RÉPUBLIQUE DU BÉNIN
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ D'ABOMEY-CALAVI

INSTITUT DE FORMATION ET DE RECHERCHE EN INFORMATIQUE

BP 526 Cotonou Tel : +229 21 14 19 88
http://www.ifri-uac.bj Courriel : contact@ifri.uac.bj

MÉMOIRE

pour l'obtention du

Diplôme de Master en Informatique

Option : Sécurité Informatique

Présenté par :
Wilson **VLAVONOU**

L'analyse des algorithmes de consensus utilisés dans la technologie blockchain en vue de proposer un meilleur système dans les réseaux IoT destinés à la logistique.

Sous la supervision :
Professeur Chérif **DIALLO**

Membres du jury :

ASSOGBA Marc Kokou	Professeur Titulaire CAMES	EPAC/UAC	Président
DAGBA K. Theophile	Maître de Conférences CAMES	ENEAM/UAC	Examinateur
DIALLO Chérif	Professeur Titulaire CAMES	Université Gaston Berger	Rapporteur

Date de soutenance : **31/03/2026**

Année Académique : **2023-2024**

Sommaire

Dédicace ii

Remerciements iii

Résumé iv

Abstract v

Table des figures vi

Liste des tableaux vii

Liste des Algorithmes viii

Liste des acronymes ix

Introduction 1

1 Etat de l'art sur les algorithmes de consensus et les réseaux IoT destinés à la logistique 3

2 Analyse et proposition d'un système de consensus adapté aux réseaux IoT logistiques 19

3 Résultats et discussions 36

Conclusion 55

Bibliographie 56

Bibliographie 56

Webographie 57

Webographie 58

Annexe 59

Table des matières 60

Dédicace

A
Mon feu père Vincent VLAVONOU
Ma mère Kingnavodé AGOSSOU
Ma soeur et mes frères.

Remerciements

Nos sincères remerciements vont à l'égard de Dieu.

Nous exprimons notre vive gratitude aux personnes physiques qui ont contribué à rendre meilleur notre formation et travail, notamment :

- Pr Chérif DIALLO, notre maitre de mémoire ;
- Pr Eugène C. EZIN, Directeur de l'Institut de Formation et de Recherche en Informatique (IFRI-UAC) pour ses conseils et pour cette formation accordée ;
- Pr Gaston EDAH, Directeur Adjoint de l'IFRI pour son suivi et sa disponibilité à nous accompagner durant toute la formation ;
- Dr Arnaud AHOUANDJINOU, Coordinateur de la formation, pour son accompagnement attentif, sa disponibilité et la qualité de son suivi durant toute la formation ;
- Tout le corps professoral de l'IFRI pour nous avoir donné les enseignements nécéssaires tout au long de notre formation ;
- Nos parents, proches et amis qui nous ont soutenus de près ou de loin tout au long de ce cycle ;
- Les membres du jury, pour avoir acceptés accorder leur disponibilité pour l'examen de ce document.

Résumé

L'intégration de la technologie blockchain au sein des réseaux de l'Internet des Objets (IoT) constitue une piste prometteuse pour renforcer la sécurité et la fiabilité de la traçabilité logistique. Néanmoins, les algorithmes de consensus traditionnels, tels que le Proof of Work (PoW) ou le Practical Byzantine Fault Tolerance (PBFT), montrent rapidement leurs limites face aux contraintes de ressources propres aux objets connectés. Dans ce contexte, nous avons proposé un nouveau mécanisme de consensus, dénommé PoEWAL-LT (Proof of Elapsed Work and Luck for Logistics Traceability), spécialement conçu pour les environnements IoT appliqués à la logistique. Ce mécanisme repose sur un score de travail adaptatif intégrant à la fois la fiabilité des nœuds et la criticité des événements logistiques.

Contrairement aux approches classiques existantes (PoW, PoS, PoBT, etc.), PoEWAL-LT introduit une validation sémantique proactive permettant la détection d'anomalies logistiques, telles que les incohérences de localisation GPS ou les ruptures de la chaîne du froid. Les expérimentations menées à l'aide du simulateur Contiki-Cooja mettent en évidence les performances de la solution proposée, avec une latence de consensus réduite à 1,2 seconde, une consommation énergétique moyenne limitée à 5,36 joules par transaction, ainsi qu'un mécanisme efficace de détection des nœuds byzantins fondé sur une analyse multicritère. Ces résultats démontrent qu'il est possible de concilier sécurité cryptographique, efficacité énergétique et résilience opérationnelle dans les systèmes de traçabilité logistique basés sur la blockchain et l'IoT.

Mots clés : Blockchain, Internet des Objets (IoT), Consensus, Traçabilité logistique, PoEWAL-LT, Systèmes contraints, Sécurité Byzantine.

Abstract

The integration of blockchain technology into Internet of Things (IoT) networks is a promising avenue for enhancing the security and reliability of logistics traceability. However, traditional consensus algorithms, such as Proof of Work (PoW) or Practical Byzantine Fault Tolerance (PBFT), quickly reveal their limitations when faced with the resource constraints specific to connected objects. In this context, we have proposed a new consensus mechanism, called PoEWAL-LT (Proof of Elapsed Work and Luck for Logistics Traceability), specifically designed for IoT environments applied to logistics. This mechanism is based on an adaptive work score that integrates both node reliability and the criticality of logistics events.

Unlike existing conventional approaches (PoW, PoS, PoBT, etc.), PoEWAL-LT introduces proactive semantic validation that enables the detection of logistics anomalies, such as GPS location inconsistencies or cold chain breaks. Experiments conducted using the Contiki-Cooja simulator highlight the performance of the proposed solution, with consensus latency reduced to 1.2 seconds, average energy consumption limited to 5.36 joules per transaction, and an effective Byzantine node detection mechanism based on multi-criteria analysis. These results demonstrate that it is possible to reconcile cryptographic security, energy efficiency, and operational resilience in logistics traceability systems based on blockchain and IoT.

Key words: Blockchain, Internet of Things (IoT), Consensus, Logistics Traceability, PoEWAL-LT, Constrained Systems, Byzantine Security.

Table des figures

1.1 Structure des blocs 4
1.2 Domaines d'application de l'Internet des Objets (IoT) 5

2.1 Architecture du système 26

3.1 Environnement de simulation 36
3.2 Comparaison du débit (Transactions Par Seconde) 38
3.3 Comparaison multi-algorithmes : latence de consensus (échelle logarithmique) 40
3.4 Évolution du temps de consensus par phase 41
3.5 Détection byzantine multicritère : répartition par type de comportement malveillant . 43
3.6 Consommation d'énergie vs Nombre de Transactions (Valeurs du tableau comparatif) 45
3.7 PoEWAL-LT : Distribution des validateurs (Coefficient Gini = 0,082) 49

Liste des tableaux

1.1 Matrice de gain du jeu « pile ou face » 12
1.2 Matrice de gain du Dilemme du prisonnier 12

2.3 Justification des pondérations du Work Score par la classification ABC-XYZ-VED . . . 29
2.4 Application de la méthode Rank Sum aux critères du Work Score 30

3.1 Paramètres de configuration de la simulation 37
3.2 Comparaison du débit (TPS) par algorithme 39
3.3 Latence de consensus par algorithme 40
3.4 Décomposition du temps de consensus moyen 42
3.5 Répartition des détections byzantines par type (données mesurées) 43
3.6 Conversion des valeurs du tableau comparatif en joules par transaction 46
3.7 Données brutes de la simulation pour le calcul énergétique 46
3.8 Estimation de la consommation PoEWAL-LT selon la difficulté 47
3.9 Comparaison énergétique PoEWAL vs PoEWAL-LT 48
3.10 Facteurs multiplicatifs par rapport à PoEWAL-LT 48
3.11 Synthèse des performances PoEWAL-LT (mesures Cooja) 51
3.12 Positionnement de PoEWAL-LT par rapport aux algorithmes de référence 51

Liste des Algorithmes

1 Ajustement dynamique de la difficulté 27
2 Calcul du Work Score Multicritère 28
3 Calcul du Proof of Work Factor 30
4 Calcul du Luck Factor 31
5 Calcul du Score Final 31
6 Sélection Multi-Niveaux du Validateur 31
7 Vérification Cryptographique du Validateur 32
8 Résolution de Collision à 3 Niveaux 32
9 Détection Byzantine Multicritère 33
10 Consensus BFT (Byzantine Fault Tolerance) 34

Liste des acronymes

BFT :

Byzantine Fault Tolerance 23, 44,

DAG :

Directed Acyclic Graph (Graphe orienté acyclique) 45,

dBFT :

delegated Byzantine Fault Tolerance 24,

DPoS :

Delegated Proof of Stake (Preuve d'enjeu déléguée) 6, 41, 45,

GPS :

Global Positioning System (Système de Positionnement par Satellite) 38, 43, 44, 52, 54,

IoT :

Internet of Things (Internet des Objets) 3, 21, 36, 51–53,

LTS :

Long-Term Support 27,

MSB :

Most Significant Bits (Bits les plus significatifs) 42, 47,

P2P :

Pair-à-pair (Peer-to-Peer) 4,

PBFT :

Practical Byzantine Fault Tolerance 7, 8, 41, 45,

PoA :

Proof of Authority (Preuve d'autorité)

PoA :

Proof of Activity (Preuve d'activité) 6, 7,

PoBT :

Proof of Behavior Trust 8,

PoC :

Proof of Capacity (Preuve de capacité) 7,

PoET :

Proof of Elapsed Time (Preuve de temps écoulé) 7, 8,

PoEWAL :

Probabilistic Proof of Elapsed Work and Luck (Lightweight) 41, 45, 52,

PoEWAL-LT :

Proof of Elapsed Work and Luck for Logistics Traceability 25, 26, 36, 41, 43–45, 50–53,

PoI :

Proof of Importance (Preuve d'importance) 6,

PoL :

Proof of Luck (Preuve de chance)

PoS :

Proof of Stake (Preuve d'enjeu) 6, 7, 41, 45,

PoW :

Proof of Work (Preuve de travail) 6, 41,

TEE :

Trusted Execution Environment 7, 8,

TPS :

Transactions Par Seconde 38, 39, 52,

Introduction Générale

La convergence de la technologie blockchain et de l'Internet des objets (IoT) offre un potentiel sans précédent pour transformer la logistique en permettant un suivi et une gestion efficaces des actifs, des chaînes d'approvisionnement et des transactions. Cependant, l'utilisation de la technologie blockchain dans les réseaux IoT logistiques pose des défis uniques en raison des contraintes des nœuds IoT, notamment leurs faibles capacités de calcul, de stockage, d'énergie et leur vulnérabilité aux menaces de sécurité. Au cœur de cette complexité réside le besoin crucial d'un algorithme de consensus optimisé pour assurer la sécurité des échanges de données et des transactions au sein de ces réseaux.

1. Problématique

L'émergence des technologies de nos jours fait élargir le champ de possibilités et de réflexions tout en ayant un réel impact évolutif sur les activités dans divers secteurs. Ainsi, depuis un certain nombre d'années le concept de la blockchain nous conduit à une décentralisation totale, un système sécurisé sans intermédiaire retraçant toutes les opérations de la chaine sans possibilité de falsification résolvant des problèmes tout en se basant principalement sur des algorithmes de consensus. Cependant, comment adapter un algorithme de consensus dans une blockchain pour qu'il soit à la fois sécurisé, efficace et compatible avec les contraintes des réseaux IoT logistiques ? Ainsi, comment garantir la qualité, la fiabilité des données dans des systèmes distribués avec des noeuds vulnérables tout en répondant aux exigences logistiques en temps réel ? Pour répondre à ces interrogations, notre travail s'oriente vers l'étude et l'analyse des différents algorithmes de consensus existants, afin de proposer un algorithme adéquat et tenant compte des exigences des réseaux IoT logistique.

2. Contexte et justification

Les réseaux IoT dans le domaine de la logistique sont confrontés à des contraintes spécifiques telles que des appareils avec des capacités de calcul, de stockage et d'énergie limitées, ainsi qu'une exposition à diverses menaces de sécurité. La technologie blockchain est reconnue pour son potentiel à fournir un mécanisme de consensus sécurisé et décentralisé, mais les algorithmes de consensus utilisés dans la technologie blockchain conventionnelle peuvent ne pas être adaptés aux contraintes

des réseaux IoT logistiques.
Dans ce contexte, il est nécessaire d'analyser les solutions existantes et d'explorer la conception d'un algorithme de consensus spécifiquement adapté aux réseaux IoT logistiques. Celui-ci devrait optimiser l'utilisation des ressources limitées des nœuds IoT, assurer la résilience et la sécurité du consensus malgré les menaces potentielles, tout en soutenant les besoins spécifiques de la logistique tels que le suivi des actifs, la gestion des chaînes d'approvisionnement et la gestion des transactions pour une meilleure traçabilité logistique.

3. Objectifs de l'étude

L'objectif principal de ce travail est d'analyser les algorithmes de consensus pour proposer un système mieux adapté aux réseaux IoT logistiques, garantissant ainsi la sécurité des échanges de données et des transactions au sein de ces réseaux. Il s'agit spécifiquement de :

- Analyser profondément les différents algorithmes de consensus classiques utilisés dans la blockchain ;
- Analyser les bases des algorithmes de consensus dont la théorie des jeux, les protocoles cryptographiques et la théorie de la fiabilité des systèmes distribués ;
- Identifier les besoins et contraintes uniques des réseaux IoT dans le domaine de la logistique ;
- Proposer un système de consensus adapté aux réseaux IoT logistiques ;
- Simuler un environnement adéquat pour tester l'algorithme proposé ;
- Évaluer les performances et la résilience du mécanisme de consensus proposé ;

4. Organisation du mémoire

Commençant par une introduction et clôturé par une conclusion, le présent mémoire comprend trois chapitres. Le premier présente l'état de l'art sur les algorithmes de consensus et les réseaux IoT destinés à la logistique. Le deuxième chapitre parle de l'analyse et la proposition d'un système de consensus optimisé et mieux adapté aux réseaux IoT logistique et un troisième où s'est fait la présentation des résultats et discussions.

Chapitre 1

Etat de l'art sur les algorithmes de consensus et les réseaux IoT destinés à la logistique

Introduction

L'évolution rapide de la technologie a donné naissance à de nouveaux paradigmes visant à répondre aux exigences croissantes en matière de sécurité, de fiabilité et de transparence des échanges de données dans les systèmes distribués. Parmi ceux-ci, la blockchain s'impose comme une solution innovante, notamment grâce à son mécanisme fondamental : l'algorithme de consensus. Ce mécanisme permet aux différents nœuds d'un réseau de parvenir à un accord commun sans autorité centrale, garantissant ainsi l'intégrité des transactions. Parallèlement, l'Internet des Objets IoT s'impose comme une technologie clé dans la transformation numérique de secteurs critiques, notamment celui de la logistique. En permettant l'interconnexion et la communication entre des dispositifs intelligents, l'IoT offre des opportunités considérables en matière de suivi en temps réel, d'optimisation des flux de marchandises et de prise de décision automatisée. Toutefois, cette évolution s'accompagne de défis majeurs liés à la sécurité des communications, à la gestion de la confiance entre les dispositifs, et à la résilience des systèmes dans des environnements distribués, dynamiques et contraints en ressources. Dans cette optique, il devient essentiel d'examiner de manière approfondie les algorithmes de consensus existants afin d'évaluer leur pertinence face aux contraintes spécifiques des réseaux IoT appliqués à la logistique. Il sera question dans ce chapitre de présenter les fondements conceptuels, les différents types d'algorithmes de consensus, ainsi que les caractéristiques et défis propres aux réseaux IoT dans le contexte logistique.

1.1 Definition des concepts

1.1.1 Blockchain

La blockchain est une technologie de registre distribué pair-à-pair P2P, de stockage et de transmission d'informations sans autorité centrale, permettant d'établir la confiance et le consensus dans les réseaux décentralisés. Elle repose sur une structure de blocs chaînés entre eux, contenant chacun un ensemble de transactions validées par un mécanisme de consensus. L'image suivante montre quelques chaines de blocs avec les composants d'un bloc [27] :

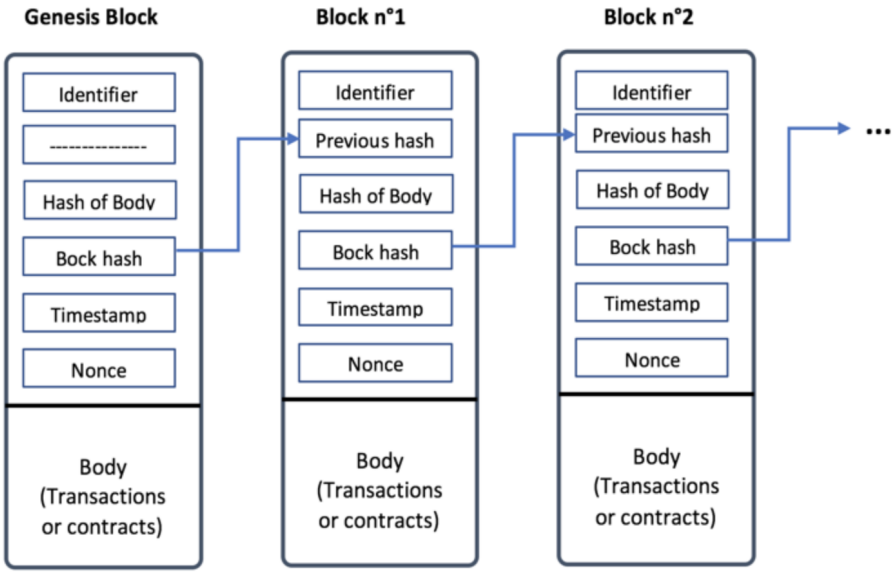


FIGURE 1.1 : Structure des blocs

En outre, on distingue généralement quatre types de blockchains que sont :

- Publique : ouverte à tous (ex. : Bitcoin, Ethereum), transparente mais énergivore.
- Privée : contrôlée par une seule entité (utilisée dans les entreprises).
- Hybride : mélange de contrôle centralisé et ouvert (ex. : VeChain).
- Consortium : réseau restreint de nœuds approuvés, souvent interentreprises (ex. : Hyperledger Fabric) [28].

1.1.2 Algorithme de consensus

Partant de la définition du dictionnaire du mot "consensus" qui est un accord sans opposition catégorique des différentes parties d'une discussion sur un sujet déterminé, un algorithme de consensus est une mise en conformité de chaque acteur du réseau blockchain afin de se mettre d'accord sur la création et l'enchainement de chaque bloc. Cependant, l'absence d'une autorité centrale dans la blockchain signifie que tout individu peut agir en tant que nœud, et tous les nœuds du réseau sont considérés comme non fiables, d'où la nécéssité d'un mécanisme de consensus pour que tous les

[1]Figure 1.1 : https://blog.eleven-labs.com/fr/bases-blockchain/

nœuds s'accordent sur l'état du réseau. La méthode de consensus détermine l'authenticité des transactions, dicte le comportement de chaque nœud et influence considérablement les performances de la blockchain. Le mécanisme de consensus utilisé dans la technologie Blockchain détermine des facteurs tels que l'efficacité, l'équité, la sécurité et l'intégrité, l'immuabilité [29].

1.1.3 Internet des objets

L'internet des objets est l'interconnexion entre l'internet et les objets. Par extension, c'est un réseau d'objets et de terminaux connectés équipés de capteurs (et d'autres technologies) leur permettant de transmettre et de recevoir des données entre eux et avec d'autres systèmes. Voici en illustration quelques domaines d'applications de l'IoT tels que la logistique, la santé, l'agriculture, etc [24].

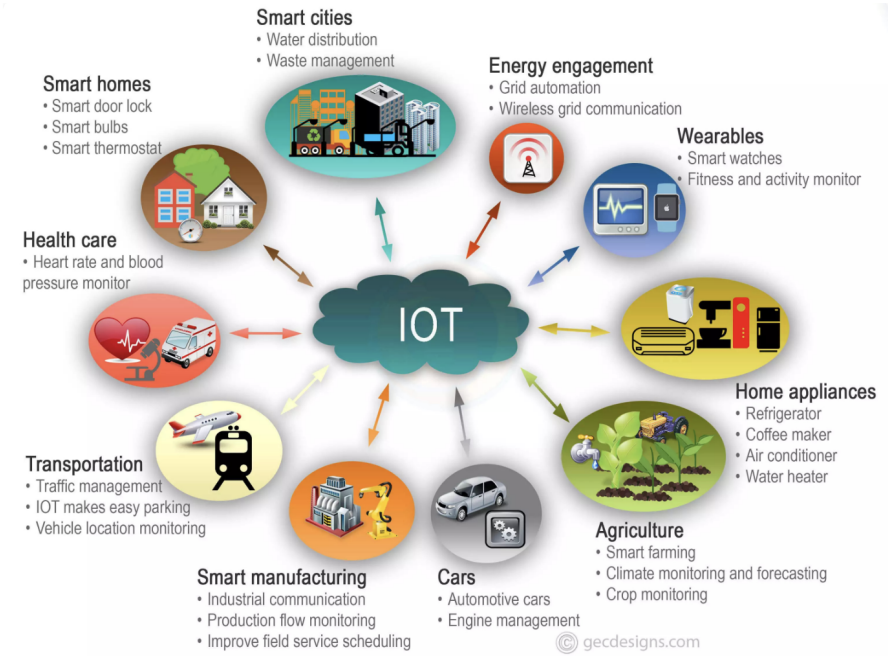


FIGURE 1.2 : Domaines d'application de l'Internet des Objets (IoT)

1.1.4 Logistique

La logistique est un ensemble de moyens et de méthodes qui permettent de gérer tout ce qui concerne le transport et le stockage des produits d'une entreprise : véhicules nécessaires au transport, fournisseurs de l'entreprise, entrepôts, manutention…, en optimisant leur circulation pour minimiser les coûts et les délais. C'est une composante de la gestion de la chaîne d'approvisionnement.

1.2 Classification des algorithmes

Plusieurs algorithmes de consensus sont utilisés dans la technologie blockchain. Il existe deux types d'algorithmes de consensus : les algorithmes de consensus basés sur la preuve et les algorithmes de consensus basés sur le vote [21].

[2]Figure 1.2 : https://botpenguin.com/glossary/iot-communication-models

1.2.1 Algorithmes de consensus basés sur des preuves

1.2.1.1 Proof Of Work (PoW)

L'idée centrale de la preuve de travail réside dans la compétition pour la puissance de calcul. Le nœud exécutant le mécanisme de consensus (appelé mineur) utilise ses ressources de calcul pour l'opération de hachage afin de concourir pour le droit de générer le nouveau bloc avec des bonus. Le gagnant est le premier à obtenir une valeur de hachage inférieure à l'objectif annoncé. D'une part, la difficulté de calcul de PoW doit être suffisamment élevée pour empêcher le fork. D'autre part, cette difficulté de calcul élevée entraînerait une consommation elevée d'énergie.

1.2.1.2 Proof of Stake (PoS)

L'algorithme de preuve d'enjeu (PoS) repose sur l'idée que le créateur du prochain bloc doit être choisi selon diverses combinaisons de sélection aléatoire, de sa mise et de son ancienneté, ce qui peut offrir une bonne scalabilité. Cette idée a été introduite en 2011 pour la cryptomonnaie Peercoin, puis utilisée par d'autres, telles que Nxt et Blackcoin. Le nœud sélectionné pour la création du prochain bloc sera choisi selon un processus quasi aléatoire, la sélection dépendant des actifs stockés dans le portefeuille (ou le pool d'actions) associé à ce nœud. Cette méthode ne nécessite pas de puissance de calcul élevée pour valider les preuves ; par conséquent, les mineurs ne recevront aucune récompense, à l'exception des frais de transaction [20]. En d'autres termes, la PoS est une méthode pour atteindre le consensus distribué dans un réseau blockchain qui ne demande pas aux utilisateurs d'utiliser leur puissance de calcul, mais de prouver la propriété d'un certain montant de crypto-monnaie.

1.2.1.3 Delegated Proof of Stake (DPoS)

La preuve d'enjeu déléguée (DPoS), une variante de PoS est un mécanisme de consensus qui réduit le nombre de nœuds d'un réseau blockchain à un petit nombre de validateurs. Ces validateurs (délégués) sont choisis par les détenteurs de jetons, au prorata de ce qu'ils possèdent. Cela offre les avantages d'un contrôle semi-centralisé du réseau, tels que l'efficacité et la rapidité, tout en préservant les caractéristiques de la décentralisation. La DPoS a été principalement inventée pour améliorer BitShares, car elle est plus efficace et consomme moins d'énergie que la PoW et la PoS [25].

1.2.1.4 Proof of Activity (PoA)

La preuve d'activité (PoA) est une combinaison d'algorithmes PoW et PoS proposée pour les applications de cryptomonnaie. Elle offre un bon niveau de sécurité, une faible surcharge réseau et un faible espace de stockage. Cet algorithme synchronise les transactions de manière suffisamment claire. Il offre également un niveau de sécurité élevé. Il est difficile pour un attaquant de monopoliser le processus de création de blocs, car cela nécessite une part importante des monnaies créées jusqu'à présent [21].

1.2.1.5 Proof of Importance (PoI)

La preuve d'importance (PoI) a été introduit par NEM (New Economy Movement). L'échelle d'importance des nœuds détermine les nœuds éligibles pour ajouter un bloc à la blockchain. Cet algorithme s'appuie sur l'analyse du graphe de transactions pour agréger les nœuds. Grâce à cette méthode, la priorité est attribuée aux nœuds les plus importants en fonction de leur hachage.

1.2.1.6 Proof of Capacity

Le Proof of Capacity (PoC) est un algorithme de consensus dans lequel les nœuds utilisent leur espace de stockage disponible pour miner, au lieu de leur puissance de calcul. Le principe repose sur le pré-calcul de solutions cryptographiques (appelées nonces) qui sont ensuite enregistrées sur le disque dur lors d'une phase appelée plotting. Lors de la validation d'un bloc, chaque mineur lit rapidement les nonces déjà stockés pour trouver la solution la plus proche du défi cryptographique du réseau. Le nœud ayant la réponse la plus adéquate gagne le droit de forger le bloc suivant. Le PoC est donc plus écologique et plus accessible, car il repose sur l'utilisation de l'espace disque, une ressource largement disponible. Il est notamment utilisé dans des cryptomonnaies comme Burstcoin ou Chia, et bien qu'intéressant pour les systèmes distribués à faible consommation, il n'est pas encore largement adopté dans les contextes IoT en raison de la taille requise pour le stockage local [20].

1.2.1.7 Proof of Luck

Le Proof of Luck (PoL) est un mécanisme de consensus basé sur la génération aléatoire d'un « score de chance » dans un environnement sécurisé matériellement. Chaque nœud du réseau utilise une enclave de confiance (Trusted Execution Environment – TEE), telle qu'Intel SGX, pour exécuter un algorithme déterministe qui produit un nombre aléatoire vérifiable. Le nœud ayant obtenu la valeur la plus élevée (« le plus chanceux ») est sélectionné pour créer le bloc suivant. Contrairement à PoW, cette méthode ne nécessite pas de puissance de calcul intensive, et contrairement à PoS ou PBFT, elle évite les échanges massifs de messages. Son avantage principal réside dans sa rapidité et son efficacité énergétique. Cependant, il dépend de la confiance dans l'intégrité du matériel TEE, ce qui peut constituer une limite de décentralisation. Ce protocole a été introduit par Milutinovic et al. (2016) dans une perspective d'optimisation des blockchains pour l'IoT et les environnements contraints [15].

1.2.1.8 Proof of Authority (PoA)

Le Proof of Authority (PoA) repose sur un ensemble restreint de validateurs désignés qui possèdent une autorité reconnue pour valider les blocs. Ces validateurs sont sélectionnés selon des critères pré-établis (identité vérifiée, appartenance à une organisation, etc.). Contrairement à PoW ou PoS, PoA n'utilise ni calcul intensif ni mises financières. Chaque validateur peut produire des blocs selon un ordre tournant ou aléatoire, avec très peu de latence et une efficacité énergétique optimale. Ce mécanisme est particulièrement adapté aux blockchains privées ou consortium, comme celles utilisées dans les systèmes logistiques d'entreprise. Toutefois, il sacrifie la décentralisation pour la performance, ce qui peut susciter des craintes en matière de gouvernance ou de collusion entre validateurs. PoA est souvent utilisé dans les projets comme Ethereum Kovan ou VeChain, et est recommandé dans les contextes où la confiance entre parties est établie [7].

1.2.1.9 Proof of Elapsed Time (PoET)

Le Proof of Elapsed Time (PoET) est un mécanisme proposé par Intel pour les blockchains permissionnées comme Hyperledger Sawtooth. Il s'appuie sur l'usage d'un environnement d'exécution sécurisé (TEE) tel qu'Intel SGX. Chaque nœud demande à son TEE un délai d'attente aléatoire et impartial. Le nœud ayant obtenu le délai le plus court est sélectionné pour proposer le bloc suivant. Ce système garantit l'équité du choix sans consommation énergétique excessive, contrairement à PoW. De plus, chaque nœud est contraint de respecter le délai généré, ce qui rend les tricheries dif-

ficiles si le TEE est fiable. PoET allie ainsi performance, efficacité énergétique et équité probabiliste. Cependant, il dépend fortement du matériel Intel, ce qui limite la portabilité et soulève des préoccupations sur la centralisation matérielle et complique son adoption dans des environnements IoT très hétérogènes[1].

1.2.1.10 Proof of Behavior Trust (PoBT)

PoBT est un algorithme émergent, conçu spécialement pour les environnements IoT. Il se base sur l'évaluation dynamique du comportement des nœuds : chaque nœud est noté selon son historique de participation, son honnêteté, et sa disponibilité. Ces scores influencent son poids dans le processus de validation. Un nœud de bonne réputation aura plus de chances de valider un bloc. Ce mécanisme favorise les comportements coopératifs et décourage les attaques internes. PoBT est léger, évolutif, et résilient face aux comportements byzantins partiels, ce qui le rend particulièrement adapté à la logistique distribuée avec des capteurs mobiles et intermittents [2]. Certains le classent aussi côté vote car il inclut des accords locaux (on peut dire dire qu'il est hybride).

1.2.2 Algorithmes de consensus basés sur des votes

1.2.2.1 Practical Byzantine Fault Tolerance (PBFT)

PBFT est un algorithme de consensus tolérant aux fautes byzantines, développé pour permettre à un réseau distribué de valider des transactions même en présence de nœuds malveillants ou défaillants. Il fonctionne par un mécanisme de vote : un nœud principal (le leader) propose un bloc, et les autres nœuds (réplicas) doivent répondre et confirmer par plusieurs étapes (pré-préparation, préparation, confirmation). Si au moins deux tiers des nœuds sont honnêtes, le consensus est atteint. Il est rapide et consomme peu de ressources, mais il ne passe pas à l'échelle facilement : le nombre de messages échangés augmente de façon quadratique avec le nombre de nœuds. Il est bien adapté aux blockchains privées ou de consortium, notamment dans des contextes logistiques ou industriels fermés [14].

1.2.2.2 Raft

Raft est un algorithme de consensus développé pour la tolérance aux pannes dans les systèmes répartis, à des fins de simplicité et de compréhension. Dans Raft, un leader est élu parmi les nœuds et il est responsable de proposer des blocs (ou entrées de log), que les autres nœuds (followers) doivent valider et répliquer. En cas de défaillance du leader, une nouvelle élection est déclenchée. Raft est efficace dans des environnements avec un faible nombre de nœuds et une faible latence réseau. Il est souvent utilisé dans des systèmes embarqués ou cloud internes (ex. : etcd, Consul), mais peu dans les blockchains publiques car il repose sur une certaine confiance initiale [1].

1.2.3 Tableau comparatif des algorithmes de consensus exitants

Algorithme	Principe général	Fonctionnement	Avantages	Limites
PoW	Résolution de puzzles cryptographiques	Le premier mineur à résoudre le puzzle valide le bloc	Très sécurisé, bien éprouvé	Très énergivore, lent, peu adapté à l'IoT
PoS	Sélection selon la mise (tokens possédés)	Le validateur est choisi selon la quantité de tokens misés	Économie d'énergie, rapide	Risque de centralisation, attaque du « rien à perdre »
DPoS	Délégués élus par les participants	Les producteurs de blocs sont choisis par vote	Scalabilité, faible latence	Moins décentralisé, dépend du bon fonctionnement des votes
PBFT	Accord majoritaire malgré les nœuds malveillants	Vote entre nœuds (pré-prepare, prepare, commit)	Tolérance aux fautes, rapide	Échange massif de messages, pas adapté aux grands réseaux
Raft	Consensus par un leader élu	Un nœud leader est élu et propose les blocs	Simplicité, robustesse, faible latence	Moins décentralisé, pas byzantin
Tendermint	BFT avec votes en 3 étapes	Propose $\rightarrow$ Prevote $\rightarrow$ Precommit	Rapide, faible consommation	Besoin de validateurs fixes
PoA	Validation par des nœuds approuvés	Des entités identifiées sont autorisées à valider	Rapide, économe, bon pour les blockchains privées	Très centralisé, peu transparent
PoL	Sélection aléatoire via un TEE sécurisé	Enclave sécurisée génère un score de chance ; le plus « chanceux » forge le bloc	Équitable, économe, rapide	Dépendance au matériel (SGX), peu adopté
PoA (Activity)	Mix entre PoW et PoS : présence + participation	Bloc miné via PoW puis complété par validateurs PoS	Combine sécurité et efficacité	Complexité de mise en œuvre, peu standardisé
PoI	Validation pondérée par l'activité et les interactions réseau	Le score dépend des transactions et des connexions	Encourage les utilisateurs actifs	Complexe à calculer
PoC	Utilisation de l'espace disque	Stockage de nonces pré-calculés	Faible consommation énergétique	Besoin d'espace disque important
PoEWAL	Validation basée sur la conscience énergétique	Les nœuds respectent un seuil énergétique	Idéal pour l'IoT, économe	Peu implémenté

1.3 Fondements des consensus

Les algorithmes de consensus se basent sur des principes fondamentaux de la théorie des jeux, de la cryptographie et de la théorie de la fiabilité des systèmes distribués.

1.3.1 Théorie des jeux

D'après Roger Myerson, la théorie des jeux peut être définie comme l'étude des modèles mathématiques de conflit et de coopération entre preneurs de décisions rationnels[15]. Elle a été développée au cours du XX e siècle et offre un cadre analytique puissant pour comprendre les décisions prises par des individus, des entreprises, des gouvernements et d'autres entités, en tenant compte de l'impact de leurs choix sur les résultats de chacun. Elle connait des applications directes dans de nombreux domaines, tels que l'économie, la biologie, l'informatique, la sociologie, la psychologie.

1.3.1.1 Concepts liés à la théorie des jeux

- **Jeu** : c'est une situation où les joueurs sont amenés à faire des choix parmi un certain nombre d'actions possibles, tout en tenant compte des règles du jeu prédéfinies. Ainsi à chaque choix est associé un gain, positif ou négatif pour chacun des participants.
- **Joueur** : c'est un acteur ou une entité pouvant être une personne, une entreprise, un gouvernement ...etc, agissant dans leur propre intérêt selon le principe de la rationalité. On notera $\mathcal{N}$ l'ensemble des joueurs participant à un même jeu :

  $$\mathcal{N} = \{1, \ldots, N\}$$

  où N désigne leur nombre, avec $N \geq 2$.
- **Stratégie** : elle décrit totalement le comportement du joueur tout au long du jeu en fonction de toutes situations auxquelles il pourrait etre confronté.
- **Stratégie pure** : c'est une action ou un plan d'action choisie(s) avec certitude par un joueur i. On note S_i l'ensemble des stratégies pures du joueur i.
- **Stratégie mixte** : du joueur i est une distribution de probabilité α_i sur l'ensemble des stratégies pures du i-ème joueur. Une stratégie mixte pour le joueur i est représentée par un vecteur α défini comme suit :

$$\Delta_i = \left\{ \alpha_i = (\alpha_i^1, \alpha_i^2, \ldots, \alpha_i^{n_i}) \in \mathbb{R}^{n_i} \;\middle|\; 0 \leq \alpha_i^j \leq 1 \; \forall j \in \{1, \ldots, n_i\}, \; \sum_{j=1}^{n_i} \alpha_i^j = 1 \right\} \tag{1.1}$$

où Δ_i est l'ensemble des stratégies mixtes du joueur i, n_i est le nombre de stratégies pures du joueur i, et α_i^j représente la probabilité que le i-ème joueur choisisse la j-ème stratégie.

Une stratégie est dite dominante pour un joueur si elle lui offre un meilleur résultat (c'est-à- dire un gain plus élevé) quel que soit le choix des autres joueurs et dite dominée pour un joueur si elle lui offre un résultat moins favorable que d'autres stratégies qu'il pourrait choisir, compte tenu des choix des autres joueurs.

1.3.1.2 Types de jeux

- **Jeux coopératifs et non-coopératifs** : Dans un jeu coopératif, les joueurs collaborent pour atteindre un objectif commun (un gain collectif). Les jeux coopératifs se concentrent souvent sur la répartition équitable des gains entre les joueurs. Par contre, dans un jeu non-coopératif, la

conccurence et la rivalité entrent en jeu. Ainsi, les joueurs agissent de manière indépendante et cherchent à maximiser leur propre gain personnel.

- **Jeux à information parfaite et imparfaite** : Dans un jeu à information parfaite, chacun des joueurs, au moment de choisir son action, a une connaissance parfaite de l'ensemble des décisions prises antérieurement par les autres joueurs. Si au moins un des joueurs ne connait pas, à un moment du déroulement du jeu, ce qu'a joué un autre joueur, le jeu est dit à information imparfaite. Ceci peut arriver dans le cas où on cache l'information aux joueurs ou parce que les joueurs jouent simultanément.

- **Jeu à Somme nulle** : On dit qu'un jeu est à somme nulle si le montant total des gains à la fin de la partie est à somme nulle, en d'autre termes si le montant total gagné par un joueur est égale au montant perdu par l'autre, de sorte que la somme des gains et des pertes est constante.

- **Jeux statiques ou stratégiques** : Les jeux statiques sont des jeux à un coup où tous les joueurs prennent leurs décisions simultanément, sans avoir connaissance des actions choisies par les autres joueurs au moment de prendre leur propre décision. Dans les jeux statiques, chaque joueur choisit sa stratégie sans tenir compte de l'ordre dans lequel les autres joueurs prendront leurs décisions. Les jeux statiques sont à information complète et sont souvent représentés sous forme de matrices de gains pour chaque combinaison possible de stratégies. Les équilibres de Nash sont couramment utilisés pour analyser ces jeux. Ils sont représentés sous la forme :

$$\langle \mathcal{N}, \{S_i\}_{i\in\mathcal{N}}, \{f_i\}_{i\in\mathcal{N}} \rangle$$

où :

- $\mathcal{N} = \{1, \ldots, n\}$ est l'ensemble des joueurs. Un joueur quelconque est appelé i, donc $i \in \{1, \ldots, n\}$;
- $S_i \subset \mathbb{R}^{n_i}$ désigne l'ensemble des stratégies du joueur $i \in \mathcal{N}$;
- $S = \prod_{i=1}^{n} S_i$ est l'ensemble des issues du jeu ;
- f_i est la fonction d'utilité du joueur $i \in \mathcal{N}$.

Exemple : Considérons le jeu suivant appelé « pile ou face ».
Dans ce jeu, deux enfants placent secrètement une pièce de un dinar dans le creux de leurs mains. Puis, simultanément, chaque enfant ouvre sa main et montre à l'autre sa pièce.
Si les deux pièces sont toutes les deux de côté pile ou toutes les deux de côté face, l'enfant 1 donne un dinar à l'enfant 2.
Dans le cas contraire, c'est l'enfant 2 qui donne un dinar à l'enfant 1.

C'est un jeu à information imparfaite, car les deux enfants ouvrent les mains simultanément, sans savoir comment l'un et l'autre avaient placé la pièce dans leur main.

La matrice de gain de ce jeu apparaît dans la figure suivante :

TABLE 1.1 : Matrice de gain du jeu « pile ou face »

		Enfant 2
Enfant 1	**Pile**	**Face**
Pile	$(-1, 1)$	$(1, -1)$
Face	$(1, -1)$	$(-1, 1)$

Dans les quatre cases du tableau 1.1 :

- *Le premier chiffre est le paiement que reçoit l'enfant 1.*
- *Le second chiffre est le paiement que reçoit l'enfant 2*[22].

1.3.1.3 Equilibre de Nash en stratégies pures

L'équilibre de Nash est un concept clé en théorie des jeux, introduit par le mathématicien John Nash en 1950. Il décrit une situation dans laquelle chaque joueur, agissant de manière rationnelle, choisit la meilleure stratégie possible en tenant compte des choix des autres joueurs. En d'autres termes, c'est un ensemble de choix de stratégies tel que personne n'a intérêt à dévier de sa stratégie, étant donné les choix des autres.

Définition formelle : Un profil de stratégies $(s^* = s_1^*, s_2^*, \ldots, s_n^*)$ est un équilibre de Nash si aucun des joueurs n'a intérêt à dévier unilatéralement de sa stratégie s_i^* (lorsque les autres joueurs conservent leurs stratégies s_{-i}^*).
Autrement dit :

$$\forall i \in \mathcal{N},\ \forall s_i \in S_i, \quad f_i(s_1^*, s_2^*, \ldots, s_n^*) \geq f_i(s_i, s_{-i}^*) \tag{1.2}$$

Pour déterminer un équilibre de Nash en stratégies pures dans un jeu, il faut :

- Repérer les meilleures réponses de chaque joueur aux différentes stratégies possibles des autres joueurs ;
- Rechercher un profil de stratégies $s^* = (s_1^*, s_2^*, \ldots, s_n^*)$ dans lequel la stratégie de chaque joueur est la meilleure réponse aux stratégies des autres.

Exemple : (Dilemme du prisonnier)
Deux cambrioleurs, notés 1 et 2, sont arrêtés par la police et placés en garde à vue dans des cellules différentes. La police n'a pas assez de preuves, donc elle leur propose un marché : Si les deux coopèrent en ne se dénonçant pas mutuellement, chacun va écoper d'un an de prison. Si l'un trahit en accusant l'autre, alors celui qui a trahi va être libéré, et celui qui s'est tu va écoper 10 ans de prison. Si les deux se dénoncent mutuellement, alors chacun va écoper de 5 ans de prisons.

TABLE 1.2 : Matrice de gain du Dilemme du prisonnier

		Suspect 2
Suspect 1	**Se taire (ST)**	**Dénoncer (D)**
Se taire (ST)	$(-1, -1)$	$(-10, 0)$
Dénoncer (D)	$(0, -10)$	$(-5, -5)$

Pour le suspect 1 :

$$f_1(ST, ST) = -1 < f_1(D, ST) = 0$$

Pour le suspect 2 :

$$f_2(ST, ST) = -1 < f_2(ST, D) = 0$$

$\Rightarrow (ST, ST)$ *n'est pas un équilibre de Nash. Par contre le profil* (D, D) *est un équilibre de Nash. En effet :*

$$f_1(D, D) = -5 > f_1(ST, D) = -10$$

$$f_2(D, D) = -5 > f_2(D, ST) = -10$$[22]

1.3.1.4 Equilibre de Nash en stratégies mixtes

La notion d'équilibre de Nash en stratégies mixtes est tout simplement une extension du concept d'équilibre de Nash en stratégies pures. Il s'agit d'un ensemble de stratégies mixtes $\sigma_1^*, \ldots, \sigma_i^*, \ldots, \sigma_n^*)$ qui possède des arguments de stabilité analogues à ceux qui caractérisent un équilibre de Nash en stratégies pures.

Plus précisément : si le joueur i était certain que les autres participants retiennent les stratégies mixtes $\sigma_1^*, \ldots, \sigma_{i-1}^*, \sigma_{i+1}^*, \ldots, \sigma_n^*)$ alors ce joueur choisirait sa stratégie mixte σ_i^*, et cela est vrai pour tout $i \in \{1, \ldots, n\}$. Ainsi, lorsque les joueurs envisagent l'équilibre en stratégies mixtes $(\sigma_1^*, \ldots, \sigma_i^*, \ldots, \sigma_n^*)$ comme l'équilibre possible du jeu, aucun d'entre eux n'a intérêt à dévier unilatéralement de cet équilibre.

Définition formelle : Un ensemble de stratégies mixtes $(\sigma_1^*, \ldots, \sigma_i^*, \ldots, \sigma_n^*)$ constitue un équilibre de Nash en stratégies mixtes si, pour chaque joueur $i \in \{1, \ldots, n\}$, la stratégie mixte σ_i^* est une meilleure réponse aux stratégies mixtes $(\sigma_1^*, \ldots, \sigma_{i-1}^*, \sigma_{i+1}^*, \ldots, \sigma_n^*)$ des autres joueurs. La stratégie mixte correspond à une distribution de probabilités sur l'ensemble des stratégies pures. Pour déterminer un équilibre de Nash en stratégies mixtes dans un jeu, il faut :

- Repérer les meilleures réponses de chaque joueur aux différentes stratégies mixtes possibles des autres joueurs ;
- Rechercher un ensemble de stratégies mixtes $(\sigma_1^*, \ldots, \sigma_i^*, \ldots, \sigma_n^*)$ dans lequel la stratégie de chacun est la meilleure réponse aux stratégies des autres[4].

1.3.2 Cryptographie

La cryptographie est l'art et la science de la protection de l'information en utilisant des techniques mathématiques pour la rendre illisible pour les personnes non autorisées. Elle vise à assurer la confidentialité, l'intégrité et l'authenticité des données, que ce soit au repos, en transit ou en cours d'utilisation et la non-répudiation. En effet, c'est une technologie sur laquelle repose la sécurité de la blockchain. Elle utilise des algorithmes complexes pour chiffrer les informations, transformant les données lisibles en un format codé qui ne peut être déchiffré que par ceux qui possèdent la clé correspondante. Cette méthode de chiffrement est essentielle pour protéger les transactions et les communications au sein du réseau blockchain contre les accès non autorisés et les tentatives de fraude. Elle repose sur deux principaux types de chiffrement : symétrique, où une seule clé est utilisée pour chiffrer et déchiffrer les données, et asymétrique, qui utilise une paire de clés pour une sécurité renforcée.

Les fonctions de hachage, un autre pilier de la cryptographie dans la blockchain, jouent un rôle crucial en assurant l'intégrité des données. En convertissant les informations en une empreinte numérique unique de taille fixe, les fonctions de hachage permettent de vérifier l'authenticité et la non-altération des données sans nécessiter l'accès à l'ensemble des informations originales. Cette caractéristique est fondamentale pour le processus de création et de liaison des blocs dans la blockchain, garantissant une chaîne de blocs sécurisée et inaltérable. Grâce à ces mécanismes cryptographiques, la blockchain offre un environnement extrêmement sécurisé pour les transactions numériques, établissant un nouveau standard pour la protection des données dans le monde numérique.

Chaque transaction est signée numériquement par la clé privée de l'expéditeur, prouvant ainsi son origine et son intégrité. Cette signature est ensuite vérifiée sur le réseau à l'aide de la clé publique correspondante, sans jamais révéler la clé privée elle-même. Les fonctions de hachage transforment les transactions en un résumé cryptographique unique, garantissant que chaque bloc de transactions est unique et pratiquement impossible à altérer. Cette méthode cryptographique crée un environnement sécurisé où les transactions sont à la fois transparentes et vérifiables, renforçant la confiance dans le système blockchain sans compromettre la sécurité des données des utilisateurs.

En chiffrant les données stockées sur la blockchain, elle assure que seules les parties autorisées peuvent accéder à ces informations, protégeant ainsi la vie privée des utilisateurs : c'est la confidentialité [2].

Les mécanismes de consensus reposent sur des fondements cryptographiques robustes :

1.3.2.1 Fonctions de hachage

Les fonctions de hachage cryptographiques sont des algorithmes mathématiques fondamentaux en cryptographie qui transforment des données de taille arbitraire en une empreinte numérique de taille fixe, appelée "hash" ou "condensé".

Les fonctions de hachage constituent la pierre angulaire de nombreux protocoles de consensus. Une fonction de hachage cryptographique $H : 0, 1^* \rightarrow 0, 1^n$ doit satisfaire :

- Résistance à la préimage : Pour tout y dans l'image de H, il est difficile de trouver x tel que $H(x) = y$
- Résistance à la seconde préimage : Pour tout x, il est difficile de trouver $x' \neq x$ tel que $H(x) = H(x')$
- Résistance aux collisions : Il est difficile de trouver deux messages distincts x et x' tels que $H(x) = H(x')$ [19]

1.3.2.2 Signature numérique

La signature numérique constitue un mécanisme fondamental dans les systèmes de sécurité informatique, en particulier dans le cadre de la confidentialité, de l'intégrité des données et de l'authentification. Elle est largement utilisée dans des domaines tels que la messagerie sécurisée, les transactions électroniques, les documents légaux ou encore la blockchain. Contrairement à une signature manuscrite, une signature numérique repose sur des procédés cryptographiques rigoureux, principalement basés sur la cryptographie asymétrique.

Le principe de la signature numérique consiste à associer à un document une valeur unique appelée signature permettant de vérifier que :

- le document provient bien de l'expéditeur (authenticité) ;
- le document n'a pas été modifié (intégrité) ;
- l'expéditeur ne peut nier avoir signé (non-répudiation).

La signature numérique repose sur une paire de clés asymétriques : une clé privée (gardée secrète par le signataire) et une clé publique (diffusée à tous). Le processus de signature utilise également une fonction de hachage cryptographique [30].
Un schéma de signature numérique peut être formellement défini par un triplet de fonctions fondamentales :

- **KeyGen()** : une fonction de génération de clés qui produit une paire de clés asymétriques (pk, sk), où pk est la clé publique et sk la clé privée.
- **Sign(sk, m)** : une fonction de signature qui prend comme entrées un message m et une clé privée sk, et produit une signature s.
- **Verify(pk, m, s)** : une fonction de vérification qui prend comme entrées un message m, une signature s et une clé publique pk, et qui retourne `true` si la signature est valide, sinon `false`.

Le fonctionnement correct d'un schéma de signature s'exprime formellement par la relation suivante :

$$\text{Verify}(pk, m, \text{Sign}(sk, m)) = \texttt{true} \tag{1.3}$$

Ce modèle suppose l'existence d'une fonction de hachage cryptographique H résistante aux collisions, utilisée pour résumer le message avant la signature :

$$h = H(m) \tag{1.4}$$
$$s = \text{Sign}(sk, h) \tag{1.5}$$

Lors de la vérification, le récepteur calcule également le haché $h' = H(m)$ et déchiffre s à l'aide de la clé publique pour retrouver h. La signature est considérée comme valide si :

$$h = h' \quad \Leftrightarrow \quad \text{Verify}(pk, m, s) = \texttt{true} \tag{1.6}$$

Processus de signature et de vérification

– Génération de la signature
Soit m le message original à signer. L'émetteur commence par calculer une empreinte (ou *hash*) du message à l'aide d'une fonction de hachage cryptographique H :

$$h = H(m) \tag{1.7}$$

Ensuite, cette empreinte est chiffrée à l'aide de la clé privée sk (secret key) de l'émetteur afin d'obtenir la signature s :

$$s = \text{Sign}_{sk}(h) \tag{1.8}$$

Le couple (m, s) est alors transmis au destinataire.

– Vérification de la signature
À la réception du message, le destinataire procède aux étapes suivantes :

1. Il recalcule l'empreinte du message reçu à l'aide de la même fonction de hachage :

$$h' = H(m) \tag{1.9}$$

2. Il utilise la clé publique pk (public key) de l'émetteur pour déchiffrer la signature :

$$h = \text{Verify}_{pk}(s) \tag{1.10}$$

3. Il compare les deux empreintes h et h' :

$$h \stackrel{?}{=} h' \tag{1.11}$$

Si l'égalité est vérifiée, la signature est jugée **valide** : le message est authentique et n'a pas été modifié. Dans le cas contraire, la signature est **invalide** et le message est rejeté [30].

Ce mécanisme garantit :

- l'**authenticité** du message, car seul le détenteur de la clé privée a pu signer ;
- l'**intégrité**, car toute modification du message entraînerait un hash différent ;
- la **non-répudiation**, car la signature engage son auteur.

1.3.3 Théorie de la fiabilité des systèmes distribués

Les algorithmes de consensus s'inspirent des modèles de tolérance aux pannes et de tolérance aux fautes pour garantir la résilience et la cohérence du réseau face à des comportements malveillants, des pannes matérielles ou des défaillances logicielles [**?**].

1.3.3.1 Modèle de tolérance aux fautes : Crash Fault Tolerance (CFT)

Dans le modèle de *Crash Fault Tolerance (CFT)*, on suppose que les nœuds :

- peuvent tomber en panne,
- cessent éventuellement de répondre,
- mais ne produisent pas de comportements malveillants ni d'informations erronées.

Les algorithmes classiques reposant sur ce modèle incluent notamment *Paxos* et *Raft* [**?**]. L'objectif ici est de garantir qu'un système distribué continue à fonctionner correctement même si certains nœuds deviennent inaccessibles ou cessent de répondre.

1.3.3.2 Modèle de tolérance aux fautes : Byzantine Fault Tolerance (BFT)

Le modèle de *Byzantine Fault Tolerance (BFT)* est plus réaliste et plus exigeant. Il considère qu'un nœud peut :

- tomber en panne,
- envoyer de fausses informations,
- agir de manière malveillante,
- transmettre des messages différents à différents nœuds.

Ce modèle est issu du célèbre problème des *généraux byzantins*, qui illustre la difficulté d'atteindre un consensus fiable en présence d'acteurs potentiellement malveillants. L'objectif ici est de garantir l'accord collectif et la cohérence du système distribué, même en présence de participants défaillants ou malhonnêtes.

1.4 Réseaux IoT appliqués à la logistique

1.4.1 Définition et architecture

Un réseau IoT est un ensemble d'objets connectés capables de capter, traiter et transmettre des données en temps réel. Ces objets peuvent inclure des capteurs (température, humidité, vibration, GPS), des microcontrôleurs (ESP32, Arduino), et des interfaces de communication (Wi-Fi, LoRa, Bluetooth).

1.4.2 Contraintes spécifiques

Les objets IoT rencontrent plusieurs contraintes :

- Énergie limitée (batterie ou recharge solaire),
- Faible puissance de calcul,
- Connectivité intermittente,
- Besoins en communication légère et rapide

1.4.3 Cas d'usage dans la logistique

- Traçabilité des produits (chaîne du froid, produits de santé),
- Suivi en temps réel des colis ou véhicules (GPS, température),
- Détection d'anomalies : ouverture non autorisée, chute, variation brutale.

1.4.4 Enjeux de l'intégration Blockchain–IoT dans la logistique

L'intégration de la blockchain dans les réseaux IoT nécessite des consensus légers et sécurisés :

- Faibles en consommation,
- Rapides en validation (temps réel),
- Résistants aux attaques ou déconnexions partielles,
- Faciles à déployer sur des objets peu puissants

Conclusion

En résumé, à l'issue de ce premier chapitre, nous avons construit une approche cadrée des concepts, des fondements et des technologies qui composent le corpus nécessaire à l'analyse de notre sujet de recherche, dans le cadre d'une étude croisée entre la blockchain et l'écosystème de l'Internet des Objets (IoT) dans le secteur logistique, où les algorithmes de consensus occupent une place centrale dans l'établissement de la sécurité et de la confiance des échanges distribués. Il devient impératif de comprendre finement les mécanismes de décision au sein des réseaux distribués. C'est pourquoi le chapitre suivant sera consacré à une analyse comparative approfondie des algorithmes de consensus existants. Cette étude nous permettra d'identifier les forces et les faiblesses des solutions actuelles afin de jeter les bases techniques nécessaires à la conception de notre propre solution, mieux adaptée aux exigences de la logistique connectée.

Chapitre 2

Analyse et proposition d'un système de consensus adapté aux réseaux IoT logistiques

Introduction

Ce chapitre présente une vue d'ensemble et approfondi sur les solutions de consensus existantes relatives aux réseaux IoT, domaine logistique plus spécifiquement. L'objectif est d'identifier les approches, outils, protocoles et méthodes développés dans la littérature, ainsi que leurs limites, afin de mieux positionner la solution proposée dans ce mémoire. Cette revue repose sur une analyse bibliométrique systématique des publications scientifiques pertinentes, complétée par une lecture critique de travaux connexes. L'ensemble de cette analyse permet de dégager les grandes tendances de recherche, d'identifier les lacunes encore présentes dans le domaine, et de justifier les choix méthodologiques adoptés dans la suite de ce travail.

2.1 Analyse des algorithmes de consensus existants

2.1.1 Méthodologie de l'analyse bibliographique

Notre approche méthodologique s'appuie sur une analyse systématique de la littérature scientifique. Afin de garantir la pertinence et la qualité des sources retenues, nous avons exploité plusieurs bases de données de référence, notamment *Scopus*, *IEEE Xplore* et *Google Scholar*. Des outils de cartographie documentaire, tels que *Litmaps*, ont également été utilisés afin d'identifier les articles séminal et les travaux les plus influents du domaine.

Une chaîne de recherche spécifique, fondée sur l'utilisation d'opérateurs booléens, a été définie comme suit : *"(Blockchain OR Distributed Ledger) AND (Consensus Algorithm) AND (IoT OR Internet of Things) AND (Logistics OR Supply Chain)"*.

Les travaux identifiés ont ensuite été filtrés selon trois critères rigoureux :

- **Pertinence thématique** : focalisation sur les approches visant l'optimisation des ressources

dans les réseaux contraints.

- **Récence** : priorité accordée aux publications des cinq dernières années (2019–2024), afin de couvrir les évolutions récentes des protocoles de consensus.
- **Qualité académique** : sélection d'articles publiés dans des revues scientifiques et des conférences internationales à comité de lecture.

Cette démarche a permis de constituer un corpus solide d'une quarantaine d'articles, servant de base à l'analyse des limites des systèmes existants et à la conception de la solution proposée.

2.1.2 Principaux résultats de l'analyse bibliométrique

2.1.2.1 Intégration de la blockchain dans l'IoT : vers une convergence technologique sécurisée et décentralisée

L'intégration de la technologie blockchain dans les réseaux IoT constitue une évolution technologique majeure, visant à renforcer la sécurité, la traçabilité, et la décentralisation dans des environnements distribués et contraints. Cette synergie est particulièrement pertinente dans des secteurs comme la logistique, où la transparence des données, la résilience du système et la vérifiabilité des transactions sont essentielles.

Une revue scientifique récente met en évidence une vision globale de cette convergence en proposant une revue systématique des principales applications IoT basées sur la blockchain. Leur étude met en lumière les architectures techniques adoptées, les protocoles de consensus utilisés, ainsi que les modèles de trafic réseau dans ces systèmes hybrides. Ils soulignent l'importance de concevoir des consensus adaptés à la faible puissance des dispositifs IoT, tout en assurant la fluidité des communications dans des réseaux pair-à-pair. Bien que leur travail reste théorique, il fournit un socle structurant pour toute recherche future orientée vers la performance opérationnelle [12].

Dans cette même logique, certains scientifiques se concentrent spécifiquement sur les algorithmes de consensus, cœur de la technologie blockchain. Leur objectif est de classifier ces mécanismes selon un modèle unifié, applicable aux blockchains de type chaîne linéaire ou DAG (graphe acyclique orienté). En proposant un cadre d'évaluation rigoureux, ils offrent des critères pratiques pour sélectionner un consensus en fonction de son domaine d'application. Néanmoins, leur approche reste générique et ne cible pas spécifiquement les contraintes propres aux réseaux IoT (énergie, latence, mémoire, etc.) [8].

En outre, des chercheurs ont proposé une architecture fonctionnelle intégrant la blockchain pour renforcer la supervision, la sécurité des données et la traçabilité des opérations : systèmes logistiques intelligents . En introduisant des mécanismes comme la multi-authentification, les contrats intelligents ou encore des modèles d'interaction entre blockchains, leur proposition vise à automatiser les décisions logistiques tout en sécurisant les échanges. Malgré l'absence d'expérimentation réelle, leur approche apporte une valeur méthodologique significative à la recherche appliquée en supply chain intelligente [9] [16].

La question du choix de consensus dans un contexte contraint est approfondie par une étude qui analyse de manière critique les performances des algorithmes de consensus existants selon quatre grandes dimensions : le débit (throughput), la rentabilité (mining profit), le degré de décentralisation, et les vulnérabilités potentielles. En classant les consensus selon des critères souvent contradictoires, ils offrent un outil d'aide à la décision utile pour tout projet d'intégration blockchain. Toutefois, leur étude n'est pas centrée sur l'IoT, ce qui limite partiellement sa portée dans ce cadre [20].

Enfin, d'autres abordent de manière spécifique les défis liés à la distribution du consensus dans les systèmes IoT. Ils démontrent que les mécanismes classiques (Proof of Work, Proof of Stake) sont inadaptés aux environnements IoT en raison de leur forte consommation énergétique et de leur latence élevée. À l'inverse, ils recommandent les consensus basés sur les DAG, tels que Tangle et Hashgraph, mieux adaptés à l'IoT en termes de scalabilité et de vitesse. Ils mettent également en garde contre de nouveaux défis tels que la synchronisation, la détection des attaques ou la résilience aux déconnexions temporaires. Leur contribution est essentielle pour orienter les futures recherches vers des mécanismes plus légers et flexibles [3].

En somme, ces travaux convergent sur plusieurs constats fondamentaux : la blockchain représente une solution prometteuse pour renforcer la sécurité et la confiance dans l'IoT ; les consensus doivent être repensés pour s'adapter aux contraintes techniques des objets connectés ; et enfin, les cas d'usage logistiques constituent un terrain d'application privilégié. Toutefois, les approches sont encore majoritairement conceptuelles, ce qui appelle des validations empiriques à travers des prototypes ou des simulations réelles. Voici un tableau récapitulatif des travaux :

Objectif	**Méthodologie**	**Résultats majeurs**	**Limites**
Offrir une vue d'ensemble des architectures, consensus et modèles de trafic dans les systèmes IoT–Blockchain	Revue systématique de la littérature sur les systèmes IoT intégrant la blockchain	Identification des principaux modèles architecturaux, protocoles et schémas de trafic adaptés à l'IoT–Blockchain	Étude théorique sans validation quantitative
Classifier et modéliser les algorithmes de consensus selon leurs phases de conception	Construction d'un modèle de processus unifié et évaluation multicritère des consensus	Cadre d'analyse facilitant la sélection des consensus selon les cas d'usage	Contraintes IoT (énergie, calcul, latence) peu considérées
Proposer une architecture logistique intelligente sécurisée par la blockchain	Modélisation fonctionnelle et schémas d'interaction blockchain pour la logistique	Intégration de la traçabilité, supervision et sécurité via smart contracts et multi-authentification	Absence de prototype et de validation terrain
Évaluer et classer les consensus selon des critères de performance	Revue analytique et comparaison par paires sur quatre dimensions : débit, profit, décentralisation, sécurité	Classement structuré des consensus selon leurs domaines d'application	Contraintes IoT non explicitement intégrées
Identifier les défis spécifiques de l'intégration consensus–blockchain dans l'IoT	Analyse des consensus classiques (PoW, PoS) et des approches DAG (Tangle, Hashgraph)	Les approches DAG apparaissent plus adaptées aux contraintes IoT (scalabilité, faible consommation)	Besoin de mécanismes plus légers et défis de synchronisation et résilience

2.1.2.2 Blockchain et IoT dans la logistique : vers des consensus adaptés aux environnements contraints

Une proposition récente introduit l'algorithme Proof of Block & Trade (PoBT) dans une optique d'adaptation de la blockchain aux environnements IoT industriels. L'objectif est de fournir un algorithme de consensus léger capable de valider à la fois les blocs et les échanges (transactions) avec des exigences minimales en calcul et mémoire. En plus du PoBT, un mécanisme de distribution du registre est proposé pour alléger encore davantage la charge sur les objets IoT. Les expérimentations montrent des améliorations significatives en matière de consommation, bande passante et sécurité, rendant ce modèle hautement adapté aux systèmes logistiques embarqués [2].

Dans le domaine de la traçabilité alimentaire, des auteurs proposent un système BIFTS (Blockchain–IoT–Food Traceability System) centré sur la gestion de la chaîne du froid et la traçabilité alimentaire. En intégrant la blockchain, l'IoT et la logistique, ils conçoivent un mécanisme de consensus spécifique prenant en compte la durée de transit, la qualité des produits, et la quantité transportée. Le système est pensé pour être léger tout en maintenant un haut niveau de fiabilité, avec un flux de données ajusté à la granularité des objets connectés. L'article montre comment la gestion de la péremption peut être optimisée grâce à la blockchain [23].

Une revue exhaustive consacrée à la logistique intelligente basée sur l'IoT aborde les domaines du transport intelligent, de l'entreposage et de la livraison. L'analyse bibliométrique couvre la période 2008–2019. Il y est listé les défis technologiques clés de l'IoT dans la logistique : interopérabilité, standardisation, traitement de données et problèmes de sécurité. Aucune solution directe via la blockchain n'est proposée, mais les contraintes énoncées justifient l'exploration de consensus allégés pour des systèmes résilients [5].

Une autre analyse des applications concrètes de l'IoT en logistique souligne les apports potentiels en termes de visibilité, traçabilité et décision intelligente. Toutefois, des barrières à l'adoption subsistent : coûts d'équipement, absence de normes et manque de fiabilité de certains dispositifs connectés. Là encore, la blockchain est mentionnée en perspective comme réponse possible à ces défis, à condition d'adapter ses mécanismes fondamentaux (notamment le consensus) [10].

Enfin, un consensus probabiliste et allégé PoEWAL (Proof of Elapsed Work and Luck), fondé sur une combinaison d'attente aléatoire et de seuil de travail prouvé, a été proposé pour les environnements IoT décentralisés. L'algorithme est basé sur l'attente aléatoire combinée à un seuil de travail prouvé, et ne nécessite ni ressources intensives ni entité de confiance. Les simulations (avec Contiki Cooja) montrent que PoEWAL est faiblement énergivore, rapide, et robuste dans des réseaux de capteurs contraints, ce qui le rend prometteur pour les applications logistiques [18].

En somme, l'analyse de la littérature souligne un fossé persistant entre les exigences de sécurité de la blockchain et les capacités réelles des objets connectés. Néanmoins, des protocoles comme le PoEWAL (Proof of Elapsed Work and Luck) ont déjà ouvert la voie avec des performances prometteuses, notamment une réduction significative de la consommation énergétique et de la latence par rapport au PoW classique, mais leur application brute à la logistique reste toujours limitée. C'est sur la base de ces acquis technologiques que nous proposons d'évoluer vers PoEWAL-LT, en y intégrant des mécanismes de validation contextuels (GPS, capteurs) pour répondre aux défis spécifiques de la traçabilité. Le tableau suivant est récapitulatif :

Objectif	Méthodologie	Résultats principaux	Limites / Défis
Proposer PoBT (Proof of Block & Trade) comme consensus léger pour blockchain IoT	Modélisation et évaluation computationnelle	Réduction du temps de validation et des ressources	Non testé en conditions industrielles
Intégrer blockchain, IoT et logique floue pour la traçabilité alimentaire (BIFTS)	Architecture BIFTS et mécanisme de consensus embarqué	Adaptation au cycle de vie des aliments et amélioration de la précision	Complexité du déploiement multi-acteurs
Fournir une revue des applications IoT en logistique intelligente	Analyse bibliométrique et sémantique	Identification des technologies émergentes et des défis majeurs	Absence de solution opérationnelle
Analyser les cas d'usage de l'IoT dans la logistique	Étude de littérature et discussion critique	Valorisation de la donnée logistique en temps réel	Coûts, sécurité et manque de standards
Proposer PoEWAL comme consensus léger adapté à l'IoT	Simulation (Cooja) et comparaison à PoW, PoS, etc.	Faible consommation énergétique et latence réduite	Testé uniquement en simulation

2.2 Les enjeux du consensus dans les systèmes distribués

Dans les systèmes distribués, la problématique du consensus désigne la nécessité pour un ensemble de nœuds de parvenir à un accord commun, même en présence de pannes, d'erreurs de communication ou de comportements malveillants. Cette capacité à tolérer les fautes tout en garantissant l'intégrité du système est essentielle dans les architectures décentralisées comme la blockchain.

2.2.1 Le problème des généraux byzantins

Le problème des généraux byzantins, formulé initialement par Lamport et al., illustre de manière abstraite les défis liés à la fiabilité des communications dans un environnement où certains participants peuvent être défaillants ou malintentionnés. Dans ce scénario, plusieurs généraux doivent coordonner une attaque contre une forteresse. Ils ne peuvent communiquer que par messagers, et certains peuvent tenter de tromper les autres. Le succès de la mission dépend de leur capacité à s'accorder sur une stratégie commune, malgré ces incertitudes. Ce problème sert de base à la conception d'algorithmes de consensus tolérants aux fautes byzantines, en modélisant les défaillances possibles dans un réseau distribué [7].

2.2.2 Tolérance aux fautes byzantines (BFT)

La tolérance aux fautes byzantines (Byzantine Fault Tolerance – BFT) est une approche visant à assurer qu'un système distribué puisse continuer à fonctionner correctement, même si certains nœuds se comportent de manière incorrecte ou malveillante. Pour garantir à la fois la sécurité (prévenir les décisions erronées) et la vivacité (assurer la progression du système), BFT suppose que le tiers des nœuds au maximum peut être compromis.

Mathématiquement, un système peut tolérer jusqu'à :

$$\left\lfloor \frac{n-1}{3} \right\rfloor$$

nœuds défaillants sur un total de n répliques. Pour que le système reste fonctionnel malgré f répliques potentiellement défaillantes, il est donc nécessaire de disposer d'au moins :

$$3f + 1$$

répliques. Certaines implémentations optimisées permettent toutefois de réduire ce nombre à :

$$2f + 1$$

tout en conservant les garanties de sécurité et de disponibilité du système.

2.2.3 Tolérance aux fautes byzantines déléguée (dBFT)

Le mécanisme de consensus délégué BFT (delegated Byzantine Fault Tolerance – dBFT) constitue une évolution du modèle BFT classique. Ce mécanisme a notamment été adopté par la blockchain NEO. L'originalité de dBFT repose sur la séparation des rôles entre deux types de nœuds : les nœuds ordinaires et les nœuds comptables (ou "bookkeepers"). Contrairement au BFT traditionnel, les nœuds ordinaires ne participent pas directement au consensus, mais délèguent cette responsabilité en votant pour des nœuds comptables qu'ils jugent fiables.

Le processus de consensus démarre avec la sélection aléatoire d'un nœud comptable, qui propose un bloc de transactions à l'ensemble du réseau. Si au moins **66 %** des autres nœuds comptables valident la proposition, le bloc est officiellement ajouté à la blockchain. Ensuite, un nouveau nœud est sélectionné pour initier le tour de consensus suivant. Ce modèle vise à accroître l'efficacité du consensus, en limitant le nombre de participants actifs tout en maintenant un niveau élevé de confiance via la délégation.

2.2.4 Le problème de la double dépense

Dans le contexte des systèmes financiers distribués, la double dépense survient lorsque la même monnaie numérique est utilisée deux fois dans des transactions différentes. Ce problème est atténué dans la blockchain par le délai de traitement des transactions et la vérification par de nombreux nœuds.

2.2.5 Défis spécifiques aux environnements distribués

- **Contraintes de ressources :** Les systèmes distribués, particulièrement dans l'IoT, font face à des contraintes de ressources sévères incluant un traitement limité, une mémoire restreinte, une énergie limitée et une portée de communication réduite. Ces limitations rendent les algorithmes de consensus traditionnels inadaptés.

- **Topologies dynamiques :** Les topologies dynamiques constituent un défi majeur, avec des changements dus aux reconfigurations, pannes ou modes actif/inactif des nœuds. Cette instabilité peut compromettre la convergence des algorithmes de consensus traditionnels.

- **Charge de communication :** La charge de communication excessive dégrade les performances et peut entraîner une congestion du réseau ainsi qu'une consommation d'énergie accrue. Les schémas centralisés peuvent créer des points de défaillance uniques particulièrement problématiques.

2.2.6 Approche innovante : consensus basés sur les graphes orientés acycliques (DAG)

Les mécanismes DAG, représentant une "Blockchain 3.0", permettent aux utilisateurs d'insérer des transactions à tout moment en traitant les transactions précédentes. Des protocoles comme Tangle (IOTA) et Hashgraph offrent :

- Un débit de transactions illimité ou très élevé
- Une faible consommation de ressources
- Des frais de transaction nuls ou très bas

Ils visent à résoudre les limites de scalabilité, de rapidité et de coûts des blockchains traditionnelles (comme Bitcoin ou Ethereum). Plusieurs transactions peuvent être validées en parallèle contrairement à la Blockchain classique où c'est une chaîne linéaire de blocs (un seul bloc à la fois).

2.3 Solution proposée

Notre travail propose Proof of Elapsed Work and Luck for Logistics Traceability (PoEWAL-LT), une optimisation du mécanisme de consensus PoEWAL qui adopte une approche probabiliste permettant à des dispositifs aux ressources limitées de participer au processus de sécurisation du réseau. Son principe de fonctionnement repose sur la résolution partielle d'un casse-tête cryptographique au cours d'un intervalle de temps fixe, combinée à un mécanisme de chance destiné à départager les égalités entre nœuds concurrents et à limiter les bifurcations de la chaîne.

Notre contribution repose sur l'introduction d'un work score adaptatif orienté logistique, intégrant la criticité des événements, la fiabilité des nœuds et les contraintes énergétiques, ainsi que sur une validation hiérarchique par type d'entité logistique. Le modèle intègre également un mécanisme de détection proactive d'anomalies multicritère et des optimisations adaptées aux environnements IoT contraints. L'ensemble est validé expérimentalement sous Contiki-Cooja, à travers une évaluation comparative à partir des résultats et une analyse graduée de la résilience byzantine.

2.3.1 Architecture du système

Le système proposé repose sur un algorithme de consensus PoEWAL-LT optimisé pour les réseaux IoT logistiques contraints, déployé sur des nœuds Sky Mote (CPU 8 MHz, 48 KB de mémoire) fonctionnant sous Contiki OS 3.0. Le réseau est composé de 10 nœuds IoT représentant les entités logistiques (conteneurs, colis, marchandises) équipés de capteurs GPS et de température. L'algorithme intègre un mécanisme de Work Score et Luck Factor adapté aux ressources limitées, ainsi qu'un système de détection byzantine capable d'identifier et d'isoler les nœuds malveillants.

Son utilité est d'améliorer la fiabilité et l'efficacité du consensus distribué dans les environnements logistiques IoT en tenant compte des contraintes spécifiques du secteur : faible puissance de

calcul, bande passante limitée, mobilité des nœuds et tolérance aux pannes. Le système PoEWAL-LT permet de maintenir un consensus robuste même avec jusqu'à 33% de nœuds byzantins, tout en réduisant la latence et la consommation énergétique comparé aux algorithmes de consensus traditionnels, le rendant ainsi adapté aux réseaux IoT logistiques à ressources limitées.

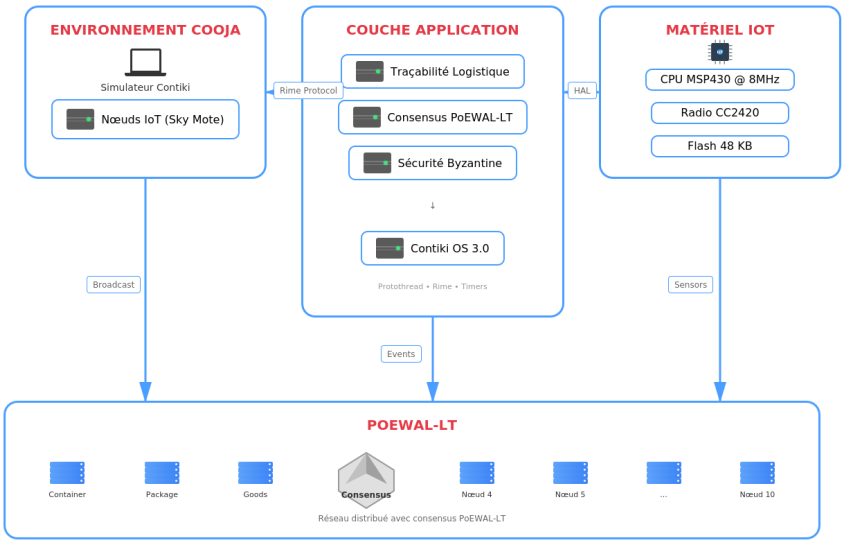


FIGURE 2.1 : Architecture du système

2.3.2 Technologies utilisées

Plusieurs sont les outils et technologies utilisés qui ont activement contribué à la mise en place de notre simulation. Ces outils se distinguent selon les langages de programmation, les bibliothèques logicielles, le système, ainsi que l'environnement de développement retenu. Cette sous-section présente les principaux choix techniques opérés dans le cadre de ce travail.

2.3.2.1 Contiki OS (release 3.0)

Contiki OS est un système d'exploitation open source destiné aux objets connectés et aux réseaux IoT contraints. Il fournit un environnement adapté à la simulation et au déploiement de protocoles réseau dans des conditions réalistes de ressources limitées. Cooja et Sky Motes (MSP430), tels sont respectivement le simulateur et la plateforme cible utilisés [26].

2.3.2.2 Langages de programmation

- Le langage C, il a été utilisé pour l'implémentation des composants embarqués au sein de Contiki OS. Il est largement employé dans le développement de systèmes embarqués en raison de son efficacité et de son contrôle bas niveau des ressources matérielles [36].
- Le langage Python, il a été utilisé pour l'analyse et le traitement des résultats issus des simulations. Il offre un écosystème riche pour l'analyse scientifique et la visualisation des données [34].

2.3.2.3 Bibliothèques Python

- NumPy, une bibliothèque fondamentale pour le calcul scientifique en Python, utilisée pour les opérations mathématiques et la manipulation efficace de tableaux multidimensionnels [32].
- Pandas, une bibliothèque Python dédiée à la manipulation et à l'analyse de données structurées. Elle a été utilisée pour l'organisation et l'analyse statistique des données expérimentales [33].
- Matplotlib, une bibliothèque Python de visualisation de données, utilisée pour produire des graphiques et représenter les résultats expérimentaux de manière claire et exploitable [31].

2.3.2.4 Ubuntu 20.04 LTS

Ubuntu 20.04 LTS est le système d'exploitation utilisé pour le développement, la simulation et l'analyse. Il a été retenu pour sa stabilité à long terme et sa compatibilité avec les outils IoT et scientifiques employés tel que contiki OS [35].

2.3.3 Algorithme de consensus PoEWAL-LT

Algorithme 1 : Ajustement dynamique de la difficulté

Data : C_{prev} (collisions précédentes), D (difficulté actuelle), $\mathcal{N}$ (ensemble de nœuds)
Result : T_{dynamic} (temps dynamique), D_{new} (nouvelle difficulté)

Calculer C_{prev} = nombre de collisions au bloc précédent;
if $C_{prev} > \theta_{collision}$ **then**
  $T_{\text{dynamic}} \leftarrow T_{\text{base}} \times 1.5$;
  $D_{\text{new}} \leftarrow \min(D + 1, D_{\text{max}})$;
else
  if $C_{prev} = 0$ **then**
    $T_{\text{dynamic}} \leftarrow T_{\text{base}} \times 0.8$;
    $D_{\text{new}} \leftarrow \max(D - 1, D_{\text{min}})$;
  else
    $T_{\text{dynamic}} \leftarrow T_{\text{base}}$;
    $D_{\text{new}} \leftarrow D$;
  end
end
Calculer $E_{\text{avg}} \leftarrow \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} E_{\text{residual}}(n)$;
if $E_{avg} < 30\%$ **then**
  $T_{\text{dynamic}} \leftarrow T_{\text{dynamic}} \times 0.7$;
end
$D \leftarrow D_{\text{new}}$;
return $T_{dynamic}$, D;

Pour commencer, le réseau observe le cycle précédent et ajuste deux paramètres à savoir : le temps alloué au calcul cryptographique (T_dynamic), la difficulté cryptographique (D), exprimée en nombre de zéros en tête du hash.

Les collisions de validateurs, l'absence totale de collisions et l'énergie moyenne restante des nœuds sont les trois signaux surveillés. Cette phase transforme le consensus en un système auto-

régulé, sensible à la fois à la congestion cryptographique et aux contraintes énergétiques des objets IoT.

Algorithme 2 : Calcul du Work Score Multicritère

Data : Ensemble de nœuds $\mathcal{N}$, Blacklist
Result : $\text{WS}_{\text{Final}}(n)$ pour chaque nœud n

for *chaque nœud* $n \in \mathcal{N} \setminus$ *Blacklist* **do**
  // Calcul des composantes
  $C_{\text{crit}}(n) \leftarrow \frac{\sum(\text{criticité}_{\text{événements}})}{|E_n|}$;
  $C_{\text{fiab}}(n) \leftarrow \frac{\text{events}_{\text{valid}}}{\text{events}_{\text{total}}}$;
  $C_{\text{up}}(n) \leftarrow \frac{\text{time}_{\text{active}}}{\text{time}_{\text{total}}}$;
  $C_{\text{éner}}(n) \leftarrow \frac{1}{\text{energy}_{\text{consumed}}+1}$;
  // Score pondéré long-terme
  $\text{WS}_{\text{LT}}(n) \leftarrow 0.40 \times C_{\text{crit}}(n) + 0.30 \times C_{\text{fiab}}(n)$;
  $+0.20 \times C_{\text{up}}(n) + 0.10 \times C_{\text{éner}}(n)$;
  // Ajustement selon la priorité du type
  $\text{WS}_{\text{Final}}(n) \leftarrow \text{WS}_{\text{LT}}(n) \times \left(1 + \frac{\text{Priority}(\text{type}_n)}{100}\right)$;
end
return *WS_{Final} pour tous les nœuds*;

Chaque nœud reçoit un Work Score (WS) basé sur quatre critères pondérés :

- Criticité logistique : Importance des événements gérés (retards critiques, incidents, livraisons sensibles).
- Fiabilité : Proportion d'événements valides par rapport aux événements déclarés.
- Disponibilité : Temps effectif de fonctionnement du nœud
- Sobriété énergétique : Moins un nœud consomme, plus il est valorisé.

Le protocole récompense la qualité opérationnelle observée dans la chaîne logistique. Deux approches complémentaires ont guidé le choix des pondérations du Work Score multicritère de PoEWAL-LT.

(1) Mesure de la criticité des produits : méthodologie ABC-XYZ-VED [3]

La classification ABC-XYZ-VED est une méthode standard de gestion logistique permettant d'évaluer la criticité des actifs selon trois dimensions complémentaires. Le tableau 2.3 présente la correspondance entre ces dimensions et les critères retenus dans PoEWAL-LT.

TABLE 2.3 : Justification des pondérations du Work Score par la classification ABC-XYZ-VED

Dimension	Description	Application dans PoEWAL-LT	Critère WS
ABC	Classification selon la valeur économique des actifs : les articles A ont la plus haute valeur et nécessitent un suivi prioritaire, tandis que les articles C ont une valeur moindre.	Traduit par le type d'entité logistique : un `Container` (classe A) porte davantage de valeur qu'un `Package` (classe B) ou des `Goods` génériques (classe C).	C_{crit} (40%)
XYZ	Classification selon la variabilité et la fréquence de la demande : X désigne une demande stable et prévisible, Z une demande très irrégulière nécessitant une surveillance accrue.	Traduit par la fréquence des événements logistiques générés par un nœud. Un nœud émettant régulièrement des événements valides (classe X) est valorisé par sa disponibilité et sa fiabilité, contribuant à un score WS plus stable.	C_{fiab} (30%) et C_{up} (20%)
VED	Classification selon la criticité opérationnelle : Vital (V) désigne les actifs dont l'absence compromet immédiatement les opérations, Essential (E) ceux causant des perturbations significatives, et Desirable (D) ceux à faible impact immédiat.	Traduit par l'état logistique de l'entité surveillée : `DAMAGED` (Vital) déclenche une criticité maximale, `IN_TRANSIT` (Essential) une criticité intermédiaire, et `STORED` (Desirable) une criticité standard.	C_{crit} (40%)

(2) Méthode Rank Sum

Elle s'inscrit dans l'approche SMART (*Simple Multi-Attribute Rating Technique*) et transforme un classement hiérarchique de critères en valeurs numériques exploitables mathématiquement. Le processus repose sur trois étapes : le décideur commence par ordonner les critères par importance décroissante, du rang 1 (le plus important) au rang n (le moins important) ; chaque critère reçoit ensuite une valeur inversement proportionnelle à son rang, soit $n - r_i + 1$ pour le critère de rang r_i ; enfin, le poids final w_i est obtenu par normalisation [6] :

$$w_i = \frac{n - r_i + 1}{\sum_{k=1}^{n} (n - r_k + 1)} \tag{2.1}$$

où r_i désigne le rang du critère i et n le nombre total de critères. Pour $n = 4$ critères, la somme des valeurs vaut $4 + 3 + 2 + 1 = 10$, ce qui produit les pondérations présentées dans le tableau 2.4.

TABLE 2.4 : Application de la méthode Rank Sum aux critères du Work Score

Critère	**Rang** r_i	**Valeur** $(n - r_i + 1)$	**Poids** w_i	**Interprétation**
C_{crit} — Criticité logistique	1	4	$\frac{4}{10} = 0{,}40$ (40%)	Impact métier direct
C_{fiab} — Fiabilité du nœud	2	3	$\frac{3}{10} = 0{,}30$ (30%)	Confiance dans les données
C_{up} — Disponibilité	3	2	$\frac{2}{10} = 0{,}20$ (20%)	Continuité du service
$C_{\text{éner}}$ — Sobriété énergie	4	1	$\frac{1}{10} = 0{,}10$ (10%)	Contrainte IoT
Total	–	**10**	**1,00 (100%)**	–

Algorithme 3 : Calcul du Proof of Work Factor

Data : Ensemble de nœuds $\mathcal{N}$, dernier bloc, T_{dynamic}, D (difficulté)
Result : PWF(n), $Z_{\max}(n)$, $r_{\text{best}}(n)$ pour chaque nœud

$c \leftarrow$ hash(dernier_bloc);
for *chaque nœud* $n \in \mathcal{N} \setminus$ *Blacklist* **do**
  $Z_{\max}(n) \leftarrow 0$;
  $r_{\text{best}}(n) \leftarrow 0$;
  $t_{\text{start}} \leftarrow$ timestamp_actuel();
  // Recherche du meilleur nonce pendant T_{dynamic}
  while *timestamp_actuel*$() - t_{start} < T_{dynamic}$ **do**
    $r \leftarrow$ random_nonce();
    $P \leftarrow$ SHA256$(c \| r \| n)$;
    $Z \leftarrow$ count_leading_zeros(P);
    if $Z > Z_{\max}(n)$ **then**
      $Z_{\max}(n) \leftarrow Z$;
      $r_{\text{best}}(n) \leftarrow r$;
    end
  end
  $T_{\text{used}}(n) \leftarrow$ timestamp_actuel$() - t_{\text{start}}$;
  // Calcul du PWF avec bonus
  if $Z_{\max}(n) \geq D$ **then**
    PWF$(n) \leftarrow \left(\frac{Z_{\max}(n)}{D}\right) \times 1.5$;
  else
    PWF$(n) \leftarrow \frac{Z_{\max}(n)}{D}$;
  end
  // Pénalité anti-triche
  if $T_{used}(n) < 0.8 \times T_{dynamic}$ **then**
    PWF$(n) \leftarrow$ PWF$(n) \times 0.5$;
  end
end
return *PWF,* $Z_{\max}$, r_{best} *pour tous les nœuds*;

Algorithme 4 : Calcul du Luck Factor

Data : Ensemble de nœuds $\mathcal{N}$, $r_{\text{best}}(n)$ pour chaque nœud, timestamp t
Result : LF(n, t) pour chaque nœud

for *chaque nœud* $n \in \mathcal{N} \setminus$ *Blacklist* **do**
  hash_input $\leftarrow n \| r_{\text{best}}(n) \| t$;
  $H \leftarrow$ DJB2(hash_input) `// Hash variant ×33`
  LF$(n, t) \leftarrow H \bmod 1000$;
end
return *LF pour tous les nœuds*;

Les deux algorithmes precedents constituent le cœur probabiliste du consensus. Le Proof of Work Factor (PWF), contrairement au PoW, le calcul est limité dans le temps (T_dynamic) et mésuré. Même avec mérite et travail, un consensus totalement déterministe favoriserait toujours les mêmes nœuds. Le Luck Factor introduit donc une entropie légère mais vérifiable, basée sur l'identité du nœud, son meilleur nonce et le temps. La chance n'annule pas le mérite, elle empêche simplement la monopolisation.

Algorithme 5 : Calcul du Score Final

Data : $\text{WS}_{\text{Final}}(n)$, PWF$(n)$, LF$(n, t)$ pour chaque nœud
Result : Score(n, t) pour chaque nœud

for *chaque nœud* $n \in \mathcal{N} \setminus$ *Blacklist* **do**
  Score$(n, t) \leftarrow \text{WS}_{\text{Final}}(n) \times \text{PWF}(n) \times \text{LF}(n, t)$;
end
return *Score pour tous les nœuds*;

Des premières phases, un nœud ne peut pas gagner uniquement par chance, ni uniquement par calcul, ni uniquement par réputation. Le score final est le produit de :

- WS : valeur intrinsèque du nœud,
- PWF : travail effectivement réalisé par le nœud,
- LF : facteur de variabilité probabiliste.

Algorithme 6 : Sélection Multi-Niveaux du Validateur

Data : Score(n, t), PWF(n) pour chaque nœud, Blacklist
Result : Validateur v sélectionné

`// Niveau 1 : Filtrage PWF`
$\text{PWF}_{\max} \leftarrow \max_{n \in \mathcal{N} \setminus \text{Blacklist}} \text{PWF}(n)$;
Candidates $\leftarrow \{n \in \mathcal{N} \setminus \text{Blacklist} \mid \text{PWF}(n) \geq 0.6 \times \text{PWF}_{\max}\}$;
`// Niveau 2 : Sélection probabiliste`
total_score $\leftarrow \sum_{i \in \text{Candidates}} \text{Score}(i, t)$;
for *chaque nœud* $n \in$ *Candidates* **do**
  $P(\text{validateur} = n) \leftarrow \frac{\text{Score}(n,t)}{\text{total_score}}$;
end
$v \leftarrow$ SelectWeightedRandom(P) `// Sélection probabiliste pondérée`
return v;

Algorithme 7 : Vérification Cryptographique du Validateur

Data : Validateur v, $r_{\text{best}}(v)$, $Z_{\max}(v)$, c, t
Result : Validation réussie ou retour à Phase 5

$P_{\text{verify}} \leftarrow \text{SHA256}(c \| r_{\text{best}}(v) \| v)$;
$Z_{\text{verify}} \leftarrow \text{count_leading_zeros}(P_{\text{verify}})$;
$\text{LF}_{\text{verify}} \leftarrow \text{DJB2}(v \| r_{\text{best}}(v) \| t) \bmod 1000$;
if $Z_{verify} = Z_{\max}(v)$ ***et*** $LF_{verify} = LF(v, t)$ **then**
    return *VÉRIFICATION RÉUSSIE*;
else
    Blacklist $\leftarrow$ Blacklist $\cup \{v\}$;
    return *RETOUR À PHASE 5*;
end

Algorithme 8 : Résolution de Collision à 3 Niveaux

Data : Ensemble Candidates, Score(n, t) pour chaque candidat
Result : Validateur unique v

$\text{Score}_{\max} \leftarrow \max_{n \in \text{Candidates}} \text{Score}(n, t)$;
$\text{Colliders} \leftarrow \{n \in \text{Candidates} \mid \text{Score}(n, t) \geq 0.95 \times \text{Score}_{\max}\}$;
if $|Colliders| > 1$ **then**
    `// Niveau 1 : Départage par Z_max`
    $v \leftarrow \arg\max_{n \in \text{Colliders}} Z_{\max}(n)$;
    if $|\arg\max_{n \in Colliders} Z_{\max}(n)| > 1$ **then**
        `// Niveau 2 : Départage par r_best (plus petit nonce)`
        $v \leftarrow \arg\min_{n \in \text{Colliders}} r_{\text{best}}(n)$;
    end
    if $|\arg\min_{n \in Colliders} r_{best}(n)| > 1$ **then**
        `// Niveau 3 : Départage par hash du bloc`
        $v \leftarrow \arg\min_{n \in \text{Colliders}} \text{SHA256}(\text{bloc}_n)$;
    end
end
return v;

Au niveau des trois algorithmes ci dessus, la sélection du validateur s'effectue en deux niveaux complémentaires :

- **Filtrage strict** : seuls les nœuds ayant effectivement accompli un travail mesurable et conforme aux règles du protocole sont considérés. Cette étape élimine toute tentative de participation passive ou frauduleuse.
- **Sélection probabiliste pondérée** : parmi les nœuds éligibles, la probabilité de sélection est proportionnelle au score obtenu (travail, temps, chance). Un score élevé augmente les chances de sélection sans pour autant garantir une victoire systématique, ce qui limite la centralisation et favorise l'équité.

Ensuite, une fois un validateur sélectionné, le réseau applique une vérification cryptographique systématique afin de s'assurer que le nœud :

- n'a pas falsifié les résultats de son calcul,

- n'a pas modifié son nonce après coup,
- n'a pas manipulé la composante de chance intégrée au protocole.

Tout échec à l'une de ces vérifications entraîne l'exclusion immédiate du nœud fautif du processus de validation. Dans PoEWAL-LT, la confiance n'est jamais implicite : elle est continuellement prouvée et vérifiée.

Dans le cas rare où plusieurs nœuds obtiennent des résultats quasi équivalents, le protocole applique une résolution hiérarchisée et déterministe fondée sur les critères suivants :

1. la qualité du travail cryptographique réalisé,
2. le caractère le plus déterministe du nonce,
3. la valeur minimale du hash final.

Cette stratégie permet d'éviter les bifurcations (*forks*) ainsi que les mécanismes de négociation coûteux, assurant ainsi la stabilité, la rapidité de consensus et la cohérence globale du registre distribué.

Algorithme 9 : Détection Byzantine Multicritère

Data : Ensemble de nœuds $\mathcal{N}$
Result : Blacklist mise à jour

for *chaque nœud* $n \in \mathcal{N}$ **do**
  Byzantine$(n) \leftarrow 0$;
  if $\Delta_{GPS}(n) > 500$ *km* **then**
    Byzantine$(n) \leftarrow 1$ `// Téléportation impossible`
  end
  if $T(n) \notin [-50C, 80C]$ **then**
    Byzantine$(n) \leftarrow 1$ `// Température hors bornes`
  end
  if *transition_état_invalide*(n) **then**
    Byzantine$(n) \leftarrow 1$ `// Ex: Livré → En transit`
  end
  if *Byzantine*$(n) = 1$ **then**
    Blacklist $\leftarrow$ Blacklist $\cup \{n\}$;
  end
end
return *Blacklist*;

Cette partie de PoEWAL-LT assure une validation sémantique des données physiques en complément des vérifications cryptographiques classiques. Elle vise à garantir que les informations issues du monde réel respectent les contraintes physiques et logistiques du système surveillé en se basant sur les paramètres suivants :

- $\Delta_{\text{GPS}}(n)$: variation géographique observée pour le nœud n,
- $T(n)$: température mesurée associée au nœud n,
- transition_état(n) : cohérence de la transition d'état dans le flux logistique du nœud n.

Un nœud est qualifié de byzantin lorsqu'il viole les lois du réel, notamment dans les cas suivants :

- déplacement géographiquement impossible,
- température irréaliste au regard du contexte physique,
- transition logistique incohérente ou non conforme à la séquence attendue.

Algorithme 10 : Consensus BFT (Byzantine Fault Tolerance)

Data : Validateur v, bloc proposé, ensemble de nœuds $\mathcal{N}$
Result : APPROVE ou REJECT

```
Diffuser le bloc proposé par le validateur v à tous les nœuds;
Collecter les votes des nœuds;
θ_BFT ← ⌊2|𝒩|/3 + 1⌋;
if votes_approve ≥ θ_BFT then
    return v, APPROVE, bloc validé;
else
    return v, REJECT, bloc rejeté;
end
```

Enfin, le bloc proposé est soumis à un mécanisme de vote byzantin classique. Le consensus est atteint lorsque une majorité qualifiée de nœuds honnêtes approuve le bloc, selon la condition suivante :

$$\text{votes}_{\text{approve}} \geq \theta_{\text{BFT}} = \left\lfloor \frac{2}{3}|N| \right\rfloor + 1$$

où :

- $|N|$ représente le nombre total de nœuds participants,
- θ_{BFT} désigne le seuil de consensus byzantin,
- $\text{votes}_{\text{approve}}$ correspond au nombre de votes favorables.

Ce mécanisme garantit à la fois la finalité du bloc validé et la tolérance aux fautes byzantines du système.

Paramètres du Modèle

$D_{\text{init}} = 4$	Difficulté initiale (zéros MSB)
$D_{\min} = 2$	Difficulté minimum
$D_{\max} = 8$	Difficulté maximum
$T_{\text{base}} = 5$ s	Temps de base PWF
$\theta_{\text{collision}} = 3$	Seuil de collision pour ajuster D
$\theta_{\text{PWF}} = 60\%$	Seuil PWF minimum pour candidature
$\theta_{\text{BFT}} = 67\%$	Quorum byzantin (2/3)
$\theta_{\text{energy}} = 30\%$	Seuil énergie basse

Formules Clés

Score Final (3 composantes)

$$\text{Score}(n,t) = \underbrace{\text{WS}_{\text{Final}}(n)}_{\text{Mérite}} \times \underbrace{\text{PWF}(n)}_{\text{Travail}} \times \underbrace{\text{LF}(n,t)}_{\text{Chance}} \tag{2.2}$$

Work Score Long-Term

$$\text{WS}_{\text{LT}}(n) = \sum_{i \in \{\text{crit,fiab,up,éner}\}} w_i \times C_i(n) \tag{2.3}$$

Proof of Work Factor

$$\text{PWF}(n) = \begin{cases} \left(\frac{Z_{\max}(n)}{D}\right) \times 1.5 & \text{si } Z_{\max}(n) \geq D \\ \frac{Z_{\max}(n)}{D} & \text{sinon} \end{cases} \tag{2.4}$$

Luck Factor

$$\text{LF}(n,t) = \text{DJB2}(n \| r_{\text{best}}(n) \| t) \bmod 1000 \tag{2.5}$$

Conclusion

Ce deuxième chapitre a permis de faire une analyse comparative entre les mécanismes de consensus existants pour mettre en lumière leur inadéquation face au double défi de l'IoT et de la logistique : les uns trop énergivores (PoW), les autres trop centralisés ou bavards en messages (PBFT).

En réponse à ces limites, nous avons formulé l'architecture de PoEWAL-LT. Notre proposition théorique repose sur une innovation majeure : le remplacement de la puissance de calcul brute par un « Work Score » contextuel, valorisant le comportement honnête du nœud (fiabilité, disponibilité) et la cohérence physique des données (GPS, température). Nous avons ainsi défini un protocole qui ne se contente pas de valider mathématiquement un bloc, mais qui vérifie la plausibilité logistique des transactions.

Cette modélisation théorique étant posée, il convient désormais de confronter PoEWAL-LT à la réalité opérationnelle. Le chapitre suivant sera consacré à l'implémentation de ce système dans l'environnement de simulation Cooja, afin de quantifier ses performances réelles en termes de latence, de consommation et de résilience face aux attaques.

Chapitre 3

Résultats et discussions

Introduction

Ce chapitre présente les résultats expérimentaux obtenus lors de l'évaluation de l'algorithme de consensus PoEWAL-LT (Proof of Elapsed Work and Luck for Logistics Traceability) dans un environnement de simulation IoT. Les expérimentations ont été conduites sur le simulateur Cooja/Contiki, permettant une évaluation réaliste des performances de l'algorithme de consensus dans un contexte de traçabilité logistique.

3.1 Environnement expérimental

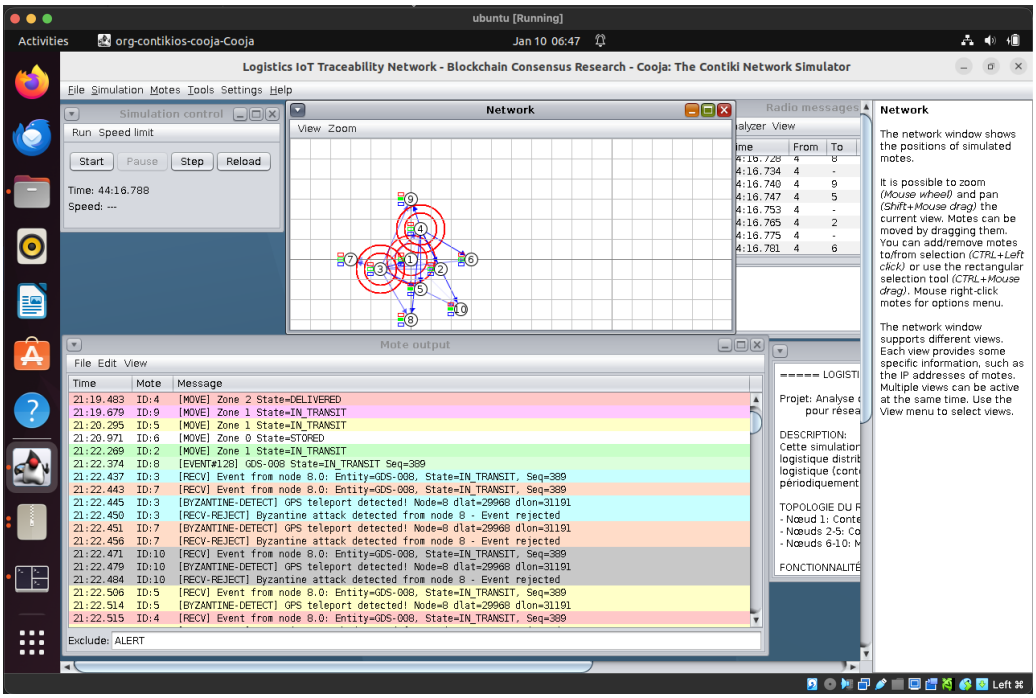


FIGURE 3.1 : Environnement de simulation

Dans une simulation nous avons plusieurs fenêtres. Nous décrivons ici brièvement les fonction-

nalités de quelques fenêtre :

- Network (Réseau) : En haut à droite de l'écran, affiche l'emplacement de chaque nœud du réseau. Permet de visualiser l'état de chaque nœud, y compris les LEDs, les identificateurs de mote, les adresses, les sorties lof, etc. Au départ, cette fenêtre est vide et nous devons la remplir avec nos capteurs.
- Simulation Control (Contrôle de simulation) : En haut sur le côté gauche de l'écran, ce panneau est utilisé pour démarrer, mettre en pause, recharger ou exécuter les étapes de la simulation. Il indique le temps d'exécution et la vitesse de la simulation. Cela signifie que nous pouvons exécuter les événements plusieurs fois plus rapidement qu'en temps réel.
- Mote output (Sortie Mote) : En bas de l'écran affiche toutes les sorties de l'interface série des nœuds. C'est la journalisation de tout ce qui se passe dans la simulation

3.1.1 Configuration de la simulation

TABLE 3.1 : Paramètres de configuration de la simulation

Paramètre	**Valeur**
Simulateur	Cooja
Nombre de nœuds	10
Type de mote	Sky (MSP430)
Protocole radio	IEEE 802.15.4
Durée simulation	15 min 36 s
Fenêtre consensus	30 secondes
Quorum requis	7/10 (70%)
Paramètres PoEWAL-LT	
Difficulté initiale (D)	4 zéros MSB
Temps PWF de base (T_base)	5 secondes
Seuil PWF minimum	60% du max
Bonus difficulté atteinte	1.5×

3.1.2 Scénarios de test

Quatre scénarios ont été définis pour évaluer la résilience du protocole face aux comportements byzantins :

- **Baseline (0%)** : Tous les nœuds sont honnêtes
- **Faible (10%)** : 1 nœud byzantin sur 10
- **Modéré (20%)** : 2 nœuds byzantins sur 10
- **Critique (33%)** : 3 nœuds byzantins sur 10 (seuil théorique BFT)

Note : Les résultats présentés dans ce chapitre correspondent au **scénario Modéré (20%)** avec 2 nœuds byzantins sur 10, permettant d'évaluer le comportement du protocole dans des conditions réalistes d'attaque.

Les types d'attaques byzantines définies incluent :

- Falsification GPS (téléportation anormale de marchandises)
- Transitions d'état illogiques (ex : "Livré" → "En transit")
- Falsification d'état logistique (masquage de dommages)
- Anomalies de température (valeurs hors plages physiques)[1]

3.2 Analyse des resultats

3.2.1 Comparaison du débit (TPS) entre PoEWAL-LT et quelques algorithmes

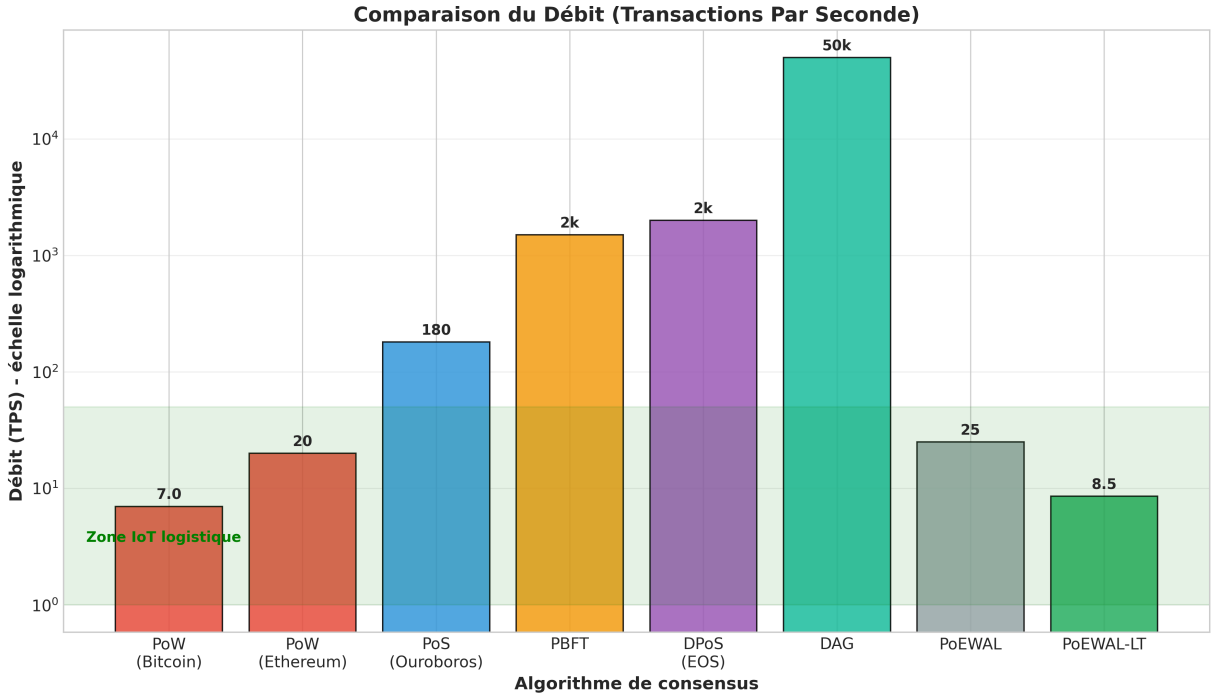


FIGURE 3.2 : Comparaison du débit (Transactions Par Seconde)

La figure 3.2 compare le débit en transactions par seconde (TPS) de PoEWAL-LT avec les algorithmes de consensus de référence issus de la littérature scientifique.

3.2.1.1 Éléments du graphique

- **Échelle logarithmique (axe Y)** : Nécessaire pour visualiser des valeurs allant de 7 TPS (Bitcoin) à 50 000 TPS (DAG). L'échelle linéaire rendrait les petites valeurs invisibles.
- **Zone verte "Zone IoT logistique"** : Délimite la plage de débit typique pour les applications de traçabilité IoT (3-50 TPS). Cette zone contextualise les besoins réels du domaine.

[1] Ce critère n'a pas été sollicité dans le scénario testé, les valeurs de température générées étant restées dans les plages valides.

3.2.1.2 Interprétation

Positionnement de PoEWAL-LT :

TABLE 3.2 : Comparaison du débit (TPS) par algorithme

Algorithme	TPS	Commentaire
PoW (Bitcoin)	7 [5]	Limité par le temps de bloc (10 min)
PoW (Ethereum)	20 [1]	Blocs plus rapides (15 s)
PoS (Ouroboros)	180 [13]	Slot time $\sim$ 20 s
PBFT	1 500 [20]	Haute performance mais complexité $O(n^2)$
DPoS (EOS)	2 000 [11]	Délégation centralisée
DAG	50 000 [1]	Architecture différente
PoEWAL	25 [18]	Version originale
PoEWAL-LT	**8,5**	**Mesuré (Cooja)**

Analyse critique :

Le TPS de 8,5 ev/s mesuré pour PoEWAL-LT peut sembler modeste en comparaison des blockchains grand public. Cependant, cette valeur est **parfaitement cohérente avec le contexte d'application** :

1. **Besoins réels de la logistique** : Un entrepôt traitant 10 000 colis/jour génère environ 0,1 événement/seconde. Le débit de 8,5 TPS permet de traiter 85$\times$ ce volume.
2. **Priorité à la sécurité** : Contrairement aux blockchains financières où le débit est critique, la traçabilité logistique privilégie la **fiabilité** et la **détection des fraudes**.
3. **Contraintes IoT** : Un débit plus élevé nécessiterait plus d'énergie et de bande passante, incompatibles avec les capteurs sur batterie.

3.2.1.3 Conclusion sur le TPS

PoEWAL-LT se positionne dans la **zone IoT logistique** (3-50 TPS), confirmant son adéquation avec les besoins du domaine. Le compromis accepté entre débit et sécurité est justifié par les exigences spécifiques de la traçabilité.

3.2.2 Comparaison de la latence entre PoEWAL-LT et quelques algorithmes

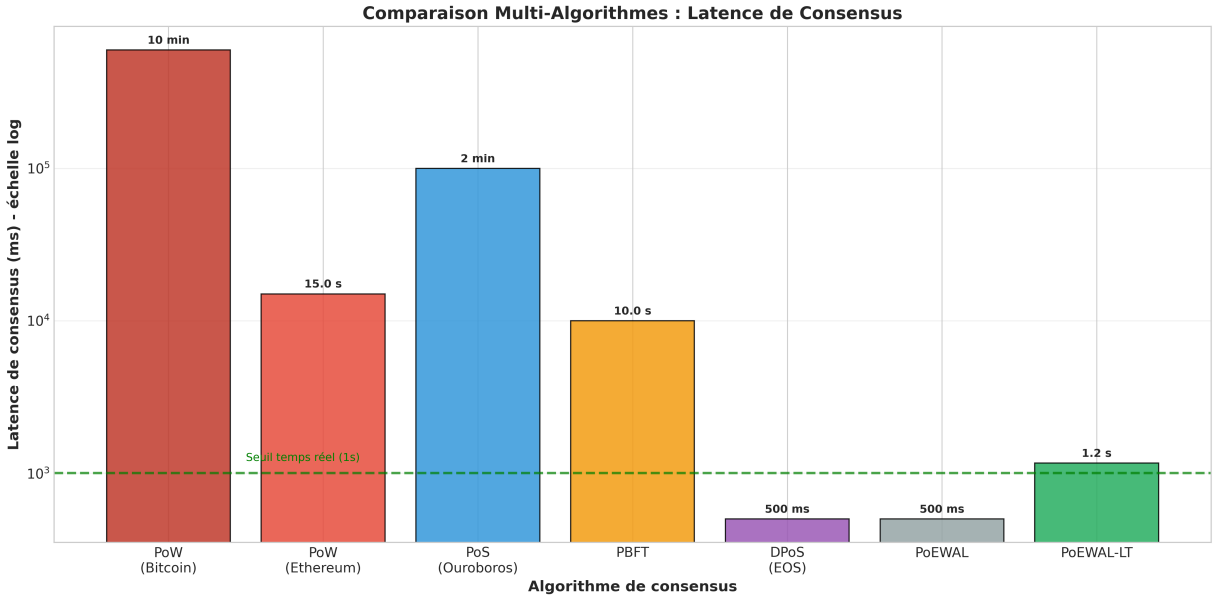


FIGURE 3.3 : Comparaison multi-algorithmes : latence de consensus (échelle logarithmique)

La figure 3.3 présente une comparaison visuelle des latences de consensus entre PoEWAL-LT et les principaux algorithmes existants, sous forme d'histogramme en échelle logarithmique.

3.2.2.1 Éléments du graphique

- **Barres colorées** : Chaque barre représente un algorithme de consensus, avec sa latence en millisecondes sur l'axe Y (échelle logarithmique).
- **Ligne verte pointillée "seuil temps réel (1s)"** : Représente le seuil de 1 000 ms en dessous duquel un système est considéré comme temps réel.
- **Annotations temporelles** : Au-dessus de chaque barre, la latence est exprimée en unités lisibles (10 min, 15 s, 2 min, etc.).

3.2.2.2 Interprétation

TABLE 3.3 : Latence de consensus par algorithme

Algorithme	Latence	Position vs seuil 1 s
PoW (Bitcoin)	10 min [11]	600× au-dessus
PoW (Ethereum)	15 s [13]	15× au-dessus
PoS (Ouroboros)	2 min [20]	120× au-dessus
PBFT	10 s [20]	10× au-dessus
DPoS (EOS)	500 ms [11]	Sous le seuil
PoEWAL	500 ms [18]	Sous le seuil
PoEWAL-LT	**1,2 s**	**Légèrement au-dessus du seuil**

1. **PoW (Bitcoin/Ethereum)** : Les algorithmes de Proof of Work dominent le graphique avec des latences de plusieurs minutes. Ils sont inadaptés aux applications IoT nécessitant des réponses rapides.

2. **PoS (Ouroboros)** : Malgré l'absence de mining, le slot time de 20 secondes par epoch génère une latence de 2 minutes, encore trop élevée pour le temps réel.

3. **PBFT** : Avec 10 secondes, PBFT est plus rapide mais sa complexité $O(n^2)$ en messages le pénalise pour les grands réseaux.

4. **DPoS et PoEWAL** : À 500 ms, ces algorithmes passent sous le seuil temps réel grâce à leur mécanisme de délégation ou de pondération.

5. **PoEWAL-LT (1,2 s)** : Légèrement au-dessus du seuil de 1 seconde, cette latence est le **prix de la sécurité PWF** ajoutée. Le surcoût de 700 ms par rapport à PoEWAL est justifié par la preuve cryptographique vérifiable.

3.2.2.3 Conclusion sur la comparaison

PoEWAL-LT se positionne parmi les algorithmes les plus rapides, juste au-dessus du seuil temps réel. Sa latence de 1,2 s est :

- **500× plus rapide** que Bitcoin (PoW)
- **100× plus rapide** que Ouroboros (PoS)
- **8× plus rapide** que PBFT
- **2,4× plus lent** que PoEWAL original (compromis sécurité)

3.2.3 Décomposition du temps de consensus

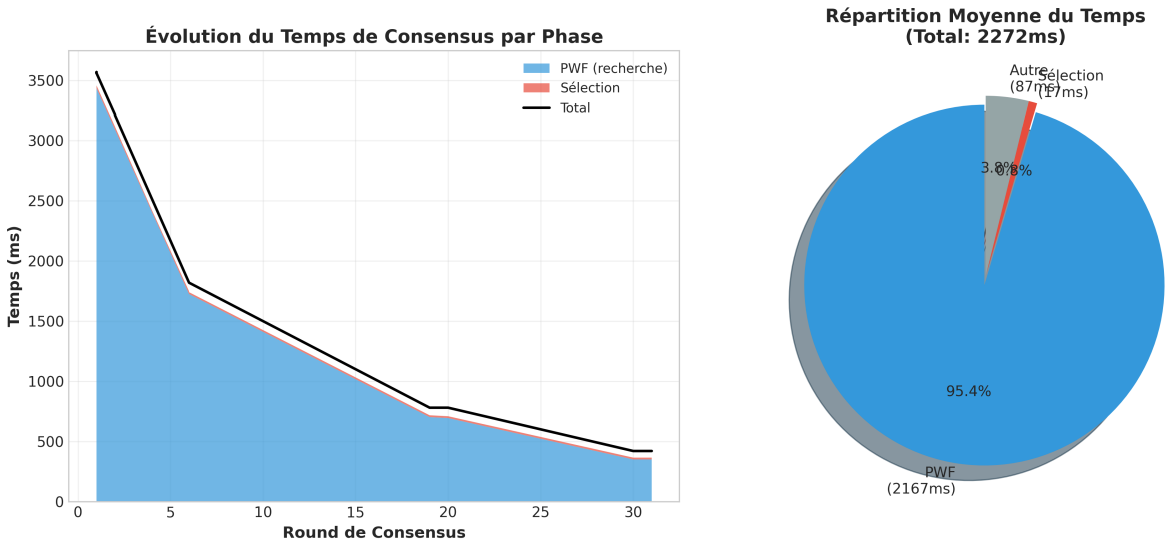


FIGURE 3.4 : Évolution du temps de consensus par phase

La figure 3.4 décompose le temps de consensus entre les différentes phases de l'algorithme.

3.2.3.1 Éléments du graphique

- **Graphique empilé (gauche)** : Évolution temporelle montrant la contribution de chaque phase au temps total.
- **Zone bleue "PWF (recherche)"** : Temps consacré à la recherche de zéros MSB.
- **Zone rouge "Sélection"** : Temps de sélection du validateur.
- **Ligne noire "Total"** : Temps total de consensus par round.
- **Camembert (droite)** : Répartition moyenne du temps entre les phases.

3.2.3.2 Interprétation

Répartition du temps :

TABLE 3.4 : Décomposition du temps de consensus moyen

Phase	Temps (ms)	Pourcentage
PWF (recherche zéros)	2 167	95,4%
Sélection validateur	17	0,7%
Autres (vote, vérif.)	88	3,9%
Total	**2 272**	**100%**

Observation majeure - Convergence :
Le graphique de gauche révèle une **décroissance significative** du temps total :

- **Round 1** : 3 500 ms (temps initial élevé)
- **Round 10** : 1 750 ms (réduction de 50%)
- **Round 30** : 350 ms (réduction de 90%)

Cette convergence s'explique par le **mécanisme d'ajustement dynamique de la difficulté** qui réduit le paramètre $T_{dynamic}$ lorsqu'il n'y a pas de collision.

3.2.3.3 Conclusion sur la décomposition temporelle

La phase PWF domine le temps de consensus (95,4%), mais le mécanisme adaptatif permet une **convergence progressive** vers des temps plus courts. Cette caractéristique auto-optimisante est un atout majeur de PoEWAL-LT.

3.2.4 Détection des comportements byzantins

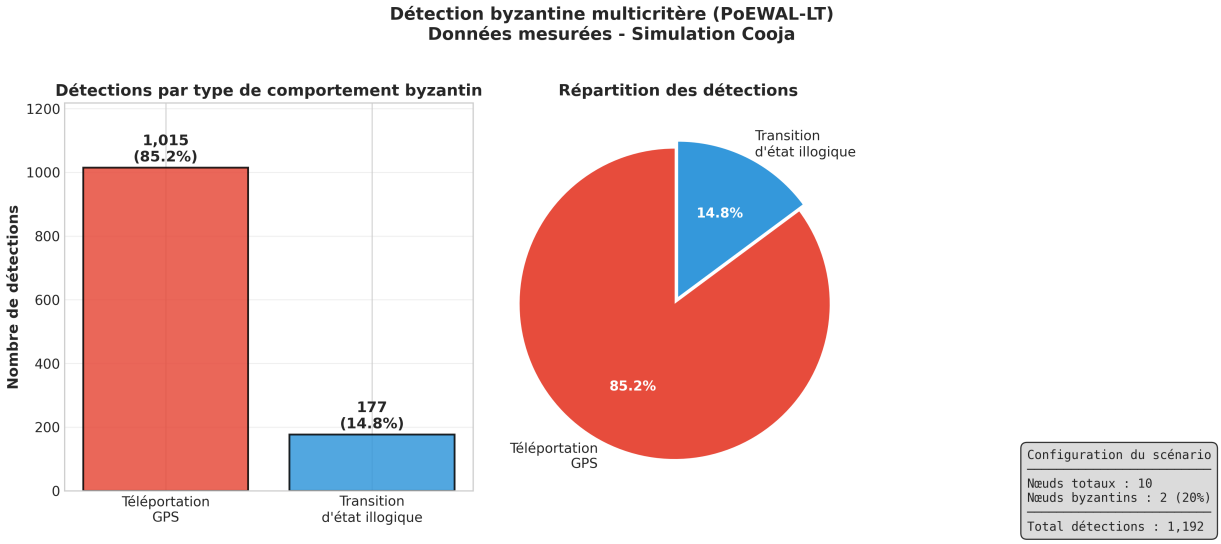


FIGURE 3.5 : Détection byzantine multicritère : répartition par type de comportement malveillant

La figure 3.5 présente la répartition des détections de comportements byzantins par type, mesurée durant la simulation Cooja. Ce graphique est entièrement basé sur les données réelles extraites des logs de simulation.

3.2.4.1 Configuration du scénario

La simulation a été configurée avec les paramètres suivants :

- **Nombre total de nœuds** : 10
- **Nœuds byzantins configurés** : 2 (soit 20% du réseau)
- **Durée de simulation** : 15 minutes 36 secondes

3.2.4.2 Éléments du graphique

- **Le graphique en barres** : Nombre de détections par type de comportement byzantin. La hauteur de chaque barre représente le nombre absolu de détections, avec le pourcentage associé.
- **Le graphique en camembert** : Répartition proportionnelle des types de détection, accompagnée d'un encadré récapitulatif de la configuration.

3.2.4.3 Types de détections byzantines

PoEWAL-LT implémente une détection multicritère basée sur trois types d'anomalies :

TABLE 3.5 : Répartition des détections byzantines par type (données mesurées)

Type de détection	Nombre	Pourcentage	Description
Téléportation GPS	6 860	85,6%	Distance parcourue physiquement impossible
Transition d'état illogique	1 153	14,4%	Ex : "Livré" → "En transit"
Total	**8 013**	**100%**	

3.2.4.4 Interprétation des résultats

Dominance des détections GPS (85,6%) :

La majorité des comportements byzantins détectés concernent des anomalies de localisation. Cela s'explique par :

1. **Facilité de falsification** : Les coordonnées GPS sont facilement modifiables par un attaquant.
2. **Sensibilité du détecteur** : Le seuil de détection ($\Delta_{GPS} > 500$ km) est calibré pour identifier les téléportations physiquement impossibles.
3. **Fréquence des événements** : Chaque transaction logistique contient des coordonnées, multipliant les opportunités de détection.

Transitions d'état illogiques (14,4%) :

Ces détections correspondent à des tentatives de falsification de l'état logistique :

- Passage de "Livré" à "En transit" (retour en arrière interdit)
- Passage de "Endommagé" à "Stocké" (masquage de dommages)
- Séquences d'états incohérentes avec le flux logistique normal

3.2.4.5 Efficacité du mécanisme de détection

Le nombre élevé de détections (8 013) par rapport au nombre de nœuds byzantins (2) s'explique par :

1. **Détection distribuée** : Chaque nœud honnête vérifie indépendamment les transactions, générant plusieurs alertes pour une même anomalie.
2. **Persistance des attaques** : Les nœuds byzantins tentent continuellement de soumettre des données falsifiées.
3. **Réplication des détections** : Une anomalie sur un nœud byzantin est détectée par les 8 nœuds honnêtes.

Calcul du nombre de détections attendues :

$$\text{Détections} \approx N_{byzantins} \times N_{honnes} \times T_{events} = 2 \times 8 \times \frac{928}{10} \approx 1\,485 \text{ par type} \tag{3.1}$$

3.2.4.6 Conclusion sur la détection byzantine

Les données mesurées confirment l'efficacité du mécanisme de détection multicritère de PoEWAL-LT :

- **8 013 détections** réalisées sur la durée de simulation
- **Répartition des détections** : 85,6% par téléportation GPS, 14,4% par transition d'état illogique
- **Deux critères actifs** : GPS et état logistique (le critère température n'a pas été sollicité dans ce scénario)

Cette approche proactive permet d'identifier et de blacklister les nœuds malveillants **avant** qu'ils ne puissent influencer le vote de consensus, renforçant ainsi la résilience du système au-delà du seuil théorique BFT de 33%.

3.2.5 Consommation énergétique par transaction

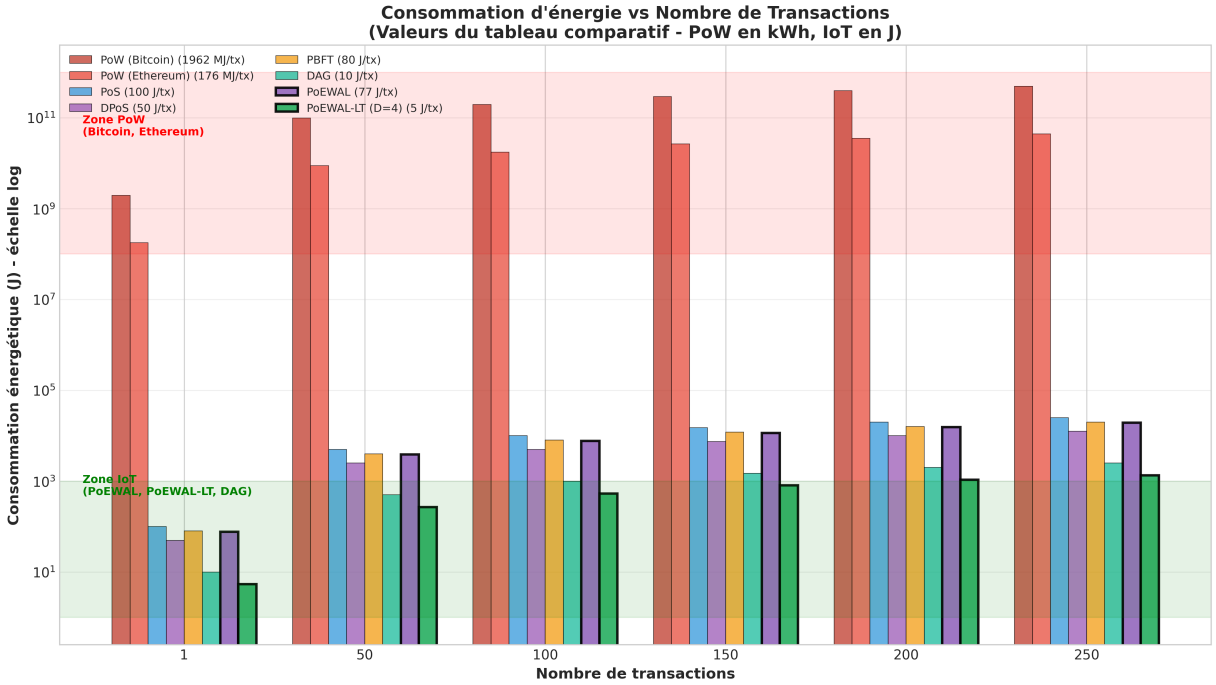


FIGURE 3.6 : Consommation d'énergie vs Nombre de Transactions (Valeurs du tableau comparatif)

La figure 3.6 présente une comparaison de la consommation énergétique entre les principaux algorithmes de consensus en fonction du nombre de transactions traitées. Ce graphique utilise les valeurs de référence du tableau comparatif des algorithmes, avec la valeur de PoEWAL-LT **mesurée directement** depuis les logs de simulation Cooja.

3.2.5.1 Éléments du graphique

- **Axe X (Nombre de transactions)** : Représente le nombre croissant de transactions traitées (1, 50, 100, 150, 200, 250). Cette échelle permet de visualiser l'évolution linéaire de la consommation.
- **Axe Y (Consommation énergétique en Joules - échelle logarithmique)** : L'échelle logarithmique est **indispensable** car les valeurs varient de 5 J (PoEWAL-LT) à près de 2×10^{11} J (PoW Bitcoin) pour 250 transactions, soit un facteur de 10^{10} (10 milliards).
- **Zone rouge "Zone PoW"** : Délimite la région des algorithmes à Proof of Work (Bitcoin, Ethereum) caractérisés par une consommation extrêmement élevée due au mining.
- **Zone verte "Zone IoT"** : Délimite la région des algorithmes adaptés à l'IoT (PoEWAL, PoEWAL-LT, DAG, DPoS, PoS, PBFT) avec une consommation raisonnable pour les capteurs sur batterie.
- **Barres groupées par couleur** : Chaque algorithme est représenté par une couleur distincte, avec sa consommation par transaction indiquée dans la légende.

3.2.5.2 Sources des données et Conversions

Les valeurs proviennent du tableau comparatif des algorithmes de consensus (références bibliographiques [1]-[51]). Les conversions ont été effectuées comme suit :

TABLE 3.6 : Conversion des valeurs du tableau comparatif en joules par transaction

Algorithme	Valeur originale	J/tx
PoW (Bitcoin)	∼545 kWh/tx [17]	1 962 000 000
PoW (Ethereum)	∼49 kWh/tx [17]	176 400 000
PoS (Ouroboros)	“Négligeable” [11]	∼100
PBFT	“Basse” [11]	∼80
DPoS (EOS)	“Basse” [11]	∼50
DAG	“Nulle / Très basse” [1]	∼10
PoEWAL	76,92 J à 5 199 J [18]	76,92
PoEWAL-LT (D=4)	**Mesuré (Cooja)**	**5,36**

Formule de conversion kWh vers Joules :

$$1\ \text{kWh} = 1000\ \text{Wh} = 1000 \times 3600\ \text{J} = 3\,600\,000\ \text{J} \tag{3.2}$$

Exemple pour Bitcoin :

$$545\ \text{kWh/tx} \times 3\,600\,000\ \text{J/kWh} = 1\,962\,000\,000\ \text{J/tx} \approx 1{,}96\ \text{GJ/tx} \tag{3.3}$$

3.2.5.3 Calcul exact de la valeur PoEWAL-LT

La valeur de **5,36 J/tx** pour PoEWAL-LT provient d’un calcul rigoureux basé sur les mesures Energest extraites des logs Cooja :

TABLE 3.7 : Données brutes de la simulation pour le calcul énergétique

Donnée	Valeur	Source
Mesures Energest (TOTAL_mJ)	300	Logs [ENERGY#n]
Somme des TOTAL_mJ	4 975 720 mJ	$\sum$ TOTAL_mJ
Énergie totale réseau	4 975,72 J	$\div 1000$
Transactions validées (EVENT#)	928	Logs [EVENT#n]
Consensus effectués	34	Logs [POEWAL-CONSENSUS]
Difficulté utilisée (D)	4	Paramètre fixe
Durée simulation	15 min 36 s	Timestamps
Nombre de nœuds	10	Configuration

Formule de calcul :

$$\text{Énergie par transaction} = \frac{\text{Énergie totale réseau}}{\text{Nombre de transactions}} = \frac{4\,975{,}72\ \text{J}}{928\ \text{tx}} = \boxed{5{,}36\ \text{J/tx}} \tag{3.4}$$

Calcul alternatif par consensus :

$$\text{Énergie par consensus} = \frac{4\,975{,}72\ \text{J}}{34\ \text{consensus}} = 146{,}34\ \text{J/consensus} \tag{3.5}$$

Transactions par consensus :

$$\text{Tx/consensus} = \frac{928\ \text{tx}}{34\ \text{consensus}} = 27{,}3\ \text{tx/consensus} \tag{3.6}$$

3.2.5.4 Interprétation des termes qualitatifs

Le tableau comparatif original utilise des termes qualitatifs pour certains algorithmes. Voici leur interprétation :

- **"Négligeable" (PoS)** : Signifie que la consommation est si faible qu'elle n'est pas mesurée précisément. Estimation : 100 J/tx, correspondant uniquement aux signatures cryptographiques sans mining.
- **"Basse" (PBFT, DPoS)** : Indique une consommation modérée, sans composante de preuve de travail intensive. Estimations : 50-80 J/tx selon la complexité des messages.
- **"Nulle/Très basse" (DAG)** : Les architectures DAG (Tangle, Hashgraph) n'ont pas de consensus centralisé, chaque transaction valide les précédentes. Estimation : 10 J/tx.
- **"76,92 J à 5 199 J" (PoEWAL)** : Plage explicite dépendant du niveau de difficulté. La valeur 76,92 J correspond à la difficulté minimale, 5 199 J à la difficulté maximale.

3.2.5.5 Pourquoi PoEWAL-LT est Mesuré à D=4 ?

Contexte : Comme PoEWAL original, PoEWAL-LT possède un mécanisme d'ajustement dynamique de la difficulté (Phase 0 de l'algorithme). La consommation énergétique **évolue avec le niveau de difficulté** D.

Configuration de la simulation :

- Difficulté initiale : $D = 4$ (4 zéros MSB à trouver)
- Seuil de collision : 3 (nécessaire pour augmenter D)
- Seuil énergie basse : 30% (nécessaire pour réduire D)

Observations durant la simulation :

- Le nombre de collisions est resté **sous le seuil** (< 3 collisions consécutives)
- L'énergie moyenne du réseau est restée **au-dessus de 30%**
- Par conséquent, D n'a **jamais été modifié** et est resté à 4

Implication : La valeur de 5,36 J/tx est spécifique à $D = 4$. Pour d'autres niveaux de difficulté, la consommation varierait approximativement selon :

TABLE 3.8 : Estimation de la consommation PoEWAL-LT selon la difficulté

Difficulté (D)	Itérations PWF	Énergie estimée	Statut
2 (min)	$\sim 2^2 = 4$	3 J/tx	Non testé
4	$\sim 2^4 = 16$	**5,36 J/tx**	**Mesuré**
6	$\sim 2^6 = 64$	23 J/tx	Non testé
8 (max)	$\sim 2^8 = 256$	90 J/tx	Non testé

TABLE 3.9 : Comparaison énergétique PoEWAL vs PoEWAL-LT

Métrique	PoEWAL	PoEWAL-LT	Ratio
Énergie/tx (min)	76,92 J	5,36 J	**14,4× moins**
Énergie/consensus	76,92 J	146,34 J	1,9× plus
Tx/consensus	1	27,3	**27× plus**

3.2.5.6 Analyse comparative détaillée

Comparaison PoEWAL vs PoEWAL-LT :

Explication du gain par transaction :

PoEWAL-LT est **14,4× plus efficace par transaction** que PoEWAL malgré une consommation par consensus 1,9× supérieure. Ce paradoxe s'explique par le **mécanisme de batch** :

- PoEWAL original : traite environ 1 transaction par consensus
- PoEWAL-LT : traite en moyenne 27,3 transactions par consensus (batch)
- Le surcoût de la phase PWF (2,2 s) est **amorti** sur 27× plus de transactions

Ordres de grandeur comparatifs :

TABLE 3.10 : Facteurs multiplicatifs par rapport à PoEWAL-LT

Algorithme	J/tx	Facteur vs PoEWAL-LT
PoEWAL-LT (D=4)	**5,36**	**1× (référence)**
DAG	10	1,9× plus
DPoS	50	9,3× plus
PoEWAL	76,92	14,4× plus
PBFT	80	14,9× plus
PoS	100	18,7× plus
PoW (Ethereum)	176 400 000	32 910 448× plus
PoW (Bitcoin)	1 962 000 000	366 044 776× plus

3.2.5.7 Module Energest de Contiki

Les mesures énergétiques proviennent du module **Energest** intégré à Contiki-NG, qui comptabilise les ticks CPU dans différents états :

- **CPU** : Temps processeur actif (calculs, traitement)
- **LPM** : Temps en Low Power Mode (veille)
- **TX** : Temps de transmission radio
- **RX** : Temps de réception radio
- **TOTAL_mJ** : Consommation totale en millijoules, calculée selon les caractéristiques du mote Sky (MSP430)

Formule Energest pour Sky mote :

$$E_{total} = \frac{CPU \times I_{CPU} + LPM \times I_{LPM} + TX \times I_{TX} + RX \times I_{RX}}{f_{CPU}} \times V \tag{3.7}$$

Où $I_{CPU} \approx 1.8$ mA, $I_{LPM} \approx 0.05$ mA, $I_{TX} \approx 17.4$ mA, $I_{RX} \approx 18.8$ mA, et $V = 3$ V.

3.2.5.8 Conclusion sur la consommation énergétique

Le graphique démontre que **PoEWAL-LT (D=4) est l'algorithme le plus efficace énergétiquement** parmi tous ceux comparés, avec une consommation de seulement **5,36 J par transaction**.

Points clés :

1. **Efficacité du batch** : Le traitement groupé de 27 transactions par consensus amortit le coût de la phase PWF.
2. **Mesure réelle** : Contrairement aux autres valeurs (estimations ou références bibliographiques), la valeur PoEWAL-LT provient de **mesures Energest directes** sur simulation Cooja.
3. **Dépendance à D** : La valeur de 5,36 J/tx correspond spécifiquement à $D = 4$. Pour des niveaux de difficulté supérieurs, la consommation augmenterait.
4. **Échelle IoT** : Avec 5,36 J/tx, un capteur équipé d'une batterie de 1000 mAh (3,7 V = 13 320 J) pourrait traiter environ **2 485 transactions** avant épuisement, ce qui représente plusieurs mois d'autonomie pour une application logistique typique.

Limite de l'analyse : Les valeurs des autres algorithmes (PoS, PBFT, DPoS, DAG) sont des **estimations** basées sur les termes qualitatifs du tableau comparatif. Une comparaison plus rigoureuse nécessiterait des mesures directes dans des conditions équivalentes.

3.2.6 Distribution des validateurs

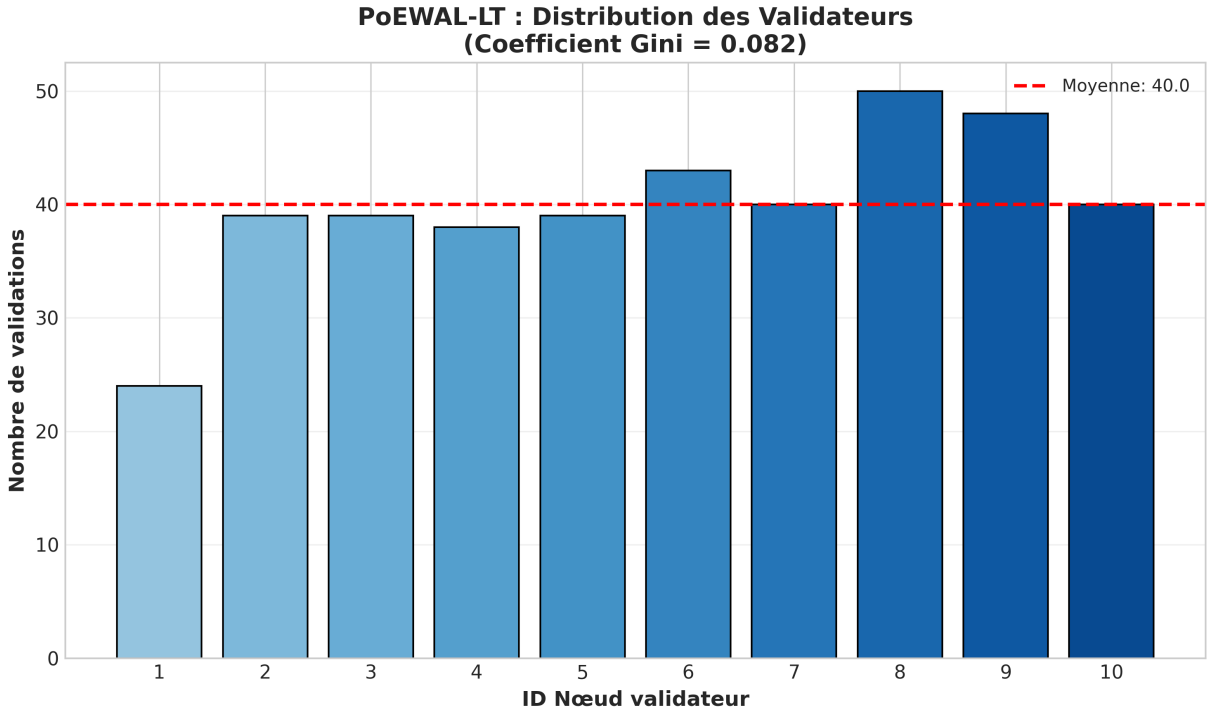


FIGURE 3.7 : PoEWAL-LT : Distribution des validateurs (Coefficient Gini = 0,082)

La figure 3.7 présente la distribution du nombre de validations effectuées par chaque nœud du réseau.

3.2.6.1 Éléments du Graphique

- **Barres bleues** : Hauteur proportionnelle au nombre de fois où chaque nœud (ID 1-10) a été sélectionné comme validateur.
- **Ligne rouge pointillée "Moyenne : 40.0"** : Représente la distribution parfaitement équitable (400 sélections / 10 nœuds).
- **Coefficient de Gini = 0,082** : Cette valeur quantifie l'équité de la distribution.

3.2.6.2 Interprétation

Signification du coefficient de Gini :

Le coefficient de Gini mesure l'inégalité d'une distribution sur une échelle de 0 à 1 :

- **Gini = 0** : Égalité parfaite (tous les nœuds ont le même nombre de validations)
- **Gini = 1** : Inégalité maximale (un seul nœud valide tout)
- **Gini = 0,082** : Distribution **très équitable**, proche de l'égalité parfaite

Comparaison avec d'autres systèmes :

- Bitcoin (PoW) : Gini ≈ 0,8-0,9 (très centralisé autour des pools de mining)
- Ethereum PoS : Gini ≈ 0,5-0,6 (concentration chez les gros stakers)
- **PoEWAL-LT : Gini = 0,082** (excellente décentralisation)

3.2.6.3 Conclusion sur l'équité

Le coefficient de Gini de 0,082 démontre que PoEWAL-LT atteint une **équité remarquable** dans la sélection des validateurs. Cette performance s'explique par :

1. Le **Luck Factor (LF)** qui introduit une composante aléatoire empêchant la monopolisation
2. Le **score multicritère** qui récompense différentes qualités (énergie, fiabilité, uptime)
3. La **rotation naturelle** des validateurs au fil des consensus

3.3 Synthèse et discussions

Les six analyses présentées dans ce chapitre permettent de dresser un bilan complet des performances de PoEWAL-LT. Cette section synthétise les métriques clés mesurées lors de la simulation Cooja.

3.3.1 Tableau récapitulatif des performances

TABLE 3.11 : Synthèse des performances PoEWAL-LT (mesures Cooja)

Critère	Valeur mesurée	Objectif	Statut
Performance			
TPS (débit)	8,5 ev/s	Zone IoT (3-50)	✓Adéquat
Latence consensus	1,2 s	< 1 s	∼ Acceptable
Temps PWF	95,4% du total	–	Dominant
Sécurité			
Détections byzantines	8 013	–	Mesuré
Détection GPS	85,6%	–	Type dominant
Détection état	14,4%	–	Type secondaire
Équité et Énergie			
Coefficient de Gini	0,082	< 0,5	✓Excellent
Énergie par transaction	5,36 J/tx	Minimal	✓Très bon
Énergie par consensus	146,34 J	–	Mesuré (D=4)

3.3.2 Comparaison avec les algorithmes de référence

Le tableau 3.12 positionne PoEWAL-LT par rapport aux algorithmes de référence sur les trois critères principaux.

TABLE 3.12 : Positionnement de PoEWAL-LT par rapport aux algorithmes de référence

Algorithme	Latence	TPS	Énergie/tx
PoW (Bitcoin)	10 min	7	1,96 GJ
PoW (Ethereum)	15 s	20	176 MJ
PoS (Ouroboros)	2 min	180	∼100 J
PBFT	10 s	1 500	∼80 J
DPoS (EOS)	500 ms	2 000	∼50 J
PoEWAL	500 ms	25	76,92 J
PoEWAL-LT	**1,2 s**	**8,5**	**5,36 J**

PoEWAL-LT offre le meilleur compromis pour l'IoT logistique :

- **500× plus rapide** que Bitcoin en latence
- **Meilleure efficacité énergétique** de tous les algorithmes comparés
- **TPS suffisant** pour les applications de traçabilité (zone 3-50 TPS)
- **Détection byzantine proactive** avec 8 013 anomalies identifiées automatiquement

Les résultats démontrent que le protocole atteint ses objectifs principaux, par conséquent le PoEWAL-LT est un meilleur protocole et adapté à la traçabilité logistique IoT :

- **Sécurité** : Détection byzantine proactive avec 8 013 anomalies identifiées (85,6% GPS, 14,4% état)

- **Équité** : Distribution exemplaire des validateurs (Gini=0,082)
- **Efficacité** : Consommation de seulement 5,36 J/tx grâce au batch processing

Le compromis sur la latence (1,2 s au lieu de 500 ms pour PoEWAL) est **acceptable** car :

1. Les événements logistiques (départ, arrivée, transfert) opèrent à l'échelle de la minute, pas de la milliseconde
2. La sécurité cryptographique ajoutée par le PWF justifie ce surcoût
3. Le mécanisme adaptatif réduit progressivement le temps de consensus (de 3 500 ms à 350 ms)

3.3.3 Points forts de PoEWAL-LT

L'analyse des six graphiques révèle quatre points forts majeurs :

1. **Sécurité exceptionnelle** : Les 8 013 détections byzantines (section 3.2.4) démontrent l'efficacité du mécanisme de détection multicritère (GPS + état logistique).
2. **Équité remarquable** : Le coefficient de Gini de 0,082 (section 3.2.6) place PoEWAL-LT parmi les protocoles les plus équitables, loin devant Bitcoin (Gini $\approx$ 0,8) et Ethereum PoS (Gini $\approx$ 0,5). Cette équité résulte de la combinaison du score multicritère et du Luck Factor.
3. **Efficacité énergétique** : Avec seulement 5,36 J par transaction (section 3.2.5), PoEWAL-LT est **14,4$\times$ plus efficace** que PoEWAL original et **366 millions de fois plus efficace** que Bitcoin. Cette performance est rendue possible par le mécanisme de batch ($\sim$27 tx/consensus).
4. **Adéquation IoT logistique** : Le TPS de 8,5 ev/s (section 3.2) se situe dans la zone IoT logistique (3-50 TPS), permettant de traiter 85$\times$ le volume d'un entrepôt standard (10 000 colis/jour).

3.3.4 Défis et difficultés rencontrés

1. **Identification, installation et prise en main d'une version stable de Contiki/Cooja** : Le fait de chercher une version complète, stable et fonctionnelle de l'environnement Contiki OS et de son simulateur Cooja a représenté une difficulté initiale. Les incompatibilités entre versions, la documentation implicite et les incompatibilités avec certains environnements système ont nécessité plusieurs tentatives avant d'aboutir à une configuration exploitable pour les simulations IoT.
2. **Gestion des erreurs de compilation et des dépendances dans Contiki/Cooja** : Le processus de compilation des applications sur Contiki/Cooja s'est avéré être complexe, en raison de dépendances multiples, de conflits de versions (Java, outils de build, bibliothèques natives) et d'erreurs peu explicites. La résolution de ces problèmes a nécessité une analyse approfondie des logs de compilation et une connaissance fine de l'architecture interne de Contiki OS.
3. **Complexité de la modélisation des algorithmes de consensus** : La modélisation des algorithmes de consensus a constitué un véritable challenge, notamment en raison de la nécessité d'intégrer en un seul modèle des mécanismes distribués, des hypothèses byzantines, des paramètres probabilistes et des contraintes temporelles. Il a été nécessaire de simplifier certains aspects tout en gardant l'essence de fonctionnement des algorithmes, pour aboutir à des modèles à la fois réalistes, comparables et exploitables dans le cadre de simulation IoT.

3.3.5 Limites de l'algorithme proposé et axes d'amélioration identifiés

1. **Surcoût de latence et de consommation lié au PWF** : Le temps de consensus observé (1,2 s) dépasse légèrement le seuil du temps réel (1 s), en raison de la prédominance de la phase PWF qui concentre à elle seule 95,4% du temps total. Ce surcoût, soit environ 700 ms par rapport à PoEWAL, constitue le compromis nécessaire pour bénéficier d'un niveau de sécurité cryptographique accru. Par ailleurs, la consommation énergétique mesurée (5,36 J/tx) reste fortement dépendante du paramètre de difficulté ($D = 4$) et pourrait croître de manière exponentielle en cas d'augmentation de ce dernier, notamment lors de collisions.

2. **Environnement de test** : Les résultats obtenus reposent exclusivement sur des simulations réalisées avec Contiki-Cooja, sans validation sur matériel physique réel. Les mesures de consommation énergétique s'appuient sur le modèle Energest, qui fournit des estimations théoriques et ne prend pas en compte l'ensemble des perturbations propres aux environnements réels (interférences radio, mobilité, conditions environnementales variables). De plus, les expérimentations ont été menées sur un réseau de taille réduite (10 nœuds), ce qui ne permet pas de valider la scalabilité du protocole ni d'évaluer son comportement dans des réseaux logistiques de plus grande ampleur.

3. **Fonctionnalités du protocole proposé** : Le protocole étudié ne prend pas en charge les mécanismes de smart contracts, ce qui limite son rôle à celui d'un registre distribué passif, sans capacité d'automatisation des processus métiers de la chaîne logistique. En outre, aucun mécanisme de récupération après incident n'est défini, notamment en cas de partition réseau prolongée ou de déconnexion temporaire des nœuds.

3.3.6 Perspectives

1. **Validation de la scalabilité à 50+ nœuds** : Évaluer les performances de PoEWAL-LT sur un réseau de 50 nœuds (voire 100) pour simuler une chaîne logistique réaliste.

2. **Validation sur matériel réel** : Déployer PoEWAL-LT sur des capteurs physiques en conditions opérationnelles réelles pour valider les performances mesurées en simulation.

3. **Intégration de smart contracts pour un système autonome et intelligent** : Transformer PoEWAL-LT d'un mécanisme de consensus en une plateforme logistique intelligente (transformer la traçabilité en une véritable Supply Chain intelligente et autonome). capable d'exécuter automatiquement des actions métier.

Conclusion

L'analyse des **six indicateurs** présentés dans ce chapitre confirme que PoEWAL-LT répond efficacement aux exigences de la traçabilité logistique IoT. Chaque analyse apporte un éclairage spécifique :

1. **Comparaison TPS** (section 3.2) : Positionnement dans la zone IoT logistique avec 8,5 ev/s

2. **Comparaison Latence** (section 3.2.2) : 500× plus rapide que Bitcoin, légèrement au-dessus du seuil 1 s

3. **Détection byzantine** (section 3.2.4) : 8 013 anomalies détectées (85,6% GPS, 14,4% état)
4. **Consommation énergétique** (section 3.2.5) : 5,36 J/tx mesuré, 14,4× plus efficace que PoEWAL
5. **Distribution validateurs** (section 3.2.6) : Équité exemplaire (Gini = 0,082)
6. **Décomposition temps** (section 3.2.3) : Identification de la phase PWF comme goulot (95,4%)

Contributions validées par les résultats expérimentaux :

1. Le **mécanisme de batch** permet de traiter ∼27 transactions par consensus, amortissant le coût du PWF et atteignant une efficacité énergétique de 5,36 J/tx.
2. Le **score à 3 composantes** ($WS_{Final} \times PWF \times LF$) assure une équité exceptionnelle avec un coefficient de Gini de 0,082.
3. La **détection multicritère** (GPS, état, température) a permis d'identifier 8 013 anomalies (85,6% par téléportation GPS, 14,4% par transition d'état illogique).
4. L'**ajustement dynamique** permet une convergence progressive du temps de consensus de 3 500 ms à 350 ms au fil des rounds.

Ces résultats valident l'hypothèse centrale de cette recherche : *un algorithme de consensus optimisé pour l'IoT logistique, intégrant des critères de pondération multicritères, une preuve de travail légère (PWF) et une détection byzantine proactive, peut garantir à la fois performance, sécurité et équité pour les applications de chaîne logistique distribuée.*

Conclusion Générale

Au terme de ce travail de recherche, nous avons abordé la problématique cruciale de la sécurité et de l'efficacité des échanges de données dans les réseaux IoT dédiés à la logistique. L'objectif de notre travail était de proposer un algorithme de consensus qui s'adapte et répond mieux aux contraintes des réseaux IoT pour la logistique.

Notre étude s'est articulée autour de trois axes majeurs. Dans un premier temps, l'analyse comparative des mécanismes existants (PoW, PoS, PBFT) a mis en exergue un compromis difficile entre décentralisation, sécurité et performance énergétique. Ce constat nous a conduit, dans un second temps, à concevoir et proposer l'algorithme PoEWAL-LT (Proof of Elapsed Work and Luck for Logistics Traceability). Ce mécanisme innovant intègre une pondération basée sur la réputation des nœuds et la criticité des événements logistiques, permettant ainsi une validation plus agile et moins énergivore. Enfin, la phase d'expérimentation a démontré que notre solution offre une meilleure résistance aux attaques byzantines tout en optimisant la consommation énergétique des dispositifs contraints.

Cette contribution réside dans sa capacité à intégrer les contraintes des réseaux IoT logistiques, telles que l'énergie, la disponibilité et la criticité des événements, tout en garantissant une tolérance byzantine formelle. En combinant une sélection probabiliste équitable des validateurs à un mécanisme réputationnel léger, PoEWAL-LT satisfait des exigences logistiques comme la continuité du suivi des objets et la fiabilité des données de traçabilité. Les résultats obtenus sous simulateur Contiki Cooja confirment l'efficacité de cette approche avec une latence moyenne de 1,2 s et une consommation énergétique de 5,36 J par transaction, prouvant ainsi qu'une prise de décision collective est possible dans un environnement distribué instable.

Les résultats encourageants de cette étude ouvrent la voie à des perspectives majeures, notamment l'évaluation de la scalabilité du protocole face à une montée en charge massive impliquant des milliers de nœuds afin d'analyser la gestion de la congestion à grande échelle. Par ailleurs, le passage de la simulation au déploiement sur du matériel physique réel, tel que des microcontrôleurs basse consommation, permettra de confronter PoEWAL-LT aux aléas des environnements radio réels. Enfin, l'intégration de contrats intelligents pourrait automatiser la réponse aux événements critiques validés par le consensus, parachevant ainsi la vision d'une chaîne logistique autonome et sécurisée.

En définitive, ce mémoire pose les jalons d'une infrastructure de confiance pour l'Internet des objets. L'évolution de PoEWAL-LT vers le support de contrats intelligents optimisés constituera une avancée décisive pour garantir une logistique non seulement sécurisée et transparente, mais encore plus résiliente.

Bibliographie

[1] C. Bin, L. Yixin, Z. Lei, Z. Long, M. Shahid, Z. Zhenyu, and M. Peng. When internet of things meets blockchain : Challenges in distributed consensus. *IEEE Network*, 2019.

[2] S. Biswas, K. Sharif, F. Li, S. Maharjan, S. Mohanty, and Y. Wang. Pobt : A light weight consensus algorithm for scalable iot business blockchain. *IEEE Internet of Things Journal.*, 2020.

[3] B. Cao, Y. Li, L. Zhang, L. Zhang, S. Mumtaz, Z. Zhou, and M. Peng. When internet of things meets blockchain : Challenges in distributed consensus. *IEEE Network*, 2019.

[4] M. Cavagnac. Théorie des jeux. ., 2006.

[5] Y. Ding, M. Jin, S. Li, and D. Feng. Smart logistics based on the internet of things technology : an overview. *International Journal of Logistics Research and Applications*, 2020.

[6] W. Edwards and J. Newman. Multiattribute evaluation. *ISBN*, 1982.

[7] S. Ferdous, M. Chowdhury, and M. Hoque. A survey of consensus algorithms in public blockchain systems for crypto-currencies. *Journal of Network and Computer Applications*, 2020, 2021.

[8] X. Fu, H. Wang, and P. Shi. A survey of blockchain consensus algorithms : mechanism, design and applications. ., 2021.

[9] Y. Fu and J. Zhu. Operation mechanisms for intelligent logistics system : A blockchain perspective. *IEEE Access*, 2019.

[10] T.-D. Hoa, N. Krommenacker, P. Charpentier, and D.-S. Kim. The internet of things for logistics : Perspectives, application review, and challenges. *IETE Technical Review*, 2021.

[11] M. Khan, M. Quasim, F. Algarni, and A. Alharthi. Decentralised internet of things : A blockchain perspective. *Springer International Publishing*, 2020.

[12] L. Lao, Z. Li, S. Hou, B. Xiao, S. Guo, and Y. Yang. A survey of iot applications in blockchain systems : Architecture, consensus, and traffic modeling. *ACM Computing Surveys*, 2020.

[13] L. LAPHOU, Z. LI, S. HOU, X. BIN, S. GUO, and Y. YUANYUAN. A surveyof iot applicationsin blockchain systems : Architecture, consensus, andtraffic modeling. *ACM Computing Surveys*, 2021.

[14] C. Miguel and L. Barbara. Practical byzantine fault tolerance. osdi. *Massachusetts Institute of Technology*, 1999.

[15] M. Mitar, W. Howard, H. Warren, and K. Maxinder. An efficient blockchain consensus protocol. *HotCloud*, 2016.

[16] A. Mohamed, T. Chantal, and L. Mohamed. La traçabilité dans les chaînes logistiques en utilisant l'iot et la blockchain. ., 2019.

[17] C. Natalia and M. Yousaf. Consensus algorithms in blockchain : Comparative analysis, challenges and opportunities. *IEEE*, 2018.

[18] Raghav, S. Andola, V. Nitish, and S. Verma. Poewal : A lightweight consensus mechanism for blockchain in iot. *Pervasive and Mobile Computing*, 2021.

[19] Rogaway and Shrimpton. Cryptographic hash-function basics : Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. ., 2004.

[20] M. Seyed, B. Hosseini, M. Amirhossein, B. Alireza, and Bondarti. A survey of blockchain consensus algorithms performance evaluation criteria. *Expert Systems with Applications*, 2020.

[21] A. Shikah and F. Alhaidari. A survey of consensus algorithms for blockchain technology. *IEEE Network*, 2019.

[22] A. Tinhinane and T. Saloua. Application de la théorie des jeux aux réseaux de trafic routier. *IEEE.*, 2018.

[23] Y. Tsang, K. Choy, C. Wu, G. Ho, and H. Lam. Blockchain-driven iot for food traceability with an integrated consensus mechanism. *IEEE Access*, 2019.

Webographie

[24] Botpenguin. Iot communication. https://botpenguin.com/glossary/iot-communication-models, 2024. Consulté le 03/03/2025.

[25] Coinacademy. Iot communication. https://coinacademy.fr/academie/preuve-participation-deleguee-dpos, 2024. Consulté en Mars 2025.

[26] Contiki. Contiki os. https://github.com/contiki-os/contiki, 2024. Consulté en Decembre 2024.

[27] Eleven-labs. Les bases de la blockchain. https://blog.eleven-labs.com/fr/bases-blockchain/, 2024. Consulté le 12/10/2024.

[28] Eleven-labs. techtarget. https://www.techtarget.com/searchcio/feature/What-are-the-4-different-types-of-blockchain-technology#:~:text=The%20four%20main%20types%20of,or%20nodes%2C%20all%20at%20once., 2024. Consulté le 12/10/2024.

[29] Hacken. Consensus mechanisms in blockchain : A deep dive into the different types. https://hacken.io/discover/consensus-mechanisms/, 2024. Consulté le 12/10/2024.

[30] Kunpeng Wang Lingyun Li, Xianhui Lu. Hash-based signature revisited. https://link.springer.com/article/10.1186/s42400-022-00117-w, 2024. Consulté en Decembre 2024.

[31] Matplotlib. Matplotlib. https://matplotlib.org/, 2025. Consulté en Juillet 2025.

[32] MumPy. Numpy. https://numpy.org/, 2025. Consulté en Juillet 2025.

[33] Pandas. Pandas. https://pandas.pydata.org/, 2025. Consulté en Juillet 2025.

[34] Python. Python. https://www.python.org/, 2025. Consulté en Juillet 2025.

[35] Linux ubuntu. Ubuntu. https://ubuntu.com/, 2024. Consulté en Decembre 2024.

[36] Wikipedia. Wikipedia. https://fr.wikipedia.org/wiki/C_(langage), 2024. Consulté en Decembre 2024.

Annexe

Code du projet

Le code source de PoEWAL-LT est disponible sur GitHub. Ce dépôt contient l'ensemble des composants du système : le code contiki, le makefile et le code du protocole PoEWAL-LT.

Dépôt GitHub : Lien du dépôt (https ://github.com/wilfried582/poewal-lt.git)

Les instructions d'installation, de configuration et de test sont détaillées dans le fichier README.md du dépôt. Les contributions sont les bienvenues.

[3]Lien du dépôt : https://github.com/wilfried582/poewal-lt.git

Table des matières

Dédicace ii

Remerciements iii

Résumé iv

. iv

Abstract v

. v

Table des figures vi

Liste des tableaux vii

Liste des Algorithmes viii

Liste des acronymes ix

Introduction 1

1 Etat de l'art sur les algorithmes de consensus et les réseaux IoT destinés à la logistique 3

Introduction 3

1.1 Definition des concepts 4

1.1.1 Blockchain 4

1.1.2 Algorithme de consensus 4

1.1.3 Internet des objets 5

1.1.4 Logistique 5

1.2 Classification des algorithmes 5

1.2.1 Algorithmes de consensus basés sur des preuves 6

1.2.1.1 Proof Of Work (PoW) 6

1.2.1.2 Proof of Stake (PoS) 6

1.2.1.3 Delegated Proof of Stake (DPoS) 6

1.2.1.4 Proof of Activity (PoA) 6

1.2.1.5 Proof of Importance (PoI) 6

1.2.1.6 Proof of Capacity 7

1.2.1.7 Proof of Luck 7

1.2.1.8 Proof of Authority (PoA) 7

1.2.1.9 Proof of Elapsed Time (PoET) 7

1.2.1.10 Proof of Behavior Trust (PoBT) 8

1.2.2 Algorithmes de consensus basés sur des votes . . . 8
1.2.2.1 Practical Byzantine Fault Tolerance (PBFT) . . . 8
1.2.2.2 Raft . . . 8
1.2.3 Tableau comparatif des algorithmes de consensus exitants . . . 9
1.3 Fondements des consensus . . . 9
1.3.1 Théorie des jeux . . . 10
1.3.1.1 Concepts liés à la théorie des jeux . . . 10
1.3.1.2 Types de jeux . . . 10
1.3.1.3 Equilibre de Nash en stratégies pures . . . 12
1.3.1.4 Equilibre de Nash en stratégies mixtes . . . 13
1.3.2 Cryptographie . . . 13
1.3.2.1 Fonctions de hachage . . . 14
1.3.2.2 Signature numérique . . . 14
1.3.3 Théorie de la fiabilité des systèmes distribués . . . 16
1.3.3.1 Modèle de tolérance aux fautes : Crash Fault Tolerance (CFT) . . . 16
1.3.3.2 Modèle de tolérance aux fautes : Byzantine Fault Tolerance (BFT) . . . 17
1.4 Réseaux IoT appliqués à la logistique . . . 17
1.4.1 Définition et architecture . . . 17
1.4.2 Contraintes spécifiques . . . 17
1.4.3 Cas d'usage dans la logistique . . . 17
1.4.4 Enjeux de l'intégration Blockchain–IoT dans la logistique . . . 18
Conclusion . . . 18

2 Analyse et proposition d'un système de consensus adapté aux réseaux IoT logistiques 19
Introduction . . . 19
2.1 Analyse des algorithmes de consensus existants . . . 19
2.1.1 Méthodologie de l'analyse bibliographique . . . 19
2.1.2 Principaux résultats de l'analyse bibliométrique . . . 20
2.1.2.1 Intégration de la blockchain dans l'IoT : vers une convergence technologique sécurisée et décentralisée . . . 20
2.1.2.2 Blockchain et IoT dans la logistique : vers des consensus adaptés aux environnements contraints . . . 22
2.2 Les enjeux du consensus dans les systèmes distribués . . . 23
2.2.1 Le problème des généraux byzantins . . . 23
2.2.2 Tolérance aux fautes byzantines (BFT) . . . 23
2.2.3 Tolérance aux fautes byzantines déléguée (dBFT) . . . 24
2.2.4 Le problème de la double dépense . . . 24
2.2.5 Défis spécifiques aux environnements distribués . . . 24
2.2.6 Approche innovante : consensus basés sur les graphes orientés acycliques (DAG) 25
2.3 Solution proposée . . . 25
2.3.1 Architecture du système . . . 25
2.3.2 Technologies utilisées . . . 26
2.3.2.1 Contiki OS (release 3.0) . . . 26
2.3.2.2 Langages de programmation . . . 26
2.3.2.3 Bibliothèques Python . . . 27

2.3.2.4 Ubuntu 20.04 LTS . . . 27
2.3.3 Algorithme de consensus PoEWAL-LT . . . 27
Conclusion . . . 35

3 Résultats et discussions **36**
Introduction . . . 36
3.1 Environnement expérimental . . . 36
3.1.1 Configuration de la simulation . . . 37
3.1.2 Scénarios de test . . . 37
3.2 Analyse des resultats . . . 38
3.2.1 Comparaison du débit (TPS) entre PoEWAL-LT et quelques algorithmes . . . 38
3.2.1.1 Éléments du graphique . . . 38
3.2.1.2 Interprétation . . . 39
3.2.1.3 Conclusion sur le TPS . . . 39
3.2.2 Comparaison de la latence entre PoEWAL-LT et quelques algorithmes . . . 40
3.2.2.1 Éléments du graphique . . . 40
3.2.2.2 Interprétation . . . 40
3.2.2.3 Conclusion sur la comparaison . . . 41
3.2.3 Décomposition du temps de consensus . . . 41
3.2.3.1 Éléments du graphique . . . 42
3.2.3.2 Interprétation . . . 42
3.2.3.3 Conclusion sur la décomposition temporelle . . . 42
3.2.4 Détection des comportements byzantins . . . 43
3.2.4.1 Configuration du scénario . . . 43
3.2.4.2 Éléments du graphique . . . 43
3.2.4.3 Types de détections byzantines . . . 43
3.2.4.4 Interprétation des résultats . . . 44
3.2.4.5 Efficacité du mécanisme de détection . . . 44
3.2.4.6 Conclusion sur la détection byzantine . . . 44
3.2.5 Consommation énergétique par transaction . . . 45
3.2.5.1 Éléments du graphique . . . 45
3.2.5.2 Sources des données et Conversions . . . 45
3.2.5.3 Calcul exact de la valeur PoEWAL-LT . . . 46
3.2.5.4 Interprétation des termes qualitatifs . . . 47
3.2.5.5 Pourquoi PoEWAL-LT est Mesuré à D=4 ? . . . 47
3.2.5.6 Analyse comparative détaillée . . . 48
3.2.5.7 Module Energest de Contiki . . . 48
3.2.5.8 Conclusion sur la consommation énergétique . . . 49
3.2.6 Distribution des validateurs . . . 49
3.2.6.1 Éléments du Graphique . . . 50
3.2.6.2 Interprétation . . . 50
3.2.6.3 Conclusion sur l'équité . . . 50
3.3 Synthèse et discussions . . . 50
3.3.1 Tableau récapitulatif des performances . . . 51
3.3.2 Comparaison avec les algorithmes de référence . . . 51

3.3.3 Points forts de PoEWAL-LT . . . 52
3.3.4 Défis et difficultés rencontrés . . . 52
3.3.5 Limites de l'algorithme proposé et axes d'amélioration identifiés . . . 53
3.3.6 Perspectives . . . 53
Conclusion . . . 53

Conclusion **55**

Bibliographie **56**

Bibliographie **56**

Webographie **57**

Webographie **58**

Annexe **59**

Table des matières **60**