ISSN 2007-9737

Protocolos criptográficos de consenso en Blockchain para el Internet de las Cosas

Luis Miguel Saldaña Trejo[1], Gina Gallegos Garcia[1,*], Rocio Aldeco Peréz[2]

[1] Instituto Politecnico Nacional,
Centro de Investigación en Computación,
México

[2] Universidad Nacional Autónoma de México,
México

{lsaldanat2021, ggallegos}@cic.ipn.mx, rocio.aldeco@ingenieria.unam.edu

Abstract. El éxito que tuvo Blockchain en el campo de las criptomonedas abre la posibilidad de aprovechar las principales ventajas de esta tecnología en otras áreas como servicios financieros, servicios públicos, servicios sociales o internet de las cosas (IoT). Usar estas caracteristicas apoyan a preservar diversos servicios de seguridad como, descentralización, integridad y pseudo-anonimato e incluso tener una completa trazabilidad en todas las transacciones. El explotar estas características que ofrece Blockchain en la red IoT, genera alta expectativa en cuanto al poder de computo, almacenamiento y transacciones descentralizadas, resistentes a la manipulación. Sin embargo, la mayoría de los protocolos de consenso existentes fueron creados para ser ejecutados por nodos con alta capacidad de cómputo y almacenamiento, características que no tienen los dispositivos que conforman una red IoT. A pesar de que han surgido nuevos protocolos que exijen menor poder de cómputo o almacenamiento, ninguno está diseñado a la medida para poder ser ejecutado por nodos con recursos limitados. Con base en lo anterior, en este artículo se presenta los protocolos criptográficos de consenso que podrían ser empleados en la red IoT.

Keywords. Blockchain, protocolo de consenso, internet de las cosas (IoT).

Cryptographic Consensus Protocols in Blockchain for the Internet of Things

Abstract. The success of Blockchain in the field of cryptocurrencies opens the possibility of leveraging the main advantages of this technology in other areas such as financial services, public services, social services, and the Internet of Things (IoT). Using these features helps preserve various security services such as decentralization, integrity, and pseudo-anonymity, and even allows for complete traceability of all transactions. Exploiting these features offered by Blockchain in the IoT network generates high expectations regarding computing power, storage, and decentralized, tamper-resistant transactions. However, most existing consensus protocols were created to be executed by nodes with high computing and storage capacity, characteristics that the devices that make up an IoT network do not have. Although new protocols have emerged that require less computing power or storage, none are tailored to be executed by nodes with limited resources. Based on the above, this article presents the cryptographic consensus protocols that could be used in the IoT network.

Keywords. Blockchain, consensus protocol, Internet of Things (IoT).

1. Introducción

En esta era digital se observa un crecimiento en el surgimiento de tecnologías revolucionarias y su uso en casi todos los aspectos de la vida cotidiana. Una de estas tecnologías emergentes es el Internet de las Cosas (IoT), una red entrelazada de dispositivos inteligentes que intercambia datos. Esta tiene un gran impacto en

varios sectores como casas inteligentes, atención médica, aplicaciones industriales, transporte y logística, por mencionar algunos [13]. Sin embargo, este entrelazamiento entre dispositivos no está exento de desafíos; algunos ejemplos son la seguridad y la centralización, que han sido cuestionados en cuanto a su implementación y expansión [15], debido a que no se tiene la certeza de que todos los dispositivos que conforman esta red sean dispositivos confiables, o que, al estar basados en una red centralizada, el tener miles de transacciones pueda ocasionar un problema de 'cuello de botella'.

En este contexto de tecnologías revolucionarias, se tiene también a la tecnología blockchain, que promete abordar dichas preocupaciones de manera innovadora y efectiva [16], con la forma en que gestiona y almacena la información.

Blockchain es una tecnología de registro distribuido que permite almacenar datos de manera segura, transparente e inmutable. Funciona como una base de datos compartida que se replica en múltiples nodos de una red, garantizando que la información no pueda ser alterada sin el consenso de la mayoría de los participantes. Cada bloque en la cadena contiene un conjunto de transacciones, un sello de tiempo y un enlace criptográfico al bloque anterior, creando así una cadena continua de bloques.

Una característica esencial de blockchain es el uso del árbol de Merkle, que es una estructura de datos que permite verificar eficientemente la integridad y consistencia de los datos. El árbol de Merkle agrupa transacciones en pares y las resume en una raíz de hash, lo que facilita la comprobación de que los datos no han sido manipulados. A pesar de que blockchain puede suponer una solución a algunos de los problemas que la tecnología IoT tiene, aún existen grandes desafíos por afrontar. El integrar estas dos tecnologías no parece ser tarea fácil, debido a que el desarrollo realizado por parte de la tecnología blockchain considera que los dispositivos que participan en el consenso cuenten con poder de procesamiento y almacenamiento, características con las que no cuentan los dispositivos de una red IoT [3, 7, 26, 29] , más si tomamos en cuenta, que esta red es un conjunto de dispositivos limitados, que van desde sensores, electrodomésticos, vehículos, hasta dispositivos móviles, los cuales, claramente no cuentan con alto poder de procesamiento, almacenamiento. Además, muchos de estos dispositivos dependen de una pequeña batería como fuente de alimentación, lo que hace más difícil la idea de que un dispositivo con tales limitaciones pueda ejecutar algoritmos que exijan alto poder de procesamiento, lo que se traduce en un alto consumo energético [4].Esta red de dispositivos genera, procesa e intercambia una gran cantidad de datos. En algunas áreas específicas, la transferencia de estos datos puede ser crítica para la seguridad o indispensables para su propio funcionamiento. De hecho para tener una transferencia de datos exitosa, se requiere asegurar tanto la comunicación como la misma información, tratando de preservar servicios de seguridad como integridad, autenticación y disponibilidad.

El crear un sistema distribuido con nodos IoT parece ser la solución a los problemas que surgen en la centralización, tales como las dificultades en términos de escalabilidad, seguridad y eficiencia. Este enfoque descentralizado permite una mayor resiliencia ante fallos, distribución equitativa de la carga de trabajo y una mejor capacidad de respuesta ante diversas condiciones del entorno. En este contexto, la tarea de garantizar la seguridad, el correcto funcionamiento, la gestión de la información y la descentralización de las transacciones recae en los **protocolos criptográficos de consenso** [28], los cuales son un conjunto de procedimientos y reglas que tienen como objetivo, permitir que los participantes de un sistema distribuido, lleguen a un acuerdo sobre la validez, el orden de las transacciones y otras decisiones críticas.

Estos protocolos criptográficos, aseguran que todos los nodos de la red lleguen a un entendimiento común y que sigan las reglas prestablecidas. Sin embargo, para que estos nodos puedan llevar a cabo sus funciones, requieren el uso de primitivas criptográficas que demandan recursos computacionales intensivos, algo que con recursos limitados los dispositivos IoT no se puede lograr. Esto crea la necesidad

ISSN 2007-9737

de diseñar un protocolo de consenso que permita llegar a un acuerdo sin requerir de demasiado poder de cómputo o capacidad de almacenamiento.

Como consecuencia, este artículo, presenta los protocolos criptográficos de consenso existentes que puedan ser o no empleados para la integración entre el Internet de las Cosas y la tecnología blockchain, tomando en cuenta las limitaciones en los dispositivos, el enfoque monetario que no existe en la red IoT, y la baja latencia requerida. Nuestro trabajo también incluye la representación matemática de los protocolos seleccionados para este ambiente. Esto nos permite caracterizar su adaptabilidad a dispositivos con recursos limitados. Asimismo, se discuten brevemente opciones para mejorar estos protocolos, asegurando su compatibilidad y rendimiento óptimo en redes IoT.

El resto del documento está organizado de la siguiente manera. La Sección 2 describe el flujo general y los elementos que conforman un protocolo de consenso. En la Sección 3 se mencionan los trabajos relacionados en donde realizan el análisis de los protocolos de consenso existentes y cuál de ellos podrían ocuparse en escenarios IoT. La Sección 4 hace la distinción de los protocolos de consenso que pueden ser rediseñados para poder ser ejecutados por dispositivos limitados. En la Sección 5 se realiza un modelado de los protocolos no descartados y por último la Sección 6 muestra las conclusiones de esta revisión y trabajo a futuro.

2. Protocolos criptográficos de consenso

En el contexto de blockchain y redes distribuidas, un protocolo criptográfico de consenso se refiere a las reglas y procedimientos que rigen cómo los nodos en la red llegan a un acuerdo sobre el estado compartido del sistema.

Los protocolos criptográficos de consenso son necesarios en entornos descentralizados, donde múltiples nodos pueden proponer y validar transacciones. Es crucial tener este tipo de mecanismo para determinar cuál de las propuestas debe ser aceptada y cuál rechazada [28].

Este proceso de consenso, asegura que todos los nodos en la red tengan una visión común del historial de transacciones y son esenciales para garantizar la coherencia y la integridad de la información almacenada. Los componentes de cada protocolo pueden variar según el método de consenso y la implementación, pero en términos generales es posible decir que todos consideran lo siguiente [11, 31]: *Nodos:* La red está compuesta por múltiples participantes o nodos que intervienen en el proceso de consenso. Estos nodos pueden ser cualquier dispositivo con capacidad de cómputo y almacenamiento. *Propuesta:* Un nodo en la red inicia una propuesta, que puede incluir un conjunto de transacciones o un cambio de estado específico. *Reglas de validación:* Cada nodo aplica un conjunto de reglas predefinidas para validar las transacciones propuestas. Estas reglas suelen incluir comprobaciones de la validez de la transacción, la coherencia de los datos, la verificación de firmas digitales y cualquier otro requisito especificado por el protocolo. *Protocolo de comunicación:* Reglas para la propagación, validación y acuerdo de mensajes entre los nodos participantes. *Mecanismos de tolerancia a fallas:* Los protocolos de consenso a menudo incorporan mecanismos para manejar las fallas o el comportamiento malicioso de los participantes. Estos mecanismos permiten mantener la integridad de la red y evitan cambios no autorizados. Los ejemplos de mecanismos de tolerancia a fallas incluyen redundancia, elección de líder, sistemas de votación y sanciones por mala conducta. Estos componentes pueden variar, ya que pueden poner énfasis en distintos aspectos y presentar diferentes niveles de descentralización, seguridad, escalabilidad y rendimiento.

2.1. Funcionamiento de los protocolos criptográficos de consenso

Para abordar brevemente cómo funciona un protocolo criptográfico de consenso, se describe el siguiente flujo general que ilustra la interacción y cooperación de los nodos en una red blockchain.

Cada paso en este proceso cumple un papel en la seguridad y confiabilidad del sistema distribuido, permitiendo un consenso descentralizado. Aunque los detalles específicos pueden variar según el protocolo de consenso, el proceso general de cada nodo incluye la validación de las transacciones propuestas, la generación del bloque y la obtención del consenso.

La validación de transacciones propuestas es un paso fundamental en los protocolos de consenso. Cada nodo en la red realiza este proceso para garantizar la validez y la integridad de las transacciones. Generalmente este proceso implica los siguientes pasos:

– Validación de la firma digital: Cada transacción incluye una firma digital creada con la clave privada del remitente. Los nodos verifican esta firma para asegurarse de que la transacción provenga de la entidad autorizada y que no haya sido alterada durante la transmisión.

– Disponibilidad de fondos (si aplica): Se verifica que el remitente tenga fondos suficientes para realizar la transacción. Esto implica verificar el saldo de la cuenta y asegurarse de que sea igual o mayor al monto de la transacción más la recompensa.

– Reglas del protocolo: Los nodos aplican las reglas específicas del protocolo para verificar la validez de la transacción. Esto puede incluir asegurarse de que la transacción no duplique gastos (double-spending), cumplir con requisitos de formato específicos, o seguir otras reglas definidas por el protocolo de consenso.

– Coherencia en las transacciones: Se revisa el historial para verificar que las transacciones anteriores del remitente sean válidas. Esto garantiza que la transacción propuesta sea compatible con las transacciones pasadas.

– Consistencia de datos: La consistencia de los datos dentro de la transacción también se verifica, asegurándose de que los campos obligatorios estén presentes y que los datos sean coherentes con las reglas del protocolo.

Una vez que un nodo ha completado exitosamente estos pasos de validación, la transacción es considerada válida y puede ser propuesta para su inclusión en un bloque [6, 8].

La elección del nuevo bloque, depende completamente de las reglas de cada protocolo de consenso, como:

– Difusión y recepción: El bloque propuesto se difunde a la red, y todos los nodos reciben la propuesta. Esto asegura que cada nodo tenga conocimiento de la transacción propuesta.

– Validación del bloque: Los nodos validan independientemente la propuesta del bloque y la comparan con su copia local de *blockchain*. Verifican la consistencia de las transacciones y la integridad del bloque.

– Consenso: La red busca un acuerdo mayoritario sobre la validez del bloque propuesto. Esto puede implicar la participación en un proceso de votación, la resolución de un problema matemático, o seguir algún mecanismo predefinido por el protocolo de consenso.

– Actualización: Una vez obtenido el consenso, todos los nodos actualizan su estado local del sistema, de acuerdo con la nueva información incluida en el bloque aceptado.

Este proceso de consenso descentralizado, garantiza que todos los participantes en la red tengan una visión común y acordada del historial de transacciones. La resistencia a la manipulación y la descentralización inherentes al consenso, son esenciales para la seguridad y la confiabilidad de una red descentralizada [5].

Otra característica importante de los protocolos de consenso es tener tolerancia a fallas [32], refiriéndose a tener la capacidad del sistema para mantener su funcionamiento normal, incluso cuando algunos de sus nodos o componentes fallan o se comportan de manera inesperada. En contextos de consenso, la tolerancia a fallos es crucial para garantizar la robustez y la confiabilidad de la red.

Algunos protocolos específicos, como los algoritmos de tolerancia a fallas bizantinas *(Byzantine Fault Tolerance)*, están diseñados

específicamente para garantizar la tolerancia a fallos en entornos distribuidos, donde algunos nodos pueden actuar de manera maliciosa.

Todas estos procesos, pueden ser personalizados por cada protocolo, en dependencia del objetivo con el que fue diseñado, el nivel de descentralización y la manera en conseguir el consenso entre los participantes.

Dado que la mayoría de los protocolos existentes requieren altos recursos, hasta la fecha se han desarrollado diversas variantes de protocolos de consenso, con el objetivo de mejorar los procesos y reducir el consumo de recursos.

El proceso descrito con anterioridad no es aplicable a IoT, la necesidad de un protocolo de consenso diseñado específicamente para este ambiente, se deriva de las características únicas y desafíos que presentan estos dispositivos [9, 22, 30], como:

– Limitaciones de recursos: Dado que la mayoría de los protocolos existentes requieren altos recursos, hasta la fecha se han desarrollado diversas variantes de protocolos de consenso, con el objetivo de mejorar los procesos y reducir el consumo de recursos.

– Eficiencia energética: La eficiencia energética es una consideración crítica para los dispositivos IoT, que a menudo funcionan con fuentes de energía limitadas o baterías. Un protocolo de consenso, diseñado para dispositivos IoT, debe minimizar el consumo de energía para garantizar la vida útil y prolongada de la batería y reducir la necesidad de recargas frecuentes.

– Ancho de banda limitado: Muchos dispositivos IoT operan en entornos con limitaciones significativas de ancho de banda. Los protocolos de consenso deben ser eficientes en la transmisión de datos para minimizar la carga en las redes de comunicación, especialmente en entornos con conectividad limitada.

– Latencia baja: Algunas aplicaciones de IoT requieren respuestas rápidas y baja latencia. Los protocolos de consenso deben ser capaces de llegar a decisiones rápidas y eficientes, para cumplir con los requisitos de tiempo real en estas aplicaciones.

– Diversidad de dispositivos y arquitecturas: Los dispositivos IoT pueden variar considerablemente en términos de arquitectura, de hardware y sistemas operativos. Un protocolo de consenso para IoT debe ser compatible con una amplia gama de dispositivos y sistemas, ofreciendo flexibilidad en su implementación.

En resumen, un protocolo de consenso diseñado para dispositivos IoT con bajos recursos computacionales, tendría que abordar específicamente los desafíos de eficiencia, escalabilidad y adaptabilidad en entornos donde los recursos son limitados, permitiendo la implementación exitosa de tecnologías de *blockchain*, aprovechando sus beneficios sin comprometer la eficiencia y la viabilidad operativa de estos dispositivos de baja potencia [12, 23].

3. Trabajos relacionados

En esta sección, se discuten los trabajos relacionados que hacen una revisión de los protocolos de consenso existentes, enfatizando las dificultades para emplear dichos protocolos en ambientes IoT.

El trabajo de Salimitari et al. [19] aborda la aplicación de la tecnología *blockchain* en redes IoT, centrándose en los protocolos de consenso que requieren alto poder computacional y mostrando que no son adecuado para dispositivos IoT con recursos limitados. Exploran medidas para reducir la carga computacional y el tiempo de convergencia. Este trabajo reconoce la necesidad de un enfoque híbrido o una modificación en los protocolos de consenso existentes para lograr una implementación exitosa de *blockchain* en redes IoT a gran escala y con baja latencia. En otro trabajo de Salimitari et al. [21] destacan la necesidad de abordar los desafíos específicos de aplicar la tecnología *blockchain* en entornos de IoT.

Así proponen un protocolo de consenso de 2 pasos (detección de valores atípicos y PBFT), resaltando que la implementación de *blockchain*

ISSN 2007-9737

dispositivos IoT es un desafío no trivial debido a las limitaciones de recursos. En general, el artículo presenta una perspectiva positiva sobre la combinación de *blockchain* e inteligencia artificial para abordar los desafíos específicos de las redes IoT. Complementando esta idea y destacando la importancia de *Blockchain* en aplicaciones de IoT, Raghav et al. [17] realizan un análisis de su propuesta de protocolo de consenso *(PoEWAL)*, el cual es la combinación del protocolo *Proof of Authority (PoAu)* y *Proof of Stake*, que tiene como objetivo reducir la cantidad de energía requerida para obtener el consenso. Este trabajo enfatiza la importancia de la eficiencia energética y la baja latencia requerida para este tipo de escenarios. Para poder garantizar la descentralización e inmutabilidad en la recopilación y compartición de datos, Uddin et al. [25] destaca los desafíos y oportunidades al integrar Blockchain con IoT, mencionando la preocupación por la seguridad de los datos de IoT y la necesidad de abordar problemas de escalabilidad. El trabajo presenta CBCIoT como un protocolo de consenso eficiente en términos de tiempo de generación de bloques y transacciones por segundo, y aunque presenta un protocolo de consenso prometedor, sugiere que aún hay desafíos por abordar, por lo que enfatiza la necesidad de futuras investigaciones, volviéndose escencial la mejora continua de la integración de *Blockchain* y IoT. Wu et al. [30] proponen un mecanismo de consenso para dispositivos IoT, destacando la necesidad de un enfoque ligero y de alto rendimiento. Proponen el uso de HMAC y la Función Aleatoria Verificable (VRF) para implementar la elección rápida y fuera de línea de los nodos de bloque. Adicionalmente, señalan desafíos actuales y destacan la importancia de seleccionar mecanismos criptográficos adecuados para dispositivos IoT, evitando operaciones criptográficas complejas que puedan sobrecargar estos dispositivos.

Otra propuesta diferente se da en el trabajo de Anita et al. [1] en donde se propone un protocolo de consenso llamado Proof-of-Improved-Participation (PoIP), que busca mejorar la autenticación del protocolo de consenso empleando un mecanismo de autenticación de control de acceso para sustituir algunos procesos de PoW y PoS.

Nuevamente Salimitari et al. [18] destacan que las limitaciones de recursos en dispositivos IoT hacen que la aplicación directa de protocolos de consenso tradicionales sea difícil, dado que los protocolos actuales a menudo requieren alto poder computacional. Destacan la necesidad de adaptar o desarrollar protocolos de consenso que sean adecuados para dispositivos IoT, para satisfacer las demandas específicas de estos entornos.

Por otro lado el trabajo [20] realiza una revisión de las limitaciones actuales de las redes IoT, destacando la aplicabilidad de blockchain para abordar estas limitaciones. Se discuten los métodos de consenso y se clasifican según su idoneidad para las redes IoT.

En esta serie de artículos, los autores abordan el tema de la aplicación de tecnología blockchain en entornos IoT, convergiendo en que existen muchos desafíos en integrar Blockchain en IoT. Además, mencionan que los protocolos de consenso actuales no pueden ser usados en su forma simple y enfatizan la importancia de contar con protocolos ligeros y de alto rendimiento, ya sea proponiendo un protocolo híbrido o realizando modificaciones a los existentes, logrando una implementación exitosa a gran escala y con baja latencia.

Por esta razón, en este trabajo se retomó y se complementó el análisis de los protocolos de consenso, descartando los protocolos que presentan más dificultades al ser empleados en el entorno IoT, y enfocándose en aquellos que ofrecen mejores soluciones en términos de eficiencia y que se pueden emplear de manera efectiva en dispositivos con recursos limitados.

4. Análisis de protocolos de consenso

La elección del protocolo de consenso juega un papel crucial en la seguridad, eficiencia y viabilidad de una blockchain, y su selección se vuelve aún más crítica en entornos de IoT. En este contexto, este trabajo se centra en el análisis detallado de varios protocolos de consenso,

con un enfoque especial en su idoneidad para escenarios IoT. Entre los protocolos de consenso más prominentes, se encuentra el *Proof of Work (PoW)*, que aunque es el más usado, presenta desafíos significativos en términos de eficiencia energética y escalabilidad. Por esta razón, el análisis, también explora protocolos como *Proof of Stake (PoS), Delegated Proof of Stake (DPoS)*, y otros protocolos emergentes que buscan atender dichas áreas de oportunidad. El objetivo principal de este trabajo es presentar aquellos protocolos de consenso que, dadas sus características y requisitos, podrían no ser óptimos para la implementación en entornos IoT, que a menudo enfrentan limitaciones de recursos y demandas específicas de rendimiento.

El análisis detallado de estos protocolos, permite una comprensión más profunda de sus fortalezas y limitaciones, lo que a su vez facilita la identificación de soluciones de consenso más adecuado para aplicaciones de IoT.

El primer protocolo que se puede mencionar como inapropiado para entornos IoT es *Proof of Work (PoW)* , ya que exige que los nodos que conforman la red posean una gran capacidad de procesamiento para participar en el proceso de consenso.

Proof of Capacity(PoC) es un protocolo de consenso que difiere de PoW, ya que implica demostrar la posesión de espacio de almacenamiento, en lugar de resolver problemas criptográficos. Aunque PoC puede ser adecuado para ciertos contextos, su implementación en escenarios IoT puede resultar poco práctico, debido a las limitaciones de almacenamiento en dispositivos, desafíos de escalabilidad y posibles problemas de competencia desleal [2].

Stellar Consensus Protocol (SCP) fue diseñado principalmente para aplicaciones enfocadas en el sector financiero, particularmente en casos que se involucran transferencia de activos digitales. Parte del objetivo de este protocolo es incrementar el rendimiento y disminuir la demanda de recursos computacionales. Sin embargo, en varias aplicaciones IoT, se busca garantizar baja latencia, por lo que existe la interrogante de, si este tipo de protocolo sería apto para ambientes IoT. En un caso hipotético, donde la aplicación IoT pueda tolerar alta latencia, sigue siendo inviable debido a que el proceso consenso de SCP involucra operaciones criptográficas e intercambio de gran cantidad de información [18].

Ripple/Ledger XRP (XRPL) es un protocolo de consenso diseñado para el área financiera. Si bien, demanda menos recursos que PoW y PoC, puede que no sea el protocolo de consenso más recomendado para aplicaciones IoT. El consenso en este protocolo ejecuta múltiples rondas de votación entre validadores para acordar el orden y la validez de las transacciones. Este proceso, introduce una sobrecarga en términos de tiempo de confirmación de la transacción que puede no ser adecuado para ambientes IoT en tiempo real [18].

BizCoin es un protocolo utilizado en ciertos sistemas, el cual emplea una variante del PBFT que ofrece ciertas ventajas en términos de seguridad y confiabilidad. Sin embargo, la literatura nos indica que emplea operaciones criptográficas y que suele existir sobrecarga de comunicación, lo que no es ideal para ambientes IoT [18].

Byzantine Agreement Protocol (BAP) es un protocolo de consenso empleado por Algorand, criptomoneda que es conocida por ofrecer un alto grado de descentralización mediante el uso del protocolo PoS y una función aleatoria verificable *(VRF)*. La criptográfia empleada en VRF puede hacer un uso intensivo del procesador y puede ser un problema cuando se trabaja con dispositivos con recursos limitados como lo son los dispositivos IoT. Otra limitante sería la latencia, esto debido a que el proceso de consenso de Algorand implica múltiples rondas de comunicación y votación entre nodos [2].

Dfinity es una red que propone un protocolo de consenso utilizando VRF (Funciones Aleatorias Verificables) para alcanzar el consenso, lo que requiere una gran capacidad de procesamiento. Esto puede presentar limitaciones significativas en dispositivos con recursos limitados, especialmente a medida que la red crece [2].

Dentro de las redes distribuidas existe la técnica llamada fragmentación que consiste en dividir los gastos generales del procesamiento de transacciones entre múltiples grupos más

ISSN 2007-9737

pequeños de nodos llamados fragmentos, esta técnica es usada por el protocolo *RSCoin*, que es un protocolo de consenso respaldado por un banco central o gobierno, que no es tolerante a fallas bizantinas, es susceptible a ataques de doble gasto y que no está descentralizado como se afirma, ya que depende de una fuente confiable de aleatoriedad para la ya comentada fragmentación, todo esto lo hace no ideal para entornos IoT [18].

OnmiLedger tambien usa fragmentación de blockchain donde cada fragmento opera de forma independiente, procesando sus transacciones y contratos inteligentes. Fue propuesto para abordar algunas limitaciones de escalabilidad, sin embargo, la latencia, problemas de seguridad, y la sobrecarga de comunicación no permite que se empleado en dispositivos IoT con recursos limitados. [18].

Otro ejemplo es *Rapidchain* requiere que cada nodo participe en la resolución de un acertijo criptográfico, lo que implica un alto consumo de energía y recursos computacionales. Además, depende de la sincronización de los nodos para garantizar la seguridad y la descentralización, sin embargo los dispositivos IoT pueden tener una conectividad intermitente o suelen desconectarse frecuentemente lo cual haría muy difícil dicha sincronización.

Con base en la descripción e identificación de las desventajas de los protocolos en términos de aplicabilidad en el IoT, se puede concluir que cada uno de estos protocolos presentados en gris en la *Tabla 1* presentan limitaciones específicas y desafíos que los hacen inadecuados para escenarios IoT. Estas limitaciones pueden incluir problemas de consumo energético elevado, alta demanda de espacio de almacenamiento, problemas de escalabilidad, donde los protocolos no pueden manejar eficientemente el volumen masivo de dispositivos conectados, o carencias en la seguridad, lo que resulta crítico cuando se trata de proteger la integridad de los datos sensibles. En este contexto, se realizó una segunda revisión a los protocolos de consenso restantes marcados en blanco en la *Tabla 1*, con el fin de evaluar su viabilidad y adaptabilidad en el entorno dinámico y exigente del IoT.

Table 1. 1era revisión de protocolos de consenso

Protocolo de consenso	Compatible con IoT	¿Porque no es compatible?
PoW	x	Nodos sin capacidad de procesamiento
PoC	x	Nodos sin capacidad de almacenamiento
PoET	✓	-
PoS	✓	-
DPoS	✓	-
LPoS	✓	-
PoI	✓	-
PoA	✓	-
PoB	✓	-
PBFT	✓	-
dPBFT	✓	-
Stellar	x	Alta latencia Nodos sin capacidad de procesamiento
Ripple/XRP	x	Alta latencia
Byzcoin	x	Alta latencia
BAP-Algorand	x	Nodos sin capacidad de procesamiento
Dfinity	x	Nodos sin capacidad de procesamiento
RSCoin	x	No es tolerante a fallas bizantinas
OnmiLedger	x	Alta latencia
Rapidchain	x	Intermitencia en conexión
Raft	✓	-

Para la segunda revisión se consideraron protocolos que prometen no exigir un alto poder de procesamiento o almacenamiento, tales como, *Proof of Stake (PoS)* que es un protocolo que promete no demandar recursos computacionales, pero con el riesgo de centralización y que asume la posesión de un token para participar en la validación de las transacciones. Esto en redes IoT genera el problema conocido como “nothing at stake” [14], que se refiere a la situación, en la que un nodo seleccionado no tiene nada que perder si tiene un mal comportamiento, en otras palabras, este protocolo de consenso requiere un concepto monetario algo que no existe en ambientes IoT. *DPoS y LPoS* tienen el mismo problema, razón por la cual, son protocolos de consenso poco recomendables para redes IoT [19].

Proof of Importance (PoI) asigna ïmportancia.[a] los participantes de la red en función de varios factores, como la cantidad de tokens que poseen, su actividad en la red y su historial de transacciones. Como es un protocolo que se basa en la actividad, implica un alto consumo de ancho de banda y de recursos computacionales, además que, similar al caso anterior necesita tokens para su funcionamiento [19].

Proof of Authority (PoA) es un protocolo que se basa en la reputación y la identidad de los nodos validadores, que son elegidos a través de un

conjunto de reglas predefinidas. Esto implica una pérdida de anonimato y privacidad para los nodos validadores, lo que puede exponerlos a ataques físicos o cibernéticos. Además, el consenso recae en un grupo específico de nodos, reintroduciendo el concepto de centralización [19].

Proof of Burn (PoB) es un protocolo de consenso en el que los participantes ”queman”tokens, enviándolos a direcciones irreversibles, para demostrar su inversión en la red. Aunque este protocolo puede ser útil en ciertos contextos, su implementación en entornos IoT podría ser impráctica debido a la destrucción de recursos. Además, al igual que los protocolos mencionados anteriormente, requiere el concepto monetario [10].

(dPBFT) es un protocolo que tiene algunos procedimientos como PBFT pero no requiere la participación de todos los nodos para obtener el consenso, mejorando su escalabilidad. Sin embargo, en el proceso de votación para la elección de los delegados, puede representar latencia e introducir cierto grado de centralización.

(Raft) es un protocolo basado en votación que no es óptimo para entornos IoT. Aunque este método tiene un alto rendimiento y baja latencia, su desempeño depende del nodo líder, que ocupa un dominio absoluto en el sistema. Esto podría hacer que el sistema sea vulnerable si el nodo líder falla o se convierte en un cuello de botella.

Table 2. 2da revisión de protocolos de consenso

Protocolo de consenso	Compatible con IoT	¿Porque no es compatible?
PoET	✓	-
PoS	x	Requiere concepto monetario
DPoS	x	Requiere concepto monetario
LPoS	x	Requiere concepto monetario
PoI	x	Requiere concepto monetario
PoA	x	Centralización
PoB	x	Requiere concepto monetario
PBFT	✓	-
dPBFT	x	Latencia Centralización
Raft	x	Centralización

En esta segunda evaluación, se descartan los protocolos de consenso que, a pesar de su baja demanda de recursos computacionales, se basan en tokens o en una economía monetaria. También, se descartan aquellos que pueden inducir alta latencia. Los protocolos que superaron estos criterios que se detallan en la *tabla 2* marcados en blanco, serán objeto de análisis en el capítulo siguiente.

5. Modelado de protocolos seleccionados

Después de la revisión de los protocolos de consenso, se puede concluir que dos de los protocolos que podrían ser empleados son PoET y PBFT. Por lo tanto, en este capítulo ahondaremos en estos protocolos y realizaremos el modelado correspondiente para evaluar su viabilidad en diferentes entornos.

Table 3. Notación protocolo PoW

Simbolo	Definición
E	Propietario
H	Función Hash
pk	Llave pública
sk	Llave privada
$\rightarrow$	Distribución
S	Firma
N_v	Nodos validadores
N_g	Nodo generador
N_T	Nodos de la red
V	Verificación
t	Estampa de tiempo
v	Versión del bloque
hB	Encabezado
rM	Cuerpo del bloque
target	Dificultad
nonce	Valor buscado
B_i	Bloque actual
B_{i-1}	Bloque anterior

5.1. Proof of work

Como se mencionó previamente, PoW es un protocolo de consenso utilizado en blockchain, conocido por su alto nivel de descentralización y seguridad. Mediante el proceso de minería, los participantes compiten para resolver problemas matemáticos complejos, garantizando así la

ISSN 2007-9737

integridad de las transacciones. A continuación, se describirán las acciones y características específicas del protocolo PoW que se tomarán como base para modelar los protocolos PBFT y PoET.

La interacción del usuario con el protocolo de consenso comienza con la preparación de la transacción. El propietario firma la información utilizando su clave privada y luego la transmite a la red (1). Esta información es recibida por los nodos validadores:

$$\begin{aligned} E &: h = H(info), \\ E &: S_E = S_{skE}(h), \\ \forall N_v \in N_T &: E \rightarrow Nv : (info, S_E). \end{aligned} \tag{1}$$

El grupo de nodos validadores tienen la tarea de validar la firma del propietario de la transacción, al igual que validar la veracidad de la información. Si ambas validaciones son correctas, esta transacción es considerada como válida y es enviada al conjunto de transacciones (2), para que una vez que se haya seleccionado el nodo generador, empiece a realizar el proceso para obtener el siguiente bloque:

$$\begin{aligned} &\forall N_v : V = ver_{pkE}(S_E), \\ &\forall N_v : V = ver_h(info). \end{aligned} \tag{2}$$

Dependiendo de la configuración, ya sea por tiempo o por número de transacciones, los nodos comienzan a competir por ser quien genere el siguiente bloque aceptado por la red, los nodos comienzan con la creación del bloque formando su encabezado, el cual incluye el hash resultante de combinar la estampa de tiempo, la versión del bloque y el hash del bloque previo. En cuanto al cuerpo del bloque, se construye a partir del árbol de Merkle, que resume todas las transacciones validadas.

Una vez definidas las estructuras del encabezado y del cuerpo, los nodos proceden a iterar el valor del *nonce* utilizando el algoritmo Hashcash, hasta encontrar el hash que cumpla con el *target* establecido por la red (3).

Cuando el primer nodo obtiene el *nonce*, procede a generar el bloque, firmándolo con su llave privada, y lo envía al resto de los nodos para su validación (4):

$$\begin{aligned} &N_g : hB = H(t|v|B_{i-1}) \\ &N_g : rM = H(rootMerkle_x) \\ &while(B_i > target) \\ &\quad N_g : B_i = H(rM|hB|nonce) \\ &\quad\quad nonce++ \\ &break, \end{aligned} \tag{3}$$

$$\begin{aligned} N_g &: S_g = S_{skg}(B_i), \\ \forall N_v \in N_T &: N_g \rightarrow N_v : (B_i, S_g). \end{aligned} \tag{4}$$

Para alcanzar el **consenso**, es necesario que los nodos validen que el bloque inicial recibido fue generado correctamente y que proviene del nodo autorizado (5).

Si las verificaciones son correctas, esta es añadida a su propia copia de la cadena de bloques y comparte su respuesta al resto de los nodos, para propagar el nuevo bloque.

$$\begin{aligned} &\forall N_v : V = ver_{pkg}(S_g), \\ &\forall N_v : V = ver_B(B_i), \\ &\forall N_v \in N_T : N_v \rightarrow N_{v+1} : (B_i). \end{aligned} \tag{5}$$

Aunque la operación de búsqueda del *nonce* es relativamente simple, su ejecución repetitiva por parte de los nodos resulta en una ineficiencia energética considerable. Esta situación obliga a los nodos a aumentar su capacidad de procesamiento para ser competitivos, especialmente frente a nodos con mayores recursos. Por esta razón, el protocolo PoW es conocido por su alto consumo energético y su tendencia a la centralización de la minería en grandes grupos.

Table 4. Notación protocolo PBFT

Simbolo	Definición
E	Propietario
H	Función Hash
pk	Llave pública
sk	Llave privada
$\rightarrow$	Distribución
S	Firma
N_p	Nodo principal
N_j, N_k	Nodos réplica
N_T	Nodos de la red
V	Verificación
v	Vista
B_x	Bloque

5.2. Practical Byzantine Fault Tolerance

PBFT es un protocolo de consenso tolerante a fallas bizantinas diseñado para lograr el consenso en un sistema distribuido, incluso cuando algunos de los nodos participantes pueden presentar fallas bizantinas, como enviar mensajes contradictorios, omitir mensajes o enviar diferentes mensajes a diferentes nodos.

El funcionamiento de este protocolo se inicia cuando el propietario prepara la transacción, firmando el hash de la información y enviándolo al nodo principal (6):

$$\begin{aligned} E : h &= H(info), \\ E : S_E &= S_{skE}(h), \\ E &\rightarrow N_p : (info, S_E, op, t, id_E). \end{aligned} \tag{6}$$

Después de cambiar la vista, se elige un nuevo nodo principal, quien se encargará de validar la firma del propietario (7):

$$N_p : V = ver_{pkE}(S_E). \tag{7}$$

Si la firma es incorrecta, la transacción es rechazada. Si es correcta, se envía al conjunto de transacciones válidas. Dependiendo de la configuración, el nodo principal puede generar el bloque con una o varias transacciones, asigna un número de secuencia para organizarlas, establece la vista correspondiente para indicar que está llevando a cabo la generación y firma digital del bloque con su llave privada (8):

$$\begin{aligned} &gen(B_x), \\ N_p : S_{B_p} &= S_{sk_p}(H(B_x)), \\ &pre-prepare \\ \forall N_j \in N_T : N_p &\rightarrow Nj : (B_x, H(B_x), S_{B_p}, v). \end{aligned} \tag{8}$$

Los nodos réplica validan la firma del nodo principal, la vista, y la integridad del mensaje (9):

$$\forall N_j : V_p = Ver_{pk_{N_p}}(B_x, H(B_x), S_{B_p}, v), \tag{9}$$

Si la verificación de todos los datos es satisfactoria, cada nodo réplica firma el bloque y lo comparte con los demás nodos réplica, lo que representa el envío del mensaje *prepare* (10):

$$\begin{aligned} N_j : S_{B_j} &= S_{sk_{N_j}}(H(B_x)), \\ &prepare \\ \forall N_k \in N_T : N_j &\rightarrow Nk : (B_x, H(B_x), S_{B_j}, v). \end{aligned} \tag{10}$$

Cada nodo réplica verifica la propuesta y espera recibir al menos 2n+1 mensajes 'prepare' de otros nodos, lo que indica que han validado la propuesta. Una vez que un nodo ha recibido suficientes mensajes 'prepare', verifica que todos sean consistentes y, si es así, envía su propio mensaje 'commit' a los demás nodos, confirmando que acepta la propuesta y está listo para ejecutar la transacción (11):

$$\begin{aligned} &\forall N_k : V_{q_j} = Ver_{pk_{N_j}}(B_x, H(B_x), S_{B_j}, v), \\ &\sum_{j=1}^{N_T} V_{q_j} \geq 2f+1 \text{ ,entonces quórum} \\ &commit \\ \forall N_y \in N_T : N_k &\rightarrow Ny : (H(B_x), v). \end{aligned} \tag{11}$$

Cada nodo réplica verifica y espera recibir *2n+1* mensajes de confirmación de otros nodos que contengan los mismos valores, al recibir estos mensajes, el nodo entra en estado confirmado.

Posteriormente, cada nodo envía una respuesta al nodo principal, informándole que la operación se ha llevado a cabo con éxito y añaden el nuevo bloque a su copia (12):

$$\begin{aligned}&\forall N_y : V_{q_k} = Ver_{pk_{N_k}}(H(B_x), v),\\&\sum_{k=1}^{N_T} V_{q_k} \geq 2f+1 \text{ ,entonces quórum}\\&\forall N_y : N_y \rightarrow Np : (Succes),\\&\quad add(B_x).\end{aligned} \tag{12}$$

Este protocolo puede alcanzar el consenso sin requerir un excesivo poder computacional, lo que lo hace adecuado para entornos donde los recursos son limitados.

5.3. Proof of Elapsed Time

PoET es un protocolo de consenso que emplea entornos de ejecución confiables (TEE) para determinar un líder aleatorio. En este protocolo se realiza un proceso de autenticación para poder participar en la contienda en la generación del bloque.

Table 5. Notación protocolo PoET

Simbolo	Definición
soft	TEE
pk	Llave pública
sk	Llave privada
$\rightarrow$	Distribución
S	Firma
N_p	Nodo
N_v	Nodos validadores
V	Verificación
a	Atestación
t	Tiempo de espera
a_t	Tiempo cumplido
cert	Certificado
B_i	Bloque actual

Para la autenticación en la red, cada nodo participante instala el software que le permite generar una atestación firmada digitalmente (13):

$$\begin{aligned}&N_p : dowload(soft),\\&gen(a),\\&soft : S_s = S_{sksoft}(a).\end{aligned} \tag{13}$$

Después de que el nodo ha generado la atestación, el siguiente paso consiste en firmarla digitalmente y transmitirla a los nodos validadores. La firma digital garantiza la autenticidad y la integridad de la atestación (14):

$$\begin{aligned}&N_p : S_a = S_{skN_p}(a),\\&\forall N_v \in N_T : N_p \rightarrow N_v : (a, S_a).\end{aligned} \tag{14}$$

Una vez que los nodos validadores reciben la atestación firmada, la verifican para permitirle al nodo poder participar en el proceso de consenso (15):

$$\forall N_v : V = ver_{pkN_p}(a, S_a). \tag{15}$$

El software descargado previamente crea un temporizador específico para ese nodo. La configuración de este temporizador depende de dos factores clave: la dificultad de la red y la cantidad total de nodos que participan (16):

$$\begin{aligned}&gen(t),\\&soft : S_t = Ssksoft(t).\end{aligned} \tag{16}$$

Los nodos esperan el tiempo asignado como parte del proceso de consenso (17). Una vez transcurrido este período, el software valida la espera. Si se cumple con éxito, se genera un certificado para el nodo, otorgándole el derecho a convertirse en el nodo generador en el proceso de consenso (18):

$$\begin{aligned}&N_p : V = ver_{pksoft}(S_t)\\&while(t < gettime)\\&\quad N_p : delay()\\&break,\end{aligned} \tag{17}$$

$$\begin{aligned} soft : V = ver_{time}(a_t), \\ gen(cert), \\ soft : S_c = S_{sksoft}(cert), \\ soft \rightarrow N_p : (cert, S_c). \end{aligned} \tag{18}$$

Una vez que el nodo ha recibido el certificado que le autoriza a generar el bloque, elabora la propuesta de bloque y la comparte con los demás nodos, incluyendo el certificado (19):

$$\begin{aligned} gen(Bi), \\ \forall N_v \in N_T : N_p : \rightarrow N_v : (cert, B_i). \end{aligned} \tag{19}$$

El consenso se alcanza cuando la mayoría de los nodos validadores pueden verificar la autenticidad tanto del certificado como del bloque generado:

$$\forall N_v : V_i = Ver_{pk_{N_i}}(cert, B_i). \tag{20}$$

Después de analizar y representar formalmente los protocolos de consenso como PoW, PBFT y PoET, obtenemos una nueva perspectiva de su funcionamiento interno y de los pasos necesarios para alcanzar el consenso. En cada uno de estos pasos, se emplean diversas primitivas criptográficas para firmas digitales, cifrado de información, garantía de integridad y validación. Estas primitivas podrían ser reemplazadas por algoritmos criptográficos más ligeros, que no demanden alto poder de procesamiento pero que sigan garantizando la seguridad sin alterar el proceso requerido [27]. Además, como se muestra en trabajos relacionados, la combinación de protocolos existentes podría ser una alternativa viable. Las propuestas actuales que combinan protocolos para aprovechar las mejores características de cada uno, como la elección de nodo con un protocolo y el consenso con otro, revelan que aún existen áreas de oportunidad y que cada protocolo aborda algunas deficiencias en la integración de Blockchain e IoT.

En cuanto al almacenamiento de la cadena de bloques, algunas propuestas sugieren almacenar solo una copia parcial del blockchain o guardar versiones de la cadena según la zona o las actividades del nodo [12, 23, 24]. Por lo tanto, diseñar un protocolo de consenso específico para dispositivos con recursos limitados se presenta como el escenario más óptimo para aprovechar de manera eficiente los recursos de estos dispositivos que interactúan en una red IoT.

6. Conclusiones y trabajo futuro

Mediante el análisis exhaustivo de diversos protocolos de consenso para su aplicabilidad en dispositivos IoT, identificamos criterios clave para la selección. Descartamos aquellos que requerían gran poder computacional, generaban altos tiempos de latencia, dependían de tokens o presentaban centralización. Como resultado, nos centramos en dos opciones: PBFT (Practical Byzantine Fault Tolerance) y PoET (Proof of Elapsed Time).

Es relevante destacar que una de las contribuciones significativas fue la representación matemática de los protocolos restantes. Esta representación proporciona una base sólida para el análisis y la comprensión de su funcionamiento, permitiendo una evaluación más precisa de su aplicabilidad en entornos de dispositivos IoT.

Además, exploramos la posibilidad de diseñar protocolos específicos para este entorno particular que podría mejorar la eficiencia y escalabilidad en sistemas IoT con recursos computacionales limitados.

Referencias

1. **Anita, N., Vijayalakshmi, M., Mercy-Shalinie, S. (2023).** Proof-of-Improved-Participation: A New Consensus Protocol for Blockchain Technology. Computer Systems Science Engineering, Vol. 44, No. 3, pp. 2007–2018. DOI: 10.32604/csse.2023.025516.

ISSN 2007-9737

2. **Auhl, Z., Chilamkurti, N., Alhadad, R., Heyne, W. (2022).** A Comparative Study of Consensus Mechanisms in Blockchain for IoT Networks. Electronics, Vol. 11, No. 17, pp. 1–23. DOI: 10.3390/electronics11172694.

3. **Bala, K., Kaur, P. D. (2022).** Changing Trends of Blockchain in IoT: Benefits and Challenges. In 12th International Conference on Cloud Computing, Data Science & Engineering, IEEE, pp. 324–329. DOI: 10.1109/Confluence52989.2022.9734206.

4. **Banafa, A. (2022).** IoT and Blockchain Convergence: Benefits and Challenges. In 8th World Forum on Internet of Things. URL: https://iot.ieee.org/articles-publications/newsletter/january-2017/iot-and-blockchain-convergence-benefits-and-challenges.html.

5. **Bano, S., Sonnino, A., Al-Bassam, M., Azouvi, S., et al. (2019).** SoK: Consensus in the Age of Blockchains. Proceedings of the 1st ACM Conference on Advances in Financial Technologies, pp. 183–198. DOI: 10.1145/3318041.335545.

6. **Bouraga, S. (2021).** A Taxonomy of Blockchain Consensus Protocols: A Survey and Classification Framework. Expert Systems with Applications, Vol. 168, pp. 1–17. DOI: 10.1016/j.eswa.2020.114384.

7. **Cp, V., Kalaivanan, S., Karthik, R., Sanjana, A. (2022).** Blockchain-based IoT Device Security. In 2nd International Conference on Artificial Intelligence and Signal Processing, IEEE, pp. 1–6. DOI: 10.1109/AISP53593.2022.9760674.

8. **Ferdous, M. S., Chowdhury, M. J. M., Hoque, M. A. (2021).** A Survey of Consensus Algorithms in Public Blockchain Systems for Crypto-Currencies. Journal of Network and Computer Applications, Vol. 182, No. 103035, pp. 1–28. DOI: 10.1016/j.jnca.2021.103035.

9. **Hagui, I., Msolli, A., Helali, A., Hassen, F. (2021).** Based Blockchain-Lightweight Cryptography Techniques for Security Information: A Verification Secure System for User Authentication. In International Conference on Control, Automation and Diagnosis, IEEE, pp. 1–5. DOI: 10.1109/ICCAD52417.2021.9638751.

10. **Karantias, K., Kiayias, A., Zindros, D. (2020).** Proof-of-burn. In Financial Cryptography and Data Security. Springer International Publishing, pp. 523–540. DOI: 10.1007/978-3-030-51280-4$_2$.

11. **Kaur, S., Chaturvedi, S., Sharma, A., Kar, J. (2021).** A Research Survey on Applications of Consensus Protocols in Blockchain. Security and Communication Networks, Vol. 2021, No. 1, pp. 1–22. DOI: 10.1155/2021/6693731.

12. **Kim, T., Noh, J., Cho, S. (2019).** SCC: Storage Compression Consensus for Blockchain in Lightweight IoT Network. In International Conference on Consumer Electronics, IEEE, pp. 1–4. DOI: 10.1109/ICCE.2019.8662032.

13. **Kumar, S., Tiwari, P., Zymbler, M. (2019).** Internet of Things is a Revolutionary Approach for Future Technology Enhancement: A Review. Journal of Big Data, Vol. 6, No. 111, pp. 1–21. DOI: 10.1186/s40537-019-0268-2.

14. **Lys, L., Forestier, S., Vodenicarevic, D., Laversanne-Finot, A. (2023).** Defending against the Nothing-at-Stake Problem in Multi-Threaded Blockchains. ArXiv. DOI: https://doi.org/10.48550/arXiv.2302.10009.

15. **MacKenzie, B., Ferguson, R. I., Bellekens, X. (2018).** An Assessment of Blockchain Consensus Protocols for the Internet of Things. In International Conference on Internet of Things, Embedded Systems and Communications, IEEE. DOI: 10.1109/IINTEC.2018.8695298.

16. **Panarello, A., Tapas, N., Merlino, G., Longo, F., Puliafito, A. (2018).** Blockchain and IoT Integration: A Systematic Survey. Sensors, Vol. 18, No. 8, pp. 1–37. DOI: 10.3390/s18082575.

17. **Raghav, Andola, N., Venkatesan, S., Verma, S. (2020).** PoEWAL: A Lightweight Consensus Mechanism for Blockchain in IoT. Pervasive and Mobile Computing, Vol. 69, pp. 1–12. DOI: 10.1016/j.pmcj.2020.101291.

18. **Salimitari, M., Chatterjee, M. (2018).** A Survey on Consensus Protocols in Blockchain

for IoT Networks. arXiv, pp. 1–15. DOI: 10.48550/arXiv.1809.05613.

19. **Salimitari, M., Chatterjee, M. (2018).** An Overview of Blockchain and Consensus Protocols for IoT Networks. ArXiv. URL: https://api.semanticscholar.org/CorpusID:52285522.
20. **Salimitari, M., Chatterjee, M., Fallah, Y. P. (2020).** A Survey on Consensus Methods in Blockchain for Resource-Constrained IoT Networks. Internet of Things, Vol. 11, pp. 1–19. DOI: 10.1016/j.iot.2020.100212.
21. **Salimitari, M., Joneidi, M., Chatterjee, M. (2019).** AI-Enabled Blockchain: An Outlier-Aware Consensus Protocol for Blockchain-Based IoT Networks. In Global Communications Conference, IEEE, pp. 1–6. DOI: 10.1109/GLOBECOM38437.2019.9013824.
22. **Seok, B., Park, J., Park, J. H. (2019).** A Lightweight Hash-Based Blockchain Architecture for Industrial IoT. Applied Sciences, Vol. 9, No. 18, pp. 1–9. DOI: 10.3390/app9183740.
23. **Shahid, A. R., Pissinou, N., Staier, C., Kwan, R. (2019).** Sensor-Chain: A Lightweight Scalable Blockchain Framework for Internet of Things. In International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE, pp. 1154–1161. DOI: 10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00195.
24. **Singla, A., Bertino, E. (2018).** Blockchain-Based PKI Solutions for IoT. In 4th International Conference on Collaboration and Internet Computing, IEEE, pp. 9–15. DOI: 10.1109/CIC.2018.00-45.
25. **Uddin, M., Muzammal, M., Hameed, M. K., Javed, I. T., Alamri, B., Crespi, N. (2021).** CBCIoT: A Consensus Algorithm for Blockchain-Based IoT Applications. Applied Sciences, Vol. 11, No. 22, pp. 1–20. DOI: 10.3390/app112211011.
26. **Uddin, M., Stranieri, A., Gondal, I., Balasubramanian, V. (2021).** A Survey on the Adoption of Blockchain in IoT: Challenges and Solutions. Blockchain: Research and Applications, Vol. 2, No. 2, pp. 1–49. DOI: 10.1016/j.bcra.2021.100006.
27. **V, G., George, G. V. S. (2024).** Integration of Blockchain with Hybrid Lightweight Cryptosystem. Multimed Tools Appl. DOI: 10.1007/s11042-024-20280-1.
28. **Wang, W., Hoang, D. T., Hu, P., Xiong, Z., Niyato, D., Wang, P., Wen, Y., Kim, D. I. (2019).** A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks. IEEE Access, Vol. 7, pp. 22328–22370. DOI: 10.1109/ACCESS.2019.2896108.
29. **Wang, X., Zha, X., Ni, W., Liu, R. P., Guo, Y. J., Niu, X., Zheng, K. (2019).** Survey on Blockchain for Internet of Things. Computer Communications, Vol. 136, pp. 10–29. DOI: 10.1016/j.comcom.2019.01.006.
30. **Wu, Y., Song, L., Liu, L., Li, J., Li, X., Zhou, L. (2020).** Consensus Mechanism of IoT Based on Blockchain Technology. Shock and Vibration, Vol. 2020, pp. 1–9. DOI: 10.1155/2020/8846429.
31. **Xiao, Y., Zhang, N., Lou, W., Hou, Y. T. (2020).** A Survey of Distributed Consensus Protocols for Blockchain Networks. IEEE Communications Surveys Tutoriales, Vol. 22, No. 2, pp. 1432–1465. DOI: 10.1109/COMST.2020.2969706.
32. **Zhan, Y., Wang, B., Lu, R., Yu, Y. (2021).** DRBFT: Delegated Randomization Byzantine Fault Tolerance Consensus Protocol for Blockchains. Information Sciences, Vol. 559, pp. 8–21. DOI: doi.org/10.1016/j.ins.2020.12.077.

Article received on 20/02/2025; accepted on 07/04/2025.
**Corresponding author is Gina Gallegos Garcia.*