

UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE INGENIERIA
MAESTRIA EN INGENIERIA DE REDES DE COMUNICACION



TESIS DE MAESTRÍA

**Diseño de un sistema de comunicación basado en
“Internet of Things” para la generación de mapas
de cosecha para la agroindustria.**

Postulante: Ing. Rene Fernando Sosa Santos

Tutor: Msc. Ing. Edgar del Carpio

La Paz – Bolivia

2021



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE INGENIERIA**



LA FACULTAD DE INGENIERIA DE LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) Visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) Copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) Copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la cita o referencia correspondiente en apego a las normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADAS EN LA LEY DE DERECHOS DE AUTOR.

Agradecimientos

Durante el desarrollo de la etapa de observación y análisis del proyecto, fue necesario la tutoría del magister ingeniero Edgar del Carpio, ya que desde la etapa de los módulos respectivos de maestría fue guiándome en tópicos selectos de tesis. Mas aun de los consejos del director del instituto de electrónica aplicada Wilber Flores centralizados en conocimientos integrales entre software y hardware y a la secretaria conocida como doña Mauge y el asistente de laboratorio y ahora gran amigo mío Paul Cori. Además, es necesario agradecer a todos mis compañeros de clases, pero en especial a Huber Machicado y a Gabriel Mamani “el cotelito”. Y por sobre todo a mi familia Norah Santos y Rene Sosa Sr, ya que ellos me dieron la venia para seguir adelante con una maestría y a la licenciada Valeria Crespo que al finalizar los módulos y dicho documento fue un gran apoyo tanto emocional como metodológico. Más aun a las ingenieras Andrea Escobar y Alejandra Berrios que desde pregrado me apoyaron con sus conocimientos para las diferentes actividades que tenía en este acápite de mi vida. A Dios ya la virgen de Copacabana muchas gracias y que este documento no sea el último, sino el principio de una vida de post grados en ingeniería.

Índice

1. Capítulo I: Marco referencial	1
1.1. Antecedentes.....	2
1.2. Identificación del Problema.....	4
1.3. Problemática	5
1.4. Justificación. -.....	5
1.4.1. Justificación académica.....	5
1.4.2. Justificación técnica.	5
1.4.3. Justificación operativa.....	6
1.4.4. Justificación legal y paradigma internacional.	6
1.5. Objetivos.....	6
1.5.1. Objetivo general.	6
1.5.2. Objetivos específicos.....	6
1.6. Alcance temático	7
1.6.1. Área de la investigación.	7
1.6.2. Tema específico.....	7
1.6.3. Nivel de investigación.....	7
1.6.4. Alcance temporal.....	7
1.6.5. Alcance espacial.....	7

1.7.	Diseño metodológico de la investigación	7
1.7.1.	Planteamiento de la hipótesis	7
1.7.2.	Enfoque de la investigación	8
1.7.3.	Universo y muestra.....	8
2.	Capitulo II: Marco Teórico	9
2.1.	Mapa de cosecha.....	10
2.2.	Modelos de intercambio de paquetes.....	11
2.2.1.	Modelo Cliente – Servidor	11
2.2.2.	Estructura de datos LoRa	11
2.2.3.	Modulación LoRa.....	12
2.2.4.	Rest.....	20
2.2.5.	Internet de las cosas (IOT)	20
2.2.6.	HTTP.....	21
2.2.7.	XMPP	21
2.2.8.	MQTT.....	22
3.	Capitulo III: Marco Practico	25
3.1.	Situación actual.....	26
3.1.1.	Personal de trabajo en San Buenaventura	26
3.1.2.	Datos a transmitir	26
3.1.3.	Detalles del terreno.....	27

3.2. Sensores de la cosechadora.....	28
3.3. Datos recolectados por los sensores	31
3.4. Propuesta de solución.	32
3.5. Parámetros de transmisión.....	33
3.5.1. Frecuencia de trabajo.	34
3.5.2. Nodo final y Gateway	34
3.5.3. Chip rate	36
3.5.4. Simbol rate (R_s)	36
3.5.5. Data rate (R_b)	36
3.5.6. Duración de chip (TC).....	37
3.5.7. Duración de símbolo (TS).....	37
3.5.8. Time on Air (ToS).....	37
3.5.9. Métodos de activación y seguridad	39
3.5.10. Link Budget.....	42
3.6. Análisis de repositorios protocolos y datos	46
3.6.1. Criterios de análisis	46
3.6.2. Restful	47
3.6.3. Protocolo de transporte.....	47
3.6.4. Modelo Pub-Sub y modelo Request-Response	48
3.6.5. Seguridad.....	49

3.6.6. Calidad de Servicio (QoS).....	49
3.6.7. Tamaño de cabecera	50
3.6.8. Soporte XML.....	50
3.6.9. Eficiencia energética	51
3.6.10. Implementación.....	52
3.7. Resumen de análisis de protocolos de aplicación.....	52
3.7.1. Topología general.....	53
3.8. Análisis de costos	55
4. CAPITULO IV: MARCO CONCLUSIVO	57
4.1. Conclusiones.....	58
4.2. Recomendaciones	58
4.3. Glosario de términos, siglas y acrónimos	59
4.4. Referencias Bibliográficas.....	61
5.....	61

Ilustración 1: Mapa de cosecha y detalle de puntos	10
Ilustración 2: Esquema de petición y respuesta del modelo cliente – servidor.....	11
Ilustración 3: Formato de mensaje LoRaWAN.....	12
Ilustración 4: Arquitectura XMPP.....	22
Ilustración 5 Comunicaciones entre nodos y broker en MQTT	23
Ilustración 6: Captura de pantalla sobre el peso que tiene el archivo de muestra sobre tomas de datos en campo.....	26
Ilustración 7 Perfil de terreno desde las oficinas del ingenio azucarero en san buenaventura y la última parcela ya cosechada.	27
Ilustración 8 Representación esquemática de componentes del monitor de rendimiento..	28
Ilustración 9 Esquema genérico de la posición de los sensores de grano.	29
Ilustración 10 Receptor equipado con doble procesador de señales GPS y Glonass con compensación del terreno en los tres ejes.	30
Ilustración 11 Sensor de humedad montado en la etapa de transporte de grano limpio. ...	30
Ilustración 12 Ejemplo de cabeza de cosechadora encendida y apagada.....	31
Ilustración 13 Grafica representativa de comunicación inalámbrica según la tasa de datos vs el alcance de la señal.....	33
Ilustración 14 Hardware de nodo Gateway RAK831 y nodo final LoraHatSx1662.....	34
Ilustración 15: Calculo de time on air para BW 125 KHz.....	38
Ilustración 16: Calculo de time on air para BW 500 KHz	39
Ilustración 17: diagrama OTAA LoRaWAN	41
Ilustración 18: Nodos de ejemplo de enlaces entre Gateway y nodo final	42
Ilustración 19: Figura comparativa de uso de batería entre red móvil y red lan inalámbrica WiFi.....	51

Ilustración 20: Esquema de la topología de comunicación entre cosechadora y servidor53

Ilustración 21: Esquema de la topología general usando comodin de mononivel55

Resumen

Actualmente se tiene una latencia en el transporte de datos de todas las cosechadoras sobre los granos recolectados. Estos datos indican si hubo por ejemplo exceso o falta de fertilizante, en un sector de la parcela. Los mapas de cosecha brindan esta y más información según los sensores que se tienen a disponibilidad. Sin embargo, este trabajo puede llegar a ser de paga gracias a las diferentes maquinarias censadoras del mercado; en adición, el análisis puede realizarlo cualquier persona profesional del área del agro y se puede prescindir de dichos servicios de pago. Solo es necesario obtener los datos de manera rápida y confiable.

Capítulo I: Marco referencial

1.1. Antecedentes

Actualmente se logró identificar preliminarmente 5 autores que apoyarán de gran manera a esta investigación. En primera instancia se tiene un estudio realizado por un investigador alemán con enfoques descriptivos en tecnologías puntuales. Posteriormente, existen 2 trabajos de investigación del sector privado del territorio argentino. Luego, se tiene dos investigaciones también del sector privado, donde uno de ellos tiene mucha relación con la investigación ya que es un resumen general sobre características que se deben tomar en cuenta para la implementación de IoT, siendo esta investigación centrada en la agroindustria.

A continuación, se realiza un repaso de los autores y las investigaciones mencionadas:

Mietz, R. (2018) realizó la investigación titulada “Tecnologías para una búsqueda basada en el estado en Internet de las cosas” para investigar sobre los requerimientos de los diferentes componentes desde la perspectiva del usuario, mecanismos flexibles y altos niveles de desarrollo de aplicaciones basadas en IoT. Para ello utilizó el método descriptivo, donde bajo parámetros ya establecidos en su proyecto como ser la base de datos a utilizar propuso que tecnología IoT es la mejor para su realidad a implementar. Los resultados alcanzados por la organización a la finalización del trabajo, fueron los de proponer una evaluación de las diferentes tecnologías basadas en estándares propietarios luego de investigar los requerimientos del usuario según sus perspectivas, introduciendo modelos probabilísticos con el requerimiento del almacenamiento de los datos en “DescentSPARQL”.

Mercado, G., Alvarez, L., Bocaccini, L., Ledda, M. (2017) realizaron una investigación titulada “Diagnóstico y Metodología para la Implementación de "Internet of Things" en el Planeamiento y Desarrollo de Ciudades Inteligentes.”. La investigación tuvo como objetivo diagnosticar, evaluar y ponderar soluciones de sustentabilidad urbana, utilizando tecnologías inteligentes (IoT), aplicadas en casos de estudio locales. Por consiguiente, se utilizó el método descriptivo/analítico en su investigación. Los resultados alcanzados por los autores al concluir su investigación fueron los de identificar los actores críticos intervinientes para la sustentabilidad urbana, además de identificar de los ejes temáticos para la implementación de IoT.

Díaz, J., Venosa, P., Fava, L., Castro, N. (2017) realizó una investigación titulada “Estrategias de IoT para Lograr Ciudades Digitales Seguras, más Inclusivas y Sustentables”. La investigación tuvo como objetivo la Investigación y desarrollo de tecnología centrada en la movilidad urbana, eficiencia energética, producción frutihortícola, salud y deporte. Se usó el método descriptivo gracias a que se necesitaba realizar un análisis, con este método, luego de identificar todas las variables necesarias en el trabajo se deben enunciar y contrarrestar con las tecnologías mencionadas en la investigación. El resultado obtenido fue el de presentar varias líneas de trabajo, destinadas a proveer a los ciudadanos una serie de servicios basados en tecnologías modernas y digitales, que les permite un uso más eficiente y una mayor interacción con los mismos, generando una retroalimentación al sistema global que permita mejorar su calidad de vida, teniendo en cuenta la sensibilidad de la información que los mismos manejan y los requerimientos de seguridad asociados.

Noguera, J. (2016) realizó una investigación titulada “Sistema de Diálogo Basado en Mensajería Instantánea para el Control de Dispositivos en el Internet de las Cosas”. Este trabajo de investigación tuvo como objetivo el Desarrollar un sistema de comunicación basado en el lenguaje natural tipo mensajería instantánea. En su tesis de doctorado utilizó el método descriptivo y analítico para tener como resultado el de implementar según el protocolo MQTT una comunicación de mensajería instantánea Im4things, obteniendo buenos resultados tanto para el usuario como por el procesamiento del servidor.

Clemente, M., Benitez, D., Anías, C., Regalado, E. (2015) realizaron una investigación titulada “Procedure For Internet of Things implementation”. Se tuvo como objetivo el de Puntualizar los requerimientos generales a partir de análisis de datos de los tipos de datos que se manejan en la solución en cuestión. Para ello se utilizó el método descriptivo y analítico identificando todas las variables de las empresas en su territorio y comparando con las directrices de los diferentes sistemas en producción. Los resultados de los autores al finalizar la investigación fueron los de elaborar un procedimiento para la implementación de IoT, que integra los principales elementos que se deben tener en cuenta en esta tecnología aplicada a la industria petrolera.

1.2. Identificación del Problema

El documento elaborado por los 145 países participantes de la Exposición Universal Milán Italia, fue trabajado en 26 mesas temáticas que abordan conceptos desde la lucha contra el hambre y la pobreza, hasta la defensa de la biodiversidad y el medio ambiente. La Soberanía Alimentaria se trató en la mesa 17 encabezada por la FAO y 11 estados y organizaciones internacionales, La directora del pabellón de Bolivia, Velasco, K. (2015), como representante del estado manifestó lo siguiente: “hasta nuestra participación, se preveía trabajar sobre la seguridad alimentaria, pero como política internacional de Bolivia, logramos hacer comprender a los demás estados la importancia de la soberanía alimentaria para el buen vivir”. El reporte final de la mesa 17 “Soberanía Alimentaria” es la herramienta fundamental para el logro de la seguridad alimentaria como se define en la Cumbre Mundial sobre la Alimentación y para ello en dicha mesa se concluyó que “la soberanía alimentaria es el derecho de cada país y sus ciudadanos en las estrategias sostenibles para la producción, distribución y consumo de alimentos que garanticen el derecho a una vida sana y nutritiva para toda la población”.

Según lo mencionado por Fajardo, H. (2020) en su entrevista para reconocer la realidad de la agroindustria en el territorio boliviano, en su calidad de becario investigador para el ministerio de desarrollo productivo, los sensores montados en las cosechadoras, realizan un mapeo del recorrido; en dicho momento, se puede analizar las secciones donde se pudo cosechar mayor cantidad de maíz, estos datos pueden ser exportados a una capa raster y pueden ser visualizados en una plataforma de un sistema de información geográfica. Sin embargo, el problema se encuentra en la centralización de la información de todas las cosechadoras que hicieron un barrido; más aún, generar una sola capa general de todo el terreno que se está cosechando.

Un ejemplo de sistemas automatizados es el de John Deere, que tiene una plataforma basada en la nube, pero la información es entregada 24 horas después de que todos los puntos fueron cargados. Impidiendo así el análisis en sitio del estado de la parcela que fue cosechada en la jornada. Brindando dos paradigmas para la agroindustria. Se puede recolectar la información de manera “manual”, en otras palabras, cada cosechadora recabe la información en un medio de almacenamiento portátil y luego de que todas terminaron de cosechar, en ese momento proceder a centralizar, filtrar y consolidar la información de la cosecha del día. O pagar un servicio web que

tiene una demora de 24 horas mínimamente, el cual es privativo tanto en su plataforma como con sus sensores.

Por lo expuesto surge la necesidad de que la industria adecue sus funciones de su gestión de la información para mejorar el uso de los recursos naturales y el impulso de los sectores productivos según establece las directrices de la soberanía alimentaria del estado plurinacional de Bolivia, de forma indiferente al uso de sensores de diferentes fabricantes y el periodo de inicio y fin de cosecha de cada cosechadora.

1.3. Problemática

¿Cuál es el mejor protocolo de comunicación IoT para optimizar el uso del ancho de banda en comunicaciones entre sensores y servidor, en soluciones aplicadas a la agroindustria?

1.4. Justificación. -

1.4.1. Justificación académica.

El presente documento va dirigido a dar un pantallazo para el uso de “IoT” en la agroindustria. Siendo este un tema nuevo, se espera que este trabajo sea un referente investigativo para posteriores investigaciones teóricas como para trabajos ya aplicados en el área de agronomía. Una de las formas teóricas que puede aportar en el crecimiento de esta investigación es que los estudios se centren en los cálculos necesarios para la generación de mapas de cosecha.

1.4.2. Justificación técnica.

Al utilizar IoT se optimizará el uso del ancho de banda a comparación de solicitudes HTTP al momento de transportar la información. Esto en algunos equipos que necesitan un chip, mínimamente 3G, de algún proveedor de servicios de Internet para poder enviar los datos necesarios. Al reducir el ancho de banda necesario para transportar la información se reducirá el costo del plan de megabytes disponibles.

1.4.3. Justificación operativa

La centralización de la información de varios equipos, no necesariamente del mismo fabricante, permitirá la detección casi en tiempo real del comportamiento de su terreno y la realización de acciones correctivas. Con el pasar del tiempo se generarán historiales del terreno para así poder calcular de manera más eficaz la compra de fertilizantes y el óptimo uso de recursos naturales.

1.4.4. Justificación legal y paradigma internacional.

El 25 de septiembre de 2015, los 193 estados miembros de las Naciones Unidas adoptaron los 17 Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030 para el Desarrollo Sostenible, que según lo previsto deberían guiar las acciones de la comunidad internacional por los próximos 15 años. El segundo objetivo habla de poner fin al hambre, lograr la seguridad alimentaria y mejorar la nutrición y promover la agricultura sostenible. Bolivia además de ser parte de las Naciones Unidas y su obligación de cumplir con la agenda 2030 mencionada, en materia de alimentos se puede encontrar la soberanía alimentaria y la ley número 300 del 15 de octubre de 2012. El proyecto permitirá poner en práctica las declaraciones de intenciones a las que normalmente se reduce la normativa vigente.

1.5. Objetivos

1.5.1. Objetivo general.

Determinar el mejor protocolo de comunicación IoT para optimizar el uso del ancho de banda en la comunicación entre sensores, servidor y usuario final para la generación de mapas de cosecha.

1.5.2. Objetivos específicos.

- Describir las características de los protocolos de conexión MQTT, HTTP y XMPP.
- Determinar cuál de los protocolos de comunicación tendría mayor impacto en la agroindustria para la generación de mapas de cosecha.
- Plantear un diseño lógico para la red de comunicación.
- Plantear una topología que pueda soportar la comunicación entre sensores y servidor.
- Describir el formato y la estructura de los datos para investigaciones posteriores.

1.6. Alcance temático

1.6.1. Área de la investigación.

Internet de las Cosas (IoT).

1.6.2. Tema específico

Diseño de un sistema de comunicación basado en “Internet of Things” para la generación de mapas de cosecha para la agroindustria.

1.6.3. Nivel de investigación

La investigación del presente trabajo se basó en un tipo de investigación teórica propositiva ya que se presenta una comparación de protocolos de comunicación cuyo objetivo es el ahorro de ancho de banda para poder transportar mayor información recolectada en campo y así proponer su implementación en el área de la agroindustria

1.6.4. Alcance temporal

La investigación usara datos del año 2019 que fueron recolectados en campo en época de cosecha.

1.6.5. Alcance espacial

Se realizará la investigación únicamente para la realidad que se encuentra cursando el territorio boliviano en la agroindustria. Con los equipos y sensores más utilizados en nuestro territorio. La investigación no se centrará en diseñar un sistema de generación de mapas de cosecha sino en el medio de comunicación entre sensores y servidor.

1.7. Diseño metodológico de la investigación

1.7.1. Planteamiento de la hipótesis

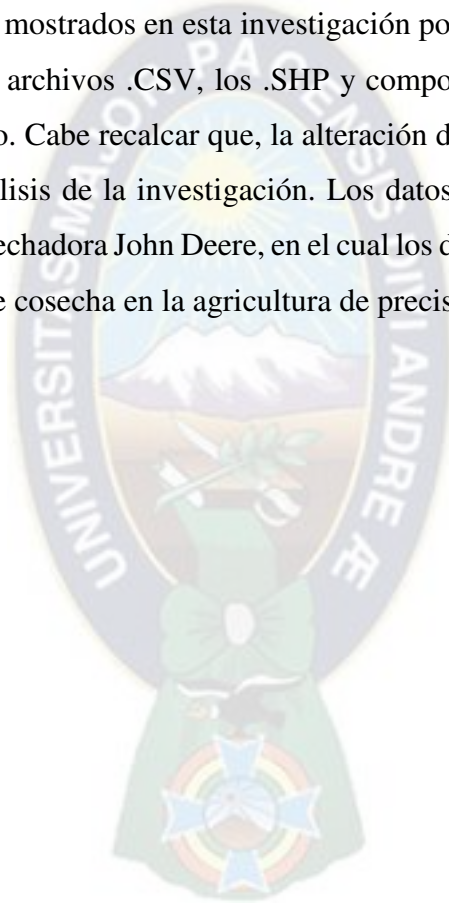
Si se aplica un protocolo de comunicaciones IoT, entonces se optimizará el uso del ancho de banda para la comunicación entre sensores y servidor en la agroindustria.

1.7.2. Enfoque de la investigación

Se hará uso del enfoque sistémico para poder tener un panorama y paradigma amplio de la realidad boliviana respecto al área tecnológica al momento de realizar la comparación y argumentar que protocolo se ajusta mejor para encontrar una solución a la problemática.

1.7.3. Universo y muestra

La recolección de datos se centra en el sector norte de Santa Cruz de la Sierra del terreno boliviano. Los datos que serán mostrados en esta investigación por solicitud del propietario de los puntos mostrados tanto en los archivos .CSV, los .SHP y componentes no indicarán la posición exacta en el territorio boliviano. Cabe recalcar que, la alteración de los valores de geolocalización no afectará en absoluto al análisis de la investigación. Los datos que se tomarán como *payload* fueron recolectados de una cosechadora John Deere, en el cual los datos a recolectar son un estándar para la generación de mapas de cosecha en la agricultura de precisión.



Capítulo II: Marco Teórico



2.1. Mapa de cosecha

(...) Un mapa de rendimiento o mapa de cosecha posee formato vectorial, donde los datos se representan utilizando puntos y un sistema de coordenadas x-y que hace referencia a las ubicaciones del mundo real. Cada punto de un mapa posee coordenadas, y la cantidad de información que se obtiene de cada punto depende del monitor de rendimiento que se utilice; estos pueden ser: altura de cabezal, ancho de franja, distancia recorrida, velocidad de avance, flujo de granos, humedad de grano, pérdidas de cosecha, entre otros.

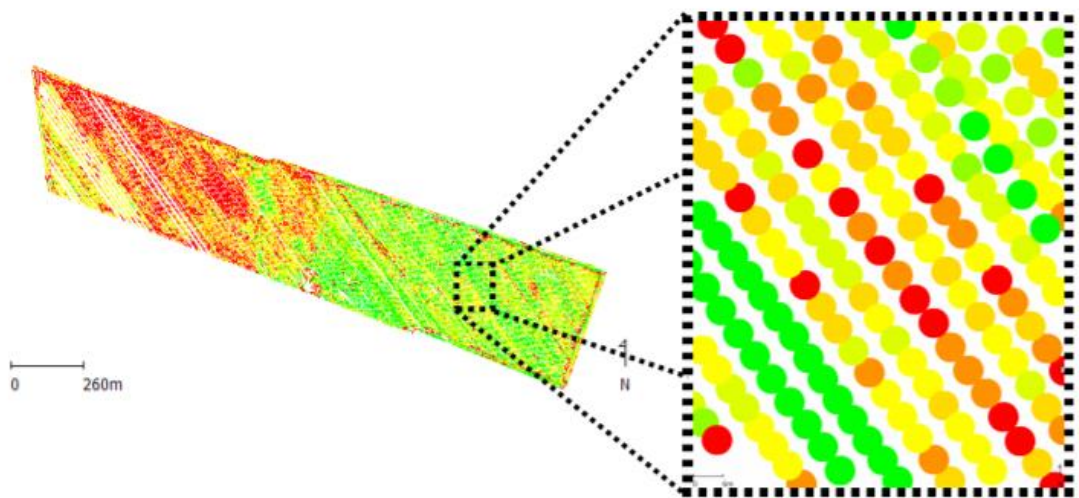


Ilustración 1: Mapa de cosecha y detalle de puntos

Fuente: (Albarenque, 2011)

Los mapas de rendimiento se obtienen a partir de los datos recopilados por una cosechadora que incluye un sistema de posicionamiento global, junto con un sistema de sensores que calculan la cantidad de grano cosechado. Existen diferentes monitores de rendimiento actualmente disponibles en el mercado. Entre ellos se encuentran los que utilizan una placa de impacto en el camino del flujo de grano. El impacto de los granos es medido por una celda de carga, dispositivo que transforma la fuerza incidente en una señal eléctrica. Adicionalmente, la mayoría de los sistemas de monitoreo de rendimiento incluyen un sistema automático de medición de humedad del grano. Esto permite que cada dato de rendimiento tenga un valor de contenido de humedad asociado, el cual puede ser utilizado para representación gráfica del rendimiento a un contenido de

humedad estándar (Albarenque, Técnicas de procesamiento de mapas de rendimiento, 2011)
(Albarenque, Técnicas para el procesamiento de mapas de rendimiento, 2011)

2.2. Modelos de intercambio de paquetes

2.2.1. Modelo Cliente – Servidor

Según (Gonzales, 2016) en su texto de teoría sobre arquitectura cliente servidor, define a cliente como programa ejecutable que participa activamente en el establecimiento de conexiones enviando una petición al servidor esperando una respuesta del mismo. Además, el servidor es un programa que ofrece un servicio que se puede obtener en una red encargado de aceptar la petición desde la red, realizar el servicio solicitado y devolver el resultado al solicitante.

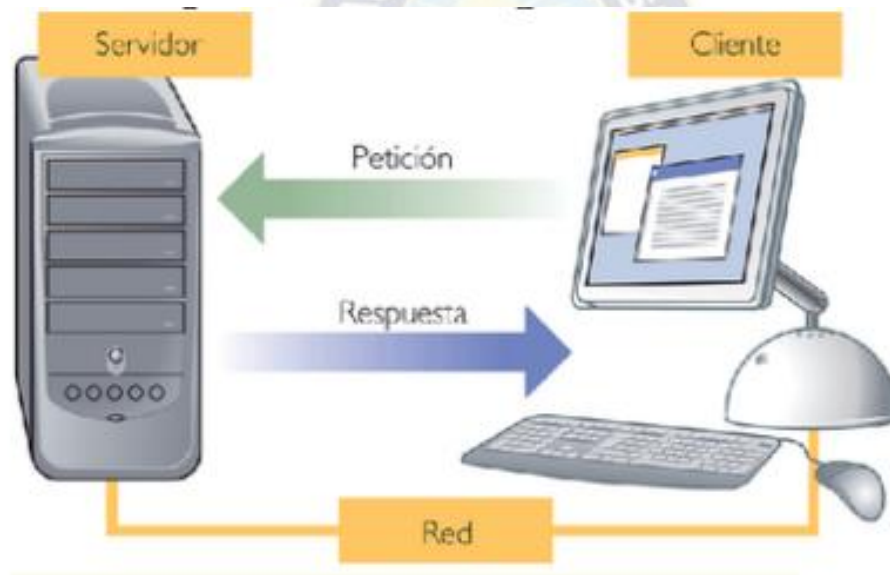


Ilustración 2: Esquema de petición y respuesta del modelo cliente – servidor

FUENTE: (Gonzales, 2016)

Existen otros modelos de los cuales se puede comparar

2.2.2. Estructura de datos LoRa

Según (Dongkyun, 2020), en su estudio sobre parámetros LoRa la estructura de bits es la siguiente:

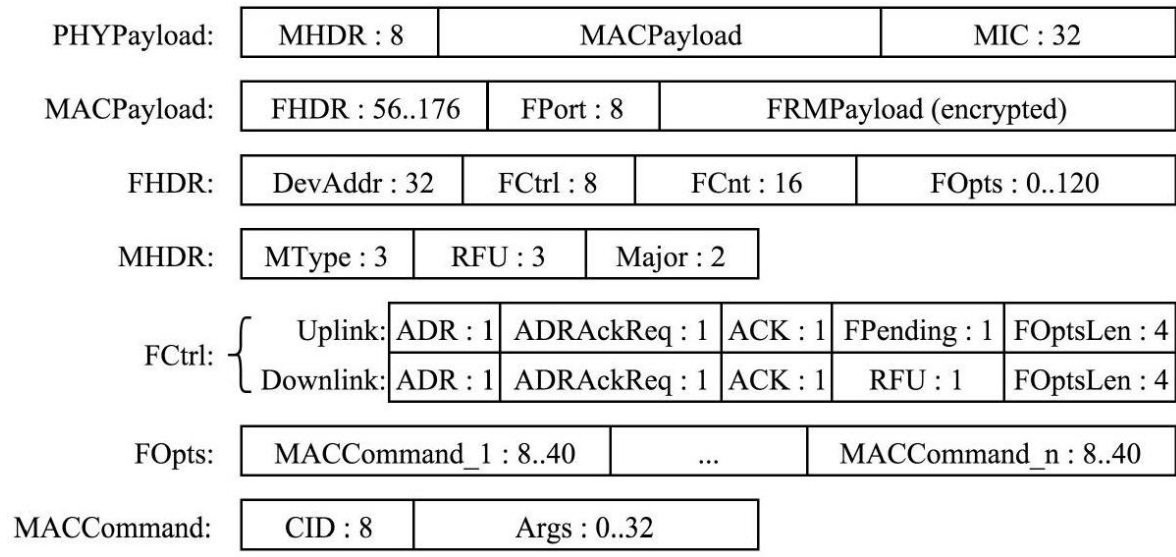


Ilustración 3: Formato de mensaje LoRaWAN

FUENTE: (Dongkyun, 2020)

En capa física se evidencia el preámbulo que comienza la secuencia de chirps de subida para abarcar toda la frecuencia, luego del preámbulo se evidencia una cabecera opcional que es usada con una tasa de codificación 4/8. Posteriormente se envía el payload de la trama. En la capa de control de acceso al medio se encuentran todos los parámetros de configuración para la transmisión entre el Gateway y el servidor de red y/o el servidor de aplicación.

2.2.3. Modulación LoRa

Según (Ordoñez , 2017) en su publicación “ Estudio de la arquitectura y el nivel de desarrollo de la red LoRaWAN y de los dispositivos LoRa.” Menciona que LoRa opera en la banda ISM por debajo de 1 Ghz (en Europa se corresponde a las bandas 868 y 433 Mhz). Al ser una banda libre se puede usar sin licencia. Utiliza como esquema de modulación de la señal una variación de la modulación DSSS de tipo Spread Spectrum denominada Chirp Spread Spectrum (CSS) que permite conexiones de bajo coste, bajo consumo (25 mA en transmisión y 10 mA en recepción) robustas frente a interferencias y resistentes al efecto Doppler. Tal y como se vio en un apartado anterior, el uso de este tipo de esquema de modulación implica una mayor complejidad de los dispositivos de comunicación y esto puede repercutir en el precio de los mismos lo que puede suponer un problema en redes de cientos de miles de dispositivos.

LoRa utiliza un método de ajuste dinámico de la potencia de emisión y la tasa de transferencia denominado ADR (Adaptative Data Rate) que principalmente permite al dispositivo final o a la red el ajuste dinámico de los parámetros antes citados en función de la distancia que separa al dispositivo final de la pasarela y del tamaño del mensaje. El objetivo perseguido es que las comunicaciones sean energéticamente eficientes empleado la máxima velocidad posible.

Con el fin de mejorar la eficiencia en el uso del espectro y la capacidad de la red LoRa permite seleccionar entre seis factores ortogonales de ensanchamiento (en adelante SF de Spreading Factor) numerados del 7 al 12 en los que para cada uno se define una relación entre potencia y tasa de transferencia. A mayor SF, mayor es la sensibilidad del receptor y por lo tanto mayor el alcance del enlace por contra la velocidad del enlace decrece (Corporation, 2016). En concreto la velocidad del enlace varía desde 0.3 kbps para un SF de 12, hasta 5486 kbps para un SF de 7. La sensibilidad de recepción varía entre -137dBm para un SF de 12 y -123 dBm para un SF de 7 lo que, como se ha comentado con anterioridad, tiene un impacto directo sobre el alcance de la conexión siendo el máximo de 14 km para un SF de 12 y de 2 km para un SF de 7¹.

2.2.3.1. OTAA.

El dispositivo final debe seguir un procedimiento de conexión con la red para poder participar en la misma. En el dispositivo final se debe configurar de manera manual con una serie de parámetros necesarios para el procedimiento de conexión, que son: un identificador global único (DevEUI) que identifica de forma única al dispositivo utilizando el esquema IEEE EUI64, el identificador de aplicación (AppEUI) y una llave criptográfica AES-128 (AppKey) específica para el dispositivo de la que se derivará la llave de sesión y la llave de sesión de aplicación.

¹ Según N. Ducrot, D. Ray, A. Saadani, O. Hersent, G. Pop, G. Remond, en su publicación “LoRa Device Developer Guide”

2.2.3.2. Clases de LoRaWAN

Existen tres tipos de clases en la red LoRaWAN de dispositivos según su funcionalidad que soporta: se divide en clases A, B y C. Es mandatorio que todos los dispositivos deban cumplir las exigencias de la clase A y es opcional las exigencias de clase B y C.

Según (N. Sornin, 2020) en su trabajo LoRaWAN specifications 1.0.2, indica que las tres clases pueden coexistir en la misma red y los dispositivos finales pueden cambiar su configuración entre clases.

Clase A.

La planificación de las transmisiones corre a cargo del propio dispositivo final. La recepción, que sólo está permitida después de una transmisión completada correctamente, está formada por dos ventanas separadas de recepción. Si en la primera ventana ya se han recibido los datos la segunda ventana se deshabilita.

Esta clase es la más eficiente desde un punto de vista energético y es adecuada para aplicaciones que prácticamente sólo requieran enviar datos.

Clase B.

Esta clase permite la creación de ventanas de recepción sin la necesidad de que se produzca una transmisión previa. Se aumenta pues la capacidad del dispositivo final de recibir datos. Mediante el envío de beacons por parte de la pasarela, ésta se sincroniza con el dispositivo final con el fin de planificar el tiempo en el que el dispositivo debe abrir la ventana de recepción.

La consecuencia inmediata de este aumento de la capacidad de recepción es el aumento del consumo eléctrico en el dispositivo final debido al coste energético del mecanismo de sincronización.

Clase C.

Este tipo de dispositivos están en modo de recepción permanente que sólo se interrumpe cuando se produce una transmisión. Esta clase de dispositivo presenta la mejor latencia de conexión entre los dispositivos finales y las pasarelas a cambio de un mayor consumo.

Las clases representan un balance entre consumo energético y capacidad de recepción. De esta forma se intentan perfilar los dispositivos en función de tres tipos de aplicaciones: las que prácticamente no necesitan enviar datos al dispositivo final, las que tienen una demanda media y finalmente las que tienen gran demanda de envío de datos al dispositivo final. Las clases son un ejemplo más del buen diseño de LoRaWAN dado que de una forma muy sencilla permiten indicar el perfil de consumo energético de un dispositivo.

En la clase de dispositivos B se introduce además la posibilidad de enviar tráfico de tipo multicast lo que permite dar soporte a aplicaciones que requieren este tipo de tráfico, así como la posibilidad de realizar pings.

2.2.3.3. Calidad de Servicio en IoT (QoS)

2.2.3.3.1. Paquetes publicados con QoS=0

El paquete de control de publicación se envía desde un publicador a servidor o desde un servidor a abonado. La función principal del paquete es transmitir la información del mensaje del tema a servidor o suscriptor. Un ejemplo es el siguiente:

Bit		7	6	5	4	3	2	1	0	Comment
Byte 0	Fixed header	0	0	1	1	0	0	0	0	MQTT control Packet type, DUP flag, QoS level, RETAIN
Byte 1		0	0	0	1	1	0	0	0	
Byte 2	Variable header	0	0	0	0	0	0	0	0	Length MSB
Byte 3		0	0	0	1	0	1	0	0	Length LSB
Byte 4-23										'home/garden/fountain'
Byte 24	Payload	0	0	1	1	0	0	0	1	'1'
Byte 25		0	0	1	1	0	0	1	1	'3'

Tabla 1: La publicación de paquetes de control con QoS = 0

FUENTE: (Junyang, 2016)

El mensaje consta de tres partes, el encabezado fijo, el encabezado variable y la carga útil.

El encabezado fijo contiene información importante sobre el mensaje. La figura a continuación ilustra las banderas en el byte 0.

Bit	7	6	5	4	3	2	1	0
Byte 0	0	0	1	1	x	x	x	x
Flag	MQTT Control Packet type				DUP flag	QoS level		RETAIN

Tabla 2: El Byte 0 de control

FUENTE: Elaboración propia

Según (Junyang, 2016) en la posición 7-4 de este byte 0 estos bits indican el tipo de paquete de control y en este caso el valor es 3. En el bit 3, si la bandera UDP se establece en 0, significa que el mensaje es la primera vez que el cliente o servidor envía y si la bandera se establece en 1, significa que el mensaje se vuelve a enviar de un intento anterior de enviar el paquete. En los bits 2-1 contiene el valor del nivel de QoS que indica el nivel de seguridad para la entrega de un mensaje de aplicación.

QoS= 0 : máximo una entrega

QoS =1: al menos una entrega

QoS=2: Exactamente una entrega.

El último bit representa una bandera de retención. En primer lugar, discutimos el mensaje de publicación que se envía. al servidor por cliente. Si el indicador de retención del paquete de publicación del cliente al servidor se establece en 1, el servidor debe almacenar el mensaje de la aplicación y su QoS y el servidor debe entregar el mensaje para futuros suscriptores cuyas suscripciones coincidan con el nombre de su tema. Entonces, cuando una nueva suscripción es

establecida, se enviará al suscriptor el último mensaje retenido que coincida con el tema. Si el cliente desea cambiar el mensaje de retención, debe enviar un mensaje de publicación con RETAIN bandera establecida en 1 y QoS establecida en 0. Entonces el servidor debe descartar cualquier mensaje retenido para ese tema y almacene el nuevo mensaje de QoS 0 como el nuevo mensaje de retención para ese tema. Puede elegir para descartarlo en cualquier momento, si esto sucede, no habrá ningún mensaje retenido para ese tema. Un mensaje publicado con un indicador de retención establecido en 1 y una carga útil que contenga cero bytes será procesado normalmente por el servidor, pero cualquier mensaje retenido con el mismo nombre de tema debe ser eliminado y no habrá ningún mensaje retenido para ese tema. Cuando el mensaje se envía a cliente por servidor, el indicador de retención debe establecerse en 1, si el mensaje se envía debido a una nueva suscripción realizada por el cliente. En otro caso, la bandera de retención debe establecerse en 0, sin importar cómo se colocó la bandera en el mensaje cuando se recibió. El byte 1 indica la longitud del encabezado variable y la carga útil. El encabezado de variable contiene campo de nombre de tema y campo de identificador de paquete. El nombre del tema debe estar presente en el primer archivo y ser una cadena codificada en UTF-8. El campo de identificador de paquete solo se usa cuando la QoS es 1 o 2. Por tanto, en la ilustración cuatro no hay un identificador de paquete. La última parte es la carga útil. Contiene los datos que es específico de la aplicación. Es válido que un paquete de publicación contenga una carga útil de longitud cero.

2.2.3.4. Paquetes publicados con QoS=1

El paquete de publicación con QoS nivel 1 es casi el mismo que el paquete con QoS nivel 0. El único campo que se debe agregar es el identificador de paquete que se utiliza para reconocer el paquete.

Bit		7	6	5	4	3	2	1	0	Comment
Byte 0	Fixed header	0	0	1	1	0	0	0	0	MQTT control Packet type, DUP flag, QoS level, RETAIN
Byte 1		0	0	0	1	1	0	0	0	
Byte 2	Variable header	0	0	0	0	0	0	0	0	Length MSB
Byte 3		0	0	0	1	0	1	0	0	Length LSB
Byte 4-23										'home/garden/fountain'
Byte 24	Packet Identifier	0	0	0	0	0	0	0	0	Message Identifier 2
Byte 25		0	0	0	0	0	0	1	0	
Byte 26	Payload	0	0	1	1	0	0	0	1	'1'
Byte 27		0	0	1	1	0	0	1	1	'3'

Tabla 3 paquete de PUBLISH con QoS=1

FUENTE: (Junyang, 2016)

El receptor debe enviar un PUBACK, si ha recibido el mensaje de publicación con QoS nivel 1 y el remitente almacenará el mensaje hasta que reciba el paquete de devolución. Si el remitente no recibe el reconocimiento en un período de tiempo razonable, enviará el mismo PUBLICAR mensaje una vez más. Si un remitente recibe el mensaje PUBACK, comparar los identificadores de paquetes en el paquete de publicación y el paquete de devolución para garantizar la llegada de publicar mensaje. La tabla 14 nos muestra la estructura de PUBACK.

bit		7	6	5	4	3	2	1	0	Comment
Byte0	Fixed header	0	1	0	0	0	0	0	0	MQTT Control Packet type
Byte1		0	0	0	0	0	0	1	0	Remaining Length
Byte2	Variable header	0	0	0	0	0	0	0	0	Packet Identifier
Byte3		0	0	0	0	0	0	1	0	

Tabla 4 Paquete PUBACK

FUENTE: Elaboración propia

El paquete solo contiene dos encabezados, el encabezado fijo y el encabezado variable. El fijo El encabezado define el tipo de paquete y la longitud restante. En el encabezado de la variable del paquete PUBACK solo tienen 2 bytes, por lo que el valor de la longitud restante es 2. El encabezado de la variable contiene el identificador del paquete de publicación.

2.2.3.5. Paquetes publicados con QoS=2

El mensaje de publicación con QoS nivel 2 es el mismo que el mensaje con QoS nivel 1. El mensaje de publicación recibido es el mismo que el mensaje de devolución en estructura y solo cambia el código de tipo de paquete de control MQTT a 2 # 0101.

bit		7	6	5	4	3	2	1	0	Comment
Byte0	Fixed header	0	1	0	1	0	0	0	0	MQTT Control Packet type
Byte1		0	0	0	0	0	0	1	0	Remaining Length
Byte2	Variable header	0	0	0	0	0	0	0	0	Packet Identifier
Byte3		0	0	0	0	0	0	1	0	

Tabla 5 Paquete PUBREC

FUENTE: Elaboración propia

El mensaje de lanzamiento de publicación es el mismo que el mensaje de devolución en estructura y solo cambia el primer código de byte a 2 # 0110 0010. Y con la misma estructura que el paquete PUBREC. El mensaje de lanzamiento de publicación es el mismo que el mensaje de devolución en estructura y solo cambia el primer código de byte a 2 # 0110 0010. El mensaje de publicación completa es el mismo que el mensaje PUBACK en estructura y solo cambia el código de tipo de paquete de control MQTT a 2 # 0111.

El paquete de desconexión se utiliza para cerrar la conexión y se envía del cliente al servidor. Se muestra la estructura del paquete de desconexión. Solo hay un encabezado fijo. El valor de longitud restante es 0, porque no hay encabezado de variable.

bit		7	6	5	4	3	2	1	0	Comment
Byte0		1	1	1	0	0	0	0	0	MQTT Control Packet type
Byte1		0	0	0	0	0	0	0	0	Remaining Length

Tabla 6 Paquete de desconexión

FUENTE: Elaboración propia.

2.2.4. Rest

Una parte importante de desarrollo de aplicaciones web es entender que se trata de una constante comunicación entre dos equipos denominados cliente y servidor. Se tiene que establecer reglar de comunicación entre ambos. Para definir como nos comunicamos con el servidor desde el cliente se usará el “REpresentational State Transfer” también conocido como REST por sus siglas en ingles ya que se tiene que separar las responsabilidades que tiene el cliente con las del servidor. Entonces para analizar si es restful nuestro protocolo se debe analizar si hay una arquitectura cliente/servidor, el principio del no estado que implica que cada petición debe ser independiente y nunca dependes de peticiones anteriores, si puede almacenar información en caché ya que el servidor puede definir respuestas cacheadas o no, si el servidor basado en capas que puedan comunicarse entre si, posee una interfaz uniforme como pieza angular ya que es el estándar para la comunicación entre el cliente y el servidor y si puede enviar código On Demand. Sin contar con la última opción, si no cumple con los puntos mencionados no se puede considerar al protocolo como Restful.

2.2.5. Internet de las cosas (IOT)

Según Nitin, N. (2017) en su artículo “Choice of Effective Messaging Protocols of IoT systems” indica que en las comunicaciones del Internet de las cosas predomina la intercomunicación entre equipos sensores, usuarios y nodos. Varias redes implementan tecnologías de radio frecuencia como ser RFID, WLAN (IEE 802.11), PAN (IEE 802.15) y WMAN (IEEE 802.16) para sus comunicaciones de bajo nivel.

Con la selección de un protocolo IoT apropiado para el sistema, el prerequisite es el entendimiento sobre el objetivo del sistema y los requerimientos de su comunicación de mensajes y datos.

En su artículo “A survey of Messaging protocols” Jaikar, S. y Lycr, K. (2018) indican que a diferencia de la web que usualmente se utiliza estándares simples de mensajería como ser el protocolo HTTP, IoT no puede confiar en un solo requisito para todo lo que necesita. Cientos de protocolos de mensajería están disponibles para los requerimientos de los sistemas de IoT. Algunos fueron diseñados para abordar aplicaciones que requieren transacciones más rápidas y confiables.

Otros han sido para abordar aplicaciones que requieren recopilación de datos en redes restringidas. Además, otros fueron diseñados para abordar aplicaciones que requieren mensajería instantánea y detección de presencia de línea. Y unos pocos fueron diseñados para abordar aplicaciones web que requieren comunicarse a través de Internet como un cliente / servidor RESTful.

Esto claramente muestra que el futuro de la tecnología IoT se queda en manos de varios protocolos de mensajería y los que se ajusten a sus necesidades. Consecuentemente es necesario investigar los pros y contras de varios protocolos famosos para determinar cuál es el mejor que se ajusta.

2.2.6. HTTP

Atmoko, R., Riantini, R. y Hasin, M. (2017) definen al protocolo HTTP como el protocolo de comunicación por el cual las redes de Internet fueron construidas. Diferentes tipos de datos, imágenes o texto son enviados o recibidos de manera rápida entre servidores y navegadores, este protocolo está basado en TCP y se asegura que los datos transmitidos de un dispositivo a otro no estén corruptos asegurando su integridad. HTTP es un protocolo abierto de comunicación que puede ser leído por cualquier dispositivo que fue desarrollado para soportar HTTP, como ser los navegadores. Una transacción HTTP consiste en dos partes: solicitud, que es enviada del cliente al servidor, y respuesta, que es enviado del servidor al cliente. El proceso de respuesta y solicitud se envía utilizando un bloque de datos con un formato específico conocido como mensaje HTTP, los mensajes enviados por HTTP solamente pueden ser en una sola dirección según sea solicitud o sea respuesta. Cabe recalcar que la versión de HTTP 3 ya se encuentra en su etapa final de revisión para posterior implementación; sin embargo, debido a que no se cuenta con la RFC concreta se realizará el estudio con su versión anterior ya implementada en todo el mundo.² (Force, 2020)

2.2.7. XMPP

La descripción del protocolo según Wang, H., Xiong, D., Wang, P. y Liu, Y. (2017) indica que Extensible Messagin and Presence Protocol fue diseñado para abordar el problema de heterogeneidad en las redes de IoT, siendo luego estandarizado por IETF. Inicialmente fue diseñado para aplicaciones que únicamente son usadas en Internet, siendo este usado para mensajería instantánea, aplicaciones de entretenimiento a tiempo real y sistemas de telepresencia.

² Según el documento que reemplazara al draf-shade-quiv-http2-mapping de la IETF en su proposición de estándar.

Google talk, Facebook chat son ejemplos donde se usaba el servicio de mensajería en tiempo real. XMPP es un protocolo IP que soporta XML.

Geng, H., Salman, T., y Jain, R. (2016) mencionan características interesantes de XMPP, mencionando que también usa la arquitectura de publicación/suscripción como MQTT, como también soporta la arquitectura solicitud/respuesta. Esto le da una gran flexibilidad para el desarrollador para escoger la arquitectura necesaria según el caso. XML aborda problemas de heterogeneidad, pero, por el contrario, agrega una sobrecarga adicional debido a la gran cantidad de etiquetas y formatos de encabezado que aumenta el consumo de potencia en dispositivos IoT. Este problema puede ser resuelto con compresiones XML usando EXI. Soporta nodos de sensores que se suspenden periódicamente, extendiendo su tiempo de vida de dichos dispositivos.

XMPP permite que dispositivos heterogéneos interactúen con cada uno, enviando mensajes instantáneos a través de Internet independientemente de los sistemas operativos subyacentes. Se lo considera un protocolo seguro ya que admite cifrado, autenticación y control de acceso para la adición de diferentes nuevas aplicaciones en paralelo a los protocolos centrales existentes.

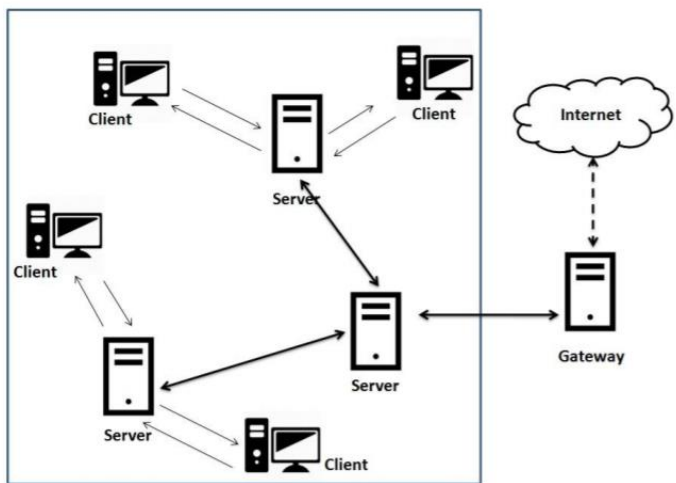


Ilustración 4: Arquitectura XMPP

Fuente (Sagar, 2018)

2.2.8. MQTT

MQTT es un protocolo de comunicación ligera de máquina a máquina diseñado para conectar dispositivos pequeños a redes restringidas. Su modelo de publicación / suscripción utiliza 2 bytes de encabezado, y para que sea confiable utiliza el modelo orientado a conexión TCP. Según el portan oficial de MQTT organization (2019), el cliente MQTT publica mensajes a un MQTT

bróker (se encuentra orientado a conexión), donde otros clientes se encuentran suscritos, cada mensaje se publica a una dirección conocida por su tópico. Los clientes pueden suscribirse a diferentes tópicos y recibir cada mensaje publicado en cada uno de los tópicos. MQTT es un protocolo binario con pequeños payloads con un máximo de 256 MB y usa protocolos de seguridad TSL/SSL.3

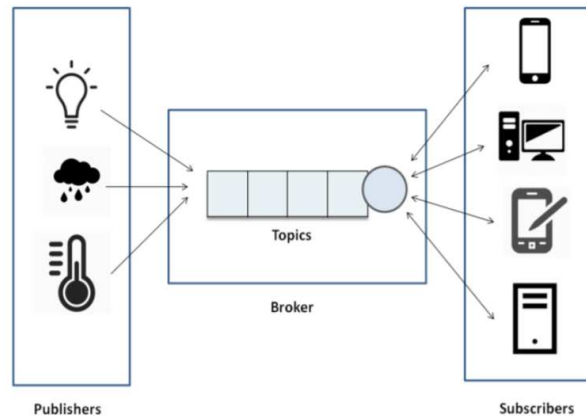


Ilustración 5 Comunicaciones entre nodos y broker en MQTT

Fuente (Sagar, 2018)

“... El funcionamiento de los blogs serviría como una analogía para explicar el funcionamiento de MQTT. Cuando en el blog se escribe algo interesante en un tópico determinado, los suscriptores a éste reciben una notificación de lo que se publicó; así que cualquier interesado en su tópico puede leer este blog. De esta misma manera funciona MQTT: los bloques de datos se envían por medio de mensajes de publicación, mientras se reciben por mensajes de suscripción; en ambos casos los datos son identificados por tópicos. “Jaikar, S. y Lyer, K. (2018).

2.2.8.1. Formato de control de paquetes

El control de paquetes sigue la siguiente estructura:

³ MQTT también en su portal habla sobre 3 grandes niveles de calidad de servicio para la entrega de mensajería. MQTT.org (2019) consultado el 26 de enero de 2020

Encabezado fijo (Fixed header), presente en todos los paquetes de control
Encabezado variable (Variable header), presente en algunos paquetes de control
Información (PayLoad), presente en algunos paquetes de control MQTT

BIT	7	6	5	4	3	2	1	0
BYTE 1	MQTT Control Packet Type				Flags especific to each MQTT Control Packet Type			
BYTE 2	Longitud restante							

Tabla 7 Formato del campo Fixer header dentro del protocolo MQTT

FUENTE: (Noguera, 2016)

En la posición Byte 1 y bits 7-4 como se aprecia en la ilustración cinco, se define el tipo de paquetes de control del protocolo MQTT teniendo su respectiva dirección o flujo de información para sus valores. Como en la posición flag bits tienen contenidos donde se puede utilizar una combinación como identificador de calidad de servicio y también bit reservados para uso futuro. Cabe recalcar que si se recibe flags validas el receptor debe cerrar la conexión de red. Por último, el restante de la longitud es el número de bytes restantes en el paquete actual, incluyendo los datos de la cabecera variable y la carga útil, para su codificación se emplea un esquema de codificación de longitud variable que utiliza un solo byte para valores de hasta 127.

2.2.8.2. Modelo de comunicación de paquetes

La característica principal del protocolo MQTT sigue el modelo cliente-servidor según la topología establecida por referencia estrella. Siendo el nodo central de la comunicación el establecimiento de un nodo central conocido como bróker. El Bróker es un servidor encargado de la transmisión, de los publicadores, o recepción de mensajes con los suscriptores. Además de mandar o recibir mensajes también se tienen tópicos donde los clientes conocidos ahora como suscriptores reciben información sobre el asunto del tópico específico. (Amr, 2018)

Capitulo III: Marco Practico



3.1. Situación actual

3.1.1. Personal de trabajo en San Buenaventura

Normalmente se cosecha una vez al año, usualmente a finales de verano o inicios del otoño en el sector del oriente (Entre febrero y marzo). Los profesionales que usarían la solución consta de operadores de la cosechadora que se encargan de monitorizar en el equipo que tanto los sensores como la misma cosechadora funcionan correctamente. Posteriormente se tiene otro equipo de trabajo que consta del ingeniero agrónomo que, gracias a varios parámetros según el tipo de cosecha, temporada, y parámetros naturales debe realizar un filtrado de los datos para poder tener los puntos necesarios para poder trabajar el mapa de cosecha. Por ejemplo, al ingeniero agrónomo no le interesan los datos recolectados cuando la cosechadora dio un giro a otra fila para seguir cosechando o si esquivó algún obstáculo en la línea de cosecha.

3.1.2. Datos a transmitir

En primera instancia hay que analizar la carga útil que se desea analizar.

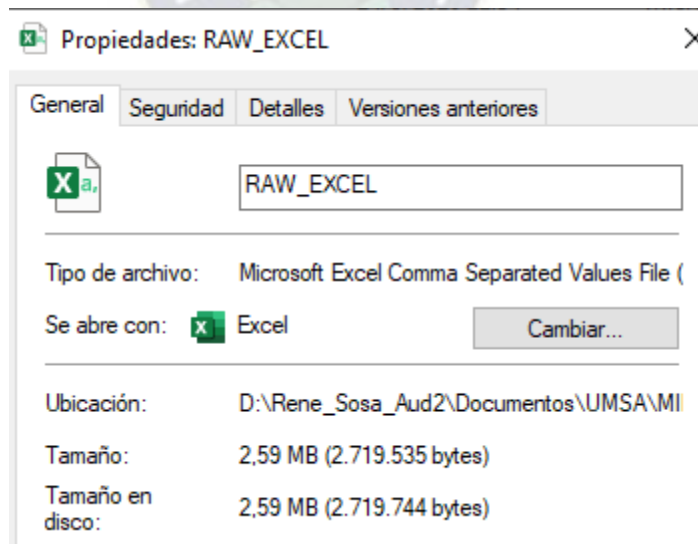


Ilustración 6: Captura de pantalla sobre el peso que tiene el archivo de muestra sobre tomas de datos en campo

FUENTE: Elaboración propia

Como se puede evidenciar en la ilustración 6 el propietario de los datos considera como la carga útil a analizar es un archivo .csv que tiene un peso de 2.59 MB que son aproximadamente 31264 muestras de aproximadamente 27 hectáreas de 1 sola cosechadora. Sin embargo, si solo se tiene un solo registro conocido como *data row*, entonces el peso del archivo .csv cambia a solo 218 B. Esta diferenciación es clave para poder seleccionar que tipo de suscripción del protocolo de capa 7 del modelo OSI se requiere.

3.1.3. Detalles del terreno

Aunque el objetivo del presente documento es plantear un sistema de comunicación basado en IoT para la generación de mapas de cosecha. Es necesario tomar el ejemplo del terreno donde se tomó la muestra de los sensores. Para tal motivo, se analizó el perfil del terreno:

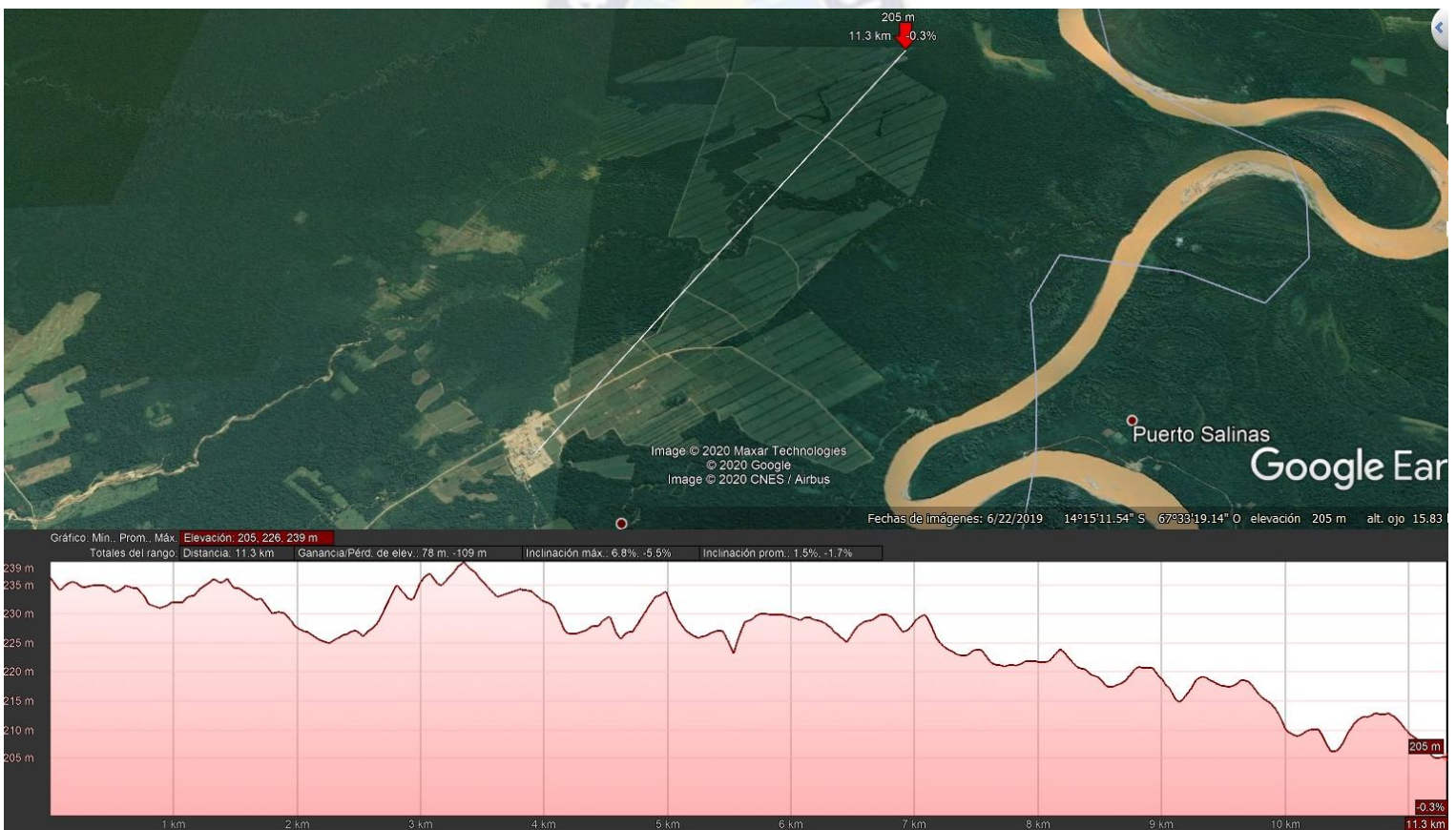


Ilustración 7: Perfil de terreno desde las oficinas del ingenio azucarero en san buenaventura y la última parcela ya cosechada.

FUENTE: Google Earth pro, 2020

Las oficinas de San Buenaventura se encuentran a una altura de 236 m (sin contar los edificios de oficinas) y la elevación más considerable del terreno es de 239 m. Cabe mencionar que el Gateway se instalará en el cuarto piso de las oficinas, por tanto, a 236 metros que es la elevación del terreno hay que sumar 10 metros más para poder sugerir la altura de la antena para cobertura, en otras palabras, la altura mínima para el Gateway es de 246 m.

3.2. Sensores de la cosechadora

Los diferentes modelos de cosechadoras, según sus fabricantes pueden incorporar diferentes equipos sensores y/o capturadores de datos en sus estaciones de trabajo; sin embargo, para los componentes mínimos para realizar un estudio para la generación de mapas de cosecha se centran en el siguiente diagrama.

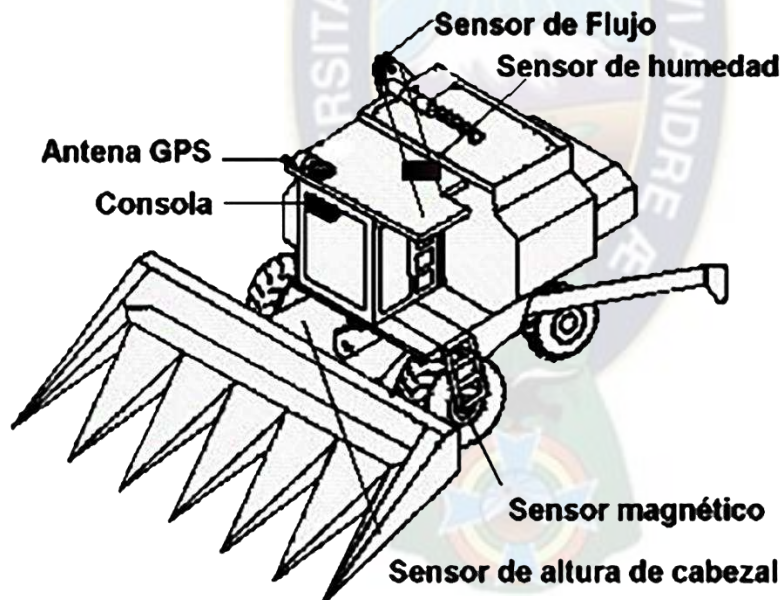


Ilustración 8 Representación esquemática de componentes del monitor de rendimiento

Fuente: (Bragachini, Mendez, & Scaramuzza, 2019)

Los primeros sensores que comienzan a capturar la cosecha son los sensores de impacto o los sensores ópticos.

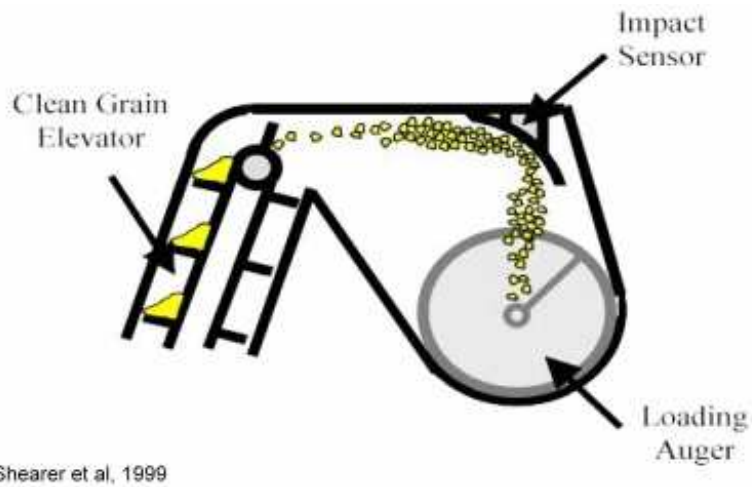


Ilustración 9 Esquema genérico de la posición de los sensores de grano.

FUENTE: (Nielsen, 2020)

Como se puede apreciar en la ilustración número seis, los sensores convierten las señales eléctricas recibidas de los mismos en información, capturándose gracias al impacto del grano, en el caso del sensor de impacto, o en la cinta transportadora de grano limpio de la cosechadora midiendo la cantidad del haz de luz luego desde el componente transmisor. En estimaciones de flujo de grano por segundo en la mayoría de los casos.



Ilustración 10 Receptor equipado con doble procesador de señales GPS y Glonass con compensación del terreno en los tres ejes.

FUENTE: (Jhon Deere, 2020)

Apreciando la ilustración número siete las estimaciones van de la mano con las distancias recorridas basadas del GPS y corregidas diferencialmente. Además, un componente esencial para el proyecto es el sensor de humedad montado sea en el contenedor de grano o en la etapa de transporte luego del cabezal de la cosechadora y el sensor de flujo de grano.

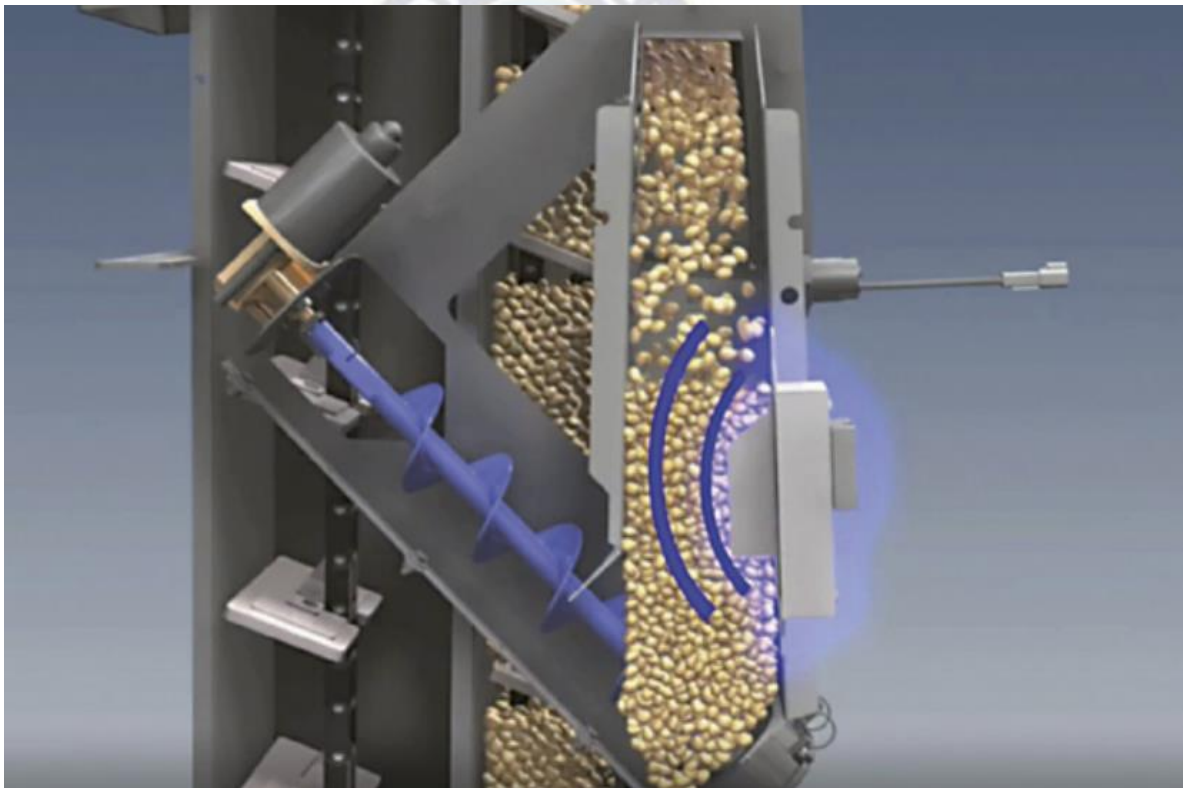


Ilustración 11 Sensor de humedad montado en la etapa de transporte de grano limpio.

FUENTE: (Bartolomé Cerutti SRL, 2020)

Luego de evidenciar en la ilustración número ocho como se monta dicho sensor de humedad, se puede capturar el valor de la humedad del grano tomando dicho valor según captura del sistema. Por otra parte, hay valores que ya vienen definidos por el fabricante como el ancho del

cabezal, si el cabezal se encuentra trabajando; también, valores que se pueden calcular como la trayectoria según los puntos de GPS y distancia recorrida.

El cabezal de la cosechadora es uno de los datos más importantes de captura, igualando así la importancia de los datos que se capturan del grano, ya que cuando un maquinista termina una trayectoria recta, no es necesario que suba el cabezal para ir cosechando en una nueva línea recta, cabe mencionar que para trasladarse a esa nueva línea debe realizar un giro en U usualmente, en estas trayectorias los datos capturados no son de importancia para la generación de mapas de cosecha ya que no contiene información de cosecha, solo de trayectoria (siendo estos puntos de interés para el técnico especialista ya que el archivo que consulta en el servidor cliente tiene que eliminar estos puntos, “limpiando” así el informe).



Ilustración 12 Ejemplo de cabeza de cosechadora encendida y apagada

FUENTE: (Vicente, 2020)

Con los datos ya mencionados ya se puede proceder a generar una estimación del rendimiento de grano seco por hectárea, según su contenido de humedad, según la elección del especialista, por punto de ubicación geográfica de la parcela.

3.3. Datos recolectados por los sensores

Actualmente los datos enviados por las cosechadoras tienen una estructura de datos ya definida en cada archivo .csv que se exporta los cuales cada columna separada por comas tiene los campos de:

- Longitud (LAT)
- Latitud (LONG)

- FLOW (VARIABLE DE CAUDAL DE COSECHA)
- GPS TIME (CAMPO DE TIEMPO EN SEGUNDOS, FRECUENCIA DE LECTURA)
- LOGGED INT (CAMPO DE REMARCADO DE 1 SEGUNDO DEL MOMENTO DE CAPTURA)
- DISTANCE I (DISTANCIA DE RECORRIDO DE LA MAQUINA)
- SWATH INCH (ANCHO DE LA COSECHADORA)
- MOISTURE (HUMEDAD DE GRANO)
- HEAD STA (1 ENCENDIDO Y 0 APAGADO, ES LA CABEZA CORTADORA DE LA COSECHADORA)
- PASS NUMBER (NUMERO DE TRAYECTORIA DE COSECHA)
- COMBINE (ID DE COSECHADORA)
- FIELD (ID DE CAMPO)
- CONSTANT (ID ADICIONAL DE CAMPO)
- CROP (CULTIVO)
- HYBRID (VARIEDAD DE CULTIVO)

3.4. Propuesta de solución.

Debido a que en las cosechadoras no se cuenta con una fuente de poder constante; en otras palabras, hay que evitar que la batería a bordo se descargue al momento de ingresar. Como serán estaciones que deben transportar cadenas de texto de manera remota la mejor solución es implementar una Low Power Wide Area Network.

El uso de LPWAN es debido a que como se necesita transmitir cadenas de texto no es necesario una tasa de datos elevada al momento de la transmisión.

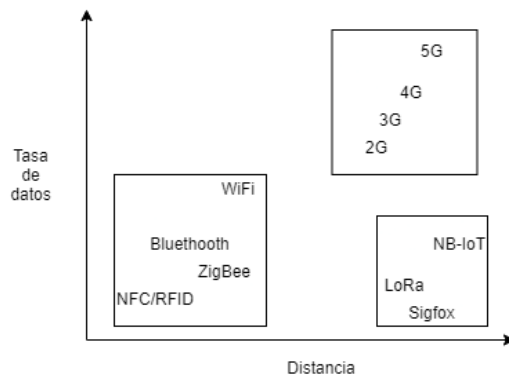


Ilustración 13 Grafica representativa de comunicación inalámbrica según la tasa de datos vs el alcance de la señal.

FUENTE: Elaboración propia

Según la ilustración diez, es de vital importancia tener gran alcance respecto a la central de información, además de crear una red de área local entre los sensores y el servidor, es por tal motivo que como no es necesario utilizar la red celular que tenga cobertura en el sitio y no se está contratando soluciones propietarias como Sigfox, se utilizará LoRa, los parámetros técnicos para llegar a esta determinación se explicarán más a detalle en el documento; sin embargo, el criterio de selección fue el alcance de un enlace con respecto al alcance en kilómetros entre el Gateway y el nodo en la cosechadora y el reducido ancho de banda necesario para la transmisión de datos.

3.5. Parámetros de transmisión

Revisando la ilustración número catorce se puede evidenciar todas las tecnologías disponibles para la transmisión de datos a bajo consumo eléctrico. Un parámetro importante es la distancia de comunicación entre emisor y receptor. Otro es tener la disposición la red el momento que deseen, vale decir que no deban depender de redes de terceros, o alquilar infraestructura para la comunicación de dispositivos. Más aun, la frecuencia de trabajo deber ser no licenciada para evitar el pago de licencia y procurar que en la mayoría de sembradíos no sea afectada por redes que estén usando el mismo canal de comunicación. Es por tal motivo que se descarta usar las redes tanto de Bluetooth, WiFi, ZigBee y NFC; además de las redes móviles de telefonía celular. Quedando así con NB-IoT, LoRa y Sigfox. Sigfox queda descartada por ser una solución

privada y no proveen equipos para su experimentación para desarrollos alternos a la empresa. NB-IoT trabaja en las frecuencias de telefonía móvil, lo cual es contraproducente para el proyecto. Dejando así la solución de LoRa para el sistema de comunicación.

3.5.1. Frecuencia de trabajo.

Tanto para el nodo final montado en la cosechadora como el Gateway de LoRa, se tiene que especificar que rango de frecuencias se va a utilizar. Según el plan nacional de frecuencias de Bolivia, se usará el rango conocido como la banda FCC US902-928, siendo esta la norma americana como banda de uso libre.

Gracias al Decreto Supremo 4272 de Reactivación Intensiva del Empleo del 23 de Junio de 2020, En los numerales dos y tres indica sobre el registro ante la autoridad de Regulación y Fiscalización de Telecomunicaciones y Transporte sin derecho a exclusividad sobre las frecuencias libres. Dando a entender que frecuencias libres están declaradas en el plan nacional de frecuencias de Bolivia en las secciones BOL 20 y BOL 34.

3.5.2. Nodo final y Gateway

Los componentes mencionados fueron seleccionados por la disponibilidad en el mercado internacional para importación desde portales de compras por Internet. El transceiver a utilizar es el Semtech SX1262 ubicado en el componente LoRa Hat para Raspberry Pi ya que trabaja con cobertura de frecuencias globales estándar como nodo final y el Gateway es el RAK831. Su hoja de especificaciones se adjuntará al anexo 1 del documento.



Ilustración 14 Hardware de nodo Gateway RAK831 y nodo final LoraHatSx1662

FUENTE: Elaboración propia

Con estos equipos se propone realizar la comunicación entre cosechadora (Nodo LoraHatSx1662) y las oficinas centrales con el GateWay (RAK831) con las siguientes especificaciones:

	SX1268 470M LoRa HAT	SX1268 433M LoRa HAT	SX1262 868M LoRa HAT	SX1262 915M LoRa HAT
RF module	SX1268		SX1262	
Frequency band	410.125 ~ 493.125MHz		850.125 ~ 930.125MHz	
Antenna	470MHz antenna (2dBi)	433MHz antenna (2dBi)	868MHz antenna (2dBi)	915MHz antenna (2dBi)
Radio rate	0.3kbps ~ 62.5kbps (programmable)			
Transmit power	22.0dBm (10/13/17/22dBm programmable)			
Receive sensitivity	-147dBm (radio rate 2.4kbps)			
Power consumption	Transmit current: 110mA (transient consumption) Receive current: 11mA Sleep current: 2uA (LoRa module software shutoff)		Transmit current: 133mA (transient consumption) Receive current: 11mA Sleep current: 2uA (LoRa module software shutoff)	
Communication interface	UART (TTL level)			
Baudrate	1200bps ~ 115200bps (9600bps by default)			
Packet length	240 Byte (32/64/128/240 Byte programmable)			
Cache	1000 Byte			
Experimental range	5km (condition: cloudless and wild, altitude 2.5m, antenna gain 5dBi, radio rate 2.4kbps)			
Power supply	5V			
Logic level	3.3V/5V			
Operating temperature	-40~85°C			
Applications	smart home, industrial sensors, building automation, smart meters, and more other IoT applications			
Applicable region	China	Europe, Asia, Africa	America, Oceania, Asia	
	Note: for reference only, please confirm the local supported LoRa bands before placing order.			

Tabla 8 Especificaciones básicas del módulo LoRa

FUENTE: ebay.com, artículo: Waveshare SX1262 Lora Sombrero Para Raspberry Pi 915MHz de modulación de Espectro ensanchado

Cabe recalcar en la tabla número ocho sobre el dato de *Experimental range*. Aunque nosotros tengamos el enlace más lejano a 10 km, y el dato menciona solo 5 km. Se debe diferenciar que estos estudios de la tabla se realizaron en zonas urbanas, donde LoRa Alliance de igual manera recomienda usar estos dispositivos para enlaces de 5 km; pero si se encuentra en zonas rurales los enlaces pueden llegar a 15 km. (iFroglab, 2020)

3.5.3. Chip rate

Según (Umber, Bounceur, & Laurent, 2017) el valor de chip rate es la equivalencia al ancho de banda descrito en el equipo, vale decir que para uplink se tiene un BW de 125 KHz (estos valores fijos gracias a los equipos sugeridos anteriormente) entonces serán 125000 chips/s y para downlink su BW es de 500 KHz, siendo 500000 chips/s. Más aún, el spreading factor del equipo seleccionado tanto para uplink como para downlink es de 7 (SF= 7) por tanto cada símbolo lleva 7 bits de información siendo 2^7 chips de los equipos seleccionados en el anterior punto.

3.5.4. Simbol rate (Rs)

La tasa de símbolos se puede calcular con el cociente del ancho de banda entre la potencia en base dos del SF.

$$R_s = \frac{BW}{2^{SF}}$$

Ecuación 1

Donde:

BW en Hz

SF = 7

Reemplazando los datos se tiene que para uplink será 977 símbolos/segundo y para downlink son 3907 símbolos/segundo.

3.5.5. Data rate (R_b)

El cálculo de la tasa de datos se puede calcular en bits/segundo con la siguiente fórmula:

$$R_b = SF * \frac{BW}{2^{SF}} * \frac{4}{4+CR}$$

Ecuación 2

Donde:

BW es en Hz

SF=7

CR=1

Para uplink nos da una tasa de datos de 5.5 kbps y para downlink es 21.9 kbps.

3.5.6. Duración de chip (T_C)

La duración de cada chip de transmisión es la inversa del ancho de banda (Hz) cuyo resultado está en segundos. Por tanto, para uplink la duración de chip es $8\mu s$ y para downlink es $2\mu s$.

3.5.7. Duración de símbolo (T_S)

La duración de cada símbolo se ve expresada por la siguiente fórmula:

$$T_S = \frac{2^{SF}}{BW}$$

Ecuación 3

Donde:

$$SF = 7$$

$$BW \text{ uplink} = 125 \text{ KHz y downlink} = 500 \text{ KHz}$$

Por tanto, para uplink la duración de símbolo es de 1.024 ms y para downlink 256 ms . Cabe recalcar que ya se puede comprobar que a mayor ancho de banda menor será la duración de símbolo. Si se da el caso de requerir un enlace de mayor alcance, se debe aumentar el SF.

3.5.8. Time on Air (ToS)

LoRa Alliance recomienda utilizar la plataforma loratool.nl para realizar el cálculo de ToA. Es por tal motivo que se tuvo los siguientes resultados:

LoRa Modem settings

Spreading factor	<input type="text" value="7"/>	7 - 12
Bandwidth	<input type="text" value="125"/> kHz	125 kHz default for LoRaWAN. 250 kHz also supported.
Code rate	<input type="text" value="1"/>	4 / (CR + 4) = 4/5. 4/5 default for LoRaWAN

Frame configuration

Payload length	<input type="text" value="218"/> bytes	
Preamble length	<input type="text" value="8"/> symbols	Default for frame = 8, beacon = 10.
Explicit header	<input checked="" type="checkbox"/> Yes	Default on for LoRaWAN
CRC	<input checked="" type="checkbox"/> Yes	Default on for LoRaWAN

Low data rate optimization	No	Enabled for bandwidth 125 kHz and Spreading factor >= 11
Preamble length	12.54 ms	
Symbol length	1.02 ms	
Symbols in frame	323	
Time on air	343.30 ms	

Ilustración 15: Calculo de time on air para BW 125 KHz

FUENTE: <https://www.loratools.nl/#/airtime>



LoRa Modem settings		
Spreading factor	<input type="text" value="7"/>	7 - 12
Bandwidth	<input type="text" value="500"/> kHz	125 kHz default for LoRaWAN. 250 kHz also supported.
Code rate	<input type="text" value="1"/>	$4 / (CR + 4) = 4/5$. 4/5 default for LoRaWAN
Frame configuration		
Payload length	<input type="text" value="218"/> bytes	
Preamble length	<input type="text" value="8"/> symbols	Default for frame = 8, beacon = 10.
Explicit header	<input checked="" type="checkbox"/> Yes	Default on for LoRaWAN
CRC	<input checked="" type="checkbox"/> Yes	Default on for LoRaWAN
Low data rate optimization	<input type="text" value="No"/>	Enabled for bandwidth 125 kHz and Spreading factor ≥ 11
Preamble length	3.14 ms	
Symbol length	0.26 ms	
Symbols in frame	323	
Time on air	85.82 ms	

Ilustración 16: Calculo de time on air para BW 500 KHz

FUENTE: <https://www.loratoools.nl/#/airtime>

Tanto en la ilustración trece como en la catorce se puede evidencias tiempos de uplink de 343.3 ms y downlink de 85.82 ms.

3.5.9. Métodos de activación y seguridad

La información de especificación se basa en el documento de LoRaWAN 1.0.2 del grupo de trabajo lora-alliance.org, debido a que la librería MCCI Arduino LMIC solamente fue probado para redes LoRAWAN 1.0.2, y tiene varios parámetros no implementados de la versión 1.1. Esto es muy importante para futuras pruebas y prototipos.

Un dispositivo final debe ser activado para poder realizar intercambio de paquetes entre el servidor de la red; en otras palabras, este proceso se realiza al inicio de la comunicación entre dispositivo y gateway. Existen 2 métodos para los mencionado. Se recomienda utilizar OTAA (el

otro método conocido como ABP ignora las solicitudes de *join-request* y *join-accept* ya que no se almacena DevEUI, AppEUI y AppKey), debido a que en el nodo final se tiene que almacenar los parametros de DevEUI, AppEUI y AppKey, y en el servidor de red debe estar registrada la misma AppKey para asegurar la integridad de los mensajes. DevEUI es similar a una dirección MAC del equipo para ser reconocido en la red. AppEUI identifica al servidor de aplicación similar a un numero de puerto convencional. Ambos con una longitud de 64 bits como *extended unique identifier*. La AppKey es una llave de 128 bits simétrica para generar el MIC que se describirá más adelante.

Al momento de inicializar el nodo final se genera un mensaje que contiene DevNonce y transporta la AppEUI y el DevEUI, conformando así la trama de *join-request* y adicionando un código de integridad de mensaje MIC. Cabe recalcar que DevNonce, AppEUI y DevEUI no son encriptados por la AppKey. Luego que el servidor de la red reciba el mensaje de *join-request*, este revisa si este nodo final no se utilizó anteriormente autenticándolo con su MIC. Si la autenticación fue un éxito se generan los valores de DevAddr que es una dirección interna entre el nodo final y el servidor de la red de 32 bits para reducir la cabecera de transmisión, AppNonce que es un numero generado randomicamente y el identificador de red. Estos parametros será parte del mensaje de respuesta en conjunto con parametros de la red a la que acaba de autenticarse el nodo final, donde su propio MIC es generado con la AppKey conformando así el mensaje *join-accept*. Finalmente, tanto nodo final como servidor de red generan llaves de sesión conocidas como *Network Sesión Key* (NwkSKey) y *Application Session Key* (AppSKey), enviando así al servidor de aplicación la AppSKey y el DevAddr como se puede observar en la siguiente ilustración.

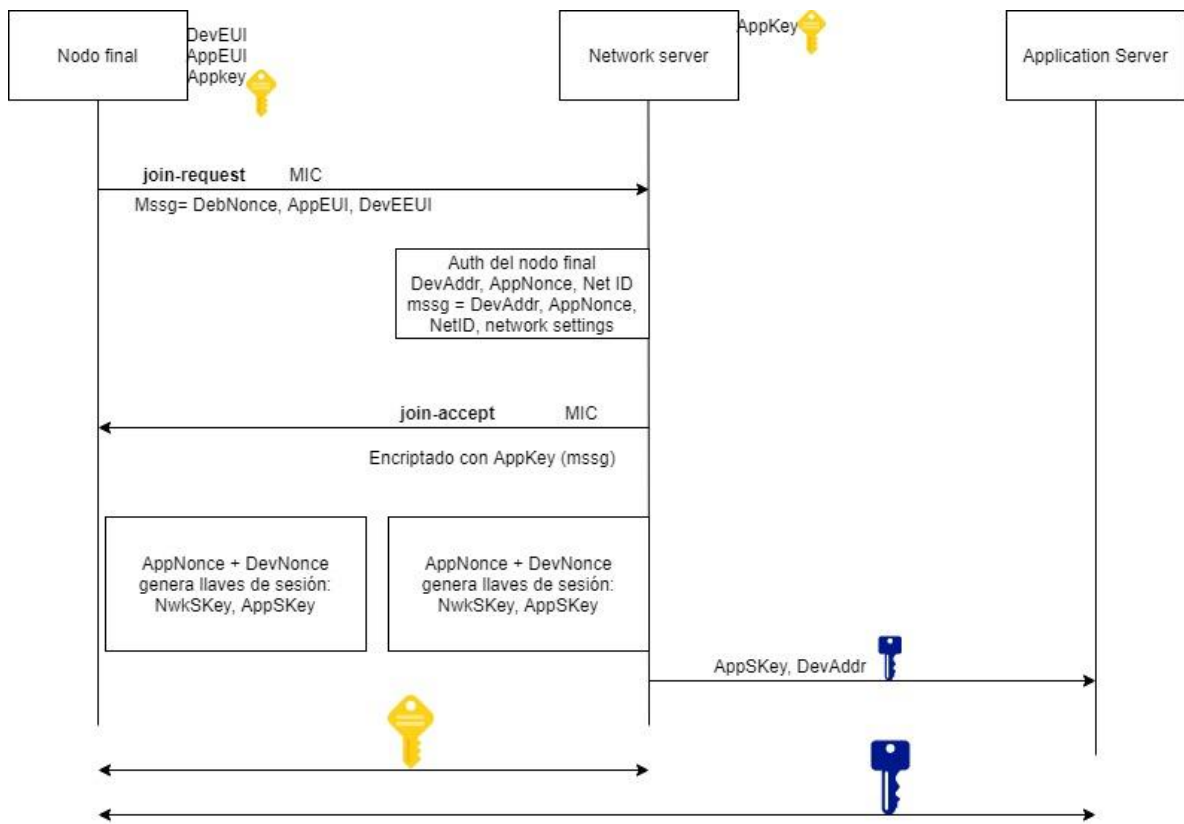


Ilustración 17: diagrama OTAA LoRaWAN

FUENTE: Elaboración propia.

Más aun, LoRaWAN aporta con 2 capas de seguridad, en la capa de red se asegura la integridad del mensaje gracias al MIC usando NwkSKey teniendo así el payload encriptado desde el dispositivo al servidor de red. En la capa de aplicación el payload es encriptado usando AppSKey, asegurando la encriptación desde el nodo final hasta el servidor de aplicación. Los mecanismos de calidad de servicio respecto al momento de enviar una trama y por condiciones externas no favorables se pueda enviar una trama se detallan están controlados por protocolos de capa de aplicación, ya que en este punto se está inicializando la comunicación para poder identificarse con el AppEUI con el servidor de aplicación.

3.5.10. Link Budget

La distancia máxima hasta la fecha son 11 km entre la central de San Buenaventura y el último sector de parcela, sea alquilada a los comunarios o propiedad del estado. Debido a la selección de la frecuencia, se tiene que basar en el estudio realizado por Okumura Hata, ya que incluye medios urbanos y rurales. El modelo Hata no tiene correcciones para distintos tipos de terreno. Según (Zalles, 2019) en su texto sobre redes móviles, explica que la expresión matemática de Hata se representa de la siguiente manera:

$$Lp_{(urbano)} = 69.55 + 23.16 \log f - 13.82 \log h_b - a(hr) + (44.9 - 6.55 \log h_b) \log D$$

Ecuación 4

Donde:

F: frecuencia de operación en MHz

h_b : altura efectiva de la antena de la estación base

hr: altura de la antena de la estación móvil en m

D: distancia del transmisor

a(hr): factor de corrección de altura efectiva de la antena móvil

$Lp_{(urbano)}$: Perdida promedio del trayecto de propagación

Para obtener los datos de altura de antena se hizo una simulación con el software Linkplanner teniendo el peor caso posible en el enlace que es el de que la cosechadora se encuentre en el punto más lejano.

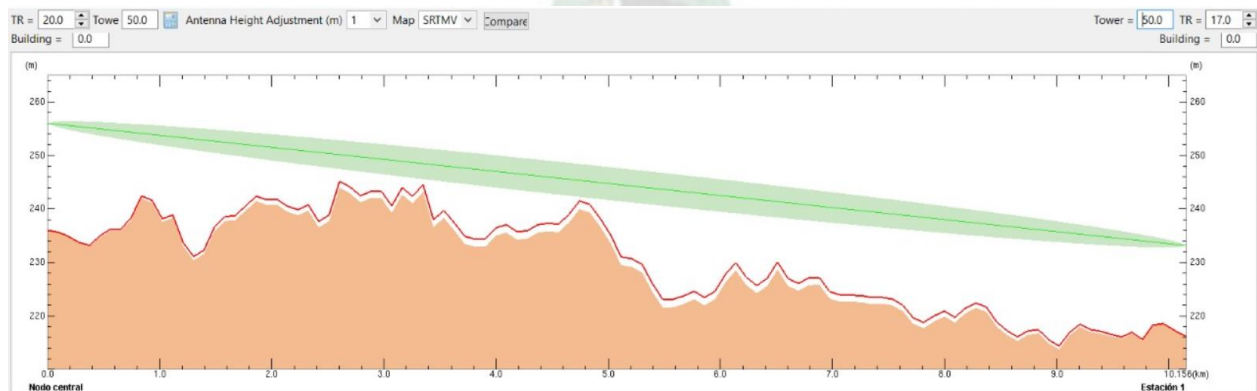


Ilustración 18: Nodos de ejemplo de enlaces entre Gateway y nodo final

FUENTE: Elaboración propia, simulación con linkplanner

Realizando una simulación con Linkplanner se utilizó supuestos para poder observar cual es la altura mínima necesaria en el nodo central (donde se instalará el Gateway) y así observar que la primera zona de Fresnel pasa con tranquilidad. Un punto que observar es que se debe corregir que la altura de la estación uno máximo puede llegar hasta 5 metros aproximadamente que es la altura de la cosechadora, es por tal motivo que al reducir altura en la estación 1 se le debe aumentar en el nodo central según la ilustración 16. Es por tal motivo que se tiene como altura del Gateway 275 metros (39 metros con relación al piso con sujeción sugerida por torre arriostrada para asegurar, según ilustración 18 la primera zona de fresnel y la línea de vista) y la altura del nodo 211 metros (5 metros con relación al piso).

La frecuencia de trabajo en Uplink será 903.9 MHz y de downlink será de 923.3 MHz según especificaciones de la empresa TTN. Tomando en cuenta que será el canal 10 SF7BW125 de uplink y el canal 0 SF7BW500 (RX 1). Estos parámetros fueron seleccionados gracias a la recomendación de la empresa The Things Network, donde al tener la disponibilidad del espectro en la zona de cobertura, es mejor utilizar estos canales para que otro desarrollo trabaje en los canales posteriores manteniendo un orden de asignación. (N. Sornin, 2020)

Antes de realizar el cálculo de pérdida de trayectoria se necesita calcular el coeficiente a para ciudades pequeñas y medianas:

$$a(hr) = (1.1 \log f - 0.7)h_r - (1.56 \log f - 0.8) \quad \text{Ecuación 5}$$

Reemplazando datos:

Upstream

$$a(hr) = (1.1 \log(903.9) - 0.7) * 5 - (1.56 \log(903.9) - 0.8) = 8.947$$

Downstream

$$a(hr) = (1.1 \log(923.3) - 0.7) * 5 - (1.56 \log(923.3) - 0.8) = 8.984$$

Reemplazando valores se tiene en la fórmula de Hata se tiene:

Upstream

$$\begin{aligned} L_{p(\text{urbano})} &= 69.55 + 23.16 \log(903.9) - 13.82 \log(39) - 8.947 + (44.9 - 6.55 \log(39)) \log 11 \\ &= 142.984 \text{ dB} \end{aligned}$$

Downstream

$$\begin{aligned} L_{p(\text{urbano})} &= 69.55 + 23.16 \log(923.3) - 13.82 \log(39) - 8.984 + (44.9 - 6.55 \log(39)) \log 11 \\ &= 143.161 \text{ dB} \end{aligned}$$

Por ultimo para ajustar el enlace por en área rural se tiene:

$$L_{p(\text{rural})} = L_{p(\text{rubano})} - 4.78(\log f)^2 + 18.33 \log f - 40.94 \text{ Ecuación 6}$$

Reemplazando valores:

Upstream

$$L_{p(\text{rural})} = 142.984 - 4.78(\log(903.9))^2 + 18.33 \log(903.9) - 40.94 = 114.459 \text{ dB}$$

Downstream

$$L_{p(\text{rural})} = 143.161 - 4.78(\log(923.3))^2 + 18.33 \log(923.3) - 40.94 = 114.543 \text{ dB}$$

3.5.10.1. Balance de potencia

Con los datos calculados en el punto 3.4.3 se puede realizar un balance de potencias y perdidas entre el Gateway y el nodo final. En resumen, se tienen los siguientes datos para balance de potencias y perdidas:

N°	ITEM	UNIDAD	CENTRAL	ESTACIÓN 1
1	NOMBRE DE LA ESTACIÓN		SB-C1	COSECHADORA 1
2	LATITUD	°	-17,3378	-14,249739
3	LONGITUD	°	-64,6126	-64,586122
4	FRECUENCIA DE OPERACIÓN uplink	MHz	903,9	903,9
5	FRECUENCIA DE OPERACIÓN downlink	MHz	923,3	923,3
6	DISTANCIA DE SALTO	Km	11	11
7	PERDIDA DE LINEA DE TX	dB	5	5
8	PERDIDA EN ESPACIO LIBRE upstream	dB	114,459	
9	PERDIDA EN ESPACIO LIBRE downstream	dB	114,543	
10	PERDIDA TOTAL DEL TRAYECTO upstream	dB	124,459	124,459
11	PERDIDA TOTAL DEL TRAYECTO downstream	dB	124,543	124,543
12	ALTURA DE ANTENA	m	39	5
13	GANANCIA DE ANTENA	dBi	2	2
14	GANANCIA TOTAL	dB	2	2
15	POTENCIA DE TRANSMISIÓN	dBm	22	20
16	SENSIBILIDAD O UMBRAL DE RECEPTOR	dBm	-148	-133

Tabla 9 Balance de potencia y pérdidas entre el Gateway LoRa y un nodo final al extremo del sector cosechado.

FUENTE: Elaboración propia

Con los datos brindados en la tabla número uno, y un ajuste a las alturas de las antenas para que el nodo final sea montado en la parte superior de la cosechadora, ya se puede calcular la potencia de recepción

$$P_{Rx} = P_{Tx} - L_{cableTx} + G_{antenaTx} - L_{os} + G_{antenaRx} - L_{cableRx} \text{ Ecuación 7}$$

Reemplazando datos se tiene:

Upstream

$$P_{Rx} = 22\text{dBm} - 5\text{ dB} + 2\text{dBi} - 114.459\text{dB} + 2\text{dBi} - 5\text{dBi} = -98.459\text{ dBm}$$

Downstream

$$P_{Rx} = 22\text{dBm} - 5\text{ dB} + 2\text{dBi} - 114.543\text{dB} + 2\text{dBi} - 5\text{dBi} = -98.543\text{ dBm}$$

Siendo estos valores bastantes lejanos a la sensibilidad del receptor, garantizando así el enlace de comunicación. Cabe recalcar que este es un caso de cosechadora en el punto más lejano respecto a las oficinas centrales, demostrando que si se encuentra en cualquier otro punto más cercano se asegura de igual manera el enlace.

Es por tal motivo que, dado el caso de ejemplo en la vida real, hablando de caso de la empresa de San Buenaventura, comparando con lo teórico entre las demás tecnologías inalámbricas existentes, debido a rango de cobertura, ancho de banda y eficiencia energética LoRa es la indicada para implementarse como transporte de datos. Para el trato de los datos a nivel de aplicación es necesario realizar el estudio mencionado a continuación.

3.6. Análisis de repositorios protocolos y datos

3.6.1. Criterios de análisis

Cualquier aplicación orientada a transmitir datos debe analizarse con varios criterios y deben ser comparados con las tecnologías existentes. Los parámetros a analizar en los protocolos MQTT, XMPP y HTTP serán:

- RESTful,
- Protocolo de transporte
- Modelo pub-sub,
- Modelo solicitud/respuesta
- Seguridad
- QoS
- Tamaño de cabecera
- Soporte XML
- Eficiencia energética.

3.6.2. Restful

Tomando en cuenta la definición de REST como estilo de arquitectura para el diseño de aplicaciones con redes basándose en protocolos cliente-servidor (por excelencia HTTP) tratando objetos como recursos que pueden ser creados o destruidos

HTTP:

Al tratarse de un protocolo enfocado al modelo cliente – servidor, también conocido por comenzar una solicitud por su *Request* y espera un *Reply* con la información solicitada. Cada vez que se solicita algo desde una página web se usa solicitudes que son consideradas Restful.

MQTT:

Para ser considerado como Restful se necesita crear, limpiar, borrar o adicionar registros mediante respuestas de requerimientos GET o POST. MQTT con la característica de trabajar a bajos niveles de energía es necesario utilizar otros métodos entre publish y subscribe, con el deseo de enviar mensajes en lugar de sondear el servidor en búsqueda de nuevos mensajes.

XMPP:

De igual manera que MQTT, no se lo considera Restful ya que usa el mismo modelo PUB/SUB y no el modelo Request/Response.

3.6.3. Protocolo de transporte

Sea o no que el servicio utilice Internet, es necesario asegurar la integridad de los datos que se envían y reciben a través de toda la red, es por tal motivo que en capa 4 del modelo OSI tanto MQTT, XMPP y HTTP harán uso del protocolo de control de transmisión conocido también como TCP. El transporte de datos ya fue abordado anteriormente con la solución LoRa

3.6.4. Modelo Pub-Sub y modelo Request-Response

El modelo request response, como su nombre indica el cliente solicita al servidor algo, por ejemplo, una petición GET, al momento de realizarla el cliente entra en modo espera y se bloquea para esperar la respuesta (este bloqueo no es al dispositivo en su totalidad ya que puede estar haciendo otras tareas en paralelo) hasta que el servidor manda el contenido solicitado. En otras palabras, el cliente siempre inicializa la solicitud. Pero tiene una problemática, si se tienen servicios anidados en serie, se crea una cadena de espera, hasta que termine de procesarlos para recién liberar los tanto los servicios anteriores como al cliente. Es por tal motivo que se implementaron los brokers, donde se utilizan colas conocidas como tópicos o canales y aunque también el cliente se puede bloquear, el primer servicio anidado publica los datos como tópico y ahí terminaría el proceso para el cliente siendo desbloqueado y los demás servicios consumen del tópico seleccionado para cuya respuesta sea respuesta sea también publicada y el siguiente servicio pueda comenzar a trabajar liberando al primer servicio, y así sucesivamente en todo el proceso anidado en serie. Es por tal motivo que es mucho más eficiente al momento de estar publicando todas las mediciones de los sensores de las cosechadoras y se queden en cola para que el servidor pueda consumir del tópico de cada cosechadora e ir armando el documento final para el mapa de cosecha.

El modelo request/response y las colas de mensajes (pub-sub) tiene sus facilidades para cierto tipo de aplicaciones. Request/response tiene como característica ser simple si se tiene 2 servicios que deban comunicarse entre sí, y aunque no necesariamente sea solicitud respuesta como se haría en una base de datos ya que es un modelo no orientado al estado trabaja muy bien con HTTP. Por otra parte, en contra se aprecia que si se tiene múltiples receptores o consumidores se van creando varios puntos de falla, dado a timeout si comenzamos a saturar a los servicios o errores de mayor gravedad como que un web service perdió la comunicación y compromete a todo el servicio. Más aún, es necesario que cliente y servidor deben estar activos para enviar y recibir información, siendo una desventaja si se quiere enviar información que no será capturada al mismo tiempo o las 24 horas del día (realizando la importancia de las colas).

Las colas de mensajes como aspectos positivos que puede escalar con múltiples suscriptores y estos pueden ser diferentes equipos con diferentes capacidades y recursos; más aun

convirtiéndolo como la mejor opción para microservicios y por sobre todo puede trabajar con clientes que no se encuentran activos porque los mensajes se encuentran almacenados en la cola del tópic. Sin embargo, se pueden evidenciar problemas al momento de entregar mensajes, al momento de publicar algo no se puede saber si evidentemente se publicó, el problema no radica que si no se recibe un ACK, como nos encontramos con TCP eso significaría un RETRY, pero hay la problemática de que es probable que se publique un mensaje dos veces, y aunque hay métodos como la de la idempotencia para resolver esto, de igual manera es un aspecto a tener en cuenta; de igual manera, siendo un suscriptor no se tiene la certeza que ya leyó el mensaje o no que fue publicado. En resumen, MQTT y XMPP utilizan el modelo pub-sub pero no HTTP (este último protocolo no usa el modelo porque va en contra de ya que este protocolo utiliza la arquitectura REST). Por otra parte, XMPP y HTTP usan el modelo request-response pero no MQTT.

3.6.5. Seguridad

Sea MQTT, HTTP y XMPP usan SSL, es necesario crear una conexión segura entre el cliente y el servidor si hablamos generalmente de HTTP o del cliente al bróker si estamos hablando de MQTT y XMPP si publicamos el servicio a la red externa. Estos protocolos son independientes del protocolo SSL/TLS para comunicaciones cifradas, así que se los debe implementar finalizando la solución.

3.6.6. Calidad de Servicio (QoS)

Para HTTP no hay un mecanismo definido exactamente para dicho protocolo; más aun, aunque se pueda usar marcadores para tráfico, no hay un marcado específico para HTTP. En MQTT existen 3 niveles de QoS que se puede programar conocidos como 0 at most once, 1 at least once y 2 exactly once. Donde el nivel 0 indica que al enviar el mensaje no se necesita reenviar en caso, nivel 1 y nivel 2 se encargan de garantizar la transmisión y recepción de los

mensajes. Y por último XQMPP integrado en su programación no cuenta con mecanismos de QoS y más aún, los mecanismos de QoS se quedaron en borrador ya que no fueron aceptados hasta la fecha, por tanto, no se implementan desde el 2015.

3.6.7. Tamaño de cabecera

Tanto HTTP como XMPP no cuentan con datos específicos de cabecera, esto es gracias a diferentes razones. HTTP para envío de documentos no necesita de cabeceras especiales o alguna etiqueta en específico, otras capas se encargan de enviar este payload. En XMPP por estar basado en etiquetas ya contiene en su payload la información necesaria de destino y origen del payload; en otras palabras, la entidad inicial debe abrir una conexión TCP con la entidad receptora para luego negociar con streams XML entre sí. Por último, MQTT si dos tipos de cabeceras, uno de longitud fija (2 Bytes) y otra opcional de longitud variable.

3.6.8. Soporte XML

El XML es una adaptación del SGML (Standard Generalized Markup Language), un lenguaje que permite la organización y el etiquetado de documentos. Esto quiere decir que el XML no es un lenguaje en sí mismo, sino un sistema que permite definir lenguajes de acuerdo a las necesidades. El XHTML, el MathML y el SVG son algunos de los lenguajes que el XML tiene la capacidad de definir. Las bases de datos, los documentos de texto, las hojas de cálculo y las páginas web son algunos de los campos de aplicación del XML. El metalenguaje aparece como un estándar que estructura el intercambio de información entre las diferentes plataformas.

Tanto HTTP por sus diversas aplicaciones como por excelencia XMPP soportan XML; sin embargo, por su contraparte MQTT no requiere el uso de etiquetas para el envío de información para publicar o consumir de un tópico. Cabe recalcar que el soporte XML no es un análisis conclusivo de optimización ya que se aconseja en lugar de utilizar etiquetas xml usar JSON o en su defecto Google protocol Buffer.

3.6.9. Eficiencia energética

No existe un estudio específico disponible al público sobre en qué entorno y pruebas se realizaron para el consumo de energía usando XMPP. Sin embargo, si existe la comparación de consumo de energía tanto con redes móviles 3G como con Wifi entre MQTT y HTTP mostrando que MQTT es más eficiente al momento de analizar el consumo de energía eléctrica.

Gracias a (Nicholas, 2020) en su estudio donde compara los protocolos MQTT y HTTP, en su estudio denota a C2DM, sin embargo no aporta al proyecto mencionarlo. Se realizó una aplicación móvil push notification con diferentes características según especifica el protocolo.

Por el lado de MQTT se hizo una aplicación simple usando un *custom wrapper* con el estándar de cliente JAVA MQTT de IBM usando una conexión SSL en ambos lados con un micro boker y certificado mutuo entre cliente y servidor. En cambio, por el lado de HTTP, se desarrolló una aplicación estilo *long polling*, es un intercambio de acusas de recibo para emular un modelo pub-sub. Basicamente el cliente hace un *request* HTTP al servidor, y se mantiene abierta la solicitud hasta que el servidor tiene información para mandar al cliente, cuando esta información llega el servidor la manda al cliente y recién cierra el *request* solicitado; para luego, el cliente pueda iniciar un nuevo *request long polling* para obtener nuevos datos. Las pruebas se realizaron con máquinas de escritorio y un dispositivo móvil HTC one, cabe recalcar que para la aplicación HTTP se tuvo que adjuntar un *read timeout*.

% Battery Used			
3G		Wifi	
HTTPS	MQTT	HTTPS	MQTT
0.02972	0.04563	0.00228	0.00276



Ilustración 19: Figura comparativa de uso de batería entre red móvil y red lan inalámbrica WiFi

FUENTE: (Nicholas, 2020)

3.6.10. Implementación

Queda evidenciado que implementar HTTP en una solución que trabaja con sensores y datos recolectados varias veces en un periodo determinado no es conveniente; más aún, XMPP al no poseer mecanismos de calidad de servicio lo convierte como el menos indicado para una aplicación de transporte de datos de campo que son claves para la generación de mapas de cosecha es por tal motivo que se recomienda la siguiente topología general

3.7. Resumen de análisis de protocolos de aplicación

Criterio	MQTT	XMPP	HTTP
Modelo	Pub/Sub	Pub/Sub - Cliente/Servidor	Cliente /Servidor
Restful	No	No	Si
Transporte	TCP	TCP	TCP
Seguridad	SSL	SSL	SSL
QoS	Si	No	No
Tamaño de cabecera (B)	2	-	-
Soporte XML	No	Si	Si
Eficiencia energética	Optima	-	No

Tabla 10: Criterios de análisis para la selección del protocolo de aplicación

FUENTE: Elaboración propia

Gracias a la tabla número 11 se pueden apreciar dos factores clave para la selección de MQTT como protocolo de aplicación, las respuestas de color rojo no son favorables para el proyecto. Al no presentar eficiencia energética, o estudios que corroboren la información, se puede argumentar que MQTT ahorrará energía en los nodos finales de las cosechadoras sin comprometer la batería del motorizado. Otro factor clave es el no ser Restful, este punto ya fue explicado anteriormente. Sin embargo, para eliminar el empate entre XMPP y MQTT se hizo énfasis en la calidad de servicio. XMPP no garantiza el envío de paquetes a su destinatario,

siendo este un punto en contra para su selección. Es por tal motivo que la propuesta se centra en MQTT.

3.7.1. Topología general

Para implementar la solución hay 2 opciones por optar debido a la solicitud y estado de los equipos sensores para poder capturar la información, en el primer caso se tendrá cuando se puede acceder a cada fila de registro de todos los sensores, vale decir todos los datos en una fila .csv. Y el segundo caso es cuando se puede acceder a cada dato de cada sensor.

3.7.1.1. Primera opción

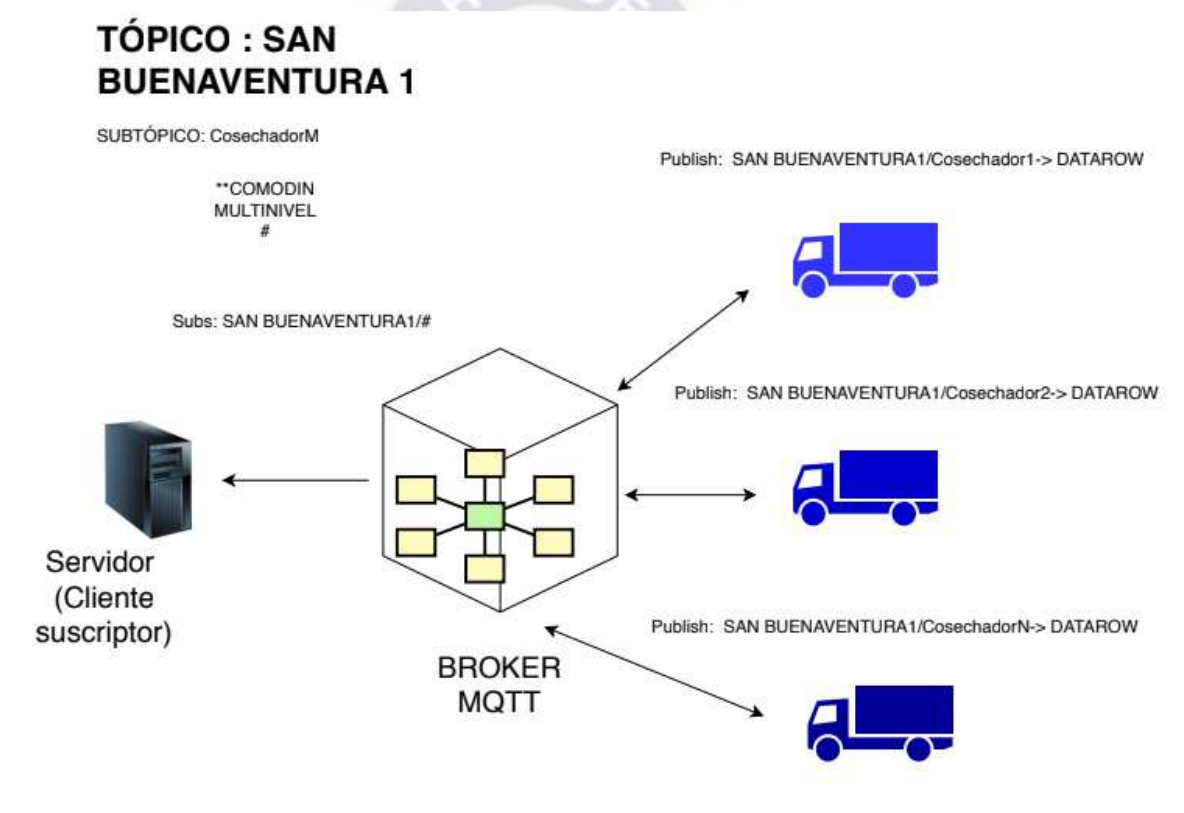


Ilustración 20: Esquema de la topología de comunicación entre cosechadora y servidor

FUENTE: Elaboración propia

Según la Ilustración número cinco, cada cosechadora tiene su subtopico donde irá publicando cada fila de registro de sensores que vaya captando a su propio ritmo para que se recepcione el *DATAROW* de cada cosechadora, se posicione en una cola y el servidor comience a consumir de la pila de registros de cada cosechadora generando un archivo consolidado para que el técnico especialista comience a filtrar los datos.

3.7.1.1.1. El servidor para el consolidado de datos

En primera instancia con las maquinas cosechadoras que se tiene en producción, su monitor de cultivo entrega la misma estructura de datos, cada columna es el dato que debe ser. Entonces la suscripción al bróker MQTT debería ser al sector en específico denominado San Buenaventura1, y haciendo uso del comodín “#” significa que el servidor será suscriptor de cualquier subnivel perteneciente a SAN BUENAVENTURA1; vale decir a cada cosechador que vaya publicando sus datos.

3.7.1.1.2. Los publicadores (cosechadoras)

Teóricamente MQTT soporta hasta 10000 clientes, es por tal motivo que lo único que necesita una nueva cosechadora es registrarse como cliente publicador de SAN BUENAVENTURA1 y podrá comenzar a transmitir la información.

3.7.1.2. Segunda opción

Dado el caso que existen varios fabricantes de diferentes países que no están obligados a mantener la estructura de los datos mencionado anteriormente, las cosechadoras tendrán que crear un nuevo subtópico donde capturará el dato específico del sensor.

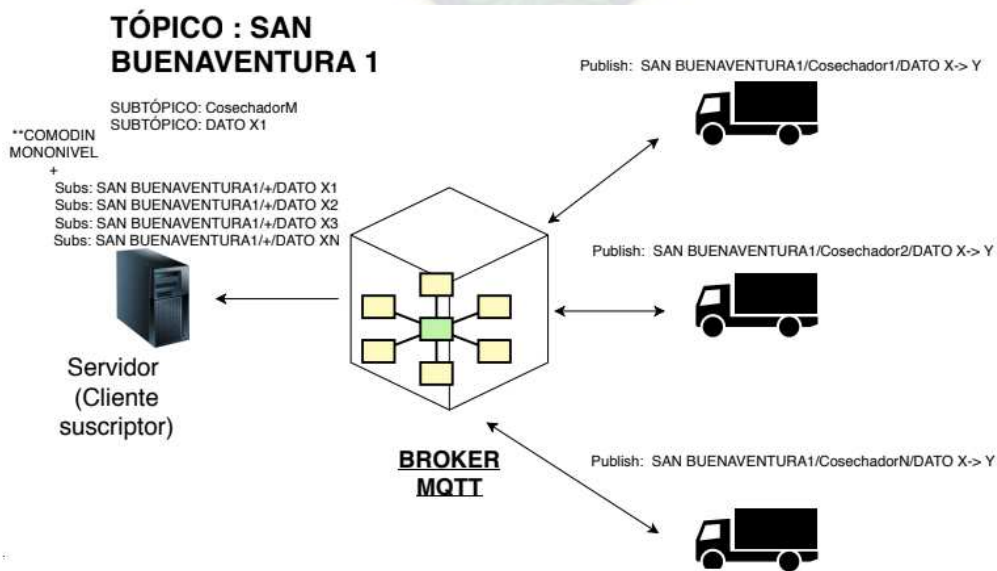


Ilustración 21: Esquema de la topología general usando comodin de mononivel

FUENTE: Elaboración propia

3.7.1.2.1. El cliente suscriptor

Dado a que el único cambio se evidencia en el envío de datos por tópicos, por ejemplo, tener un subtema de longitud, otro de longitud, otro de humedad de grano y sucesivamente. Entonces se debe utilizar un comodín mono nivel; esto es debido a que se tiene como constantes a las cosechadoras, pero se debe captar los datos específicos que van captando los diferentes sensores montados. Debido a la diversidad de datos que se pueden capturar por fabricante de equipos se debe seleccionar lo que desea capturar y lo que se desea ignorar.

3.7.1.2.2. Los publicadores

Se debe hacer un análisis de todos los datos que puede brindar las cosechadoras, cada cosechadora mínimamente debe brindar los datos especificados anteriormente. Si existe el caso de tener datos relevantes no considerados, el técnico especialista de filtrado de datos debe decidir si es conveniente o no crear nuevos subtópicos.

3.8. Análisis de costos

Con costos aproximados a la tabla a continuación se puede llegar a un estimado de inversión de equipos, hay que tomar en cuenta que como puede variar el terreno según el área donde se desea realizar la implementación de esta tecnología no se toma en cuenta los costos de trabajos y materiales de planta externa.

Equipos	Bs.
Nodofinal LoRaHat + Raspberrypi 3b+	890
Gateway LoRa	1047
Procesador Ryzen 5 2400G	1400
Memoria RAM 8GB X2	740
Fuente ATX	700
HDD 2TB	720
Case generico+perifericos	880
Monitor 22"	650
Total	7027

Tabla 11 Precios referenciales de hardware de 1 nodo central y 1 nodo final

FUENTE_ Elaboración propia

Para un cálculo diario del costo de servicio se tiene que analizar que la cosecha aproximadamente dura un mes. Por tanto, se usará 30 días, entonces el precio por día se calcula de la siguiente forma:

$$\text{costo}_{\text{día}} = \frac{7027 \text{ Bs}}{30 \text{ días}} = 234.23 \text{ Bs/día} \quad \text{Ecuación 8}$$

Por tal motivo el costo a recuperar aproximado por día de uso de servicio de todas las cosechadoras es de 234.23 Bs por día. Vale recalcar que no se está tomando el precio de las torres de montaje para las antenas, ya que la investigación se centra en el sector agroindustrial cuyos terrenos pueden variar de sector en sector.



CAPITULO IV: MARCO

CONCLUSIVO



4.1. Conclusiones

Luego de realizar con éxito el estudio de factibilidad de enlace. Se logró identificar que, aunque XMPP pueda brindar un menor payload para la transmisión de información de sensores, la ausencia de mecanismos de QoS lo convierte en el protocolo menos recomendable para transmisión de información sensible, y como el proyecto está orientado al Internet de las cosas y no a la presentación de una página web de presentación de datos, MQTT es el protocolo recomendado para implementar en el sector agrícola. La topología recomendada para levantar el servicio de MQTT en cualquier sector de la agroindustria, no necesariamente en el sector de estudio de ejemplo (San Buenaventura), es la topología en estrella, donde todos los nodos se encontrarán dentro el área de cobertura de nodo central, cuyo protocolo de comunicación es LoRa creando la red LoRaWAN para la comunicación entre *end node* y *gateway*; además que es recomendado gracias al poco consumo de energía eléctrica y se puede implementar con baterías si es necesario o adaptarlo al sistema de energía de la cosechadora en el caso del nodo final, y del gateway con un cargador de celular. Los datos deben ser enviados desde las cosechadoras como una fila de registros con un orden determinado y común entre los nodos finales para poder registrarlos en el tópico respectivo de la cosechadora; caso contrario, se debe crear un registro por cada sensor y dato que se recolecta (Este último reduciría considerablemente el payload pero no afecta tanto en la lógica MQTT como en la transmisión LoRa). Más aún, al ser un modelo Pub/sub, considerado un servicio de mensajería asíncrona, al compilar en un componente centras de infraestructura, asegura que al momento de conectarse se recibirá toda la información necesaria, haciendo uso eficiente del ancho de banda; a comparación del modelo cliente servidor, sea o no usando sockets, para poder recibir la información se debe realizar solicitudes constantes. Es por tanto que no es recomendable enviar los datos inalámbricamente como un documento consolidado de todas las mediciones de la cosechadora al final de la jornada, sino enviar al servidor para almacenarlas y en su defecto el especialista (como usuario final) comience a filtrar los datos que no le interesa y pueda gracias a sus cálculos generar mapas de cosecha optimizando así su tiempo de trabajo y productividad.

4.2. Recomendaciones

A pesar del éxito de la investigación del presente documento, se recomienda realizar pruebas tanto de enlace como de integridad en capa de aplicación del modelo TCP/IP en el territorio boliviano de las ofertas en el mercado priorizando el sector rural preliminarmente para corroborar

que los modelos y teoría de las hojas de especificaciones se aplican sin observaciones y no se deba tomar en cuenta otros factores externos al análisis.

4.3. Glosario de términos, siglas y acrónimos

IoT	La ITU (International Telecommunication Union) y el IERC (Internet of Things European Research Cluster), lo definen como: “una infraestructura de Red Global y dinámica con la auto-configuración de las capacidades basadas en protocolos de comunicación estándar e inter-operables, donde “cosas” físicas y virtuales tienen identidad, atributos físicos, personalidades virtuales y utilizan interfaces inteligentes que se integran a la perfección en la red de información”.
Centro de Datos	Se refiere a un nodo de cálculo y almacenamiento de datos en algunos contextos. Varía en escala entre un centro de datos centralizado, un centro de datos regional y un centro de datos de extremo.
Latencia	En el contexto de la transmisión de datos en red, el tiempo que tarda una unidad de datos (normalmente una trama o un paquete) en viajar desde su dispositivo de origen hasta su destino previsto. Medido en términos de mili-segundos en puntos únicos o repetidos en el tiempo entre dos o más puntos finales.

Orquestación El proceso de determinar dónde y cómo se debe procesar una carga de trabajo de aplicaciones en el gradiente de n-niveles de los recursos de computación, almacenamiento de datos y red proporcionados por el edge. En la mayoría de los casos, la orquestación de la carga de trabajo se realiza mediante un sistema automatizado que tiene en cuenta los requisitos de rendimiento, tiempo y coste de un operador de aplicaciones en ese punto.

IoT	Internet of Things
MQTT	Message Query Telemetry Transport
M2M	Machine 2 (to) Machine
LoRa	Long Range
WiFi	Wireless Fidelity
JSON	Java Script Object Notation
API	Application Pprogramming Interface
HTTP	HTypeperText Transfer Protocol
XML	eXtensible Markup Language
SOAP	Simple Object Access Protocol
IT	Tecnologías de la información
ONG	Organismo no gubernamental
RF	Radio Frecuencia

4.4. Referencias Bibliográficas

Albarenque, S. V. (2011). *Técnicas de procesamiento de mapas de rendimiento*.

Albarenque, S. V. (2011). *Técnicas para el procesamiento de mapas de rendimiento*.

Amr. (2018). Obtenido de <https://1sheeld.com/mqtt-protocol/>

Bartolomé Cerutti SRL. (2020). Obtenido de Detección de la humedad en tiempo real:

<https://bartolomecerutti.com.ar/product/deteccion-de-la-humedad-en-tiempo-real/>

Bragachini, M., Mendez, A., & Scaramuzza, F. (2019). *Proyecto de Agricultura de Presición*.

INTA Manfredi.

Corporation, S. (Abril de 2016). *AN1200.2 LORA MODULATION BASICS*. Obtenido de

<http://www.semtech.com/images/datasheet/an120>

Dongkyun, K. (11 de 2020). *MPDI*. Obtenido de Us national library of medicine:

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5038744/#:~:text=However%2C%20in%20practice%2C%20the%20LoRaWAN,between%2013%20and%2028%20bytes.>

Force, I. E. (23 de September de 2020). *HTTP*. Obtenido de <https://tools.ietf.org/html/rfc7230>

Gonzales, A. (2016). *Redes de computadores Arquitectura Cliente - Servidor*. Universidad

Tecnica Federico Santa Maria.

iFroglab. (septiembre de 2020). *techdesign*. Obtenido de [https://www.techdesign.com/hardware-](https://www.techdesign.com/hardware-solutions/LoRa-Node-iL-LORA1272-Node-Ultra-Remote-15km)

[solutions/LoRa-Node-iL-LORA1272-Node-Ultra-Remote-15km](https://www.techdesign.com/hardware-solutions/LoRa-Node-iL-LORA1272-Node-Ultra-Remote-15km)

Jhon Deere. (2020). Obtenido de [https://www.deere.com.ar/es/agricultura-de-](https://www.deere.com.ar/es/agricultura-de-precisi%C3%B3n/receptores-y-monitores/starfire-6000/)

[precisi%C3%B3n/receptores-y-monitores/starfire-6000/](https://www.deere.com.ar/es/agricultura-de-precisi%C3%B3n/receptores-y-monitores/starfire-6000/)

Junyang, B. (2016). *Communication between industrial network and public network using*

Profine and MQTT protocol. Valencia: Universidad politecnica de España.

Moya, J. M., & Pastor, R. C. (2006). *Sistemas de telefonía*. Madrid, España: Editorial Paraninfo.

N. Sornin, M. L. (Septiembre de 2020). *LoRaWAN Specifications 1.0.2*. Obtenido de www.lora-alliance.org

Nicholas, S. (2020). *Power Profiling: HTTPS Long Polling vs. MQTT with SSL, on Android*.

Obtenido de <https://stephendnicholas.com/posts/power-profiling-mqtt-vs-https>

Nielsen, B. (2020). *Corny News Network*. Obtenido de

<https://www.agry.purdue.edu/ext/corn/news/timeless/yldmoncalibr.html>

Noguera, J. (2016). *Sistema de Diálogo Basado en Mensajería Instantánea para el Control de Dispositivos en el Internet de las Cosas* .

Ordoñez , I. (2017). *Estudio de la arquitectura y el nivel de desarrollo de la red LoRaWAN y de los dispositivos LoRa*. Cataluña: Universitat Oberta de Catalunya.

Sagar, P. I. (2018). *A survey of Messagin protocols for IoT Systems*.

Umber, N., Bounceur, A., & Laurent, C. (2017). A Study of LoRa Low Power and Wide Area Network. *ATSIP'2017*.

Vicente, M. (2020). *Partes de una cosechadora*. Obtenido de

<https://blogsmanuelvicentedelgadocascos.blogspot.com/2014/02/partes-de-una-cosechadora.html>

Zalles, V. (2019). *Telecomunicaciones, satelites y fibra óptica*. La Paz.

ANEXO 1

Key Features

- Compact size 80.0 x 50.0 x 5.0mm
- LoRa™ modulation technology
- Frequency band 433,470, 868,915MHz
- Orthogonal spreading factors
- Sensitivity down to -142.5 dBm
- Maximum link budget 162 dB
- SPI interface
- SX1301 base band processor
- Emulates 49 x LoRa demodulators
- 12 parallel demodulation paths
- 1 (G)FSK demodulator
- 2 x SX1257 Tx/Rx front-ends High frequency
- 2 x SX1255 Tx/Rx front-ends low frequency
- Supply voltage 5 V
- RF interface optimized to 50
- Output power level up to 23 dBm
- GPS receiver (optional)
- Range up to 15 km (Line of Sight)
- Range of several km in urban environment
- Status LEDs
- HAL is available from

https://github.com/RAKWireless/RAK831_LoRaGateway

1.2 Applications

- Smart Metering

- Wireless Star Networks
 - Home, Building, Industrial automation
- Remote Control
- Wireless Sensors
- M2M, IoT
- Wireless Alarm and Security Systems

AK831 Datasheet

2. Module Package

In the following the RAK831 module package is described. This description includes the RAK831 pinout as well as the modules dimensions.

2.1 Pinout Description

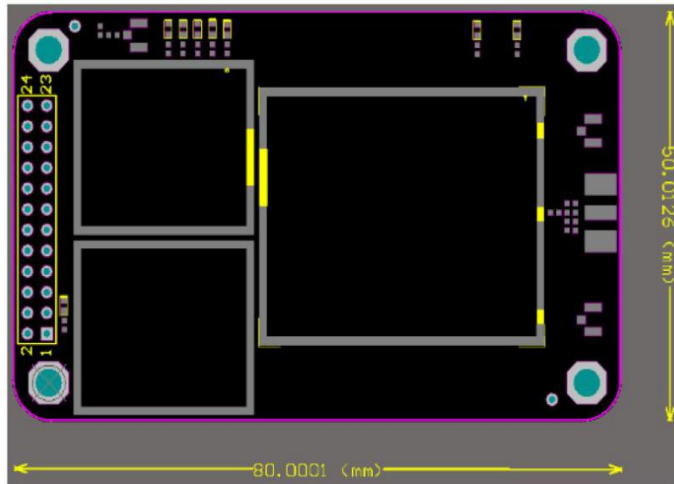
The RAK831 provides headers at the bottom side, which have a pitch of 2.54 mm. The description of the pins is given by below Table .

Pin	Name	Type	Description
1	+5V	POWER	+5V Supply Voltage
2			
3	GND	GND	GND
4	LNA_EN_A	Input	SX1301 Radio C Sample Valid
5	GND	GND	GPS Module LDO:Enable Pin

6	GND	GND	GND
7	RADIO_EN_A	Input	SX1257_A_EN
8	PA_G8	Input	PA GAIN 0
9	RADIO_EN_B	Input	SX1257_B_EN
10	PA_G16	Input	PA GAIN 1
11	PA_EN_A	Input	PA EN
12	GND	GND	GND
13	RADIO_RST	RST	SX1257_A_B RESET
14	GND	GND	GND
15	CSN	SPI	SX1301 SPI_NSS
16	MOSI	SPI	SX1301 SPI_MOSI
17	MISO	SPI	SX1301 SPI_MISO
18	SCK	SPI	SX1301 SPI_CLK
19	RESET	RST	SX1301 RESET
20	GPIO0	GPIO	SX1301 GPIO
21	GPIO1	GPIO	SX1301 GPIO
22	GPIO2	GPIO	SX1301 GPIO
23	GPIO3	GPIO	SX1301 GPIO
24	GPIO4	GPIO	SX1301 GPIO

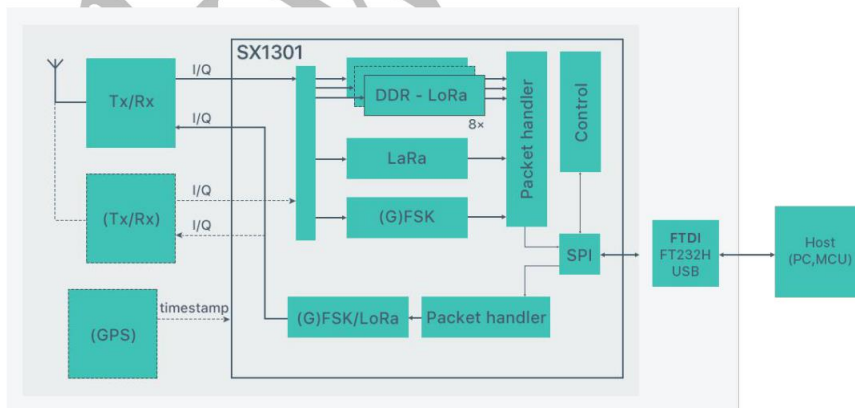
2.2 Module Dimensions

The outer dimensions of the RAK831 are given by 80.0 x 50.0 mm ± 0.2 mm. The RAK831 provide four drills for screwing the PCB to another unit each with a drill diameter of 3 mm.



3.Module Overview

The Concentrator Module is currently available in one versions with SPI interface.



3.1 SX1301

The RAK831 includes Semtech’s SX1301 which is a digital baseband chip including a massive digital signal processing engine specifically designed to offer

breakthrough gateway capabilities in the ISM bands worldwide. It integrates the LoRa concentrator IP.

The SX1301 is a smart baseband processor for long range ISM communication. In the receiver part, it receives I and Q digitized bit stream for one or two receivers (SX1257), demodulates these signals using several demodulators, adapting the demodulators settings to the received signal and stores the received demodulated

packets in a FIFO to be retrieved from a host system (PC, MCU). In the transmitter part, the packets are modulated using a programmable (G)FSK/LoRa modulator and sent to one transmitter (SX1257). Received packets can be time-stamped using a GPS PPS

RAK831
Datasheet

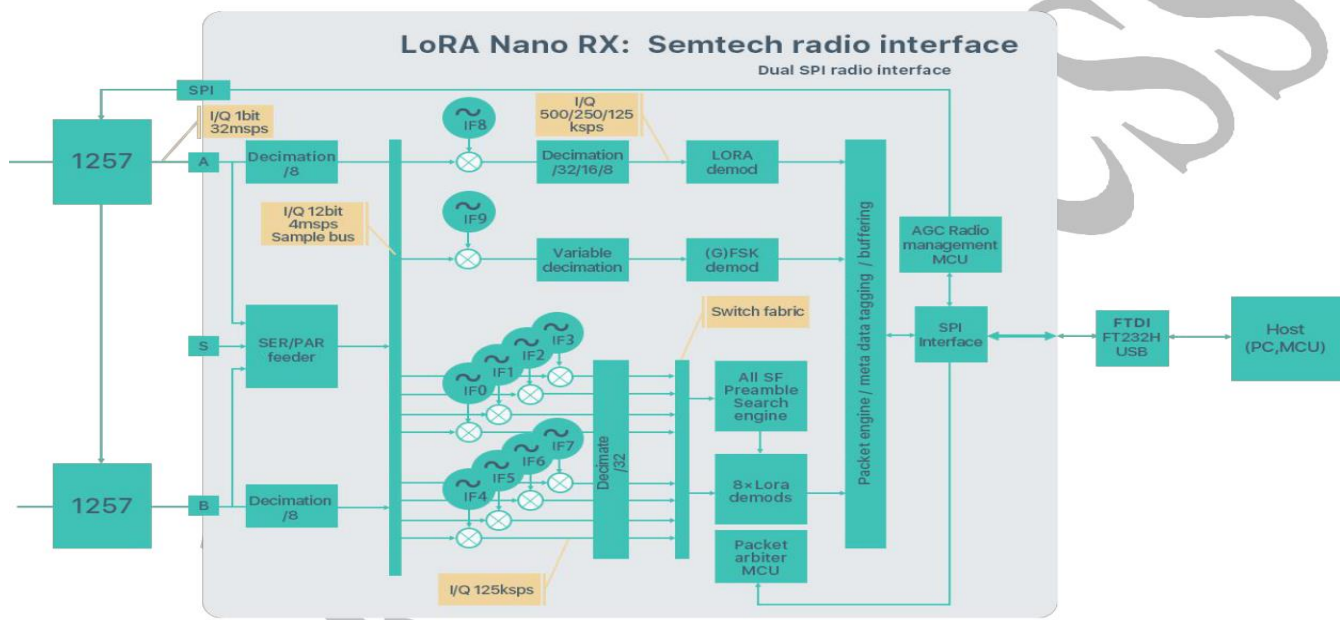
input.

The SX1301 has an internal control block that receives microcode from the host system (PC, MCU). The microcode is provided by Semtech as a binary file to load into the SX1301 at power-on (see Semtech application support for more information).

The control of the SX1301 by the host system (PC, MCU) is made using a Hardware Abstraction Layer (HAL). The Hardware Abstraction Layer source code is provided by Semtech and can be adapted by the host system developers.

It is highly recommended to fully re-use the latest HAL as provided by Semtech on <https://github.com/Lora-net>.

3.1.1 Block Diagram



The SX1301 digital baseband chip contains 10 programmable reception paths. Those paths have differentiated levels of programmability and allow different use cases. It is important to understand the differences between those demodulation paths to make the best possible use from the system.