



NUEVAS REALIDADES PARA LA EDUCACIÓN EN INGENIERÍA:
CURRÍCULO, TECNOLOGÍA, MEDIO AMBIENTE Y DESARROLLO

13 - 16
DE SEPTIEMBRE

2022

CARTAGENA DE INDIAS,
COLOMBIA



Internet of medical things and coding as a service (IoMTCaaS) for monitoring people affected by Covid-19

Yair Rivera, Elmer Vega, Enrique Jonathan Peña Cantillo

**Corporación Universitaria Americana
Barranquilla, Colombia**

Resumen

Con la necesidad cada vez mayor de recursos computacionales y de transferencia de datos que exigen los servicios de telemedicina, no es realista que los dispositivos médicos móviles (IoMT, Internet of Medical Things) locales (con fuentes limitadas) implementen el procesamiento intenso de la información a gran escala para el envío continuo de los datos sobre una red inalámbrica móvil. Hablamos de la adquisición y envío de datos sanitarios móviles a través de gadgets médicos wearables y de la aplicación de estos datos en la monitorización de diversas condiciones de salud, especialmente en pacientes afectados por el COVID-19 en áreas remotas y de difícil acceso. Un monitoreo remoto permite la obtención de variables biométricas como el nivel de oxígeno en sangre, el ECG, la presión sanguínea, el asma, etc. y otros servicios relacionados con la atención médica como la videoconferencia. El rendimiento del sistema es muy importante para el sistema de salud móvil basado en el IoT. Por ello, los particulares o las empresas se inclinan por externalizar sus necesidades de envío de datos y generación de servicio a través de soluciones cerradas que exigen un consumo significativo del ancho de banda. Sin embargo, con la gran cantidad de recursos compartidos en el ancho de banda, la externalización conlleva a problemas de retardo y rendimiento del sistema, lo que hace que se genere un aumento del consumo energético producto de los errores del sistema, algo no ideal para dispositivos IoT. Recientemente, se han llevado a cabo numerosos trabajos basados en la codificación distribuida de última generación en la universidad americana. El objetivo es desarrollar un mecanismo (Middleware) complementario de codificación (RLNC, Random Lineal Network Coding) controlado desde la capa de aplicación, el cual permite reducir los errores del sistema y por consiguiente minimizar los tiempos de la comunicación en dispositivos médicos 5G-IoT. El esquema planteado basado en los códigos rateless, es un proceso de baja complejidad que permite subsanar fallas originadas por las variaciones del canal

inalámbrico, como consecuencia garantiza un mínimo retardo sobre la red, especialmente en aquellas con poca cobertura, ubicadas en zonas remotas y de difícil acceso.

Palabras clave: IoMT; COVID-19; Middleware; RLNC; 5G-IoT; rateless

Abstract

With the ever-increasing need for computational and data transfer resources demanded by telemedicine services, it is unrealistic for local (with limited sources) mobile medical devices (IoMT, Internet of Medical Things) to implement intensive large-scale information processing for continuous data delivery over a mobile wireless network. We are talking about the acquisition and sending of mobile health data through wearable medical gadgets and the application of this data to monitor various health conditions, especially in patients affected by COVID-19 in remote and hard-to-reach areas. Remote monitoring allows obtaining biometric variables such as blood oxygen level, ECG, blood pressure, asthma, etc. and other healthcare-related services such as videoconferencing. System performance is significant for the IoT-based mobile health system. Therefore, individuals or enterprises are inclined to outsource their data delivery and service generation needs through closed solutions that require substantial bandwidth consumption. However, with many shared resources in bandwidth, outsourcing leads to system delay and performance issues, resulting in increased power consumption due to system errors, which is not ideal for IoT devices. Recently, a lot of work based on state-of-the-art distributed coding has been carried out at the American university. The objective is to develop a complementary coding middleware mechanism (RLNC, Random Linear Network Coding) controlled from the application layer, which reduces system errors and minimizes communication times in 5G-IoT medical devices. The proposed scheme based on rateless codes is a low complexity process that allows overcoming failures caused by variations in the wireless channel, thus ensuring a minimum delay on the network, especially in those with poor coverage, located in remote areas and difficult to access.

Keywords: IoMT; COVID-19; Middleware; RLNC; 5G-IoT; rateless

1. Introducción

La masificación de servicios sobre la web y la implementación de dispositivos IoT han abonado el camino para mejorar la calidad de la vida humana, en especial la salud. El desarrollo de arquitecturas de conectividad ha desarrollado plataformas de servicios sanitarios con facultades inteligente y omnipresente, sobre todo en áreas remotas y de difícil acceso. Una nueva denominación de servicios, Internet de las Cosas de la Salud (IoHT) ha permitido el desarrollo de componentes para el tratamiento de grandes volúmenes de datos sanitarios en un formato descentralizado¹. Gracias al desarrollo de sensores biométricos es posible la capturas las señales (temperatura, presión arterial, nivel de respiración, entre otros) medicas de un paciente y almacenarlas para su posterior análisis. De igual manera proporcionar acceso a los datos a través de una interfaz de programación de aplicaciones (API) y con ayuda de funciones provenientes del protocolo HTTP ofrece una abstracción de acceso a recursos en un formato menos complejo^{2,3}. Este nivel de abstracción hace



que muchas compañías decidan crear extensiones de servicios como clientes de datos como servicio (DaaS) basados en IoT y de software como servicio (SaaS) basado en IoT, la fusión de esto se define como IoTaaS. IoTaaS se considera como un proveedor de servicios que ofrece a los usuarios de IoT varios niveles de acceso que van desde los extremos finales definido por sensores, actuadores y dispositivos de seguimiento de la ubicación como la parte central de los datos definida en la nube, todo un middleware conformado por micro servicios que podrían desplegarse de forma dinámica a través de contenedores y orquestadores^{4,5}. Ahora bien, en la actualidad y debido a la crisis sanitaria generada por el coronavirus covid-19, son varias las propuestas que se han realizado desde IoT y que contribuyen a diferentes avances. Se han propuesto un modelo de rastreo de contactos de COVID-19 basado en el IoT^{6,7}. algunos que van desde la propuesta de un túnel epidémico inteligente que es una tecnología de asistencia para la desinfección de COVID19 solamente, hasta otros sistemas basados en un sistema que utiliza un casco inteligente para detectar únicamente la temperatura corporal⁸. Tecnologías que van desde un reconocimiento inteligente, la localización, el seguimiento, la supervisión y la gestión de datos de posibles infectados, hasta un informe a sus posibles contactos, todo con la finalidad de hacer un cerco epidemiológico. Sin embargo, no hay una arquitectura clara de servicios basado en contenedores, y peor aun, como son sistemas realizados para sistemas urbanos, cuando se aplica en áreas de difícil acceso, cuando se despliegan algunos recursos del alto rendimiento estos presentan un retardo en la red algo exagerado. Sistemas como el envío de datos a través de *video streaming*, presentan un consumo en la red debido al bajo rendimiento originado por la conexión inalámbrica de los datos. Nuestro modelo ofrece una arquitectura de servicio basado en contenedores y orquestadores, el cual garantiza una demanda dinámica de servicios, en este caso con un servicio agregado de codificación rateless. Lo que en otras palabras que los servicios de alto rendimiento tendrán la finalidad de segmentar y codificar los datos acorde a las exigencias del medio de transmisión, lo que al final me permite segmentar y codificar los datos de forma distribuida, lo que al final garantizaría una QoS en el sistema⁹.

2. Modelo propuesto

Inicialmente la arquitectura a estos modelos se definía sobre una arquitectura en la nube, una solución ideal para la descarga de cómputo y almacenamiento, un sistema fácilmente escalable. Sin embargo, las arquitecturas en la nube están alojados en zonas remotas, de ahí que las consultas a estos sistemas resultan con una alta latencia. Para solucionar esta restricción, se define una extensión Edge o extensión perimetral. Con esta modificación sobre la arquitectura como una extensión, los recursos informáticos están disponibles en el borde de la red y más cerca de los dispositivos finales, en otras palabras, se presenta de forma generalizada una reducción en la latencia de comunicación. esto tiene como consecuencia la reducción de las demandas del ancho de banda sobre la red. Esta abstracción perimetral permite que al final La movilidad y la distribución se presenten como una capacidad clave de las implementaciones de IoT^{10,11}, ver figura 1.



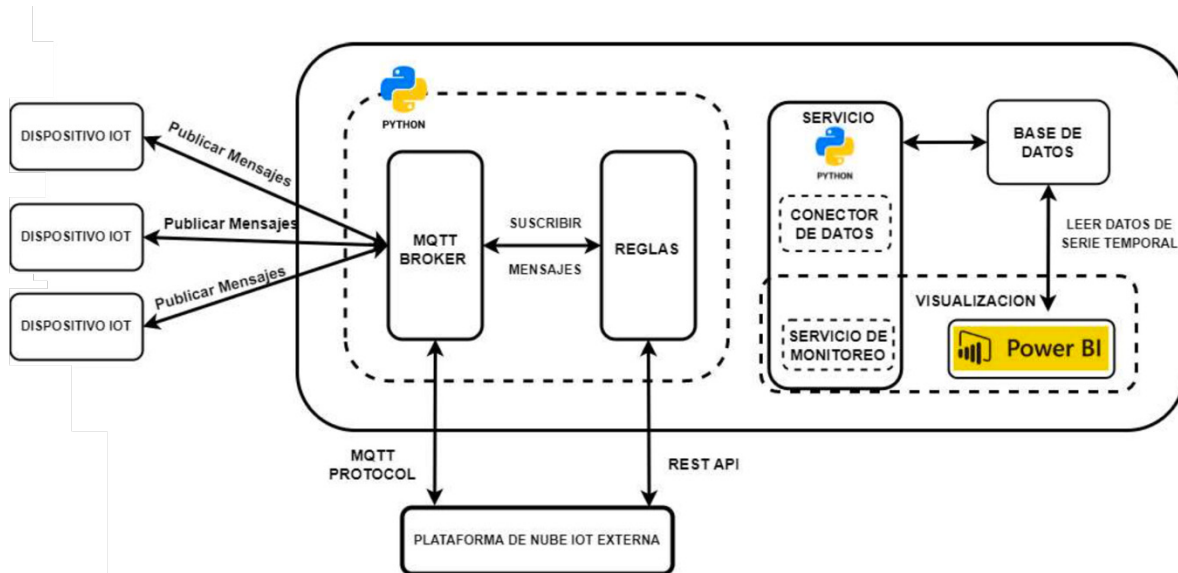


Fig 1. Arquitectura de servicios Propuesta.

La arquitectura presenta unos componentes que se pueden desplegar sin problemas de versiones o librerías compartidas, ya que se encuentran en contenedores dinámicos definidos por Docker. El servidor principal IoT se encuentra configurado sobre un sistema Node-Red, el cual me permite la recepción y el registro de los datos desde cada dispositivo en un *Dashboard*. El servidor IoT se ubica en un contenedor Docker, desde donde el cual se despliegan los micro servicios disparados por el protocolo MQTT con ayuda de un agente MOSCA. Este último encargado de definir las reglas para el servidor Node.js para IoT: Pub/Sub con MQTT, Testing y WebSockets [2]. Gracias a la orquestación de servicios implementado por Kubernetes, si se interrumpe un servidor intermediario, los otros intermediarios pueden ayudar a respaldar los datos del entorno. Por otro lado, para el reporte de los datos desde su almacenamiento una opción válida es a través de Power BI, la cual es una herramienta empresarial que permite unir diferentes fuentes de datos, analizarlos y presentarlos a través de informes y paneles.

Gracias a las publicaciones del protocolo MQTT es posible obtener un envío masivo de información desde cada dispositivo, los cuales son consultados a través de suscripciones masivas a través de una API definida por la plataforma. Un sistema que permite el envío y almacenamiento de la información a través de la capa de aplicación, ver figura 2.

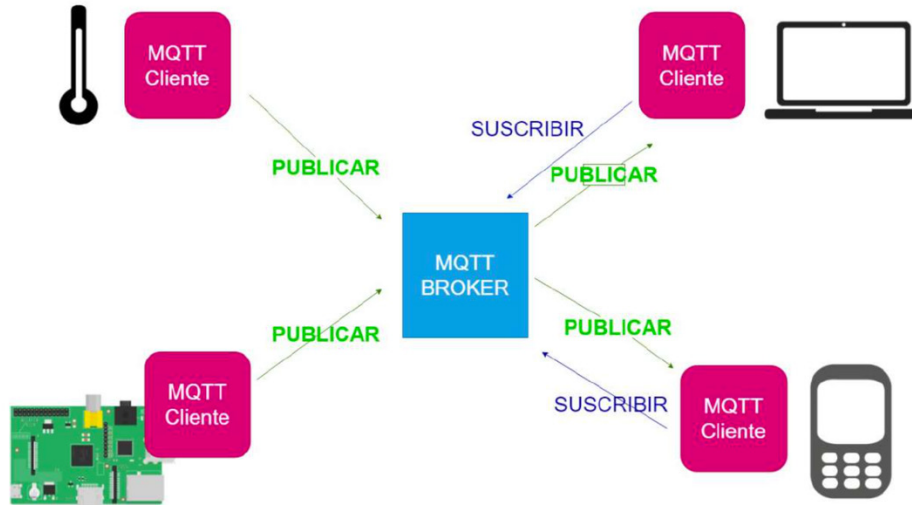


Fig 2. Sistema de suscripción y publicación de datos a través del protocolo MQTT.

El sistema también tiene muchos tipos de complementos de extensión para combinar con diversas plataformas de servicios en la nube de IoT e interactuar con las opciones de almacenamiento de datos de Edge hub, en donde por es posible almacenar temporalmente datos de señales biomédicas (ECG, temperatura, frecuencia del pulso y saturación de oxígeno para producir signos vitales para aplicaciones médicas posteriores) 12–14.

3. Procesamiento distribuida Edge

La parte de acceso de los datos está formado por sensores de baja potencia conectados a través de la red inalámbrica wifi con ayuda del modem esp8266, la red de sensores está formado por un grupo de dispositivos cuya topología está definida por el protocolo que se implemente, adicionalmente la contención del medio esta normalizado por el protocolo de acceso al medio. Esta contención al medio estaría regida por una topología ad-hoc o otra de arquitectura, ver figura 3.

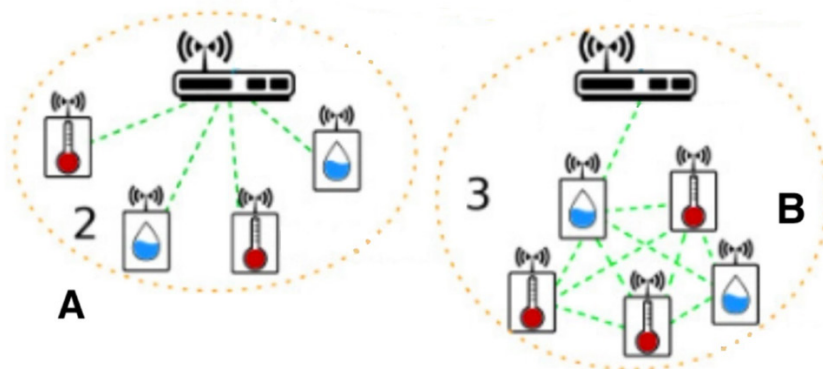


Fig 3. Tipos de arquitecturas inalámbricas.



La subred de la A es una red de censado, en donde hay un dispositivo central que administra la información y es el encargado de regular todo el tráfico, es un tipo de acceso al medio Multiple Access (MA). LA subred B es una topología ad-hoc, una topología distribuida con un acceso al medio Carrier Sense (CA), de donde los miembros de una red solo pueden enviar datos a través de ella cuando el medio de transmisión no esté ocupado. Para que esto sea posible en todo momento se está realizando un reconocimiento de la red para comprobar el estado del canal (detección de portadora). Solo cuando esté libre, se pueden enviar los datos. Cada nodo final tiene conexión con otro nodo master utilizando WiFi y en distancias relativamente cortas. El nodo master se conecta a distancias mayores con el nodo gateway utilizando Lora, de ahí que el nodo master debe trabajar con dos modem para trabajar con dos protocolos de comunicación, los modem del protocolo Master deben ser esp8266 y Módem LoRa RN2483, , ver figura 4.15,16.

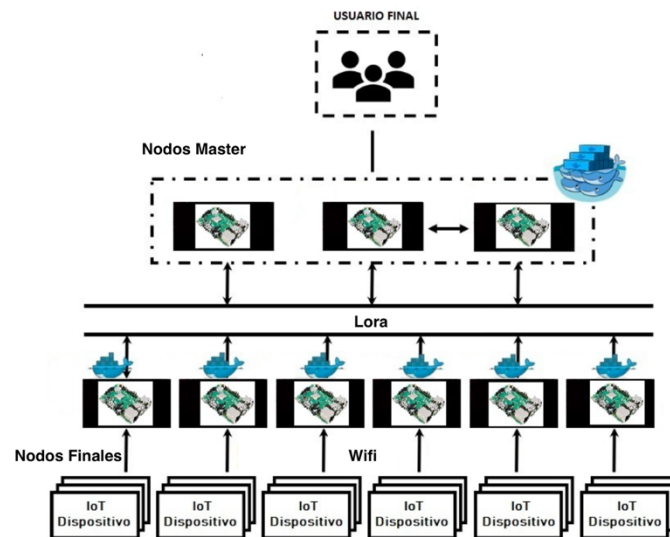


Fig 4. Arquitectura Nodos Master y Nodos sensores

El micro servicio que se genera en los nodos master genera mecanismos encargados de administrar el flujo de los datos, por tal motivo, en este punto la forma en que se envía el tráfico puede ser por reenvío o por agregación. Un tráfico por reenvío cada trama es redirigida al Gateway sin ninguna función de procesamiento, por el contrario, por agregación, se genera un procesamiento extra con la finalidad de agrupar varios datos y enviarlo en una sola trama de datos. Sin embargo, un procesamiento extra por agregación es compensado por un ahorro energético, ya que gracias a esta funcionalidad algunos nodos de la red para eliminar redundancia. Cada nodo final se conecta a la red a través del protocolo UDP/IP, un protocolo con muy poca sobrecarga y actividad de administración, el uso de UDP tiene la ventaja de ahorro de consumo de energía respecto a TCP, dado que con UDP no se tienen los mensajes propios de establecimiento de sesión/conexión necesarios para usar TCP. básicamente la funcionalidad del nodo master es recibir toda la información de la subred de sensores y retransmitirla hacia el nodo gateway. El nodo gateway concentra los paquetes LoRa de cada uno de los nodos master de la red, y los reenvía hacia la nube toda la información^{17,18}. El usuario final conformado por el sistema de registros médicos, permite de alguna manera monitorear los signos vitales de cada paciente afectado por el covid-19 en tiempo real, así como su ubicación en el espacio. Cabe anotar que existen tecnologías compatibles para el geoposicionamiento de personas, en donde se puede registrar tanto su

trayectoria como su ubicación de forma remota. Por tanto, en la medida que se presente o se pronostique una lectura anormal de sus signos vitales se puede desplegar un equipo de emergencias, dependiendo de la ubicación del paciente.

4. Codificación lineal de los datos

En cada nodo master es posible generar una codificación lineal de los datos a través de la codificación RLNC (Random Lineal Network Coding) y el operador Xor \otimes , gracias a que es el nodo de generar la agregación de los datos, por tanto, es posible realizar la segmentación para hacer la codificación en base al mapeo de campos finitos para explotar sus propiedades matemáticas en la transmisión de datos digitales. Después de haber segmentado la información en el origen, se establece una combinación lineal de segmentos de paquetes o símbolos seleccionados de forma aleatoria. Para ejercer una combinación lineal es necesario generar coeficientes lineales a través de una matriz de codificación, ver Ec1^{19,20}.

$$v_i, \forall_i \in [1, g], \quad (\text{Ec1}).$$

Donde v_i representa cada una de las entradas de la matriz de codificación, y g es el rango de la matriz o número de paquetes seleccionados para conformar la generación. Cada símbolo es codificado a través de operaciones formalizadas dentro del subespacio vectorial $GF(q)$, donde q es una potencia prima. Para empezar el proceso de codificación, se seleccionan aleatoriamente segmentos de paquetes nativos del sistema P_j , donde j es el número de segmento de paquete o entrada a combinar. Ver Ec2 para apreciar la combinación lineal.

$$C_i = \bigoplus_{j=1}^g v_{i,j} \cdot P_j, \forall_i \in [1, g]. \quad (\text{Ec2})$$

Luego, cada paquete codificado C_i en el nodo master es enviado al nodo gateway para su decodificación, pero para esto el nodo gateway tantas entradas codificadas, lo suficiente para completar una matriz de rango completo para calcular su inversa y recuperar los segmentos de paquetes nativos. Cada bloque codificado está definido como el producto vectorial $C = V \times P$, donde V es el vector de coeficientes generados en el espacio $GF(q)$ y P es el conjunto de paquetes seleccionados. Por tal motivo es necesario encontrar $V^{-1} - 1$ para obtener todos los paquetes nativos P . La ventaja del modelo es que es posible recuperar toda la información solo con información original intrínsecamente combinada linealmente^{21,22}. Es decir, si tenemos el dato codificado $p1' = c1p1 + c1p2$, este tiene dos grados de libertad para su decodificación, ya que si el terminal remoto tiene $c1p1$ o $c2p2$, este puede obtener el complemento a través del codificador *xor*, por ejemplo: $p1' \text{ xor } c1p1 = c2p2$ o de otra manera $p1' \text{ xor } c2p2 = c1p1$. Lo que al final permite obtener la información sin que el sistema genere una retransmisión de los datos. En la siguiente imagen podemos tener un ejemplo de cómo a través de micro servicios de codificación ofrecidos por nodos de reenvío en la red, lo cual implementado en áreas remotas y de difícil acceso es posible codificar un *streaming* de datos médicos.



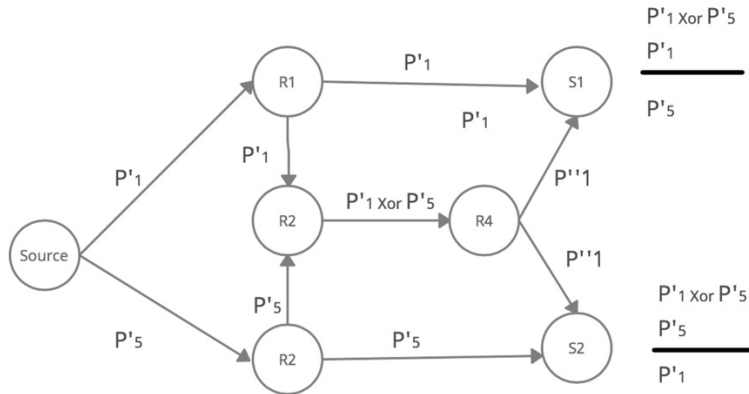


Fig 5. Codificación de datos: Coding as a Service (IoMTCaaS).

Gracias a que la codificación es definida a nivel de aplicación, cada nodo está en la capacidad de seleccionar que streaming codificar y la extensión de datos a codificar, es decir ya sea en una codificación básica $GF(2)$ o en una extensión binaria $GF(2^n)$, para $n > 1$, esto dependería de varios parámetros de la red que van desde la complejidad y las capacidades de procesamiento de cada nodo de la red hasta las codificaciones de borradura del canal.

No todos los dispositivos IoT están en capacidades de codificar extensión binaria, algunos realizarían codificaciones binarias con una baja complejidad otros realizarían codificaciones con extensiones binarias lo que se traduce en representar la misma información con más símbolos del sistema lo que es lo mismo, en generar menos información redundante.

5. Conclusión

La propuesta define una arquitectura de conectividad basada en componentes IoT para el envío y almacenamiento de datos a gran escala provenientes de sensores biomédicos para realizar el seguimiento a personas afectadas por el covid-19 en áreas alejadas y de difícil acceso. El esquema se basa en tres componentes: un parte final, en donde se realiza solamente el censado y el geo posicionamiento del paciente, un nodo master en donde se ejerce un procesamiento temporal para el empaquetamiento de los datos de diferentes nodos finales y finalmente una parte alojada en la nube, en donde se realiza el filtrado, análisis y pronóstico de los datos a través de micro servicios que se despliegan de forma dinámica. El rendimiento general del sistema de alguna manera está condicionado con técnicas de codificación basadas en Network Coding, en términos eficientes el esquema realiza combinaciones lineales para tener mayor probabilidad de decodificación en el receptor final de los datos.

6. Referencias

- Zhimin G, Zhuo L, Jiakuan F, et al. Research on security hardening technology of container image File configuration for power intelligent IoT Terminal. In: 2021 IEEE Sustainable Power and Energy Conference (ISPEC). IEEE; 2021. Accessed June 4, 2022. <https://ieeexplore.ieee.org/document/9735992/>



- Sanabria-Russo L, Pubill D, Serra J, Verikoukis C. IoT Data Analytics as a Network Edge Service. In: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE; 2019. Accessed June 4, 2022. <https://ieeexplore.ieee.org/document/8845207/>
- Cheng B, Papageorgiou A, Cirillo F, Kovacs E. GeeLytics: Geo-distributed edge analytics for large scale IoT systems based on dynamic topology. In: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). IEEE; 2015. Accessed June 4, 2022. <https://ieeexplore.ieee.org/document/7389116/>
- Rattanapoka C, Chanthakit S, Chimchai A, Sookkeaw A. An MQTT-based IoT Cloud Platform with Flow Design by Node-RED. In: 2019 Research, Invention, and Innovation Congress (RI2C). IEEE; 2019. Accessed June 4, 2022. <https://ieeexplore.ieee.org/document/8999942/>
- Kim S, Kim C, Kim J. Reliable smart energy IoT-cloud service operation with container orchestration. In: 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS). IEEE; 2017. Accessed June 4, 2022. <https://ieeexplore.ieee.org/document/8094152/>
- COVID-SAFE: An IoT-Based System for Automated Health Monitoring and Surveillance in Post-Pandemic Life | IEEE Journals & Magazine | IEEE Xplore. Accessed March 29, 2021. <https://ieeexplore.ieee.org/document/9220167>
- Ting DSW, Carin L, Dzau V, Wong TY. Digital technology and COVID-19. *Nat Med.* 2020;26(4):459-461. doi:10.1038/s41591-020-0824-5
- Roy IP, Rahman M, Hasan M, Hossain MS. IoT-based COVID-19 Suspect Smart Entrance Monitoring System. *Proc 5th Int Conf I-SMAC (IoT Soc Mobile, Anal Cloud), I-SMAC 2021.* Published online 2021:11-19. doi:10.1109/I-SMAC52330.2021.9641016
- Kwon M, Park H. Distributed Topology Design for Network Coding Deployed Large-scale Sensor Networks. Published online 2017. <http://arxiv.org/abs/1712.00631>
- Tansangworn N. Development of IoT Edge Hub for Wireless Sensor Networks based on Docker Container. In: 2020 IEEE International Conference on Smart Internet of Things (SmartIoT). IEEE; 2020. Accessed June 4, 2022. <https://ieeexplore.ieee.org/document/9191990/>
- Bedhief I, Kassar M, Aguilu T. SDN-based architecture challenging the IoT heterogeneity. In: 2016 3rd Smart Cloud Networks & Systems (SCNS). IEEE; 2016. Accessed June 4, 2022. <https://ieeexplore.ieee.org/document/7870558/>
- Zourmand A, Hing ALK, Hung CW, AbdulRehman M. Internet of Things (IoT) using LoRa technology. In: 2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS). IEEE; 2019. Accessed June 4, 2022. <https://ieeexplore.ieee.org/document/8825008/>
- T M, T A, M A V, M DK, G R. IoT and LORA Based Substation Monitoring and Control System. In: 2021 6th International Conference on Communication and Electronics Systems (ICCES). IEEE; 2021. Accessed June 4, 2022. <https://ieeexplore.ieee.org/document/9489173/>
- Leonardi L, Battaglia F, Bello L Lo. RT-LoRa: A Medium Access Strategy to Support Real-Time Flows Over LoRa-Based Networks for Industrial IoT Applications. *IEEE Internet Things J.* 2019;6(6). Accessed June 4, 2022. <https://ieeexplore.ieee.org/document/8845658/>
- Zhang C, Wang L, Jiao L, Wang S, Shi J, Yue J. A novel orthogonal LoRa multiple access algorithm for satellite Internet of Things. *China Commun.* 2022;19(3). Accessed June 4, 2022. <https://ieeexplore.ieee.org/document/9745442/>
- Nugraha AT, Wibowo R, Suryanegara M, Hayati N. An IoT-LoRa System for Tracking a Patient with a Mental Disorder: Correlation between Battery Capacity and Speed of Movement. In: 2018 7th International Conference on Computer and Communication Engineering (ICCC). IEEE; 2018. Accessed June 4, 2022. <https://ieeexplore.ieee.org/document/8539316/>
- Incháustegui CIR, Rodríguez F, Gutiérrez S. Development and Testing of Gateway LoRa for Cloudino IoT Open Source Platform. In: 2020 IEEE ANDESCON. IEEE; 2020. Accessed June 4, 2022. <https://ieeexplore.ieee.org/document/9272130/>
- Choi CS, Jeong JD, Lee IW, Park WK. LoRa based renewable energy monitoring system with open IoT platform. In: 2018 International Conference on Electronics, Information, and Communication (ICEIC). IEEE; 2018. Accessed June 4, 2022. <https://ieeexplore.ieee.org/document/8330550/>



- Wunderlich S, Gabriel F, Pandi S, Fitzek FHP, Reisslein M. Caterpillar RLNC (CRLNC): A Practical Finite Sliding Window RLNC Approach. IEEE Access. 2017;5:20183-20197. doi:10.1109/ACCESS.2017.2757241
- Wunderlich S, Gabriel F, Pandi S, Fitzek FHP. We don't need no generation- A practical approach to sliding window RLNC. 2017 Wirel Days, WD 2017. Published online 2017:218-223. doi:10.1109/WD.2017.7918148
- Julio YR, Garcia IG, Marquez J. R-IoT: An architecture based on recoding RLNC for IOT wireless network with erase channel. In: Advances in Intelligent Systems and Computing. Vol 1137 AISC. Springer; 2020:579-588. doi:10.1007/978-3-030-40690-5_55
- 22. Gabriel F, Wunderlich S, Pandi S, Fitzek FHP, Reisslein M. Caterpillar RLNC With Feedback (CRLNC-FB): Reducing Delay in Selective Repeat ARQ Through Coding. IEEE Access. 2018;6. doi:10.1109/ACCESS.2018.2865137

Los puntos de vista expresados en este artículo no reflejan necesariamente la opinión de la Asociación Colombiana de Facultades de Ingeniería.

Copyright © 2022 Asociación Colombiana de Facultades de Ingeniería (ACOFI)

