Uso de redes descentralizadas en sistemas ciber-físicos basados en la placa ESP32

Ciro Edgardo Romero†1, and Alejandro Elustondo*2

†*Dpto. de I + D + i, C&S Informática S.A.*
Buenos Aires, Argentina
[1]cromero@cys.com.ar
**XDK2MAM*
Buenos Aires, Argentina
[2]alejandro.elustondo@nakama.io

***Abstract*—Existe una tendencia en desarrollar sistemas ciber-físicos basados en las placas de desarrollo ESP32. De forma paralela, se están incorporando redes sin autoridad central que favorezcan la seguridad y la confianza en los sistemas informáticos. Eventualmente estos conceptos convergen en único sistema que presentan complejidades de desarrollo de alta incertidumbre; propios del uso de nuevas tecnologías. Este trabajo describe experiencias en el desarrollo de sistemas capaces de recolectar variables ambientales integrados con redes descentralizadas, mal llamadas redes Blockchains. Se presenta una prueba de concepto, que busca comprender los problemas a resolver cuando este tipo de proyectos se integran en una red descentralizada. Al mismo tiempo, utiliza diferentes tecnologías abiertas, utilizadas en un entorno descentralizado, y analizar la viabilidad para el desarrollo productivo.**

Palabras clave: ESP32; IOTA; Internet de las cosas; Blockchain.

I. Introducción

Hace unos años apareció un concepto llamado Internet de las Cosas (IoT), que utiliza componentes electrónicos para interactuar con servicios informaticos. Estos sistemas han sido impulsados por los avances de la potencia informática, la miniaturización electrónica y las interconexiones a través de internet [1]. Estos sistemas forman soluciones complejas, incorporando componentes informáticos digitales, capaces de interactuar directamente con el mundo que los rodea. En este tipo de desarrollo, la arquitectura del sistema define la ruta de un dispositivo a otro. Este camino determina la tolerancia a fallos del sistema y la capacidad de respuesta. Una de las topologías típicas de estos sistemas es la de cliente-servidor. Cuando se implementa este formato, los dispositivos son clientes de un sistema informático central, que auspicia como orquestador central. Este tipo de arquitectura se conoce como centralizada, y posee varias problemáticas conocidas y estudiadas [2].

II. Problemática de la centralización

Una de las problemáticas de la arquitectura clientes-servidor, es que todas las operaciones informáticas se llevan a cabo en un único dispositivo; el que funciona como servidor. Esto crea un punto crítico, donde una falla provoca que todo el sistema colapse. En sistemas IoT con este diseño, todos los datos recopilados˜ desde diferentes dispositivos están bajo la autoridad del servidor. Por tal motivo, estos elementos suelen ser objetivo de varios tipos de ataques de seguridad y privacidad. [3]. Una opción para mitigar las debilidades y vulnerabilidades pueden utilizar redes descentralizada, también conocida como DLT (por sus siglas en inglés, *Distributed Ledger Technology*).

A. Redes descentralizadas

La tecnología descentralizada resuelve (parcialmente) los problemas de seguridad que existen en un entorno público no confiable, donde dispositivos están conectados a través de internet [4]. Estos entornos son redes existentes, a las que se conectan dispositivos para acceder a Internet. La capacidad de mantener la integridad de las transacciones, elimina la necesidad de una autoridad central. Esta arquitectura ayuda a mitigar las deficiencias del modelo cliente-servidor, estableciendo una comunicación" entre iguales" para cada uno de los nodos [5]. La mejora en la tecnología DLT complementa la seguridad de sistemas IoT; tradicionalmente deficiente. Sin embardo, todavía existen algunos problemas con su adopción, como la escalabilidad, el algoritmo de consenso, la protección de datos, la eficiencia, la disponibilidad, el almacenamiento, la interoperabilidad, la estandarización, etc. Además, no existe consenso sobre modelos de referencia o mejores prácticas que definan como integrar tecnologías relevantes para cada dominio [6].

III. Adquisición de datos de forma distribuida

La implementación de un sistema ciber-físico utiliza una arquitectura que incluye interacciones, de comunicación independientes, entre diferentes elementos [7]. Con base en este concepto, se puede

entender que habrá dispositivos para tomar mediciones, y que estas serán enviadas a través de un canal de comunicación. Por otro lado, habrá dispositivos que reciban dichas mediciones, y realizaran acciones en consecuencia. En la figura 1 se muestra un esquema de posibles elementos que formarían parte de un sistema IoT.

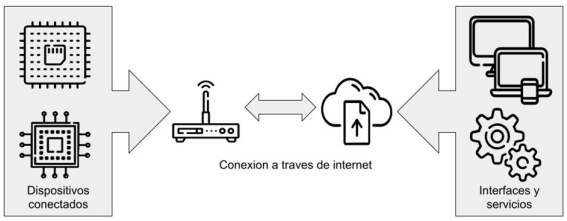

Fig. 1. Ejemplo de sistema IoT

IV. Sistemas IoT descentralizados

Desde una perspectiva teórica, los sistemas ciber-físicos pueden integrar objetos inteligentes con humanos de forma estandarizada. El objetivo del paradigma IoT es conectar operadores humanos y/o consumidores de sistemas entre sí, a través de canales seguros y confiables. Implementar una red descentralizada, es una decisión asertiva para aumentar la confianza en el diseño. Al mismo tiempo, pueda integrarse˜ con otras redes para permitir una interoperabilidad [8]. Actualmente, el salto tecnológico necesario para lograr este objetivo se logra a través de la tecnología DLT, que proporciona una red de registros única, consensuada y descentralizada [5]. Los respectivos administradores actúan como agentes confiables, verificando la identidad y las credenciales de la red. La red estructura los datos con formato Blockchain y enlaces seguros para que la información sea rastreable [9].

La implementación de la tecnología DLT conlleva su propio conjunto de desafíos. El costo asociado a cada transacción es un punto de análisis a considerar, ya que puede ser muy elevado dependiendo del tráfico que requiera la red.

A. Estructuras de datos descentralizadas

En la arquitectura tipo Blockchain, existe el conocido como algoritmo de consenso. Bajo esta política se aceptan, o no, los bloques que sean insertados dentro de la red. La necesidad de que se llegue a un consenso, antes de insertar un nuevo bloque, provoca que no se mantenga esta sincronicidad. Esto provoca" bloques huérfanos", los cuales comprometen el rendimiento general de la red [10]. Esta problemática se ve resuelta con la propuesta de *IOTA Foundation* de utilizar una arquitectura propia conocida como *The Tangle*. La misma está basada en un concepto matemático llamado Grafo Acíclico Dirigido (DAG). Dicha arquitectura diseñada para admitir la transferencia de datos y valor, mientras utiliza una estrategia para aquellos bloques sin trazabilidad. En esta estructura, los bloques" huérfanos" se fusionan nuevamente en la red [11].

Otra problemática en redes tipo Blockchain, es el tiempo que deben esperar las transacciones hasta que se incluyan en un bloque. Debido a las limitaciones en el tamaño del bloque, y al tiempo de producción de este, se crea congestión y tiempos de espera extensos. En el caso de *The Tangle* cada transacción se adjunta a transacciones cercanas. En consecuencia, el protocolo puede procesar varias cantidades de transacciones en paralelo [12]. En la figura 2 se muestra la comparativa entre una arquitectura Blockchain y por la arquitectura DAG.

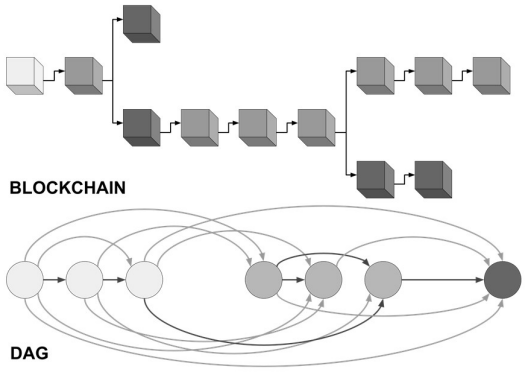

Fig. 2. Arquitectura Blockchain vs. DAG

V. Prueba de concepto

A continuación, se presenta una prueba de concepto, que intenta integrar dispositivos conectados a través internet (un sistema IoT típico) con una red descentralizada tipo Blockchain. A partir de esta experiencia se abordaron algunos de los problemas mencionados anteriormente, así como la propuesta de soluciones. El objetivo principal es dejar un registro del proceso de desarrollo IoT, con las características antes mencionadas, que sirva como base para implementaciones similares en diferentes aplicaciones.

A. Consideraciones en sistemas ciber-físicos

Al considerar un entorno descentralizado, se pueden mostrar diferentes elementos que conforman todo el conjunto de forma compleja. Desde este punto de vista, resulta interesante realizar una prueba de concepto utilizando entidades con comportamiento básico para simplificar las interacciones que se encuentran en el sistema. En el sentido más simple, un entorno descentralizado puede describirse como un sistema de IoT. Sobre esta idea, se tienen que abordar las problemáticas de:

- Capacidades limitadas en los dispositivos
- Diversidad de tecnologías coexistiendo
- Seguridad informática
- Costos económicos y de computo
- Rigurosidad en el tratamiento de datos
- Eficiencia frente al diseño

- Complejidad del sistema general

B. Sensor de recolección de datos

La placa ESP32 es una buena opción para el desarrollo de sistemas IoT. Desarrollado por *Espressif Systems*, representa una familia de microcontroladores económicos. Gracias a su bajo consumo de energía y herramientas de código abierto, es adecuado para varios tipos de implementación. Al mismo tiempo, cuenta con documentación extensa y varias comunidades activas con ejemplos de programación en C/C++, Python, entre otros lenguajes de programación [13].

El sistema explicado en el presente trabajo se basa en la experiencia obtenida por el Departamento de Investigación, Desarrollo e Innovación de la empresa C & S Informática S.A, donde se abordaron algunas de las problemáticas anteriores [14]. A partir del trabajo mencionado, se logró la comunicación entre nodos, basados en la placa ESP32, y la red descentralizada. En la figura 3 se ilustra el dispositivo desarrollado, explicado anteriormente.

Fig. 3. Prototipo de sensor con conexión a internet

Los datos recopilados por los nodos se envían a la red IOTA, diseñada para simular la comunicación típica de sensores inteligentes en un sistema descentralizado. El objetivo es desarrollar sistema que sean fáciles de mantener desde una perspectiva unitaria, utilizando lenguajes de programación de alto nivel.

C. Código *bare-metal*

Un lenguaje de programación de código interpretado, fácil de aprender y legible, es Python [15]. Este lenguaje multiparadigma es capaz de soportar programación imperativa, programación funcional y, parcial- mente, orientada a objetos. Además, es un lenguaje dinámico y compatible con diversas plataformas. En la comunidad de Python existe una reimplementación, sencilla y eficiente, optimizada para ejecutarse en microcontroladores. Este subconjunto pequeño de la biblioteca estándar, es conocido como Micropython [16].

D. Microcontrolador y sensores

Como parte del sistema IoT, se implementó un dispositivo que actúa como un nodo que recopila variables de entorno. El nodo contiene el módulo BMP180, que se basa en el sensor del mismo nombre para medir la presión, temperatura y humedad [17]. La placa realiza la configuración de sus componentes, así como los servicios diseñados para cada finalidad. Cuenta con un LED RGB, que relaciona colores con el estado de la placa. De esta forma, podrás comprobar el estado del dispositivo cuando este desconectado del

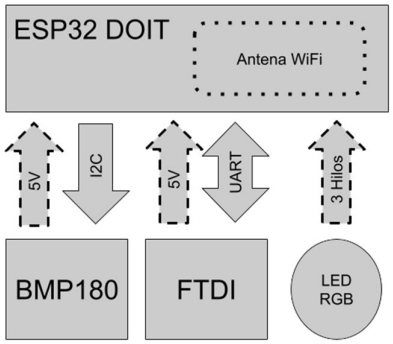

Fig. 4. Elementos que componen el nodo sensor

ordenador. En la figura 4 se muestran los componentes del nodo sensor basado en la ESP32.

El dispositivo se conecta a internet y realiza una configuración del reloj interno, a través del protocolo de tiempo de red (NTP, por sus siglas en inglés) [18]. El funcionamiento general se inicia, cuando el microcontrolador se comunica con los sensores para tomar medidas. Estos datos se almacenan en memoria y se suma un identificador único de dispositivo. Adicional- mente se recuperan la hora y fecha de medición. Antes de enviar un mensaje, se debe comprimir mediante una función criptográfica, conocida como" hash" [19]. Este es un algoritmo que convierte cualquier bloque de datos en una nueva cadena de longitud fija. Independientemente de la longitud de los datos de entrada, el valor de salida siempre tiene la misma longitud.

Finalmente, los datos se envían desde una función *request* en formado JSON. A través de este dispositivo, se restablece el funcionamiento básico del sistema de recolección y distribución de datos. Al utilizar sensores como nodos conectados a Internet, las variables físicas se pueden convertir en variables que pueden ser utilizadas por los sistemas informáticos. Estos datos luego se transmiten a la red descentralizada para que otros sistemas los utilicen.

E. Red distribuida

La propuesta de valor en un sistema distribuido, es poder crear una red topológica colectiva, de tal manera que ninguno de ellos tenga la exclusividad de filtrar la información. Siguiendo las recomendaciones de la propia comunidad IOTA, se utilizó el nodo transaccional *Hornet*. Este tipo de nodo, contiene dos funcionalidades dentro de un sistema descentralizado. En primer lugar, permite a los clientes interactuar con *The Tangle* y comunicarse con los nodos. Esto lo hace como una interfaz de envió y recepción de mensajes. Por otro lado, permite a los clientes sondear nodos en busca de nuevos mensajes y otros eventos. Esta es útil para monitorear *The Tangle* y actualizar parámetros [20].

F. Comunicación en la red

Cada nodo se puede identificar de forma individual, mediante un identificador único (del inglés *peer identity*). Las conexiones con un nodo IOTA, se establecen a través de un protocolo conocido como *peer discovery*. Este se utiliza para exponer una interfaz que proporcione una lista verificada de nodos. La implementación del protocolo proporciona una lista codificada y confiable, administrada por la comunidad y sus colaborares [21]. Estos protocolos se identifican y autentican, utilizando criptografía de clave asimétrica. Se utilizo un nodo publico provisto por la misma *IOTA Foundation*. Los dispositivos descriptos en la sección anterior, se comunican con el nodo a través de *requests* configuradas en una ruta especificada en una API intermedia, que implementa la función *hash*. En la figura 5 se muestra un diagrama para ilustrar el esquema de comunicación.

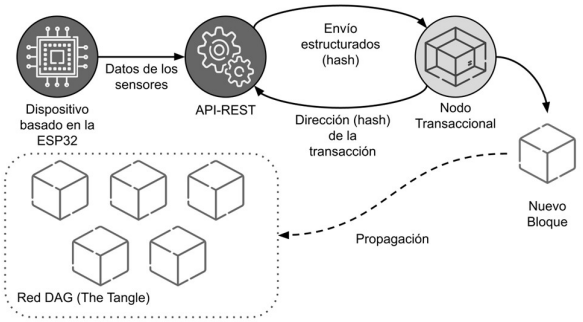

Fig. 5. Comunicación entre elementos del sistema

Al enviar el mensaje, en el caso de que falten, el nodo intentara auto completar los campos de identificación de red (networkId), el identificador de mensaje (parentMessageIds) y un número arbitrario conocido como" nonce". Si faltara el cuerpo principal del mensaje con la información (llamada *payload*, en inglés), el mensaje se generará igualmente pero vacío. A continuación, se ilustra estructura del mensaje:

```
{"networkId": "00457867963673787875",
    "parentMessageIds":  [
    "g31pxjk4q9wzmhrgptxx ( . . . )",
    "49ch624cnwtc4a7qzh88 ( . . . )",
    "q83igenu16j80013k7mj ( . . . )",
    "c4m08e672rvcr35233xf ( . . . )"
  ],
  "payload": { "type": 2
    ,
     "index":  "4 kjr7ier33rhkiwtc9y0",
     "data":  "bbwquq2kn ( . . . )"
  },
   "nonce":  "4402323"
}
```

Si la comunicación es exitosa, el mensaje se almacenará en *The Tangle*. Cuando la transacción es exitosa, se devuelve un *hash* como identificador de la transacción. El mismo puede ser utilizado por otros elementos dentro del sistema, si es necesario localizar y decodificar la información insertada.

VI. Conclusiones

La disponibilidad de placas ESP32 en el mercado hace que sea una buena opción para el desarrollo de diversos sistemas integrados. Además, la baja curva de aprendizaje de Micropython hace que este conjunto de opciones sean factores óptimos en el desarrollo de una prueba de concepto.

Al implementar proyectos productivos, los servicios integrados de transmisión de datos y cifrado proporcionan un nivel de seguridad interesante. Los sistemas ciber-físicos pueden aumentar la confianza en los usuarios del sistema.

La red IOTA reduce significativamente los costos asociados con la implementación. Esto beneficia a los desarrolladores de sistemas IoT, que utilizan placas como la ESP32 y otras con capacidades similares. El código que se comunica con redes descentralizadas se puede integrar en cualquier tipo de sistema de medición inteligente. Sus métodos son perfectamente transportables a otros lenguajes para otros microprocesadores. En otras palabras, se puede distribuir un sistema de producción ya instalado y agregar hardware y software en pequeñas cantidades.

El resultado final del trabajo realizado es un sistema mínimo perfectamente funcional capaz de escalar en implementaciones que requieran mediciones ambientales sin depender de un servidor central.

VII. Trabajo futuro

Actualmente existen microprocesadores con una capacidad de computo superior la versión de ESP32 utilizada en este trabajo. Espressif lanzo la versión ESP32-C6 mejorada. Este nuevo kit de desarrollo muestra más y mejores prestaciones que sus predecesores. Según el fabricante, es un microcontrolador ideal para aplicaciones IoT.

Por otra parte, la *IOTA Foundation* se encuentra en proceso de actualización de la red para mejorar la

integración. Promoviendo una versión a IOTA 2.0, dentro de la cual se promociona un proyecto llamado *Shimmer Network* [1]. Es ultimo permite una Inter operatividad con otras redes Blockchain, así como una extensión de sus prestaciones.

References

[1] K. Rose, S. Eldridge, and L. Chapin, "The internet of things: An overview," *The internet society (ISOC)*, vol. 80, pp. 1–50, 2015.

[2] C. Rowland, E. Goodman, M. Charlier, A. Light, and A. Lui, *Designing connected products: UX for the consumer Internet of Things*. " O Reilly Media, Inc.", 2015.

[3] M. ATT&CK, "Mitre att&ck," *URL: https://attack. mitre. org*, 2021.

[4] B. Farahani, F. Firouzi, and M. Luecking, "The convergence of iot and distributed ledger technologies (dlt): Opportunities, challenges, and solutions," *Journal of Network and Computer Applications*, vol. 177, p. 102936, 2021.

[5] J. Kehrli, "Blockchain explained," *Netguardians [en lınia].[Data de consulta: 25 de juny de 2017]¡ https://www. netguardians. ch/news/2016/11/17/blockchain-explained-part-1*, 2016.

[6] H. F. Atlam and G. B. Wills, "Intersections between iot and distributed ledger," in *Advances in Computers*. Elsevier, 2019, vol. 115, pp. 73–113.

[7] R. Hadidi, J. Cao, M. S. Ryoo, and H. Kim, "Robustly executing dnns in iot systems using coded distributed computing," in *Proceedings of the 56th Annual Design Automation Conference 2019*, 2019, pp. 1–2.

[8] R. Buyya and A. V. Dastjerdi, *Internet of Things: Principles and paradigms*. Elsevier, 2016.

[9] V. Gisbert Soler and A. I. Perez Molina, "Blockchain vs iso 9001: 2015," *3C Tecnologıa*, vol. 8, no. 2, pp. 37–48, 2019.

[10] I. Fundation, "Coordinator. part 2: Iota is a dag, not a blockchain," https://blog.iota.org/ coordinator-part-2-iota-is-a-dag-not-a-blockchain-2df8ec85200f/, 11 2018.

[11] ——, "The transparency compendium," https://blog.iota.org/ the-transparency-compendium-26aa5bb8e260/, 06 2017.

[12] W. F. Silvano and R. Marcelino, "Iota tangle: A cryptocurrency to communicate internet-of-things data," *Future Generation Computer Systems*, vol. 112, pp. 307–319, 2020.

[13] A. Maier, A. Sharp, and Y. Vagapov, "Comparative analysis and practical implementation of the esp32 microcontroller module for the internet of things," in *2017 Internet Technologies and Applications (ITA)*. IEEE, 2017, pp. 143–148.

[14] C. E. Romero, A. M. Elustondo, R. K. Der Boghosian, and M. C. Fontela, "Nodo experimental de registro e inmutabilidad de variables ambientales," in *III Simposio Argentino de Informatica Industrial e Investigacion Operativa (SIIIO 2020)-JAIIO 49 (Modalidad virtual)*, 2020.

[15] V. Frittelli, D. Serrano, R. Teicher, F. Steffolani, M. Tartabini, J. Fernandez, and G. Bett, "Uso de python como lenguaje inicial en asignaturas de programacion," *Editor Responsable*, vol. 132, 2013.

[16] N. H. Tollervey, *Programming with MicroPython: embedded programming with microcontrollers and Python*. " O'Reilly Media, Inc.", 2017.

[17] Bosch, "Bmp180," https://ae-bst.resource.bosch.com/media/ _ tech/media/product _flyer/BST-BMP180-FL000.pdf, 04 2013.

[18] IETF, "Simple network time protocol (sntp) version 4 for ipv4, ipv6 and osi," https://datatracker.ietf.org/doc/html/rfc4330, 06 2006.

[19] B. Preneel, "Cryptographic hash functions," *European Transactions on Telecommunications*, vol. 5, no. 4, pp. 431–448, 1994.

[20] I. Fundation, "Hornet. community driven iota node," https:// wiki.iota.org/hornet/welcome, 11 2021.

[21] T. Neudecker and H. Hartenstein, "Network layer aspects of permissionless blockchains," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 838–857, 2018.

[1] Detalles del proyecto: https://shimmer.network/