

# Algorithmes de consensus

## 3.1 Introduction

Après avoir compris les principes de la blockchain et découvert son architecture, nous mettrons l'accent sur le consensus dans la blockchain : son utilité, son mode de fonctionnement ou encore les différentes formes de consensus qui peuvent exister.

L'enjeu n'est pas de rentrer à un niveau de détail technique mais bien d'évaluer et de vulgariser ce processus clé d'une blockchain. Alors, dans ce chapitre après avoir connaître les critères de l'efficacité d'un bon algorithme de consensus blockchain. Nous présentons en détail quelques algorithmes de consensus les plus utilisés dans la blockchain et listons leurs différents avantages et inconvénients.

## 3.2 Critères de l'efficacité d'un mécanisme de consensus

Parvenir à un consensus au sein d'un réseau de nœuds distribués ne semble pas chose aisée. Les algorithmes de consensus doivent [16] :

- Être résilients aux pannes de nœuds, aux retards de transmission, aux messages égarés ou corrompus .
- Faire face aux nœuds malveillants tentant de manipuler le consensus ou de le retarder.

Pour ce faire, une pluralité de mécanismes existent ( PoW, PoS, DPoS, PoET, PoI, etc). Chacun d'entre eux dispose de ses propres caractéristiques en matière de synchronisation, d'émission de message (fréquence, taille), de tolérance aux pannes, de prévention contre les nœuds malveillants, de performance et de sécurité des messages échangés.

Ainsi, Le système blockchain, parvenir à un consensus garantit que l'ensemble des nœuds du réseau s'accordent sur un même état du registre et des données qui y sont stockées. Pour déterminer plus précisément l'efficacité d'un mécanisme de consensus, on évalue ce dernier selon trois critères principaux [16] :

- Terminaison (Liveness) : Tous les noeuds opérationnels participant au consensus doivent finalement produire une valeur.
- Sûreté / Consistance (Safety) : Tous les noeuds opérationnels doivent s'accorder en temps réel sur l'une des valeurs proposées par l'un des noeuds. Cette valeur doit être valide selon les règles définies par le mécanisme.
- Tolérance aux fautes (Fault Tolerance) : Le mécanisme doit être capable de fonctionner même si un ou plusieurs noeuds sont défaillants.

Il est crucial de pouvoir satisfaire les trois propriétés listées ci-dessus si on souhaite résoudre, dans son intégralité, le problème du consensus. Malheureusement, Fischer, Lynch et Patterson, trois chercheurs en informatique, ont montré en 1985 qu'aucun algorithme déterministe de consensus ne permettait de garantir en même temps ces trois propriétés au sein d'un système asynchrone tel qu'un réseau de noeuds distribués (FLP Impossibility).

Ainsi, en règle générale, puisque la tolérance aux fautes est absolument vitale dans le cadre d'un réseau de noeuds distribués, les mécanismes de consensus doivent choisir entre la sûreté et la terminaison en fonction des exigences de l'application pour laquelle a été conçue la plateforme décentralisée [16].

En matière de tolérance aux fautes, les mécanismes de consensus traditionnels opérant dans un réseau de noeuds distribués et connus se sont d'abord évertués à faire face aux fautes "fail-stop" où un noeud ne répond plus à la suite d'un problème matériel ou logiciel[18].

La section suivante, présente un ensemble des algorithmes de consensus, les plus connus dans la littérature, en expliquant leur principe et en mettant l'accent sur leurs avantages et inconvénients.

### 3.3 Algorithmes de consensus

Parmi les algorithmes de consensus existent, on peut citer : preuve de travail, preuve d'enjeu , preuve d'enjeu délégué,ect.Dans cette section,nous détaillant le principe de chaque algorithme.

#### 3.3.1 Preuve de travail (POW : proof of work)

Faisant sa première apparition en 1993, le concept de preuve de travail a été développé pour prévenir les attaques d'altération de service et autres abus de service tels que le spam, sur un réseau en imposant du travail à l'utilisateur du service, généralement en se servant de la puissance de calcul de son ordinateur. [20]

En 2009, Bitcoin a introduit une manière innovante d'utiliser la preuve de travail, comme algorithme de consensus. Dans ce cas, PoW est utilisé pour valider les transactions qui sont regroupées en blocs, qui sont liés entre eux pour former une blockchain. Depuis lors, PoW

s'est propagé pour devenir un algorithme de consensus largement utilisé et est maintenant déployé par de nombreuses crypto-monnaies.[16]

### 3.3.1.1 Principe de fonctionnement

- Les participants de la blockchain (mineurs) doivent résoudre un problème de calcul complexe afin d'ajouter un bloc de transactions dans la blockchain.
- Fondamentalement, la preuve de travail est un concept qui accorde de l'importance à la puissance de calcul des noeuds. Si un noeud dispose de 10 pour 100 de la capacité de calcul totale, alors il parviendra à miner en moyenne 10 pour 100 des nouveaux blocs. Le noeud doit prouver qu'il dispose bien de cette puissance de calcul en faisant travailler son matériel informatique. Cela est fait pour s'assurer que les mineurs mettent de l'argent / des ressources (machines d'extraction) pour faire le travail, ce qui montre qu'ils ne nuiront pas au système de blockchain, car nuire au système entraînera une perte de leur investissement [16].
- La difficulté du problème peut être modifiée lors de l'exécution, pour garantir un temps de blocage constant. Parfois, il y a une situation dans laquelle plus d'un mineur résout le problème simultanément. Dans ce cas, les mineurs choisissent l'une des chaînes et la chaîne la plus longue est considérée comme gagnante. Donc, en supposant que la plupart des mineurs travaillent sur la même chaîne, celle qui se développera le plus rapidement sera la plus longue et la plus fiable.

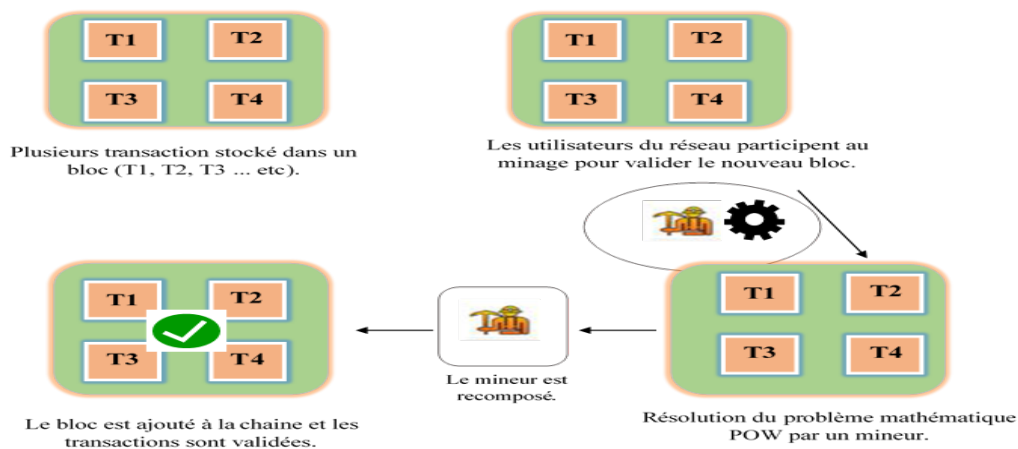


FIGURE 3.1 – POW dans une blockchain[16].

La version de PoW de Bitcoin est basée sur l'utilisation des haches. Son fonctionnement est illustré dans les étapes suivantes :

- Les mineurs sont chargés d'ajouter de nouveaux blocs dans la blockchain. Pour ce faire, les mineurs doivent essayer de deviner un nombre pseudo aléatoire (Nonce).

- Ce nombre, lorsqu'il est combiné avec les données fournies dans le bloc et transmis via une fonction de hachage, doit produire un résultat qui correspond aux conditions données, par exemple, un hachage commençant par quatre zéros.
- Lorsqu'un résultat correspondant est trouvé, les autres noeuds vérifient la validité du résultat et le noeud mineur est récompensé par la récompense de bloc [17] (actuellement "12.5 bitcoins", pour la blockchain de Bitcoins).
- Par conséquent, il est impossible d'ajouter un nouveau bloc dans la chaîne principale sans trouver d'abord un nonce valide, ce qui génère à son tour la solution d'un bloc spécifique (appelé hachage de bloc). Chaque bloc validé contient un hachage de bloc qui représente le travail effectué par le mineur.

Les étapes de l'algorithme de PoW sont illustrées dans le diagramme de figure 3.2 :

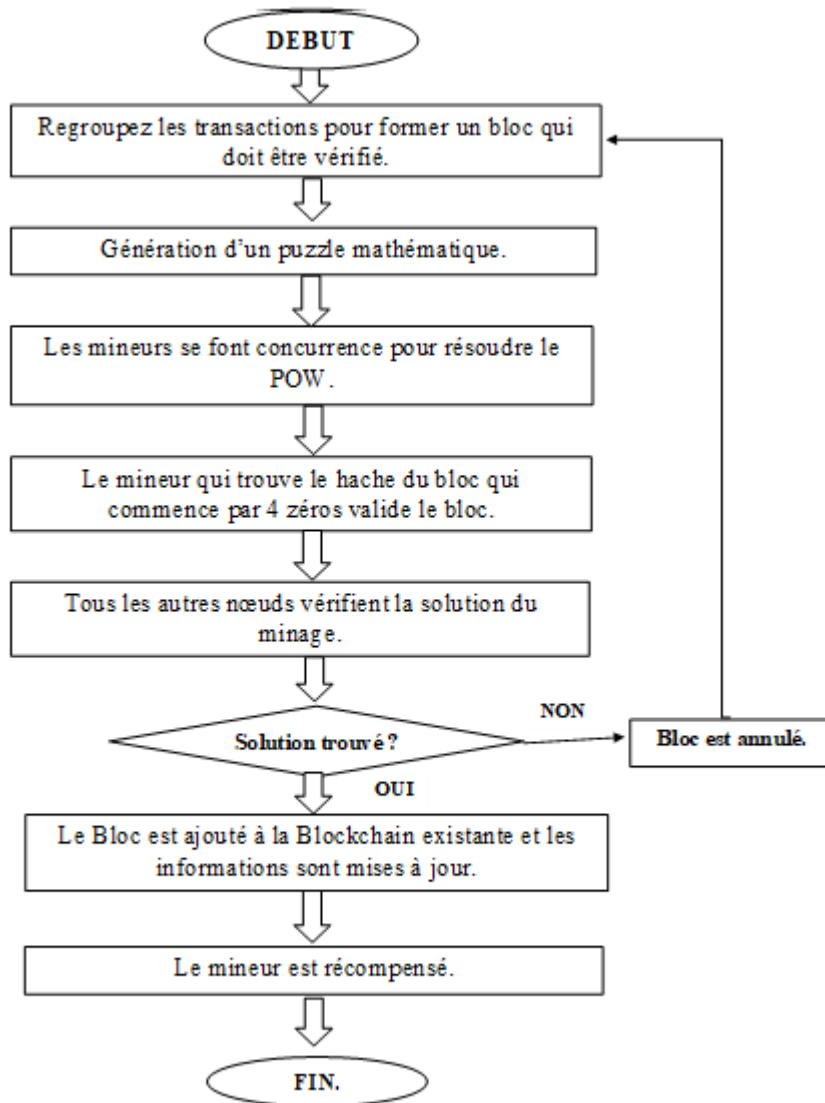


FIGURE 3.2 – Organigramme de l'algorithme PoW.

### 3.3.1.2 Avantages de POW

On peut citer les avantages suivants :

- La preuve de travail aide à protéger le réseau contre de nombreuses attaques différentes. Une attaque réussie nécessiterait beaucoup de puissance de calcul et beaucoup de temps pour faire les calculs et donc elle serait inefficace car le coût encouru serait supérieur aux récompenses potentielles pour attaquer le réseau [15].
- La preuve de travail est sécuritaire, l'historique de la chaîne de blocs est impossible à falsifier [15].
- Il est nécessaire de posséder au moins 51 pour 100 de la puissance de calcul du réseau pour le saturer de fausses transactions [15].

### 3.3.1.3 Inconvénients de POW

Comme inconvénients, on peut signaler les points suivants :

- L'exploitation minière nécessite un matériel informatique coûteux qui consomme une grande quantité d'énergie, même si c'est ce qui garantit la sécurité du réseau, ces calculs d'exploration de données ne peuvent pas être utilisés au-delà. [17]
- Quantité massive de puissance de traitement et d'énergie sous forme d'électricité. [17]
- Coûts élevés associés aux ressources matérielles. [17]
- Temps de latence nécessaire pour valider une transaction. [17]
- Le gain décroissant des mineurs augmenterait le risque de corruption du réseau, puisque la fiabilité de la chaîne de blocs nécessite qu'aucun opérateur hostile ne détienne plus de la moitié de la puissance de calcul de la chaîne (attaque de type 51 pour 100) [15].

## 3.3.2 Preuve d'enjeu (POS : proof of stake)

La preuve d'enjeu a été créée comme une alternative à la preuve de travail (PoW), afin de résoudre les problèmes inhérents à cette dernière. Bien qu'ils partagent tous deux le même objectif de parvenir à un consensus dans la blockchain, le processus pour atteindre l'objectif est assez différent.

- La preuve d'enjeu, elle est directement liée à la monnaie puisqu'elle lie la capacité à valider un bloc à la quantité d'argent (ou jeton) détenue. Sans crypto-monnaie, il n'y a donc pas de mécanisme de consensus, ce qui est le fondement cardinal d'une blockchain [17].

- Cette approche pour la gestion du réseau, ne reposant pas sur la surconsommation d'énergie. Cette fois, le système accorde de l'importance à la part des jetons électroniques qu'un individu possède par rapport au nombre total en circulation dans le réseau.
- Si un individu possède 10 pour 100 du nombre total de jetons, il pourra miner en moyenne 10 pour 100 des blocs générés [17].
- Le concept sous-jacent est que si un individu est impliqué de façon notable dans une Blockchain (comprendre qu'il possède une partie importante des jetons, et a donc un intérêt financier considérable) il sera particulièrement sensible au bon fonctionnement de cette dernière.

Les étapes de l'algorithme de PoS sont illustrées dans le diagramme de figure 3.3 :

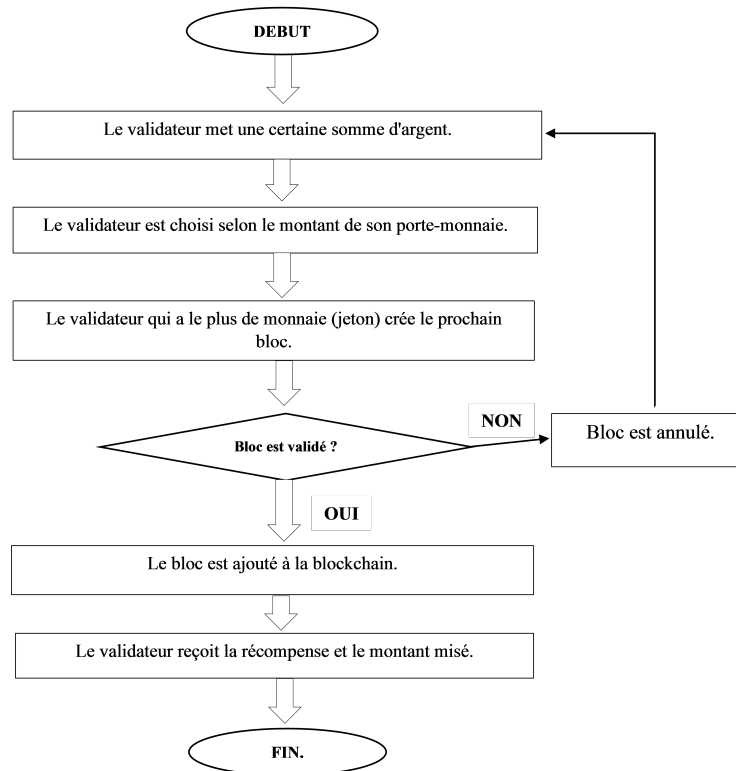


FIGURE 3.3 – Organigramme de l'algorithme PoS.

### 3.3.2.1 Avantages de POS

On peut citer les points suivants comme avantages :

- Le principal avantage de PoS est que la validation d'un bloc ne repose pas sur de puissants calculs algorithmiques qui consommeraient énormément d'énergie et auraient par conséquent un impact négatif sur l'environnement.

- La preuve d'enjeu est beaucoup plus rentable que les autres méthodes et n'entraîne pas de fuite de valeur vers des monnaies fiduciaires.
- La fidélisation de la détention et de la thésaurisation des jetons. En effet, les investisseurs (mineurs virtuels, ou forgeurs) ont un plus grand intérêt à conserver leur récompense de leur preuve d'enjeu plutôt qu'à immédiatement la revendre au prix du marché. Les mineurs virtuels (PoS) auront plutôt tendance à conserver leur récompense pour maximiser leurs chances futures d'être sélectionné à la validation des prochains blocs et percevoir ainsi de nouvelles récompenses. [17]
- La stabilité de fonctionnement de réseau avec PoS est gardée comme il est présenté dans la figure 3.1. Plus les mineurs rejoignent le réseau, plus la création des blocs augmente. Automatiquement le temps moyen de création diminue et ce qui augmente la difficulté du minage. Dans ce moment les validateurs des blocs quittent le réseau, cela implique une diminution de création de bloc. Enfin, le temps moyen de la création de bloc devient à la normale.

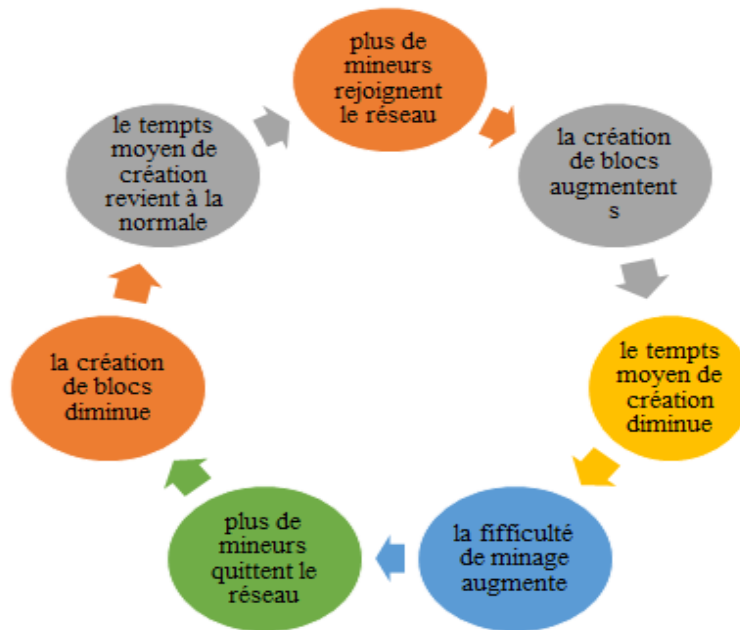


FIGURE 3.4 – Schéma des avantages de POS.

### 3.3.2.2 Inconvénient de POS

Malgré ses avantages, nous pouvons trouver les problèmes suivants :

- Favoriser les personnes ayant le plus de jeton favorise la thésaurisation. C'est à dire amasser de l'argent sans le dépenser. Stocker des jetons pour miner va donc limiter les échanges de jeton et donc nuire au développement de la crypto monnaie. [18]

- Monopolisation de la richesse, en effet plus on a de jeton plus l'on peut forger et plus on est reçoit de l'argent. Les riches restent riches.

### 3.3.3 Preuve d'enjeu Délégué (DPOS : delegated proof of stake)

L'algorithme de consensus preuve d'enjeu délégué (DPoS) a été développé par Daniel Larimer, en 2014. Bitshares, Steem, Ark et Lisk sont quelques-uns des projets de crypto-monnaie qui utilisent l'algorithme de consensus DPoS.[17]

- Une blockchain basée sur DPoS compte les parties prenantes sous-traitent leur travail à un tiers. En d'autres termes, ils peuvent voter pour quelques délégués qui sécuriseront le réseau en leur nom. Les délégués peuvent également être appelés témoins et ils sont chargés de parvenir à un consensus lors de la génération et de la validation de nouveaux blocs.
- Le droit de vote est proportionnel au nombre de pièce détenues par chaque utilisateur. Le système de vote varie d'un projet à l'autre, mais en général, chaque délégué présente une proposition individuelle lorsqu'il demande des votes.
- Habituellement, les récompenses recueillies par les délégués sont partagées proportionnellement avec leurs électeurs respectifs. Par conséquent, l'algorithme DPoS crée un système de vote qui dépend directement de la réputation des délégués.
- Si un noeud élu se comporte mal ou ne fonctionne pas efficacement, il sera rapidement expulsé et remplacé par un autre.
- En ce qui concerne les performances, les chaînes de blocs DPoS sont plus évolutives, pouvant traiter plus de transactions par seconde, par rapport à PoW et PoS.

Les rôles des délégués sont :

- S'assurer que leur noeud est toujours opérationnel.
- Collecte des transactions sur le réseau.
- Signer et diffuser ces blocs, valider les transactions.
- S'il y a des problèmes de consensus, le département de la sûreté et de la sécurité permet de les résoudre d'une manière juste et démocratique.

Les délégués n'ont pas le pouvoir de modifier les détails d'une transaction. Toutefois, en tant que validateurs, ils pourraient théoriquement exclure certaines transactions d'un bloc. Néanmoins, cela n'a que très peu d'effet puisque le prochain bloc créé inclura ces transactions, ce qui donnera au prochain délégué les frais associés à leur validation.



Ainsi, les transactions ne seront que légèrement retardées. De plus, cela conduirait inévitablement à ce que le délégué malhonnête soit éliminé par le reste du réseau. Essentiellement, un réseau DPoS est autogéré et contrôlé par tous ses participants, en veillant à ce que les meilleurs intérêts du réseau demeurent la priorité.

### 3.3.3.1 Avantages de DPoS

On cite comme avantages :

- Efficacité d'un point de vue énergétique, et plus rapide qu'un mécanisme de POW et POS, il permet de valider les transactions en quelques secondes seulement.
- Réduit les interactions entre les noeuds et permet dans un plus grand nombre de transactions et des validation plus rapides.

### 3.3.3.2 Inconvénients de DPoS

Le consensus de DPoS a cependant lui aussi ses limites qui sont :

- Risque de centralisation excessive avec des risques de vulnérabilité.
- Le DPoS sacrifie la décentralisation pour favoriser la vitesse de validation des transactions du réseau.[19]
- Les participants possédant le plus de jetons peuvent voter pour eux-mêmes afin de s'élire vérificateur des transactions
- Un système ploutocratique, encourageant toujours l'enrichissement des plus riches : le vote des participants a plus de poids selon l'importance de leur portefeuille, et les délégués pouvant se permettre de redistribuer une grande partie de leurs gains auront toujours plus de chances d'être élus.

### 3.3.4 Preuve de temps écoulé (PoET : Proof of Elapsed Time)

Le concept du consensus POET a été inventé début 2016 par Intel , le célèbre géant de la fabrication de puces. Il offre un outil de haute technologie prêt l'emploi pour résoudre le problème informatique de l'élection aléatoire .[17]

- PoET est un algorithme de mécanisme de consensus qui est utilisé sur des réseaux blockchains, autorisés pour décider des droits d'extraction ou des gagnants de blocs sur le réseau. Les réseaux blockchains autorises sont ceux qui exigent que tout participant potentiel s'identifie avant d'être autorisé à se joindre.
- Basé sur le principe d'un système de loterie équitable où chaque noeud est également susceptible d'être gagnant, le mécanisme PoET est basé sur la répartition des chances de gagner équitablement parmi le plus grand nombre possible de participants au réseau.

- Le fonctionnement de l'algorithme PoET est le suivant.
  - Chaque noeud du réseau de chaînes de blocs génère un temps d'attente aléatoire et se met en veille pendant cette durée spécifiée.
  - Celui qui se réveille en premier c'est-à-dire celui qui a le temps d'attente le plus court se réveille et valide un nouveau bloc dans la chaîne de blocs, diffusant les informations nécessaires à l'ensemble du réseau de pairs.
  - Le même processus se répète ensuite pour la découverte du bloc suivant.
- Le mécanisme de consensus du réseau PoET doit garantir deux facteurs importants.
  - Premièrement, que les noeuds participants sélectionnent véritablement un temps qui est en effet aléatoire et non une durée plus courte choisie exprès par les participants pour gagner,
  - Deuxièmement, le gagnant a en effet terminé le temps d'attente.
- Essentiellement, le flux de travail est similaire au mécanisme de consensus suivi par l'algorithme de preuve de travail (POW) de Bitcoin, mais sans sa consommation d'énergie élevée. Au lieu d'être gourmande en ressources, elle permet au processeur d'un mineur de dormir et de passer à d'autres tâches pendant la durée spécifiée, augmentant ainsi son efficacité.
- Le mécanisme d'exécution du code de confiance dans un environnement sécurisé prend également en charge de nombreuses autres nécessités du réseau. Il garantit que le code de confiance s'exécute en effet dans l'environnement sécurisé et n'est modifiable par aucun participant externe. Il garantit également que les résultats sont vérifiables par les participants et entités externes, améliorant ainsi la transparence du consensus du réseau [20]

#### 3.3.4.1 Avantage du POET

- Le faible coût de la participation augmente la probabilité que la population de validateurs soit importante, augmentant la robustesse de l'algorithme de consensus.
- Plus de personnes peuvent participer facilement, donc décentralisé.
- Il est simple pour tous les participants de vérifier que le mineur a été légitimement sélectionné.
- Le coût du contrôle du processus d'élection des chefs est proportionnel à la valeur obtenue de ce dernier. [21]

### 3.3.4.2 Inconvénient du POET

- Même si ça marche, l'utilisateur doit avoir un matériel spécialisé. Il ne peut donc pas être adopté en masse.
- Ne convient pas aux blockchains publiques.

### 3.3.5 Preuve d'importance (POI : proof of importance)

La preuve d'importance est un algorithme de consensus blockchain qui prend en compte la productivité globale des utilisateurs du réseau. Il a d'abord été utilisé par NEM (New Economy Movement), une société de technologie blockchain visant à traiter les transactions plus efficacement et à introduire la réputation du crypto-système. [17]

- Cet algorithme est conçu pour récompenser les utilisateurs très fidèles de la blockchain. Par conséquent, encourager une plus grande utilisation de la plateforme.
- C'est un algorithme qui dépend de combien d'utilisateurs actifs sur le réseau. Plus ils sont actifs, plus ils reçoivent de récompenses.
- Chaque utilisateur est noté, et plus son score est élevé, plus les récompenses sont importantes.
- Le nombre de pièces peut faire balancer les votes naturellement étant donné que les transactions d'un montant élevé peuvent être admissibles à davantage de transactions. Cependant, l'algorithme dépend principalement des activités surveillées par chaque utilisateur et pas nécessairement du montant qu'ils effectuent.[22]

#### 3.3.5.1 Avantages de POI

Les riches peuvent ne pas continuer à s'enrichir injustement sur la plateforme car le montant d'argent en possession d'un individu n'est pas le seul facteur à prendre en compte lors de la mesure de la réputation d'un compte.

#### 3.3.5.2 Inconvénients du POI

Un problème brûlant avec cette méthode est l'utilisation de transactions factice qui auraient récompensé les gens pour avoir envoyé des transactions de va-et-vient pour tromper l'algorithme. L'utilisation de transactions factices est un problème que NEM (New Economy Movement : un groupe d'organisations qui tentent de restructurer le système économique) et d'autres acteurs majeurs doivent encore épuiser.

## 3.4 Conclusion

Comme nous avons vu, aucun des algorithmes de consensus même en utilisant une énorme quantité de ressources, ne fournit pas une solution parfaite aux exigences de mécanisme de consensus dans une blockchain. Chacun d'eux a les faiblesses qui mettent le réseau en danger. Cependant, il vaut la peine d'envisager des algorithmes de consensus uniquement comme outils pour assurer la stabilité du réseau. Enfin, à partir de tous consensus discutés dans ce chapitre, il paraît que la preuve de travail fournit aux réseaux la plus grande fiabilité car elle empêche les attaques à longue portée. Les performances de cet algorithme et l'algorithme de PoS sont traités, et évalués dans le prochain chapitre.

# Comparaison des algorithmes de consensus

## 4.1 Introduction

L'objectif d'un mécanisme de consensus dans une Blockchain est de laisser les membres du réseau s'accorder sur l'état actuel de l'historique des transactions en l'absence d'un organe centralisé chargé de mettre à jour ce registre .

Comme nous avons déjà vu dans le chapitre précédent, il existe un ensemble des algorithmes de consensus, qui sont variés en termes de performance (logicielle et matérielle) et de sécurité. Ces critères sont utilisés, dans notre projet, pour comparer deux algorithmes de consensus discutés dans ce présent chapitre. À cet égard, nous avons implémenté deux algorithmes de consensus preuve de travail et preuve d'enjeu, avec l'utilisation de deux fonctions de hachage, le SHA256 et le MD5.

Dans ce chapitre, nous présentons les différents moyens utilisés pour réaliser ce projet et nous présentons les différents moyens et outils. Nous commençons tout d'abord avec l'idée générale de notre projet , puis nous distinguons l'environnement de programmation et les outils utilisés. Enfin nous discutons sur le processus complet de notre système Blockchain et ses algorithmes pour analyser les résultats de la simulation obtenus.

## 4.2 Idée de travail

Dans notre travail, nous comparons deux algorithmes de consensus, preuve de travail et preuve d'enjeu, en utilisant deux machines avec des capacités différentes . Nous évaluons les deux algorithmes en termes de temps de hachage et de minage chez les utilisateurs de blockchain .

L'idée de travail est basée sur une comparaison de l'ampleur de l'effet de deux algorithmes de consensus : preuve de travail et preuve d'enjeu, avec l'utilisation de deux fonctions de hachage, le SHA256 et le MD5.

Nous avons adapté les algorithmes comme suivant :

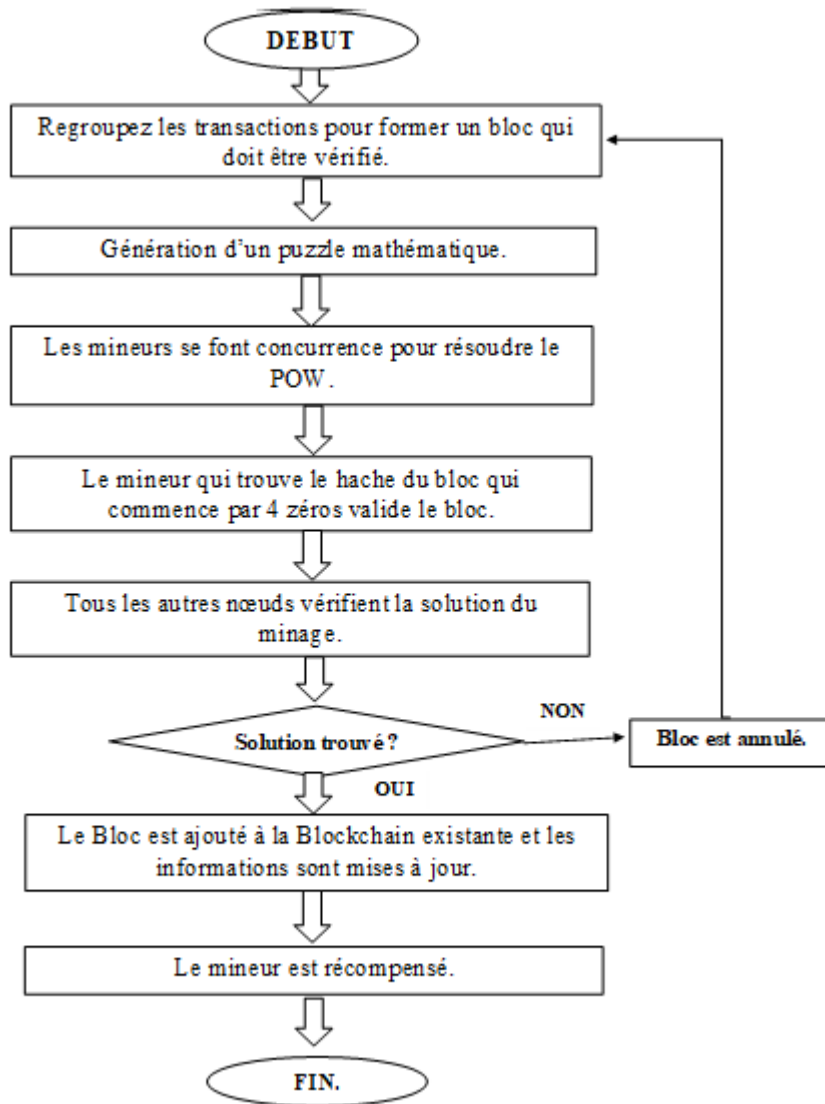


FIGURE 4.1 – Organigramme de l'algorithme PoW.

Pour l'algorithme de preuve de travail : (voir la figure 4.1)

- Nous avons adapté l'algorithme de preuve de travail de Bitcoin, dont la preuve de travail consiste à la détermination du nonce qui, concaténé avec les autres composants de l'en-tête du bloc, permettra d'aboutir un hache commençant par un nombre minimum de zéros (i.e. inférieur à un seuil). Le pseudo-code de PoW de Bitcoin peut être exprimée comme suit (voir la figure 4.2) :

```

Tant que (valeur du hash) >= SEUIL :
  1. Calculer le hash de l'en-tête du bloc
  2. Si (valeur du hash) < SEUIL, retourner le hash
  (bloc miné)
  Sinon, incrémenter le nonce de 1
  
```

FIGURE 4.2 – Pseudo-Code de PoW Bitcoin.

- Le premier noeud parvenant à exhiber un hash gagnant acquiert ainsi le droit d'ajouter le bloc afférent à la chaîne et de réclamer la récompense de minage
- Dans notre solution, nous avons utilisé deux fonctions de hachage : SHA256 et MD5.

Pour l'algorithme de preuve d'enjeu PoS : nous avons utilisé le fonctionnement illustré dans la figure 4.3 .

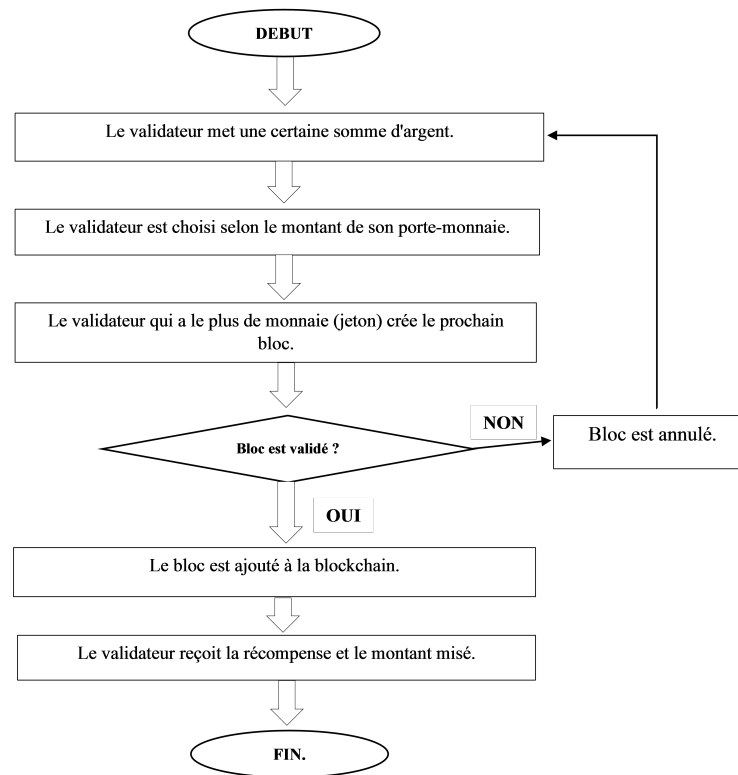


FIGURE 4.3 – Organigramme de l'algorithme PoS.

## 4.3 Environnement de développement

### 4.3.1 Matériel

Nous avons utilisé deux machines avec les caractéristiques différentes qui sont présentées dans le tableau (voir le tableau 4.1).

Caractéristiques	PC 1	PC 2
Processeur	1.70 GHz Intel® core™i3-CPU	2.13 GHz Intel® pentium®
RAM	4.00 GO	3.00 GO
Carte graphique	Intel® HD Graphics 4005	Intel® Graphics
Système d'exploitation	Windows 10 (64 bits)	Windows 7 (64 bits)

TABLE 4.1 – Caractéristiques des machines utilisés.

### 4.3.2 Logiciel

Il existe plusieurs outils et langages de programmation pour implémenter ce projet. Nous avons construit notre application en utilisant les outils suivants :

- Python : Python est un langage de programmation de haut niveau interprété et orienté objet avec une sémantique dynamique. Il est très sollicité par une large communauté de développeurs et de programmeurs. Python est un langage simple, facile à apprendre et permet une bonne réduction du cout de la maintenance des codes. Les bibliothèques (packages) python encouragent la modularité et la réutilisabilité des codes. Python et ses bibliothèques sont disponibles (en source ou en binaires) sans charges pour la majorité des plateformes et peuvent être redistribués gratuitement
- Python Flask : est un micro framework open-source de développement web en Python. Il est classé comme micro framework car il est très léger.
- HTML et CSS : Diverses options sont disponibles pour développer des Interfaces Graphiques Utilisateurs (GUI). Dans notre projet, nous avons utilisé HTML et CSS à cause de sa rapidité et facilité de créer des applications graphiques
- JavaScript (AJAX ) : utilisant pour envoyer et récupérer des données vers et depuis un serveur de facon asynchrone,
- Pycharm : Vu que PyCharm est l'API Python est la plus complète, nous avons choisi de travailler avec. C'est un bon environnement de développement dédié au langage Python.



## 4.4 Solutions proposées

Nous avons implanté une Blockchain et simulé son fonctionnement sur localhost et sur un réseau de deux ordinateurs.

### 4.4.1 Blockchain implantée

Les structures des transactions et blocks adaptées sont présentées dans les lignes suivantes.

#### 4.4.1.1 Structure de transaction

Chaque transaction contient les champs suivants :

- Clé prive et clé publique de l'envoyeur qui sont générées lors de la création du porte-monnaie.
- Clé publique de destinataire.
- Montant a envoyé.

#### 4.4.1.2 Structure de bloc

Le bloc dans la Blockchain implantée ayant la structure suivante :

- Nonce : est le nombre de cycle pour que le mineur obtient le hachage généré. La solution est la génération d'un hache qui commence par 2 zéro.
- Hash (hachage) : Le hachage des données actuel.
- Temps : l'horodatage de validation du bloc (date et heure).
- Minage : le hash généré commence par 2 zéros (au lieu de 4 zéro pour minimiser un peu le temps d'exécution ).



FIGURE 4.4 – Exemple d’un bloc.

## 4.4.2 Localhost

Nous avons testé notre solution en premier lieu localement sur une seule machine. Sur le local host, nous avons utilisé 5 Noeud, en changeant à chaque fois le numéro de port. Par exemple :

- Adresse du 1<sup>er</sup> noeud : `http ://127.0.0.1 :5001/ port :5001`
- Adresse du 2<sup>ème</sup> noeud : `http ://127.0.0.1 :5002/ port :5002`

## 4.4.3 Réseau câblé

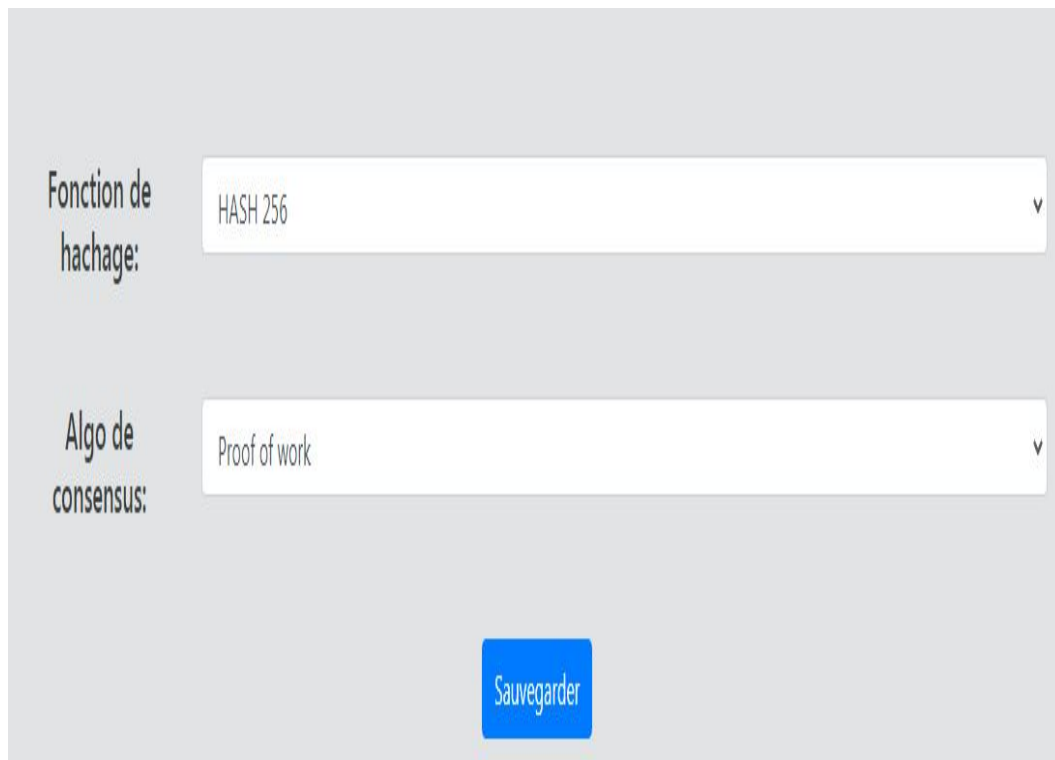
Le principe du réseau est presque le même du localhost, juste que chaque noeud a une adresse IP différente c’est-à-dire chaque utilisateur de réseau participe avec sa propre machine. Ce réseau est câblé via un fil Ethernet qui nous facilite la tâche de la transmission des données (transactions).

### 4.4.3.1 Quelques fenêtres de notre application

Notre application contient plusieurs fenêtres ( hash, block, transaction, Blockchain, paramètres ,ect) dans chaque noeud permis ces fenêtres nous présentent les plus importants dans les lignes suivants :

#### 4.4.3.1.1 Paramètres

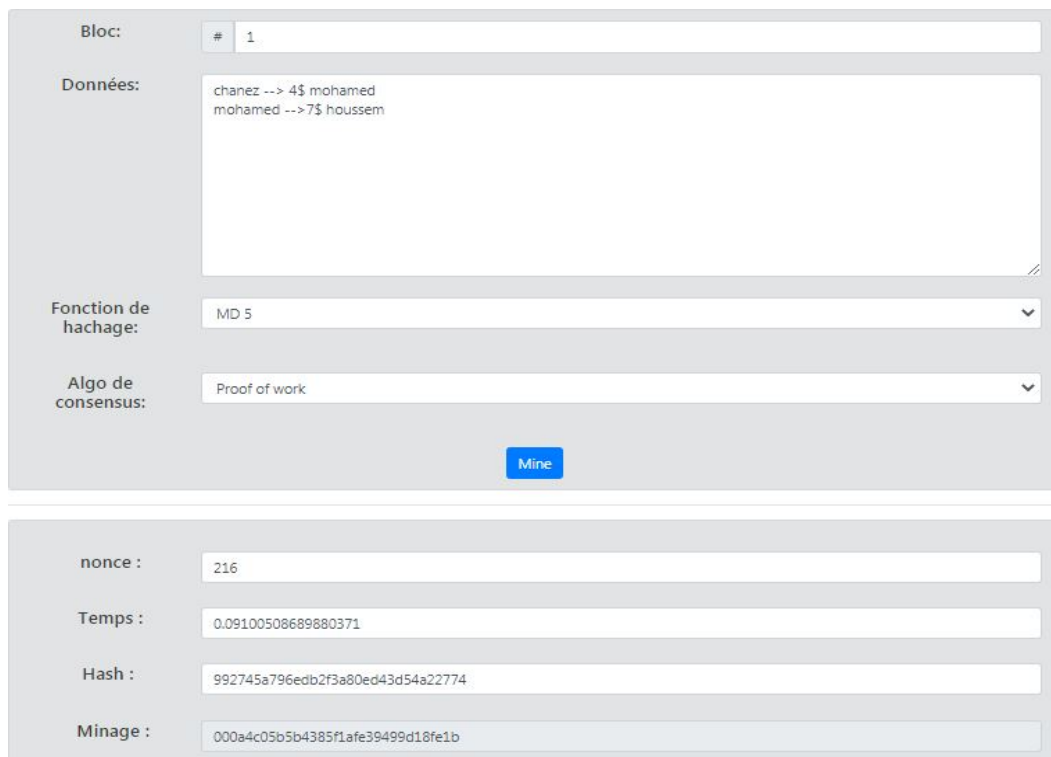
- Cette fenêtre sert à sélectionner l’algorithme de consensus qu’on veut appliquer sur le réseau Blockchain.
- Si le hachagé impliqué (POW ou POS et HASH 256 ou MD5 ).



The image shows a web interface for configuring mining parameters. It features two dropdown menus: 'Fonction de hachage:' (Hashing function) set to 'HASH 256' and 'Algo de consensus:' (Consensus algorithm) set to 'Proof of work'. A blue 'Sauvegarder' (Save) button is positioned at the bottom center.

FIGURE 4.5 – Fenêtre paramètre.

**4.4.3.1.2 Minage d'un bloc** Exemple minage d'un bloc de données avec PoW en fonction de Hash MD5.



The image displays a mining interface with two main sections. The top section is for inputting data: 'Bloc:' (Block) is set to '# 1', and 'Données:' (Data) contains 'chanez --> 4\$ mohamed' and 'mohamed --> 7\$ houssem'. The 'Fonction de hachage:' (Hashing function) is set to 'MD 5' and 'Algo de consensus:' (Consensus algorithm) is set to 'Proof of work'. A blue 'Mine' button is at the bottom. The bottom section shows the results: 'nonce : 216', 'Temps : 0.09100508689880371', 'Hash : 992745a796edb2f3a80ed43d54a22774', and 'Minage : 000a4c05b5b4385f1afe39499d18fe1b'.

FIGURE 4.6 – Fenêtre bloc.

#### 4.4.3.1.3 Création du porte-monnaie

- Sur cette page, nous devons créer un porte-monnaie (wallets) pour stocker de l'argent afin de se permettre d'effectuer un paiement (ou une transaction).
- La création d'un porte-monnaie consiste à affecter une clé privée masquée et une clé publique que tout le monde peut la voir.

créer un porte monnaie argent: \$ 10

porte monnaie créé

La clé publique d'expéditeur

30819f300d06092a864886f70d010101050003818d0030818902818100b2c1fd83065a816007318365bea6eb9838470b55ffb96575309039e528f291fdede525

Clé publique du récepteur

Entrer la clé

FIGURE 4.7 – Création d'un porte-monnaie.

**4.4.3.1.4 Transaction** Pour générer une transaction et l'envoyer, il faut avoir :

- La clé privée de l'expéditeur. (Invisible).
- La clé publique de l'expéditeur
- La clé publique du récepteur de monnaie.
- Le montant de l'argent à envoyer.

créer un porte monnaie

argent: \$ 10

porte monnaie créé

La clé publique d'expéditeur

30819f300d06092a864886f70d010101050003818d0030818902818100b2c1fd83065a816007318365bea6eb9838470b55ffb96575309039e528f291fdede525

Clé publique du recepneur

30819f300d06092a864886f70d010101050003818d00308189028181009ddcddf405900035eb38810ba789616b9a61ea0f9d58df0bc0b668e23bcd3e0fef8e27

La somme d'argent

2.5

possible (e.g. 1.50)

Envoyer

FIGURE 4.8 – Fenêtre transaction.

#### 4.4.3.1.5 Fenêtre Blockchain

- Liste des transactions : tableau de transactions et le Bouton Mine (seront affichés chez le mineur).
- Bouton Mise à Jour Blockchain : mise à jour de la liste des Blocs.
- Liste des Blocs : Chaque bloc affiche ses propres informations.
- Bouton Voir : affiche la liste de transaction de ce Bloc.

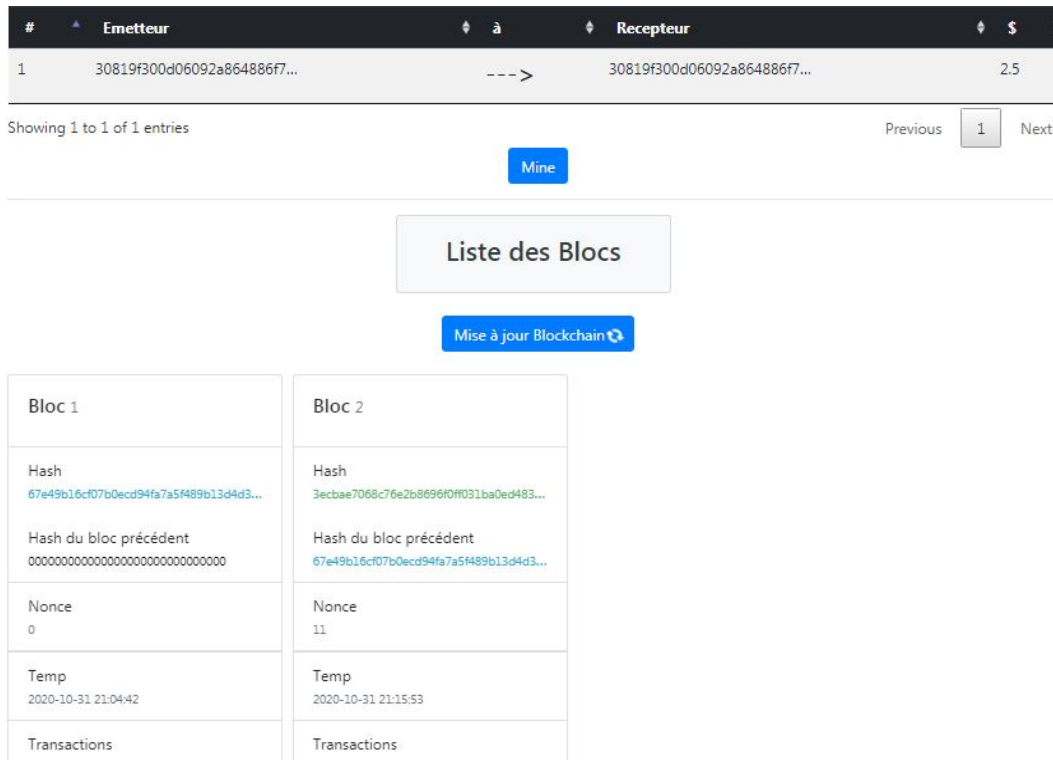


FIGURE 4.9 – Fenêtre Blockchain.

## 4.5 Résultats expérimentaux

Dans cette partie du chapitre, nous analysons les performances d'algorithmes de consensus.

Nous avons utilisé deux machines de différente capacité, pour tester deux algorithmes de consensus (PoW et PoS), tout en utilisant deux fonctions de hachage (SHA256 et MD5). Les résultats sont présentés et discutés dans les lignes suivantes.

### 4.5.1 Pour PC1

#### 4.5.1.1 PoW et PoS avec SHA256 pour PC1

Dans ce cas on a appliqué les algorithmes de consensus PoW puis PoS avec la fonction de hachage SHA256, dont le PC1 est mineur. Les résultats sont présentés dans le (tableau 4.3), en termes de temps de hachage et temps de minage.

PoW /SHA256			PoS /SHA256		
Blocs\nbrs de transaction	Temps de hachage (ms)	Temps de minage (ms)	Blocs\nbrs de transaction	Temps de hachage (ms)	Temps de minage (ms)
Bloc 1 (1transaction)	1.144	288.629	Bloc 12 (1transaction)	1.19	5.21
Bloc 3 (2transaction)	2.228	505.506	Bloc 13 (2transaction)	1.99	5.41
Bloc 4 (1transaction)	1.273	88.26	Bloc 14 (1transaction)	1.22	4.57
Bloc 8 (4transaction)	3.39	364.69	Bloc 16 (1transaction)	2.34	12.87

TABLE 4.2 – Résultat du PoW /SHA256 et PoS/ SHA256 du pc1.

#### 4.5.1.2 PoW et PoS avec MD5 pour PC1

Les résultats d'application des algorithmes de consensus PoW et PoS avec la fonction de hachage MD5 dont le pc1 est mineur sont présentés dans le tableau 4.4, en terme de temps de hachage et temps de minage.

PoW /MD5			PoS /MD5		
Blocs\nbrs de transaction	Temps de hachage (ms)	Temps de minage (ms)	Blocs\nbrs de transaction	Temps de hachage (ms)	Temps de minage (ms)
Bloc 1 (1transaction)	0.004	2.16	Bloc 12 (1transaction)	0.003	2.21
Bloc 3 (2transaction)	0.007	2.65	Bloc 13 (2transaction)	0.05	2.25
Bloc 4 (1transaction)	0.003	3.51	Bloc 14 (1transaction)	0.006	2.15
Bloc 8 (4transaction)	0.012	5.46	Bloc 16 (1transaction)	0.013	4.18

TABLE 4.3 – résultat du PoW /MD5 et PoS/ MD5 du pc1.

### 4.5.2 Pour PC2

#### 4.5.2.1 PoW et PoS avec SHA256 pour PC2

Dans ce cas on a appliqué les algorithmes de consensus PoW puis PoS avec la fonction de hachage SHA256, dont le PC2 est mineur. Les résultats sont présentés dans le tableau 4.5, en termes de temps de hachage et temps de minage.

PoW /SHA256			PoS /SHA256		
Blocs\nbrs de transaction	Temps de hachage (ms)	Temps de minage (ms)	Blocs\nbrs de transaction	Temps de hachage (ms)	Temps de minage (ms)
Bloc 1 (1transaction)	1.89	366.35	Bloc 12 (1transaction)	2.03	6.9
Bloc 3 (2transaction)	3.099	940.494	Bloc 13 (2transaction)	3.62	8.23
Bloc 4 (1transaction)	2.14	166.73	Bloc 14 (1transaction)	2.46	7.8
Bloc 8 (4transaction)	6.12	1045.21	Bloc 16 (1transaction)	6.84	10.32

TABLE 4.4 – Résultat du PoW /SHA256 et PoS/ SHA256 du pc2.

#### 4.5.2.2 PoW et PoS avec MD5 Pour PC2

Les résultats d'application des algorithmes de consensus PoW et PoS avec la fonction de hachage MD5 dont le pc2 est mineur sont présentés dans le tableau 4.6 , en terme de temps de hachage et temps de minage.

PoW /MD5			PoS /MD5		
Blocs\nbrs de transaction	Temps de hachage (ms)	Temps de minage (ms)	Blocs\nbrs de transaction	Temps de hachage (ms)	Temps de minage (ms)
Bloc 2 (1transaction)	0.009	2.89	Bloc 12 (1transaction)	0.01	2.13
Bloc 5 (2transaction)	0.001	3.56	Bloc 13 (2transaction)	0.01	2.19
Bloc 6 (1transaction)	0.009	2.25	Bloc 14 (1transaction)	0.09	2.13
Bloc 8 (4transaction)	0.01	5.75	Bloc 16 (4transaction)	0.01	3.12

TABLE 4.5 – Résultat du PoW /MD5 et PoS/ MD5 du pc2.

## 4.6 Analyse et discussion du résultat

Comme les résultats montrent :

- Le temps de minage dans la preuve de travail est plus élevé par rapport à la preuve d'enjeu.



- Le temps de hachage de SHA256 est beaucoup plus lent que MD5.

Ces deux temps (minage et hachage) dépendent de nombre de transactions, fonction de hachage et performance de la machine.

Par exemple, lorsque on applique un consensus avec les SHA256, le hachage et le minage prend assez de temps à cause de sa complexité. Par contre le MD5 est plus rapide, dans ce cas on déduit que le mode de fonctionnement de l'algorithme de hachage est très important au niveau de rapidité.

Par ailleurs, on remarque la différence des résultats entre les deux machines est très claire, ce qui indique que la performance et la capacité de la machine joue un rôle très important en termes de temps et de la consommation d'électricité.

D'après les tableaux de teste, on remarque que l'algorithme de consensus de la preuve de travail est lent par rapport à la preuve d'enjeu en minage peu importe la fonction de hachage utilisé car la preuve de travail repose sur la puissance de calcul, à partir de là on déduit que la preuve d'enjeu est meilleure que la preuve de travail soit au niveau de temps ou la consommation d'énergie.

## 4.7 Conclusion

Dans ce chapitre nous avons abordé la partie implémentation de notre application de l'analyse des algorithmes de consensus dans la Blockchain. Nous avons commencé par présenter le réseau Blockchain qui a été utilisé pour générer et tester et évaluer les performances des algorithmes. Nous avons également présenté les différences prétraitements et les démarches réalisé pour effectuer des transactions et réalisé la chaine des blocs ainsi les métriques utilisées pour mesurer les performances des consensus. Enfin, nous avons terminé par la simulation et par la discussion des résultats obtenus.