Welches Konsensprotokoll bietet die beste Kombination aus Leichtgewichtigkeit, Skalierbarkeit und Effizienz für die Integration von IoT-Geräten?

Fatih Dönmez
Freie Universität Berlin
fatih.doenmez@fu-berlin.de

***Abstract*—Der Erfolg der Blockchain-Technologie als Grundlage für Kryptowährungen hat die Tür zu weiteren Anwendungsgebieten geöffnet. Ihre inhärenten Sicherheitsmechanismen und Widerstandsfähigkeit gegenüber Angriffen sind die Hauptvorteile, die den Einsatz der Blockchain in anderen Bereichen attraktiv machen. Die Blockchain ist auf einen Konsensmechanismus angewiesen, um sich auf neue Daten zu einigen. Die meisten derzeit für Kryptowährungen verwendeten Konsensmechanismen erfordern jedoch hohe Rechenleistung und sind daher für ressourcenbeschränkte Systeme ungeeignet. Dies bedeutet, dass herkömmliche Konsensmechanismen wie Proof of Work (PoW) ungeeignet sind. In diesem Artikel vergleichen wir mehrere gängige Konsensmechanismen, um herauszufinden, welche Mechanismen sich für den Einsatz im IoT eignen und welche Kompromisse erforderlich sind.**

***Index Terms*—Article, Konsensprotokolle, Verteilte Systeme, IoT-Geräte**

I. Einleitung

A. Motivation

Die fortschreitende Entwicklung der Internet of Things (IoT) stellt eine transformative Kraft in modernen Technologien dar, die das Potential besitzt, sowohl das Alltagsleben als auch diverse Industriebereiche grundlegend zu verändern. Durch die exponentielle Zunahme vernetzter Geräte entstehen jedoch auch erhebliche Herausforderungen, insbesondere im Hinblick auf die Sicherheit und das Datenmanagement innerhalb dieser Netzwerke. In diesem Kontext gewinnt die Auswahl eines geeigneten Konsensprotokolls, das Leichtgewichtigkeit, Skalierbarkeit und Effizienz vereint, an entscheidender Bedeutung.

Das Konzept des Konsensprotokolls, ursprünglich entwickelt für Kryptowährungen und Blockchain-Technologien, dient dazu, Übereinstimmung in einem verteilten Netzwerk ohne zentrale Autorität zu schaffen. Diese Protokolle sind essenziell, um sicherzustellen, dass Transaktionen oder Datenübermittlungen zwischen Geräten in einem IoT-Netzwerk verlässlich und sicher ablaufen. Die Anforderungen an solche Protokolle im IoT-Kontext sind jedoch signifikant unterschiedlich im Vergleich zu jenen in typischen Blockchain-Anwendungen wie Kryptowährungen.

Die Motivation dieses Überblicks liegt darin, durch eine detaillierte Analyse und Bewertung der verschiedenen verfügbaren Konsensprotokolle eine fundierte Empfehlung aussprechen zu können, welche die spezifischen Anforderungen und Herausforderungen von IoT-Netzwerken am effektivsten adressiert.

B. Problemstellung

Erstens muss das Konsensprotokoll extrem leichtgewichtig sein, da IoT-Geräte oft durch limitierte Rechenkapazitäten und minimale Speicherressourcen gekennzeichnet sind [14]. Diese Geräte sind nicht in der Lage, komplexe Berechnungen durchzuführen, die in traditionellen Konsensprotokollen wie Proof of Work erforderlich sind. Daher muss ein idealer Ansatz für IoT-Geräte diese Beschränkungen berücksichtigen und Protokolle anbieten, die minimale Rechenleistung erfordern.

Zweitens muss das Protokoll skalierbar sein, um mit der stetig wachsenden Anzahl von IoT-Geräten, die miteinander kommunizieren, umgehen zu können. Schätzungen zufolge wird die Anzahl der IoT-Geräte bis zum Jahr 2025 auf über 75 Milliarden ansteigen [2]. Ein effektives Konsensprotokoll muss in der Lage sein, effizient mit einer derart massiven Skala an Transaktionen und Datenaustausch umzugehen, ohne dabei an Leistungsfähigkeit zu verlieren.

Drittens ist die Effizienz des Protokolls von höchster Bedeutung, besonders in Bezug auf den Energieverbrauch. IoT-Geräte sind häufig von Batterien abhängig und befinden sich an Standorten, an denen ein häufiger Batteriewechsel unpraktisch wäre. Ein energieeffizientes Konsensprotokoll ist daher entscheidend, um die Langlebigkeit und Funktionalität dieser Geräte zu gewährleisten.

Angesichts dieser Anforderungen stellt sich die Frage, welches Konsensprotokoll am besten geeignet ist, die Balance zwischen Leichtgewichtigkeit, Skalierbarkeit und Effizienz zu finden.

C. Ziel des Papers

Das primäre Ziel dieses Papers besteht darin, eine systematische und umfassende Analyse verschiedener Konsensprotokolle durchzuführen, um festzustellen, welches Protokoll die optimale Kombination aus Leichtgewichtigkeit, Skalierbarkeit und Effizienz für die Integration von IoT-Geräten bietet. Diese Untersuchung zielt darauf ab, praxisorientierte Einsichten zu liefern, die Entscheidungsträgern und Entwicklern helfen können, informierte Entscheidungen über die Implementierung von Konsensmechanismen in IoT-Umgebungen zu treffen.

D. Gliederung

Das Paper ist in mehrere Schlüsselabschnitte gegliedert, um eine systematische Untersuchung und Diskussion über Konsensprotokolle für IoT-Geräte zu ermöglichen:

a) Einleitung: Erläutert die Bedeutung von Konsensprotokollen für IoT und definiert die Kernziele des Papers. Das erste Kapitel besteht aus vier Abschnitten. Während die Motivation erklärt, warum das Problem relevant ist, gibt die Problemstellung einen kurzen Überblick über den Beitrag des Papers zum diskutierten Thema. Sie definiert das spezifische Problem und zeigt, wie die Forschungsarbeit zur Lösung beitragen wird. Im Abschnitt Ziel des Papers werden die Ziele des Papers festgelegt, und der letzte Abschnitt, Gliederung, liefert die Struktur des gesamten Papiers.

b) Verwandte Arbeiten: Dieses Kapitel bietet einen grundlegenden Überblick über Konsensprotokolle in verteilten Systemen, erläutert die wichtigsten Arten und ihre Funktionsweise. Weiterhin wird auf die speziellen Herausforderungen eingegangen, die die Ressourcenknappheit in IoT-Netzwerken mit sich bringt, und warum konventionelle Konsensprotokolle oft nicht optimal für solche Umgebungen sind.

c) Konsensprotokolle in IoT: In diesem Abschnitt werden verschiedene Konsensprotokolle, die speziell für den Einsatz in ressourcenbeschränkten IoT-Umgebungen entwickelt wurden, vorgestellt und verglichen. Es werden die technischen Merkmale, die Leistungsfähigkeit und die Eignung der verschiedenen Ansätze diskutiert, um einen Einblick in zukünftige Entwicklungen und Potenziale zu geben.

d) Schlussfolgerung: In diesem Abschnitt werden die Ergebnisse diskutiert und zusammengefasst. Es wird aufgezeigt, welche Einschränkungen bei der Optimierung von Konsensprotokollen für IoT bestehen.

II. Verwandte Arbeiten

A. Konsens in verteilten Systemen

Konsensprotokolle in verteilten Systemen sind Mechanismen, die eine gemeinsame Übereinkunft unter allen Knoten ermöglichen und so für Datenintegrität und -konsistenz sorgen. Konsens bedeutet in diesem Zusammenhang, dass unabhängige Prozesse innerhalb eines komplexen Systems gemeinsame Entscheidungen treffen, z. B. die Reihenfolge von Transaktionen. Solche Systeme benötigen fehlertolerante Mechanismen, da einzelne Prozesse ausfallen können, ohne das Gesamtsystem zu beeinträchtigen. Daher sind folgende Eigenschaften wichtig:

- **Terminierung**: Jeder fehlerfreie Prozess muss zu einem Beschluss kommen.
- **Übereinstimmung**: Alle korrekt funktionierenden Prozesse müssen sich auf denselben Wert einigen.
- **Integrität**: Wenn alle korrekt funktionierenden Prozesse denselben Wert vorgeschlagen haben, muss dieser Wert entschieden werden.

Die Wahl eines geeigneten Konsensprotokolls wird durch die zu tolerierenden Fehlerarten bestimmt. Protokolle wie Paxos, Raft und praktische byzantinische Fehlertoleranz (PBFT) versuchen, diese Anforderungen zu erfüllen, und setzen dabei unterschiedliche Schwerpunkte. Dies führt zu einem Spannungsfeld zwischen Sicherheit, Effizienz und Ressourcennutzung. So muss jedes Protokoll hinsichtlich Fehlertoleranz, Netzwerkgröße und Kommunikationsaufwand abgewogen werden, um eine optimale Lösung für spezifische Anwendungsfälle zu bieten.

B. Berechnungsmodell für Konsensalgorithmen

Konsensalgorithmen sind seit Jahrzehnten ein aktives Forschungsfeld, beginnend in den 1970er Jahren mit Leslie Lamports Pionierarbeit. Diese Forschung mündete in die Entwicklung grundlegender Algorithmen, die bis heute als Meilensteine gelten. Der Einzug neuer Technologien wie Blockchain brachte zu Beginn des neuen Jahrtausends maßgeschneiderte Konsensalgorithmen mit sich, die eine detaillierte Betrachtung des Berechnungsmodells erforderlich machen.

a) Byzantinische vs. Absturzfehler: Um einen geeigneten Konsensmechanismus zu wählen, muss man die verschiedenen Fehlerarten verstehen. Absturzfehler entstehen, wenn ein Prozess plötzlich aufhört zu arbeiten und nicht fortgesetzt wird. Dagegen sind byzantinische Fehler komplexer und disruptiver, meist aufgrund von böswilligen Aktivitäten. Sie leiten sich vom ”Problem der byzantinischen Generäle” ab und sind schwerer zu beheben als Absturzfehler, da sie Prozesse teils funktionsfähig und teils ausgefallen erscheinen lassen.

b) Synchrone vs. Asynchrone Kommunikation: In synchronen Systemen arbeiten alle Prozesse mit einer synchronisierten Uhr, was jeden Verarbeitungsschritt und jede Nachrichtübertragung begrenzt. In verteilten Systemen hingegen ist die Kommunikation zwischen den Prozessen asynchron, ohne globale oder einheitliche Uhr. Der FLP-Unmöglichkeitssatz (Fischer, Lynch, Paterson) [11] zeigt, dass in einem voll asynchronen System mit mindestens einem Absturzfehler ein deterministischer Konsensalgorithmus nicht möglich ist.

c) Genehmigte vs. Genehmigungsfreie Netzwerke: Ein weiterer wichtiger Faktor des Berechnungsmodells ist die Topologie des Netzwerks. In einem geschlossenen, genehmigten Netzwerk sind die Prozesse im Voraus festgelegt und können sich gegenseitig authentifizieren.

In genehmigungsfreien Netzwerken kann jeder ohne Voranmeldung teilnehmen, wobei Maßnahmen gegen Sybil-Angriffe nötig sind, bei denen einzelne Akteure durch mehrere Identitäten Einfluss gewinnen. Fortschrittliche Konsensalgorithmen, die für genehmigte Netzwerke entwickelt wurden, können in genehmigungsfreien Netzwerken versagen, weshalb zusätzliche Überlegungen erforderlich sind.

C. Voting-basierte Konsensalgorithmen

Mit der Weiterentwicklung des Konsensproblems in verteilten Systemen wurde es notwendig, die Algorithmen in Kategorien einzuordnen. Frühere Implementierungen setzten auf verschiedene abstimmungsbasierte Mechanismen. Diese gewährleisten zuverlässige Fehlertoleranz und basieren auf starken mathematischen Grundlagen für Sicherheit und Stabilität. Aufgrund ihrer demokratischen Natur sind sie jedoch ineffizient und langsam, insbesondere in großen Netzwerken.

Practical Byzantine Fault Tolerance (pBTF), entwickelt von Barbara Liskov und Miguel Castro im Jahr 1999 [5], sollte eine praktische Lösung für das Problem der byzantinischen Generäle in verteilten Systemen bieten. Der Algorithmus repliziert byzantinische Zustandsmaschinen mithilfe von Abstimmungen bei Zustandsänderungen. Die Fehlertoleranz erfordert, dass nicht mehr als ein Drittel aller Knoten fehlerhaft

sein dürfen. Ein Knoten fungiert als primärer Knoten, während die anderen sekundäre Knoten sind.

Die Kernkonsensphasen gliedern sich in drei Hauptteile:

- **Prepare-Phase**: Der primäre Knoten sendet "Prepare"-Nachrichten an alle sekundären Knoten.
- **Commit-Phase**: Der primäre Knoten sendet "Commit"-Nachrichten an alle sekundären Knoten.
- **View Change-Phase**: Der primäre Knoten sendet "View Change"-Nachrichten an alle sekundären Knoten.

In jeder Konsensrunde wechselt der primäre Knoten in einem Prozess namens "View Change". Obwohl dieser Mechanismus optimiert ist und eine geringe Latenz aufweist, leidet er unter Einschränkungen wie einer Anfälligkeit für Sybil-Angriffe und eingeschränkter Skalierbarkeit.

D. Proof-basierte Konsensalgorithmen

Die Einführung der Blockchain-Technologie und Distributed-Ledger-Netzwerke brachte größere, offene Netzwerke hervor, die sich besser für proof-basierte Konsensalgorithmen eignen. Teilnehmer müssen hier einen Nachweis erbringen, um an der Entscheidungsfindung beteiligt zu werden. [13]

- **Proof-of-Work (PoW)**: Eine Partei erbringt einen Nachweis über geleistete Rechenarbeit. Satoshi Nakamoto nutzte dieses Konzept 2008 erstmals für Bitcoin mit der Hashcash-Methode. PoW ist effizienter als byzantinisch fehlertolerante Protokolle, benötigt jedoch viel Energie.
- **Proof-of-Stake (PoS)**: Knoten setzen einen Einsatz, z. B. Kryptowährung, und ein Algorithmus wählt Kandidaten aus, die Aufgaben ausführen. Erfolgreiche Knoten erhalten ihren Einsatz zurück plus eine Belohnung. PoS ist energieeffizienter, fördert Dezentralisierung und hat geringere Einstiegshürden. Dennoch bleiben Risiken wie 51%-Angriffe. Ethereum wechselte kürzlich von PoW zu PoS.

E. Herausfordeungen in IoT Geräten

Das IoT verknüpft physische Objekte der realen Welt mit dem Internet. Im Zentrum des IoT-Ökosystems steht eine Vielzahl intelligenter Geräte, die Sensoren und Aktuatoren enthalten und mit Rechen- und Kommunikationsfunktionen ausgestattet sind [8]. Das System führt verschiedene Aufgaben wie Sensordaten-Erfassung, Kommunikation, Berechnung und Aktuation aus, um Daten in der IoT-Umgebung zu sammeln und zu verarbeiten. Die verwendeten Sensoren messen verschiedene Parameter wie Temperatur, Luftfeuchtigkeit oder Druck. Diese Geräte erfassen Daten aus ihrer Umgebung und senden sie entweder direkt oder über ein Gateway an einen Server. Basierend auf dem Feedback des Servers können IoT -Geräte Befehle an Aktuatoren senden, um bestimmte Parameter anzupassen.

IoT-Anwendungen finden in zahlreichen realen Umgebungen Einsatz, etwa in intelligenten Häusern, im Transportwesen, in Gesundheitssystemen, im Einzelhandel, in intelligenten Städten, bei der Umweltüberwachung und im Energiemanagement [3]. Aufgrund der vielen Vorteile, die IoT bietet, investieren weltweit Industrien, Forschungseinrichtungen und Regierungen beträchtliche Mittel in diese Technologien. Dies hat zu einer Fragmentierung und einem starken Wettbewerb auf dem IoT-Markt geführt, der aufgrund mehrerer Standards von Inkompatibilität geprägt ist.

Das Betriebssystem (OS) bietet eine Abstraktionsschicht für die Hardware, indem es die Ressourcen jedes IoT-Geräts verwaltet [15]. Es stellt eine Programmierschnittstelle bereit und verwaltet die Prozessorzeit. IoT-Geräte arbeiten in ressourcenbeschränkten, parallelen Umgebungen, und das OS muss ein geeignetes Ausführungsmodell bieten, um diese parallelen Anwendungen zu unterstützen. Dieses Modell muss speichereffizient sein [9]. Ebenso muss das OS (da es batteriebetrieben ist) einen Energiesparmodus bereitstellen, wenn keine Anwendung aktiv ist [4]. Die Energieeffizienz der Kommunikationskomponenten stellt eine besondere Herausforderung für ein OS dar, da diese während der Kommunikationsphasen aktiviert sein müssen. Daher verwaltet ein OS die Energieeffizienz mithilfe verschiedener Mechanismen wie einem separaten Radio-Duty-Cycling-Verfahren [12], einem virtuellen Trägererkennungsmechanismus mit Netzwerk-Allokationsvektor und zeitmultiplex-basierten Verfahren (TDMA). Da nicht alle IoT-Geräte über Flash-Speicher verfügen, ist ein geeignetes Dateisystem notwendig, um die Speicheranforderungen mancher Anwendungen zu erfüllen. Das Dateisystem muss Daten effizient auf Sektoren abbilden, um das Schreiben und Lesen zu optimieren. Daher muss ein OS eine vollständige Dateisystem-Schnittstelle bereitstellen [18].

Die Kommunikationsbedürfnisse verschiedener Anwendungen werden durch eine geeignete Kommunikationsarchitektur abgedeckt. Angesichts der Ressourcenknappheit der Geräte müssen die Kommunikationsprotokolle bei Datenerfassung, Ereigniserkennung und -verfolgung, Gerätesynchronisation, Nachbarschaftserkennung und Datenübertragung energie- und speichereffizient sein [6], [7]. Um die Herausforderungen im Ressourcenmanagement für IoT-Geräte mit eingeschränkten Fähigkeiten zu meistern, wurden verschiedene Mechanismen und Verfahren entwickelt, die in fünf Bereiche unterteilt werden.

Das Flussdiagramm des Ressourcenmanagements im OS für Geräte mit geringen Fähigkeiten ist in Abbildung 1 dargestellt. [16]

Die aufgezeigten Herausforderungen, die das Ressourcenmanagement, die Kommunikation und die Hardwarebeschränkungen von IoT-Geräten betreffen, unterstreichen den Bedarf an spezifischen Lösungsansätzen. Insbesondere Konsensprotokolle spielen hier eine entscheidende Rolle. Sie ermöglichen es verteilten Systemen, trotz der inhärenten Ressourcenknappheit und Netzwerkunsicherheiten, zuverlässige Entscheidungen zu treffen. Ein effektives Konsensprotokoll gewährleistet die korrekte Koordination der verschiedenen IoT-Geräte, selbst wenn einige Komponenten ausfallen oder kompromittiert werden. Es trägt dazu bei, die Synchronisation zu verbessern, Energie effizient zu nutzen und die Datenintegrität zu sichern. Der folgenden Abschnitt befasst sich mit den verschiedenen Konsensprotokollen, die speziell für das IoT entwickelt wurden.

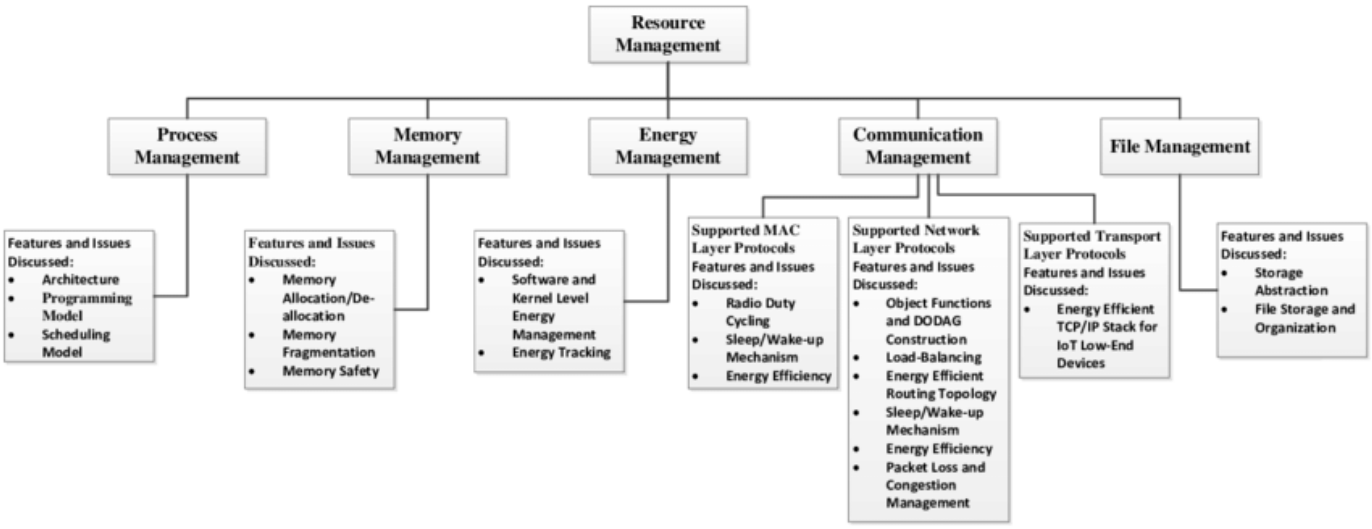

Fig. 1. Resources Management Classification

TABLE I
ÜBERBLICK ÜBER DIE IOT-SPEZIFISCHEN KONSENSMECHANISMEN

Konsensprotokoll	Ähnlich zu	Dezentralisiert	Merkmale
PoSCS	PoS	Nein	Reputationssystem
Microchain	PoS	Teils	Kryptosortierung
PoBT	Weder noch	Teils	Handels- und Blockvalidierung
HPoC	PoW	Teils	IoT Endgeräte
CBPoW	PoW	Teils	Kreditsystem

TABLE II
EIGNUNG DES KONSENSMECHANISMUS FÜR IOT-GERÄTE

Konsensprotokoll	Prozessorauslastung	Sicherheit	Speicher	TPS	Geeignet?
PoSCS	Gering	Hoch	Gering	Variable	Teils
Microchain	Gering	Hoch	Gering	Moderat	Ja
PoBT	Gering	Hoch	Gering	Hoch	Ja
HPoC	Moderat	Hoch	Moderat	Hoch	Teils
CBPoW	Gering	Hoch	Gering	Moderat	Ja

III. KONSENSPROTOKOLLE IN IOT

A. Proof of Block and Trade

Der Proof of Block and Trade (PoBT)-Konsensalgorithmus wurde entwickelt, um die Herausforderungen der Skalierbarkeit und Effizienz in Blockchain-Systemen im Kontext des Internets der Dinge (IoT) zu bewältigen. Entwickelt von S. Maharjan, S. Mohanty und Y. Wang, zielt PoBT darauf ab, den Rechenaufwand zu reduzieren und die Speichernutzung zu optimieren, indem es Trades und Blöcke effizient und sicher validiert und dabei die Systemintegrität aufrechterhält.[1]

Transaktionen, die von IoT-Geräten stammen, werden als ausführbare Smart Contracts vorbereitet. Der Besteller überprüft mithilfe des PoBT-Algorithmus die Anmeldeinformationen und genehmigt anschließend neue Blöcke, die an alle Knoten im Netzwerk verteilt werden. Jeder Knoten prüft die Signatur des Blocks, fügt ihn seinem eigenen Ledger hinzu und aktualisiert den globalen Status, um die Unveränderlichkeit der Transaktionen und die Integrität der Blockchain zu gewährleisten.

Gezielte Optimierungen von PoBT reduzieren die Rechenzeit bei Handelsvalidierungs- und Blockerstellungsprozessen erheblich, was zu einer verbesserten Systemleistung führt. Ein Ledger-Verteilungsmechanismus reduziert den Speicherbedarf für IoT-Knoten, sorgt für effiziente Speicherung und Abruf von Blockchain-Daten und trennt lokale Transaktionen klar von globalen Trades, was die Bandbreitennutzung verbessert.

Der Algorithmus minimiert Kommunikationsverzögerungen, sorgt für schnelle Blockschließzeiten und erhöht die Transaktionsraten. Diese klare Trennung von Trades reduziert die Belastung der Blockchain-Peer-Knoten, ohne die Sicherheit zu beeinträchtigen.

PoBT zeigt eine effektive Lösung für Skalierbarkeits- und Effizienzprobleme in IoT-Blockchain-Netzwerken, bringt jedoch auch offene Fragen hinsichtlich der Vergleichbarkeit mit etablierten Konsensmechanismen wie PoW mit sich:

- **Eingeschränktes Benchmarking:** Der PoBT-Algorithmus bietet keinen direkten Vergleich mit etablierten öffentlichen Blockchain-Konsensmechanismen wie PoW. Dies erschwert eine direkte Bewertung der Leistungsfähigkeit in Anwendungsfällen, die ein hohes Maß an Dezentralisierung und Sicherheit erfordern.
- **Bedenken hinsichtlich der Skalierbarkeit:** Ohne Vergleichsdaten mit herkömmlichen Blockchain-Algorithmen

bleiben die genauen Auswirkungen von PoBT auf die Blockchain-Leistung unbekannt, was zu Unsicherheiten bezüglich Skalierbarkeit und Sicherheit führt.
- **Herausforderungen bei der Risikobewertung:** Unternehmen könnten Schwierigkeiten haben, die Risiken und Vorteile zu bewerten, da ein Vergleich mit bekannten Konsensalgorithmen fehlt. Dies erschwert eine klare Analyse der Stärken und Schwächen.

Ohne Benchmarks mit anderen Blockchain-Konsensmechanismen könnte die Akzeptanz von PoBT beeinträchtigt sein. Vergleiche mit traditionellen Algorithmen sind notwendig, um eine fundierte Einschätzung von PoBT in der Praxis zu ermöglichen. Der PoBT-Algorithmus sorgt durch eine individuelle Handelsüberprüfung und die Einbindung mehrerer Knoten im Konsensprozess für eine hohe Validierungssicherheit. Neue Blöcke werden nur dann genehmigt, wenn sie als gültig anerkannt wurden, und unveränderlich ins Ledger eingefügt. Dadurch bleibt das System vor böswilligen oder ungenauen Transaktionen geschützt und gewährleistet so eine robuste Handelsvalidierung in IoT-Blockchain-Netzwerken.

B. Hierarchical Proof of Capability

Der hierarchische Proof-of-Capability (HPoC) - Konsensmechanismus, entwickelt von Zixiang Nie, Maosheng Zhang, Yueming Lu, Zubair Asghar, Asad Masood und Shakeel Ahmad, wurde speziell für das Internet der Dinge (IoT) konzipiert. Er gewährleistet Datensicherheit und bietet eine schlanke Blockchain-Lösung. Durch die Kombination mit einem asynchronen Proof-of-Work (PoW)-Mechanismus verbessert HPoC die Rechen-, Speicher- und Kommunikationsfähigkeiten von IoT-Edge-Geräten. Ziel ist es, in Edge-Geräteknoten mit eingeschränkten Ressourcen Blöcke mit niedriger Latenz, geringem Stromverbrauch und hoher Stabilität zu generieren.[14]

Der Mechanismus verwendet verschiedene Knotenklassen mit unterschiedlicher Speicherkapazität. Knoten der Klasse 1 besitzen die größte Kapazität und führen eine digitale Cache-Tabelle. Deren Größe beeinflusst den Schwierigkeitsgrad und den Zeitaufwand im Konsensprozess. Der flexibel steuerbare Mechanismus kann bis zu 2000 Transaktionen pro Sekunde (TPS) verarbeiten, was ihn ideal für ressourcenbeschränkte IoT-Edge-Computing-Szenarien macht.

Die hierarchische Struktur verschiedener Knotenklassen mit unterschiedlicher Speicherkapazität steigert die Effizienz und Leistungsfähigkeit. Allerdings könnte der Konsensbildungsmechanismus von HPoC Anpassungen benötigen, um die Widerstandsfähigkeit gegenüber byzantinischen Fehlertoleranzproblemen (BFT) zu stärken. Weitere spezifische Einschränkungen werden jedoch nicht ausdrücklich genannt.

C. Credit-Based PoW

Huang und seine Kollegen schlagen ein kreditbasiertes Proof-of-Work (PoW)-System vor, das sich für IoT-Geräte eignet. [10] Ihr Konsensmechanismus passt die Schwierigkeit der PoW-Aufgaben eines Geräts dynamisch an, je nachdem, wie gut es die Konsensregeln befolgt. Der Gesamtscore eines Knotens ergibt sich durch die Summe seines positiven und negativen Scores. Ein positiver Score steigt, wenn der Konsensmechanismus befolgt wird, während der negative Score durch Regelverstöße wächst. Im Fokus des Papiers stehen zwei spezifische Angriffe, die den Score eines Clients verringern könnten: Lazy Tips und Double-Spending.

Lazy Tips ist ein Problem, das vor allem bei Direct Acyclic Graphs (DAGs) auftritt, bei dem ein Angreifer neue Transaktionen vermeidet, indem er auf alten, bereits existierenden Transaktionen aufbaut. Dies kann dem Netzwerk schaden, da ehrliche Knoten keine Bestätigung ihrer neuen Transaktionen erhalten. Huang und seine Kollegen bestrafen Nutzer auch, wenn sie versuchen, ihre Tokens doppelt auszugeben. Wird ein Knoten als böswillig identifiziert, kommt eine Strafmaßnahme zum Einsatz, bei der die Anzahl seiner böswilligen Transaktionen über einen bestimmten Zeitraum mit einem Strafkoeffizienten multipliziert wird, um den Kreditbetrag zu reduzieren.

Da die Knoten zwei frühere Transaktionen bestätigen müssen, bevor sie ihre eigenen einreichen können, macht ein niedriger PoW-Kredit das Hinzufügen einer Transaktion zeitaufwendig und rechenintensiv. Das kreditbasierte PoW (CBPoW) nutzt ein gestuftes Knotennetzwerk, in dem Lite-Knoten für das Sammeln von Daten und das Übertragen von Transaktionen verantwortlich sind, während Full-Nodes die Pflege des ”Tangle” übernehmen.

D. Proof of Supply Chain Share

PoSCS, ein Konsensmechanismus von Tsang et al., ist speziell auf die verderbliche Lebensmittel-Lieferkette (Perishable Food Supply Chain, PFSC) ausgerichtet [17]. Das Projekt verwendet ein Framework, das ein IoT-Netzwerk zur Überwachung und Kommunikation, eine Blockchain zur Verwaltung der Lebensmitteldaten während des gesamten Lieferkettenzyklus und eine Datenbank zur Archivierung von Lieferketteninformationen umfasst. Die Autoren betonen, dass PoW aufgrund des rechenintensiven Miningprozesses für das IoT ungeeignet ist. Daher schlagen sie einen Konsensmechanismus ähnlich dem Proof-of-Stake (PoS) vor, bei dem jedoch anstelle einer Währung ein Reputationssystem verwendet wird.

Jeder Knoten, der am Konsens beteiligt ist, hat vier Komponenten, die seinen Ruf bestimmen: Einflussfaktor (INF), Interessenfaktor (INT), Engagementfaktor (DEV) und Zufriedenheitsfaktor (SAT). Diese Faktoren werden mit drei Strategien gewichtet: der Interessen-zuerst-Strategie, der moderaten Strategie und der Engagement-zuerst-Strategie. Diese Gewichtung verhindert, dass der Konsensmechanismus Teilnehmer bevorzugt, die versuchen, einen einzelnen Faktor zu maximieren. Außerdem berücksichtigt das System das Versandvolumen, indem es das eingehende und ausgehende Volumen eines bestimmten Akteurs im Lieferkettennetzwerk betrachtet. Diese Faktoren und Gewichte dienen zur pseudorandomisierten Auswahl eines Block-Erstellers, der einen Block generieren muss.

Die Block-Ersteller müssen außerdem eine kleine Menge an PoW-Mining betreiben, um die Blockerstellungszeit zu steuern. Anstatt dass alle Knoten am PoSCS-Konsensmechanismus teilnehmen, ist nur der Block-Ersteller dazu verpflichtet. Das

System setzt auf eine hybride Architektur, die Blockchain und Cloud kombiniert. Die Blockchain zeichnet die Daten über ein bestimmtes Objekt in der Lieferkette parallel zu einer herkömmlichen Datenbank auf. Nachdem das Objekt seinen Weg durch die Lieferkette abgeschlossen hat, wird es aus dem IoT-Gerätespeicher entfernt und in der Cloud archiviert.

E. Microchain

Microchain schlägt einen leichtgewichtigen Konsensmechanismus vor, der speziell für das Internet der Dinge (IoT) entwickelt wurde [19]. Ähnlich wie beim Proof-of-Stake (PoS) werden bei Microchain eine Reihe von Validatoren ausgewählt, um einem Komitee beizutreten. Aus diesem Komitee wird dann ein Knoten bestimmt, der den nächsten Block erzeugt. Das Komitee dient dazu, einen pseudorandomisierten Netzwerk-Ausschnitt auszuwählen, um voreingenommene oder bösartige Blockproduzenten zu vermeiden. Microchain verwendet dabei ein Komitee, das als "Dynasty" bezeichnet wird. Hierbei werden geeignete Validatoren ausgewählt, um der Gruppe beizutreten.

Der Konsensmechanismus von Microchain besteht aus zwei Hauptkomponenten: Proof of Credit (PoC) und Voting-based Chain Finality (VCF). PoC ist ein PoS-Mechanismus, bei dem ein Kreditgewicht die Wahrscheinlichkeit eines Knotens erhöht, ausgewählt zu werden, um einen Block zu erstellen. Knoten haben in einer bestimmten Dynasty mit höherem Kreditgewicht eine größere Chance, ausgewählt zu werden.

VCF dient als Werkzeug zur Auflösung von Gabelungen. Es verlängert die Kette durch das Hinzufügen neuer Blöcke und schützt die Blockchain vor böswilligen oder versehentlichen Umstrukturierungen, indem Checkpoints hinzugefügt werden. Xu et al. nutzen eine verifizierbare Zufallsfunktion (VRF) für die Auswahl der Knoten, die in eine Dynasty aufgenommen werden. Microchain geht davon aus, dass Netzwerke synchron arbeiten, und kann somit zwei Garantien bieten: Persistenz und Lebendigkeit. Persistenz bedeutet, dass alle Nutzer die gleiche Historie der Blockchain teilen und wenn ein ehrlicher Knoten eine Transaktion als endgültig ansieht, wird dies auch von allen anderen ehrlichen Knoten so akzeptiert. Lebendigkeit stellt sicher, dass eine gültige Transaktion, die von einem ehrlichen Knoten eingereicht wird, schließlich in einem neuen Block aufgezeichnet wird.

IV. Schlussfolgerung

In dieser Arbeit haben wir ressourcenbeschränkte IoT-Geräte und die Einschränkungen aktueller Konsensmechanismen im Kontext des IoT erörtert. Zu Beginn haben wir Kriterien zur Bewertung der Konsensmechanismen festgelegt, wie etwa Geschwindigkeit, Sicherheit, Dezentralisierung, Speichernutzung und TPS. Danach wurden mehrere Konsensmechanismen im Einzelnen besprochen und ihr genereller Ablauf erläutert. Diese IoT-orientierten Konsensmechanismen modifizieren bereits bestehende PoW- und PoS-Ansätze, eliminieren jedoch energieintensive Mining-Prozesse und monetäre Systeme.

Die Analyse betrachtete die Vor- und Nachteile jedes Konsensmechanismus sowie deren Eignung für das IoT. Die Ergebnisse zeigen, dass Microchain, CBPoW und PoBT für das IoT geeignet sind. Microchain weist eine angemessene Leistung für das IoT in privaten Umgebungen auf, während CBPoW Probleme mit der lokalen Blockchain-Speicherung auf IoT-Geräten löst und PoBT über einen Ledger-Verteilungsmechanismus verfügt, um den Speicherbedarf von IoT-Knoten zu senken und so die Ressourenauslastung innerhalb des Netzwerks zu optimieren. Daher wurden diese als geeignet eingestuft. Die weitere Mechanismen wurden als teilweise geeignet bewertet. Diese teilweise empfohlenen Mechanismen lösen Probleme mit monetären Systemen, Rechenaufwand und Speicheranforderungen. Allerdings führen sie auch neue Herausforderungen ein, wie Abhängigkeit von der Cloud, synchronisierte Uhren sowie unklare oder unüberprüfte Leistungsangaben.

References

[1] Sujit Biswas et al. "PoBT: A Lightweight Consensus Algorithm for Scalable IoT Business Blockchain". In: *IEEE Internet of Things Journal* 7 (Dec. 2019), pp. 2327–4662. DOI: 10.1109/JIOT.2019.2958077.

[2] Peter Brown. "75.4 Billion Devices Connected to the Internet of Things by 2025". In: *Electronics360* (2016).

[3] Muhammad Burhan et al. "IoT Elements, Layered Architectures and Security Issues: A Comprehensive Survey". In: *Sensors (Basel, Switzerland)* 18 (2018).

[4] Stefano Buzzi et al. "A Survey of Energy-Efficient Techniques for 5G Networks and Challenges Ahead". In: *IEEE Journal on Selected Areas in Communications* 34.4 (2016), pp. 697–709. DOI: 10.1109/JSAC.2016.2550338.

[5] M. Castro. "Practical Byzantine fault tolerance". In: (1999), pp. 173–186. DOI: 10.1145/296806.296824.

[6] Adam Dunkels, Fredrik Österlind, and Zhitao He. "An adaptive communication architecture for wireless sensor networks". In: *ACM International Conference on Embedded Networked Sensor Systems*. 2007. URL: https://api.semanticscholar.org/CorpusID:11778569.

[7] Prabal Dutta and Adam Dunkels. "Operating systems and network protocols for wireless sensor networks". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 370 (2012), pp. 68–84. URL: https://api.semanticscholar.org/CorpusID:1199659.

[8] Ala Al-Fuqaha et al. "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications". In: *IEEE Communications Surveys and Tutorials* 17.4 (2015), pp. 2347–2376. DOI: 10.1109/COMST.2015.2444095.

[9] Jason L. Hill and David E. Culler. "A wireless embedded sensor architecture for system-level optimization". In: 2002.

[10] Junqin Huang et al. "Towards Secure Industrial IoT: Blockchain System With Credit-Based Consensus Mechanism". In: *IEEE Transactions on Industrial Informatics* 15.6 (2019), pp. 3680–3689. DOI: 10.1109/TII.2019.2903342.

[11] Ivan Klianev. "Different Perspectives on FLP Impossibility". In: *ArXiv* abs/2210.02695 (2022). DOI: 10.48550/arXiv.2210.02695.

[12] Jeong Gil Ko et al. “Pragmatic low-power interoperability: ContikiMAC vs TinyOS LPL”. In: *2012 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*. 2012, pp. 94–96. DOI: 10.1109/SECON.2012.6276358.

[13] Shijie Lin. “Proof of Work vs. Proof of Stake in Cryptocurrency”. In: *Highlights in Science, Engineering and Technology* (2023). DOI: 10.54097/hset.v39i.6683.

[14] Zixiang Nie, Maosheng Zhang, and Yueming Lu. “HPoC: A Lightweight Blockchain Consensus Design for the IoT”. In: *Applied Sciences* 12.24 (2022). ISSN: 2076-3417. DOI: 10.3390/app122412866. URL: https://www.mdpi.com/2076-3417/12/24/12866.

[15] Ramon Serna Oliver, Ivan Shcherbakov, and Gerhard Fohler. “An operating system abstraction layer for portable applications in wireless sensor networks”. In: *ACM Symposium on Applied Computing*. 2010.

[16] Nenad Tomić. “A Review of Consensus Protocols in Permissioned Blockchains”. In: *Journal of Computer Science Research* 3 (Apr. 2021). DOI: 10.30564/jcsr.v3i2.2921.

[17] Y. P. Tsang et al. “Blockchain-Driven IoT for Food Traceability With an Integrated Consensus Mechanism”. In: *IEEE Access* 7 (Sept. 2019), pp. 129000–129017. DOI: 10.1109/ACCESS.2019.2940227.

[18] Nicolas Tsiftes and Adam Dunkels. “A database in every sensor”. In: *ACM International Conference on Embedded Networked Sensor Systems*. 2011. URL: https://api.semanticscholar.org/CorpusID:15279428.

[19] Ronghua Xu et al. *Microchain: A Hybrid Consensus Mechanism for Lightweight Distributed Ledger for IoT*. 2019. arXiv: 1909.10948 [cs.DC].

List of Figures