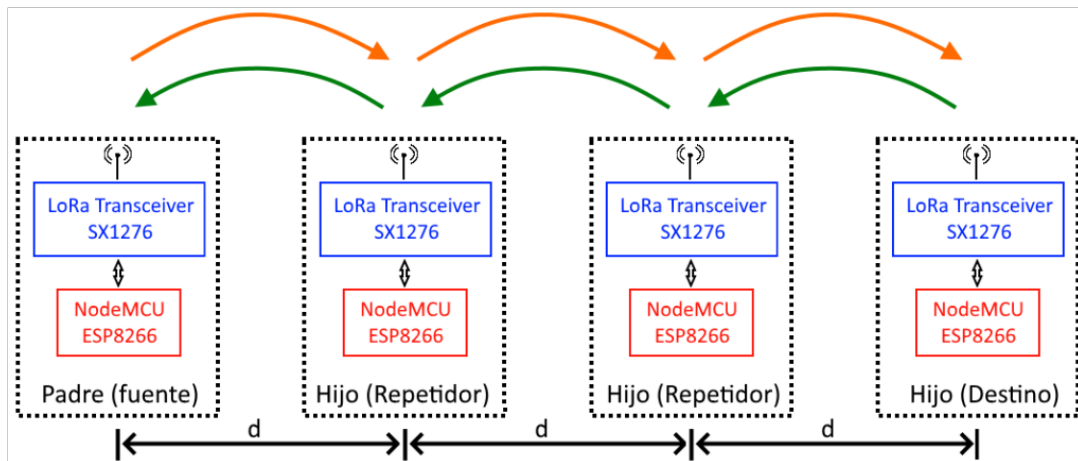


Figura 2.6 Distribución de Equipos

2.3.2 INTERVALO DE MEDIDA

Para la obtención de datos, se siguió el denominado promediado espacial que es un proceso de medición propuesto por [34]. Conforme a dicho procedimiento, se debe efectuar las mediciones entre 20 lambda y 40 lambda para una frecuencia dada.

Si bien LoRa, realiza saltos en frecuencia pseudoaleatorios, para efectuar el cálculo en Ecuaciones 2.2, 2.3 y 2.4, se tomó en cuenta la frecuencia en que funcionan los dispositivos. Una frecuencia de 915 MHz para nuestro estudio.

$$\lambda = \frac{c}{f} = \frac{3 * 10^8}{915 * 10^6} \left[\frac{m/s}{Hz} \right] = 0.3279 m$$

Ecuación 2.2 Lambda para la frecuencia central

$$\text{Límite inferior} = 20 * \lambda = 6.56 m$$

Ecuación 2.3 Límite inferior

$$\text{Límite superior} = 40 * \lambda = 13.11 m$$

Ecuación 2.4 Límite inferior

Para este trabajo se escogió la separación entre nodos de 10 metros, así como sus múltiplos hasta un máximo de 250 metros como se indica en la tabla 2.5.

Tabla 2.5 Múltiplos de la separación escogida entre nodos

Distancia entre nodos – d (m)					
10	20	40	80	150	250

2.3.3 IMPLEMENTACION DE LOS NODOS

2.3.3.1 Física

Si bien en el mercado existe nodos comerciales que se podían emplear para cumplir los objetivos fijados en este trabajo. Se decidió efectuar la construcción de estos debido principalmente a costos.

Los diagramas empleados, así como esquematización y diseños en PCB se los puede encontrar en el Anexo 3. Finalmente, la estructura final del nodo, así como su montaje se puede visualizar en el Anexo 4.

Los módulos usados se deben preparar con una configuración y modificación, para adecuar los factores de transmisión al método elegido.

El chip SX1276 (LoRa) es manejado por el microprocesador de la tarjeta NodeMCU ESP8266 por medio del estándar de comunicaciones SPI, quien aparte de realizar la configuración de los parámetros de transmisión (Frecuencia, Spread Factor, potencia, etc.), también se encarga de la recopilación de los datos enviados y recibidos.

En Figura 2.7, elaborada por mi persona. Se muestra la configuración general en bloques de los nodos

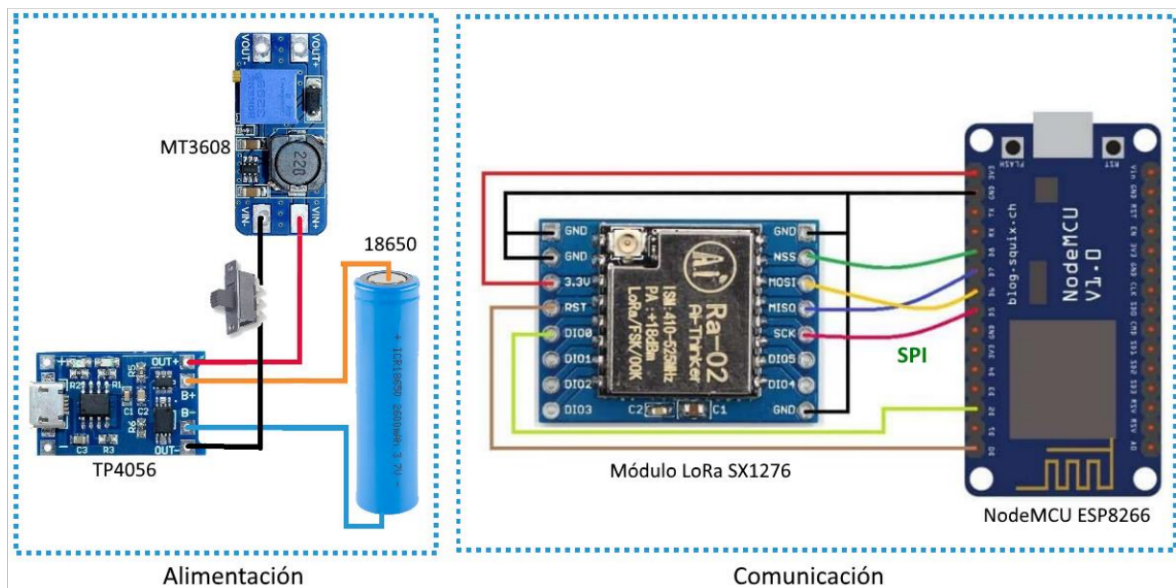


Figura 2.7 Configuración General de los nodos

2.3.3.2 Diseño del diagrama de flujo de la operación de los nodos

Hay más de una forma de transferir diversos paquetes desde un nodo de origen a un nodo de destino. Antes de emplear la retransmisión, una posibilidad es enviar todos los paquetes al primer repetidor. Cuando todos los paquetes llegan a éste, comienza a retransmitir. Otro recurso es retransmitir cada paquete tan pronto como llegue a cada punto de retransmisión. Siendo este último mecanismo de transmisión teóricamente más rápido que el anterior.

Esto se lo puede verificar comparando la Figura 2.8 (a) cuando en el instante 15 todos los paquetes se hallan en el tercer repetidor, contra la Figura 2.8 (b), en la cual los paquetes se encuentran en el tercer repetidor en el instante 12. Dicha figura fue elaborada por mi persona.

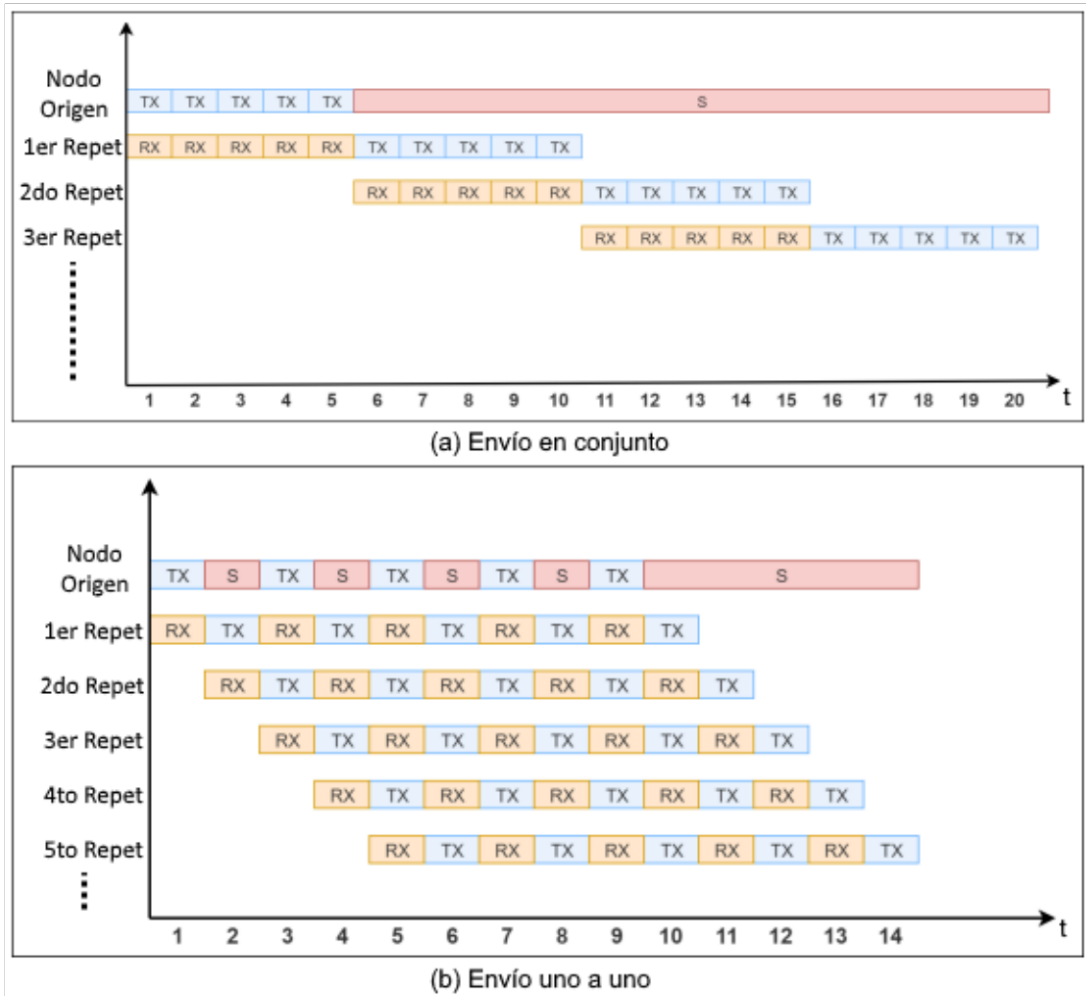


Figura 2.8 Comparativa transferencia de paquetes

En Figura 2.9, elaborada por mi persona, se presenta el diagrama de flujo para la transmisión de datos desde un nodo origen a un nodo destino, mediante el empleo de un repetidor. Es muy parecida a la conexión extremo-extremo que se tiene con dos nodos, pero a diferencia de esta, cuando el repetidor fija un enlace con el nodo, este responde con un paquete ACK/WAIT al nodo.

2.3.4 PROGRAMACIÓN DE LOS NODOS CON IDE ARDUINO Y LIBRERÍA RADIOHEAD

2.3.4.1 IDE Arduino y Librería RadioHead

Teniendo un modelo de referencia como el visto en el apartado anterior, se utilizará la IDE de Arduino y la librería RadioHead Packet Radio para microprocesadores embebidos. Esta librería proporciona una completa biblioteca orientada a objetos, para remitir/admitir datos por medio de una amplia variedad de radios [35].

RadioHead consta de dos conjuntos principales de clases: Controladores y Gestores.

- Los controladores proporcionan acceso de bajo nivel a una serie de radios.
- Los gestores proporcionan funciones de envío y recepción de mensajes de alto nivel.

Para el transceptor LoRa HPD13A V1.1 se empleará el controlador RH_RF95, el cual funciona con los módulos Semtech SX1276/77/78/79, Modtronix inAir4 e inAir9, y HopeRF RFM95/96/97/98 y otras radios similares con capacidad LoRa; en conjunto con los gestores RHReliableDatagram y RHRouter, con los cuales se puede implementar una conexión multisalto desde el nodo de origen al nodo de destino a través de 0 o más nodos intermedios, con direccionamiento manual [35].

2.3.4.2 Codificación del diagrama de flujo

A continuación, se presenta la conformación de cada uno de los dispositivos empleados para hacer la toma de medidas.

La configuración general viene dada por la declaración de variables a utilizar.

Inclusión de librerías a utilizar:

```
1 #include <RHRouter.h>
2 #include <RH_RF95.h>
3 #include <SPI.h>
```

Un direccionamiento para lograr que la comunicación sea llevada a cabo de manera lineal, cumpliendo así con la topología señalada con anterioridad.

```

5 #define PADRE_ADDRESS 1
6 #define REPETIDOR1_ADDRESS 2
7 #define REPETIDOR2_ADDRESS 3
8 #define DESTINO_ADDRESS 4

```

Cumpliendo con la normativa de LoRa Alliance, se fija la frecuencia de 915 MHz, con una potencia de transmisión de 17 dBm, y un tamaño de preámbulo de 8. Si bien se puede realizar combinaciones con el ancho de banda, SF y CR. Se utilizó a configuración por default de la normativa.

```

10 // CONFIG FIJA
11 #define RADIO_FREQUENCY 915.0
12 #define RADIO_TX_POWER 17
13 #define RADIO_PREAMBLELENGTH 8
14
15 // CONFIG VARIABLE
16
17 // Defaut, parámetros regionales LoRa
18 #define RADIO_SIGNALBANDWIDTH 125000
19 #define RADIO_SPREADFACTOR 7
20 #define RADIO_CODINGRATE 5
21 /*

```

Declaración de pines para comunicación SPI e Instancia Singleton para el controlador de Radio.

```

36 // DEFINO PINES SPI PARA NodeMCU ESP8266
37 #define RFM95_CS 15
38 #define RFM95_RST 5
39 #define RFM95_INT 4
40
41 // DEFINO INSTANCIA Singleton DEL CONTROLADOR DE RADIO
42 RH_RF95 rf95(RFM95_CS, RFM95_INT);

```

A cada nodo le corresponde una dirección única, por lo cual esta debe ser definida. En este caso revisamos la configuración del nodo origen o padre.

```

44 // DEFINO LA DIRECCION DEL NODO
45 RHRouter manager(rf95, PADRE_ADDRESS);

```

La estructuración del lazo Setup y Loop cambia un tanto para el padre como para los hijos/destino, ya que se necesita un direccionamiento específico para cada nodo.

Las tablas de direccionamiento elaboradas en base a la Figura 2.6 para indicar los saltos a realizar del nodo padre, repetidor 1, repetidor 2 y destino, se detallan a continuación en la Tabla 2.6, Tabla 2.7, Tabla 2.8 y Tabla 2.9, respectivamente.

Tabla 2.6 Direccionamiento para Nodo Origen

Nodo (me ubico en)	Hacia donde quiero llegar	Siguiente salto para llegar
Origen	Repetidor 1	Repetidor 1
Origen	Repetidor 2	Repetidor 1
Origen	Destino	Repetidor 1

Tabla 2.7 Direccionamiento para Nodo Repetidor 1

Nodo (me ubico en)	Hacia donde quiero llegar	Siguiente salto para llegar
Repetidor 1	Origen	Origen
Repetidor 1	Repetidor 2	Repetidor 2
Repetidor 1	Destino	Repetidor 2

Tabla 2.8 Direccionamiento para Nodo Repetidor 2

Nodo (me ubico en)	Hacia donde quiero llegar	Siguiente salto para llegar
Repetidor 2	Origen	Repetidor 1
Repetidor 2	Repetidor 1	Repetidor 1
Repetidor 2	Destino	Destino

Tabla 2.9 Direccionamiento para Nodo Destino

Nodo (me ubico en)	Hacia donde quiero llegar	Siguiente salto para llegar
Destino	Origen	Repetidor 2
Destino	Repetidor 1	Repetidor 2
Destino	Repetidor 2	Repetidor 2

En el lazo de Setup del archivo a utilizar, aparte de la configuración de los parámetros, se debe introducir estos direccionamientos para que se realicen los saltos de manera lineal.

```

47 void setup()
48 {
49     Serial.begin(115200);
50     if (!manager.init())
51         Serial.println("Inicio Fallido");
52     rf95.setFrequency(RADIO_FREQUENCY);
53     rf95.setTxPower(RADIO_TX_POWER);
54     rf95.setSignalBandwidth(RADIO_SIGNALBANDWIDTH);
55     rf95.setSpreadingFactor(RADIO_SPREADFACTOR);
56     rf95.setPreambleLength(RADIO_PREAMBLELENGTH);
57     rf95.setCodingRate4(RADIO_CODINGRATE);
58     // ASIGNO LOS SALTOS PARA CADA NODO (PARA NODO CLIENTE)
59     manager.addRouteTo(REPETIDOR1_ADDRESS, REPETIDOR1_ADDRESS);
60     manager.addRouteTo(REPETIDOR2_ADDRESS, REPETIDOR1_ADDRESS);
61     manager.addRouteTo(DESTINO_ADDRESS, REPETIDOR1_ADDRESS);
62 }

```

En el lazo Loop del nodo Origen se colocó hacia donde quiero que se dirija el mensaje, para este caso hacia el Destino.

```

65 void loop()
66 {
67     while(1)
68     {
69         Serial.println("Enviando al servidor destino");
70         uint8_t data[] = "Hola vengo desde el Transmisor!";
71         //HACIA DONDE QUIERO ENVIAR LA INFORMACIÓN
72         if (manager.sendtoWait(data, sizeof(data), DESTINO_ADDRESS) != RH_ROUTER_ERROR_NONE)
73             Serial.println("SendtoWait Fallido");
74         else
75     {

```

La programación empleada para cada nodo (origen y repetidores) se presenta detalladamente en Anexo 5 y 6.

2.3.5 CARACTERÍSTICAS DEL ENVÍO DE PAQUETES

Los dispositivos empleados admiten una vasta serie de configuraciones para la conexión, pero se limitó el Spreading Factor a un solo valor, SF7 para el estudio, conservando también un BW de 125 KHz y Coding Rate igual a 4/5, esto con el fin de preservar las especificaciones de los parámetros regionales.

Los paquetes incluyen en su contenido un texto que indica la procedencia de envío, la dirección local, la dirección hacia el próximo salto, la id de identificación del mensaje y la longitud del buffer que no sobrepasar 255 bytes sino la información enviada será recortada.

Al usar la configuración de SF7, cada paquete se transmitirá con un BW de 125 KHz y 5.4 Kbps con la una velocidad de bits. Si bien se recomienda aumentar el intervalo entre

transmisiones con el fin de disminuir o aumentar la probabilidad de colisiones, para este estudio no se lo empleará, ya que no se requiere aumentar retardos innecesarios en los nodos. Además, el sistema está diseñado provee retransmisiones (hasta tres reintentos con un tiempo de espera por defecto de 200 ms), esto según la documentación de la biblioteca RadioHead [36], con lo que la pérdida de paquetes será ínfima.

2.3.6 CARACTERÍSTICAS DE LA RECEPCIÓN DE PAQUETES

Al ser este un trabajo de evaluación de los retardos en redes LoRa multisalto con topología lineal, y por facilidad en el guardado de los paquetes recibidos, se eligió no añadir código extra que genere un archivo .csv, con el propósito de no generar más retardos de procesamiento.

Los valores de la carga útil, RSSI, SNR, número de paquetes enviados y recibidos, se los puede obtener y acopiar de una forma más idónea realizando una captura del monitor serial de la IDE de Arduino.

Si bien existen dos formas para formas de realizar los múltiples saltos a través de los nodos, la unidireccional y la bidireccional, se escogió la segunda como se señala en 2.3.1, esto con la finalidad de obtener una monitorización constante en los nodos desde un solo punto. Los tiempos de retardos en este caso medirán la ida y retorno de paquetes.

En Figura 2.10 se visualiza las capturas efectuadas con el monitor serial y guardadas en un archivo de texto (formato .txt)

```
final a 10 metros.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
13:56:43.672 -> SNR:9
13:56:43.672 -> #Pckts Rx: 20
13:56:43.672 -> #Pckts Tx: 20
13:56:43.672 ->
13:56:43.672 -> Enviando al servidor destino
13:56:44.135 -> Recibió una petición de : 0x4
13:56:44.135 -> Mensaje: Hola de vuelta a ti desde Destino
13:56:44.135 -> RSSI: -71
13:56:44.135 -> SNR:9
13:56:44.135 -> #Pckts Rx: 22
13:56:44.135 -> #Pckts Tx: 22
13:56:44.135 ->
13:56:44.135 -> Enviando al servidor destino
13:56:44.599 -> Recibió una petición de : 0x4
13:56:44.599 -> Mensaje: Hola de vuelta a ti desde Destino
13:56:44.599 -> RSSI: -68
13:56:44.599 -> SNR:9
13:56:44.599 -> #Pckts Rx: 24
13:56:44.599 -> #Pckts Tx: 24
13:56:44.599 ->
13:56:44.599 -> Enviando al servidor destino
13:56:45.061 -> Recibió una petición de : 0x4
13:56:45.061 -> Mensaje: Hola de vuelta a ti desde Destino
13:56:45.061 -> RSSI: -68
```

Figura 2.10 Obtención de paquetes mediante desde el origen al destino mediante tres saltos

2.3.7 EVALUACIÓN MATEMÁTICA DEL RETARDO

En la sección 1.4.1, ya se examinó que el retardo en el nodo está compuesto por el retardo de la cola y de procesamiento, pero según lo examinado en [36], no se tiene un retardo de la cola usando ese gestor (driver). Por tanto, se empleará un diagrama símil al retardo en comunicación de paquetes, en Figura 2.11, elaborada por mi persona, se simboliza el envío de dos paquetes con dimensión $H+p$ [bits] a un RX, por intermedio de 2 nodos intermedios.

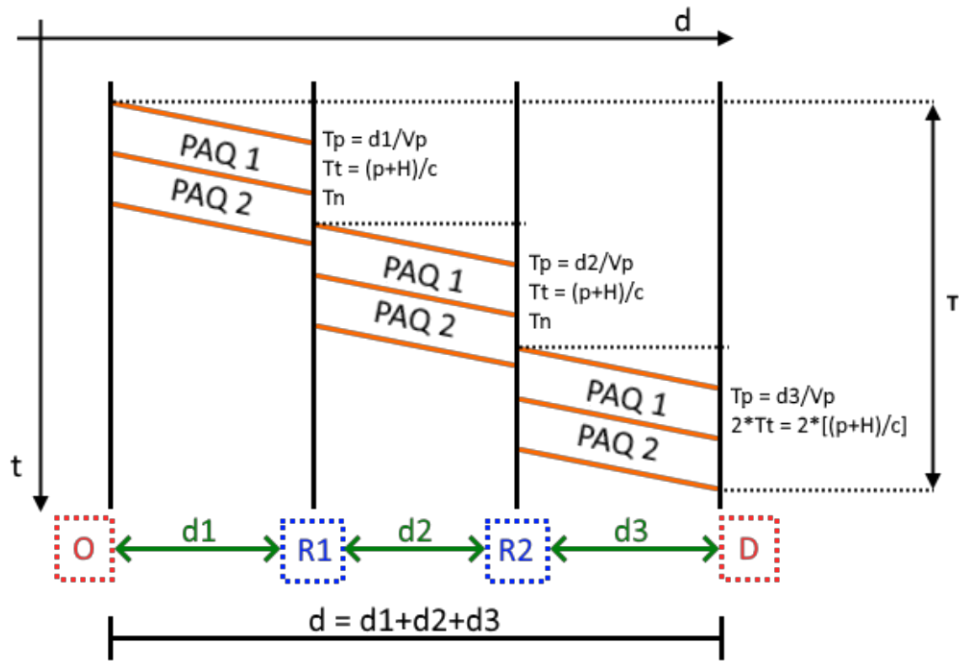


Figura 2.11 Diagrama envío de paquetes

Y la expresión usual del retardo (extremo a extremo con tamaño fijo de paquetes) a evaluar viene dado en Ecuación 2.5:

$$Retardo = \frac{d}{V_p} + n * \frac{p + H}{c} + n * T_n + \frac{m}{p} * \frac{p + H}{c}$$

Ecuación 2.5 Expresión del retardo extremo a extremo con tamaño fijo de paquetes

Donde:

- \$d\$ = Distancia entre el nodo origen y el nodo destino (metros - kilómetros)
- \$V_p\$ = Velocidad de propagación (m/s)
- \$n\$ = Número de nodos intermedios
- \$p\$ = Tamaño del paquete sin la cabecera (bits)
- \$H\$ = Tamaño de la cabecera (bits)
- \$c\$ = Velocidad de transmisión de cada enlace (bps - kbps)
- \$T_n\$ = Tiempo de procesamiento en cada nodo (ms)
- \$m\$ = Tamaño del mensaje (bits)

2.3.8 COMPARATIVA DE DATOS ENVIADOS Y RECIBIDOS

Para este estudio la obtención de estos valores se los obtuvo de la siguiente manera:

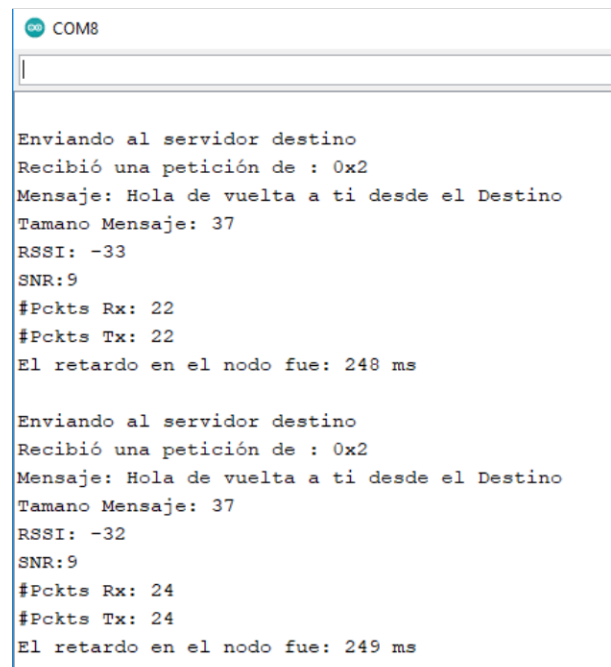
Las distancias como se visualizó con anterioridad en la tabla 2.5, la velocidad de propagación en el aire es muy cercana a la velocidad de la luz en el vacío según [37], tanto el tamaño de la cabecera, el tamaño del mensaje, como el valor del tamaño del paquete sin cabecera se obtuvieron de las comparativas entre las Figuras 2.12 – 2.14 y Figuras 2.13 – 2.15 respectivamente.

```
// Envío de respuesta al Nodo Origen
uint8_t data[] = "Hola de vuelta a ti desde el Destino";
if (manager.sendtoWait(data, sizeof(data), from) != RH_ROUTER_ERROR_NONE)
    Serial.println("sendtoWait failed");
```

Figura 2.12 Payload del Destino

```
Serial.println("Enviando al servidor destino");
uint8_t data[] = "Hola vengo desde el cliente!";
//DEBO PONER LA DIRECCION DESTINO
if (manager.sendtoWait(data, sizeof(data), SERVER1_ADDRESS) != RH_ROUTER_ERROR_NONE)
    Serial.println("sendtoWait failed");
```

Figura 2.13 Payload del Origen

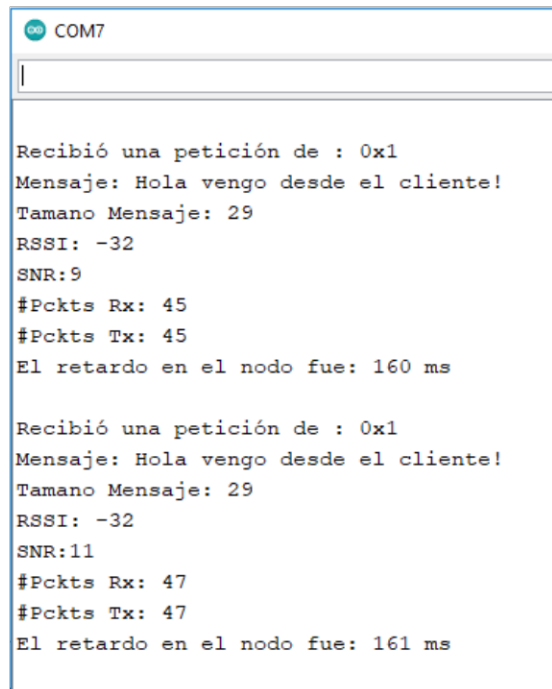


```
COM8

Enviando al servidor destino
Recibió una petición de : 0x2
Mensaje: Hola de vuelta a ti desde el Destino
Tamano Mensaje: 37
RSSI: -33
SNR:9
#Pckts Rx: 22
#Pckts Tx: 22
El retardo en el nodo fue: 248 ms

Enviando al servidor destino
Recibió una petición de : 0x2
Mensaje: Hola de vuelta a ti desde el Destino
Tamano Mensaje: 37
RSSI: -32
SNR:9
#Pckts Rx: 24
#Pckts Tx: 24
El retardo en el nodo fue: 249 ms
```

Figura 2.14 Mensaje recibido en el nodo origen



```
COM7

Recibió una petición de : 0x1
Mensaje: Hola vengo desde el cliente!
Tamano Mensaje: 29
RSSI: -32
SNR:9
#Pckts Rx: 45
#Pckts Tx: 45
El retardo en el nodo fue: 160 ms

Recibió una petición de : 0x1
Mensaje: Hola vengo desde el cliente!
Tamano Mensaje: 29
RSSI: -32
SNR:11
#Pckts Rx: 47
#Pckts Tx: 47
El retardo en el nodo fue: 161 ms
```

Figura 2.15 Mensaje recibido en el nodo destino

De las Figuras 2.14 y 2.16 también se obtuvo el valor del retardo aproximado producido en cada nodo debido al tiempo de procesamiento (T_n), utilizando la función millis en la IDE de Arduino para restar el tiempo adquirido al final del lazo Loop del tiempo obtenido al inicio de este. De manera adicional se empleó el software desarrollado por Semtech para la adquisición de algunos indicadores importantes en la comunicación como la Performance RF y de tiempo, al igual que el consumo de energía que se produciría en cada nodo. Estos factores se presentan en Figura 2.16

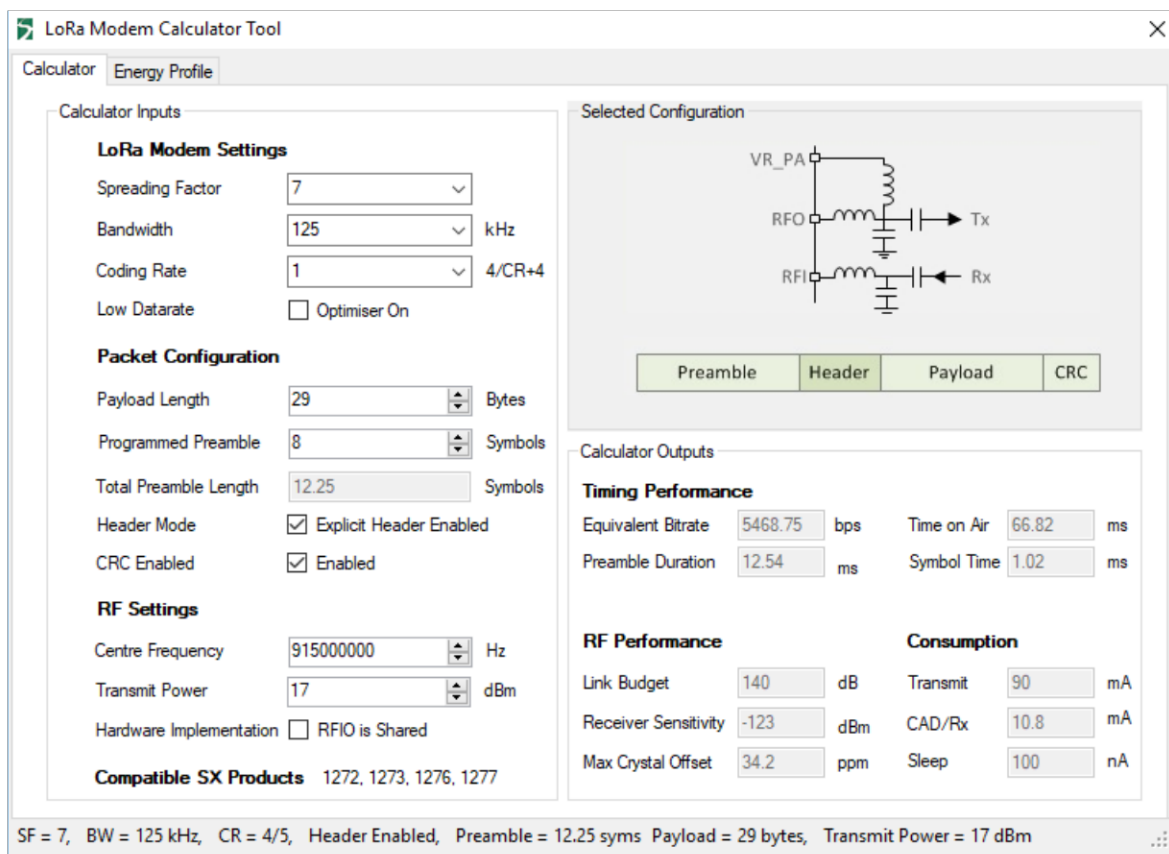


Figura 2.16 Indicadores de la performance del nodo

Los valores de cada variable para el cálculo del retardo end-end multisalto se pueden apreciar en la Tabla 2.10

Tabla 2.10 Valores de los parámetros empleados para la obtención del retardo multisalto

d1, d2, d3 (m)	Vp (ms)	n	p (bits)	H (bits)	c (kbps)	Tn (ms)	m (bits)
10	299792458	1 - 2	224	8	5.4	160 – 161 248 - 249	232- 296
20							
40							
80							
150							
250							

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

En este capítulo se dará cumplimiento los objetivos finales del presente trabajo, los cuales consisten en evaluar como los niveles de atenuación e interferencia en ambientes urbanos y rurales influyen al sistema planteado. Para ellos se presentan mediciones y comparaciones del RSSI y SNR obtenidos para los diferentes ambientes expuestos en la sección 3.1. Por último, se presenta una expresión matemática que me permita para evaluar y predecir el retardo para topologías multisalto usando tecnología LoRa.

3.1 ESCENARIO DE PRUEBAS

Para ofrecer una generalidad a la evaluación del retardo en redes LoRa multisalto con topología lineal, se seleccionaron dos escenarios en zonas: rural y urbana en la ciudad de Quito, Ecuador. Esto con el fin de evaluar el comportamiento de la comunicación con el cambio de factores físicos.

3.1.1 ENTORNO URBANO – Av. GRAL. IGNACIO DE VEINTIMILLA

La Figura 3.1 muestra el tramo escogido de la Avenida General Ignacio de Veintimilla ubicada en el Centro-Norte de la ciudad de Quito, dicho tramo está delimitado por la Av. Amazonas (Origen) y la Av. 12 de octubre. Este trecho, fue elegido ya que no exhibe una gran curvatura y por su cercanía a la Escuela Politécnica Nacional.

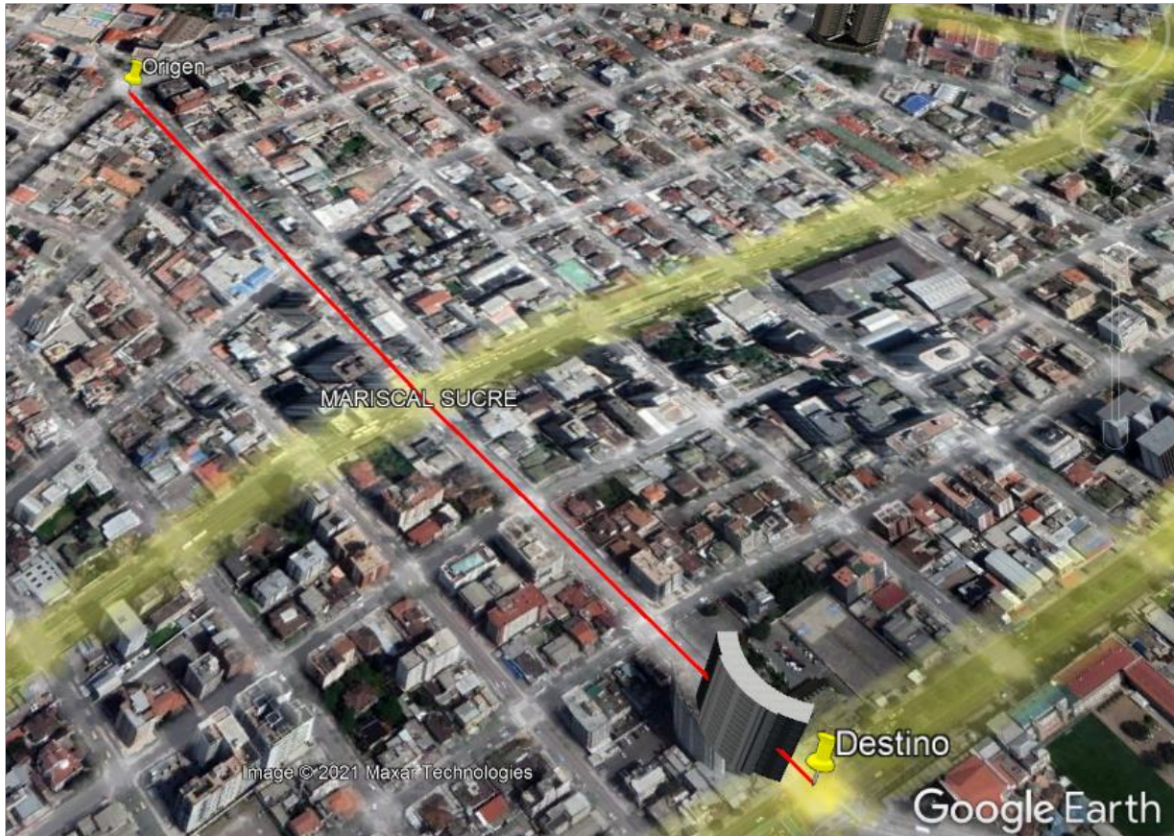


Figura 3.1 Entorno Urbano - Av. Veintimilla

Topografía

Uno de los factores más relevantes en la propagación de una señal es la topografía, debido a que esta repercute claramente en la potencia recibida. La Figura 3.2 muestra el perfil de elevación obtenido en Google Earth para este tramo.



Figura 3.2 Entorno Urbano - Perfil de Elevación

3.1.2 ENTORNO RURAL – PARQUE BICENTENARIO

Este entorno se simuló en este parque, debido a la accesibilidad del lugar al ser un espacio céntrico. El sitio seleccionado, que se puede visualizar en Figura 3.3 y en Figura 3.5 ya que para este entorno fue en dos secciones.

La sección A o Ambiente Rural 1 en medio de la pista principal del parque, no presenta obstáculos estructurales a considerar, solo personas ya sea a pie o en bicicleta. La sección B o Ambiente Rural 1 se localiza a pocos metros de la sección A, pero a diferencia de esta se apreciaron en esencia dos clases de flora sembrada, ambas con una altura no mayor a dos metros.

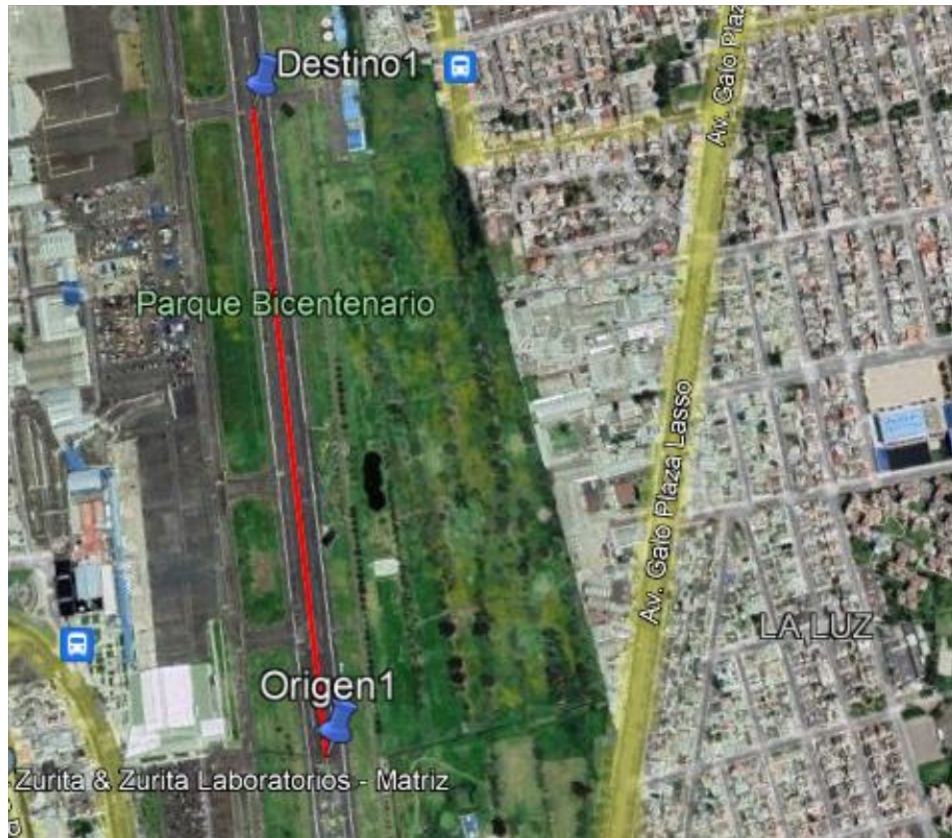


Figura 3.3 Simulación Entorno Rural - Parque Bicentenario Sección A

Topografía

En Figura 3.4, se denota que la topografía de esta sección del parque Bicentenario no presenta grandes declives, por la que se la puede considerar plana (siente metros de altura entre el origen y destino según el software).

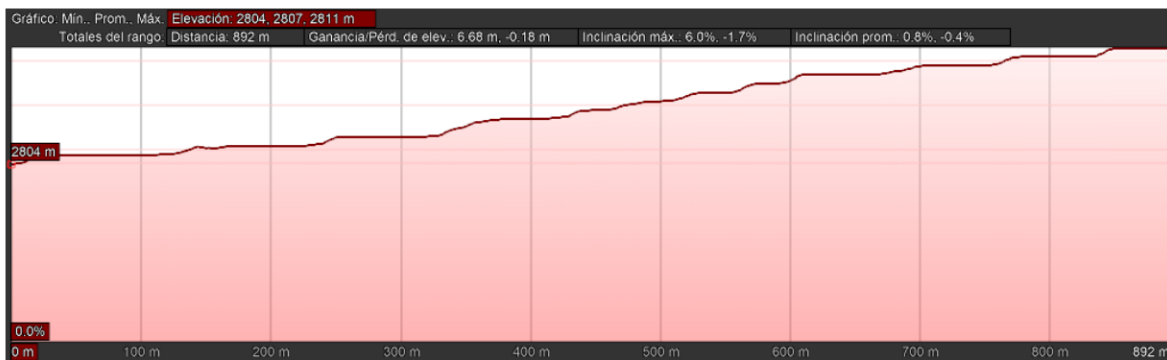


Figura 3.4 Simulación Entorno Rural - Perfil de Elevación Sección A

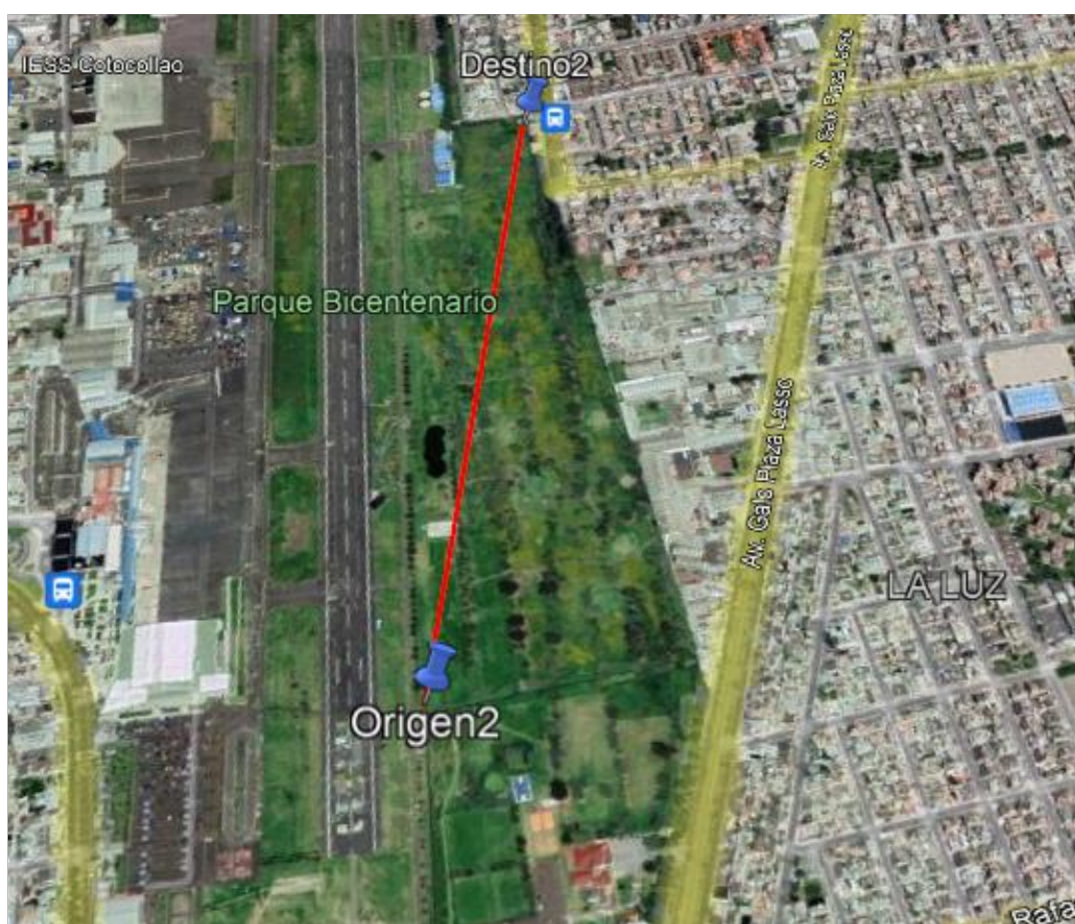


Figura 3.5 Simulación Entorno Rural - Parque Bicentenario Sección B

Topografía

En Figura 3.6, la topografía de esta sección del parque Bicentenario sin bien cuenta picos (principalmente por montículos pequeños de tierra) no presenta declives exorbitantes (siente metros de altura entre el origen y destino al igual que la sección anterior).



Figura 3.6 Simulación Entorno Rural - Perfil de Elevación Sección B

3.2 RESULTADOS

3.2.1 MEDICIONES – AMBIENTE RURAL 1

Para la obtención de las mediciones se realizó la variación de distancias como se especificó en la Tabla 2.5, donde la línea de vista entre el nodo origen, repetidor(es) y nodo destino estaban completamente libre y se encontraban lo más cerca posible de la misma altura.

En Figura 3.1 se puede observar que a partir de una distancia de ~10 metros vemos una clara diferencia en el RSSI entre tener un salto adicional para la comunicación, pero no existe una diferencia en cuanto a la SNR. A medida que aumenta la distancia, los valores de RSSI y SNR disminuyen a un ritmo rápido al principio que después de la primera medición.

Esto se ajusta a la teoría aprendida a lo largo de la carrera, ya que el valor de la RSSI disminuiría rápidamente, pero la cobertura sería todavía transitable, ya que el ruido de la señal era mínimo y no había obstrucciones. En este escenario la tecnología LoRa es muy adecuada y la limitación viene dada principalmente por la propia especificación de LoRa, es decir, el equilibrio entre el alcance y el ancho de banda.

Los retardos no varían en demasía entre los obtenidos en las pruebas y los calculados con la Ecuación 2.5. En la Tabla 3.1 y 3.2, se muestra un resumen de los datos sobre los cuales se trabajó.

La línea punteada de color azul identifica el valor del “Parámetro” vs Distancia con un solo repetidor entre el Nodo Origen y Nodo Destino, a su vez, la roja representa el valor de “Parámetro” vs Distancia usando dos repetidores entre el Nodo Origen y Nodo Destino.

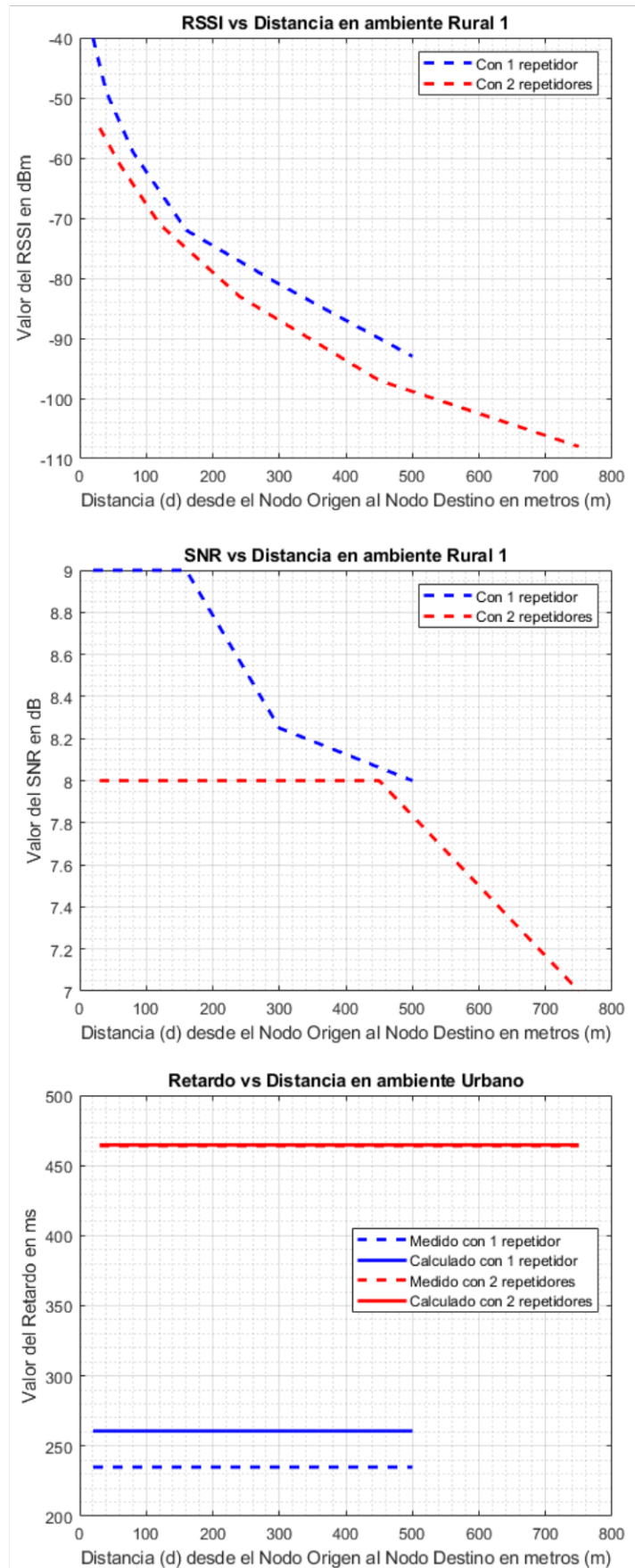


Figura 3.1 RSSI vs Distancia para el Ambiente Rural 1

Tabla 3.1 Resumen de los valores obtenidos en el Ambiente Rural 1 con un nodo repetidor

Distancia entre nodos 1_2_3	Distancia Total	RSSI (dBm)	SNR (dB)	Retardo Obtenido (ms)	Retardo Calculado (ms)
10	20	-40	9	235	260.7354498
20	40	-49	9	235	260.7354499
40	80	-59	9	235	260.73545
80	160	-72	9	235	260.7354503
150	300	-81	8.25	235	260.7354507
250	500	-93	8	235	260.7354514

Tabla 3.2 Resumen de los valores obtenidos en el Ambiente Rural 1 con dos nodos repetidores

Distancia entre nodos 1_2_3_4	Distancia Total	RSSI (dBm)	SNR (dB)	Retardo Obtenido (ms)	Retardo Calculado (ms)
10	30	-55	8	464	464.6984128
20	60	-61	8	464	464.6984129
40	120	-71	8	464	464.6984131
80	240	-83	8	464	464.6984135
150	450	-97	8	464	464.6984142
250	750	-108	7	464	464.6984152

3.2.2 MEDICIONES – AMBIENTE RURAL 2

En este experimento se preparó de forma similar al anterior, en el sentido de que se intentó mantener una diferencia de altura de para los nodos implicados. Como se demuestra en Figura 3.2, podemos ver una disminución drástica que es más severa que la que vimos con la línea de vista sin obstáculos (en teoría). Por otra parte, la SNR comienza a disminuir considerablemente después de 300 metros e incluso alcanza valores negativos para 750 metros. Esto indicaría que este entorno tiene un gran impacto en la SNR y también en el RSSI. El máximo valor de RSSI parece alcanzar los -120 dBm, ya que no se observó ningún paquete con un valor inferior. Así mismo, como en el apartado anterior se muestra en Tablas 3.3 y 3.4 un resumen de los datos para el Ambiente Rural 2

Tabla 3.3 Resumen de los valores obtenidos en el Ambiente Rural 2 con un nodo repetidor

Distancia entre nodos 1_2_3	Distancia Total	RSSI (dBm)	SNR (dB)	Retardo Obtenido (ms)	Retardo Calculado (ms)
10	20	-58	10	278	260.7354498
20	40	-61	9	278	260.7354499
40	80	-73	9	278	260.73545
80	160	-85	8.25	278	260.7354503
150	300	-102	8	278	260.7354507
250	500	-112	5	278	260.7354514

Tabla 3.4 Resumen de los valores obtenidos en el Ambiente Rural 2 con dos nodos repetidores

Distancia entre nodos 1_2_3_4	Distancia Total	RSSI (dBm)	SNR (dB)	Retardo Obtenido (ms)	Retardo Calculado (ms)
10	30	-69	9	493	464.6984128
20	60	-73	9	493	464.6984129
40	120	-85	8.56	493	464.6984131
80	240	-96	6	493	464.6984135
150	450	-113	4	493	464.6984142
250	750	-119	-2	493	464.6984152

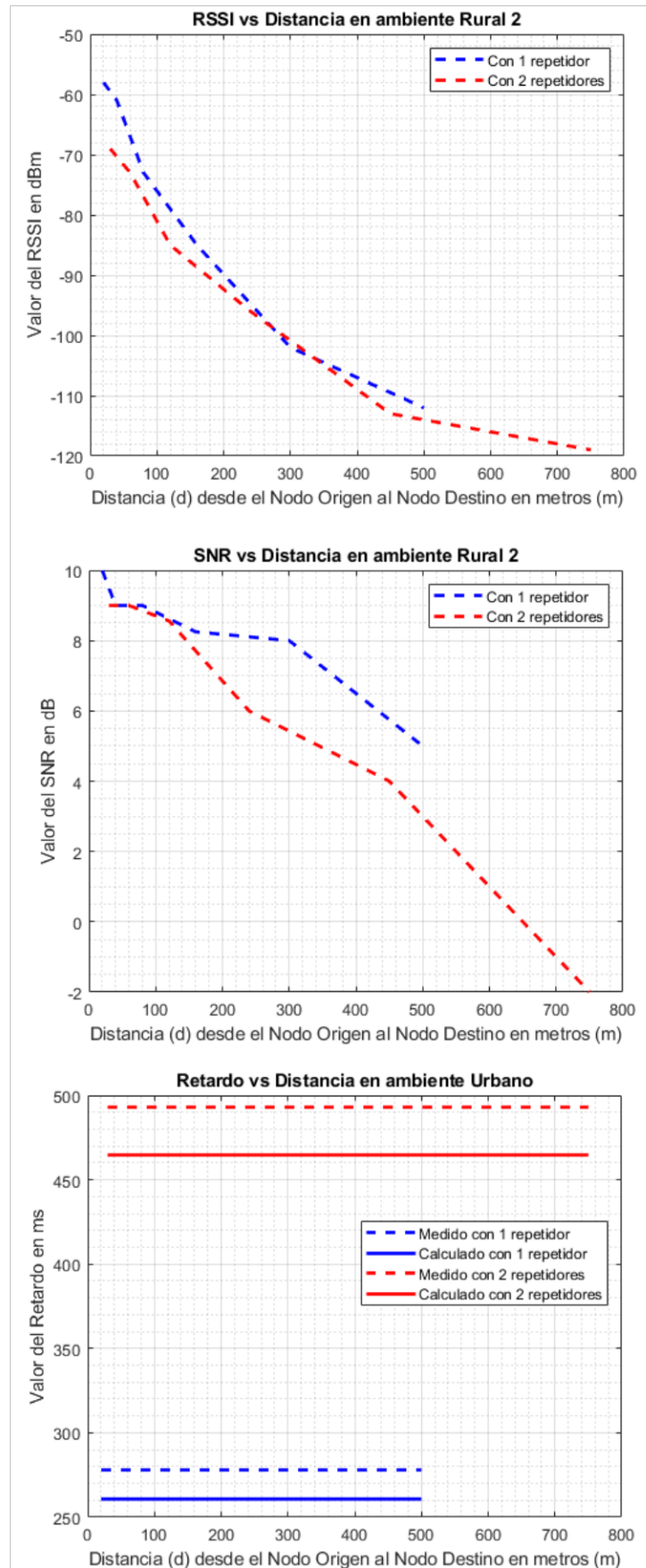


Figura 3.2 RSSI vs Distancia para el Ambiente Rural 2

3.2.3 MEDICIONES – AMBIENTE URBANO

En Figura 3.3 se aprecia una caída más drástica de la RSSI y la SNR que en los experimentos anteriores y se puede observar que los valores son más bajos a 200 metros que los de los experimentos anteriores a 400 metros. En Tablas 3.5 y 3.6, se muestra un resumen de los datos para el Ambiente Urbano.

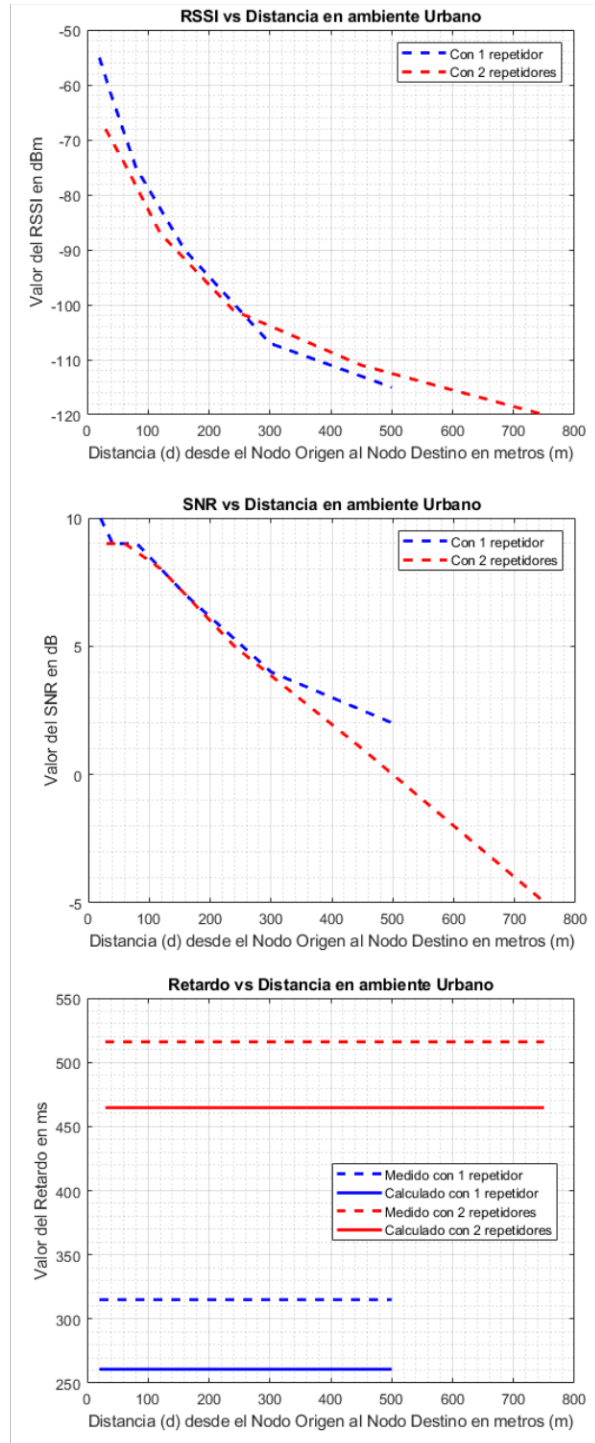


Figura 3.3 RSSI vs Distancia para el Ambiente Urbano

Tabla 3.5 Resumen de los valores obtenidos en el Ambiente Urbano con un nodo repetidor

Distancia entre nodos 1_2_3	Distancia Total	RSSI (dBm)	SNR (dB)	Retardo Obtenido (ms)	Retardo Calculado (ms)
10	20	-55	10	315	260.7354498
20	40	-62	9	315	260.7354499
40	80	-75	9	315	260.73545
80	160	-90	7	315	260.7354503
150	300	-107	4	315	260.7354507
250	500	-115	2	315	260.7354514

Tabla 3.6 Resumen de los valores obtenidos en el Ambiente Urbano con dos nodos repetidores

Distancia entre nodos 1_2_3_4	Distancia Total	RSSI (dBm)	SNR (dB)	Retardo Obtenido (ms)	Retardo Calculado (ms)
10	30	-68	9	516	464.6984128
20	60	-74	9	516	464.6984129
40	120	-87	8	516	464.6984131
80	240	-101	5	516	464.6984135
150	450	-111	1	516	464.6984142
250	750	-120	-5	516	464.6984152

Cabe mencionar que los datos utilizados para realizar las gráficas y tablas en este apartado, se extrajeron de los archivos de texto (.txt) mencionados en la sección 2.5.6, con muestras mayores 700 paquetes enviados y recibidos durante las pruebas.

3.2.4 EXPRESIÓN MATEMÁTICA DEL RETARDO MULTISALTO

Para determinar si la ecuación 2.5 que evalúa el retardo en una comunicación extremo a extremo con tamaño fijo de paquetes es válida para este modelo, se realizó cálculos de varianza, desviación estándar y errores relativos – absolutos.

Para una muestra la varianza viene dada en Ecuación 3.1

$$S^2 = \frac{\sum(x - \bar{X})^2}{n - 1}$$

Ecuación 3.1 Varianza para una muestra

La media de la muestra se señala en la ecuación 3.2

$$\bar{X} = \frac{\sum x}{n}$$

Ecuación 3.1 Media para una muestra

El error absoluto y error relativo están representadas en Ecuaciones 3.3 y 3.4, respectivamente.

$$Ea = |V_{REAL} - V_{MEDIDO}|$$

Ecuación 3.3 Error Absoluto

$$Er = \frac{Ea}{V_{REAL}}$$

Ecuación 3.4 Error Relativo

3.2.4.1 Desviación Estándar y errores del retardo para un nodo repetidor

$$\begin{aligned} \bar{X} &= \frac{\sum x}{n} \\ &= \frac{260.7354498 + 260.7354499 + 260.73545 + 260.7354503 + 260.7354507 + 260.7354514}{6} \\ &= 260.7354503 \end{aligned}$$

$$S^2 = \frac{\sum(x - \bar{X})^2}{n - 1} = 1.83852 * 10^{-13} \text{ ms}^2$$

$$S = 4.287786 * 10^{-7} \text{ ms}$$

$$Ea \text{ Ambiente Rural 1} = |235 - 260.7354503| = 25.7354503$$

$$Ea \text{ Ambiente Rural 2} = |278 - 260.7354503| = 17.2645497$$

$$Ea \text{ Ambiente Urbano} = |315 - 260.7354503| = 57.2645497$$

$$Er \text{ Ambiente Rural 1} = \frac{25.7354503}{235} = 10.95\%$$

$$Er \text{ Ambiente Rural 2} = \frac{17.2645497}{278} = 6.21\%$$

$$Er \text{ Ambiente Urbano} = \frac{57.2645497}{315} = 18.18\%$$

3.2.4.2 Desviación Estándar del retardo para dos nodos repetidores

$$\begin{aligned}\bar{X} &= \frac{\sum x}{n} \\ &= \frac{464.6984128 + 464.6984129 + 464.6984131 + 464.6984135 + 464.6984142 + 464.6984152}{6} \\ &= 464.6984136\end{aligned}$$

$$S^2 = \frac{\sum(x - \bar{X})^2}{n - 1} = 8.61667 * 10^{-13} \text{ ms}^2$$

$$S = 9.2826 * 10^{-7} \text{ ms}$$

$$Ea \text{ Ambiente Rural 1} = |464 - 464.6984136| = 0.694136$$

$$Ea \text{ Ambiente Rural 2} = |493 - 464.6984136| = 28.3$$

$$Ea \text{ Ambiente Urbano} = |516 - 464.6984136| = 51.3$$

$$Er \text{ Ambiente Rural 1} = \frac{0.694136}{464} = 0.14\%$$

$$Er \text{ Ambiente Rural 2} = \frac{28.3}{493} = 5.74\%$$

$$Er \text{ Ambiente Urbano} = \frac{51.3}{516} = 9.94\%$$

3.3 CONCLUSIONES

El objetivo principal de este trabajo era hacer una evaluación del retardo y rendimiento de una red LoRa Multisalto para una topología lineal. A partir de esta evaluación, analizar cómo la limitación establecida por estos factores afecta a varios casos de uso.

Se pueden observar varias restricciones cuando hay obstrucciones en el camino ya que la señal cae rápidamente hasta los 300 metros, después de lo cual la tasa de caída de la señal se reduce y, en función del entorno, puede llegar a detenerse, como en el caso del experimento en el ambiente rural 1, donde la señal mantiene una buena calidad constante durante toda la prueba. En las pruebas en el ambiente rural 2 y en ambiente urbano la calidad es significativamente peor hasta el punto de que con la configuración de los nodos, en un entorno urbano alcanzar distancias superiores a 400 m es imposible debido a la gran cantidad de ruido y materiales densos existentes. Este problema se puede sortear elevando los nodos hasta el punto de conseguir una línea de vista libre o cercana a ella.

Esto significa que las implementaciones a gran escala que despliegan nodos se tendrían que colocar en torres de radio o terrazas de edificio para evitar obstrucciones, con ello no se obtendrían problemas para alcanzar distancias de +5 km. En el caso de implementaciones a menor escala, pueden surgir problemas ya que las barreras en forma de casas, terreno y/o vegetación, ruido ambiental se suman rápidamente y, en la mayoría de los casos, limitan el alcance de las implementaciones a nivel de ciudad o selva. En algunos casos, elevar la puerta de enlace a una altura óptima podría ser imposible. Otro factor es si el dispositivo es móvil, ya que tener las antenas omnidireccionales deja una gran huella en el rendimiento ya que no hay una focalización en la señal.

En cuanto a los retardos los valores en los errores no son tan significativos y podemos intuir que, incrementando el número de saltos, es decir, en número de repetidores entre los nodos origen y destino la ecuación empleada para evaluarlos se amolda a la implementación.

Al superar el umbral RSSI de -120dBm hace que el paquete se corrompa, es decir, que sea ilegible o se pierda por completo. Sin embargo, la SNR parece tener menos efecto, ya que un paquete es aparentemente tan legible a ~ -5 dB como a un valor positivo. Esto significa que para la mayoría de las implementaciones es muy importante tener en cuenta la pérdida de paquetes al planificar la colocación, ya que el RSSI en este estudio se incrementó según se aumentaba los repetidores y distancia, como en la mayoría de los

casos no hay retransmisiones en LoRa si el dispositivo está en el límite del rango de comunicación, el paquete puede perderse para siempre.

3.4 RECOMENDACIONES

Se recomienda realizar las mediciones desde alturas considerables con el fin de tener una línea de vista idónea, así como también, conectar el módulo LoRa SX1276 a una antena comercial a la frecuencia de trabajo con el propósito de tener la mejor eficiencia posible.

No todos los parámetros de configuración para el transceptor modelo HPD13A V1.1 fueron aceptadas, así que se recomienda realizar las pruebas pertinentes en un ambiente controlado o laboratorio, y no fiarse tanto de la literatura, antes de realizar las prueba en campo.

Si es necesario se sugiere realizar las pruebas con más nodos intermedios entre el nodo origen y destino, además del envío de datos obtenidos de sensores, ya que el payload se verá afectado y por ende los retrasos transmisión y procesamiento.

Se podría realizar las pruebas con otro tipo de microcontroladores, ya sea un Atmega 328p (Arduino y sus variantes), NodeMCU ESP32, una Blue Pill STM32 e incluso una Raspberry Pi Pico, con el fin de memorar los tiempos de procesamiento en el nodo. Teniendo en cuenta que para cada tarjeta de desarrollo exista una librería que describa las características de LoRa compatible.

REFERENCIAS BIBLIOGRÁFICAS

- [1] «Regional Parameters», *The Things Network*, 27 de abril de 2021. <https://www.thethingsnetwork.org/docs/lorawan/regional-parameters/> (accedido 16 de noviembre de 2021).
- [2] baeldung, «Propagation Delay vs Transmission Delay | Baeldung on Computer Science», 22 de junio de 2021. <https://www.baeldung.com/cs/propagation-vs-transmission-delay> (accedido 15 de diciembre de 2021).
- [3] «Delays in Computer Network», *GeeksforGeeks*, 23 de marzo de 2020. <https://www.geeksforgeeks.org/delays-in-computer-network/> (accedido 21 de enero de 2022).
- [4] «What Is the Internet of Things (IoT)?», *What is IoT*. <https://www.oracle.com/internet-of-things/what-is-iot/> (accedido 16 de noviembre de 2021).
- [5] D. Zorbas, K. Abdelfadeel, P. Kotzanikolaou, y D. Pesch, «TS-LoRa: Time-slotted LoRaWAN for the Industrial Internet of Things», *Computer Communications*, vol. 153, pp. 1-10, mar. 2020, doi: 10.1016/j.comcom.2020.01.056.
- [6] K. Mekki, E. Bajic, F. Chaxel, y F. Meyer, «A comparative study of LPWAN technologies for large-scale IoT deployment», *ICT Express*, vol. 5, n.º 1, pp. 1-7, mar. 2019, doi: 10.1016/j.icte.2017.12.005.
- [7] «LPWA: definición y detalles». <https://www.paessler.com/es/it-explained/lpwa> (accedido 21 de enero de 2022).
- [8] Olatinwo, A. Abu-Mahfouz, y Hancke, «A Survey on LPWAN Technologies in WBAN for Remote Health-Care Monitoring», *Sensors*, vol. 19, p. 5268, nov. 2019, doi: 10.3390/s19235268.
- [9] M. Farooq y D. Pesch, «Analyzing LoRa: a Use Case Perspective», feb. 2018. doi: 10.1109/WF-IoT.2018.8355224.
- [10] «All About LoRa and LoRaWAN: LoRa: Symbol Generation». <http://www.sghosly.com/p/lora-is-chirp-spread-spectrum.html> (accedido 25 de enero de 2022).
- [11] A. Lavric y V. Popa, «Internet of Things and LoRa™ Low-Power Wide-Area Networks: A survey», en *2017 International Symposium on Signals, Circuits and Systems (ISSCS)*, jul. 2017, pp. 1-5. doi: 10.1109/ISSCS.2017.8034915.
- [12] «an1200.22.pdf». Accedido: 15 de diciembre de 2021. [En línea]. Disponible en: <https://www.frugalprototype.com/wp-content/uploads/2016/08/an1200.22.pdf>
- [13] M. Bor y U. Roedig, «LoRa Transmission Parameter Selection», en *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, Ottawa, ON, jun. 2017, pp. 27-34. doi: 10.1109/DCOSS.2017.10.
- [14] M. Bor, J. Vidler, y U. Roedig, «LoRa for the Internet of Things», *EWSN*, pp. 361-366, feb. 2016.
- [15] «What is the Technology Behind LoRa Frequency», *MOKOSmart #1 Smart Device Solution in China*, 12 de marzo de 2020. <https://www.mokosmart.com/lora-frequency/> (accedido 25 de enero de 2022).
- [16] M. Knight y B. Seeber, «Decoding LoRa: Realizing a Modern LPWAN with SDR», *Proceedings of the GNU Radio Conference*, vol. 1, n.º 1, Art. n.º 1, sep. 2016, Accedido: 15 de diciembre de 2021. [En línea]. Disponible en: <https://pubs.gnuradio.org/index.php/grcon/article/view/8>
- [17] T. Bouguera, J.-F. Diouris, J.-J. Chaillout, R. Jaouadi, y A. Guillaume, «Energy consumption model for sensor nodes based on LoRa and LoRaWAN», *Sensors*, vol. 18, p. 2104, jun. 2018, doi: 10.3390/s18072104.
- [18] «Modulación LoRa: Long Range Modulation | by edgar solera | Exploración de aplicaciones usando tecnología LoRa | Medium». <https://medium.com/pruebas-de-laboratorio-de-la->

- modulaci%C3%B3n-lora/modulaci%C3%B3n-lora-4ad74cabd59e (accedido 15 de diciembre de 2021).
- [19] «What is LoRaWAN® Specification - LoRa Alliance®». <https://loro-alliance.org/about-lorawan/> (accedido 15 de diciembre de 2021).
- [20] «Tecnología LoRA y LoRAWAN - Catsensors». <https://www.catsensors.com/es/lorawan/tecnologia-lora-y-lorawan> (accedido 25 de enero de 2022).
- [21] L. Casals Ibáñez, B. Mir Masnou, R. Vidal Ferré, y C. Gomez, «Modeling the energy performance of LoRaWAN», *Sensors*, vol. 17, p. 2364, oct. 2017, doi: 10.3390/s17102364.
- [22] «(PDF) On the Limits of LoRaWAN Channel Access». https://www.researchgate.net/publication/312485284_On_the_Limits_of_LoRaWAN_Channel_Access (accedido 15 de diciembre de 2021).
- [23] C. E. Acosta, F. Gil-Castiñeira, y E. Costa-Montenegro, «Red inalámbrica de sensores con topología lineal sin capa de red», *Revista de Investigación en Tecnologías de la Información*, vol. 9, n.º 17, Art. n.º 17, ene. 2021, Accedido: 21 de enero de 2022. [En línea]. Disponible en: <https://www.riti.es/ojs2018/inicio/index.php/riti/article/view/274>
- [24] M. Y. Lee, A. S. Azman, S. Subramaniam, y F. Feroz, «Performance Analysis of Linear Topology Wireless Sensor Network in Oil and Gas Industry», *IOP Conference Series: Materials Science and Engineering*, vol. 765, p. 012070, mar. 2020, doi: 10.1088/1757-899X/765/1/012070.
- [25] D. E. Gorla, «Reproducibilidad y replicabilidad en la investigación en ciencias naturales: ¿Hay una crisis?», *Ecología Austral*, vol. 31, n.º 1, Art. n.º 1, mar. 2021, doi: 10.25260/EA.21.31.1.0.1060.
- [26] «SX1276». <https://www.semtech.com/products/wireless-rf/lora-core/sx1276> (accedido 21 de enero de 2022).
- [27] «MODULO TRANSCEIVER LORA RFM95 RFM95W SX1276 868/915MHZ TRANSCEPTOR», *Starware*. <https://tienda.starware.com.ar/producto/modulo-transceiver-lora-rfm95-rfm95w-sx1276-868915mhz-transceptor/> (accedido 25 de enero de 2022).
- [28] «NodeMCU v2 ESP8266 WiFi». <https://naylampmechatronics.com/espressif-esp/153-nodemcu-v2-esp8266-wifi.html> (accedido 21 de enero de 2022).
- [29] «Convertidor Voltaje DC-DC Step-Up 2A MT3608», *Naylamp Mechatronics - Perú*. <https://naylampmechatronics.com/conversores-dc-dc/119-convertidor-voltaje-dc-dc-step-up-2a-mt3608.html> (accedido 21 de enero de 2022).
- [30] «ELEVADOR DC-DC MT3608 – TECHNICAL HANDS SAS». <https://technicalhands.com.co/product/elevador-dc-dc-mt3608/> (accedido 25 de enero de 2022).
- [31] «Módulo Cargador de batería litio TP4056 micro-USB», *Naylamp Mechatronics - Perú*. <https://naylampmechatronics.com/baterias/194-cargador-de-bateria-litio-tp4056-micro-usb.html> (accedido 21 de enero de 2022).
- [32] «Batería 18650 3.7V 2200mAh», *UNIT Electronics*. <https://uelectronics.com/producto/bateria-18650-3-7v-2200mah/> (accedido 21 de enero de 2022).
- [33] «Liion Battery Diagram Vector Illustration Rechargeable: vector de stock (libre de regalías) 686973958». <https://www.shutterstock.com/es/image-vector/liion-battery-diagram-vector-illustration-rechargeable-686973958> (accedido 25 de enero de 2022).
- [34] W. C. Y. Lee, «Estimate of local average power of a mobile radio signal», *IEEE Transactions on Vehicular Technology*, vol. 34, n.º 1, pp. 22-27, feb. 1985, doi: 10.1109/T-VT.1985.24030.
- [35] «RadioHead: RadioHead Packet Radio library for embedded microprocessors». <http://www.airspayce.com/mikem/arduino/RadioHead/> (accedido 21 de enero de 2022).
- [36] «RadioHead: RHReliableDatagram Class Reference». <https://www.airspayce.com/mikem/arduino/RadioHead/classRHReliableDatagram.html> (accedido 21 de enero de 2022).

[37] «Electromagnetic wave - Knowino».

https://www.tau.ac.il/~tsirel/dump/Static/knowino.org/wiki/Electromagnetic_wave.html
(accedido 21 de enero de 2022).

ANEXOS

ANEXO I. Diagrama de Pines del Transceptor SX1276 modelo HPD13A V1.1

ANEXO II. Pinout de modulo ModeMCU ESP8266

ANEXO III. Esquematización y diseños en PCB

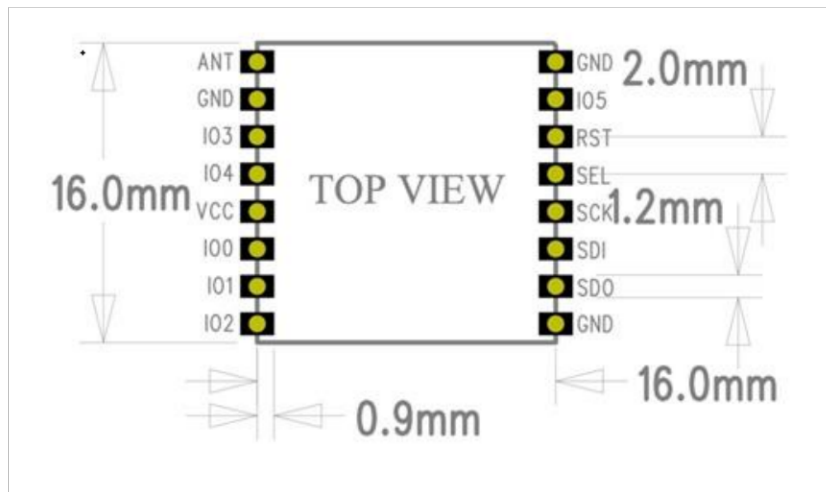
ANEXO IV. Montaje final del nodo

ANEXO V. Código Nodo Origen

ANEXO VI. Código Nodo Repetidor / Destino

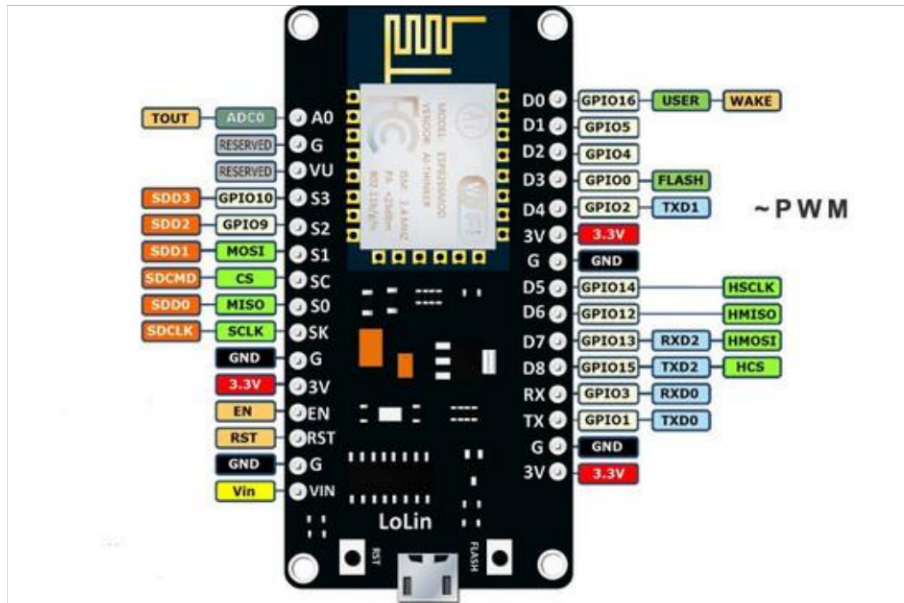
ANEXO I

Diagrama de Pines del Transceptor SX1276 modelo HPD13A V1.1



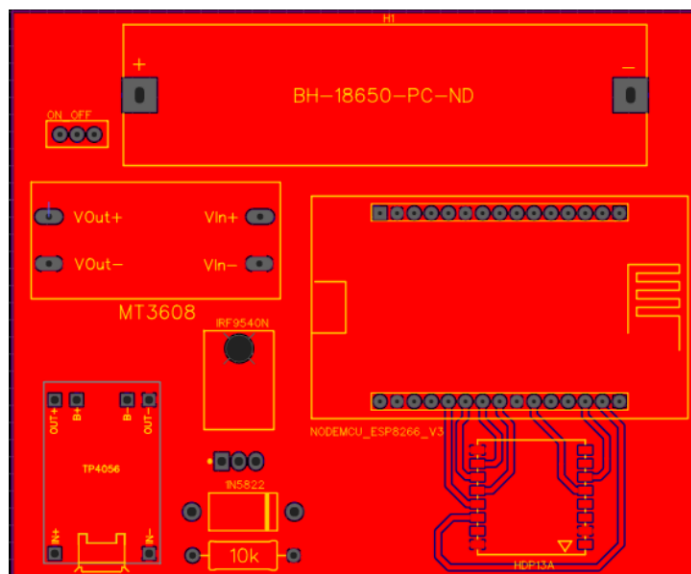
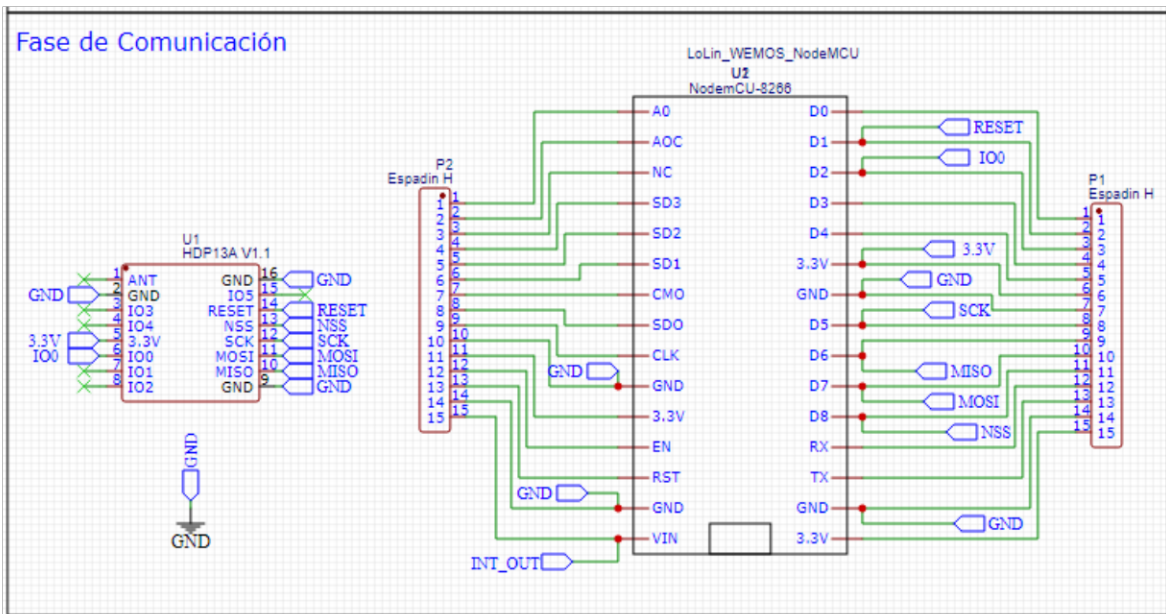
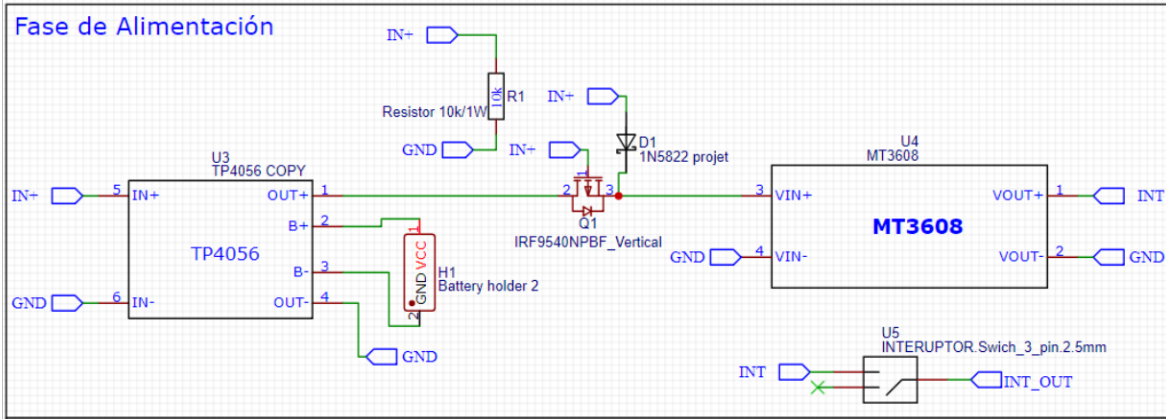
ANEXO II

Pinout de modulo ModeMCU ESP8266



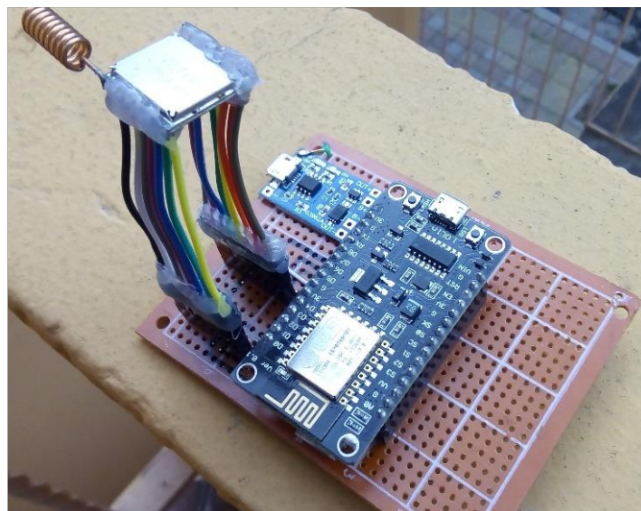
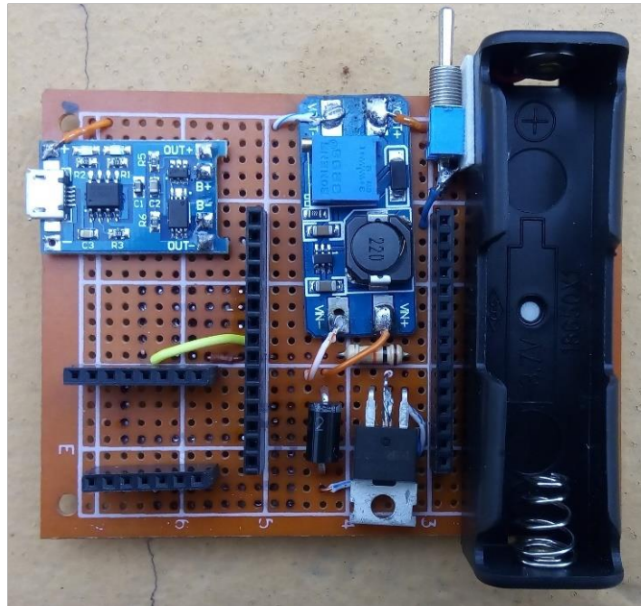
ANEXO III

Esquematzación y diseños en PCB



ANEXO IV

Montaje final del nodo



ANEXO V

Código Nodo Origen

```
#include <RHRouter.h>
#include <RH_RF95.h>
#include <SPI.h>

unsigned long previo,actual;

#define ORIGEN_ADDRESS 1
#define REPET1_ADDRESS 2
#define REPET2_ADDRESS 3
#define DESTIN_ADDRESS 4

// CONFIG FIJA
#define RADIO_FREQUENCY 915.0
#define RADIO_TX_POWER 17
#define RADIO_PREAMBLELENGTH 8

// CONFIG Variable

// Defaut, medio rango
#define RADIO_SIGNALBANDWIDTH 125000
#define RADIO_SPREADFACTOR 7
#define RADIO_CODINGRATE 5
/*
/*
// Rango -> Veloz y Corta Distancia
#define RADIO_SIGNALBANDWIDTH 500000
#define RADIO_SPREADFACTOR 7
#define RADIO_CODINGRATE 5
*/
/*
// Rango -> Lento y Larga Distancia
#define RADIO_SIGNALBANDWIDTH 125000
#define RADIO_SPREADFACTOR 12
#define RADIO_CODINGRATE 8
*/

// Instancia Singleton para el controlador de radio (para NodeMCU ESP8266 en este caso)
#define RFM95_CS 15
#define RFM95_RST 5
#define RFM95_INT 4

RH_RF95 rf95(RFM95_CS, RFM95_INT);

// DEFINO LA DIRECCION DEL NODO (nodo origen para este caso)
RHRouter manager(rf95, ORIGEN_ADDRESS);
```

```

void setup()
{
  Serial.begin(115200);
  if (!manager.init())
    Serial.println("init failed");
  // Configuración con los parametros de comunicación
  rf95.setFrequency(RADIO_FREQUENCY);
  rf95.setTxPower(RADIO_TX_POWER);
  rf95.setSignalBandwidth(RADIO_SIGNALBANDWIDTH);
  rf95.setSpreadingFactor(RADIO_SPREADFACTOR);
  rf95.setPreambleLength(RADIO_PREAMBLELENGTH);
  rf95.setCodingRate4(RADIO_CODINGRATE);

  // ASIGNO LOS SALTOS PARA CADA NODO (PARA NODO ORIGEN)
  manager.addRouteTo(REPET1_ADDRESS, REPET1_ADDRESS);
  manager.addRouteTo(REPET2_ADDRESS, REPET1_ADDRESS);
  manager.addRouteTo(DESTIN_ADDRESS, REPET1_ADDRESS);
}

void loop()
{
  while(1)
  {
    previo = millis();
    Serial.println("Enviando al servidor destino");
    uint8_t data[] = "Hola vengo desde el origen!!";
    //DEBO PONER LA DIRECCION DESTINO
    if (manager.sendtoWait(data, sizeof(data), DESTIN_ADDRESS) != RH_ROUTER_ERROR_NONE)
      Serial.println("sendtoWait failed");
    else
    {
      uint8_t buf[RH_ROUTER_MAX_MESSAGE_LEN];
      uint8_t len = sizeof(buf);
      uint8_t from;
      if (manager.recvfromAckTimeout(buf, &len, 3000, &from))
      {
        Serial.print("Recibió una petición de : 0x");
        Serial.println(from, HEX);
        Serial.print("Mensaje: ");
        Serial.println((char*)buf);
        Serial.print("Tamano Mensaje: ");
        Serial.println(len);
        Serial.print("RSSI: ");
        Serial.println(rf95.lastRssi(), DEC);
        Serial.print("SNR:");
        Serial.println(rf95.lastSNR(), DEC);
        Serial.print("#Pckts Rx: ");
        Serial.println(rf95.rxGood(), DEC);
        Serial.print("#Pckts Tx: ");
        Serial.println(rf95.txGood(), DEC);

      }
    }
    else
    {
      Serial.println("No hay respuestas de los otros nodos");
    }
  }
}

```

```
actual = millis();

Serial.print("El retardo en el nodo fue: ");
Serial. print(actual-previo);
Serial.println(" ms");
Serial.println(" ");

}

}
```

ANEXO VI

Código Nodo Repetidor / Destino

```
#include <RHRouter.h>
#include <RH_RF95.h>
#include <SPI.h>

unsigned long previo,actual;

#define ORIGEN_ADDRESS 1
#define REPET1_ADDRESS 2
#define REPET2_ADDRESS 3
#define DESTIN_ADDRESS 4

// CONFIG FIJA
#define RADIO_FREQUENCY 915.0
#define RADIO_TX_POWER 17
#define RADIO_PREAMBLELENGTH 8

// CONFIG Variable

// Defaut, medio rango
#define RADIO_SIGNALBANDWIDTH 125000
#define RADIO_SPREADFACTOR 7
#define RADIO_CODINGRATE 5
// DIRECCION DEL NODO
const uint8_t MI_DIRECCION = DESTIN_ADDRESS;
/*
/*
// Rango -> Veloz y Corta Distancia
#define RADIO_SIGNALBANDWIDTH 500000
#define RADIO_SPREADFACTOR 7
#define RADIO_CODINGRATE 5
*/
/*
// Rango -> Lento y Larga Distancia
#define RADIO_SIGNALBANDWIDTH 125000
#define RADIO_SPREADFACTOR 12
#define RADIO_CODINGRATE 8
*/

// Instancia Singleton para el controlador de radio (para NodeMCU ESP8266 en este caso)
#define RFM95_CS 15
#define RFM95_RST 5
#define RFM95_INT 4

RH_RF95 rf95(RFM95_CS, RFM95_INT);

// DEFINO LA DIRECCION DEL NODO (nodo destino para este caso)
RHRouter manager(rf95, MI_DIRECCION);
```

```

void setup()
{
    Serial.begin(115200);
    if (!manager.init())
        Serial.println("init failed");
    // Configuración con los parametros de comunicación
    rf95.setFrequency(RADIO_FREQUENCY);
    rf95.setTxPower(RADIO_TX_POWER);
    rf95.setSignalBandwidth(RADIO_SIGNALBANDWIDTH);
    rf95.setSpreadingFactor(RADIO_SPREADFACTOR);
    rf95.setPreambleLength(RADIO_PREAMBLELENGTH);
    rf95.setCodingRate4(RADIO_CODINGRATE);

    // ASIGNO LOS SALTOS PARA CADA NODO
    switch (MI_DIRECCION)
    {
        case REPET1_ADDRESS:
            manager.addRouteTo (ORIGEN_ADDRESS, ORIGEN_ADDRESS);
            manager.addRouteTo (REPET2_ADDRESS, REPET2_ADDRESS);
            manager.addRouteTo (DESTIN_ADDRESS, REPET2_ADDRESS);
            break;
        case REPET2_ADDRESS:
            manager.addRouteTo (ORIGEN_ADDRESS, REPET1_ADDRESS);
            manager.addRouteTo (REPET1_ADDRESS, REPET1_ADDRESS);
            manager.addRouteTo (DESTIN_ADDRESS, DESTIN_ADDRESS);
            break;
        case DESTIN_ADDRESS:
            manager.addRouteTo (ORIGEN_ADDRESS, REPET2_ADDRESS);
            manager.addRouteTo (REPET1_ADDRESS, REPET2_ADDRESS);
            manager.addRouteTo (REPET2_ADDRESS, REPET2_ADDRESS);
            break;
        default:
            Serial.println ("No hay definida una dirección");
            exit (EXIT_FAILURE);
            break;
    }
    //manager.printRoutingTable(Serial);

    Serial.println ("Esperando la llegada de mensajes...");
}

void loop()
{
    previo = millis();
    if (rf95.available())
    {
        uint8_t buf[RH_ROUTER_MAX_MESSAGE_LEN];
        uint8_t len = sizeof(buf);
        uint8_t from;
    }
}

```

```

if (manager.recvfromAck(buf, &len, &from))
{
    Serial.print("Recibió una petición de : 0x");
    Serial.println(from, HEX);
    Serial.print("Mensaje: ");
    Serial.println((char*)buf);
    Serial.print("Tamano Mensaje: ");
    Serial.println(len);
    Serial.print("RSSI: ");
    Serial.println(rf95.lastRssi(), DEC);
    Serial.print("SNR:");
    Serial.println(rf95.lastSNR(), DEC);
    Serial.print("#Pckts Rx: ");
    Serial.println(rf95.rxGood(), DEC);
    Serial.print("#Pckts Tx: ");
    Serial.println(rf95.txGood(), DEC);

    // Envío de respuesta al Nodo Origen

    uint8_t data[] = "Hola de vuelta a ti desde el Destino"; //Este mensaje varia segun el nodo
    if (manager.sendtoWait(data, sizeof(data), from) != RH_ROUTER_ERROR_NONE)
        Serial.println("sendtoWait failed");
}
    actual = millis();

Serial.print("El retardo en el nodo fue: ");
Serial.print(actual-previo);
Serial.println(" ms");
Serial.println(" ");
}
}

```