

Arquitectura de comunicación para la digitalización de la agricultura en torno a la maquinaria agrícola

Communication architecture for the digitization of agriculture around agricultural machinery

Natalia Iglesias^{*1}, Pilar Bulacio^{*2} and Elizabeth Tapia^{*3}

^{*} *Facultad de Ciencias Exactas, Ingeniería y Agrimensura, UNR*

[†] *Instituto CIFASIS (CONICET-UNR)*

Rosario, Argentina

¹iglesias@cifasis-conicet.gov.ar

²bulacio@cifasis-conicet.gov.ar

³tapia@cifasis-conicet.gov.ar

Resumen—Se presenta una arquitectura de comunicación para habilitar servicios en la nube en torno a la maquinaria agrícola en zonas rurales con infraestructura de conectividad limitada. La arquitectura propuesta se compone de tres módulos, *Unidad a bordo, gateway y servicio en la nube*, que extienden el alcance de la cobertura de comunicación y permiten el intercambio de datos desde la maquinaria agrícola a la nube. La arquitectura se implementa con hardware de bajo costo y librerías de código abierto que permiten una rápida implementación. Los resultados obtenidos, en términos de latencia de comunicación, indican que la solución es adecuada para aplicaciones de monitoreo en maquinarias agrícolas que utilizan ISOBUS.

Palabras clave: LoRa; agricultura digital; monitoreo.

Abstract— A communication architecture is presented to enable cloud services around agricultural machinery in rural areas with limited connectivity infrastructure. This architecture is made up of three modules named: *on-board unit, gateway and cloud service*. Which together allow extending the scope of communication coverage, enabling the exchange of data from agricultural machinery to the cloud. The architecture is developed using low-cost hardware and open-source libraries that allow its rapid implementation. The results obtained, in terms of communication latency, indicate that the solution is suitable for monitoring applications in agricultural machinery using ISOBUS.

Keywords: LoRa; digital agriculture; monitoring.

I. INTRODUCCIÓN

La digitalización de la agricultura se refiere a la creación de valor a partir de los datos [1] mediante el uso de TICs (Tecnologías de la Información y la Comunicación). La agricultura digital provee soluciones a las demandas de la agricultura de precisión. Estas demandas atienden necesidades de gestión agronómica del campo a partir de la transformación de datos precisos en conocimiento procesable que impulse y respalde la toma de decisiones informadas [2]. Por tanto, se espera que la digitalización de la agricultura proporcione una optimización de la productividad agrícola de manera sustentable [3].

La agricultura digital introduce nuevas oportunidades de negocios [4] como la gestión y mantenimiento de la maquinaria agrícola (MA) a través del monitoreo remoto de variables de uso y el diagnóstico de fallas. Esto es posible debido a la aplicación de tres tecnologías que habilitan la transformación digital en la MA: la *electrificación* - en términos de eficiencia energética y operación (control de velocidad y punto de fuerza)-, la *conectividad* y el *aprendizaje de máquina*, que en conjunto permiten el desarrollo de aplicaciones inteligentes. Debemos notar que la disponibilidad y asequibilidad de infraestructura de conectividad y energía eléctrica requerida para la transformación digital del sector agroindustrial [5] no siempre se encuentra disponible. En particular, en lo que a conectividad refiere existen áreas geográficas en el mundo, incluida Argentina, donde la infraestructura de conectividad -como la tecnología celular- aún no ha sido desplegada. Siendo esto, un obstáculo para el desarrollo de la agricultura digital. En este contexto, en ausencia de infraestructura de conectividad tradicional, la elección de tecnologías alternativas disponibles para el despliegue de la agricultura digital en áreas rurales remotas es muy limitada. Una primera opción es la utilización de conectividad satelital [6], como la utilizada en escenarios de comunicaciones marítimas y manejo de emergencia [7]. Brevemente, los satélites de órbita terrestre geoestacionaria (GEO) se encuentran en una órbita alta (≈ 36000 km), lo que produce un gran retardo de comunicación (≈ 600 ms) y atenuación de la señal. Los satélites de órbita terrestre baja (LEO) se encuentran en una órbita más baja (≥ 1000 km), lo que produce un retardo de comunicación más bajo (≈ 40 ms) y una atenuación de señal relativamente pequeña. Por lo que, en términos de latencia, los satélites LEO son preferibles en soluciones de comunicación a los satélites GEO. Sin embargo, este tipo de comunicación resulta costosa en términos de implementación de la comunicación sobre dispositivos finales (\geq us\$ 200) y uso (alto costo de envío de mensajes) (\geq us\$ 8 planes por mes). Una segunda opción es la utilización de soluciones de comunicación inalámbricas emergentes en el contexto de IoT (Internet of Things) tales como las tecnologías LPWAN (Low Power

Wide Area Network) que proporcionan soluciones de bajo consumo de energía, largo alcance y bajo costo (\geq us\$ 50) [8]. Entre estas soluciones se encuentran SigFox [9] y LoRa [10] dos tecnologías líderes con algunas diferencias técnicas analizadas en [11] - [12], y resumidas de forma cualitativa en la Fig. 1 en términos de requerimientos de diseño para este trabajo (II-A). Finalmente, como una tercera opción se presentan redes híbridas que persiguen la convergencia de redes satelitales y terrestres basadas en soluciones IoT [13] donde la comunicación satelital es usada como backhaul y la comunicación LPWAN sobre las subredes en el campo [14]. Esta última opción de comunicación busca brindar conectividad en zonas remotas - embarcaciones marítimas, plataformas polares, etc- donde no es posible llegar con otro tipo de conectividad al tiempo que los costos de implementación y uso son reducidos a un nivel intermedio comparados con la primera opción, dado que la conectividad satelital es embebida en el dispositivo de borde de red y no sobre los dispositivos finales reduciendo así el número de transceivers satelitales necesarios para la comunicación. En este contexto, el presente trabajo busca brindar una solución de comunicación en áreas rurales con escasa infraestructura de conectividad a fin de promover el desarrollo de la agricultura digital. Como requisitos de diseño la solución debe ser factible de implementar en el contexto socio-económico argentino actual, de bajo costo de implementación y uso, proveer comunicación bidireccional, consumir bajos recursos de hardware sobre los dispositivos finales y permitir el despliegue de redes agrícolas privadas. Dado que estas características son ideales para la generación de servicios tecnológicos en el agro, acorde a la economía del conocimiento. Entre las opciones de comunicación previamente descritas, la segunda opción basada en tecnología LPWAN LoRa es seleccionada. Mientras que, la primera opción basada en conectividad satelital es descartada debido a su alto costo y la tercera opción es considerada como viable en situaciones donde la extensión de la cobertura de conectividad brindada por la opción dos no sea suficiente. La solución de comunicación presentada en este trabajo resuelve la transferencia de datos desde la maquinaria agrícola a la nube para su procesamiento en áreas con infraestructura de comunicación limitada. Su desempeño es evaluado en términos de tiempos de comunicación entre los módulos que integran la arquitectura a fin de contrastarlos con los requeridos por el protocolo de comunicaciones ISOBUS dentro de la MA.

II. DISEÑO E IMPLEMENTACIÓN

II-A. Especificaciones generales

La actividad agrícola es heterogénea en términos de geografía, cultivos, capacidades tecnológicas y estrategias comerciales. En particular, el tamaño del área cultivada y el tipo de cultivo definen el tipo y tamaño de maquinaria agrícola a utilizar. En la práctica, maquinarias de diferentes fabricantes deben trabajar juntas. Por lo que existe una necesidad de compatibilidad de interconexión entre dispositivos de diferentes marcas. En el contexto de la maquinaria agrícola, esta necesidad de compatibilidad es abordada por la norma ISO 11783 [15], más conocida como ISOBUS que especifica la comunicación entre los diferentes dispositivos

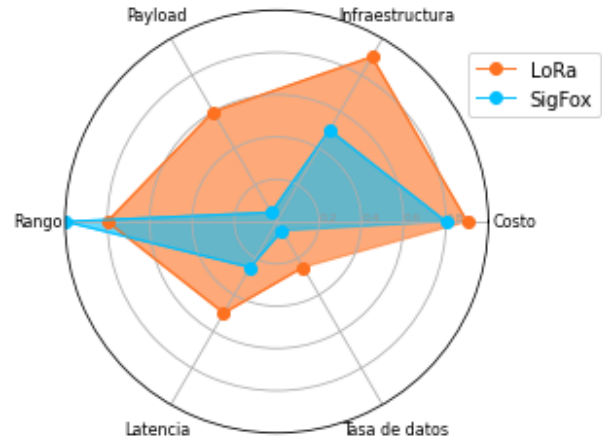


Figura 1. Comparación de SigFox y LoRa.

electrónicos, llamados ECU (Electronic Control Unit), alojados en la maquinaria agrícola mediante la normalización de los procesos de recolección, procesamiento, almacenamiento y transferencia y visualización de los datos. Note que, un requisito clave en la agricultura digital, habilitada por las tecnologías de IoT es la recopilación eficiente de datos. En particular, la interconexión inalámbrica, entre la electrónica a bordo de la maquinaria agrícola y los servicios en la nube, que permite la transferencia de esos datos para su recopilación resulta relevante [2]. Si bien, la norma ISOBUS, cubre aspectos relevantes a la transferencia de datos desde el tractor a los servicios en la nube [16], en la actualidad, estos lineamientos están siendo revisados [17] para lograr compatibilidad con toda la cadena de valor agroindustrial. En lo que respecta a la elección de la tecnología inalámbrica a utilizar (WiFi, Bluetooth, celular, etc) para la comunicación con la nube, ésta es libre pero condicionada por los requerimientos de métodos y formatos de datos ISOBUS.

En este contexto, las especificaciones generales de diseño del sistema son delineadas en función de los requisitos de estandarización agrícola, digitalización de servicios agrícolas y cobertura limitada de conectividad celular. En conjunto, con los requerimientos actuales de IoT en cuanto a adaptabilidad, seguridad, bajo costo, modularidad y escalabilidad [18]. Brevemente, en relación a los requerimientos funcionales del sistema, éste debe posibilitar la conectividad inalámbrica de largo alcance entre la MA y la nube. El mismo deberá estar integrado por tres módulos: un nodo a bordo, un nodo intermedio y un servicio en la nube. El nodo a bordo estará alojado en la maquinaria agrícola y deberá adaptar los métodos y formatos para que los datos recibidos desde la interfaz cableada interna de la maquinaria puedan ser transferidos a través de la interfaz inalámbrica al nodo intermedio. El nodo intermedio actuará como Gateway entre el nodo de abordaje y el servicio en la nube. Esto deberá permitir la extensión del área de conectividad a través del reacondicionamiento de los datos. El servicio en la nube alojará aplicaciones relacionadas con la gestión de la MA accesibles por los usuarios finales. Debemos notar que, la comunicación entre el nodo a bordo y el nodo intermedio debe ser posible sin la disponibilidad de infraestructura de conectividad a fin de proporcionar conexión de largo alcance

en zonas rurales. El sistema deberá considerar los estándares de conectividad vigentes, tanto los específicos a la MA como los relacionados a IoT para redes de largo alcance de cobertura, considerando la movilidad de los nodos. En particular, el desarrollo deberá minimizar los tiempos de transmisión tanto como sea posible. El sistema deberá hacer uso de dispositivos asequibles en consonancia con la especificación RFC 7228 [19] de dispositivos restringidos.

II-B. Arquitectura

La Fig. 2 muestra la arquitectura de la solución propuesta. La arquitectura se divide en tres módulos. Un módulo a nivel de la maquinaria agrícola constituido por la *Unidad a bordo*. Un módulo intermedio de interconexión constituido por un *Gateway* y un módulo superior en la nube donde se alojan los *servicios*. Las siguientes subsecciones presentan información detallada del hardware y software utilizado para la implementación del sistema. La elección del hardware y software se realizó bajo los requerimientos de IoT utilizando dispositivos restringidos en procesamiento, memoria y almacenamiento.

II-B1. Unidad a bordo: La Unidad a bordo se encuentra alojada en la MA y es la encargada de interconectar la red de comunicación ISOBUS interna de la MA con los servicios externos a través de conectividad inalámbrica de largo alcance. Para la selección del protocolo inalámbrico de largo alcance se priorizó aquellos que trabajan en bandas de frecuencia no licenciadas, permitieran la implementación de redes privadas y que el costo de implementación, uso y mantenimiento fuera reducido. Bajo estos requerimientos, las redes LPWAN como LoRa son consideradas apropiadas como tecnología de conectividad de última milla para IoT en áreas rurales. Las características principales son: largo alcance de transmisión, bajo consumo de energía y baja tasa de transferencia de datos.

Para la implementación de la Unidad a bordo se utilizó una plataforma de hardware basada en microcontrolador de 8 bits y 16 MHz, con 256 kB de memoria ROM y 8 kB de memoria RAM, 1 puerto de conectividad SPI (Serial Peripheral Interface) y 4 puertos USARTs que pueden ser utilizados en modo SPI. A la plataforma se le incorporó una interfaz de comunicación CAN [20], una memoria SD y una interface de comunicación LoRa [21]. Todas estas interfaces fueron interconectadas con el microprocesador utilizando el protocolo SPI. Para la implementación de la comunicación LoRa se utilizó la librería de código abierto LMIC [22] a fin de permitir una migración transparente al protocolo LoRaWAN [23] en un futuro si así se requiriese. LoRa basa su funcionamiento en una técnica de modulación de espectro expandido que incorpora cinco parámetros configurables: frecuencia de la portadora (FREQ), ancho de banda (BW), factor de dispersión (SF), tasa de codificación (CR) y potencia de transmisión (PT). A su vez para la transferencia de los datos, LoRa especifica una estructura de trama de longitud variable en función del tamaño del campo de datos (Payload). La configuración de estos valores influyen sobre la tasa de bits, el rango de comunicación, el tiempo de transmisión de las tramas (ToA) y la resistencia a las interferencias. Para la parametrización del protocolo LoRa los valores de SF, BW, CR, PT fueron acotados

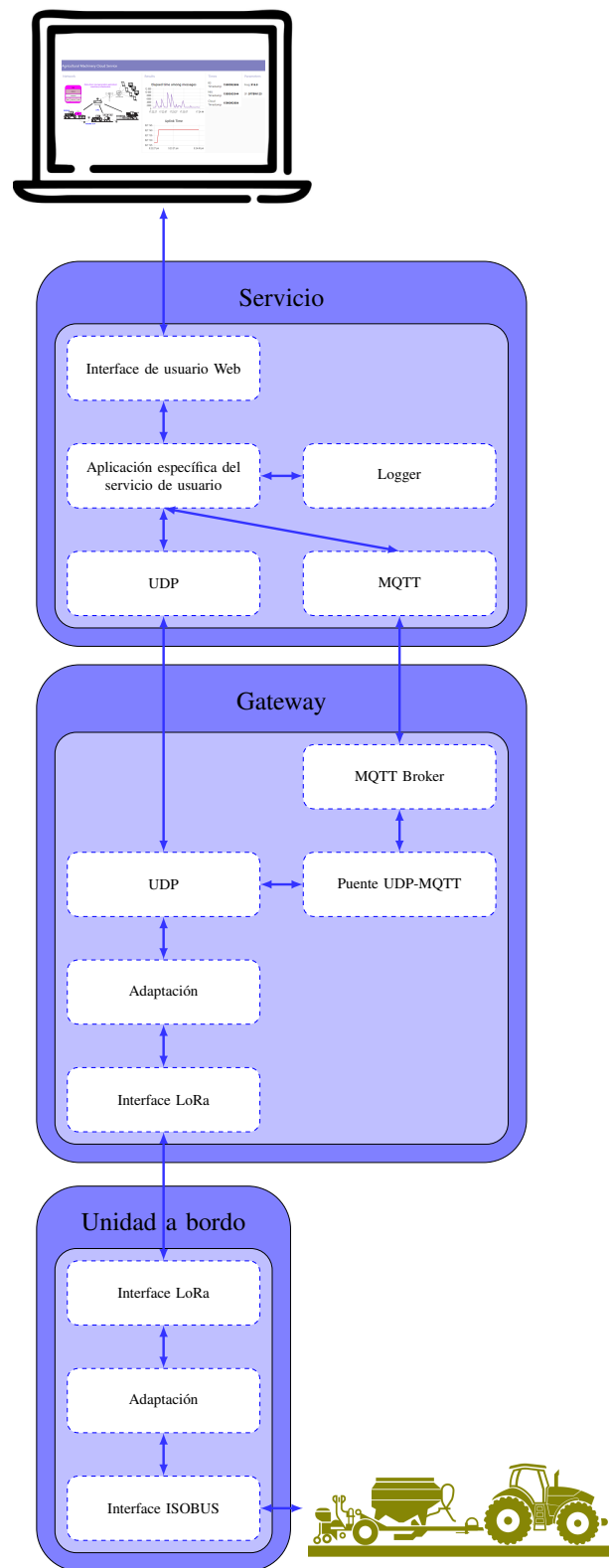


Figura 2. Arquitectura del sistema propuesta.

al subconjunto de valores definidos en LoRaWAN para la región AU915 a la cual Argentina adhiere [24].

Respecto a su funcionamiento, la Unidad a bordo, al inicializarse configura los periféricos y protocolos asociados a ellos. Luego se prepara para recibir de forma continua mensajes por el bus ISOBUS. Los mensajes ISOBUS recibidos son analizados sintácticamente y puestos en cola con

priorización, de acuerdo a la prioridad definida por ISO-BUS, para su adaptación y retransmisión sobre la interfaz inalámbrica LoRa. La transferencia de datos entre la Unidad a bordo y el Gateway se realiza mediante la transferencia de datos crudos a fin de mantener una carga útil reducida, con el objetivo de maximizar el rendimiento de la comunicación.

II-B2. Gateway: El Gateway se encuentra alojado en un lugar externo a la MA. Su posición debe ser elevada en altura a fin de posibilitar una línea de visión RF libre de obstáculos con la Unidad a bordo tanto como se pueda. El Gateway actúa como puerta de enlace entre la red LoRa y la red backhaul TCP/IP. Para la implementación del Gateway se utilizó una plataforma Pi 3 Model B+ basada en el procesador Cortex-A53 de 64 bits a 1.4 GHz, con 1 GB de RAM y conectividad LAN inalámbrica integrada. A la cual, se le incorporó una interfaz de comunicación LoRa [25] mediante el protocolo SPI. El Gateway implementado trabaja sobre un único canal de comunicación en la frecuencia de 916.8 MHz, la cual puede ser modificada en el momento de la configuración del dispositivo.

Respecto a su funcionamiento, el Gateway, al inicializarse configura los periféricos y protocolos asociados a ellos. Luego se prepara para recibir los mensajes desde la interfaz LoRa. Los mensajes recibidos son analizados sintácticamente y adaptados al formato JSON (JavaScript Object Notation) [26], ampliamente utilizado para transmitir información en aplicaciones web y sistemas IoT independientemente de la plataforma y protocolo que se esté usando. Posteriormente, el mensaje JSON es transmitido a un servidor remoto donde se aloja el Servicio mediante la interfaz TCP/IP. Para la transmisión del mensaje JSON dos opciones de protocolos de mensajes fueron implementadas. La primera opción consiste en la transmisión del mensaje JSON crudo utilizando el protocolo de transporte UDP [27] basado en mensajes. Mientras que, la segunda opción hace uso del protocolo de mensajes de capa de aplicación MQTT [28]. A diferencia de la primera opción, la segunda opción hace uso del protocolo de transporte TCP [29]. Para la implementación de la transmisión vía UDP se utilizó la librería de código abierto [30]. Para la implementación de la transferencia de mensajes vía MQTT, sobre el Gateway se creó un cliente MQTT utilizando [31], el cual publica mensajes con el tópico *gateway/GatewayID/event/up* sobre un MQTT Broker [32], también creado sobre el Gateway.

II-B3. Servicio en la Nube: El Servicio en la nube se encuentra alojado en un servidor remoto. La implementación de la aplicación del Servicio es dependiente de los datos recolectados y los requerimientos del cliente. Como caso de estudio en este trabajo se implementó una aplicación que recolecta datos sobre la velocidad del tractor y parámetros asociados a la red de comunicación LoRa a fin de realizar un análisis de rendimiento sobre esta. Para la implementación del Servicio se utilizó una Notebook i7 a 2.7 GHz con 8 GB de memoria RAM. La programación del Servicio se realizó bajo el paradigma de programación basada en eventos utilizando la herramienta Node-Red [33]. Respecto a su funcionamiento, el Servicio, de acuerdo a la configuración seleccionada puede conectarse vía UDP o MQTT al Gateway. En ambos casos, el Servicio escucha de forma continua el puerto de comunicación. Al recibir un

mensaje desde el puerto de comunicación, el Servicio guarda automáticamente el mensaje recibido en un archivo *log*. Casi al mismo tiempo que el mensaje recibido es analizado y particionado en base a una estructura “clave-valor” definida en los mensajes. Posteriormente, los datos recibidos son visualizados por los usuarios finales sobre una *Interface de visualización Web*. El usuario final también puede descargar el archivo *log* desde la Interface Web para su procesamiento offline.

III. EVALUACIÓN Y RESULTADOS

Con el objetivo de verificar la arquitectura de comunicación propuesta en términos del tiempo de ida y vuelta (RTT) de los mensajes. Y así, evaluar su viabilidad en el entorno de la MA, dos mediciones fueron realizadas. La primera medición de RTT se realizó sobre el enlace LoRa que interconecta los módulos Unidad a bordo y Gateway. La segunda medición de RTT se realizó sobre el enlace TCP/IP que interconecta los módulos Gateway y Servicio.

Las evaluaciones fueron realizadas en un entorno de laboratorio configurado de la siguiente manera. Para la parametrización del protocolo LoRa se establecieron los valores: SF = 7, BW = 125 kHz, CR = 4/5, PT= 14 dB y FREQ = 916.8 Mhz a fin de obtener la mayor tasa de transferencia de datos [34] sin sobrecargar las unidades de procesamiento y memoria sobre los dispositivos. Para la parametrización del protocolo MQTT, a fin de evaluar su desempeño versus el protocolo UDP, se establecieron los valores: Puerto TCP = 1883, QoS = 0 sin ACK - los mensajes se envían una vez y el entorno operativo hace el mejor esfuerzo por entregarlo -, además durante esta primera prueba de concepto no se utilizaron mecanismos de encriptación y autenticación de la comunicación. La distancia entre la Unidad a bordo y el Gateway fue de 7 m. El Gateway se ubicó a una altura de 2,5 m. La Unidad a bordo inicia el proceso de ensayo transmitiendo mensajes periódicos cada 1 segundo al Gateway, quien los retransmitirá al módulo Servicio. Los ensayos se realizan con tamaños de mensaje (Payload) de entre 8 Bytes y 48 Bytes a fin de evaluar los tiempos de comunicación, la carga sobre el bus y el desempeño de los buffers de comunicación. Para cada tamaño de mensaje se realizaron 3 ensayos de 1000 mensajes cada uno durante 3 días diferentes de prueba.

En particular, para la medición de RTT sobre el enlace LoRa que interconecta los módulos Unidad a bordo y Gateway, los mensajes enviados por la Unidad a bordo son recibidos por el Gateway y retransmitidos nuevamente a la Unidad a bordo. La Unidad a bordo se encarga de almacenar los Timestamp de envío y recepción de cada mensaje y calcular el valor RTT en base a la diferencia de tiempo entre ellos. Luego los registros de RTT medidos son extraídos y procesados con scripts desarrollados en Python.

Los resultados obtenidos de la medición de RTT son resumidos en el Cuadro I. Los resultados obtenidos concuerdan con los resultados esperados. Dado que en un enlace LoRa el valor RTT puede ser estimado como $2T_{oA} + T_{bs}$, donde T_{oA} refiere al Tiempo de Aire y T_{bs} al tiempo de procesamiento (Fig. 3). Un análisis analítico [34] sobre el Tiempo de Aire (T_{oA}) para BW= 125 KHz y CR = 4/5 fue realizado. La Fig. 4 muestra como el T_{oA} aumenta al

aumentar el tamaño del payload, lo cual resulta lógico para cada valor de SF entre 7 y 10. De igual modo, la pendiente de la característica $ToA vs Payload$ aumenta al aumentar el valor de SF, de forma que para un mismo tamaño de payload los tiempos de transmisión serán mayores para una parametrización LoRa de SF=10 que para SF=7. Un SF = 7 permite lograr menores tiempos de transmisión y mayores tasas de transmisión. En relación a la Fig. 4, resulta importante remarcar también que el eje vertical se encuentra truncado en el valor de 400 ms dado existen regulaciones sobre el uso compartido del espectro no licenciado que restringen la transmisión de forma continua sobre un mismo canal por más de 400 ms. Notar también sobre la Fig. 4 que, la línea rayada roja hace referencia a los tiempos de respuesta especificados en ISOBUS para comunicaciones del tipo *Solicitud-Respuesta*. Esta indica que si un mensaje enviado desde la MA requiere una respuesta, ésta deberá ser recibida en un plazo menor a 200 ms, caso contrario el mensaje de solicitud será nuevamente emitido por la MA pudiendo generarse un efecto no deseado que deteriore el sistema de comunicación.

Cuadro I
 ESTADÍSTICAS DESCRIPTIVAS PARA LOS VALORES RTT MEDIDOS.

Payload [bytes]	RTT [ms]				
	Media	Desvío	Min	Mediana	Max
8	87.245	0.340	86.120	87.224	88.372
16	117.721	0.296	116.744	117.812	118.128
24	138.461	0.301	137.396	138.488	138.788
32	159.041	0.356	158.072	159.148	159.528
40	179.769	0.287	178.780	179.856	180.136
48	210.490	0.135	209.428	210.496	211.316

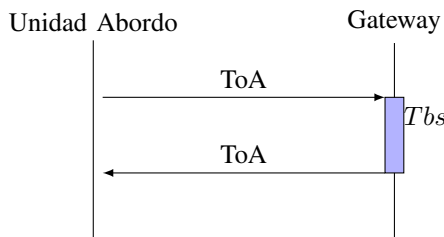


Figura 3. Medición de RTT sobre la conexión LoRa.

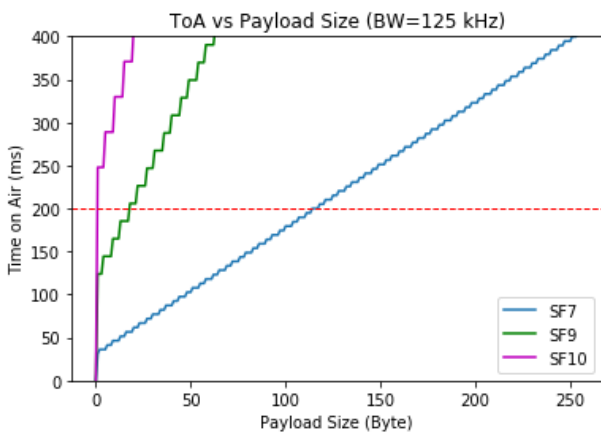


Figura 4. Tiempo de Aire de paquetes LoRa.

Como métrica para comparar los resultados medidos de RTT sobre el enlace LoRa con diferentes tamaños de Payload se utilizó el coeficiente de variación ($Desvio/Media$), $CV = (0.39, 0.25, 0.22, 0.21, 0.16, 0.11)$. Los resultados muestran que a medida que aumenta el tamaño de carga útil el RTT aumenta y el coeficiente de variación disminuye. El mejor rendimiento es obtenido para un tamaño de payload de 48 bytes. Sin embargo, el valor RTT para una payload de 48 bytes excede los 200 ms de tiempo de espera especificados por ISOBUS. Por lo que, éste tamaño de payload es descartado para ser utilizado en un escenario de comunicación ISOBUS-LoRa. Por otro lado, el peor rendimiento es obtenido para un payload de 8 bytes por lo que también es descartado. Como recomendación se sugiere la utilización de un tamaño de payload de entre 16 y 40 bytes. En particular, un payload de 16 bytes es suficiente para la transmisión de mensajes individuales ISOBUS. Además, un mensaje con payload de 16 bytes, al tener un valor bajo de ToA hace un uso menor del canal de comunicación disminuyendo las posibilidades de colisiones. Finalmente, el tiempo Tbs medido fue en promedio de 4.918 ms con una desviación estándar de 0.364 para todos los casos.

En relación a la segunda medición de RTT realizada sobre el enlace TCP/IP que interconecta los módulos Gateway y Servicios, debemos notar que los relojes internos de los diferentes módulos fueron sincronizados al servidor NTP (<https://www.pool.ntp.org/zone/ar>). Todo el tráfico intercambiado durante los ensayos fue capturado utilizando la herramienta Wireshark. Dos instancias de Wireshark se ejecutaron simultáneamente una sobre el Gateway y otra sobre el Servidor. Luego, todas las capturas recolectadas fueron procesadas y analizadas utilizando scripts desarrollados en Python. Además, dado que los valores de RTT medidos sobre el enlace TCP/IP son dependientes de la carga sobre la red, durante la ejecución de los ensayos se midió latencia de red entre ambos módulos utilizando la herramienta PING. Los valores de RTT obtenidos desde PING fueron: mín = 2 ms, medio = 6 ms y máx = 95 ms. Durante los ensayos los mensajes recibidos por el Gateway desde la Unidad a bordo son retransmitidos en forma simultánea por los protocolos UDP y MQTT al módulo de Servicio. La Fig. 5 muestra los valores de RTT medidos sobre ambas transferencias. En particular, puede observarse que el tiempo RTT para transferencias usando MQTT fue aproximadamente 5 veces superior a UDP. Específicamente, para un payload de 16 bytes, el valor medio RTT para MQTT fue de 47.44 ms con un desvío de 1.86, mientras que el valor medio de RTT para UDP fue de 8.19 ms con un desvío de 1.83. De igual forma, para un payload de 40 bytes, el valor medio RTT para MQTT fue 47.48 ms con un desvío de 1.96, mientras que el valor medio de RTT para UDP fue de 8.41 ms. Nótese también que tanto para MQTT como para UDP el tamaño de payload utilizado no muestran diferencias significativas sobre los valores de RTT utilizando el mismo protocolo. Esto es debido a que el mayor tiempo consumido durante la transferencia se debe al tamaño de las cabeceras de los protocolos TCP/IP. Lo cual también puede ser evaluado en términos de uso promedio de la capacidad de canal, donde para igual cantidad de mensajes con payload de 40 bytes la tasa de transferencia fue de 3894 bps para MQTT y de

1798 bps (aprox. la mitad) para UDP. Sobre estos resultados, no se contempla la sobrecarga generada por el intercambio de paquetes TCP de establecimiento y mantenimiento de la conexión MQTT.

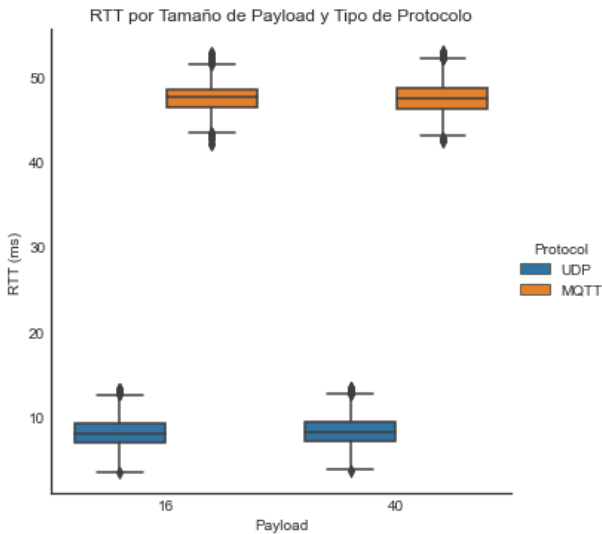


Figura 5. RTT medido para protocolos UDP y MQTT.

IV. DISCUSIÓN

Los resultados obtenidos muestran que los tiempos RTT de la arquitectura de comunicación basada en LoRa son del orden del tiempo de espera de retransmisión ISOBUS (200 ms). Por lo que, la arquitectura propuesta resulta más adecuada para tareas de monitoreo de la MA que utiliza ISOBUS, que para tareas que requieran acción remota en tiempos breves. Los ensayos experimentales indican que los valores de RTT son afectados por la parametrización del protocolo LoRa y el tamaño del payload como se observó en la sección III. A la vez que, la selección del valor SF a utilizar afecta la tasa de transferencia de datos y la inmunidad al ruido del sistema [35]. En relación al análisis sobre el efecto que tiene la distancia sobre los valores RTT, este requiere la realización de estudios adicionales. Sin embargo, algunas primeras observaciones realizadas en ambiente de laboratorio indicarían que los valores de RTT no son afectados de forma significativa por la distancia.

A fin de evaluar el peor caso en términos de recursos de Hw, pero el mejor caso en términos de costos, la arquitectura implementada utiliza un Gateway de un único canal. Esto significa que el Gateway no puede recibir y enviar mensajes LoRa de forma simultánea. Por defecto, el Gateway, se encuentra siempre en estado de RX y ocasionalmente en caso de tener que enviar una respuesta conmuta su radio LoRa de modo RX a TX. Durante dicho período de TX, si algún mensaje es enviado por una Unidad a bordo al Gateway, el mensaje será perdido. El protocolo LoRa utiliza el protocolo ALOHA puro [36] como mecanismo de acceso al medio, esto hace que la carga operativa de la red sea de un 18 % respecto al límite máximo soportado. Dicho límite máximo [35] dependerá de la configuración de las Unidades a bordo. Para una red de un canal y cuyas Unidades a bordo sean configuradas con un mismo valor $SF=7$, la capacidad ideal del canal máxima será de $(SF * \frac{BW}{2^{SF}} * CR) = 5468$ bps.

Mientras que para una red de un canal, cuyas Unidades a bordo sean configuradas con diferentes valores de SF entre 7 y 10, la capacidad del canal será de 11326 bps, debido a la ortogonalidad de las señales LoRa. Así, la carga operativa de la red resulta en 984 bps y 2038 bps respectivamente. Lo que se traduce en que el Gateway puede soportar aprox. 4 Unidades a bordo “activas” simultáneamente todas configuradas con mismo $SF = 7$ y transmitiendo de forma organizada un payload de 16 bytes. Lo cual, para un modelo de negocio de red privada y de muy bajo costo en el entorno de la MA, puede resultar adecuado. Esta solución es factible de escalar con baja inversión utilizando un Gateway de 8 canales de RX con un canal adicional independiente para TX (\geq us\$ 170). En cuanto al rango de cobertura del Gateway, éste es altamente dependiente del ambiente, la altura donde esté emplazado, y la ganancia de la antena utilizada. Pudiendo brindar un alcance entre cientos de metros a kilómetros. En particular, el Gateway implementado tiene un link budget teórico de 137 dB.

V. CONCLUSIONES

En este trabajo se presentó la arquitectura modular de un sistema de comunicación que permite extender el área de cobertura de los servicios de monitoreo de la maquinaria agrícola desde la nube. Los resultados experimentales muestran que la arquitectura implementada es válida para tareas de monitoreo donde, los datos se envían a la nube, procesan y analizan facilitando la toma de decisión informada por parte del trabajador de campo.

Actualmente, se están llevando a cabo trabajos de actualización de la arquitectura desde el protocolo LoRa a LoRaWAN, a fin de permitir la integración con redes públicas basadas en tecnología LoRa, en caso de ser necesario. Mientras que, la Unidad a bordo basada en una plataforma de 8 bits esta siendo actualizada a una plataforma de 32 bits, manteniendo las características de recursos restringidas, con el objetivo de reutilizar aplicaciones y herramientas previamente desarrolladas para ISOBUS.

REFERENCIAS

- [1] CEMA, “Digital farming: What does it really mean?” Feb. 2017, (Accedido Septiembre 2020). [Online]. Available: https://www.cema-agri.org/images/publications/position-papers/CEMA_Digital_Farming_-_Agriculture_4.0_13_02_2017_0.pdf
- [2] A. Villa-Henriksen, G. T. Edwards, L. A. Pesonen, O. Green, and C. A. G. Sorensen, “Internet of things in arable farming: Implementation, applications, challenges and potential,” *Biosystems Engineering*, vol. 191, pp. 60 – 84, 2020. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1537511020300039>
- [3] B. Basso and J. Antle, “Digital agriculture to design sustainable agricultural systems,” *Nature Sustainability*, vol. 3, pp. 254 – 256, apr 2020.
- [4] E. C. for Latin America and the Caribbean (ECLAC), “Data, algorithms and policies: redefining the digital world (lc/cmsi.6/4),” 2018. [Online]. Available: https://repositorio.cepal.org/bitstream/handle/11362/43515/7/S1800052_en.pdf
- [5] S. V. Nikola M. Trendov and M. Zeng, “Digital technologies in agriculture and rural areas. briefing paper,” 2019. [Online]. Available: <http://www.fao.org/3/ca4887en/ca4887en.pdf>
- [6] I. del Portillo, B. G. Cameron, and E. F. Crawley, “A technical comparison of three low earth orbit satellite constellation systems to provide global broadband,” *Acta Astronautica*, vol. 159, pp. 123 – 135, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0094576518320368>

- [7] S. D. Ilcev, *Global Mobile Satellite Communications Theory: For Maritime, Land and Aeronautical Applications*, 2nd ed. Springer Publishing Company, Incorporated, 2016.
- [8] LoRaAlliance, "Agricultural applications." [Online]. Available: <https://loro-alliance.org/lorawan-vertical-markets/agriculture/>
- [9] Sigfox, online. Accessed: 16 Nov 2020. [Online]. Available: <https://www.sigfox.com>
- [10] Semtech, "What is lora?" (Accedido Septiembre 2020). [Online]. Available: <https://www.semtech.com/lora/what-is-lora>
- [11] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of lpwan technologies for large-scale iot deployment," *ICT Express*, vol. 5, no. 1, pp. 1 – 7, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2405959517302953>
- [12] B. Foubert and N. Mitton, "Long-range wireless radio technologies: A survey," *Future Internet*, vol. 12, p. 13, 01 2020.
- [13] I. Lysogor, L. Voskov, A. Rolich, and S. Efremov, "Study of data transfer in a heterogeneous lora-satellite network for the internet of remote things," *Sensors*, vol. 19, no. 15, 2019. [Online]. Available: <https://www.mdpi.com/1424-8220/19/15/3384>
- [14] M. Palattella and N. Accettura, "Enabling internet of everything everywhere: Lpwan with satellite backhaul," *2018 Global Information Infrastructure and Networking Symposium (GIIS)*, pp. 1–5, 2018.
- [15] ISO, "Iso11783 (all parts), tractors and machinery for agriculture and forestry — serial control and communications data network," 2019.
- [16] J. Backman, R. Linkolehto, M. Koistinen, J. Nikander, A. Ronkainen, J. Kaivosoja, P. Suomi, and L. Pesonen, "Cropinfra research data collection platform for iso 11783 compatible and retrofit farm equipment," *Computers and Electronics in Agriculture*, vol. 166, p. 105008, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168169918317381>
- [17] AEF, "Agricultural industry electronics foundation," 2020, online. Accessed: 30 Jun 2020. [Online]. Available: <http://aef-online.org/>
- [18] E. G. Petrakis, S. Sotiriadis, T. Soultanopoulos, P. T. Renta, R. Buyya, and N. Bessis, "Internet of things as a service (itaas): Challenges and solutions for management of sensor data on the cloud and the fog," *Internet of Things*, vol. 3–4, pp. 156 – 174, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2542660518300350>
- [19] C. Bormann, M. Ersue, and A. Keränen, "Terminology for Constrained-Node Networks." RFC 7228, May 2014. [Online]. Available: <https://rfc-editor.org/rfc/rfc7228.txt>
- [20] Elecrow, "Can-bus shield-v1.4." [Online]. Available: <https://www.elecrow.com/canbus-shield-p-1133.html>
- [21] H. RF, "Rfm95/96/97/98(w) - low power long range transceiver module," 2014.
- [22] IBM, "Library: Lorawan mac on embedded systems." [Online]. Available: <https://github.com/mcci-catena/ibm-lmic/>
- [23] LoRa-Alliance, "Lorawan 1.0.3 specification," 2018.
- [24] —, "Rp002-1.0.1 lorawan regional parameters," February 2020, status: Final.
- [25] Dragino, "Raspberry pi hat featuring gps and lora technology." [Online]. Available: <https://www.dragino.com/products/lora/item/106-lora-gps-hat.html>
- [26] T. Bray, "The JavaScript Object Notation (JSON) Data Interchange Format," RFC 7159, Mar. 2014. [Online]. Available: <https://rfc-editor.org/rfc/rfc7159.txt>
- [27] "User datagram protocol," RFC 768, Aug. 1980. [Online]. Available: <https://rfc-editor.org/rfc/rfc768.txt>
- [28] OASIS, "Message queuing telemetry transport," 2019, online. Accessed: 30 Jun 2020. [Online]. Available: <http://mqtt.org/>
- [29] "Transmission Control Protocol," RFC 793, Sep. 1981. [Online]. Available: <https://rfc-editor.org/rfc/rfc793.txt>
- [30] Semtech, "Lora network packet forwarder." [Online]. Available: https://github.com/lora-net/packet_forwarder
- [31] ChirpStack, "Gateway bridge: abstracts packet forwarder protocols into protobuf or json over mqtt." [Online]. Available: <https://github.com/brocaar/chirpstack-gateway-bridge>
- [32] Eclipse, "Eclipse mosquito - an open source mqtt broker." [Online]. Available: <https://mosquitto.org/>
- [33] O. Foundation, "Node-red," online. Accessed: Sept 2020. [Online]. Available: <https://nodered.org/>
- [34] Semtech, "Sx1276/77/78/79 137 mhz to 1020 mhz low power long range transceiver," 2019.
- [35] —, "An1200.22. lora modulation basics."
- [36] N. Abramson, "The aloha system: another alternative for computer communications," in *Proceedings of the November 17-19, 1970, fall joint computer conference*, 1970, pp. 281–285.