UNIVERSITÉ DE LORRAINE | BIBLIOTHÈQUES UNIVERSITAIRES

AVERTISSEMENT
DROIT D'AUTEUR

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document. Toute contrefaçon, plagiat ou reproduction illicite encourt une poursuite pénale.

Contact bibliothèque : ddoc-theses-contact@univ-lorraine.fr
(Cette adresse ne permet pas de contacter les auteurs)

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

UNIVERSITÉ DE LORRAINE
34 COURS LÉOPOLD — CS 25233
54052 NANCY CEDEX
TÉL : 03 54 50 54 00
CONTACT@UNIV-LORRAINE.FR
WWW.UNIV-LORRAINE.FR

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Dynamiques de la sélection et des comportements des validateurs de blockchains tolérantes aux byzantins

THÈSE

soutenance prévue le 18 décembre 2025

pour l'obtention du

Doctorat de l'Université de Lorraine

(Spécialité Automatique, Traitement du Signal et des images, Génie Informatique)

par

Rachid Guedjali

Composition du jury

Président :	BADONNEL Rémi	Professeur, Université de Lorraine
Rapporteurs :	RACHEDI Abderrazak	Professeur, Université Gustave Eiffel
	SOUIHI Sami	Chercheur CEA-LIST (Dept DILS, Labo LICIA)
Examinateurs :	CHINNICI Marta	*Senior researcher*, ENEA C.R. Casaccia, Italie
	GEORGES Jean-Philippe	Professeur, Université de Lorraine (Directeur de thèse)
	KUBLER Sylvain	*Research scientist*, Université du Luxembourg (Co-directeur)
Invité :	LEDUC Guilain	*Research & development specialist*, Université du Luxembourg

CRAN — UMR 7039 CNRS UL

Mis en page avec la classe thesul.

Table des matières

Introduction générale **1**

Chapitre 1 Fondements techniques de la blockchain **4**

1.1 Introduction 4

1.2 Évolution historique et contexte d'émergence 5

1.2.1 Précurseurs cryptographiques et concepts fondateurs 5

1.2.2 De Bit Gold à Bitcoin : la quête de la monnaie numérique décentralisée . 5

1.2.3 Évolutions post-Bitcoin et diversification des paradigmes 6

1.3 Du système distribué à blockchain 6

1.3.1 Système distribué 6

1.3.2 Propriétés fondamentales : Safety et Liveness 7

1.3.3 Centralisation et décentralisation 8

1.3.4 Réplication de machines à états (SMR) 9

1.3.5 Blockchain et consensus 10

1.3.6 Fiabilité des Transactions : ACID et CAP 11

1.3.7 Modèles temporels 12

1.4 Architecture fondamentale de la blockchain 13

1.4.1 Principes cryptographiques fondamentaux 13

1.4.2 Structure interne d'un bloc 15

1.4.3 Processus de construction et de validation d'un bloc 16

1.4.4 Le Concept de Cross-Chain et l'Interopérabilité 16

1.5 Topologies des réseaux blockchain et leurs implications 19

1.5.1 Classification par modèle d'accès 19

1.5.2 Implications sur la sécurité et la performance 20

1.6 Conclusion 21

Chapitre 2 Consensus BFT : sélection des validateurs et dynamiques de vote **22**

2.1 Introduction 22

2.2 Consensus à preuve (Proof-based) 23

2.2.1 Proof of Work 23
2.2.2 Proof of Stake 23
2.2.3 Proof of Reputation : La preuve par le capital de confiance 24
2.3 Consensus à vote (Vote-based) 24
2.3.1 Le problème des généraux byzantins : 25
2.3.2 Garanties théoriques (sécurité, vivacité, finalité) : 25
2.3.3 Protocole de consensus PBFT et séquence d'exécution 26
2.3.4 Protocoles dérivés modernes : HotStuff et Tendermint 27
2.3.5 Optimisations modernes et défis des consensus à vote 31
2.4 Justification du cadre d'étude "vote-based" 31
2.5 Problématique 33
2.5.1 Problème 1 : la Sélection des Validateurs 33
2.5.2 Problème 2 : dynamiques des votes des validateurs 36
2.6 Conclusion et positionnement 42

Chapitre 3 Optimisation d'un protocole de sélection de validateurs dans un environnement BFT 43

3.1 Introduction 43
3.2 État de l'art 43
3.2.1 Enjeux de la sélection de validateurs 44
3.2.2 Méthodes existantes de sélection des validateurs 44
3.2.3 Détection de nœuds déviants 45
3.3 Métriques pour la sélection des validateurs et la détection des nœuds malveillants 47
3.3.1 Métriques de sélection des validateurs 47
3.3.2 Métriques et techniques de détection des nœuds malveillants 49
3.4 Compromis sécurité–performance dans la sélection BFT 50
3.4.1 Compromis sécurité–performance 50
3.4.2 Manifestations opérationnelles du compromis 51
3.4.3 Sabine[Leduc et al., 2022b] : Un exemple d'optimisation adaptative du compromis Sécurité/Débit 51
3.5 Modélisation par graphes d'une blockchain BFT 53
3.5.1 Concepts fondamentaux 53
3.5.2 Clique et performance réseau 54
3.5.3 Optimisation 55
3.6 Problème de clique 55
3.6.1 Énoncé du problème 55
3.6.2 Décentralisation 57

3.6.3 La densité de la clique . . . 58
3.7 Sélection centrée dynamique . . . 58
3.7.1 Algorithmes . . . 59
3.7.2 Simulations . . . 61
3.8 Conclusion . . . 67

Chapitre 4 Modélisation comportementale et détection des déviances des validateurs 68

4.1 Introduction . . . 68
4.2 Dynamique d'opinion des validateurs dans un réseau byzantin . . . 69
4.2.1 Modèles de dynamique d'opinion . . . 69
4.2.2 Formalisation du problème . . . 70
4.2.3 Modèle de dynamique par les proportions . . . 71
4.2.4 Implémentation et algorithme . . . 74
4.2.5 Étude par simulations . . . 75
4.3 Analyse de similarité comportementale des validateurs . . . 79
4.3.1 Métrique de similarité . . . 81
4.3.2 Méthodologie de clustering avancé . . . 83
4.3.3 Simulations . . . 83
4.4 Extension de l'analyse de similarité comportementale des validateurs . . . 86
4.4.1 Cadre conceptuel et modélisation . . . 87
4.4.2 Métriques probabilistes pour la comparaison . . . 87
4.4.3 Clustering et analyse comportementale . . . 89
4.4.4 Simulations . . . 91
4.5 Conclusion . . . 95

Conclusions et perspectives 97

Publications 99

Bibliographie 100

Table des figures

1.1 Représentation schématique d'un système distribué. Le schéma illustre l'interconnexion de nœuds hétérogènes (ordinateurs, serveurs, bases de données) au sein d'un réseau, mettant en évidence les caractéristiques fondamentales (tolérance aux pannes, évolutivité, hétérogénéité) ainsi que les défis majeurs (absence d'horloge globale, communication asynchrone) qui caractérisent ces systèmes. 7
1.2 Architecture d'un système distribué centralisé avec nœud contrôleur 9
1.3 Architecture de consensus dans une blockchain. 10
1.4 Architecture schématique d'une blockchain. Chaque bloc contient un en-tête et un corps composé de transactions. La liaison cryptographique entre blocs assure l'intégrité et l'immutabilité de la chaîne. 15
1.5 Classification des blockchains selon leur modèle d'accès et leur structure de gouvernance 19

2.1 Architecture en cinq phases du protocole PBFT. Les flèches indiquent les échanges de messages entre les nœuds et le client. 27
2.2 Architecture en trois couches du processus de sélection optimisée de validateurs. La couche 1 collecte les données, la couche 2 les analyse grâce à diverses méthodes, et la couche 3 partitionne le réseau pour une sélection ciblée des validateurs. . . . 34
2.3 Convergence conceptuelle entre les dynamiques d'opinion sociale et les technologies blockchain 38

3.1 Les phases de Sabine [Leduc et al., 2022b] 52
3.2 Modélisation du réseau blockchain par un graphe complet G. Le comité de validateurs C forme une clique où tous les nœuds sont interconnectés directement, garantissant une communication optimale. 54
3.3 L'ensemble de validateurs à des instants différents [Leduc et al., 2023]. 56
3.4 Mesure de la décentralisation après 2000 itérations en fonction du nombre de validateurs sur une chaîne de 200 nœuds. 62
3.5 Évolution du poids moyen (latence moyenne) des 25 validateurs au cours de 2000 itérations sur une chaîne de 200 nœuds. 63
3.6 Mesure de la décentralisation après 2000 itérations en fonction du nombre de validateurs sur une chaîne de 200 nœuds. 64
3.7 Évolution de la décentralisation pour $n = 25$ validateurs sur une chaîne de 200 nœuds en fonction des itérations. 65
3.8 Évolution de la latence moyenne pour $n = 25$ validateurs sur une chaîne de 200 nœuds en fonction des itérations. 66

4.1 Transitions possibles pour la proportion x de validateurs votant 1 dans un système à $n = 3$ validateurs. Les flèches indiquent les probabilités de passage $\gamma(x|x')$ entre états. 73
4.2 Schéma d'architecture du modèle markovien : les états discrets x_i et leurs probabilités $P(x_i)$ interagissent avec la matrice de transition $\gamma(x|x')$ pour produire le vecteur de probabilités et la concentration $c(t)$. 75
4.3 Évolution de la concentration $c(t)$ pour différents horizons n_γ dans un scénario de référence. 76
4.4 Évolution du nombre de validateurs votant 1 et de leur concentration au cours du temps, $n = 10$. 77
4.5 Probabilité de chaque proportion de validateurs votant 1 au cours du temps, $n = 10$, $n_\gamma = 20$. 78
4.6 Évolution de la concentration $c(t)$ lors d'une attaque continue et progressive. La dérive persistante vers $c(t) = 1.0$ reflète l'influence croissante des validateurs corrompus. 79
4.7 Illustration des 4 états des nœud sur deux itérations. 80
4.8 Comparaison schématique des divergences : (gauche) divergence KL unidirectionnelle P→Q ; (droite) divergence JSD symétrique entre P et Q via leur moyenne M. 82
4.9 Matrice de similarité fondée sur la divergence de Jensen-Shannon, illustrant les relations comportementales entre les 10 nœuds. 82
4.10 Matrice des divergences de Jensen-Shannon entre itérations successives pour chaque nœud. Les valeurs faibles (jaune) indiquent une stabilité comportementale, les valeurs élevées (rouge) une forte variation. 85
4.11 Évolution de la divergence moyenne de Jensen-Shannon entre itérations successives. La décroissance observée témoigne de la stabilisation comportementale du réseau. 85
4.12 Évolution de la stabilité des clusters au fil des itérations de consensus. 86
4.13 Distribution de probabilité des 4 états pour un nœud v. 88
4.14 Visualisation du coefficient de Bhattacharyya pour deux nœuds, chacun avec 4 états et deux distributions. Les barres vertes représentent $\sqrt{P(i) \cdot Q(i)}$ pour chaque état, et la somme de ces barres correspond à la valeur du BC. 89
4.15 Schéma conceptuel de la distance de Wasserstein pour un nœud à 4 états. Les barres représentent les probabilités de chaque état pour les distributions P (bleu) et Q (rouge), et les flèches vertes illustrent le transport de masse nécessaire pour transformer P en Q. 90
4.16 JSD moyenne entre tours successifs du consensus. 92
4.17 Coefficient de Bhattacharyya moyen entre tours successifs du consensus. 92
4.18 Distance de Wasserstein moyenne entre tours successifs du consensus. 92
4.19 Évolution de la taille des clusters au fil des itérations, illustrant la dynamique et la stabilité des regroupements de nœuds. 94
4.20 Évolution de la distance de Wasserstein intra-cluster. 94
4.21 Évolution du coefficient de Bhattacharyya intra-cluster. 95

Introduction générale

Dans un paysage numérique en constante évolution, la fiabilité des systèmes distribués s'impose comme un enjeu technologique majeur du XXIe siècle. Ces architectures décentralisées, qui orchestrent la collaboration de multiples nœuds autonomes, constituent l'épine dorsale d'infrastructures critiques allant des plateformes financières globales aux systèmes de santé numérisés, en passant par les réseaux de l'Internet des Objets et les écosystèmes de *cloud computing* [Van Steen and Tanenbaum, 2017]. La transformation digitale des sociétés contemporaines s'accompagne d'une dépendance exponentielle vis-à-vis de ces infrastructures distribuées, où la continuité de service, l'intégrité des données et la résilience face aux attaques représentent des impératifs non négociables.

La problématique fondamentale qui guide ces systèmes complexes réside dans la nécessité de maintenir une cohérence globale malgré l'absence de coordinateur central et la possibilité de défaillances partielles. Les pannes, qu'elles soient accidentelles dues à des erreurs matérielles, des bogues logiciels ou des latences réseau ou malveillantes résultant d'attaques ciblées ou de comportements byzantins menacent constamment l'intégrité des opérations. Historiquement, cette problématique a trouvé sa formulation la plus aboutie dans le célèbre problème des généraux byzantins [Lamport et al., 1982], qui pose les bases théoriques de la coordination dans un environnement distribué où certains participants peuvent adopter un comportement arbitraire. Les travaux fondateurs de Lamport et al. ont démontré qu'atteindre un consensus dans de telles conditions exige des protocoles sophistiqués capables de tolérer jusqu'à un tiers de participants défaillants.

C'est dans ce contexte exigeant que les protocoles de tolérance aux fautes byzantines (BFT) ont émergé comme solution élégante et robuste. Leur principe fondamental repose sur la capacité d'un système distribué à fonctionner correctement malgré la présence d'un certain nombre de nœuds défaillants ou malveillants, pour autant que la proportion de ces nœuds ne dépasse pas un seuil critique établi théoriquement. Le protocole PBFT (Practical Byzantine Fault Tolerance) développé par Castro et Liskov [Castro and Liskov, 1999] a marqué un tournant décisif en proposant une implémentation pratique capable de fonctionner efficacement dans des environnements partiellement synchrones, ouvrant ainsi la voie à des applications concrètes dans divers domaines critiques.

L'avènement des technologies blockchain a provoqué un regain d'intérêt remarquable pour ces protocoles BFT. Alors que les blockchains publiques comme Bitcoin [Nakamoto, 2008] privilégient initialement les mécanismes de preuve de travail (Proof of Work), l'émergence de blockchains de consortium et privées a mis en lumière les avantages des approches BFT, notamment leur efficacité énergétique supérieure, leur finalité transactionnelle immédiate et leur capacité à gérer des débits élevés. Des protocoles modernes comme Tendermint [Buchman et al., 2018], HotStuff [Yin et al., 2019] ou encore les variantes optimisées de PBFT ont élargi le spectre d'application des consensus byzantins, permettant leur déploiement dans des contextes industriels exigeants où la performance et la sécurité doivent coexister harmonieusement.

Cependant, malgré ces avancées significatives, les implémentations contemporaines de protocoles BFT se heurtent à des limitations persistantes qui entravent leur adoption à grande échelle. La rigidité des mécanismes de sélection des validateurs constitue l'un des verrous les plus contraignants. Dans de nombreuses architectures, la composition du comité de validation reste statique ou évolue selon des schémas prévisibles, créant ainsi des vulnérabilités exploitables par des adversaires sophistiqués. La complexité communicationnelle quadratique des protocoles classiques [Castro and Liskov, 1999] limite par ailleurs la scalabilité des systèmes, restreignant de facto la taille des comités de validation et réduisant potentiellement la décentralisation.

Un défi plus subtil réside dans l'incapacité des approches traditionnelles à capturer la dynamique comportementale des validateurs. Les modèles binaires qui catégorisent les nœuds comme simplement "honnêtes" ou "byzantins" peinent à rendre compte des états intermédiaires et des transitions comportementales progressives qui caractérisent les environnements réels. Un validateur peut ainsi évoluer graduellement d'un comportement conforme vers des attitudes déviantes, que ce soit sous l'effet de compromissions partielles, de pannes intermittentes ou d'attaques sophistiquées comme les campagnes de slow-adaptation. La détection précoce de ces potentielles dérives comportementales représente un enjeu de sécurité majeur pour lequel les solutions existantes restent insuffisantes.

C'est précisément dans cet espace problématique que s'inscrit notre recherche doctorale. Notre travail se propose de relever ces défis through une approche intégrée combinant modélisation théorique rigoureuse et validation expérimentale poussée. Nous postulons que l'optimisation dynamique de la sélection des validateurs, couplée à une analyse fine de leur comportement temporel, peut significativement améliorer la robustesse et l'efficacité des protocoles BFT dans des environnements distribués complexes.

La première contribution de cette thèse réside dans le développement d'un modèle novateur de sélection des validateurs fondé sur une approche centrée dynamique. Contrairement aux méthodes statiques ou purement aléatoires, notre protocole intègre simultanément des critères de performance réseau, en privilégiant les validateurs à faible latence mutuelle et des impératifs de décentralisation en assurant une rotation équitable des rôles de validation. L'utilisation d'une métrique de distance compensée, ajustée dynamiquement en fonction de l'historique de participation des nœuds, permet de maintenir un équilibre optimal entre efficacité opérationnelle et résilience à long terme. Cette approche s'inscrit dans la lignée des travaux récents sur l'optimisation des performances des protocoles BFT [Leduc et al., 2023, Abraham et al., 2018a].

Notre deuxième contribution majeure concerne la modélisation des dynamiques comportementales within les réseaux de validateurs. En nous appuyant sur des concepts empruntés à la physique statistique et aux systèmes complexes [Castellano et al., 2009a], nous avons développé un cadre formel pour analyser l'évolution temporelle des comportements des validateurs. L'approche par dynamique d'opinion que nous proposons permet de capturer les transitions d'état subtiles qui échappent aux modèles classification binaires traditionnels. La formalisation markovienne de l'évolution des proportions de validateurs adoptant un comportement donné offre une base mathématique solide pour la détection précoce d'anomalies comportementales.

La troisième contribution originale de nos travaux réside dans le développement d'une méthodologie complète d'analyse de similarité comportementale. En exploitant des métriques avancées comme la divergence de Jensen-Shannon [Menéndez et al., 1997], le coefficient de Bhattacharyya [Bhattacharyya, 1943] et la distance de Wasserstein [Villani and Villani, 2009], nous proposons un cadre unifié pour quantifier les similarités et différences entre les profils comportementaux des validateurs. Le couplage de ces métriques avec des algorithmes de clustering densité comme HDBSCAN *Hierarchical Density-Based Spatial Clustering of Applications with Noise*) [Campello et al., 2013] permet d'identifier des patterns comportementaux collectifs et de détecter des dé-

viations significatives par rapport aux comportements attendus.

L'architecture de ce manuscrit reflète la progression logique de notre démarche de recherche. Le premier chapitre établit les fondements techniques essentiels à la compréhension des blockchains et des systèmes distribués, en retraçant l'évolution historique de ces technologies et en présentant leur architecture fondamentale. Nous y analysons les propriétés théoriques qui guident ces systèmes, notamment les concepts de safety et liveness [Alpern and Schneider, 1985], les modèles temporels de synchronisation, et les principes cryptographiques garantissant l'intégrité des données.

Le deuxième chapitre se concentre spécifiquement sur les protocoles de consensus BFT, en offrant une analyse comparative approfondie des différentes familles de protocoles. Nous y examinons les compromis fondamentaux entre sécurité, performance et décentralisation, et nous identifions les limitations des approches existantes en matière de sélection des validateurs et de gestion des dynamiques comportementales. Ce chapitre sert de fondement théorique à nos contributions ultérieures.

Le troisième chapitre présente une adaptation d'un protocole de sélection centrée dynamique des validateurs. Nous y détaillons le modèle mathématique sous-jacent, l'algorithme d'optimisation et les métriques d'évaluation que nous avons développées. Les résultats de simulations extensives démontrent l'efficacité de notre approche pour maintenir simultanément de faibles latences moyennes et un degré élevé de décentralisation, dépassant ainsi les performances des méthodes traditionnelles.

Le quatrième et dernier chapitre élargit la perspective en proposant un cadre méthodologique complet pour la modélisation comportementale et la détection des déviances dans les réseaux de validateurs BFT. Nous y présentons nos travaux sur la dynamique d'opinion des validateurs, l'analyse de similarité comportementale et les techniques de clustering avancé. La validation expérimentale sur des réseaux simulés de grande taille confirme l'efficacité de notre approche pour détecter divers scénarios d'attaque, depuis les déviations transitoires jusqu'aux compromissions progressives.

Au-delà de ces contributions spécifiques, cette thèse vise à établir les bases d'une nouvelle approche pour la conception de systèmes de consensus byzantins plus adaptatifs et résilients. En intégrant des mécanismes d'observation fine du comportement des validateurs et des boucles de rétroaction pour l'ajustement dynamique des paramètres opérationnels, nous ouvrons la voie vers des protocoles capables de s'adapter automatiquement aux conditions changeantes des environnements distribués réels.

Les perspectives de recherche ouvertes par ces travaux sont nombreuses et prometteuses. L'intégration de techniques d'apprentissage automatique pour la détection en temps réel d'anomalies comportementales, l'extension des modèles à des environnements multi-chaînes hétérogènes, et l'optimisation des compromis sécurité-performance dans des contextes opérationnels contraints représentent autant de directions futures passionnantes. Nous espérons que les avancées présentées dans ce manuscrit contribueront à l'émergence d'une nouvelle génération de systèmes distribués, plus robustes, plus adaptatifs et mieux armés pour faire face aux défis sécuritaires du monde numérique de demain.

Chapitre 1

Fondements techniques de la blockchain

1.1 Introduction

La blockchain représente l'une des innovations technologiques les plus marquantes des dernières décennies, promettant de transformer fondamentalement notre conception des systèmes de confiance et des échanges numériques. Née avec le Bitcoin de Satoshi Nakamoto en 2008, cette technologie a rapidement évolué au-delà de son application monétaire initiale pour s'étendre à des domaines aussi variés que la finance décentralisée, la gestion de supply chain, l'identité numérique et les systèmes de gouvernance.

Ce premier chapitre se propose d'établir les fondements théoriques et techniques essentiels à la compréhension de la technologie blockchain. Notre objectif est triple : retracer son évolution historique, analyser son architecture fondamentale, et situer ses principes dans le cadre plus large des systèmes distribués. Cette approche nous semble indispensable pour appréhender correctement les enjeux techniques et les compromis de conception qui caractérisent les différentes implémentations blockchain.

Nous débuterons par un retour sur les origines conceptuelles de la blockchain (Section 1.2), en mettant en lumière les travaux précurseurs en cryptographie et en systèmes distribués qui ont pavé la voie à son émergence. Des premières tentatives de création de monnaies numériques décentralisées aux innovations fondamentales introduites par Bitcoin et Ethereum, nous examinerons comment cette technologie synthétise plusieurs décennies de recherche académique.

Dans un second temps (Section 1.3), nous analyserons la blockchain en tant que système distribué particulier, en explorant ses propriétés fondamentales de *safety* et *liveness* (Section 1.3.2), ses modèles temporels (Section 1.3.7), et le paradigme de réplication de machines à états (SMR, Section 1.3.4) qui en constitue le fondement théorique. Cette perspective nous permettra de comprendre les défis spécifiques que doit relever la blockchain dans des environnements ouverts.

Nous aborderons également les garanties de fiabilité des transactions à travers le prisme des propriétés ACID et du théorème CAP (Section 1.3.6), ainsi que les enjeux de centralisation et décentralisation (Section 1.3.3) qui influencent directement l'architecture des systèmes blockchain.

Enfin, nous détaillerons l'architecture technique concrète des blockchains (Section 1.4), depuis les principes cryptographiques qui garantissent leur sécurité (Section 1.4.1) jusqu'aux différentes topologies de réseau (Section 1.5) qui déterminent leurs propriétés opérationnelles. Une attention particulière sera portée aux mécanismes de consensus, dont le rôle crucial justifiera l'approfondissement qui fera l'objet des chapitres suivants.

Ce cadrage théorique et technique constitue un prérequis essentiel pour la suite de notre travail, qui se concentrera plus spécifiquement sur l'étude comparative des protocoles de consen-

sus, avec un accent particulier sur la famille des protocoles BFT (Byzantine Fault-Tolerant). En établissant solidement ces bases fondamentales, nous nous donnons les moyens d'analyser avec rigueur les avancées récentes dans ce domaine et leurs implications pour le développement de systèmes blockchain performants et sécurisés.

1.2 Évolution historique et contexte d'émergence

La blockchain ne s'est pas développée dans un vide technologique, mais résulte de la convergence de recherches en informatique distribuée, en cryptographie et en théorie des jeux. La compréhension de cette genèse historique est cruciale pour appréhender les principes architecturaux fondamentaux qui structurent cette technologie.

1.2.1 Précurseurs cryptographiques et concepts fondateurs

Les bases conceptuelles de la blockchain remontent aux travaux pionniers des années 1980-1990. [Haber and Stornetta, 1990] proposent l'un des premiers systèmes pratiques pour horodater des documents numériques de manière sécurisée, utilisant déjà des chaînes de hachages cryptographiques. Leur système, bien que centralisé, établit le principe fondamental de l'enchaînement temporel par preuves cryptographiques.

Parallèlement, les recherches sur les protocoles byzantins, initiées par [Lamport et al., 1982], posent les bases théoriques de la coordination dans des environnements décentralisés avec participants potentiellement malveillants. Le problème des généraux byzantins et sa résolution partielle ouvrent la voie aux systèmes tolérants aux fautes dans des contextes adversariaux.

La notion de preuve de travail (*Proof of Work*), souvent associée à Bitcoin, trouve en réalité ses origines dans les travaux de Dwork and Naor [1992] sur la prévention du spam et des attaques par déni de service. Leur système *Hashcash* utilise déjà la résolution de puzzles cryptographiques comme mécanisme de coût computationnel.

1.2.2 De Bit Gold à Bitcoin : la quête de la monnaie numérique décentralisée

Avant Bitcoin, plusieurs tentatives de création de monnaies numériques décentralisées ont échoué, principalement à cause du problème de double dépense dans des environnements pair-à-pair. Les propositions de [Back et al., 2002] avec Hashcash et de [Szabo, 2005] avec Bit Gold contenaient déjà de nombreux éléments constitutifs des blockchains modernes, mais manquaient d'un mécanisme de consensus robuste pour un réseau entièrement décentralisé.

L'innovation fondamentale de Satoshi Nakamoto en 2008 [Nakamoto, 2008] réside dans la combinaison ingénieuse de ces briques technologiques :

— L'utilisation d'une chaîne de blocs horodatée pour l'enregistrement immuable des transactions

— La preuve de travail comme mécanisme de consensus et de sélection des validateurs

— Un système d'incitations économiques alignant les intérêts des participants

— La résolution élégante du problème de double dépense par la règle de la chaîne la plus longue

Cette synthèse créative transforme des concepts théoriques en un système opérationnel résistant à la censure et fonctionnant sans autorité centrale.

1.2.3 Évolutions post-Bitcoin et diversification des paradigmes

L'émergence d'Ethereum [Buterin et al., 2013] marque une seconde étape fondamentale avec l'introduction de la programmabilité via les contrats intelligents. Cette innovation étend la blockchain du simple registre transactionnel à une plateforme de calcul décentralisée, ouvrant la voie aux applications décentralisées.

Parallèlement, la montée en puissance des problématiques d'évolutivité et de la consommation énergétique excessive du mécanisme de consensus originel de Bitcoin, la Preuve de Travail (*Proof of Work, PoW*), a stimulé la recherche sur des alternatives. Dans la Preuve de Travail, les nœuds appelés mineurs sont en compétition pour résoudre un problème cryptographique complexe, consommant d'importantes ressources computationnelles pour valider les transactions et créer de nouveaux blocs. La Preuve d'Enjeu (*Proof of Stake, PoS*), initialement proposée par [King and Nadal, 2012] puis perfectionnée dans des implémentations comme Ethereum 2.0, représente une évolution majeure. Ce paradigme remplace les mineurs par des validateurs qui sont choisis pour créer le bloc suivant en fonction de la quantité de cryptomonnaie qu'ils possèdent et "misent" (leur *stake*) en garantie, réduisant ainsi considérablement la consommation énergétique du réseau.

Cette évolution historique illustre comment la blockchain incarne la maturation progressive de concepts distribués et cryptographiques, transformés en systèmes pratiques grâce à des innovations architecturales fondamentales.

1.3 Du système distribué à blockchain

Avant d'aborder les spécificités de la blockchain, il est essentiel de comprendre les principes des systèmes distribués, sur lesquels reposent ses fondements techniques. Ces systèmes permettent à plusieurs nœuds indépendants de coopérer pour accomplir des tâches communes tout en maintenant une apparence de cohérence et d'unité pour l'utilisateur final. La blockchain peut être vue comme une évolution particulière de ces systèmes distribués, intégrant des notions de consensus, de tolérance aux fautes et de sécurité cryptographique dans un environnement ouvert et potentiellement hostile. Cette section présente d'abord les caractéristiques clés des systèmes distribués, examine les notions de centralisation et de décentralisation, explore les modèles temporels qui influencent la faisabilité des algorithmes distribués, puis introduit la réplication de machines à états (SMR), un paradigme central pour la cohérence et l'atomicité dans les systèmes distribués.

1.3.1 Système distribué

Un système distribué peut être défini comme un ensemble d'ordinateurs autonomes, interconnectés par un réseau, qui coopèrent afin d'accomplir une tâche commune. À l'utilisateur final, un tel système apparaît comme une entité unique et cohérente, bien qu'il soit constitué d'une multiplicité de composants indépendants. Deux caractéristiques fondamentales en découlent [Van Steen and Tanenbaum, 2017] : d'une part, l'absence de mémoire partagée globale (chaque nœud dispose uniquement de sa mémoire locale) et, d'autre part, l'absence d'horloge universelle fiable permettant de synchroniser l'ensemble des processus.

Comme illustré dans la figure 1.1, un système distribué vise généralement à offrir une transparence vis-à-vis de l'utilisateur, en masquant la complexité de la localisation des données, de la réplication ou encore de la tolérance aux pannes. Les propriétés recherchées incluent notamment :

— **la tolérance aux pannes**, c'est-à-dire la capacité à fonctionner, malgré des défaillances partielles des nœuds ou des canaux de communication ;

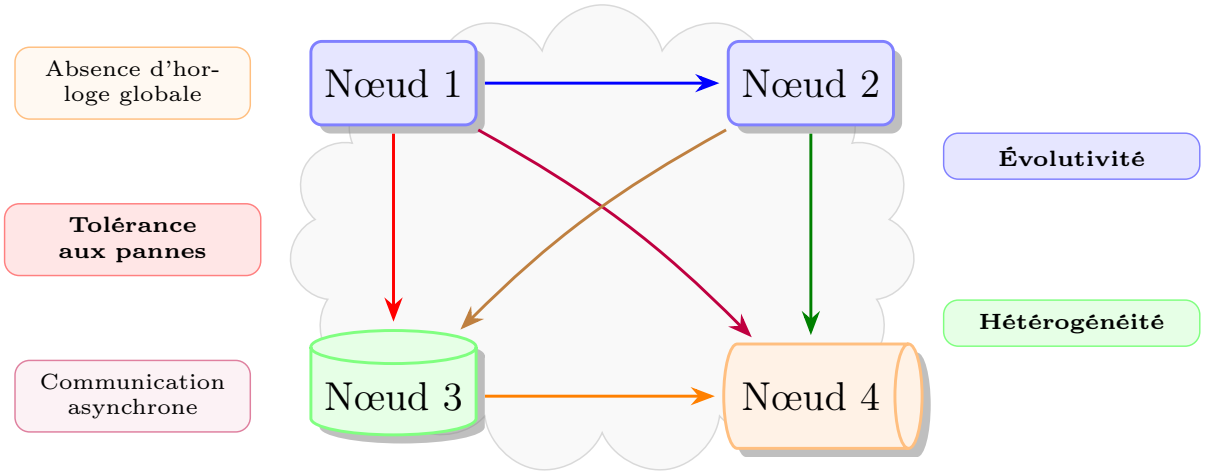


FIGURE 1.1 – Représentation schématique d'un système distribué. Le schéma illustre l'interconnexion de nœuds hétérogènes (ordinateurs, serveurs, bases de données) au sein d'un réseau, mettant en évidence les caractéristiques fondamentales (tolérance aux pannes, évolutivité, hétérogénéité) ainsi que les défis majeurs (absence d'horloge globale, communication asynchrone) qui caractérisent ces systèmes.

— **l'évolutivité**, permettant au système de s'adapter à une augmentation du nombre de nœuds et du volume de données échangées ;

— **l'hétérogénéité**, qui reflète la coexistence de matériels, de systèmes d'exploitation et de logiciels différents ;

Ces caractéristiques s'accompagnent toutefois de plusieurs défis. L'absence d'horloge globale complique la coordination des processus et impose le recours à des mécanismes logiques de synchronisation.(par exemple, les horloges de Lamport [Lamport, 1978]).

La communication est généralement asynchrone : les messages peuvent être retardés, dupliqués ou perdus. Le problème du *consensus distribué* se pose alors, c'est-à-dire parvenir à un accord entre processus même en présence de défaillances ou de comportements malveillants (pannes byzantines). La sécurité des échanges et les performances du système (latence, bande passante, cohérence) constituent également des enjeux majeurs.

De nombreux exemples de systèmes distribués existent : les systèmes de fichiers distribués (par exemple, *Google File system* [Ghemawat et al., 2003a,b], *Hadoop Distributed File System*(HDFS) [Shvachko et al., 2010]), les bases de données réparties (*Google Cloud Spanner* [Corbett et al., 2013] ou encore *Apache Cassandra*[Lakshman and Malik, 2010]). Les environnements modernes de calcul, tels que le cloud computing [Qian et al., 2009], reposent également sur ces principes. Enfin, la blockchain peut être vue comme une évolution particulière des systèmes distribués, caractérisée par l'absence d'autorité de confiance et la nécessité de mettre en place un consensus robuste dans un environnement ouvert et potentiellement hostile.

1.3.2 Propriétés fondamentales : Safety et Liveness

La conception d'algorithmes distribués corrects nécessite une formalisation rigoureuse de leurs garanties. Les propriétés de sécurité *(Safety)* et de vivacité *(Liveness)*, introduites par Lamport [Alpern and Schneider, 1985], constituent le cadre théorique fondamental pour spécifier et vérifier la correction des systèmes distribués. Ces deux propriétés offrent une grille de lecture

essentielle pour analyser les compromis inhérents à la distribution, particulièrement dans des environnements contraints par les résultats d'impossibilité comme le théorème FLP.

1. **Propriété de "Safety" (sécurité)** :
   Une propriété de *safety* stipule que *« quelque chose de mauvais ne se produit jamais »*. Il s'agit d'une propriété qui doit être vérifiée à chaque instant de l'exécution du système. Une violation de *safety* est généralement irrémédiable et souvent detectable immédiatement.

   (a) **Cohérence forte** : Dans une base de données distribuée, la cohérence forte garantit que toute opération de lecture renvoie la valeur de la dernière écriture qui l'a précédée dans le temps. Une lecture qui renvoie une valeur ancienne alors qu'une valeur plus récente a déjà été observée par un autre client constitue une *violation de la safety.*

   (b) **Coordonnateur unique (élection de leader)** : Dans un système utilisant un protocole d'élection de leader, une fois qu'une élection est décidée, il ne doit jamais y avoir deux processus qui se considèrent simultanément comme le leader. Une telle situation représenterait une *violation directe de la safety* du protocole.

   Formellement, une propriété de safety peut être définie comme une assertion qui doit être vraie pour tout préfixe de toute exécution du système.

2. **Propriété de Liveness (vivacité)** :
   Une propriété de *liveness* stipule que *« quelque chose de bon finit par se produire »*. Il s'agit d'une propriété qui garantit que le système n'atteint pas un état bloqué. Contrairement à la safety, une violation de *liveness* peut être temporaire et n'est pas toujours détectable immédiatement.

   (a) **Traitement des requêtes client** : Toute requête valide soumise à un serveur doit être éventuellement traitée et recevoir une réponse. Le système ne doit ni ignorer ni perdre une requête indéfiniment.

   (b) **Protocole d'élection de leader** :Si le leader actuel tombe en panne, un nouveau leader doit être élu après un certain temps. Le système ne doit pas rester indéfiniment sans leader opérationnel, garantissant ainsi la propriété de vivacité du protocole.

1.3.3 Centralisation et décentralisation

Il est important de noter que la notion de distribution ne détermine pas automatiquement la centralisation ou la décentralisation d'un système.

— Dans un **système distribué centralisé**, un ou plusieurs nœuds jouent un rôle de coordination ou de contrôle unique, comme dans *Google File System* avec son serveur maître qui gère les métadonnées, ou certaines bases de données distribuées où un maître central arbitre les écritures. La figure 1.2 illustre ce type d'architecture avec un nœud contrôleur central.

— Dans un **système distribué décentralisé**, tous les nœuds ont un pouvoir équivalent et prennent des décisions collectivement, comme dans les blockchains publiques (Bitcoin, Ethereum) où aucun nœud n'est autorité centrale.

Ainsi, si les systèmes distribués centralisés offrent une gestion simplifiée et souvent de meilleures performances grâce à un point de contrôle unique, ils présentent aussi une vulnérabilité accrue en cas de défaillance ou de compromission de ce nœud central. À l'inverse, les architectures décentralisées améliorent la tolérance aux pannes, mais au prix d'une complexité accrue pour atteindre un consensus entre nœuds.

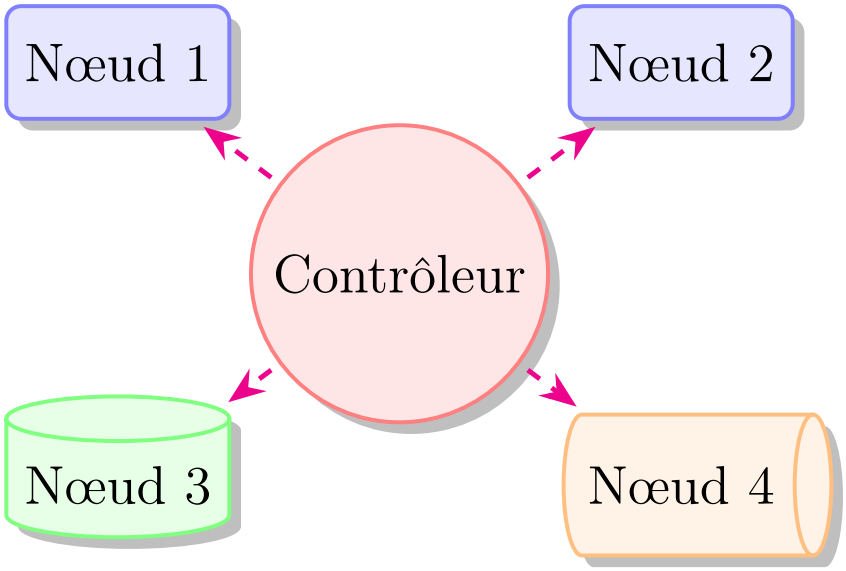


FIGURE 1.2 – Architecture d'un système distribué centralisé avec nœud contrôleur

1.3.4 Réplication de machines à états (SMR)

La réplication de machines à états (SMR) [Schneider, 1990] représente un paradigme fondamental pour la conception de systèmes distribués tolérants aux pannes. Son principe architecturale consiste à maintenir la cohérence d'un service à travers plusieurs réplicas en garantissant que chacun d'eux exécute la même séquence de commandes d'état dans un ordre identique. Cette approche assure que tous les nœuds non défaillants évoluent selon un état identique, offrant ainsi une perception d'un système unique et cohérent même en cas de défaillances partielles.

Formellement, un système basé sur SMR s'articule autour de trois composants principaux :

— Les **clients**, qui initient les opérations en soumettant des requêtes au système.

— Les **réplicas**, chargés de maintenir une copie locale de l'état et d'appliquer chaque commande de manière déterministe et atomique. Ici, l'atomicité signifie que chaque commande est exécutée dans son intégralité ou pas du tout, aucune exécution partielle ne peut altérer l'état du réplica.

— Un **service de diffusion atomique** (*atomic broadcast*), qui assure la livraison ordonnée et fiable des commandes à l'ensemble des réplicas. Ce mécanisme complète l'atomicité des réplicas en garantissant que toutes les réplicas voient et appliquent exactement les mêmes commandes dans le même ordre, assurant ainsi un séquencement logique global et prévenant toute divergence d'état

La viabilité pratique de ce modèle est étroitement tributaire des hypothèses temporelles (sous section 1.3.7) du système distribué sous-jacent. Dans un environnement synchronisé, où les délais de communication et d'exécution sont bornés, l'établissement d'un ordre total est réalisable selon des mécanismes relativement simples. À l'opposé, dans un modèle asynchrone pur où aucun délai n'est garanti, le théorème d'impossibilité de Fischer, Lynch et Paterson établit l'impossibilité de résoudre le consensus déterministe en présence de défaillances. C'est dans le modèle partiellement synchrone, qui suppose une stabilisation eventuelle des délais, que des protocoles comme Paxos [Lamport, 2001] et Raft [Ongaro and Ousterhout, 2014] ont été développés pour fournir une implémentation efficace de la SMR pour des systèmes réels.

Fondamentalement, la réplication de machines à états se ramène au problème du consensus : les réplicas doivent s'accorder sur l'ordre d'inclusion de chaque commande dans la séquence logique globale. Cette question du consensus constitue également le cœur du fonctionnement des blockchains

1.3.5 Blockchain et consensus

Une blockchain est un registre numérique distribué qui enregistre des transactions de manière sécurisée et vérifiable. Selon leur mode d'accès, on distingue deux types principaux :

— les **blockchains publiques**, ouvertes à tous, où n'importe quel participant peut consulter la blockchain, proposer des transactions et vérifier les informations stockées ;

— les **blockchains avec permission** (ou privées), où seules certaines entités autorisées peuvent écrire ou valider les transactions. Ces réseaux restreignent l'accès et le contrôle à un groupe défini de participants.

Le fonctionnement de ces réseaux repose sur un mécanisme de consensus, un protocole permettant à l'ensemble des nœuds du réseau de s'accorder sur l'état du registre partagé. Le consensus assure que toutes les copies de la blockchain maintiennent une séquence cohérente de blocs valides, préservant ainsi l'intégrité du système sans nécessiter de confiance préalable entre les nœuds.

Au cœur du réseau se trouvent les transactions, unités fondamentales représentant des transferts d'actifs ou l'exécution d'opérations. Chaque transaction est cryptographiquement signée par son émetteur et vérifiée par les nœuds selon des règles prédéfinies. Plusieurs transactions validées sont ensuite regroupées pour former un bloc.

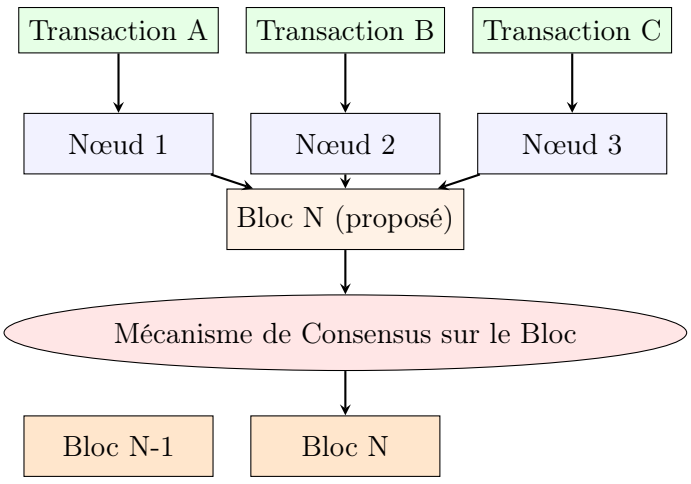


FIGURE 1.3 – Architecture de consensus dans une blockchain.

Comme montré à la figure 1.3, les transactions sont d'abord vérifiées par les nœuds avant d'être regroupées en blocs, et le consensus valide ensuite le bloc choisi pour l'ajout à la blockchain.

Chaque bloc contient :

— un en-tête avec des métadonnées importantes (hachage du bloc précédent, horodatage, racine de Merkle [Merkle, 1987] des transactions) ;

— la liste des transactions validées.

L'ajout de nouveaux blocs à la blockchain suit les règles du mécanisme de consensus, qui garantit que tous les participants partagent la même version de l'historique transactionnel.

Cependant, cette coordination n'est pas totalement sécurisée. Certaines attaques exploitent les mécanismes de consensus, comme l'attaque des 51%, où un acteur contrôlant la majorité du réseau peut réorganiser temporairement la blockchain, ou l'attaque de double dépense, où un utilisateur tente de dépenser deux fois les mêmes actifs en créant des transactions conflictuelles.

1.3.6 Fiabilité des Transactions : ACID et CAP

Une blockchain est une structure de données distribuée qui enregistre de manière chronologique et immuable l'historique des transactions effectuées entre ses participants. Chaque transaction doit respecter certaines propriétés pour garantir l'intégrité et la fiabilité du système. Ces propriétés sont généralement liées aux principes **ACID** (Atomicity, Consistency, Isolation, Durability) [Härder and Reuter, 1983].

Ces propriétés ACID assurent la fiabilité au sein de chaque blockchain, mais dans un environnement distribué et multi-chaînes, des compromis entre cohérence, disponibilité et tolérance aux partitions doivent également être pris en compte.

ACID

Les propriétés ACID constituent un cadre fondamental pour la fiabilité des transactions dans les systèmes distribués, y compris les blockchains. Elles définissent les contraintes nécessaires pour garantir que chaque transaction est exécutée de manière correcte, cohérente et sécurisée, même en présence de défaillances ou de comportements malveillants.

1. **L'atomicité (A) :** L'atomicité d'une transaction entre deux nœuds d'une blockchain est définie par son exécution qui survient intégralement et sans interruption, autrement, elle ne survient pas.
   Dans le cas d'une transaction inter-chaines, cette transaction peut être considérée comme deux transactions séparées, où chacune est exécutée dans une seule blockchain [Ou et al., 2022]. Donc, l'atomicité telle qu'elle est définie dans le cas d'une transaction au sein d'une seule blockchain n'est plus valable. D'où, il faudrait trouver un moyen de résoudre ce problème.
2. **La cohérence(C) :** La cohérence d'un système distribué décrit sa capacité à pouvoir (lire) retourner la dernière version écrite. Nous pouvons aussi définir la cohérence comme la capacité des nœuds du système distribué à converger et atteindre un état unique.
3. **L'isolation (I) :** L'isolation se réfère au fait que les transactions doivent être exécutées de manière indépendante et isolée les unes des autres.
4. **Durabilité(D) :** Une fois qu'une transaction est validée et enregistrée sur la blockchain, elle devient permanente et ne peut pas être modifiée ou supprimée. Cela assure la durabilité des données et l'intégrité de l'historique des transactions.

Le théorème CAP :

Egalement connu sous le nom de théorème de Brewer, a été formulé par Éric Brewer en 2000 [Brewer, 2000] et prouvé par Seth Gilbert et Nancy Lynch en 2002 [Gilbert and Lynch, 2002]. Ce théorème énonce qu'il est impossible pour un système distribué de garantir simultanément les trois contraintes suivantes, associées aux lettres "CAP" :

1. **Cohérence (Consistency)** : Chaque demande de lecture doit renvoyer la dernière écriture la plus récente. En d'autres termes, toutes les opérations de lecture doivent refléter les modifications les plus récentes effectuées sur le système. Cette propriété est essentielle pour maintenir l'intégrité des données et garantir leur cohérence.
2. **Disponibilité (Availability)** : Le système doit être en mesure de répondre aux requêtes qui lui sont soumises, même en présence de défaillances ou de pannes. Cela signifie que le système doit être opérationnel et accessible pour les utilisateurs, sans interruptions majeures de service.
3. **Tolérance au partitionnement (Partition Tolerance)** : Le système doit être capable de fonctionner correctement même en présence de partitions ou de déconnexions dans le réseau. Une partition se produit lorsqu'un réseau distribué est divisé en sous-réseaux isolés qui ne peuvent pas communiquer entre eux. La tolérance au partitionnement garantit que le système peut continuer à fonctionner de manière cohérente malgré ces partitions.

Le théorème CAP peut être interprété comme une manifestation particulière du compromis entre *safety* et *liveness* (section 1.3.2) :

— **Cohérence (C)** → *Safety* : Garantir la cohérence empêche « quelque chose de mauvais » (lecture de données obsolètes).

— **Disponibilité (A)** → *Liveness* : Garantir la disponibilité assure que « quelque chose de bon finit par se produire » (service répond toujours).

1.3.7 Modèles temporels

Dans un système distribué, l'absence d'horloge globale et la variabilité des délais de communication rendent difficile la coordination des processus et la garantie de l'ordre des événements. Pour raisonner sur ces systèmes, il est nécessaire d'introduire des modèles temporels, qui posent des hypothèses sur le comportement du temps et des communications. Ces modèles influencent directement la faisabilité et la conception des algorithmes distribués, notamment en matière de consensus.

Systèmes synchrones

Dans un système synchrone, on suppose que les délais de communication et le temps de calcul de chaque processus sont bornés et connus. Chaque message envoyé parvient donc à destination dans un délai maximum garanti, et chaque processus exécute ses opérations à une vitesse bornée. Ce modèle simplifie grandement la conception des algorithmes distribués, car il permet d'utiliser des mécanismes de coordination déterministes. Toutefois, il est rarement réaliste dans les environnements ouverts comme Internet, où les retards de communication peuvent être arbitrairement longs, ils sont soumis à une forte variance et à priori inconnus.

Systèmes asynchrones

Dans un système asynchrone, aucune borne n'est supposée sur les délais de transmission des messages ou sur la vitesse de calcul des processus. Les messages peuvent être retardés indéfiniment, ce qui reflète mieux le comportement réel des réseaux à grande échelle, de même, des messages peuvent être perdus.

Cependant, ce modèle entraîne une limitation fondamentale : Fischer, Lynch et Paterson ont démontré dans leur célèbre résultat FLP [Fischer et al., 1985] qu'il est impossible de garantir

un consensus déterministe en présence d'au moins une panne de processus dans un système entièrement asynchrone. Plus précisément, dans un tel système, rien ne permet de distinguer de manière fiable si un processus est en panne ou s'il est simplement très lent à répondre. Par exemple, considérons un système distribué où plusieurs processus doivent s'accorder sur une valeur unique (comme l'ordre d'exécution d'une transaction). Si un des processus cesse d'envoyer des messages, les autres ne peuvent pas savoir s'il est définitivement tombé en panne ou s'il subit seulement un délai réseau. Dans les deux cas, l'incertitude empêche les autres processus de décider de façon déterministe sans risquer de violer la propriété d'accord ou de terminaison. Ce paradoxe illustre la barrière fondamentale posée par FLP : dans un modèle purement asynchrone, aucune hypothèse de temps ne peut être exploitée pour lever cette ambiguïté, rendant impossible la construction d'un protocole de consensus tolérant aux fautes qui garantisse toujours la convergence.

Ce résultat souligne l'importance l'hypothèse temporelle dans la conception des algorithmes distribués.

Systèmes partiellement synchrones

Le modèle partiellement synchrone constitue un compromis entre les deux précédents. Il suppose que des bornes existent sur les délais de communication et la vitesse des processus, mais que celles-ci ne sont pas connues à l'avance, ou bien qu'elles ne sont garanties qu'après un certain temps. Ce modèle est plus réaliste pour de nombreux environnements pratiques, où les délais peuvent varier mais finissent par se stabiliser. De nombreux protocoles de consensus modernes, y compris ceux utilisés dans les blockchains (par exemple, le protocole de Nakamoto [Nakamoto, 2008]dans Bitcoin), reposent implicitement sur ce type d'hypothèses.

1.4 Architecture fondamentale de la blockchain

L'architecture fondamentale d'une blockchain repose sur l'agencement de plusieurs briques technologiques en une structure cohérente et résiliente. Comme le montre la Figure 1.4, celle-ci est matérialisée par une chaîne de blocs immuable, où chaque bloc est cryptographiquement lié à son prédécesseur via son hachage, formant ainsi un registre séquentiel et tamponné dans le temps. Cette structure est maintenue par un réseau pair-à-pair de nœuds qui appliquent un protocole de consensus.

1.4.1 Principes cryptographiques fondamentaux

La blockchain, en tant que registre distribué et immuable, tire sa robustesse de l'association de concepts de systèmes distribués et de principes cryptographiques éprouvés. La cryptographie n'est pas un simple accessoire ; elle est le ciment qui garantit la confidentialité, l'intégrité, l'authentification et la non-répudiation des transactions et des blocs. Cette section détaille les trois piliers cryptographiques essentiels au fonctionnement de toute blockchain : les fonctions de hachage, les signatures numériques et le chiffrement asymétrique.

Rôle des fonctions de hachage

Une fonction de hachage cryptographique est une fonction mathématique qui prend en entrée des données de taille arbitraire (un message, un fichier, etc.) et produit en sortie une chaîne de

caractères de taille fixe, appelée *empreinte* (*hash* en anglais). Ces fonctions doivent satisfaire des propriétés fondamentales [Katz and Lindell, 2007] :

1. **Résistance à la préimage (ou « sens unique »)** : Étant donné une empreinte h, il est computationnellement impossible de trouver un message m tel que $hash(m) = h$. Cette propriété protège les données originales.
2. **Résistance à la seconde préimage** : Étant donné un message m_1, il est impossible de trouver un message différent m_2 tel que $hash(m_1) = hash(m_2)$. Cette propriété garantit qu'un fichier ne peut être substitué à un autre sans changer son empreinte.
3. **Résistance aux collisions** : Il est impossible de trouver deux messages distincts m_1 et m_2 tels que $hash(m_1) = hash(m_2)$.

Dans le contexte des blockchains, les fonctions de hachage assurent plusieurs rôles cruciaux : elles permettent de chaîner les blocs les uns aux autres (d'où le terme *blockchain*), de vérifier rapidement l'intégrité des données, et d'implémenter des mécanismes de preuve de travail (*Proof of Work*) en exploitant leurs propriétés de pseudo-aléatoirité. Des algorithmes comme SHA-256[Nakamoto, 2008] (utilisé dans Bitcoin).

Signatures numériques et gestion des clés

Les signatures numériques sont l'équivalent électronique des signatures manuscrites, mais offrant des garanties de sécurité bien supérieures. Elles reposent sur la cryptographie asymétrique, qui utilise une paire de clés mathématiquement liées : une clé privée, gardée secrète par son propriétaire, et une clé publique, librement distribuable.

Le processus de signature et de vérification fonctionne comme suit [Stallings, 2017] :

1. **Signature** : Pour signer un message (par exemple, une transaction), l'émetteur génère une signature à l'aide de sa clé privée et du contenu du message lui-même. La signature est donc unique à la fois à l'émetteur et au message.
2. **Vérification** : Tout destinataire peut vérifier l'authenticité de la signature en utilisant la clé publique de l'émetteur et le message reçu. Si la vérification est positive, elle prouve deux choses de manière indissociable :
   - **Authentification** : Le message a bien été signé par le détenteur de la clé privée correspondante.
   - **Intégrité** : Le message n'a pas été altéré depuis sa signature.

Dans le contexte blockchain, les signatures numériques sont au cœur de l'autorisation des transactions. Seul le détenteur de la clé privée associée à une adresse (dérivée de la clé publique) peut signer une transaction pour dépenser les fonds qui y sont associés. Cela assure la **non-répudiation** : un utilisateur ne peut nier avoir effectué une transaction qu'il a signée. La gestion sécurisée des clés privées est donc un enjeu primordial, car leur perte équivaut à la perte des actifs numériques, et leur vol permet à un attaquant de les dépenser. L'algorithme ECDSA (Elliptic Curve Digital Signature Algorithm) est couramment utilisé dans de nombreuses blockchains pour son efficacité et sa sécurité [Antonopoulos, 2014].

La blockchain est une technologie de registre distribué (*Distributed Ledger Technology* ou DLT) qui enregistre les transactions de manière décentralisée, transparente, sécurisée et immuable. Son architecture repose sur une suite de blocs liés entre eux par des fonctions de hachage cryptographique, suivant le principe de chaînage temporel introduit par [Haber and Stornetta, 1990]. Chaque bloc contient un ensemble de transactions validées, garantissant la cohérence et l'intégrité du registre à travers l'ensemble du réseau.

1.4.2 Structure interne d'un bloc

Comme illustré dans la figure 1.4, la blockchain est constituée d'une séquence ordonnée de blocs, chacun contenant des informations essentielles permettant d'assurer la continuité, la traçabilité et la sécurité du registre distribué.

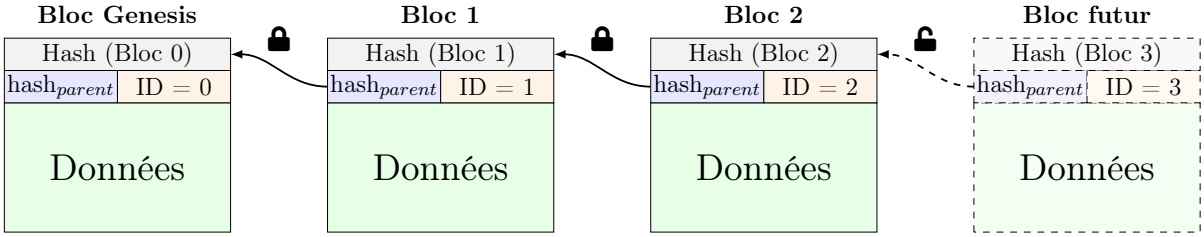


FIGURE 1.4 – Architecture schématique d'une blockchain. Chaque bloc contient un en-tête et un corps composé de transactions. La liaison cryptographique entre blocs assure l'intégrité et l'immutabilité de la chaîne.

Chaque bloc agit ainsi comme un maillon d'une chaîne d'enregistrements sécurisés : il référence son prédécesseur par un identifiant unique (le hachage) et renferme les données validées par le réseau. Cette structure enchaînée forme un registre distribué où chaque ajout dépend de la validité cryptographique de l'ensemble précédent, assurant cohérence et continuité dans le temps.

L'en-tête du bloc (*block header*). L'en-tête est la partie la plus critique du bloc : elle assure la connexion cryptographique entre les blocs successifs et garantit la cohérence globale de la chaîne. Dans la représentation de la figure 1.4, elle se compose de trois champs fondamentaux :

- **Le hachage du bloc précédent($hash_{parent}$).** Il s'agit d'une empreinte cryptographique unique (généralement calculée avec une fonction de hachage comme SHA-256) représentant le contenu du bloc antérieur. Ce champ lie le bloc courant à son prédécesseur et forme la « chaîne » de la blockchain. Toute altération d'un bloc antérieur modifierait son hachage et invaliderait ainsi tous les blocs suivants, rendant la falsification immédiatement détectable.
- **L'identifiant du bloc (*ID*).** Chaque bloc possède un numéro ou identifiant unique ($ID = 0, 1, 2, \ldots$) reflétant son ordre d'insertion dans la chaîne. Bien qu'il ne soit pas indispensable à la sécurité cryptographique, cet identifiant facilite le suivi séquentiel, la gestion des index et la vérification de la cohérence du registre par les nœuds du réseau.
- **Le hachage du bloc courant (`Hash`).** Ce champ correspond à l'empreinte cryptographique du bloc lui-même, calculée à partir de l'ensemble des éléments de son en-tête et de ses données. Il agit comme une signature numérique immuable, garantissant que tout changement, même minime, dans les données du bloc entraîne un hachage totalement différent. Ce mécanisme de sensibilité aux modifications, propre aux fonctions de hachage cryptographiques, constitue le pilier de l'intégrité de la blockchain.

Le corps du bloc (*Données*). Le corps contient les données transactionnelles validées, représentées dans la figure 1.4 sous la mention `Données`. Dans une blockchain publique telle que Bitcoin ou Ethereum, il s'agit d'un ensemble de transactions signées numériquement, organisées de manière à permettre une vérification rapide de leur intégrité. Les systèmes plus avancés utilisent des structures arborescentes comme les arbres de Merkle [Merkle, 1987], dans lesquels chaque transaction est associée à un hachage, permettant de vérifier la présence d'une transaction

sans parcourir l'ensemble du bloc. Ainsi, le corps du bloc est à la fois un espace de stockage et un ancrage cryptographique assurant la cohérence interne du registre.

L'interdépendance entre ces composantes garantit les propriétés de traçabilité, d'intégrité et d'immutabilité caractéristiques des blockchains. Le lien unidirectionnel entre blocs, symbolisé dans la figure par les flèches et les cadenas assure que chaque nouveau bloc dépend cryptographiquement du précédent.

1.4.3 Processus de construction et de validation d'un bloc

La création d'un nouveau bloc dans la chaîne suit un processus séquentiel rigoureux qui combine principes cryptographiques et mécanismes de consensus distribués. Ce processus peut légèrement différer selon le type de protocole utilisé (preuve de travail, preuve d'enjeu, tolérance aux fautes byzantines, etc.), mais les étapes fondamentales demeurent similaires :

1. **Propagation des transactions** : Les utilisateurs créent et signent numériquement des transactions à l'aide de leurs clés privées, puis les diffusent sur le réseau. Ces transactions sont reçues par les nœuds validateurs qui les stockent temporairement dans une mémoire d'attente.
2. **Vérification et regroupement** : Les nœuds participants vérifient la validité des transactions : signatures, solvabilité, format et les regroupent dans un bloc candidat. Le contenu de ce bloc est ensuite résumé par une racine de Merkle incluse dans son en-tête.
3. **Validation du bloc selon le mécanisme de consensus** : Selon le protocole adopté :
   - Dans la **preuve de travail (*Proof of work)*, PoW)**, les nœuds doivent résoudre un problème computationnel difficile, trouver un nonce produisant un hachage conforme à une cible de difficulté.
   - Dans la **preuve d'enjeu (*Proof of stake*,PoS)**, un ou plusieurs validateurs sont sélectionnés de manière pseudo-aléatoire en fonction de leur mise en jeu (*stake*) pour proposer et valider le bloc.
   - Dans les **protocoles byzantins** (*Practical Byzantine Fault Tolerance*(PBFT), Tendermint, HotStuff, etc.), les nœuds échangent des messages signés et votent pour atteindre un accord collectif, sans résoudre de problème computationnel.
4. **Propagation et validation du bloc** : Une fois le bloc produit, il est diffusé à l'ensemble du réseau. Chaque nœud vérifie la validité du bloc (intégrité du hachage, validité des transactions, respect du protocole de consensus) avant de l'ajouter à sa propre copie de la blockchain.
5. **Ajout à la chaîne** : Après validation par la majorité des nœuds selon les règles du consensus, le bloc est ajouté à la chaîne existante. Ses transactions deviennent alors permanentes, immuables et publiquement vérifiables.

Cette architecture garantit à la blockchain des propriétés fondamentales : décentralisation (absence d'autorité centrale), transparence (registre partagé), immutabilité (impossibilité de modifier rétroactivement les blocs) et **résilience** face aux fautes ou comportements malveillants.

1.4.4 Le Concept de Cross-Chain et l'Interopérabilité

Le *cross-chain* désigne l'établissement d'un lien d'interopérabilité et de connexion entre au moins deux blockchains distinctes. L'objectif de cette connexion est de permettre l'exécution de transactions et d'échanges entre ces différentes chaînes. Pour réaliser cette connexion, la

littérature suggère l'intégration d'un ou plusieurs nœuds d'une blockchain dans la deuxième chaîne.

L'interconnexion d'un ensemble de blockchains est un défi, principalement à cause de leur **hétérogénéité**. Cette diversité découle notamment des différences dans les *protocoles de consensus* (tels que la Preuve de Travail, ou les protocoles de type *Byzantine Fault Tolerance* et dans le type de blockchains à relier (publiques ou privées).

Problématiques de l'Atomicité et de la Cohérence

Le passage d'une architecture mono-chaîne à une architecture *cross-chain* nécessite la réévaluation ou l'adaptation des propriétés **ACID** (Atomicité, Cohérence, Isolation, Durabilité), souvent d'une manière "faible", pour répondre aux défis spécifiques posés par cette technologie.

Atomicité L'atomicité, qui est traditionnellement basée sur le principe du "tout ou rien" pour garantir la non-perte d'actifs, n'est plus strictement valable dans le contexte inter-chaînes [Mohanty et al., 2022]. Dans le cas d'une transaction inter-chaînes, celle-ci peut être considérée comme deux transactions séparées, chacune exécutée dans une seule blockchain [Robinson, 2020]. L'atomicité stricte est difficile à garantir, car le processus implique des étapes complexes pour l'échange de données et l'atteinte d'un consensus entre les nœuds de différentes blockchains. Cela peut entraîner une exécution partielle ou une interruption, laissant un état intermédiaire. Pour gérer ces situations, une approche d'**atomicité faible** a été introduite.

Cohérence En *cross-chain*, la **cohérence** est définie comme la possibilité que les transactions et les mises à jour associées soient soit effectuées, soit rejetées par toutes les blockchains impliquées [Robinson, 2020]. Compte tenu de l'augmentation des données et des risques de sécurité liés à leur transfert, il est souvent préférable d'effectuer les calculs dans chaque blockchain et de ne transférer que les résultats [Zhang et al., 2022] . Cependant, cette approche pose un problème de cohérence, car si les blockchains n'ont pas accès aux calculs internes des autres, elles ne peuvent pas juger de la cohérence de ces calculs entre elles.

Schémas d'Interopérabilité des Chaînes Hétérogènes

La littérature identifie cinq types de schémas pour connecter des blockchains hétérogènes [Wang et al., 2020, Deng et al., 2018, Ou et al., 2022, Mao et al., 2022] :

1. Le notaire (*Notary Scheme*)
2. Chaîne latérale/Relais (*Sidechains/Relays*)
3. *Hash Timelock Contract* (*HTLC*)
4. *Distributed Private Key Control* (*DPKC*)
5. Couplage étroit (*Tight Coupling*)

Certains de ces schémas, comme le *notaire* et *DPKC*, introduisent un **tiers de confiance**, ce qui affaiblit la **décentralisation** du système et introduit de nouvelles problématiques de sécurité.

Sidechains et Relays

La technique *sidechains/relays* est une solution au *cross-chain*.

— Une *chaîne de relais* (*relay chain*) agit comme un pont qui relaie deux autres chaînes [Ou et al., 2022, Pang, 2020].

— Les *sidechains* (chaînes latérales) ajoutent des blockchains secondaires à la chaîne principale pour apporter de nouvelles fonctionnalités ou améliorer son fonctionnement sans la manipuler directement. L'idée des *sidechains* est apparue en 2012 et un document à ce sujet a été publié en 2014 par l'équipe de développeurs de Bitcoin [Back et al., 2014].

Protocole Hash Timelock Contract (HTLC)

Le protocole *HTLC* est spécifiquement applicable dans le cas d'*échanges d'actifs (Swap)* [Shadab et al., 2020, Bennink and Everts, 2018, Herlihy, 2018]. Ce mécanisme se déroule en plusieurs étapes impliquant l'échange d'un secret s et de son hash h :

1. Une partie (A) génère le secret s et son hash h.
2. Les deux parties publient des contrats sur la blockchain de l'autre, intégrant h et un délai d'exécution (Δ).
3. L'exécution des contrats dépend de la réception du secret s avant l'expiration du délai, sinon les actifs sont remboursés et les sous-transactions sont annulées.

Dans ce cadre, l'**atomicité** est définie par le transfert du secret s entre les blockchains dans des délais précis, faute de quoi il y a annulation et remboursement. Une limite majeure de cette technique est le **coût élevé** de son implémentation, nécessitant deux contrats pour un seul échange.

Protocoles *Notary Scheme*

Les protocoles basés sur le schéma du *notaire* [Ou et al., 2022, Mao et al., 2022] introduisent un intermédiaire dont le rôle est d'assurer la collecte, la vérification et la confirmation des transactions inter-blockchains. Dans la littérature, on trouve 3 protocoles basés sur le schéma du "notaire" :

1. *Notaire centralisé* : Un seul notaire contrôle la validation, entraînant la perte de la centralisation.
2. *Mécanisme de signature multi-notaire* : Plusieurs notaires indépendants participent au consensus. Un sous-ensemble est sélectionné aléatoirement, et un pourcentage de signatures est requis pour valider les transactions.
3. *Mécanisme de signature distribué* : Une clé unique est partitionnée et distribuée sur une sélection de notaires. Un pourcentage de consensus permet de recomposer la clé pour la signature et l'exécution des transactions. Ce protocole exige une confiance envers les notaires.

Contrôle des Clés Privées Distribuées (DPKC)

Le protocole *DPKC* est présenté comme une version plus sécurisée et décentralisée que le mécanisme de signature distribué.

— Il partitionne la clé privée et distribue les fragments sur tous les nœuds.

— Les utilisateurs détiennent une partie de la clé, et chaque nœud détient une partie de la clé privée de l'utilisateur.

— Ces manipulations permettent de **décentraliser le système**. Si un utilisateur subit une attaque, ses transactions ne s'effectuent pas car il ne détient qu'une partie de sa clé privée.

1.5 Topologies des réseaux blockchain et leurs implications

La structure du réseau sous-jacent à une blockchain influence profondément ses propriétés de sécurité, de performance et de gouvernance. Contrairement à une vision homogène, les blockchains adoptent différentes topologies qui reflètent leurs objectifs de conception et leurs modèles de confiance.

1.5.1 Classification par modèle d'accès

La diversité des cas d'usage des technologies blockchain a conduit à l'émergence de différents modèles architecturaux, principalement distingués par leurs politiques de contrôle d'accès et de gouvernance. Cette classification, qui oppose les approches sans permission (*permissionless*) aux systèmes permissionnés (*permissioned*), définit des compromis fondamentaux entre ouverture, performance et confidentialité.

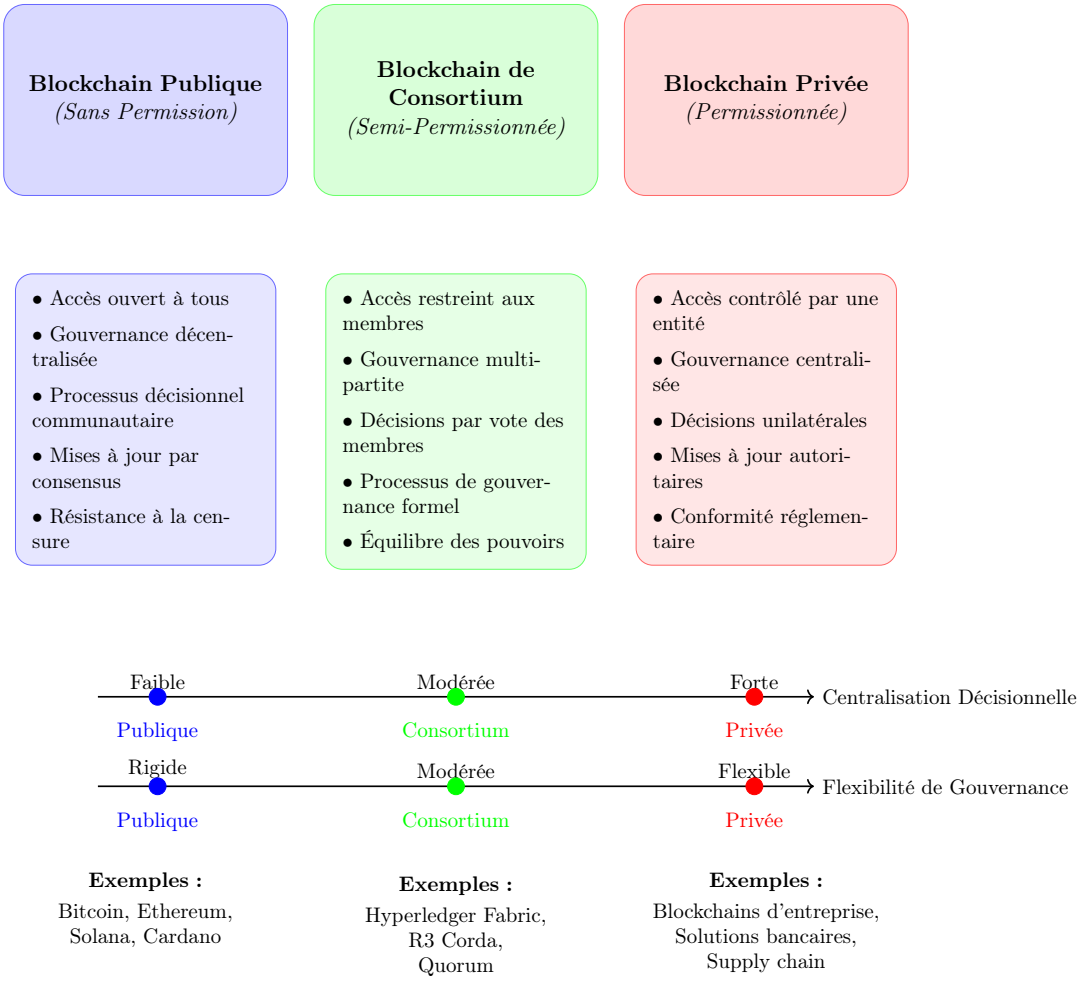


FIGURE 1.5 – Classification des blockchains selon leur modèle d'accès et leur structure de gouvernance

Comme illustré dans la Figure 1.5, on distingue principalement trois types de blockchains selon leurs politiques d'accès :

Blockchains publiques sans permission

Les blockchains publiques, telles que Bitcoin et Ethereum, sont caractérisées par :

— **Accès ouvert** : Tout participant peut rejoindre le réseau sans autorisation préalable

— **Anonymat pseudonyme** : Les identités sont représentées par des adresses cryptographiques

— **Décentralisation forte** : Aucune entité unique ne contrôle le réseau

— **Résistance à la censure** : Les transactions valides ne peuvent être empêchées

Ce modèle maximise la neutralité et la résilience, mais au prix de limitations en termes de performance et de confidentialité des données.

Blockchains privées permissionnées

À l'opposé, les blockchains privées restreignent l'accès à un ensemble prédéfini de participants de confiance. Elles présentent des caractéristiques distinctes :

— **Contrôle d'accès strict** : Seules les entités autorisées peuvent participer au consensus

— **Identités connues** : Les participants sont identifiés et authentifiés

— **Performance élevée** : L'absence de mécanismes coûteux comme la preuve de travail permet des débits transactionnels importants

— **Gouvernance centralisée** : Une entité ou un consortium contrôle les règles du réseau

Ce modèle convient particulièrement aux applications d'entreprise où la confidentialité et la conformité réglementaire sont prioritaires.

Blockchains de consortium

Les blockchains de consortium représentent un compromis entre les deux modèles précédents. Un ensemble prédéfini d'organisations collabore pour opérer le réseau, avec des mécanismes de consensus qui peuvent varier selon la configuration. Cette approche est particulièrement adaptée aux écosystèmes industriels où plusieurs acteurs doivent collaborer tout en maintenant une certaine indépendance.

1.5.2 Implications sur la sécurité et la performance

Le choix de la topologie a des conséquences directes sur les propriétés du système :

Sécurité et modèle de menace : Les blockchains publiques doivent résister à des adversaires puissants pouvant contrôler une fraction significative des ressources du réseau. Les mécanismes de consensus comme la preuve de travail sont conçus pour rendre économiquement prohibitif le contrôle du réseau. À l'inverse, les blockchains permissionnées reposent sur la réputation et l'identité des participants comme garanties de sécurité.

Performance et évolutivité : Le compromis entre décentralisation et performance est fondamental. Les blockchains publiques, avec leurs nombreux nœuds distribués géographiquement, souffrent de limitations en termes de débit transactionnel et de latence. Les blockchains permissionnées, avec un nombre réduit de validateurs, peuvent atteindre des performances comparables aux systèmes centralisés traditionnels.

Confidentialité : La transparence des blockchains publiques pose des défis importants pour la confidentialité des données. Diverses techniques cryptographiques comme les preuves à divulgation nulle de connaissance (zk-SNARKs) [Guan et al., 2020] ont été développées pour adresser ces limitations, mais elles introduisent des complexités supplémentaires.

Cette analyse des topologies met en lumière la diversité des approches blockchain et la nécessité d'adapter l'architecture au cas d'usage spécifique.

1.6 Conclusion

Ce premier chapitre a retracé l'émergence de la blockchain comme aboutissement de plusieurs décennies de recherche en informatique distribuée, en cryptographie et en théorie des jeux. Nous avons montré comment cette technologie synthétise des concepts fondamentaux tels que les chaînes de hachage horodatées, la tolérance aux fautes byzantines et les preuves de travail en un système opérationnel permettant la coordination décentralisée sans autorité centrale.

L'analyse architecturale a mis en évidence que la blockchain s'appuie sur des principes éprouvés des systèmes distribués, notamment la réplication de machines à états (SMR) et les propriétés de *safety* et *liveness*, tout en introduisant des spécificités liées à son environnement ouvert et potentiellement hostile. Les mécanismes cryptographiques, en particulier les fonctions de hachage et les signatures numériques, garantissent l'intégrité et l'authentification des données, tandis que les différentes topologies de réseau (publiques, privées, consortium) reflètent des compromis distincts entre décentralisation, performance et contrôle d'accès.

Cependant, la robustesse et la cohérence de ces systèmes reposent ultimement sur leur mécanisme de consensus, qui détermine la façon dont les nœuds s'accordent sur l'état du registre partagé. Le paysage des protocoles de consensus est aujourd'hui dominé par deux grandes familles : les protocoles de type PoW et PoS, d'une part, et les protocoles byzantins BFT, d'autre part.

Dans le chapitre suivant, nous présenterons une analyse comparative de ces deux familles de protocoles, en mettant en lumière leurs propriétés, leurs garanties de sécurité et leurs compromis respectifs. Comme annoncé, ce travail de thèse se concentrera principalement sur la famille des protocoles BFT, dont nous étudierons les fondements théoriques, les variantes modernes et les applications concrètes. Cette orientation répond à la nécessité de concevoir des systèmes à la fois performants, évolutifs et résistants aux attaques, en particulier dans des environnements exigeants où la confiance entre participants est limitée.

Chapitre 2

Consensus BFT : sélection des validateurs et dynamiques de vote

2.1 Introduction

Si les piliers fondamentaux que sont les systèmes distribués, la cryptographie et les propriétés de fiabilité constituent le socle architectural de la blockchain, c'est le consensus qui en incarne le principe vital. Ce mécanisme est l'alchimie qui transforme une nébuleuse de nœuds méfiants en une source unique et incontestable de vérité. Le chapitre précédent a dressé le cadre théorique contraignant de cet exercice, exposant les limites fondamentales énoncées par les théorèmes FLP et CAP : l'impossibilité d'un consensus déterministe parfaitement asynchrone et l'inévitable arbitrage entre cohérence, disponibilité et tolérance aux partitions. C'est dans ce paysage théorique exigeant que les protocoles de consensus doivent être conçus et évalués.

Ce chapitre a pour ambition de cartographier le vaste territoire des mécanismes de consensus qui animent les blockchains, en établissant une passerelle solide entre la théorie établie et les implémentations pratiques. Nous débuterons par un retour aux sources, avec le problème emblématique des Généraux Byzantins, qui formalise la difficulté du consensus en environnement hostile. Nous en approfondirons les garanties formelles, sécurité, vivacité et finalité, qui servent de boussole pour évaluer la robustesse de tout protocole.

Sur ces fondations, nous présenterons un aperçu structurant des protocoles, les classant en deux grandes familles aux philosophies antagonistes : les mécanismes *Proof-based*, qui reposent sur une preuve cryptographique d'engagement de ressources (calcul, réputation), et les mécanismes *Vote-based*, qui s'appuient sur une coordination explicite et des échanges de messages structurés. Cette dichotomie reflète un choix architectural fondamental aux implications profondes sur la sécurité, la performance et la gouvernance des systèmes.

Le consensus est la pierre angulaire de toute blockchain, transformant une collection de nœuds méfiants en un système cohérent et unique de vérité. Ce chapitre s'articule autour de trois axes majeurs. Les sections 2.3 et 2.2 établissent les fondements théoriques du consensus distribué et présentent une analyse comparative des grandes familles de protocoles, respectivement *Vote-based* et *Proof-based*, en examinant leurs compromis fondamentaux.La section 2.5 présente la problématique centrale de ce chapitre, identifiant les défis majeurs liés à la rigidité des mécanismes de sélection et à la détection des nœuds déviants. La section 2.5.1 se concentre sur l'optimisation de la sélection des validateurs, analysant les arbitrages entre performance et sécurité. Enfin, la section 2.5.2 aborde les dynamiques de vote sous l'angle des modèles d'opinion, ouvrant des perspectives interdisciplinaires pour la conception de systèmes de consensus plus robustes et

adaptatifs.

Ce chapitre vise à établir les bases théoriques et pratiques nécessaires à la compréhension des enjeux du consensus distribué. L'analyse systématique qui y est menée constituera le socle sur lequel s'appuieront les contributions présentées dans les chapitres suivants.

2.2 Consensus à preuve (Proof-based)

Les protocoles de consensus *proof-based* reposent sur un critère objectif permettant de sélectionner le nœud qui proposera le prochain bloc ou validera une transaction. Ce paradigme s'appuie sur la preuve cryptographique d'engagement d'une ressource spécifique, créant une barrière économique contre les comportements malveillants. Trois mécanismes majeurs dominent cette approche : *Proof of Work (PoW)*, *Proof of Stake (PoS)* et *Proof of Reputation (PoR)*. Chacun présente des avantages spécifiques, ainsi que des limitations et risques propres, qui influencent la sécurité, la performance et la décentralisation du réseau.

2.2.1 Proof of Work

Le mécanisme PoW, popularisé par Bitcoin [Nakamoto, 2008] et repris par Ethereum [Wood et al., 2014], constitue le modèle de référence des consensus à preuve. Il repose sur la résolution de puzzles cryptographiques asymétriques : la vérification d'une solution est rapide, mais la recherche d'une solution correcte exige un effort computationnel très élevé. La sécurité du protocole découle directement de ce coût de calcul, rendant économiquement prohibitive toute tentative de modification de l'historique des transactions.

La principale limitation du PoW réside dans cette exigence computationnelle, qui entraîne une consommation énergétique élevée.

Cependant, le PoW présente également des vulnérabilités fondamentales, notamment vis-à-vis de la centralisation et des attaques stratégiques :

— **Attaque à** 51% : Cette attaque survient lorsqu'un acteur ou un cartel de mineurs contrôle plus de 50% de la puissance de calcul totale du réseau. Une telle domination permet de créer une chaîne alternative plus longue, réorganisant la blockchain et invalidant des transactions confirmées, ouvrant la voie aux doubles dépenses [Aponte-Novoa et al., 2021]. La faisabilité économique de cette attaque dépend du coût d'acquisition de la puissance de calcul nécessaire.

— **Selfish Mining** : Dans cette stratégie, un comité de minage malveillant découvre un bloc mais ne le diffuse pas immédiatement. Il continue à miner secrètement sur sa chaîne privée tandis que le reste du réseau travaille sur l'ancienne chaîne publique. Lorsque l'écart devient suffisant, le comité révèle sa chaîne, rendant invalide le travail des mineurs honnêtes [Eyal, 2015]. Cette tactique permet au comité d'augmenter artificiellement sa part de récompenses au-delà de sa puissance de calcul réelle, favorisant une centralisation progressive du réseau.

En résumé, la sécurité du PoW repose entièrement sur l'impossibilité économique d'acquérir la majorité de la puissance de calcul, un principe qui devient discutable avec l'émergence de grands comités miniers et la concentration du pouvoir de minage.

2.2.2 Proof of Stake

Le Proof of Stake (PoS) représente une évolution majeure des mécanismes de consensus, remplaçant la dépense énergétique du Proof of Work par un engagement économique direct.

Dans ce paradigme, formalisé initialement dans Blackcoin Vasin et al. [2014] et déployé à grande échelle avec Ethereum 2.0, la validation des blocs est confiée à des acteurs ayant immobilisé une quantité significative de jetons en garantie.

Cette approche introduit cependant des vulnérabilités conceptuelles distinctes :

— **Double validation opportuniste (Nothing-at-Stake)** : En cas de fork, l'absence de coût marginal incite les validateurs à approuver toutes les branches simultanément, compromettant la résolution du conflit.

— **Réécriture historique (Long-Range Attack)** : La possibilité de recréer une chaîne alternative à partir d'un point ancien de l'histoire, rendue possible par l'acquisition d'anciennes clés de validation. Cette vulnérabilité menace particulièrement les nouveaux nœuds rejoignant le réseau Deirmentzoglou et al. [2019].

Les protocoles PoS modernes combattent ces risques par des mécanismes cryptéconomiques sophistiqués, incluant la confiscation des fonds en jeu (*slashing*) pour punir les validateurs malveillants.

2.2.3 Proof of Reputation : La preuve par le capital de confiance

Le Proof of Reputation (PoR) constitue une alternative novatrice particulièrement adaptée aux environnements de consortium où l'identité des participants est partiellement connue. Contrairement aux mécanismes basés sur des ressources physiques ou économiques, le PoR utilise la réputation numérique comme garantie principale.

Cette approche, conceptualisée dans Repucoin Yu et al. [2019] et affinée par Zhuang et al. Zhuang et al. [2019], transforme l'historique comportemental des nœuds en capital de confiance. Les métriques de réputation, mises à jour dynamiquement selon des algorithmes tels que celui proposé par Qiu et al. Qiu et al. [2022], créent un système d'incitations où les validateurs préservent scrupuleusement leur intégrité pour maintenir leur influence dans le réseau.

Le PoR introduit cependant des vulnérabilités spécifiques :

— **Collusion stratégique** : Des validateurs peuvent former des coalitions pour manipuler artificiellement leurs scores de réputation, créant des oligopoles qui détournent le mécanisme de consensus à leur avantage.

— **Subjectivité des métriques** : La définition et le calcul de la réputation introduisent nécessairement une part d'arbitraire, pouvant conduire à des biais systémiques et à des attaques par manipulation des paramètres d'évaluation.

Ces limitations rendent le PoR particulièrement vulnérable dans les contextes où la confiance initiale entre les participants n'est pas solidement établie, bien qu'il reste optimal pour les écosystèmes enterprise où les relations de confiance préexistent et peuvent être formalisées.

2.3 Consensus à vote (Vote-based)

Les mécanismes de consensus par vote s'appuient sur une coordination explicite et des échanges de messages structurés entre validateurs identifiés pour parvenir à un accord sur l'état du registre. Contrairement aux approches *proof-based* qui reposent sur une compétition ouverte, les systèmes *vote-based* fondent leur sécurité sur des protocoles de communication formels garantissant la tolérance aux fautes byzantines. Le modèle BFT constitue un cadre générique de tolérance aux fautes byzantines, sans toutefois spécifier la manière dont les échanges d'informations entre nœuds sont réalisés pour atteindre le consensus.

2.3.1 Le problème des généraux byzantins :

Cette section établit les bases théoriques indispensables à la compréhension des protocoles de consensus blockchain. Elle présente d'abord le problème fondateur des généraux byzantins, qui formalise la difficulté du consensus en environnement distribué hostile. Ensuite, elle détaille les garanties théoriques, sécurité, vivacité et finalité qui permettent d'évaluer rigoureusement la robustesse des différents protocoles.

Le problème des généraux byzantins constitue la formulation classique de la tolérance aux fautes dans les systèmes distribués. Introduit par Lamport et al. [Lamport et al., 1982], il illustre la difficulté fondamentale d'atteindre un accord entre plusieurs entités ("les généraux") lorsqu'une partie d'entre elles peut adopter un comportement défaillant ou malveillant (les "traîtres").

Dans sa version originale, plusieurs généraux d'une armée byzantine assiègent une ville. Chacun doit décider d'attaquer ou de battre en retraite, mais la réussite de l'opération exige que tous les généraux loyaux adoptent la même décision. La communication ne se fait que par messages, et certains généraux peuvent être des traîtres, envoyant volontairement des informations contradictoires.

Ce problème illustre les conditions nécessaires pour parvenir à un consensus distribué en présence de fautes arbitraires (dites *fautes byzantines*). Lamport et al. [Lamport et al., 1982] démontrent que :

— pour atteindre l'accord en présence de f généraux traîtres, le système doit compter au moins $3f + 1$ participants ;

— un protocole BFT peut donc tolérer au maximum un tiers des nœuds défaillants, soit $f < \frac{n}{3}$.

Cette borne théorique dépend fortement du modèle de communication. En particulier, l'existence ou non de garanties temporelles (systèmes synchrones, asynchrones ou partiellement synchrones) conditionne la possibilité d'obtenir des propriétés comme la *liveness*.

Le problème des généraux byzantins a ouvert la voie à la formalisation des notions de *safety* et de *liveness*, qui constituent encore aujourd'hui les deux piliers de l'analyse des protocoles de consensus.

Dans le domaine des blockchains, ce problème sert de *base conceptuelle* : chaque validateur peut être vu comme un "général", et le consensus doit être atteint même si certains participants agissent de manière déviante (censure, collusion, fork). Les protocoles modernes tels que PBFT [Castro and Liskov, 1999], Tendermint [Buchman et al., 2018] ou HotStuff [Yin et al., 2019] s'inscrivent directement dans cette lignée, en adaptant les résultats théoriques Lamport et al. à des environnements réels, où scalabilité et performance sont aussi cruciales que la robustesse face aux fautes byzantines.

2.3.2 Garanties théoriques (sécurité, vivacité, finalité) :

Les notions de sûreté (*safety*) et de vivacité (*liveness*), introduites dans le chapitre précédent, permettent d'évaluer la robustesse des protocoles de consensus blockchain face aux pannes et comportements malveillants. La sûreté garantit qu'aucune valeur incorrecte n'est décidée : tous les nœuds honnêtes adoptent la même valeur et toute valeur décidée provient d'une proposition légitime. La vivacité, quant à elle, assure la progression du système, de sorte que toutes les transactions valides finissent par être intégrées.

La finalité décrit l'irréversibilité des décisions. Dans les systèmes *Proof of Work (PoW)*, la finalité est probabiliste : une transaction devient de plus en plus irréversible à mesure que de nouveaux blocs sont ajoutés par-dessus, la probabilité de réorganisation décroissant de façon

exponentielle [Nakamoto, 2008]. Dans les protocoles BFT ou de preuve d'enjeu, la finalité est absolue : un bloc validé ne peut plus être modifié ni réorganisé. Cette irréversibilité découle du processus de validation collective, où une majorité qualifiée de validateurs honnêtes (souvent au moins deux tiers) doit approuver chaque bloc avant qu'il ne soit considéré comme finalisé. Ainsi, une fois que les votes nécessaires sont obtenus, le bloc devient immuable, ce qui élimine toute incertitude sur l'état du registre et garantit une cohérence forte à tous les nœuds du réseau.

Ces garanties sont soumises à des dépendances fondamentales. Le compromis sécurité/vivacité (théorème CAP) [Brewer, 2000, Gilbert and Lynch, 2002] impose des choix architecturaux selon les priorités du réseau. Le théorème FLP montre que dans un système purement asynchrone, aucune stratégie déterministe ne peut garantir à la fois sûreté et vivacité. Enfin, les modes de synchronisation influencent directement la décentralisation et la vitesse de validation.

Selon le paradigme de consensus, l'implémentation de ces garanties varie. Les protocoles PoW offrent une sécurité probabiliste émergeant de la compétition minière et une vivacité assurée par l'ajustement dynamique de la difficulté, tandis que les protocoles BFT garantissent une sûreté absolue et une finalité immédiate sous l'hypothèse que moins d'un tiers des validateurs sont byzantins Castro and Liskov [1999].

2.3.3 Protocole de consensus PBFT et séquence d'exécution

L'architecture illustrée par la Figure 2.1 schématise le déroulement idéal d'une exécution du protocole de consensus PBFT [Castro and Liskov, 1999]. Ce protocole constitue une approche déterministe de consensus dans les environnements distribués tolérant les fautes byzantines. Fondé sur le principe de la duplication de machines à états [Schneider, 1990], il assure que l'ensemble des nœuds honnêtes exécute les requêtes dans le même ordre, garantissant ainsi la cohérence forte du système (modèle CP selon le théorème CAP) [Ongaro and Ousterhout, 2014, Fischer et al., 1985].

Le protocole PBFT est largement adopté dans les blockchains privées et de consortium en raison de son efficacité énergétique et en raison de son déterminisme comparée aux approches probabilistes (comme Pow). Pour tolérer jusqu'à f nœuds défectueux ou malveillants, le réseau doit comporter au minimum $N \geq 3f + 1$ nœuds [Castro and Liskov, 1999]. Bien que le protocole soit robuste, il est communicationnellement coûteux, nécessitant un nombre de messages quadratique $\mathcal{O}(n^2)$ selon le nombre de validateurs n [Liu et al., 2021, Qi et al., 2020].

Le processus implique un **nœud primaire** (coordinateur) et plusieurs **nœuds secondaires** (backups), qui interagissent à travers cinq phases séquentielles et authentifiées, illustrées sur la Figure 2.1.

1. **Phase `REQUEST` : soumission du client** L'exécution débute lorsque le **client** envoie une requête (`Request`) contenant le message de modification m, un horodatage t et son identité c au nœud primaire [Castro and Liskov, 1999]. Cette étape correspond à la première interaction entre le client et le réseau, initiant la demande de consensus.
2. **Phase `PRE-PREPARE` : diffusion du primaire** Le nœud primaire attribue un numéro de séquence n à la requête et la diffuse via un message `Pre-Prepare`. Ce message contient le numéro de vue v, le condensat cryptographique $d(m)$ et le numéro n. Les sauvegardes n'acceptent ce message que si $d(m)$ correspond bien à m et si n est valide. Cette phase marque la première diffusion en multicast du protocole.
3. **Phase `PREPARE` : établissement de l'ordre total** Les nœuds sauvegardes envoient ensuite un message `Prepare` (incluant v, n, $d(m)$ et leur identité i) à tous les autres nœuds. Un nœud est considéré comme *prepared* s'il reçoit au moins $2f$ messages `Prepare` concordants.

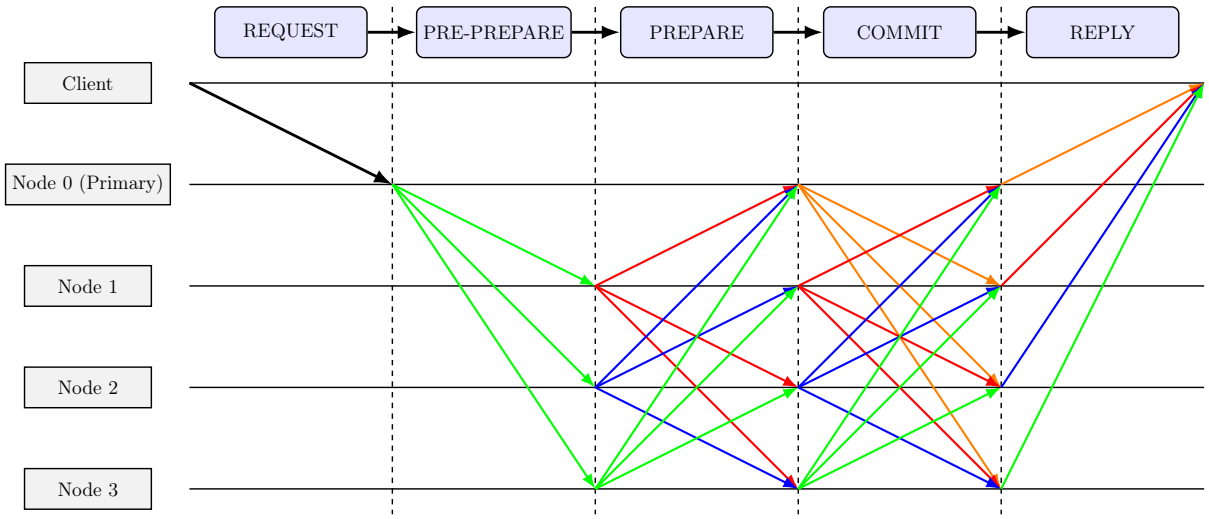


FIGURE 2.1 – Architecture en cinq phases du protocole PBFT. Les flèches indiquent les échanges de messages entre les nœuds et le client.

Cette étape assure un ordre total des requêtes, garantissant que tous les nœuds honnêtes appliquent les mêmes transactions dans la même séquence.

4. **Phase `COMMIT` : validation atomique** Une fois l'état `Prepared` atteint, chaque nœud diffuse un message `Commit`. Lorsqu'un nœud reçoit $2f + 1$ messages `Commit` valides (y compris le sien), il considère la transaction comme validée localement (`Commit-local`). Cette phase garantit l'atomicité et la durabilité de la transaction avant son exécution effective.
5. **Phase `REPLY` : confirmation au client** Chaque nœud envoie ensuite un message `Reply` au client, incluant la réponse r et l'horodatage t. Le client accepte la transaction uniquement s'il reçoit au moins $f+1$ réponses identiques, assurant ainsi la tolérance à un comportement byzantin du primaire. Ce mécanisme de majorité garantit que même si certains nœuds agissent de manière malveillante, la décision finale reste correcte.

Ainsi, la Figure 2.1 illustre la nature fortement interactive et redondante du protocole PBFT, où la sécurité repose sur la majorité honnête et la synchronisation stricte des échanges entre réplicas. Cette redondance communicationnelle permet d'assurer la cohérence forte, la tolérance aux fautes byzantines, et la détermination finale de chaque opération dans les blockchains autorisées.

2.3.4 Protocoles dérivés modernes : HotStuff et Tendermint

L'évolution des protocoles de consensus tolérants aux fautes byzantines a donné naissance à une nouvelle génération de mécanismes de réplication de machines à états SMR plus efficaces et plus simples à déployer. Ces protocoles, tout en conservant la robustesse théorique des algorithmes fondateurs tels que PBFT, cherchent à améliorer la scalabilité, la latence et la simplicité du changement de vue. Un mécanisme de changement de vue (*view change*) est crucial pour assurer la vivacité (*liveness*) d'un protocole BFT en cas de défaillance du leader. Dans les approches classiques comme PBFT, ce mécanisme constitue un protocole auxiliaire complexe, nécessitant une communication quadratique $\mathcal{O}(n^2)$ pour collecter les preuves de l'échec du leader avant de pouvoir installer un nouveau primaire.

Les protocoles de consensus tolérants aux fautes byzantines opèrent généralement dans le modèle de communication partiellement synchrone introduit par Dwork et al. [Dwork et al., 1988], où le réseau peut être asynchrone avant un certain temps de stabilisation global (GST, *Global Stabilization Time*).

Parmi ces protocoles modernes, **Tendermint** et **HotStuff** se distinguent par leur conception modulaire, leur capacité à maintenir la cohérence malgré des environnements ouverts (permissioned ou permissionless) et leur compatibilité avec des architectures de blockchain. Tous deux reposent sur un modèle de **consensus BFT dirigé par un leader**, dans lequel un nœud propose un bloc et les autres valident collectivement son engagement.

Tendermint : finalité déterministe et simplification du consensus.

Le protocole Tendermint [Buchman, 2016] a été conçu pour résoudre le problème du consensus dans la blockchain sans recourir à la preuve de travail *PoW*. Il élimine ainsi la consommation énergétique excessive et la latence élevée associées au minage.

Tendermint utilise un protocole basé sur des tours successifs (*rounds*) pour déterminer le prochain bloc à chaque hauteur de la chaîne de blocs. Le processus fait intervenir des validateurs qui participent en diffusant des signatures cryptographiques, ou votes, pour se mettre d'accord sur le bloc suivant. Chaque tour est structuré autour de trois étapes principales : `Propose`, `Prevote` et `Precommit`. Ces étapes, ainsi que les étapes spéciales `Commit` et `NewHeight`, composent le protocole.

1. **Étape Propose.** Chaque tour est associé à un proposeur désigné (choisi en général par un algorithme round-robin [Buchman, 2016] en fonction de sa puissance de vote). Au début de cette étape, le proposeur diffuse une proposition de bloc à ses pairs via *gossip*. Le protocole de diffusion *gossip* [Buchman et al., 2018](ou épidémique) permet une dissémination rapide et résiliente de l'information dans un réseau pair-à-pair. Chaque nœud relaie les messages reçus à un sous-ensemble de pairs choisis aléatoirement, garantissant qu'en $O(\log n)$ étapes, la proposition atteint presque tous les participants, même en cas de défaillances partielles.

   Si le proposeur détient un verrou sur un bloc provenant d'un tour antérieur, il propose ce bloc verrouillé en y incluant une preuve de verrouillage (*proof-of-lock*). Cette preuve de verrouillage (*proof-of-lock*) sert à informer les autres validateurs que le bloc a déjà obtenu un quorum de pré-commits dans un tour précédent (c'est-à-dire le nombre minimal de votes nécessaires pour qu'une décision soit considérée valide, généralement $2f + 1$ sur n nœuds), garantissant ainsi que les nœuds ne voteront pas pour un bloc concurrent et préservant la cohérence du consensus.
2. **Étape Prevote.** Au début de cette étape, chaque validateur prend une décision.
   - S'il est verrouillé (*locked*) sur un bloc proposé d'un tour précédent, il signe et diffuse un `prevote` pour ce bloc verrouillé.
   - Sinon, s'il a reçu une proposition acceptable pour le tour en cours, il signe et diffuse un `prevote` pour ce bloc proposé.
   - S'il n'a reçu aucune proposition (ou une proposition invalide), il signe un `prevote` spécial dit *nil prevote*.

   Il est important de noter qu'aucune opération de verrouillage n'a lieu durant cette étape `Prevote`.
3. **Étape Precommit (Verrouillage).** Au début de cette étape, chaque validateur prend à nouveau une décision.

- Si le validateur a reçu plus des deux tiers (2/3) des prevotes pour un bloc acceptable donné, il signe et diffuse un `precommit` pour ce bloc.
- Il se verrouille (*locks onto*) sur ce bloc et libère tout verrou antérieur. Un nœud ne peut avoir un verrou que sur un seul bloc à la fois.
- Si le nœud a reçu plus des 2/3 des *nil prevotes*, il se déverrouille simplement.

4. **Transition vers Commit.** À la fin de l'étape `Precommit`, si un nœud a reçu plus des 2/3 des precommits pour un bloc particulier, il entre dans l'étape spéciale `Commit`.

Chaque tour est incrémenté d'une petite durée fixe par rapport au tour précédent. Cela permet au réseau de finalement parvenir à un consensus dans un environnement partiellement synchrone, surmontant ainsi le résultat d'impossibilité FLP [Fischer et al., 1985].

- **Modèle et tolérance aux fautes.** Tendermint fonctionne dans un modèle partiellement synchrone. Il tolère jusqu'à un tiers de validateurs byzantins ($n = 3f + 1$), en accord avec les bornes classiques du consensus BFT.
- **Performances et limites.** Tendermint repose sur un **noyau synchrone**, où les propositions sont soumises à des intervalles prédéterminés dépendant d'une borne supérieure Δ du délai de propagation des messages. Cette contrainte empêche la réactivité optimiste (*optimistic responsiveness*), c'est-à-dire la capacité à progresser au rythme du délai réel du réseau. En cas de retard prolongé ou de perte de supermajorité, le protocole peut suspendre temporairement la progression, affectant ainsi la vivacité.
- **Complexité.** La complexité d'authentification est de l'ordre de $\mathcal{O}(n^2)$ par décision de consensus après le GST, pouvant être réduite à $\mathcal{O}(n)$ en utilisant des **signatures à seuil** (*threshold signatures*). En cas de défaillance de f leaders, le coût total peut atteindre $\mathcal{O}(fn^2)$ (ou $\mathcal{O}(fn)$ avec signatures agrégées).

Tendermint illustre un compromis clair entre **cohérence forte** et **disponibilité limitée**, conformément au théorème CAP : le système garantit l'unicité de la chaîne et la finalité stricte des blocs au détriment d'une certaine tolérance à la latence réseau.

HotStuff : linéarité et réactivité optimiste.

HotStuff [Yin et al., 2019], développé initialement pour la blockchain *Libra* [Baudet et al., 2019] de Meta, marque une étape clé dans la simplification et l'optimisation des protocoles BFT. C'est le premier protocole BFT partiellement synchrone à atteindre simultanément la linéarité en communication et la réactivité optimiste. La linéarité en communication signifie que le nombre total de messages échangés pour parvenir à un consensus croît linéairement avec le nombre de nœuds ($O(n)$), contrairement aux protocoles classiques comme PBFT dont la complexité est quadratique. Cette propriété améliore considérablement la scalabilité du protocole dans les grands réseaux. Quant à la réactivité optimiste, elle désigne la capacité du protocole à progresser au rythme réel du réseau plutôt qu'à attendre une borne supérieure prédéfinie du délai de communication (Δ). Autrement dit, dès que les messages requis sont effectivement reçus, le consensus peut avancer, ce qui réduit significativement la latence en conditions normales de réseau.

- **Structure à trois phases.** HotStuff adopte une structure à trois étapes : `prepare`, `pre-commit` et `commit` qui simplifie la logique du protocole et réduit la complexité du changement de leader. Cette triple phase garantit la cohérence du système même en cas de basculement fréquent des leaders.

— **Changement de vue linéaire (*Linear View Change*).** Après le GST, un leader correct n'envoie que $\mathcal{O}(n)$ authenticateurs pour parvenir à une décision de consensus, même lors d'un changement de vue. Le coût total de communication pour un consensus complet est ainsi ramené à $\mathcal{O}(n^2)$ dans le pire cas, mais reste $\mathcal{O}(n)$ par vue en conditions normales. Les **signatures à seuil** permettent de regrouper les votes en un unique *Quorum Certificate* (QC), réduisant considérablement la charge cryptographique. Dans le protocole **HotStuff**, les **signatures à seuil** jouent un rôle central dans l'optimisation de la communication et de la vérification des votes. Au lieu que chaque validateur envoie et publie sa propre signature individuelle, les contributions cryptographiques de $n - f$ validateurs (où f est le nombre maximal de nœuds byzantins tolérés) sont combinées pour produire une unique signature agrégée QC. En outre, les signatures à seuil renforcent la sécurité en garantissant qu'aucune minorité de nœuds byzantins ne peut forger un QC valide sans atteindre le seuil requis de $n - f$ signataires honnêtes.

— **Réactivité optimiste.** HotStuff se distingue par sa capacité à progresser au rythme du réseau réel : un leader n'attend que les $n - f$ premières réponses pour avancer. Cette propriété est rendue possible par la structure à trois phases et le prédicat `safeNode`, qui permet d'ignorer des verrous obsolètes. En revanche, une version simplifiée à deux phases perdrait cette réactivité et se comporterait comme Tendermint.

— **Version chaînée (*Chained HotStuff*).** Dans cette variante pipelinée, chaque phase `prepare` déclenche directement la vue suivante, où le leader suivant poursuit le processus. Les blocs sont ainsi validés après quatre vues consécutives (v1–v4), améliorant la latence tout en maintenant la sécurité.

Comparaison des performances et de la complexité.

HotStuff offre des performances en latence comparables à celles de BFT-SMaRt [Bessani et al., 2014] tout en surpassant PBFT en débit et en scalabilité. Sa structure modulaire et son coût de communication linéaire en font une solution adaptée aux blockchains à haut volume de transactions. Le tableau suivant synthétise la complexité et la réactivité de plusieurs protocoles BFT après le GST :

Protocole	**Complexité d'authentificateur**			**Réactivité**
	Leader correct	Défaillance leader	f défaillances leader	
DLS [Dwork et al., 1988]	$\mathcal{O}(n^4)$	$\mathcal{O}(n^4)$	$\mathcal{O}(n^4)$	×
PBFT [Castro and Liskov, 1999]	$\mathcal{O}(n^2)$	$\mathcal{O}(n^3)$	$\mathcal{O}(fn^3)$	✓
SBFT [Gueta et al., 2019]	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(fn^2)$	✓
Tendermint Buchman [2016]	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$	$\mathcal{O}(fn^2)$	×
HotStuff Yin et al. [2019]	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(fn)$	**! (Optimiste)**

TABLEAU 2.1 – Comparaison des protocoles BFT en termes de complexité d'authentificateur et de réactivité [Yin et al., 2019].

En résumé, Tendermint et HotStuff incarnent deux évolutions complémentaires du consensus byzantin : le premier privilégie la finalité immédiate et la cohérence stricte, tandis que le second optimise la réactivité et la scalabilité grâce à un design linéaire et pipeliné. Ces innovations préparent le terrain pour les mécanismes de vote hiérarchiques et les consensus hybrides étudiés dans la section suivante.

2.3.5 Optimisations modernes et défis des consensus à vote

L'optimisation algorithmique pure, bien que nécessaire, ne suffit pas à adresser le problème fondamental de scalabilité des protocoles BFT traditionnels. Des structures arborescentes, comme celles employées par HotStuff, réorganisent la communication non plus en maillage complet mais selon une topologie hiérarchique. Au lieu que chaque nœud communique avec tous les autres, les messages transitent par des leaders intermédiaires qui agrègent les votes avant de les propager. Le *pipelining*, quant à lui, permet un traitement en flux continu des propositions de blocs, où les différentes phases de consensus se chevauchent temporellement, réduisant significativement la latence.

Cette optimisation repose sur deux mécanismes complémentaires :

— **Comités probabilistes** : Sélection aléatoire d'un sous-groupe de validateurs pour chaque tour de consensus, comme dans **Algorand** [Gilad et al., 2017] qui utilise un processus de sélection cryptographique vérifiable pour former des comités de taille réduite tout en maintenant une sécurité probabiliste élevée.

— **Sharding de consensus** : Partitionnement du réseau en plusieurs comités traitant des transactions en parallèle, approche mise en œuvre dans **Omniledger** [Luu et al., 2016], où différents *shards* traitent des ensembles disjoints de transactions avec comités BFT dédiés.

Cependant, ces consensus présentent des vulnérabilités structurelles. Leur vivacité est conditionnée par des hypothèses de synchronie partielle, les rendant sensibles aux attaques par déni de service. Une limitation fondamentale réside dans leur vulnérabilité aux attaques Sybil [Yu et al., 2006] : dans un environnement *permissionless*, un attaquant peut créer un grand nombre d'identités factices pour prendre le contrôle du processus de vote.

Pour contrer cette menace, les implémentations modernes combinent les mécanismes de vote avec des systèmes *proof-based* :

— **Tendermint** utilise un *Proof of Stake* pour sélectionner les validateurs.

— **HotStuff** est souvent déployé avec un système de réputation ou de *stake*.

— Les **consortiums** résolvent le problème en exigeant une identité vérifiée.

En synthèse, les approches *vote-based* excellent dans les contextes contrôlés (consortiums, entreprises) où elles offrent une finalité immédiate et un haut débit, mais leur adoption dans les réseaux entièrement ouverts reste conditionnée à la résolution des défis de scalabilité et de résistance aux attaques Sybil.

2.4 Justification du cadre d'étude "vote-based"

Les deux familles de protocoles de consensus présentent des caractéristiques fondamentalement différentes qui déterminent leur applicabilité selon les contextes. Les mécanismes *proof-based* excellent dans les environnements ouverts et permissionless où l'identité des participants n'est pas vérifiée, tandis que les approches *vote-based* sont particulièrement adaptées aux environnements contrôlés où les validateurs sont identifiés.

La différence fondamentale réside dans leur approche de la sécurité : les protocoles *proof-based* reposent sur des barrières économiques et cryptographiques pour décourager les comportements malveillants, alors que les protocoles *vote-based* s'appuient sur des mécanismes formels de coordination et de détection des fautes. Cette distinction a des implications majeures sur leurs performances, leur scalabilité et leur modèle de sécurité. Le tableau comparatif suivant tableau 2.2 résume les principales différences entre les deux approches :

Caractéristique	**Proof-based (PoW)**	**Vote-based (BFT)**
Modèle de sécurité	Probabiliste	Absolu
Finalité	Après plusieurs blocs	Immédiate
Consommation énergétique	Élevée	Faible
Débit de transactions	Limité	Élevé
Latence	Variable	Prédictible

TABLEAU 2.2 – Comparaison des caractéristiques entre protocoles Proof-based et Vote-based

Dans le cadre de cette thèse, nous nous focalisons spécifiquement sur les protocoles *vote-based* pour plusieurs raisons fondamentales :

1. **Finalité immédiate** : Contrairement aux protocoles *proof-based* comme le Proof of Work où la probabilité d'une réorganisation de la chaîne (fork) décroît exponentiellement avec le nombre de blocs ajoutés après une transaction nécessitant typiquement plusieurs confirmations pour atteindre un niveau de sécurité acceptable contre les doubles dépenses, les approches BFT garantissent une finalité absolue dès la validation d'un bloc. Dans Bitcoin, chaque bloc supplémentaire ajouté après une transaction réduit la probabilité qu'une chaîne alternative plus longue ne la remplace, mais cette probabilité n'atteint jamais zéro. Dans les protocoles BFT, une fois qu'un bloc reçoit les votes requis des validateurs (généralement 2/3 de l'ensemble), il devient immuable et ne peut être réorganisé. Cette propriété est essentielle pour les applications financières, les systèmes de vote électronique ou toute application nécessitant une irréversibilité immédiate des transactions sans période d'incertitude.
2. **Efficacité énergétique** : Les protocoles BFT éliminent la course à la puissance de calcul caractéristique du Proof of Work, qui consomme des quantités d'énergie comparables à celles de pays entiers. En remplaçant la compétition minière par des échanges de messages coordonnés, les protocoles BFT permettent une consommation énergétique réduite de plusieurs ordres de grandeur, tout en maintenant un niveau de sécurité équivalent.
3. **Réduction de latence par sélection dynamique** : Les protocoles BFT permettent une sélection dynamique des validateurs optimisée selon des critères de performance réseau, notamment la proximité géographique et la qualité des connexions. Cette capacité à choisir des validateurs formant une clique à faible latence permet de réduire significativement les délais de communication, contrairement aux protocoles *proof-based* où la sélection des mineurs est aléatoire et non optimisée pour les performances.
4. **Performance prédictible** : La complexité communicationnelle quadratique ($O(n^2)$) des protocoles BFT, bien que limitante pour la scalabilité, permet une modélisation précise des temps de réponse. Cette prédictibilité est cruciale pour les applications nécessitant des garanties de latence strictes, contrairement aux protocoles *proof-based* dont les performances peuvent varier significativement.
5. **Richesse de l'analyse comportementale** : La nature explicite des votes dans les protocoles BFT fournit des données structurées pour l'analyse comportementale et la détection d'anomalies, au cœur de nos contributions méthodologiques.

Cette focalisation nous permet d'approfondir les défis spécifiques aux environnements BFT, particulièrement la sélection optimale des validateurs et la détection des comportements byzan-

tins, qui constituent les verrous scientifiques principaux adressés dans les chapitres suivants de cette thèse.

2.5 Problématique

Les protocoles de consensus BFT constituent une pierre angulaire de la sécurité et de la résilience dans les systèmes distribués et, en particulier, dans les blockchains. Cependant, malgré leurs garanties formelles, plusieurs limites subsistent et freinent leur adoption à grande échelle.

Premièrement, la plupart des mécanismes actuels reposent sur des ensembles de validateurs définis de manière statique ou semi-statique. Cette rigidité rend difficile l'adaptation du protocole face à des changements dynamiques du réseau, comme l'évolution progressive des comportements. Cette incapacité à s'ajuster en temps réel fragilise la robustesse des systèmes distribués dans des environnements ouverts et hétérogènes.

Deuxièmement, la détection des nœuds déviants demeure un défi majeur. Dans la pratique, les comportements malveillants ou irréguliers ne se traduisent pas toujours par des fautes franches et immédiatement détectables. Ils peuvent au contraire apparaître sous la forme d'anomalies subtiles, intermittentes ou contextuelles, ce qui complique leur identification rapide et fiable. L'absence de mécanismes adaptés pour surveiller, caractériser et isoler ces comportements accroît la vulnérabilité des réseaux aux attaques sophistiquées et aux manipulations stratégiques.

Ces limitations mettent en évidence un besoin urgent de méthodes capables de mieux s'adapter aux environnements dynamiques, de renforcer la détection des comportements anormaux, et de rendre le processus de sélection des validateurs plus robuste et plus efficace.

La problématique de la sélection de validateurs dans un réseau blockchain repose sur la nécessité de concilier trois objectifs : la performance (faible latence), la sécurité (résilience aux comportements malveillants) et la décentralisation (répartition équitable des rôles). Le schéma présenté à la Figure 2.2 est une proposition de cette thèse et illustre une architecture en trois couches qui reflète cette complexité

La première couche est dédiée à l'acquisition de données, où sont collectés les délais de communication en temps réel, l'historique de participation des nœuds et les protocoles de sélection existants. Ces informations constituent la base indispensable pour toute prise de décision. La deuxième couche s'appuie sur des méthodes d'analyse comportementale combinant réputation, clustering, métriques probabilistes et dynamique d'opinions. Cette étape vise à identifier des tendances collectives et individuelles permettant de mieux caractériser la fiabilité des nœuds. Enfin, la troisième couche correspond à la prise de décision : partition fiable des validateurs, surveillance renforcée ou isolement des nœuds suspects. Cette étape est complétée par un mécanisme de rétroaction, garantissant l'adaptation continue du protocole face aux évolutions du réseau.

Cette architecture traduit ainsi le cœur de la problématique : comment exploiter des données hétérogènes et dynamiques pour prendre des décisions de sélection robustes, tout en assurant un compromis entre efficacité et équité dans un environnement distribué et potentiellement hostile.

2.5.1 Problème 1 : la Sélection des Validateurs

Dans les blockchains basées sur un consensus BFT, un arbitrage fondamental existe entre la performance et la sécurité. Alors que la sécurité du consensus augmente avec le nombre de validateurs, les performances (débit, latence) diminuent de manière quadratique en raison de la complexité des communications ($O(n^2)$). Pour concilier ces exigences, une approche architecturale courante consiste à partitionner le réseau en deux sous-ensembles : un groupe restreint de nœuds,

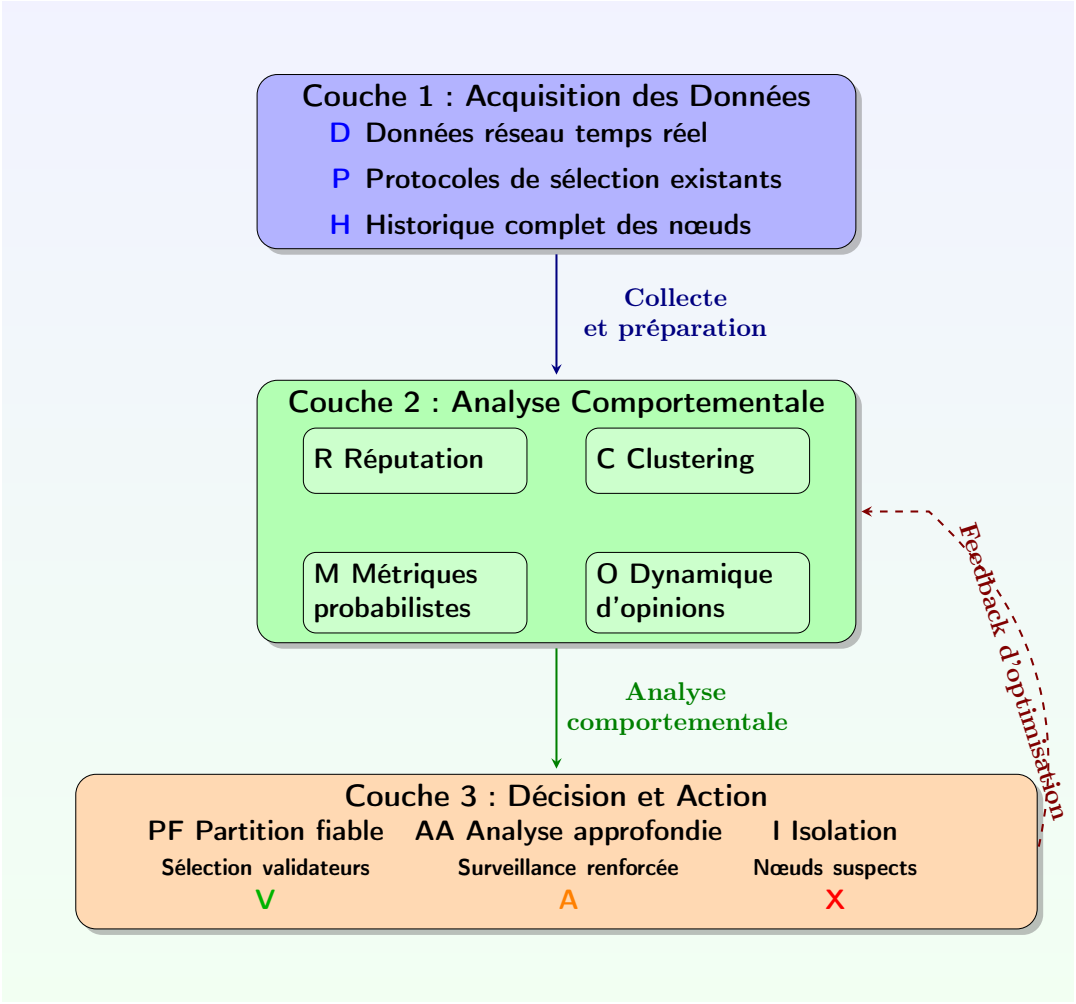


FIGURE 2.2 – Architecture en trois couches du processus de sélection optimisée de validateurs. La couche 1 collecte les données, la couche 2 les analyse grâce à diverses méthodes, et la couche 3 partitionne le réseau pour une sélection ciblée des validateurs.

les "validateurs", exécute le protocole de consensus, tandis que les autres, les "non-validateurs", répliquent la chaîne. Cette stratégie soulève une problématique centrale : comment définir les critères de sélection de ce comité de validateurs ?

Cet état de l'art analyse les différentes stratégies proposées dans la littérature, en les classant selon leur objectif principal : l'optimisation des performances par des approches déterministes, ou la garantie de la sécurité par des approches aléatoires.

Sélection Déterministe pour l'Optimisation des Performances

Une première famille de méthodes vise à améliorer activement le débit et la latence en sélectionnant les validateurs sur la base de critères mesurables [Almakhdhub et al., 2019, Leduc et al., 2022a]. Ces approches sont particulièrement pertinentes pour les blockchains privées où les participants sont identifiés [Androulaki et al., 2018].

Approches basées sur la Proximité Réseau : Il a été montré que le facteur le plus déterminant dans la performance du protocole PBFT est le temps de transmission des messages [Kolasa et al., 2020]. Il est donc pertinent de sélectionner un groupe de validateurs présentant une faible latence inter-nœuds [Leduc et al., 2023]. Mathématiquement, cela revient à identifier une clique de taille donnée qui minimise la somme des poids des arêtes, un problème connu sous le nom de partitionnement en cliques. Ce problème étant NP-complet [Cerioli et al., 2008], sa résolution exacte est trop coûteuse en temps. Des solutions approchées, comme les algorithmes génétiques, ont été explorées, mais leur temps de calcul reste un obstacle à une sélection dynamique et fréquente.

Approches basées sur l'Apprentissage par Renforcement Profond (DRL) : Pour surmonter la complexité des méthodes exactes, des stratégies basées sur le Deep Reinforcement Learning (DRL) ont été proposées. L'étude de [Liu et al., 2019a] est emblématique : un agent DRL apprend à sélectionner les validateurs en temps réel pour maximiser le débit. Le modèle prend en compte des facteurs multiples comme la localisation, la capacité de calcul et l'historique de participation. Pour éviter la centralisation, où l'algorithme convergerait vers un petit groupe de nœuds surperformants, des contraintes sont ajoutées en utilisant l'indice de Gini pour garantir une répartition équitable des sélections. Malgré des gains de performance annoncés, ces modèles présentent des défis majeurs, notamment un besoin considérable en données d'entraînement et une prédictibilité qui les rend potentiellement manipulables par un attaquant.

Sélection Sécurisée par des Mécanismes Aléatoires

Dans les blockchains publiques où le risque d'attaque Sybil est élevé [Yu et al., 2006], la priorité est donnée à l'imprévisibilité et à la robustesse de la sélection. Les approches hybrides, qui combinent un premier consensus basé sur une preuve (PoX) avec un consensus BFT subséquent, sont couramment utilisées pour sélectionner le comité de manière sécurisée.

Sélection par PoW et PoS : Le protocole PoW garantit que la participation est proportionnelle à la puissance de calcul, prévenant ainsi les attaques Sybil. Des protocoles comme ByzCoin [Kogias et al., 2016] et ASHWAChain [Arora et al., 2020] utilisent ce mécanisme : l'ajout d'un bloc PoW donne le droit de rejoindre le comité de validateurs BFT . Pour pallier la consommation énergétique de la PoW, la Preuve d'Enjeu (PoS) sélectionne les validateurs en fonction

de leur mise financière .Tendermint [Buchman et al., 2018], par exemple, sélectionne les nœuds ayant les plus grands dépôts pour former le comité.

Sélection par Fonctions Aléatoires Vérifiables (VRF) : Pour introduire un aléa à la fois imprévisible et publiquement vérifiable, les Fonctions Aléatoires Vérifiables sont de plus en plus utilisées [Goldberg et al., 2018]. Une VRF est une fonction pseudo-aléatoire qui fournit une preuve cryptographique de l'exactitude de sa sortie, garantissant un tirage au sort impartial dans un environnement déterministe.

— **Algorand** [Gilad et al., 2017] combine PoS et VRF. La probabilité d'être sélectionné par la VRF est pondérée par l'enjeu du nœud.

— **Gosig** [Li et al., 2020] utilise également une VRF, mais la sélection y est uniforme : tous les nœuds ont la même probabilité d'être choisis, ce qui favorise une décentralisation plus équitable.

Ces approches garantissent une haute sécurité, mais au détriment de l'optimisation des performances réseau, car la sélection aléatoire ne tient aucun compte de la topologie du réseau.

Synthèse

L'analyse de l'état de l'art révèle un arbitrage fondamental entre deux familles de solutions pour la sélection des validateurs :

1. Les approches déterministes et optimisées, qui privilégient la performance en construisant le comité le plus efficace possible. Elles sont pertinentes dans des environnements contrôlés mais sont sujettes à la centralisation et sont prévisibles.
2. Les approches aléatoires et sécurisées, qui privilégient la robustesse. En utilisant des mécanismes de preuve et des VRF, elles garantissent une sélection impartiale et résistante aux attaques Sybil, ce qui est indispensable pour les blockchains publiques. Cependant, elles sacrifient l'optimisation des performances.

Cette dualité suggère qu'un **compromis optimal** pourrait résider dans des approches adaptatives, cherchant à concilier l'efficacité opérationnelle avec une maximisation robuste de la sécurité. Le choix d'une stratégie dépend donc intrinsèquement du modèle de menace et des exigences opérationnelles spécifiques à chaque cas d'usage, où l'idéal serait de maintenir un équilibre dynamique entre performance garantie et sécurité optimisée.

2.5.2 Problème 2 : dynamiques des votes des validateurs

Le deuxième problème de cette thèse concerne les dynamiques des votes des validateurs dans les systèmes blockchain BFT. Alors que le premier problème porte sur la sélection des validateurs, le problème 2 s'intéresse à la manière dont ces validateurs interagissent lorsqu'ils émettent leurs votes et comment ces interactions locales influencent la convergence vers un consensus global. La littérature existante propose plusieurs approches pour analyser le comportement collectif des agents dans des réseaux distribués. Les modèles de formation d'opinion sociale [Castellano et al., 2009a] étudient comment les agents ajustent leurs opinions en fonction de celles de leurs voisins et permettent d'intégrer différents types d'interactions, de renforcement ou antagonistes [Krawiecki and Gradowski, 2024, Carro et al., 2016]. Ces modèles fournissent un cadre analytique pour simuler la convergence, la polarisation ou les blocages dans un réseau. Certains travaux s'inspirent également de concepts issus de la physique statistique pour modéliser la propagation de préférences ou de votes dans des réseaux [Varma et al., 2017]. Parallèlement, plusieurs méthodes ont

été proposées pour détecter les nœuds malveillants dans les systèmes distribués afin de sécuriser le processus de consensus. Les approches topologiques telles que SybilGuard [Yu et al., 2006] les méthodes basées sur la détection de communautés exploitent la structure du graphe pour limiter l'influence d'identités multiples ou isolées. Les modèles probabilistes [Nischwitz et al., 2021] utilisent des distributions statistiques ou des inférences Bayésiennes pour estimer la probabilité qu'un nœud soit déviant. Les méthodes basées sur l'apprentissage automatique, comme BEML [Musa Baig et al., 2023], les SVM, réseaux de neurones ou Isolation Forest, cherchent à identifier des comportements anormaux à partir de signaux observables tels que la latence, les patterns de vote ou l'activité réseau. Enfin, des approches inspirées des dynamiques d'opinion et de l'influence sociale [Jedrzejewski and Sznajd-Weron, 2019] visent à détecter les nœuds qui impactent de façon disproportionnée le réseau. Ces méthodes offrent chacune des avantages spécifiques : robustesse contre certaines attaques, cadre analytique solide ou capacité à détecter des anomalies complexes. Cependant, elles présentent aussi des limites significatives : dépendance à la topologie ou aux données et coûts computationnels élevés. Étudier les votes des validateurs sous l'angle des dynamiques d'opinion permet ainsi de dépasser la simple analyse structurelle et de développer un cadre capable de comprendre l'émergence du consensus global, d'identifier les facteurs pouvant ralentir ou perturber cette convergence et de proposer des modèles adaptés pour améliorer la robustesse et la stabilité des systèmes blockchain BFT.

La Figure 2.3 schématise la convergence conceptuelle entre les dynamiques d'opinion et les mécanismes de consensus en blockchain : elle montre comment des règles locales d'interaction (renforcement ou antagonisme) combinées à la structure du réseau peuvent conduire soit à l'émergence d'un ordre global (consensus), soit à la polarisation ou au blocage du processus décisionnel selon la topologie et la nature des influences.

En résumé, cette section expose la deuxième problématique de la thèse : analyser les dynamiques de vote des validateurs, en lien avec leur influence locale et l'émergence du consensus global, afin de compléter les méthodes de sélection des validateurs et de garantir la résilience des systèmes distribués.

Modèles de dynamiques d'opinion classiques

Les modèles de dynamique d'opinion visent à expliquer l'émergence de comportements collectifs (consensus, polarisation, fragmentation) à partir de règles d'interaction locales entre agents [Noorazar, 2020, Grabisch and Rusinowska, 2020]. Ces modèles supposent que les agents ajustent leur opinion en fonction de celles des autres, sans chercher à inférer une vérité fondamentale [Grabisch and Rusinowska, 2020]. Les opinions peuvent être discrètes, souvent binaires (e.g., {oui, non}, $\{+1, -1\}$), ou continues [Grabisch and Rusinowska, 2020, Krawiecki and Gradowski, 2024].

Dans les modèles à opinions binaires, les agents sont souvent représentés par des *spins* $S_i \in \{-1, +1\}$. Le comportement collectif est caractérisé par des quantités macroscopiques comme la concentration c (la proportion des agents ayant l'opinion $+1$) ou l'aimantation (magnetization) m (le paramètre d'ordre), liées par la relation $\mathbf{m = 2c - 1}$ ou $\mathbf{c = (1 + m)/2}$. Le désordre maximal (phase paramagnétique, PM) est atteint lorsque $\mathbf{c = 1/2}$ ($\mathbf{m = 0}$), tandis que la dominance d'une opinion (phase ferromagnétique, FM) correspond à $\mathbf{c \neq 1/2}$.

Parmi les modèles à opinions binaires, plusieurs sont considérés comme paradigmatiques [Krawiecki and Gradowski, 2024, Grabisch and Rusinowska, 2020] :

— **Le modèle de l'électeur (Voter Model)** : un agent adopte l'opinion d'un de ses voisins choisi au hasard [Krawiecki and Gradowski, 2024, Castellano et al., 2009a, Jedrzejewski and Sznajd-Weron, 2019, Peralta et al., 2018]. Ce mécanisme, le plus simple de l'imita-

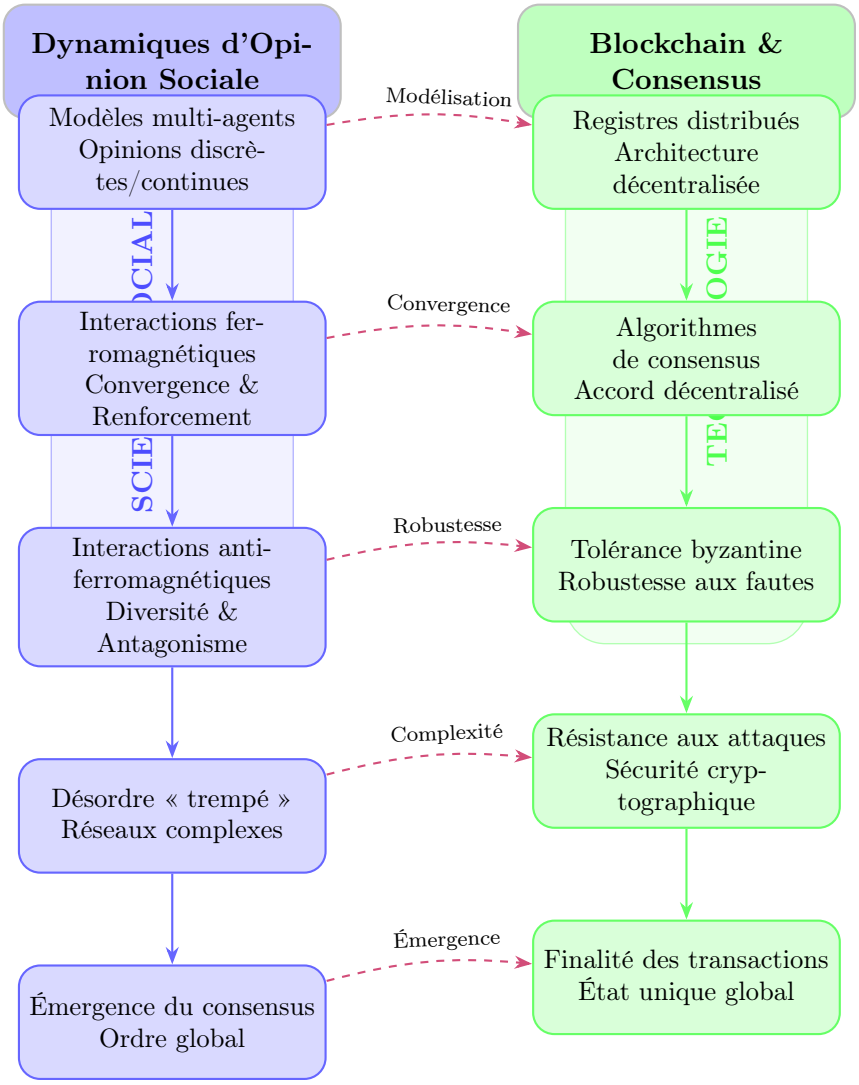


FIGURE 2.3 – Convergence conceptuelle entre les dynamiques d'opinion sociale et les technologies blockchain

tion aléatoire, conduit à des **états absorbants** correspondant au consensus total. Ses variantes « bruyantes » (noisy voter models) introduisent la possibilité de changements d'opinion spontanés (bruit ou non-conformité), ce qui supprime les états absorbants et rétablit l'ergodicité [Krawiecki and Gradowski, 2024, Carro et al., 2016]. Le temps moyen pour atteindre le consensus dans un système fini est proportionnel à la taille du système N.

— **Le modèle q-voter (q-Voter Model)** : Un agent est influencé par un groupe de q voisins (un « q-lobby »). Si ces q voisins sont unanimes, l'agent adopte leur opinion [Krawiecki and Gradowski, 2024, Sznajd-Weron, 2005, Sznajd-Weron et al., 2011, Jedrzejewski and Sznajd-Weron, 2019, Gradowski and Krawiecki, 2020]. Ce modèle non linéaire repose sur le principe psychologique selon lequel l'unanimité d'un groupe d'influence exerce une forte pression sociale [Krawiecki and Gradowski, 2024, Sznajd-Weron, 2005, Jedrzejewski and Sznajd-Weron, 2019].

  Le modèle incorpore un **paramètre d'indépendance** p ($0 \leq p \leq 1$), qui correspond à la probabilité pour un agent d'ignorer ses voisins et de changer d'avis aléatoirement [Krawiecki and Gradowski, 2024, Peralta et al., 2018, Gradowski and Krawiecki, 2020]. L'indépendance agit comme un bruit interne ou une *température sociale*. L'introduction de ce paramètre peut conduire à des **transitions de phase discontinues** (premier ordre) lorsque la taille du lobby q est suffisamment grande (e.g., $q \geq 6$ ou $q \geq 5$), distinguant la phase FM (ordonnée, $m \neq 0$) de la phase PM (désordonnée, $m = 0$). Le **point tricritique (TCP)** sépare la transition discontinue de la transition continue (deuxième ordre) dans le diagramme de phase (p, q).

— **Le modèle du vote à la majorité (Majority-Vote Model)** : Ce modèle suppose que les agents d'un groupe sélectionné aléatoirement adoptent l'opinion majoritaire au sein de ce groupe [Krawiecki and Gradowski, 2024, Castellano et al., 2009a, Jedrzejewski and Sznajd-Weron, 2019]. Dans l'approche de champ moyen, les points fixes de la concentration p_+ (fraction positive) sont $\mathbf{p_+ = 0, 1/2, 1}$. Le point $\mathbf{p_+ = 1/2}$ est instable, garantissant que le système converge vers l'opinion initialement majoritaire.

  L'équation de taux pour l'évolution de la concentration p_+ (pour un groupe de taille $r = 3$) est donnée par :

$$\frac{\mathbf{dN_+}}{\mathbf{dt}} = \mathbf{3}(\mathbf{p_+^2 p_-} - \mathbf{p_+ p_-^2}) = -\mathbf{6 p_+}(\mathbf{p_+} - \mathbf{1/2})(\mathbf{p_+} - \mathbf{1})$$

  où N_+ est le nombre d'agents avec opinion $+1$, et $p_- = 1 - p_+$.

— **Le modèle de seuil (Threshold Model)** : Un agent adopte une opinion si la proportion de ses voisins partageant cette opinion atteint un certain **seuil personnel** μ_i [Castellano et al., 2009a, Jedrzejewski and Sznajd-Weron, 2019]. C'est un modèle clairement **conformiste**. Des variantes **anticonformistes** existent, où un agent prend l'opinion opposée si le seuil est atteint. Les points fixes de la fonction de transition G (qui régit l'évolution de la concentration $c(t)$) correspondent aux états absorbants du processus.

Formalisme Analytique des Dynamiques Binaires : L'analyse de la dynamique stochastique en champ moyen (MFA) commence par l'approximation des variables fluctuantes par leur espérance, valide pour les graphes complètement connectés.

1.MFA et Équation de Taux L'évolution de la concentration $c(\tau)$ au temps τ peut être

modélisée par une équation de taux (*rate equation*) :

$$\mathbf{c}(\tau + \mathbf{1}) = \mathbf{c}(\tau) + \frac{\mathbf{1}}{\mathbf{N}} \sum_{\mathbf{x}'} \mathbf{P}(\mathbf{x}', \tau)[\gamma^+(\mathbf{x}') - \gamma^-(\mathbf{x}')] \tag{2.1}$$

où $\gamma^+(x')$ et $\gamma^-(x')$ sont les probabilités de transition pour augmenter ou diminuer la concentration c (changement d'opinion).

2. Approximations de Paires L'**Approximation de Paire (PA)** est considérée comme plus précise que la MFA, notamment sur les réseaux non complets, car elle incorpore les corrélations dynamiques entre voisins immédiats. Un concept clé introduit par la PA est celui des liens actifs : un lien est actif s'il connecte des agents avec des opinions incompatibles, que l'interaction soit renforçante ou antagoniste. La *SHPA (Signed Homogeneous Pair Approximation)* est une version raffinée qui distingue les concentrations de liens actifs selon qu'ils sont renforçants ou antagonistes, conduisant à des prédictions plus précises pour les réseaux signés.

Modèles à Opinions Continues : Dans ces modèles, l'opinion $\mathbf{x_i(t)}$ d'un agent est une variable réelle, typiquement dans $[-1, 1]$.

— **Le modèle de DeGroot** : Modèle linéaire classique en temps discret [Grabisch and Rusinowska, 2020, Varma et al., 2020]. L'opinion globale évolue par une relation matricielle impliquant la **matrice stochastique par lignes W** (matrice des poids de confiance) :

$$\mathbf{x(t+1)} = \mathbf{W x(t)} \tag{2.2}$$

Si le graphe d'interaction est suffisamment connexe (par exemple, $\mathbf{W}$ est primitive), le système atteint un consensus. Le consensus atteint $\mathbf{x}(\infty)$ est donné par $\mathbf{x(0)}^{\mathbf{T}} \cdot \mathbf{v}$, où $\mathbf{v}$ est l'unique vecteur propre gauche de $\mathbf{W}$ associé à la valeur propre 1. La somme des opinions $\sum x_i(t)$ est conservée lorsque la dynamique est régie par le Laplacien ou lorsque la matrice $\mathbf{W}$ est doublement stochastique.

— **Dynamiques Linéaires Hybrides** : Les modèles de DeGroot en temps continu (Éq. 2.3) sont souvent utilisés dans les systèmes multi-agents ou les modèles hybrides (combinaison de flux continu et de sauts discrets). La dynamique converge vers l'équilibre de manière exponentielle, la vitesse étant régie par la deuxième plus petite valeur propre (connectivité algébrique) λ_2 de la matrice Laplacienne $\mathbf{L}$.

$$\dot{\mathbf{x}}(\mathbf{t}) = -\mathbf{L x(t)} \tag{2.3}$$

— **Le modèle à confiance limitée** *(Bounded Confidence Model, BCM)* : Ces modèles, comme le modèle de **Hegselmann-Krause (HK)**, supposent que les agents n'interagissent et ne moyennent les opinions que de ceux qui sont dans leur borne de confiance ε [Grabisch and Rusinowska, 2020, Morarescu and Girard, 2010]. La règle d'actualisation du modèle HK est donnée par :

$$\sigma_{\mathbf{i}}(\mathbf{t+1}) = \frac{\mathbf{1}}{|\mathbf{I_i(\Sigma(t))}|} \sum_{\mathbf{j} \in \mathbf{I_i(\Sigma(t))}} \sigma_{\mathbf{j}}(\mathbf{t}) \tag{2.4}$$

où

$$\mathbf{I_i(\Sigma(t))} = \{\, \mathbf{j} \mid |\sigma_{\mathbf{i}}(\mathbf{t}) - \sigma_{\mathbf{j}}(\mathbf{t})| < \varepsilon \,\} \tag{2.5}$$

est l'ensemble de confiance de l'agent i à l'instant t.

Contrairement au modèle de DeGroot, le BCM conduit généralement à la fragmentation ou au clustering (formation de communautés) [Frasca et al., 2019, Hegselmann, 2015]. La dynamique converge vers un état stable en un nombre fini de pas . La stabilisation est due aux *splits* (fissures) du profil d'opinion, qui créent des sous-profils ou des communautés qui cessent d'interagir entre elles . L'évolution de l'opinion dans le modèle HK peut être réinterprétée dans le cadre des systèmes hybrides (combinaison de flux continus et de sauts discrets pour l'activation/désactivation des liens).

Convergence entre blockchain et formation d'opinion

La convergence entre les modèles de dynamique d'opinion et la technologie blockchain s'articule principalement autour du concept de **consensus**. Dans les systèmes multi-agents, le consensus est atteint lorsque tous les agents parviennent à un accord sur une valeur commune. Cet objectif est formellement identique à celui des protocoles de consensus byzantins qui, au cœur des blockchains, permettent à un réseau de nœuds distribués de s'accorder de manière irréversible sur un état global, typiquement, la séquence et la validité des transactions. Le processus par lequel les nœuds d'un réseau décentralisé s'accordent sur un état unique et partagé du registre peut être modélisé comme une dynamique d'opinion [Chen and Shu, 2015, Spanos et al., 2005].

Dans ce cadre, les nœuds du réseau sont les « agents » et les blocs ou transactions candidats sont les « opinions » [Spanos et al., 2005]. Les règles de mise à jour des opinions sont définies par le protocole de consensus, qui, comme dans les modèles d'opinion, régit l'émergence d'un comportement global à partir d'interactions locales [Morarescu and Girard, 2010].

— **Influence locale et propagation globale :** Similairement aux modèles d'opinion où les agents sont influencés par leur voisinage, les nœuds d'une blockchain prennent des décisions basées sur les informations locales qu'ils reçoivent de leurs pairs. Ces décisions locales se propagent ensuite à travers le réseau pour aboutir à un consensus global [Chen and Shu, 2015].

— **Robustesse et agents non-conformistes :** Les dynamiques d'opinion étudient comment le consensus peut être atteint malgré la présence de « bruit » ou d'agents non-conformistes (indépendants ou anticonformistes) [Krawiecki and Gradowski, 2024, Castellano et al., 2009a, Jedrzejewski and Sznajd-Weron, 2019]. De manière analogue, les protocoles de consensus des blockchains doivent être robustes face aux pannes byzantines, c'est-à-dire des nœuds au comportement arbitraire ou malveillant [Chen and Shu, 2015]. Les modèles d'opinion fournissent ainsi un cadre pour analyser la résilience de ces protocoles.

— **Vitesse de convergence :** La vitesse à laquelle un consensus est atteint est un enjeu critique. En dynamique d'opinion, cette vitesse est souvent liée à la connectivité du réseau, mesurée par la deuxième plus petite valeur propre de la matrice laplacienne du graphe (connectivité algébrique). Dans les blockchains, cette vitesse détermine la performance du système (le temps de confirmation des transactions) [Spanos et al., 2005].

Les modèles d'opinion offrent donc un appareillage théorique puissant pour analyser et concevoir les mécanismes de formation d'accord dans les systèmes décentralisés, un problème fondamental que la technologie blockchain vise à résoudre de manière pratique et sécurisée.

Applications des blockchains aux systèmes de vote décentralisés

Les systèmes de vote représentent une application naturelle où les dynamiques d'opinion et la technologie blockchain se rencontrent. Les modèles de vote, comme le modèle de l'électeur *Voter*

Model ou le modèle du vote à la majorité *Majority-Vote Model*, sont des piliers de l'étude de la formation d'opinion [Krawiecki and Gradowski, 2024, Castellano et al., 2009a]. L'utilisation de la blockchain pour des systèmes de vote électronique vise à garantir la sécurité, la transparence et la résistance à la censure.

Cependant, le consensus dans un réseau pair-à-pair est vulnérable aux **attaques Sybil**, où un attaquant crée de multiples fausses identités pour influencer le résultat [Chen and Shu, 2015]. Les approches pour contrer ce type d'attaque s'inspirent des mécanismes de pondération de l'influence présents dans les modèles de dynamique d'opinion :

1. **La puissance de calcul (Proof-of-Work)** : Le protocole de Bitcoin repose sur le principe « une-CPU-un-vote ». La décision majoritaire est représentée par la chaîne la plus longue, qui a nécessité le plus grand effort de calcul. Cette approche est cependant vulnérable si un adversaire détient une puissance de calcul dominante [Chen and Shu, 2015].
2. **Les graphes de confiance sociaux** : Pour résister aux attaques Sybil sans dépendre de la puissance de calcul, des systèmes peuvent se baser sur des graphes de confiance. Chaque nœud ne prend en compte que les opinions des nœuds qu'il a explicitement choisi de suivre (*followees*). Les identités Sybil, n'étant suivies par aucun nœud honnête, n'ont alors aucune influence sur le consensus [Chen and Shu, 2015]. Cette approche s'inspire directement des modèles d'opinion où l'influence est locale et dépendante de la topologie du réseau [Krawiecki and Gradowski, 2024, Castellano et al., 2009a]. Le framework Sky, par exemple, utilise explicitement la dynamique d'opinion dans un réseau de confiance pour atteindre le consensus [Chen and Shu, 2015].

2.6 Conclusion et positionnement

Ce chapitre a établi les fondements théoriques et pratiques des protocoles de consensus, avec une focalisation particulière sur les mécanismes BFT. L'analyse a révélé que la performance et la sécurité de ces protocoles reposent essentiellement sur deux problématiques interdépendantes : l'optimisation de la sélection des validateurs et la compréhension des dynamiques de vote.

La classification des approches *proof-based* et *vote-based* a mis en lumière l'arbitrage fondamental entre efficacité opérationnelle et robustesse sécurité. Il convient de préciser qu'au sein de cette thèse, nous ne nous concentrons pas sur l'implémentation pratique de protocoles de type vote-based tels que PBFT, Tendermint ou HotStuff, mais plutôt sur la capacité à garantir que l'hypothèse du vote majoritaire byzantine est respectée. Autrement dit, l'objectif est de s'assurer que le consensus demeure fiable malgré les défaillances et, surtout, face aux attaques coordonnées de coalitions de nœuds malveillants.

Notre étude des dynamiques de vote et des modèles d'opinion collective fournit un cadre théorique pour analyser ces phénomènes. La convergence entre ces modèles et les mécanismes BFT permet de mieux comprendre la formation du consensus, la polarisation des votes et les influences potentielles exercées par des coalitions byzantines. Ces perspectives interdisciplinaire sont essentielles pour détecter des comportements subtils et prévenir des violations de l'hypothèse du vote majoritaire.

Les éléments présentés dans ce chapitre constituent ainsi le socle nécessaire pour aborder les verrous identifiés : l'adaptativité de la sélection des validateurs face aux dynamiques réseau changeantes, et la détection des comportements byzantins coordonnés à travers l'analyse des patterns de vote. Ces fondations ouvrent la voie au chapitre suivant, qui propose une méthodologie d'optimisation du protocole de sélection des validateurs, intégrant à la fois les contraintes de performance réseau et les impératifs de sécurité distribuée.

Chapitre 3

Optimisation d'un protocole de sélection de validateurs dans un environnement BFT

3.1 Introduction

Le chapitre précédent a établi que les protocoles de consensus BFT, bien que garantissant une finalité immédiate et une efficacité énergétique supérieure, se heurtent à un verrou fondamental : l'arbitrage inévitable entre sécurité et performance. Ce compromis se cristallise particulièrement autour de la sélection des validateurs, où l'augmentation du nombre de participants renforce la résilience byzantine, mais dégrade quadratiquement les performances en raison de la complexité communicationnelle. Si les approches existantes, qu'elles soient déterministes ou aléatoires, apportent des solutions partielles, elles peinent à concilier adaptation dynamique, équité de participation et optimisation des performances réseau.

Face à ce constat, ce chapitre propose une contribution méthodologique directe visant à dépasser ces limitations. Nous présentons ici un protocole de sélection de validateurs fondé sur une approche centrée dynamique, dont l'objectif est de maintenir en temps réel l'équilibre optimal entre faible latence et décentralisation effective. Notre approche s'appuie sur une modélisation graphe du réseau et introduit le concept de distance compensée, permettant d'ajuster dynamiquement l'éligibilité des nœuds en fonction de leur historique de participation.

La structure de ce chapitre s'articule autour de quatre axes majeurs. Nous débutons par une analyse approfondie de l'état de l'art (Section 3.2), examinant les stratégies existantes de sélection de validateurs et les méthodes de détection des nœuds déviants. Nous présentons ensuite un cadre formel de modélisation par graphes (Section 3.5) qui permet de formaliser le problème d'optimisation de la sélection comme une recherche de clique pondérée. La Section 3.6 approfondit cette modélisation en explicitant les métriques de performance et de décentralisation qui guident notre approche. Enfin, nous détaillons notre protocole de sélection centrée dynamique (Section 3.7) et en validons l'efficacité avec des simulations extensives évaluant simultanément la latence moyenne et le degré de décentralisation.

3.2 État de l'art

Dans cette section, nous présentons un panorama des approches existantes relatives aux protocoles de consensus tolérants aux fautes byzantines, avec un accent particulier sur les enjeux

et méthodes liés à la sélection des validateurs.

3.2.1 Enjeux de la sélection de validateurs

Dans les protocoles de consensus BFT, les validateurs jouent un rôle central puisqu'ils sont chargés de proposer, vérifier et valider les blocs. La fiabilité du consensus dépend donc directement de leur comportement, qu'il soit honnête ou malveillant(fonctionnel ou défaillant). La sélection des validateurs constitue ainsi un enjeu critique, conditionnant la sécurité, la vivacité et les performances du système distribué.

Un premier défi réside dans la scalabilité. Les protocoles fondateurs comme PBFT souffrent d'une complexité quadratique en communication, car chaque nœud doit échanger avec l'ensemble des autres. La désignation d'un sous-ensemble de validateurs actifs est une réponse naturelle pour réduire ce coût, mais elle introduit la question de savoir combien et quels nœuds choisir pour préserver la sûreté du consensus.

Un second enjeu est lié au dynamisme du réseau. Dans des environnements ouverts tels que les blockchains publiques ou les réseaux IoT, les nœuds peuvent rejoindre ou quitter le système de manière imprévisible. Un mécanisme de sélection efficace doit donc être capable de s'adapter à ces changements sans compromettre la robustesse du consensus.

La résilience face aux comportements malveillants représente également une contrainte majeure. Un adversaire peut tenter d'infiltrer le groupe des validateurs pour perturber le protocole ou influencer la décision collective. La sélection doit donc limiter les risques d'attaques de collusion ou de type Sybil.

Enfin, la question de l'équité et des incitations ne peut être négligée. Un bon mécanisme de sélection doit éviter la concentration du pouvoir entre quelques acteurs dominants, tout en garantissant une incitation suffisante pour que les validateurs participent honnêtement au protocole.

En résumé, la sélection des validateurs se situe au cœur d'un compromis entre sécurité, performance, adaptabilité et équité. Ces enjeux expliquent la diversité des approches proposées dans la littérature, que nous détaillons dans la sous-section suivante.

3.2.2 Méthodes existantes de sélection des validateurs

Les protocoles de consensus tolérants aux fautes byzantines reposent sur des mécanismes de sélection des validateurs garantissant la sûreté et la vivacité du système. Parmi eux, le *Practical Byzantine Fault Tolerance* (PBFT) [Castro and Liskov, 1999] constitue un modèle de référence. Ce protocole assure des propriétés fortes de cohérence dans des environnements à synchronie partielle. Toutefois, il présente deux limitations majeures : (i) une assignation statique des validateurs, peu adaptée aux environnements dynamiques, et (ii) une complexité quadratique des échanges, limitant sa scalabilité dans les réseaux de grande taille.

Pour dépasser ces contraintes, des approches de sélection dynamique ont émergé, cherchant à concilier sécurité, équité et performance :

— **Sélection aléatoire** : *Algorand* introduit une *sortition cryptographique* [Gilad et al., 2017], garantissant un échantillonnage aléatoire et équitable des validateurs à chaque tour de consensus. De même, Gosig [Li et al., 2020] combine un mécanisme de preuve d'enjeu (Proof-of-Stake) et un tirage aléatoire sécurisé, améliorant la robustesse et l'efficacité du protocole dans des environnements massivement distribués.

— **Sélection par réputation** : certaines approches [Qiu et al., 2022] attribuent un score de confiance aux nœuds en fonction de leur historique de comportement, ce qui permet

d'écarter les validateurs déviants et de renforcer la fiabilité du consensus.

— **Sélection optimisée** : d'autres travaux exploitent la proximité topologique entre nœuds [Leduc et al., 2023], afin de réduire la latence et d'optimiser la communication, notamment dans les architectures géodistribuées.

Une synthèse comparative des principales approches de sélection des validateurs est présentée dans le tableau 3.1. Ce tableau met en évidence leurs avantages respectifs, leurs limites structurelles ainsi que les protocoles représentatifs associés, illustrant ainsi la diversité des stratégies employées au sein des environnements BFT.

Type d'approche	**Avantages principaux**	**Limites identifiées**	**Exemples représentatifs**
Sélection statique	Simplicité de mise en œuvre, stabilité	Manque d'adaptabilité, vulnérabilité aux changements réseau	PBFT classique [Castro and Liskov, 1999]
Sélection aléatoire	Équité, résistance aux collusions	Imprévisibilité des performances, coût cryptographique	Algorand [Gilad et al., 2017], Gosig [Li et al., 2020]
Sélection par réputation	Adaptative, historique comportemental	Complexité du calcul, vulnérabilité aux manipulations	Approches basées sur la confiance [Qiu et al., 2022]
Sélection centrée	Performance réseau, faible latence, décentralisation	Dépendance aux métriques	Méthodes basées sur la proximité [Leduc et al., 2023]

TABLEAU 3.1 – Comparatif des méthodes de sélection des validateurs en environnement BFT

Ces mécanismes ont permis d'améliorer la résilience et l'adaptabilité des protocoles BFT. Cependant, ils restent limités par leur incapacité à modéliser les comportements réels et évolutifs des validateurs au cours de l'exécution du consensus. Pour surmonter cette difficulté, des recherches récentes explorent de nouvelles pistes, telles que l'intégration de modèles inspirés des *dynamiques d'opinion* [Jedrzejewski and Sznajd-Weron, 2019] ou l'usage de l'*apprentissage automatique*, comme dans le framework BEML [Musa Baig et al., 2023], pour détecter les anomalies comportementales et adapter la composition des comités de validation.

En résumé, bien que les protocoles récents aient considérablement amélioré la sélection dynamique des validateurs et la réduction de la complexité des communications, plusieurs défis persistent : (i) la prise en compte explicite des comportements intermédiaires des validateurs, (ii) l'intégration originale de modèles d'opinion pour mieux comprendre les dynamiques collectives, et (iii) la conception de mécanismes véritablement adaptatifs et résilients face à la variabilité des conditions réseau et des acteurs.

3.2.3 Détection de nœuds déviants

La détection des comportements déviants constitue une dimension essentielle à la sélection des validateurs dans les réseaux blockchain. Même lorsqu'un sous-ensemble de nœuds est choisi pour participer au consensus, la robustesse du protocole dépend de la capacité à identifier et isoler rapidement ceux qui adoptent un comportement malveillant ou anormal.

Plusieurs approches ont été explorées pour prévenir ces comportements. Au niveau des attaques d'identité et d'infiltration, des mécanismes fondés sur la structure du graphe et la topologie du réseau visent à limiter les attaques de type Sybil. SybilGuard [Yu et al., 2006] constitue une référence notable, proposant des techniques basées sur la connectivité du réseau pour limiter l'impact d'identités multiples malveillantes. Cette méthode reste cependant dépendante des propriétés structurelles du réseau et peut perdre de son efficacité dans des topologies fortement dynamiques ou manipulées. Si les attaquants parviennent à établir de nombreuses connexions vers des nœuds légitimes ou si la topologie évolue rapidement, la distinction entre régions fiables et Sybil devient floue, réduisant la robustesse du mécanisme.

La détection d'anomalies (*Anomaly Detection, AD*) joue un rôle clé pour identifier ou prédire l'occurrence de vulnérabilités dans un délai permettant une réaction appropriée. Une anomalie est définie comme un comportement déviant du modèle attendu, parfois appelée *outlier*, *novelty* ou *activité malveillante* [Hassan et al., 2022]. Dans les réseaux blockchain, l'absence d'autorité centrale et la nature décentralisée des transactions imposent des défis spécifiques pour la détection.

L'apprentissage automatique a récemment été exploré comme levier majeur pour détecter ces anomalies. Des frameworks combinant plusieurs modèles permettent de classifier ou d'identifier des comportements anormaux à partir de signaux observables tels que les latences, patterns de vote ou activités réseau. Le framework BEML (*Blockchain Ensemble Stacked Machine Learning*) illustre cette approche en proposant une architecture empilée pour la détection d'anomalies sur des traces blockchain [Musa Baig et al., 2023]. BAD (*Blockchain Anomaly Detection*) [Signorini et al., 2020] complète cette approche en exploitant les méta-données des blocs, notamment les forks et blocs orphelins, offrant ainsi une perspective plus complète pour la détection d'anomalies. Ces méthodes présentent néanmoins des limites pratiques, notamment le besoin en données labellisées de qualité et le coût computationnel d'entraînement.

Plusieurs défis persistent dans la mise en œuvre de ces approches. En l'absence d'autorité centrale, parvenir à un consensus sur la nature d'un comportement anormal reste difficile, surtout face à des nœuds malveillants. La nature immuable des décisions sur la blockchain impose une détection à très haute précision afin d'éviter les faux positifs. La détermination des attributs pertinents pour distinguer les comportements malveillants constitue un défi méthodologique, tout comme la programmation des contrats, qui limite l'exécution en temps réel des modèles de détection. Enfin, la collecte et l'analyse des transactions nécessitent des mécanismes de confidentialité, tels que la *differential privacy* ou les preuves à divulgation nulle de connaissance.

D'un point de vue probabiliste, des modèles ont été proposés pour caractériser et prédire les défaillances de nœuds dans des environnements BFT. Par exemple, des modèles de type Bernoulli ont été employés pour estimer la probabilité de défaillance et analyser l'impact statistique de pannes sur la sécurité et la disponibilité des protocoles PBFT [Nischwitz et al., 2021]. Ces approches offrent un cadre analytique utile pour raisonner sur des scénarios de panne, mais reposent souvent sur des hypothèses d'indépendance ou de stationnarité des événements, qui peuvent être violées en présence d'adversaires adaptatifs.

En résumé, la littérature actuelle regroupe trois grandes familles d'approches : défenses topologiques contre les attaques d'identité, modélisation probabiliste des pannes et comportements déviants, et détection par apprentissage automatique, enrichie par des solutions dédiées comme BEML et BAD [Musa Baig et al., 2023]. Chacune présente des avantages et des limites spécifiques. Les approches topologiques sont efficaces contre certaines attaques d'identité mais dépendent fortement de la stabilité structurelle du réseau. Les modèles probabilistes offrent des outils analytiques mais reposent sur des hypothèses restrictives. Enfin, les solutions d'apprentissage automatique apportent une adaptabilité théorique mais nécessitent des données et présentent

des défis de robustesse. Une synthèse de ces approches est présentée dans le tableau 3.2, mettant en évidence leurs forces, faiblesses et domaines d'applicabilité.

Méthode de détection	**Forces**	**Faiblesses**	**Applicabilité**
Approches topologiques	Résistance aux attaques Sybil, indépendance des données	Dépendance à la stabilité du réseau, rigidité	SybilGuard [Yu et al., 2006]
Modèles probabilistes	Cadre analytique solide, prédictibilité	Hypothèses restrictives, sensibilité aux distributions	Modèles Bernoulli [Nischwitz et al., 2021]
Apprentissage automatique	Adaptabilité, détection d'anomalies complexes	Besoin important en données, coût computationnel	BEML [Musa Baig et al., 2023]
Dynamiques d'opinion	Modélisation des interactions sociales, perspective théorique	Complexité d'implémentation, validation empirique	Modèles q-voter [Jedrzejewski and Sznajd-Weron, 2019]

TABLEAU 3.2 – Analyse des techniques de détection des comportements byzantins

En conséquence, la détection de nœuds déviants dans les environnements BFT reste un problème ouvert, notamment lorsque l'on exige simultanément des garanties de robustesse, une faible latence décisionnelle et une applicabilité dans des réseaux dynamiques. Cette section met ainsi en lumière la nécessité d'approches complémentaires capables d'exploiter les signaux disponibles tout en restant opérationnelles et résilientes dans des contextes adversariaux. La problématique que nous formulons ensuite s'appuiera sur ces constats pour identifier les lacunes précises que la thèse vise à combler.

3.3 Métriques pour la sélection des validateurs et la détection des nœuds malveillants

La sécurité et les performances des systèmes de registre distribué (DLT), et en particulier des blockchains basées sur la Tolérance aux Fautes Byzantines, dépendent étroitement de la qualité et de la fiabilité des nœuds participant au processus de consensus [Abd-El-Malek et al., 2005]. L'évaluation de la performance des validateurs et la détection des entités malveillantes reposent sur un ensemble de métriques sophistiquées.

3.3.1 Métriques de sélection des validateurs

La sélection des validateurs est souvent motivée par la recherche d'un compromis optimal entre la sécurité (maximisée par un grand nombre de validateurs) et le débit (qui diminue avec l'augmentation du nombre de validateurs) [Abd-El-Malek et al., 2005]. Les métriques employées visent à identifier le sous-ensemble de nœuds (une sous-clique) qui minimise les délais de réponse tout en assurant une décentralisation suffisante [Abraham et al., 2018a, 2016].

Critères d'efficacité et de performance

— **Délai de Réponse (ou Latence) :** Le délai de réponse est la métrique privilégiée pour mesurer l'efficacité d'un sous-graphe de validateurs dans un graphe pondéré par le délai. L'objectif est de trouver une sous-clique de taille donnée minimisant la somme des poids des arêtes (le délai de réponse) [Abraham et al., 2018a, 2016]. Le délai correspond au délai entre l'émission d'une transaction et son engagement sur un nouveau bloc [Abraham et al., 2019].

— **Capacité (Débit Maximal) :** La capacité est définie comme le débit maximal réalisable lorsque la blockchain est dans un état stable et non saturé, mesuré en transactions réussies par seconde (Tx/s) [Burchert and Wattenhofer, 2018, Abraham et al., 2018b]. Pour les systèmes comme Sabine, le but est d'adapter la taille du comité de validateurs pour que le débit de sortie corresponde au débit de la transaction demandée (d_{req}) [Leduc et al., 2022b].

— **Distance Compensée ($d'_{i,j}$)** : Utilisée dans les protocoles de sélection centrique, cette métrique modifie la distance réelle entre un nœud central (j) et un autre nœud (i) en fonction de l'historique de participation ($\kappa_{i,t}$) de ce dernier [Leduc et al., 2023]. Les nœuds ayant peu participé (faible $\kappa_{i,t}$) sont virtuellement rapprochés du centre, augmentant leur probabilité de sélection et favorisant ainsi le roulement. La formule utilisée est (3.15) où γ est un terme positif ajustant la force de la recherche de nouveaux validateurs, et influençant le diamètre du comité de validateurs.

Métriques de réputation et de confiance

Les modèles de réputation visent à récompenser les nœuds fiables et rapides, tout en imposant un roulement pour maintenir la décentralisation.

— **Réputation Locale ($R_{l,i,j}$)** : Mesure l'accord entre deux nœuds, i et j, sur l'ajout d'un bloc, en privilégiant les nœuds dont l'approbation (α_{ij}) a été décisive pour atteindre le seuil de 2/3 nécessaire à la phase de `commit` [Zeng et al., 2022, Bellini et al., 2020]. Elle est définie comme :

$$R_{l,i,j} = \frac{\alpha_{ij} + 1}{\alpha_{ij} + \beta_{ij} + 2} \tag{3.1}$$

où β_{ij} est le nombre de fois où i n'a pas eu besoin de j pour passer à l'étape suivante.

— **Réputation Globale ($R_{g,i}$)** : Elle est calculée comme la moyenne pondérée des réputations locales données par les autres validateurs. Le poids w_k d'un nœud k est basé sur sa réputation globale relative de l'itération précédente :

$$w_k = \frac{R_{g,k}}{\sum_{j=1}^{n} R_{g,j}} \tag{3.2}$$

— **Facteur de Décroissance** : Pour garantir la décentralisation et le roulement, la réputation globale finale peut être multipliée par un terme exponentiel $e^{-\lambda_i t}$, où t est le nombre de périodes depuis la dernière sélection du nœud i en tant que validateur. Le facteur λ_i (proportionnel à $\mu \times \mathrm{Nb}_{\mathrm{Node}}$) contrôle la vitesse de ce roulement.

Métriques de décentralisation et de quorum

— **Indice de Gini (G)** : Cet indice mesure l'inégalité de la participation des nœuds dans la sélection [Leduc et al., 2023]. Une valeur de G proche de 0 indique une égalité parfaite de participation, favorisant la décentralisation .

— **Écart-type Relatif (σ_r)** : Mesure la dispersion du taux de participation par rapport à un idéal. L'objectif est de réduire σ_r pour assurer une distribution équitable des rôles [Leduc et al., 2023].

— **Taille et Poids du Quorum** : Dans les protocoles BFT et leurs variantes (comme Big-FooT), la sécurité repose sur le respect d'une majorité. Le quorum (Q) requis pour le consensus est souvent défini par $Q(n) = \lceil 2n/3 \rceil$, où n est le nombre total de validateurs [Saltini, 2022]. La sécurité implique qu'au moins 2/3 des validateurs disposent de la chaîne à jour.

— **Comités Probabilistes (Algorand)** : Algorand utilise une sélection probabiliste de comités de consensus. Chaque comité (PV_p pour la proposition, SV_p pour le vote "soft", CV_p pour le vote "cert") a une taille attendue (Expected size) et une taille de quorum (Quorum size) prédéfinies pour assurer la sécurité et la vivacité [Gilad et al., 2017].

3.3.2 Métriques et techniques de détection des nœuds malveillants

La détection des nœuds malveillants est cruciale pour la résilience du système face aux attaques Sybil et byzantines.

Détection basée sur la confiance et la réputation

Les systèmes de gestion de la confiance, notamment dans les réseaux de capteurs (WSN) [Zhang, 2025] ou IoMT [Ali et al., 2023], utilisent des scores pour identifier les nœuds peu fiables.

— **Score de Réputation Directe ($D_t(i,j)$)** : Une métrique de base pour mesurer la fiabilité d'un nœud j vu par un nœud i [Zhang, 2025].

— **Score de Réputation Indirecte ($I_t(i,j)$)** [Zhang, 2025] : Calculé en intégrant les recommandations de nœuds adjacents communs (k) pour éviter les désinformations conjointes . Ce calcul pondère les scores de réputation recommandés ($R_{t,k}(i,j)$) par un poids ϕ_k, qui est fonction du score de réputation directe entre le nœud i et le nœud k :

$$\mathrm{I}_t(i,j) = \frac{\sum_{k=1}^{s} \phi_k \cdot R_{t,k}(i,j)}{\sum_{k=1}^{s} \phi_k} \quad \text{avec } R_{t,k}(i,j) = D_t(i,k) \times D_t(k,j) \quad \text{et } \phi_k = \frac{D_t(i,k)}{\sum_{k=1}^{s} D_t(i,k)}$$

— **Valeur de Confiance et Seuils** [Ali et al., 2023] : Les systèmes de gestion de la confiance (par exemple, dans IoMT via logique floue) calculent une valeur de confiance finale. Un nœud est identifié comme Sybil ou non fiable si sa valeur de confiance est inférieure à un seuil prédéfini (par exemple, $TH = 0.5$).

Métriques de résilience et de sécurité

Dans les systèmes qui exploitent la structure du réseau social pour la défense Sybil (comme SybilGuard [Yu et al., 2006]), la sécurité est mesurée par des métriques liées à la topologie du réseau et aux chemins aléatoires.

— **Densité des Nœuds Malveillants (f)** : Dans les modèles de consensus, la résilience maximale est atteinte si la densité de nœuds malveillants est inférieure à 1/3 ($f < n/3$).

— **Arêtes d'Attaque (g)** : Mesure la résilience d'un mécanisme de défense Sybil (comme SybilGuard). g représente les liens entre la région honnête et la région Sybil, et est un facteur déterminant pour la probabilité qu'un nœud honnête en accepte un autre [Lesniewski-Laas and Kaashoek, 2010, Yu et al., 2006].

— **Groupes d'Équivalence Sybil (g)** : Mesurer et borner le nombre de groupes Sybil distincts qu'un adversaire peut contrôler est un objectif crucial pour les systèmes de réplication de fichiers. Si ce nombre est borné à g, placer le fichier sur $g+1$ groupes garantit au moins une bonne copie [Yu et al., 2006].

— **Longueur de Route Aléatoire (w)** : Dans Whānau [Lesniewski-Laas and Kaashoek, 2010], le paramètre w (mixing time) est estimé et utilisé pour la procédure de configuration afin de construire des tables de routage résistantes aux attaques Sybil.

— **Contrainte de Stabilité des Validateurs (BigFooT)** [Saltini, 2022] : Pour empêcher un leader malveillant de proposer une séquence infinie de validateurs byzantins, BigFooT impose une contrainte de sécurité : si l'ensemble de validateurs change, il ne peut pas changer à nouveau avant $|B| \leq f(|V(C)|)$ blocs supplémentaires.

Métriques basées sur l'apprentissage automatique (ML)

L'utilisation de l'apprentissage automatique pour la sélection et la validation introduit des métriques spécifiques basées sur les données comportementales.

— **Facteurs de Réputation (Tableau de Nœuds)** : Pour les modèles basés sur l'apprentissage (RbMNS/ALICIA) [Maskey et al., 2021], les entrées utilisées pour former le modèle de réputation incluent :

1. Le nombre total de blocs validés (t_endorsed).
2. Le nombre total de blocs ajoutés (t_added).
3. Le nombre total de drapeaux levés (t_flagged/flagged count).
4. Le taux net de drapeaux levés (flagged_rate).
5. L'âge du nœud (age).

— **Échec de Validation** : Si la contre-validation d'un événement par des RSU échoue, le compteur de drapeaux ('flag count') du RSU fautif est augmenté, conduisant à une baisse de sa réputation et, potentiellement, à son retrait du consensus s'il passe sous un seuil.

3.4 Compromis sécurité–performance dans la sélection BFT

La conception des mécanismes de sélection des validateurs dans les protocoles à tolérance aux fautes byzantines repose sur un compromis structurel entre sécurité et performance. Ce dilemme fondamental influence l'ensemble des décisions architecturales et opérationnelles des systèmes distribués, depuis la composition des comités de validation jusqu'à la fréquence de rotation des nœuds actifs.

3.4.1 Compromis sécurité–performance

Les protocoles BFT reposent sur l'hypothèse qu'au plus un tiers des validateurs peuvent être compromis. En théorie, l'augmentation du nombre de validateurs renforce la résilience du système en diluant l'influence des acteurs malveillants. Cependant, cette redondance accrue s'accompagne d'une dégradation notable des performances, liée à la complexité quadratique des échanges de messages ($O(n^2)$). La tension entre robustesse et efficacité devient ainsi un enjeu central de la conception des protocoles BFT modernes.

3.4.2 Manifestations opérationnelles du compromis

Taille du comité de validation : La taille du comité de validation illustre parfaitement cette dualité. D'un côté, des comités étendus renforcent la tolérance aux pannes et limitent les risques de collusion. De l'autre, des comités restreints permettent de réduire la latence et d'accroître le débit transactionnel. Le défi consiste donc à identifier un point d'équilibre entre robustesse byzantine et efficacité opérationnelle, équilibre qui pourra être ajusté dynamiquement selon les conditions du réseau.

Fréquence de rotation des validateurs. La rotation des validateurs constitue un second paramètre clé. Une rotation fréquente améliore la sécurité en réduisant la fenêtre d'attaque et l'exposition aux validateurs compromis. À l'inverse, une rotation trop rapide accroît la surcharge protocolaire, perturbe les canaux de communication établis et dégrade les performances. Le compromis réside alors dans une périodicité équilibrée, maintenant à la fois la fraîcheur du comité et la stabilité du réseau.

Critères de sélection et conséquences architecturales. Les critères utilisés pour sélectionner les validateurs traduisent également des objectifs contradictoires. Les approches orientées sécurité privilégient la diversité géographique, l'indépendance organisationnelle et l'hétérogénéité technique, tandis que les approches centrées sur la performance cherchent à optimiser la proximité réseau, la bande passante et la latence. Cette opposition se reflète dans plusieurs contraintes architecturales :

- **Débit maximal** : limité par la taille minimale nécessaire pour garantir la sécurité byzantine ;
- **Latence minimale** : contrainte par la dispersion géographique exigée pour la résilience ;
- **Scalabilité** : restreinte par la nécessité de maintenir un consensus fort sur un ensemble de validateurs croissant ;
- **Métriques hybrides** : imposant la définition de critères pondérés combinant sécurité et performance selon le contexte de risque.

En somme, la reconnaissance de ces limitations structurelles est essentielle pour formuler une problématique réaliste et concevoir des stratégies de sélection adaptatives. Celles-ci doivent viser un équilibre optimal entre résilience byzantine, performance opérationnelle et scalabilité, tout en restant compatibles avec les contraintes dynamiques des environnements distribués.

3.4.3 Sabine[Leduc et al., 2022b] : Un exemple d'optimisation adaptative du compromis Sécurité/Débit

Le consensus PBFT est un exemple largement adopté, est prisé dans les blockchains privées et de consortium pour son efficacité énergétique et sa capacité à atteindre un débit de transactions élevé, en particulier par rapport aux protocoles basés sur la preuve (*Proof-based protocols*). Néanmoins, l'adoption de BFT confronte directement la sécurité et la performance (débit).

La Figure 3.1 illustre l'architecture fonctionnelle du protocole Sabine. Le modèle d'apprentissage construit une relation empirique entre la capacité de la chaîne, la latence et le nombre de validateurs. Ce modèle est ensuite exploité par le contrôleur pour ajuster dynamiquement la taille du comité de validateurs (n) en fonction du débit requis (d_{req}) et du débit observé (d_{commit}). Le schéma met également en évidence la boucle de rétroaction continue entre le contrôleur et la chaîne BFT, garantissant une adaptation autonome aux conditions réseau.

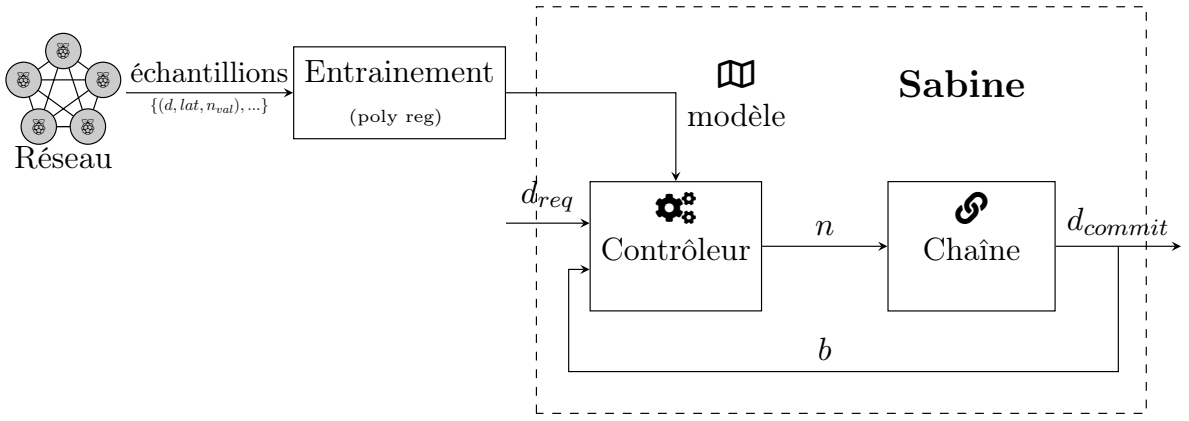


FIGURE 3.1 – Les phases de Sabine [Leduc et al., 2022b]

La nature intrinsèque du compromis

Dans les protocoles BFT, la sécurité dépend directement du nombre de nœuds validateurs (n) impliqués : plus ce nombre est grand, plus le niveau de sécurité est élevé, car le protocole BFT est conçu pour tolérer jusqu'à un tiers d'adversaires parmi les validateurs. Cependant, l'augmentation du nombre de validateurs se traduit par une augmentation de la latence et, par conséquent, une diminution du débit (*throughput*). Le coût de communication pour le processus de *gossip* (l'accord) est initialement quadratique, ce qui contraint généralement les réseaux BFT à un maximum d'environ une centaine de validateurs.

Traditionnellement, le choix du nombre de validateurs est effectué manuellement lors de la configuration initiale de la plateforme, définissant un nombre constant (n) pour toute sa durée de vie opérationnelle [Androulaki et al., 2018]. Cette stratégie, typique des *Consortium Blockchains* [Belotti et al., 2019], représente une réponse fixe à un compromis dynamique. Étant donné que les exigences de performance peuvent évoluer, une valeur fixe de n empêche l'adaptation du débit maximal et, par conséquent, de la sécurité, aux besoins réels en temps réel.

Le Protocole Sabine et l'auto-adaptation

Pour surmonter la limitation d'un nombre fixe de validateurs, le protocole **Sabine** (**S**elf-**A**daptive **B**lockcha**I**n co**N**s**E**nsus) [Leduc et al., 2022b] propose un nouvel algorithme basé sur BFT qui auto-adapte continuellement la taille du comité de validateurs. La Figure 3.1 illustre les principales phases de ce mécanisme adaptatif.

L'objectif principal de Sabine est d'optimiser ce compromis en temps réel en s'assurant que le débit de transactions validé en sortie (d_{commit}) réponde, dans la mesure du possible, au débit de transactions requis/entrant (d_{req}) par l'application. Ce problème est formalisé comme une contrainte d'optimisation (3.3) visant à maximiser la sécurité (représentée par n_t) tout en satisfaisant la demande de débit, sous réserve d'un nombre minimal de validateurs (N_{limit}) :

$$\underset{N}{\operatorname{argmax}} \left(n_t \,\middle|\, d_{\text{commit}}(n_t, \text{lag}_t) = d^t_{\text{req}}\right) \quad \text{s.t.} \quad n > N_{\text{limit}} \tag{3.3}$$

Puisqu'il est difficile d'identifier un modèle mathématique universel de la capacité d'une chaîne BFT en fonction des délais réseau (*lag*), Sabine utilise le *Machine Learning* (ML) pour modéliser les performances spécifiques du réseau.

Le mécanisme d'auto-adaptation de Sabine s'articule autour des étapes suivantes :

1. **Phase d'Entraînement :** Des échantillons de débit maximal (Capacité) sont collectés en faisant varier le nombre de validateurs (n) et un délai simulé (lag). Ces échantillons (triplets (capacité, lag, n)) sont utilisés pour construire un modèle M (via régression polynomiale) qui associe la capacité de la chaîne à n et lag. Ce modèle garantit que le débit (M) est décroissant par rapport à l'augmentation de n et de lag .
2. **Contrôle Opérationnel :** En phase opérationnelle, le contrôleur utilise le modèle M ainsi que ses fonctions inverses (μ et φ). Il commence par estimer le délai actuel du réseau (lag_{test}) en utilisant le débit de *commitment* de blocs (b_t) et le nombre de validateurs précédent (n_{t-1}).
$$lag_{test} = \mu(b_t \cdot \text{BlockSize}, n_{t-1}) \tag{3.4}$$
Ensuite, le nombre idéal de validateurs (n_t) est déterminé en comparant le débit requis (d^t_{req}) à la bijection restreinte du modèle M pour ce délai estimé :
$$n_t = \varphi(d^t_{req}, lag_{test}) \tag{3.5}$$
3. **Action sur la Chaîne :** Sabine est exécuté dans chaque nœud. Périodiquement, le Proposer applique l'algorithme pour déterminer n_t et insère une transaction spéciale dans la chaîne BFT pour informer tous les nœuds de la mise à jour du nombre de validateurs, assurant ainsi la cohérence du système.

Résultats expérimentaux de l'optimisation

Dans une expérience où la demande de débit variait fortement, Sabine a réduit l'erreur relative entre le débit requis et le débit validé à seulement $7,79\%$ (erreur absolue de $1,71$ transactions). En comparaison, une chaîne fixe configurée pour la sécurité maximale (50 validateurs) a produit une erreur relative élevée de $39,5\%$ face à la même demande. Même une chaîne fixe optimisée pour le débit moyen (22 nœuds) affichait une erreur relative de $24,1\%$.

En adaptant dynamiquement le comité de validateurs (maintenu en moyenne à $21,3$ validateurs dans l'exemple de demande testé), Sabine permet d'augmenter la sécurité (en augmentant n) pendant les périodes de faible demande, tout en maximisant le débit lors des périodes de forte demande. Le contrôle de Sabine a également démontré sa résilience en cas de variation de délais inconnus, compensant le ralentissement en réduisant le nombre de validateurs pour maintenir le débit proche du niveau requis.
Sabine se concentre sur la question de **combien** de validateurs choisir, le rendant compatible avec des mécanismes hybrides existants qui se concentrent sur **quels** validateurs choisir.

3.5 Modélisation par graphes d'une blockchain BFT

Cette section introduit le cadre formel utilisé pour analyser et optimiser le protocole de sélection. La théorie des graphes offre une abstraction puissante pour modéliser la topologie du réseau et les performances du consensus.

3.5.1 Concepts fondamentaux

Nous modélisons le réseau pair-à-pair comme un graphe non orienté, complet et pondéré $G = (V, E, w)$, où :

— $V = \{v_1, v_2, ..., v_{\mathcal{N}}\}$ est l'ensemble des $\mathcal{N}$ nœuds du réseau.

— $E \subseteq V \times V$ est l'ensemble des arêtes, représentant les canaux de communication directs. Le graphe étant complet, toute paire de nœuds distincts est connectée ($|E| = \frac{\mathcal{N}(\mathcal{N}-1)}{2}$).

— $w : E \to \mathbb{R}^+$ est une fonction de pondération qui associe à chaque arête (i, j) le délai de communication entre les nœuds i et j.

Un **comité de validateurs** est un sous-ensemble $C \subset V$ de taille $|C| = n$, avec $n \ll \mathcal{N}$. Dans un protocole BFT classique, un tour de consensus nécessite que chaque validateur communique avec tous les autres. La latence de finalité est donc directement influencée par les pires délais de communication au sein de C.

3.5.2 Clique et performance réseau

Dans ce modèle, la structure idéale pour un comité C est une clique. Formellement, une clique $K \subseteq V$ est un sous-ensemble de nœuds où toute paire de nœuds distincts est connectée par une arête dans E, c'est-à-dire :

$$\forall u, v \in K, u \neq v \Rightarrow (u, v) \in E$$

Le sous-graphe induit $G[C] = (C, E_C, w_C)$, où $E_C = \{(u, v) \in E \mid u, v \in C\}$ et w_C est la restriction de w à E_C, est alors un graphe complet.

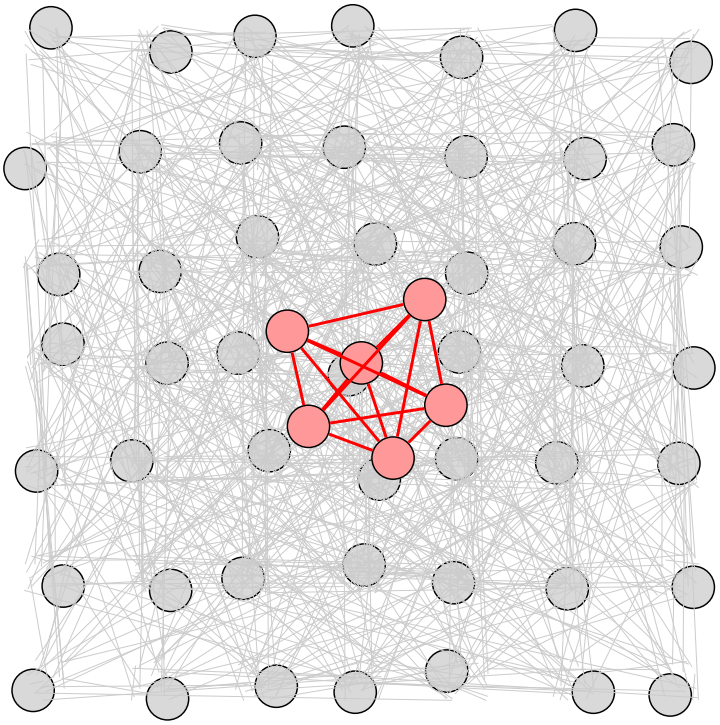


FIGURE 3.2 – Modélisation du réseau blockchain par un graphe complet G. Le comité de validateurs C forme une clique où tous les nœuds sont interconnectés directement, garantissant une communication optimale.

Si C forme une clique dans le graphe des latences G, la communication est optimale pour plusieurs raisons fondamentales :

1. **Latence minimale** : Les messages sont échangés directement entre chaque paire de validateurs, sans délai de routage multi-sauts. Dans un protocole BFT typique nécessitant des échanges. Dans une clique, cette valeur est minimisée pour le comité sélectionné.

2. **Robustesse** : La défaillance d'un nœud n'affecte pas les chemins de communication entre les autres validateurs. Contrairement à un réseau maillé où la défaillance d'un nœud central peut disrupteur multiples chemins, dans une clique, chaque connexion est indépendante.

La recherche d'un comité performant revient donc à identifier un sous-graphe optimal de taille n dans G.

3.5.3 Optimisation

Cependant, dans un graphe complet pondéré par des latences réalistes (reflétant la géographie et l'hétérogénéité des infrastructures), une clique parfaite de taille n peut ne pas exister, ou ses poids peuvent être élevés. L'objectif devient alors une optimisation : trouver un sous-ensemble C de taille n qui se rapproche le plus possible d'une clique.

On peut formuler cela comme la minimisation de la latence moyenne ou maximale au sein du comité. La fonction $g(z)$ de l'équation (3.9), qui représente la somme des poids des arêtes dans le comité, est une mesure naturelle de cette "qualité de clique". Minimiser $g(z)$ revient à sélectionner les n nœuds dont les interconnexions sont en moyenne les plus rapides.

Cette modélisation met en lumière le cœur du problème : la sélection du comité est un problème d'optimisation combinatoire sur un graphe. La solution optimale, comme le confirme la section suivante, est complexe à trouver, ce qui justifie le développement du protocole de sélection centrée dynamique présenté en section 3.7.

3.6 Problème de clique

Le problème peut être modélisé comme la recherche d'un sous-graphe de taille donnée dans un graphe fini pondéré, où les poids représentent les délais de communication entre les nœuds.

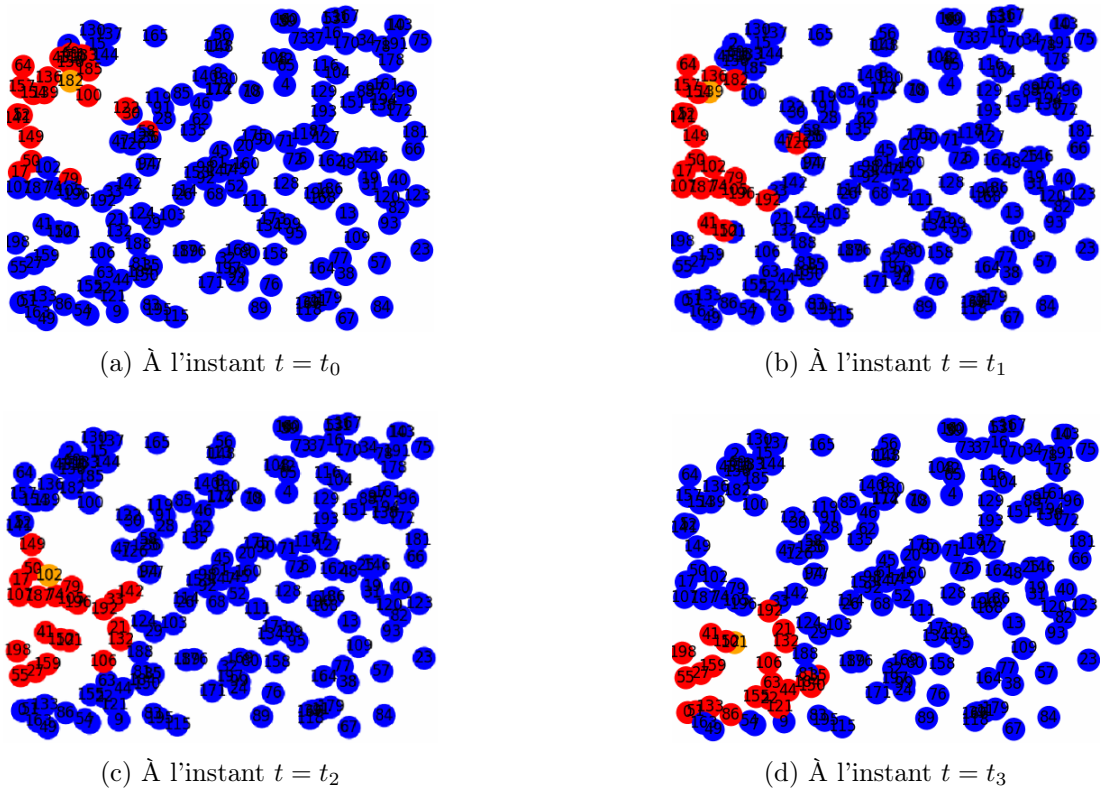
Dans ce modèle, on suppose que le graphe est complet, ce qui signifie que chaque nœud peut communiquer directement avec n'importe quel autre nœud. Cette complétude est généralement assurée par des superpositions P2P (peer-to-peer) utilisées pour acheminer les messages dans les systèmes distribués.

L'objectif du problème est de trouver une clique de taille spécifiée qui minimise les poids associés au graphe, en se concentrant sur les délais de réponse (RTT). L'idée principale est de sélectionner un sous-graphe qui optimise une certaine mesure de distance, en supposant que les nœuds plus proches puissent communiquer plus rapidement et accélérer le protocole ou les échanges d'informations.

En résumé, le problème consiste à sélectionner une clique dans un graphe complet pondéré, en utilisant le délai de réponse (RTT) comme mesure d'efficacité. L'objectif est d'optimiser la communication entre les nœuds en favorisant les connexions les plus rapides, afin d'accélérer le protocole ou les opérations réalisées dans le système distribué.

3.6.1 Énoncé du problème

Il est difficile de trouver une solution exacte pour le problème de sélection d'un sous-graphe de taille donnée qui minimise une mesure précise, comme la somme des poids des arêtes. Ce problème est en effet connu sous le nom de problème de recherche de clique, et il est classé parmi les problèmes NP. L'objectif est de déterminer une clique de taille n dans un graphique complet de taille N, en minimisant la somme des poids des bords, comme spécifié dans le système d'équation (3.9).



(a) À l'instant $t = t_0$

(b) À l'instant $t = t_1$

(c) À l'instant $t = t_2$

(d) À l'instant $t = t_3$

FIGURE 3.3 – L'ensemble de validateurs à des instants différents [Leduc et al., 2023].

$$\min \quad g(z) \tag{3.6}$$

$$\text{s.t.} \quad \sum_{j=1}^{\mathcal{N}} z_j = n \tag{3.7}$$

$$z = (z_1, \ldots, z_{\mathcal{N}}) \in \mathbb{B}^{\mathcal{N}} \tag{3.8}$$

$$g(z) = \sum_{j=1}^{\mathcal{N}} z_j \left(\sum_{i=j}^{\mathcal{N}} z_i \, d_{ij} \right) \tag{3.9}$$

Dans un réseau pair-à-pair, où la topologie et les distances entre les nœuds sont connues, il est possible de construire un graphe pondéré en fonction de ces distances. Ce graphe est non dirigé et complet, représentant ainsi la connectivité du réseau. L'objectif est alors de trouver une clique d'une taille spécifique dans ce graphe qui minimise la somme des poids des arêtes.

Pour résoudre ce problème de manière optimale, il faudrait trouver un algorithme qui puisse déterminer efficacement la sélection de la clique minimale optimale. Cependant, étant donné que le problème est NP-complet, trouver une solution exacte pour des graphes de grande taille peut être coûteux en termes de temps de calcul.

En conséquence, des approches alternatives, telles que l'utilisation d'algorithmes d'approximation, d'heuristiques ou de techniques d'optimisation, sont souvent employées. Ces méthodes visent à trouver des solutions sous-optimales dans des délais raisonnables, compte tenu des contraintes imposées par la NP-complétude du problème.

3.6.2 Décentralisation

Un risque potentiel lié à un algorithme de sélection est l'émergence d'un point fixe ou d'un schéma périodique, où un même sous-groupe de nœuds est régulièrement choisi comme validateurs. Une telle concentration du pouvoir va à l'encontre des principes de décentralisation et peut introduire des vulnérabilités similaires à celles d'un tiers de confiance, en conférant un contrôle excessif à certains acteurs du réseau.

Pour éviter ce phénomène, il est crucial de concevoir des algorithmes de sélection intégrant une part d'aléatoire et prenant en compte des critères de diversité. Une telle approche garantit une répartition équitable des responsabilités, réduit les risques de centralisation, et préserve la robustesse ainsi que la résilience du système distribué.

Pour évaluer le degré de décentralisation dans un réseau, on utilise diverses mesures. L'une d'entre elles est l'écart-type (σ) du taux de participation des nœuds. Un écart-type élevé indique une différence significative de participation, signalant un système moins décentralisé. On peut comparer cette mesure de décentralisation à l'écart-type du pire scénario possible (σ_w, où le groupe de validateurs reste inchangé.Cet écart-type dépend du nombre de nœuds et de validateurs, et il peut être calculé selon l'équation (3.10).

$$\sigma_w(n, \mathcal{N}) = \frac{1}{2} \cdot \sqrt{1 - \left(2 \cdot \frac{n}{\mathcal{N}} - 1\right)^2} \tag{3.10}$$

En utilisant l'écart-type du pire scénario, on peut déterminer l'écart par rapport à cette valeur à l'aide de l'écart-type relatif (σ_r), défini par l'équation (3.11). L'objectif est de réduire cet écart-type relatif, indiquant ainsi une diminution par rapport au pire scénario et une progression vers un système plus décentralisé. La réduction de l'écart-type relatif favorise une répartition équilibrée et équitable de la participation entre les nœuds, encourageant ainsi la décentralisation et limitant la concentration du pouvoir dans des sous-groupes spécifiques.

$$\sigma_r(x, n, \mathcal{N}) = \frac{\sigma(x)}{\sigma_w(n, \mathcal{N})} \tag{3.11}$$

L'étude de Liu et al. [Liu et al., 2019b] examine le degré de décentralisation en utilisant l'indice de Gini [Gini, 1912], qui mesure l'inégalité dans une distribution. Dans le contexte de la sélection des validateurs, l'indice de Gini est utilisé pour évaluer la répartition du nombre de fois où chaque nœud est choisi comme validateur. Nous considérons une période infinie et accumulons les sélections de validateurs pour évaluer la décentralisation et l'impact de l'algorithme de sélection.

L'indice de Gini varie de 0 à 1, où 0 représente une égalité parfaite (chaque nœud est sélectionné un nombre égal de fois) et 1 représente une inégalité parfaite (un nœud est exclusivement sélectionné). L'objectif est de minimiser autant que possible l'indice de Gini pour améliorer la décentralisation du système. Bien qu'il soit pratiquement impossible d'atteindre un indice de Gini de 1, réduire cet indice contribue à promouvoir une distribution plus équilibrée des rôles de validateur et renforce ainsi la décentralisation du réseau.

$$G(x) = \frac{\sum_{i=1}^{\mathcal{N}} \sum_{j=1}^{\mathcal{N}} |x_i - x_j|}{2\mathcal{N}^2 \overline{x}} \tag{3.12}$$

Où $x = (x_i)_{1 \leq i \leq \mathcal{N}}$ et x_i représente le nombre de fois que le nœud i est sélectionné comme validateur pendant une période spécifique L'indice de Gini diminue naturellement à mesure que le nombre de validateurs augmente, car cela implique une plus grande participation des nœuds dans la prise de décision. Toutefois, l'indice de Gini est limité par une valeur maximale, notée

G_w, qui dépend du nombre de validateurs n et du nombre de nœuds $\mathcal{N}$, comme défini dans l'équation (3.13).

$$G_w(n, \mathcal{N}) = 1 - \frac{n}{\mathcal{N}} \tag{3.13}$$

Pour évaluer le niveau de décentralisation, il est crucial de comparer l'indice de Gini avec sa valeur maximale. C'est pourquoi nous introduisons le concept de l'**indice relatif de Gini** dans l'équation (3.14).

$$G_r(x, n, \mathcal{N}) = \frac{G(x)}{G_w(n, \mathcal{N})} \tag{3.14}$$

L'indice relatif de Gini représente un terme à minimiser, indiquant une répartition plus équilibrée de la participation entre les nœuds et un niveau de décentralisation plus élevé.

3.6.3 La densité de la clique

En plus de la diversité des nœuds validateurs, le protocole de sélection proposé vise également à minimiser la latence en favorisant la sélection de nœuds qui sont proches les uns des autres au sein de l'ensemble des validateurs. Cette approche permet de réduire les délais de communication et d'améliorer l'efficacité du réseau. La figure 3.3 illustre ce principe en montrant une sélection de 25 validateurs formant une structure dense et géographiquement proche, ce qui reflète une optimisation spatiale du sous-graphe sélectionné. Cette optimisation spatiale se traduit directement par une réduction de la latence entre les nœuds. Dans le système représenté par l'équation (3.9), la problématique de minimisation de la latence est résolue en minimisant la fonction g.

3.7 Sélection centrée dynamique

Dans notre protocole, nous proposons de former initialement un petit groupe de validateurs proches, puis de le déplacer en sélectionnant les voisins du groupe précédent comme nouveaux validateurs. Pour cela, nous utilisons un mécanisme de poids qui rapproche les nœuds mal sélectionnés d'un côté et de l'autre côté, les nœuds régulièrement sélectionnés.
Cette approche permet de créer une nouvelle mesure de distance appelée distance compensée. Cependant, lors de chaque nouvelle sélection, nous privilégions toujours les nœuds les plus proches du centre de l'ancien groupe de validateurs. Ainsi, nous cherchons à maintenir la proximité géographique et à minimiser la latence au sein du groupe de validateurs sélectionnés.

Dans notre protocole, nous considérons deux mécanismes de sélection des validateurs : le premier, décrit dans [Leduc et al., 2023], et le second, proposé dans le cadre de nos travaux, qui présente de meilleures performances selon différentes métriques.
Version originale[Leduc et al., 2023] :
1) Nous utilisons un algorithme déterministe pour déterminer le nœud central, c'est-à-dire celui qui minimise la somme des distances entre ce nœud et les autres nœuds du réseau.
2) Une fois le nœud central sélectionné, nous choisissons les $n-1$ validateurs les plus proches de ce centre pour former l'ensemble des validateurs.

Version optimisée :
Développé dans la section 3.7.1, le processus de sélection des validateurs se déroule en deux étapes complémentaires : 1) La sélection du point central se fait de manière aléatoire parmi tous les nœuds du réseau, sans tenir compte de leurs distances respectives.
2) Après avoir sélectionné le point central de manière aléatoire, nous procédons à la sélection des

$n-1$ validateurs les plus proches de ce centre.

En résumé, notre protocole propose deux approches pour la sélection des validateurs : une sélection déterministe du point central basée sur les distances, et une sélection aléatoire du point central. Ces approches sont suivies de la sélection des $n-1$ validateurs les plus proches du centre choisi. Le processus est répété selon un ensemble de règles pour introduire de la variation dans la sélection des validateurs.

À chaque itération t, nous sélectionnons un nœud j comme centre. À partir de la matrice d'adjacence $(d_{ki})_{k,i\leq\mathcal{N}}$ qui représente les temps de réponse entre les différents nœuds et qui est partagée entre eux, nous calculons une nouvelle distance d'_{ij} entre le centre j et tous les autres nœuds i, en utilisant l'équation (3.15). Cette nouvelle distance d'_{ij} est calculée en tenant compte de la valeur $\kappa_{i,t}$, décrite dans l'équation (3.16). La valeur $\kappa_{i,t}$ est définie de manière récursive pour $t > 2$ et représente la participation du nœud i dans l'ensemble des consensus. Elle permet de déterminer si un nœud se rapproche ou s'éloigne du centre en fonction de sa participation.

$$d'_{ij} = d_{ij} \times e^{\kappa_{i,t}/\gamma} \tag{3.15}$$

$$\kappa_{i,t} = \begin{cases} \kappa_{i,t-1} + \dfrac{1}{n}, & \text{si } i \text{ était validateur} \\ \kappa_{i,t-1} - \dfrac{1}{\mathcal{N}-n}, & \text{sinon} \end{cases} \tag{3.16}$$

Plus précisément, si la valeur $\kappa_{i,t}$ est élevée, cela signifie que le nœud i a une forte participation dans les consensus précédents, et il tendra à s'éloigner du centre j. En revanche, si la valeur $\kappa_{i,t}$ est faible, cela indique que le nœud i a une participation limitée dans les consensus précédents, et il tendra à se rapprocher du centre j. Ainsi, en utilisant la matrice d'adjacence et les valeurs $\kappa_{i,t}$, nous ajustons les distances entre le centre et les autres nœuds à chaque itération, permettant aux nœuds de se rapprocher ou de s'éloigner du centre en fonction de leur participation dans les consensus précédents.

3.7.1 Algorithmes

Version originale "Modèle 1"

L'exécution du protocole de sélection des validateurs présenté dans [Leduc et al., 2023] s'effectue en cinq étapes principales :

1. Séparation des anciens validateurs et des non-validateurs, afin d'établir une distinction en fonction du nombre de participations passées de chaque nœud en tant que validateur, selon l'équation (3.15).

2. Sélection du centre parmi les validateurs actuels pour constituer le prochain ensemble de validateurs. Le centre correspond au validateur minimisant la distance (3.15) avec l'ensemble des autres validateurs.

3. Réajustement des positions : les nœuds ayant été fréquemment validateurs sont éloignés du centre, tandis que ceux ayant été moins souvent sélectionnés en sont rapprochés, en appliquant une seconde fois (3.15).

4. Sélection des $n-1$ nœuds les plus proches du centre (selon la distance (3.15)) comme validateurs pour l'itération suivante.

5. Mise à jour de la valeur κ pour l'ensemble des nœuds : elle est augmentée de $\frac{1}{n}$ si le nœud est sélectionné comme validateur et diminuée de $\frac{1}{\mathcal{N}-n}$ dans le cas contraire, conformément à (3.16).

Le protocole est répété à chaque nouvelle sélection de l'ensemble des validateurs, garantissant ainsi une mise à jour continue de leur composition au fil des itérations.

Version optimisée "Modèle 2"

Dans cette version optimisée du protocole, la sélection du centre est réalisée de manière aléatoire. Cette caractéristique introduit un avantage en termes de sécurité, car elle empêche qu'un attaquant puisse prédire à l'avance quel nœud sera choisi comme centre. En réduisant la prévisibilité, on diminue le risque de comportements stratégiques ou de collusion entre certains nœuds du réseau.

Toutefois, une sélection reposant uniquement sur le hasard pourrait conduire, par simple effet statistique, à ce que certains validateurs soient choisis de manière répétée, tandis que d'autres resteraient longtemps exclus du processus. Un tel déséquilibre compromettrait les objectifs d'équité et de diversité recherchés par le protocole.

Pour éviter ce phénomène, l'algorithme introduit une contrainte basée sur l'historique de participation des nœuds. L'idée est de limiter l'éligibilité des validateurs les plus souvent sélectionnés, tout en donnant davantage de chances aux validateurs moins sollicités. Ainsi, la probabilité d'être choisi comme centre dépend directement du niveau de participation passé.

Formellement, l'ensemble des nœuds éligibles à la sélection comme centre à l'itération t est défini par :

$$C_t^P = \{i \mid P_i^{t-1} \leq m_i^{t-1},\ \forall i \in V_{t-1}\} \tag{3.17}$$

où :

— i désigne un validateur à l'itération $(t-1)$,

— P_i^{t-1} correspond à la participation cumulée du nœud i dans l'ensemble des validateurs jusqu'à l'itération $(t-1)$.

Afin de formaliser le fonctionnement du protocole, nous présentons ci-dessous les différentes étapes de sélection et de mise à jour exécutées à chaque itération :

1. **Sélection du centre :** la détermination du nœud centre dépend de l'itération conformément à (3.17) :
   a- lors de la première itération, le centre est choisi de manière totalement aléatoire parmi l'ensemble des nœuds du réseau,
   b- aux itérations suivantes, le centre est sélectionné aléatoirement parmi l'ensemble C_t^P, défini en fonction de la participation historique des validateurs.
2. **Séparation des nœuds :** distinction entre les validateurs de l'itération précédente et les non-validateurs, afin de tenir compte du nombre de participations passées de chaque nœud, selon l'équation (3.15).

3. **Réajustement des positions :** application d'un mécanisme de correction basé sur la participation : les nœuds ayant été fréquemment validateurs sont éloignés du centre, ceux ayant été moins souvent sélectionnés en sont rapprochés,en appliquant une seconde fois la distance définie en (3.15).
4. **Sélection des validateurs :** les $n-1$ nœuds les plus proches du centre (selon la distance (3.15)) sont désignés comme validateurs pour l'itération suivante.
5. **Mise à jour de la valeur κ :** à la fin de chaque itération, la valeur de κ est réajustée pour l'ensemble des nœuds : si le nœud est sélectionné comme validateur, sa valeur est augmentée de $\frac{1}{n}$, dans le cas contraire, elle est diminuée de $\frac{1}{\mathcal{N}-n}$, conformément à l'équation (3.16).

Ainsi, le protocole combine une composante aléatoire, qui renforce la sécurité en rendant la prédiction des validateurs plus difficile, et une composante corrective, qui garantit une répartition équitable de la participation entre l'ensemble des nœuds du réseau.

3.7.2 Simulations

Afin d'évaluer les performances du protocole proposé, une série de simulations a été réalisée. Pour cela, un réseau de 200 nœuds a été généré en les projetant aléatoirement sur un plan de dimension 50×50, la distance entre deux nœuds étant assimilée au temps de transmission d'un message. Le nombre de validateurs est ensuite fait varier entre la valeur minimale (4) et la valeur maximale (200), chaque configuration donnant lieu à une simulation distincte. Par ailleurs, le paramètre γ, défini de manière arbitraire, est testé avec différentes valeurs afin d'observer son influence sur le comportement du protocole.

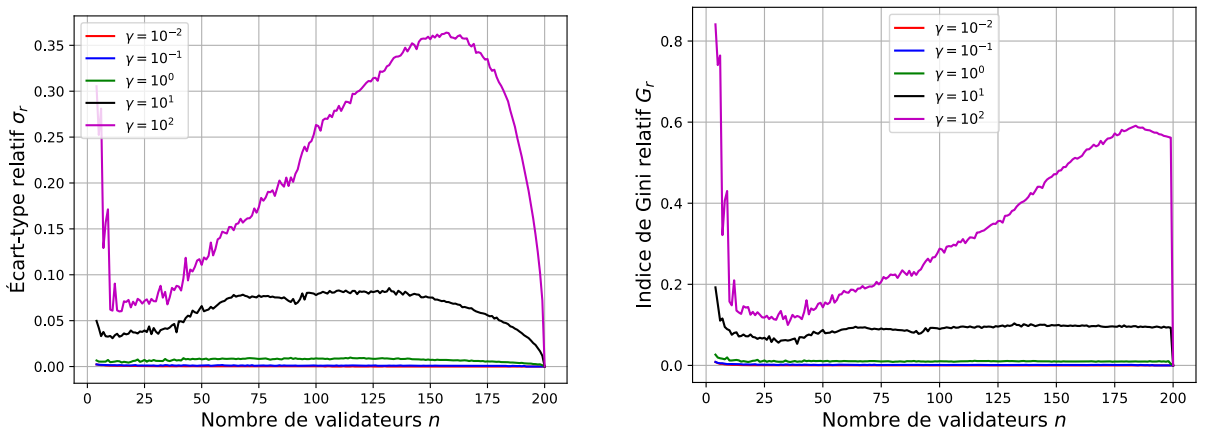
Au début de chaque simulation, le centre est initialement sélectionné de façon aléatoire, puis l'algorithme de sélection est exécuté pendant 200 itérations afin d'atteindre une situation proche de l'état stationnaire. Enfin, 2000 itérations supplémentaires sont effectuées, au cours desquelles des mesures sont collectées à chaque étape, notamment le délai moyen de transmission et la participation des nœuds. Dans ce processus, l'équation (3.17) est utilisée en restreignant l'ensemble C_t^P aux 30% des validateurs ayant le moins participé, de manière à constituer le groupe éligible pour la sélection du centre à l'itération suivante.

Décentralisation :

L'évaluation de la décentralisation est réalisée à l'aide de deux indicateurs : l'écart-type relatif et l'indice de Gini relatif. Ces mesures permettent d'analyser dans quelle mesure la sélection des validateurs est répartie équitablement entre l'ensemble des nœuds du réseau.

Les résultats montrent que les valeurs mesurées lors de la sélection des validateurs restent généralement très faibles, à l'exception notable du cas où $\gamma = 100$. Pour les valeurs de γ inférieures ou égales à 1, les deux indicateurs présentés dans les figures 3.4a et 3.4b se rapprochent de zéro. Cela met en évidence que le protocole proposé favorise une forte décentralisation : les validateurs sont sélectionnés de manière équilibrée parmi les nœuds du réseau, limitant ainsi le risque de concentration du pouvoir sur un petit nombre d'entre eux.

En revanche, lorsque γ prend une valeur plus élevée ($\gamma = 100$), les indicateurs augmentent de façon significative. Cette évolution suggère une concentration accrue des validateurs autour du centre sélectionné, ce qui peut réduire le degré de décentralisation du protocole.



(a) Écart-type relatif

(b) Indice de Gini relatif

FIGURE 3.4 – Mesure de la décentralisation après 2000 itérations en fonction du nombre de validateurs sur une chaîne de 200 nœuds.

Latence optimale :

En plus de la diversité des nœuds validateurs, le protocole de sélection proposé vise également à minimiser la latence en favorisant la sélection de nœuds qui sont proches les uns des autres au sein de l'ensemble des validateurs. Cette approche permet de réduire les délais de communication et d'améliorer l'efficacité du réseau.

Dans le système représenté par l'équation (3.9), la problématique de minimisation de la latence est résolue en minimisant la fonction g.

L'évaluation de la latence dans notre proposition de protocole sera peu coûteuse en termes de calculs, car elle sera minimisée par rapport à un seul nœud appelé "centre". Cela réduit la complexité et les ressources nécessaires pour évaluer la latence dans le système.

Le poids moyen des validateurs peut être utilisé comme un indicateur de la densité de l'ensemble des validateurs, ainsi que de la latence moyenne du comité de validateurs. Lorsque le poids moyen atteint des valeurs élevées, cela signifie que les distances entre les validateurs sont importantes, ce qui indique une faible densité de l'ensemble. En d'autres termes, les validateurs sont dispersés sur la chaîne de 200 nœuds, et il y a moins de validateurs à proximité les uns des autres. Cela peut entraîner une latence plus élevée.

Inversement, lorsque le poids moyen est faible, cela indique des distances plus courtes entre les validateurs, ce qui suggère une densité plus élevée de l'ensemble. En d'autres termes, les validateurs sont regroupés plus étroitement et il y a plus de validateurs proches les uns des autres.

Ainsi, le poids moyen des validateurs peut fournir une indication de la densité de l'ensemble. Comme l'illustre la figure 3.5, lorsque le coefficient d'éloignement γ est inférieur à 1, la latence moyenne est particulièrement élevée en raison de la dispersion marquée des validateurs. Pour $\gamma = 1$, la latence est déjà significativement réduite grâce à une meilleure concentration des validateurs. Enfin, pour des valeurs plus grandes telles que $\gamma = 100$, les validateurs sont regroupés de manière optimale autour du centre, ce qui entraîne une forte diminution du poids moyen et donc une latence minimale.

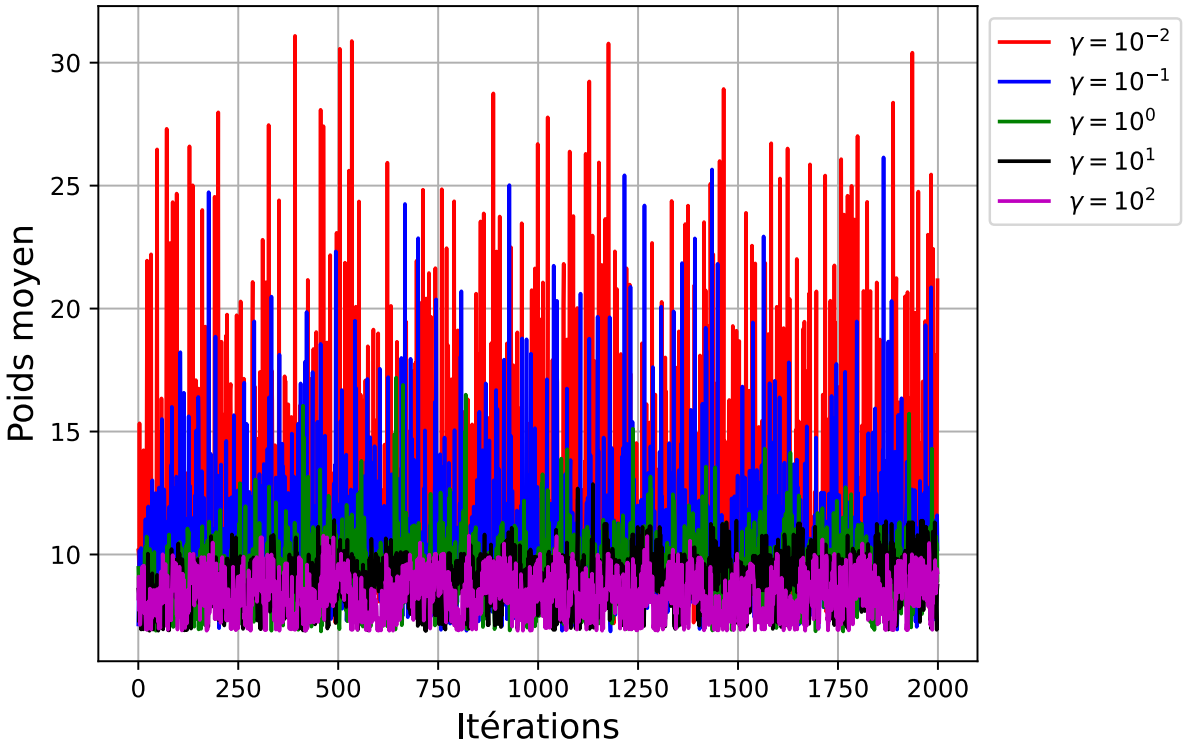
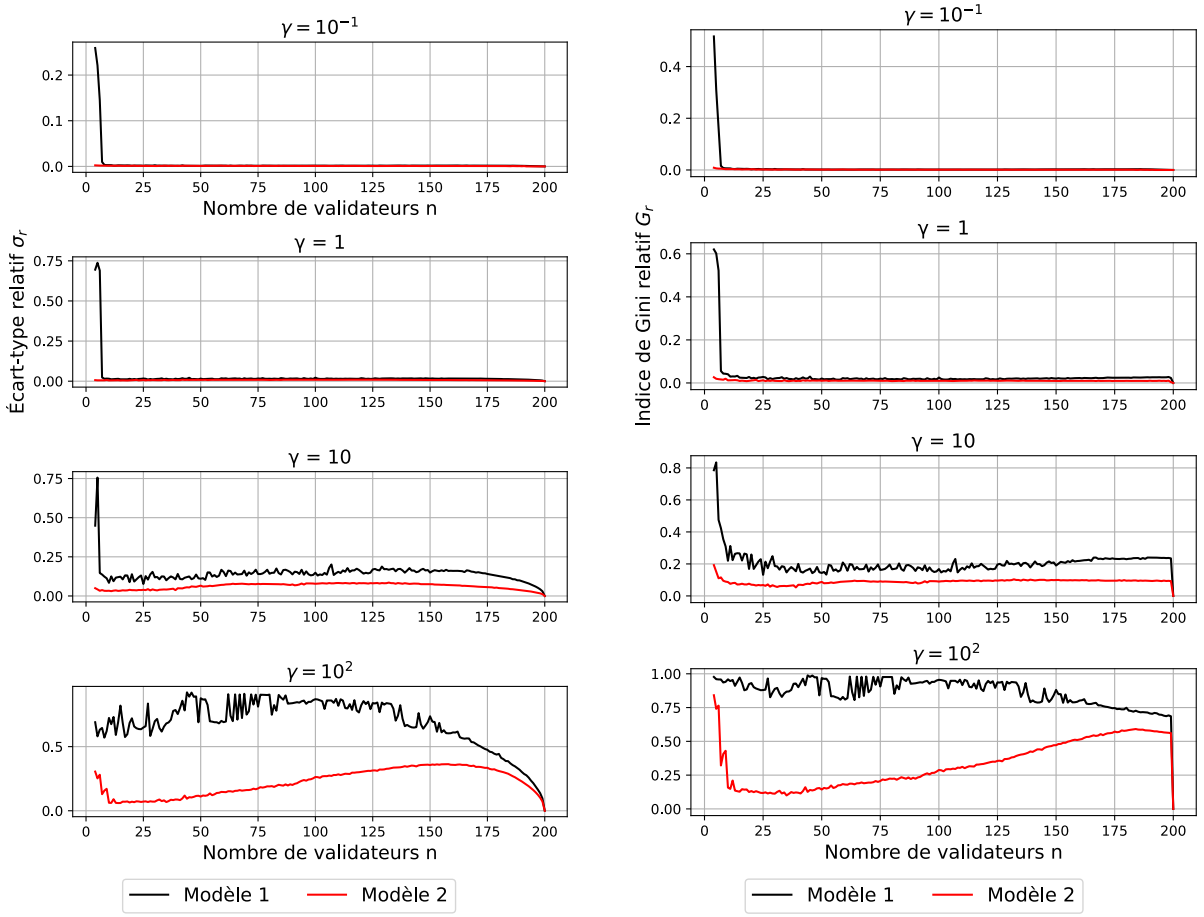


FIGURE 3.5 – Évolution du poids moyen (latence moyenne) des 25 validateurs au cours de 2000 itérations sur une chaîne de 200 nœuds.

Comparaison des résultats

Afin d'évaluer l'impact des optimisations introduites dans le protocole, nous avons procédé à une analyse comparative entre la version originale (modèle 1) et la version optimisée (modèle 2). Cette étude se concentre sur deux indicateurs principaux : la décentralisation des validateurs et la latence du réseau. L'objectif est de quantifier les gains apportés par les modifications proposées et de déterminer leur effet sur la distribution des validateurs ainsi que sur la performance temporelle du réseau.

La figure 3.6 permet d'observer les différences structurelles entre les deux modèles. On remarque que le modèle 1 présente systématiquement des valeurs supérieures pour l'écart-type et l'indice de Gini relatifs, ce qui traduit une distribution plus inégalitaire des validateurs au sein du réseau. Cette tendance se confirme pour l'ensemble des valeurs du coefficient d'éloignement $\gamma \in \{10^{-1}, 1, 10, 10^2\}$. L'analyse dynamique avec un nombre fixe de 25 validateurs (figure 3.7)



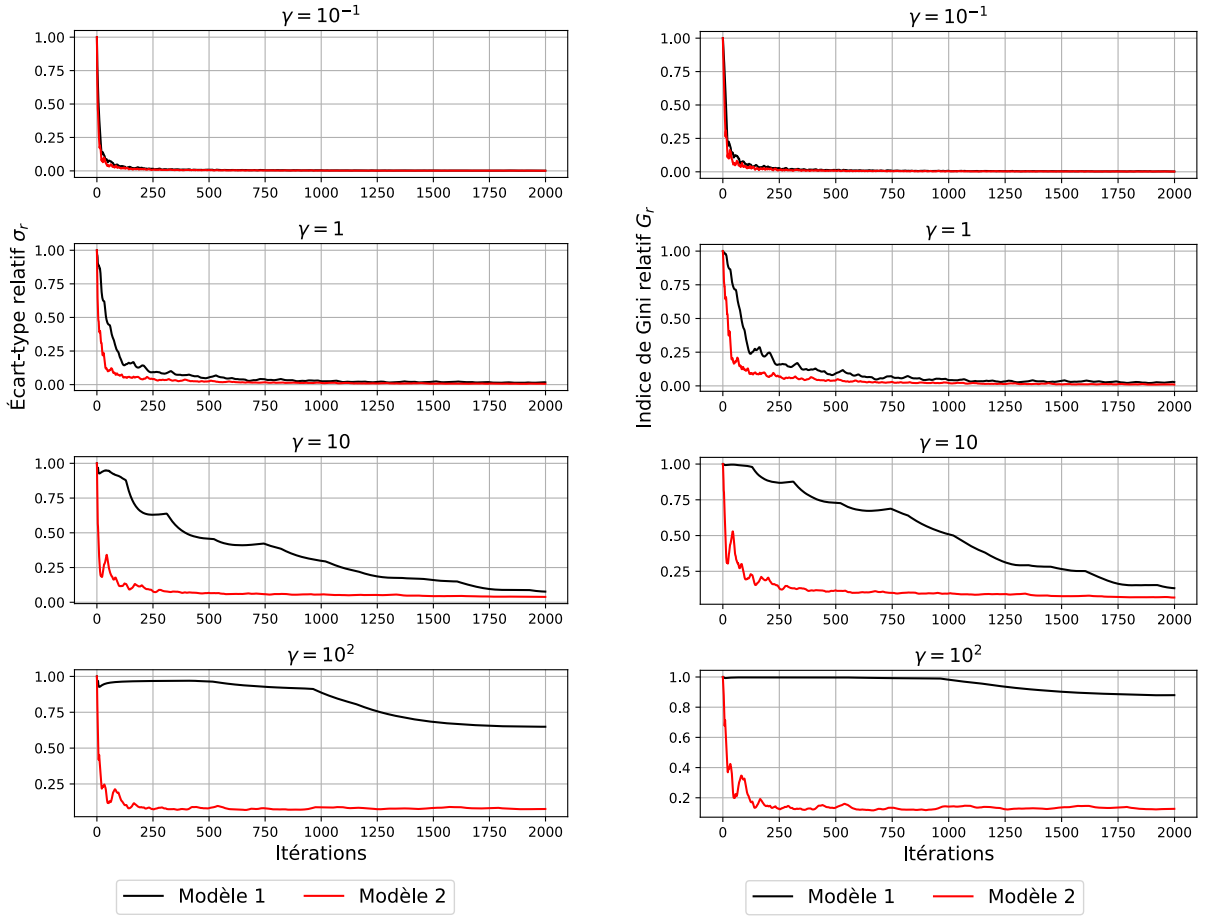
(a) Écart-type relatif

(b) Indice de Gini relatif

FIGURE 3.6 – Mesure de la décentralisation après 2000 itérations en fonction du nombre de validateurs sur une chaîne de 200 nœuds.

confirme la tendance observée précédemment : le modèle 1 conserve des valeurs élevées d'écart-type et d'indice de Gini, traduisant une distribution moins équilibrée. Le modèle 2 maintient quant à lui des valeurs nettement plus faibles et stables au fil des itérations, ce qui démontre la robustesse de la répartition des validateurs même en régime transitoire. L'augmentation de

l'écart-type et de l'indice de Gini dans le modèle 1 indique que certains validateurs sont fortement surreprésentés tandis que d'autres sont sous-utilisés. En revanche, le modèle 2 montre une réduction systématique de ces indicateurs, suggérant une répartition plus homogène et donc une décentralisation plus efficace. La figure 3.8 illustre l'évolution de la latence moyenne pour les



(a) Écart-type relatif

(b) Indice de Gini relatif

FIGURE 3.7 – Évolution de la décentralisation pour $n = 25$ validateurs sur une chaîne de 200 nœuds en fonction des itérations.

25 validateurs. On constate que la latence décroît avec l'augmentation de γ, conformément aux hypothèses du protocole : des valeurs faibles de γ favorisent des validateurs éloignés du nœud central, augmentant le temps de communication, tandis que des valeurs plus élevées sélectionnent des validateurs plus proches, réduisant la latence.En combinant les résultats de décentralisation et de latence, il apparaît que le modèle 2 permet d'atteindre un compromis optimal. Pour des valeurs intermédiaires de γ (10^0 et 10), le protocole assure simultanément un haut degré de décentralisation et une latence maîtrisée. Cette observation met en évidence la nécessité d'un paramétrage fin de γ pour concilier ces objectifs souvent antagonistes.

En conclusion, l'analyse comparative entre le modèle 1 et le modèle 2 démontre que l'optimisation du protocole améliore significativement la distribution des validateurs, tout en maintenant des temps de communication faibles. L'usage de tests statistiques renforçant la significativité des résultats permettrait de quantifier rigoureusement l'amélioration apportée par le modèle 2 et de

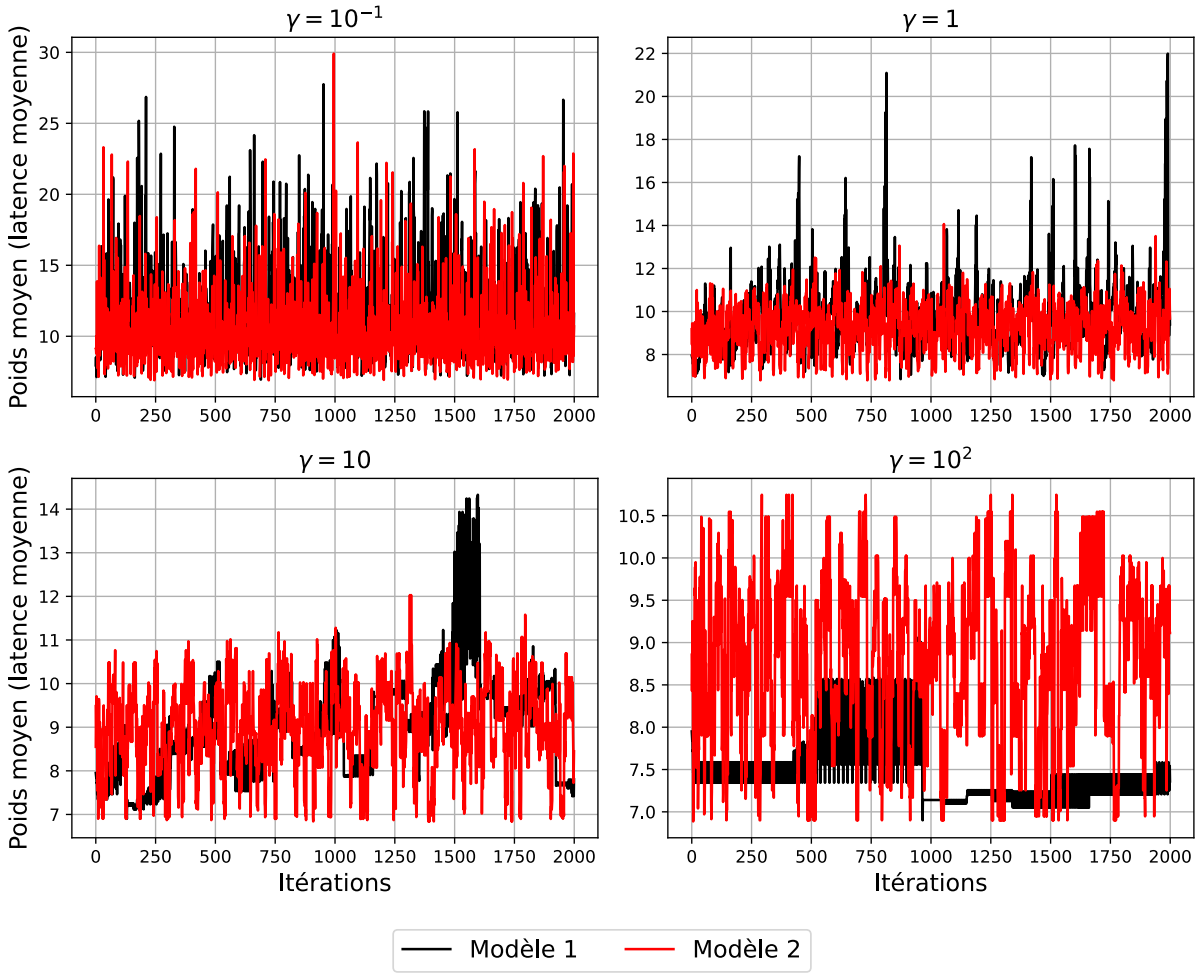


FIGURE 3.8 – Évolution de la latence moyenne pour $n = 25$ validateurs sur une chaîne de 200 nœuds en fonction des itérations.

fournir un argument scientifique solide pour sa mise en œuvre dans des réseaux décentralisés.

3.8 Conclusion

Ce chapitre a présenté une contribution fondamentale à l'optimisation de la sélection des validateurs dans les protocoles BFT. Face au compromis structurel entre performance réseau et décentralisation, nous avons proposé et validé un protocole de sélection centrée dynamique qui introduit une approche novatrice pour maintenir cet équilibre.

Notre modélisation par graphes a formalisé le problème comme une recherche de clique pondérée, tandis que les métriques d'évaluation (écart-type relatif, indice de Gini) ont permis de quantifier rigoureusement la décentralisation. Les simulations sur un réseau de 200 nœuds ont démontré l'efficacité de notre approche, avec le modèle optimisé atteignant des performances supérieures en termes de latence moyenne tout en maintenant une excellente répartition des rôles de validation.

Les résultats soulignent particulièrement l'importance du paramètre γ dans l'arbitrage entre performance et équité : les valeurs intermédiaires (autour de $\gamma = 1$) offrent le meilleur compromis, tandis que les extrêmes privilégient soit la décentralisation (γ faible) soit la performance réseau (γ élevé) au détriment de l'autre critère.

Ces avancées en optimisation de sélection ouvrent naturellement la voie au chapitre suivant, qui étendra cette réflexion en intégrant des dimensions comportementales plus fines. La modélisation des dynamiques d'opinion et l'analyse de similarité comportementale permettront de compléter notre approche en ajoutant une capacité de détection des déviances, essentielle pour renforcer la sécurité des systèmes BFT face à des attaques sophistiquées.

Chapitre 4

Modélisation comportementale et détection des déviances des validateurs

4.1 Introduction

La robustesse des systèmes distribués modernes, en particulier des plateformes blockchain, repose fondamentalement sur la fiabilité de leur mécanisme de consensus. Les protocoles de type BFT représentent l'état de l'art pour garantir la cohérence et la sécurité des réseaux en présence de nœuds défaillants ou malveillants. Cependant, les implémentations classiques comme PBFT souffrent d'une limitation majeure : elles considèrent le comportement des validateurs comme une propriété binaire et statique (honnête ou byzantin), sans capacité d'adaptation aux dynamiques temporelles et aux attaques sophistiquées.

Ce chapitre présente nos contributions fondamentales à la modélisation comportementale et à la détection des déviances dans les réseaux de validateurs BFT. Notre approche s'articule autour de trois axes principaux, structurés en plusieurs sections :

La première partie (Section 4.2) est consacrée à la modélisation par dynamique d'opinion. Nous y développons un modèle probabiliste original qui formalise l'évolution des comportements au sein d'un réseau de consensus. Après avoir présenté le cadre théorique et la formalisation du problème (Sous-section 4.2.2), nous introduisons notre modèle de dynamique par les proportions (Sous-section 4.2.3) qui surmonte le problème de complexité computationnelle grâce à une approximation de champ moyen. Nous détaillons ensuite l'implémentation et l'algorithme d'inférence (Sous-section 4.2.4) avant de valider notre approche par une étude complète basée sur des simulations (Sous-section 4.2.5).

La deuxième partie (Section 4.3) aborde l'analyse de similarité comportementale des validateurs. Nous y définissons un espace d'états comportementaux riche pour les validateurs et introduisons des métriques avancées de similarité, notamment la divergence de Jensen-Shannon (Sous-section 4.3.1). Nous présentons également notre méthodologie de clustering avancé basée sur HDBSCAN pour identifier les patterns comportementaux au sein du réseau.

La troisième partie (Section 4.4) étend notre cadre méthodologique à l'analyse complète des comportements dans un réseau PBFT. Nous y proposons un ensemble de métriques probabilistes complémentaires (coefficient de Bhattacharyya, distance de Wasserstein) et détaillons notre méthodologie de clustering et d'analyse comportementale (Sous-section 4.4.3).

4.2 Dynamique d'opinion des validateurs dans un réseau byzantin

La résilience des systèmes distribués, et en particulier des blockchains, repose fondamentalement sur la robustesse de leur mécanisme de consensus. Parmi ceux-ci, la Tolérance aux Fautes Byzantines constitue un cadre théorique et pratique essentiel pour garantir la cohérence et la sécurité du réseau même en présence de nœuds défaillants ou malveillants [Castro and Liskov, 1999]. Cependant, un écueil majeur des implémentations BFT actuelles, comme PBFT, réside dans leur staticité relative : le comportement des validateurs est souvent considéré comme une entité binaire (honnête ou byzantin) et figé, sans capacité d'adaptation à une évolution temporelle ou à des attaques sophistiquées. La sécurité du système exige que moins d'un tiers de ces validateurs soit byzantin. La sélection de ce sous-ensemble de validateurs est donc critique et fait l'objet de recherches actives : sélection aléatoire [Gilad et al., 2017], basée sur la réputation [Qiu et al., 2022], ou encore optimisée par proximité géographique [Leduc et al., 2023]. Néanmoins, peu de mécanismes intègrent un suivi dynamique et prédictif du comportement des validateurs comme critère central de leur renouvellement.

L'objectif de cette section est de présenter et de formaliser un modèle novateur de *dynamique d'opinion* probabiliste, développé dans le cadre de nos travaux de recherche et publié dans [Guedjali et al., 2025]. Ce modèle a pour ambition de transcender les limitations des approches statiques en offrant un cadre analytique pour :

1. **Analyser et prédire** l'évolution du comportement des validateurs au sein d'un réseau BFT.
2. **Simuler** l'émergence et la propagation d'états ou d'opinions parmi les validateurs.
3. **Détecter précocement** des vulnérabilités de sécurité et des anomalies comportementales susceptibles de compromettre l'intégrité du consensus.
4. **Proposer des mécanismes correctifs** dynamiques, tels que le remplacement ciblé de validateurs, afin de maintenir la résilience du système.

En surveillant la proportion de validateurs adoptant des comportements ou des patterns de vote similaires, notre modèle permet une détection fine et proactive d'activités suspectes. Les simulations menées démontrent son efficacité pour identifier et mitiger diverses formes de menaces, qu'elles soient transitoires ou persistantes, contribuant ainsi à renforcer la sécurité globale du réseau blockchain.

4.2.1 Modèles de dynamique d'opinion

Les modèles de dynamique d'opinion, issus de la physique statistique et de la sociologie mathématique, étudient comment les opinions individuelles évoluent sous l'influence des interactions sociales au sein d'un réseau. Ils se divisent schématiquement en deux catégories :

— **Modèles continus** : Les opinions sont représentées par des variables réelles évoluant de manière continue, souvent via des équations différentielles. Les modèles de Friedkin-Johnsen [Friedkin and Johnsen, 1990] ou de Hegselmann-Krause [Hegselmann and Krause, 2002] en sont des archétypes.

— **Modèles discrets** : Les opinions prennent un nombre fini de valeurs (e.g., binaires : 0/1). Le modèle de l'électeur (*voter model* [Castellano et al., 2009b]) et le modèle de Sznajd [Sznajd-Weron, 2005] en sont des exemples canoniques. Ces modèles sont particulièrement adaptés pour modéliser des processus de vote.

L'application de ces modèles au contexte blockchain, où les "opinions" correspondent aux votes des validateurs sur la validité des blocs, est prometteuse mais encore peu explorée. La principale difficulté réside dans la complexité computationnelle liée au nombre exponentiel d'états possibles du système (2^n pour n validateurs avec un vote binaire), décrite par l'équation maîtresse générale (4.2).

4.2.2 Formalisation du problème

Soit un réseau blockchain utilisant un protocole de consensus BFT et fonctionnant avec un comité de n validateurs. À chaque hauteur de bloc (ou tour de consensus), chaque validateur i émet un vote $v_i(t) \in \{0, 1\}$ sur la proposition de bloc (1 pour "valide", 0 pour "invalide"). L'état global du système au temps t, discrétisé en tours de consensus, peut alors être représenté par le vecteur binaire :

$$\mathbf{m}(t) = (v_1(t), v_2(t), \ldots, v_n(t)) \tag{4.1}$$

Ce vecteur $\mathbf{m}(t)$ décrit complètement la configuration du système à l'instant t, c'est-à-dire l'ensemble des décisions momentanées de tous les validateurs. Ainsi, l'évolution de la blockchain peut être vue comme une succession d'états binaires représentant les différents votes du réseau à chaque tour de consensus.

L'évolution probabiliste de l'état $\mathbf{m}$ est régie par une équation maîtresse qui décrit le taux de variation de la probabilité $P(\mathbf{m}, t)$ que le système soit dans l'état $\mathbf{m}$ au temps t [Castellano et al., 2009a] :

$$\frac{\mathrm{d}P(\mathbf{m}, t)}{\mathrm{d}t} = \sum_{\mathbf{m}'} \left[W_{\mathbf{m}',\mathbf{m}} P(\mathbf{m}', t) - W_{\mathbf{m},\mathbf{m}'} P(\mathbf{m}, t) \right], \tag{4.2}$$

où $W_{\mathbf{m},\mathbf{m}'}$ représente le taux de transition de l'état $\mathbf{m}$ vers l'état $\mathbf{m}'$.

L'équation maîtresse (4.2) exprime le principe fondamental de conservation de la probabilité dans l'espace des états du système. Elle décrit comment la probabilité d'observer une configuration donnée $\mathbf{m}$ varie au cours du temps sous l'effet des transitions vers ou depuis d'autres configurations possibles. Autrement dit, la dérivée temporelle $\frac{\mathrm{d}P(\mathbf{m},t)}{\mathrm{d}t}$ correspond à la différence entre le flux de probabilité entrant dans l'état $\mathbf{m}$ et le flux sortant de ce même état. Cette équation s'applique à tout système stochastique à états discrets où les transitions entre configurations peuvent être décrites par des taux de passage.

La fonction $P(\mathbf{m}, t)$ correspond à la probabilité que le système occupe l'état $\mathbf{m}$ à l'instant t, et elle obéit à la contrainte de normalisation :

$$\sum_{\mathbf{m}} P(\mathbf{m}, t) = 1, \tag{4.3}$$

ce qui garantit que la somme des probabilités sur toutes les configurations possibles reste égale à un, autrement dit, que le système se trouve nécessairement dans un état donné à chaque instant.

Le taux de transition $W_{\mathbf{m},\mathbf{m}'}$ quantifie la probabilité par unité de temps de passer de l'état $\mathbf{m}$ à l'état $\mathbf{m}'$. Ces taux de transition traduisent la dynamique du protocole de consensus : ils représentent la tendance des validateurs à modifier leur vote en fonction des informations reçues du réseau (par exemple, nouveaux messages, observations du comportement des pairs, ou évolution de la proposition de bloc).

Ainsi, le premier terme de l'équation (4.2),

$$\sum_{\mathbf{m}'} W_{\mathbf{m}',\mathbf{m}}\, P(\mathbf{m}', t), \tag{4.4}$$

représente le flux de probabilité *entrant* dans l'état $\mathbf{m}$, c'est-à-dire la somme des contributions provenant de tous les autres états $\mathbf{m}'$ susceptibles d'évoluer vers $\mathbf{m}$. Chaque terme $W_{\mathbf{m}',\mathbf{m}}P(\mathbf{m}',t)$ exprime le produit du taux de transition de $\mathbf{m}'$ vers $\mathbf{m}$ par la probabilité d'être initialement dans $\mathbf{m}'$.

De manière symétrique, le second terme,

$$\sum_{\mathbf{m}'} W_{\mathbf{m},\mathbf{m}'}\, P(\mathbf{m},t), \tag{4.5}$$

désigne le flux de probabilité *sortant* de l'état $\mathbf{m}$ vers l'ensemble des états $\mathbf{m}'$ accessibles depuis celui-ci. Il représente la perte de probabilité associée aux transitions où le système quitte l'état $\mathbf{m}$.

La variation nette $\frac{\mathrm{d}P(\mathbf{m},t)}{\mathrm{d}t}$ résulte donc de l'équilibre entre ces deux contributions : le flux entrant et le flux sortant. Cette formulation met en évidence le fait que l'évolution globale de la probabilité est gouvernée par un mécanisme de redistribution continue entre les états accessibles, tout en préservant la somme totale des probabilités (équation (4.3)). Autrement dit, le modèle décrit la dynamique d'un système conservatif où la masse de probabilité se déplace dans l'espace des configurations au fur et à mesure des interactions entre validateurs.

Cependant, la résolution directe de l'équation maîtresse (4.2) devient rapidement intraitable pour des valeurs réalistes de n. En effet, le nombre total d'états possibles croît de manière exponentielle avec la taille du comité : pour $n = 20$, on obtient déjà $2^{20} \approx 1$ million d'états distincts, et pour $n = 100$, ce nombre dépasse 10^{30}. Cette explosion combinatoire rend impossible toute approche analytique exacte ou même toute simulation exhaustive de la dynamique de $P(\mathbf{m},t)$.

Il est donc nécessaire d'introduire une simplification ou une approximation permettant de réduire la dimension du problème tout en conservant les propriétés essentielles du système. Une approche courante consiste à regrouper les états présentant des caractéristiques équivalentes, par exemple selon le nombre total de votes positifs ou négatifs, de manière à décrire la dynamique à un niveau plus agrégé. Cette réduction permettra, dans la suite du travail, d'obtenir un modèle probabiliste plus simple, capable de capturer la dynamique globale du consensus tout en restant analytiquement ou numériquement exploitable.

4.2.3 Modèle de dynamique par les proportions

Pour réduire la complexité induite par la dimensionnalité élevée, nous proposons de ne plus modéliser l'état complet $\mathbf{m}$ mais une statistique agrégée : la *proportion* de validateurs partageant une opinion ou un comportement commun. Soit $x(t)$ la variable aléatoire représentant la proportion de validateurs votant 1 au temps t. Ainsi, $x(t) \in \{0, \frac{1}{n}, \frac{2}{n}, ..., 1\}$. Cette réduction drastique de l'espace d'état, qui passe de 2^n à seulement $n+1$ états, est rendue possible grâce à l'utilisation d'une *Approximation de Champ Moyen (MFA)* [Gradowski and Krawiecki, 2020].

La MFA repose sur l'hypothèse que les interactions entre les validateurs ne dépendent pas de chaque paire individuelle, mais plutôt d'une interaction moyenne avec l'ensemble du réseau. Autrement dit, chaque validateur perçoit un champ moyen généré par le collectif, ce qui permet de décrire la dynamique globale du système à partir de variables agrégées plutôt que de suivre l'état de chaque valideur de manière explicite.

Équation d'évolution markovienne

Sous l'hypothèse markovienne, l'évolution de la probabilité $P(x,t)$ est décrite par une chaîne de Markov en temps discret, où le temps t est indexé par les tours de consensus.

Nous avons choisi l'hypothèse markovienne car elle permet de modéliser de manière naturelle les attaques par coalition. En effet, dans ces scénarios, l'effet de stratégies coordonnées entre plusieurs validateurs peut être représenté comme un changement instantané de l'état courant du système, sans nécessiter de suivre l'ensemble des décisions passées. Cette approche simplifie l'analyse des comportements stratégiques, tout en conservant la capacité de capturer les dynamiques critiques liées aux coalitions.

Dans ce cadre, chaque tour de consensus agit comme un point de réinitialisation informationnelle où l'état courant, typiquement la proportion x de validateurs - résume l'essentiel de l'information pertinente. Cette variable agrégée intègre les influences des tours passés, permettant de négliger les détails historiques tout en conservant la dynamique essentielle du système. Cette "mémoire limitée" correspond au fonctionnement réel des protocoles BFT où les décisions se basent principalement sur l'état présent, et fournit un cadre analytique traitable pour étudier les attaques de coalition, puisque l'effet des coordinations se traduit directement par des modifications de x.

L'évolution du système est régie par l'équation maîtresse :

$$P(x, t+h) = \sum_{x'} \gamma(x|x')P(x', t) \tag{4.6}$$

avec la condition de normalisation $\sum_x P(x,t) = 1$.

D'un point de vue statistique, l'hypothèse markovienne permet aussi de capturer l'évolution stochastique d'un système multi-agent sans avoir à représenter explicitement toutes les interactions individuelles. Les dépendances temporelles de plus haut ordre (par exemple, la corrélation entre plusieurs tours consécutifs) sont négligeables dès lors que les validateurs actualisent leurs décisions sur la base de l'état global observé au tour précédent, ce qui est le cas dans les protocoles où les messages ou les votes passés n'ont plus d'effet une fois le consensus atteint. Par ailleurs, les protocoles BFT incluent souvent des mécanismes de synchronisation et de redémarrage (timeouts, nouvelles propositions, reconfigurations de comité) qui garantissent la "perte de mémoire" du système entre les tours successifs.

La probabilité de transition $\gamma(x|x')$ représente la probabilité que la proportion x' de validateurs adoptant une stratégie donnée évolue vers x au tour suivant. Pour un système composé de n validateurs, les transitions possibles peuvent être indexées par le nombre de validateurs k changeant d'état entre deux tours consécutifs. On peut alors exprimer la probabilité de transition sous la forme suivante :

$$\gamma(x|x') = \begin{cases} \gamma^{+n}(x') & \text{si } x = x' + 1 \\ \gamma^{+(n-1)}(x') & \text{si } x = x' + \frac{n-1}{n} \\ \vdots & \\ \gamma^{+k}(x') & \text{si } x = x' + \frac{k}{n} \\ \vdots & \\ \gamma^{0}(x') & \text{si } x = x' \\ \vdots & \\ \gamma^{-k}(x') & \text{si } x = x' - \frac{k}{n} \\ \vdots & \\ \gamma^{-n}(x') & \text{si } x = x' - 1 \end{cases} \tag{4.7}$$

où $\sum_{k=-n}^{n} \gamma^k(x') = 1$.

Cette formulation traduit le fait que, d'un tour à l'autre, le système évolue par des changements discrets correspondant à un certain nombre k de validateurs modifiant leur état. L'ensemble des transitions possibles est donc borné entre les cas extrêmes où tous les validateurs changent ($k = n$) et où aucun ne change ($k = 0$).

L'approche markovienne offre enfin un cadre mathématique cohérent pour relier la dynamique microscopique (les décisions individuelles des validateurs) à la dynamique macroscopique (la proportion globale x de votes). Elle permet d'exprimer l'évolution du système, facilitant les analyses de stabilité, de convergence et de fluctuations autour des états d'équilibre. En d'autres termes, elle constitue une abstraction pertinente du comportement collectif des validateurs, en intégrant à la fois les aléas individuels et les interactions globales de manière probabiliste et contrôlable.

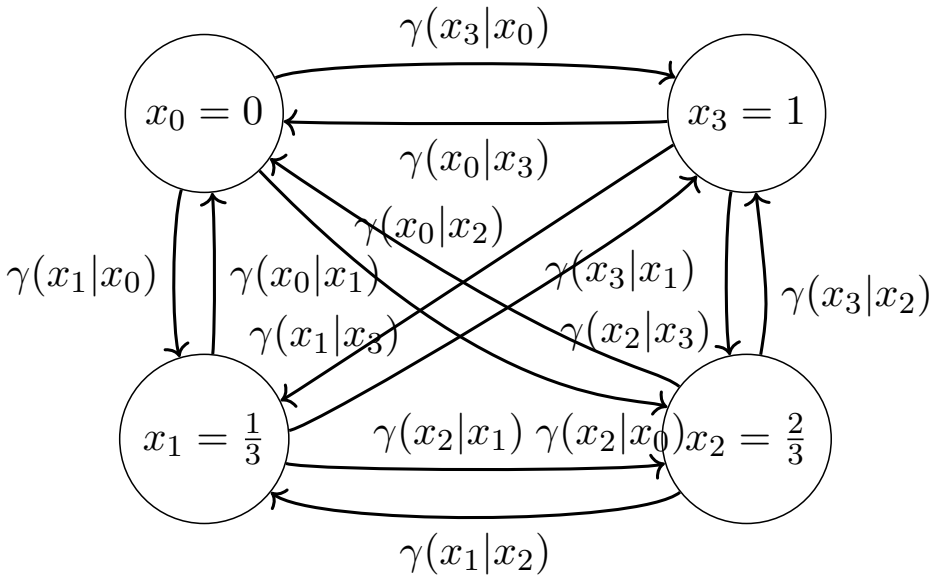


FIGURE 4.1 – Transitions possibles pour la proportion x de validateurs votant 1 dans un système à $n = 3$ validateurs. Les flèches indiquent les probabilités de passage $\gamma(x|x')$ entre états.

La Figure 4.1 illustre le graphe des transitions pour un petit système à trois validateurs. Chaque nœud représente une proportion possible de validateurs votant 1, et chaque flèche correspond à une transition possible entre états. Cette représentation compacte constitue la base pour les méthodes d'approximation, comme l'Approximation de Champ Moyen, permettant de traiter des systèmes beaucoup plus grands.

La concentration et son évolution

La concentration $c(t)$ représente une mesure centrale de la cohésion du système, définie comme l'espérance mathématique de la proportion x de nœuds adoptant un comportement commun au temps t :

$$c(t) = \sum_x x\, P(x,t), \tag{4.8}$$

où $P(x,t)$ décrit la distribution de probabilité des états du système.

L'évolution temporelle de cette concentration s'obtient en considérant son état à l'itération

suivante :

$$c(t+h) = \sum_{x} x\, P(x, t+h). \tag{4.9}$$

En injectant l'équation maîtresse markovienne (4.6) et la formulation discrète des transitions (4.7), on dérive l'expression fondamentale :

$$c(t+h) = \sum_{x'} P(x', t) \sum_{k=-n}^{n} \left(x' + \frac{k}{n} \right) \gamma\big(x' + \tfrac{k}{n} \,|\, x'\big)\,. \tag{4.10}$$

Le développement de (4.10) conduit à la forme opérationnelle :

$$c(t+h) = c(t) + \frac{1}{n} \sum_{x'} P(x', t) \sum_{k=1}^{n} k \left(\gamma^{+k}(x') - \gamma^{-k}(x') \right), \tag{4.11}$$

où $\gamma^{+k}(x')$ et $\gamma^{-k}(x')$ capturent respectivement les transitions augmentant et diminuant la concentration.

L'évolution de $c(t)$ dépend directement de la différence entre les probabilités de transition $\gamma^{+k}(x')$ et $\gamma^{-k}(x')$. En conditions normales, cette différence tend progressivement vers zéro, et $c(t)$ se stabilise autour d'une valeur d'équilibre. En revanche, si des nœuds malveillants coordonnent leurs actions pour fausser le consensus, les transitions deviennent asymétriques : $\gamma^{+k}(x')$ et $\gamma^{-k}(x')$ divergent brusquement, entraînant des variations anormales de $c(t)$. Ainsi, surveiller l'évolution de $c(t)$ permet de détecter de telles attaques, puisque toute tentative de manipulation se traduit par une rupture de la dynamique attendue.

4.2.4 Implémentation et algorithme

L'implémentation du modèle repose sur une architecture computationnelle organisée autour de la matrice de transition $\gamma(x|x')$, qui orchestre les flux de données entre les différents modules. Le schéma fonctionnel présenté en figure 4.2 illustre cette organisation et les dépendances entre les composants.

Le processus algorithmique se décompose en quatre phases principales :

1. **Initialisation des états discrets :**
   - Définir l'espace d'états $\{x_0 = 0, x_1 = \frac{1}{n}, \ldots, x_n = 1\}$ pour n validateurs.
   - Initialiser le vecteur de probabilités $(P_0, P_{\frac{1}{n}}, \ldots, P_1) \in \mathbb{R}^{n+1}$.
   - Construire la matrice de transition $\gamma(x|x')$ avec des valeurs initiales estimées ou uniformes.
2. **Propagation markovienne :**
   - Calculer la distribution à $t + h$ via l'équation maîtresse (4.6).
   - Mettre à jour le vecteur de probabilités dans $\mathbb{R}^{n+1}$.
3. **Calcul des indicateurs de consensus :**
   - Évaluer la concentration $c(t) \in \mathbb{R}$ comme métrique agrégée de cohésion.
   - Surveiller l'évolution de $c(t)$ pour détecter les anomalies.
4. **Adaptation et réestimation :**
   - Mettre à jour $\gamma(x|x')$ à partir des nouvelles observations.
   - Ajuster les paramètres du modèle en fonction des écarts détectés.

Cette architecture permet une estimation séquentielle des probabilités de transition tout en maintenant un cadre computationnel efficace pour la détection d'anomalies dans la dynamique de consensus.

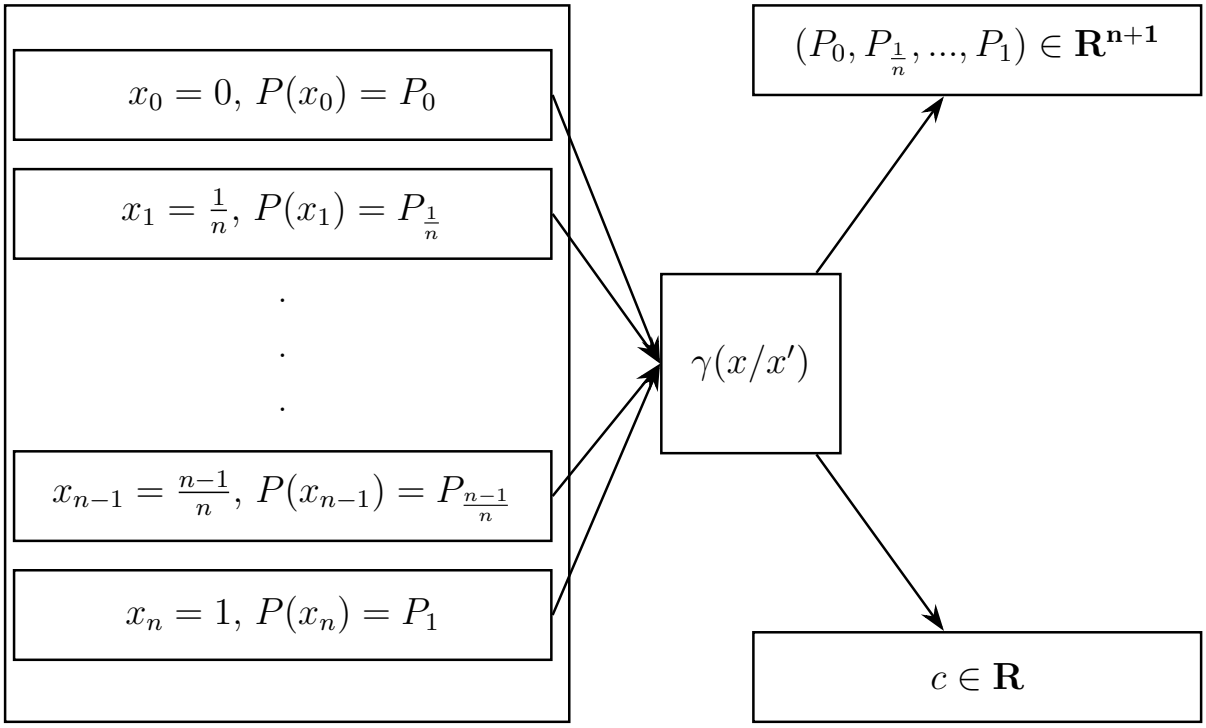


FIGURE 4.2 – Schéma d'architecture du modèle markovien : les états discrets x_i et leurs probabilités $P(x_i)$ interagissent avec la matrice de transition $\gamma(x|x')$ pour produire le vecteur de probabilités et la concentration $c(t)$.

4.2.5 Étude par simulations

Afin de valider notre modèle markovien et d'évaluer sa capacité à détecter les comportements anormaux dans les protocoles de consensus, nous avons mené une campagne de simulations complète basée sur des historiques de vote synthétiques. Ces simulations couvrent un comité de $n = 10$ validateurs et explorent trois scénarios représentatifs : un fonctionnement nominal, une attaque transitoire, et une attaque continue. L'objectif est de quantifier la sensibilité de la concentration $c(t)$ comme indicateur d'anomalie et de caractériser la réponse du modèle aux différentes formes de manipulation. Les paramètres de simulation, incluant notamment l'horizon d'estimation n_γ, sont variés pour analyser leur impact sur la détection.

Scénario de référence (réseau sain)

Dans le scénario de référence, chaque validateur vote indépendamment selon une distribution équilibrée entre les options disponibles. Pour un système binaire, ce comportement correspond à une loi de Bernoulli de paramètre $\mu = 0.5$.

La concentration $c(t)$, définie comme l'espérance mathématique $c(t) = \sum_x xP(x,t)$, évolue selon l'équation d'évolution (4.11). En régime stationnaire, l'équilibre des probabilités de transition assure que $\gamma^{+k}(x') = \gamma^{-k}(x')$, conduisant à une valeur moyenne stable $E[c(t)] = 0.5$.

Les fluctuations autour de cette moyenne sont gouvernées par la somme des transitions individuelles des validateurs. Par le théorème central limite, pour un nombre suffisant de validateurs, la concentration suit approximativement une loi normale :

$$c(t) \sim \mathcal{N}\left(0.5, \sigma_c^2(t)\right)$$

où la variance $\sigma_c^2(t)$ est déterminée par la dynamique des probabilités de transition dans l'équation d'évolution.

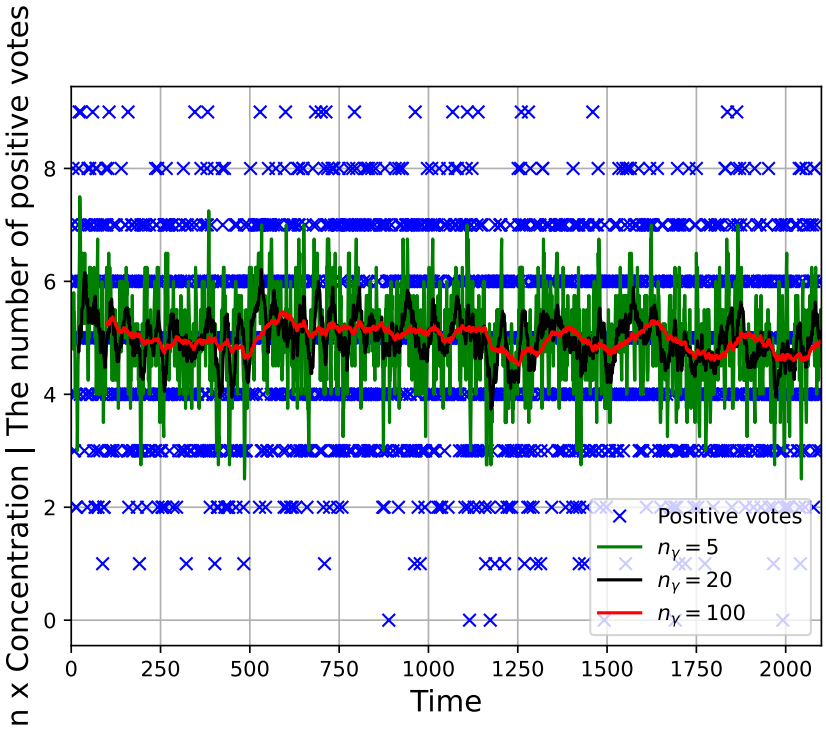


FIGURE 4.3 – Évolution de la concentration $c(t)$ pour différents horizons n_γ dans un scénario de référence.

La figure 4.3 illustre cette dynamique pour $n = 10$ validateurs. L'analyse de l'impact de l'horizon n_γ révèle le compromis fondamental entre réactivité et stabilité : un horizon court ($n_\gamma = 5$) offre une détection rapide mais sensible au bruit, tandis qu'un horizon long ($n_\gamma = 100$) procure une estimation stable mais moins adaptative.

Cette caractérisation du comportement de référence établit la baseline essentielle pour la détection d'anomalies, les déviations significatives par rapport à la distribution normale attendue signalant des comportements pathologiques.

Attaque transitoire

Pour évaluer la capacité de détection du modèle face à des comportements malveillants temporaires, nous simulons une attaque transitoire entre les itérations $t = 500$ et $t = 1000$. Durant cette période, 7 validateurs sur 10 présentent une susceptibilité accrue à la corruption et votent systématiquement 1, indépendamment des conditions de stabilité du protocole. Cette configuration est particulièrement critique car le nombre de validateurs influencés dépasse fréquemment le seuil des deux-tiers (6) requis pour le consensus dans de nombreux protocoles BFT, permettant potentiellement l'approbation de transactions malveillantes.

La figure 4.4 illustre l'évolution du nombre de validateurs votant 1 et de leur concentration $c(t)$ pour $n = 10$. La courbe de concentration révèle une déviation brutale par rapport au

comportement stable observé dans le scénario de référence (figure 4.3), avec des valeurs atteignant régulièrement $0,7$ à $0,9$ pendant la période d'attaque. Ces valeurs dépassent significativement le seuil critique des deux-tiers, signalant une situation de compromission potentielle du consensus.

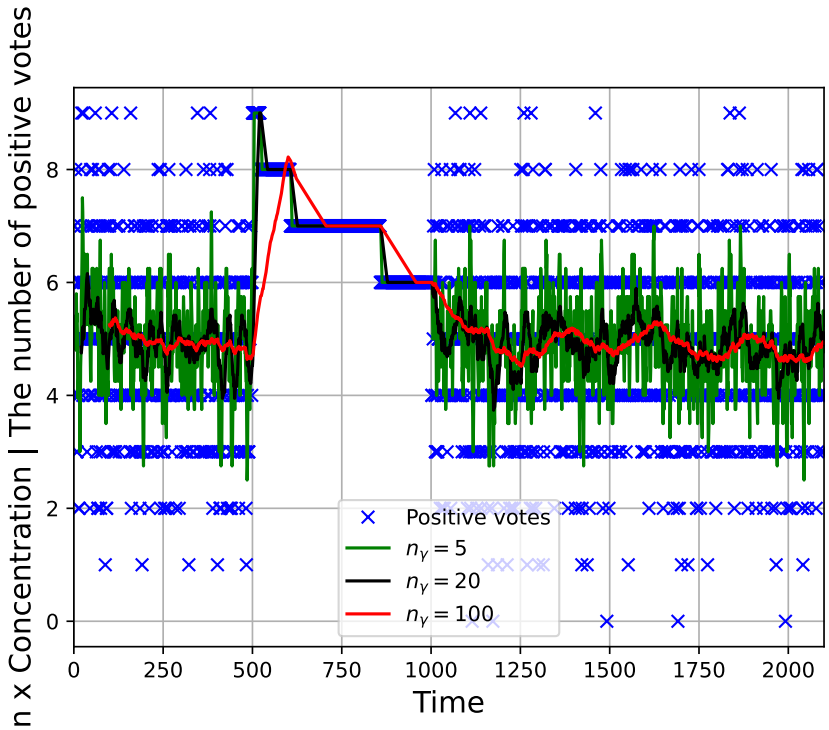


FIGURE 4.4 – Évolution du nombre de validateurs votant 1 et de leur concentration au cours du temps, $n = 10$.

L'analyse de l'impact de l'horizon n_γ révèle des compromis importants pour la détection d'attaques transitoires. Un horizon court ($n_\gamma = 5$) permet une détection plus rapide de l'apparition de l'attaque grâce à sa sensibilité accrue aux variations récentes, mais au prix d'une plus grande instabilité dans l'estimation des probabilités de transition. À l'inverse, un horizon long ($n_\gamma = 100$) offre une estimation plus stable et robuste au bruit, mais peut introduire un délai dans la détection de l'attaque et de sa fin.

La figure 4.5 complète cette analyse en présentant l'évolution temporelle de la distribution de probabilité $P(x, t)$ pour $n_\gamma = 20$. On observe une augmentation marquée des probabilités pour les états $x \geq 0,7$ pendant l'attaque, avec des pics prononcés entre $0,7$ et $0,9$ qui correspondent à des situations où 7 à 9 validateurs montrent un comportement corrompu. Cette distribution confirme que le système atteint fréquemment des états où le consensus peut être manipulé.

L'analyse démontre que le modèle détecte efficacement les situations où le nombre de validateurs susceptibles d'être corrompus dépasse fréquemment le seuil de sécurité. La détection rapide de tels écarts permet d'engager des contre-mesures immédiates, telles que le remplacement des validateurs compromis, avant que l'intégrité du consensus ne soit définitivement compromise. Cette capacité est essentielle pour maintenir la robustesse des protocoles BFT face aux attaques transitoires.

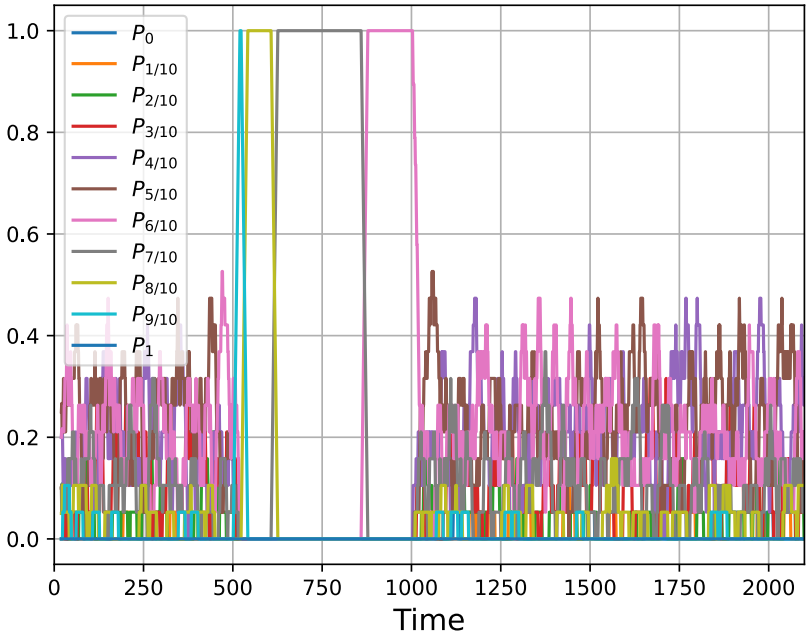


FIGURE 4.5 – Probabilité de chaque proportion de validateurs votant 1 au cours du temps, $n = 10$, $n_\gamma = 20$.

Attaque continue

Ce scénario modélise une attaque persistante où un adversaire prend progressivement le contrôle d'un nombre croissant de validateurs sur une période étendue. Contrairement à l'attaque transitoire, cette stratégie vise à établir une influence durable sur le réseau en corrompant graduellement les validateurs, simulant une compromission qui échapperait aux détections ponctuelles.

La figure 4.6 illustre l'évolution de la concentration $c(t)$ lors de cette attaque progressive. On observe une dérive persistante de $c(t)$, reflétant l'augmentation continue de l'influence des validateurs corrompus. Cette progression contraste fortement avec la stabilité du scénario de référence et la brusque variation de l'attaque transitoire.

L'analyse de la dérive de $c(t)$ permet une détection précoce de la compromission progressive. Le modèle identifie plusieurs phases critiques :

— Phase initiale ($c(t) < 0.6$) : détection précoce permettant une intervention proactive.

— Phase critique ($c(t) > 0.67$) : le seuil des deux-tiers est franchi, le consensus devient vulnérable.

— Phase de compromission ($c(t) > 0.8$) : perte de résilience avérée du système

L'impact de l'horizon n_γ se révèle déterminant dans ce scénario. Un horizon long ($n_\gamma = 100$) permet de lisser les fluctuations et de détecter la tendance à long terme, essentielle pour identifier les attaques progressives. À l'inverse, un horizon court ($n_\gamma = 5$) pourrait masquer la progression graduelle en étant trop sensible aux variations locales.

La détection de cette dérive progressive ouvre plusieurs possibilités d'action :

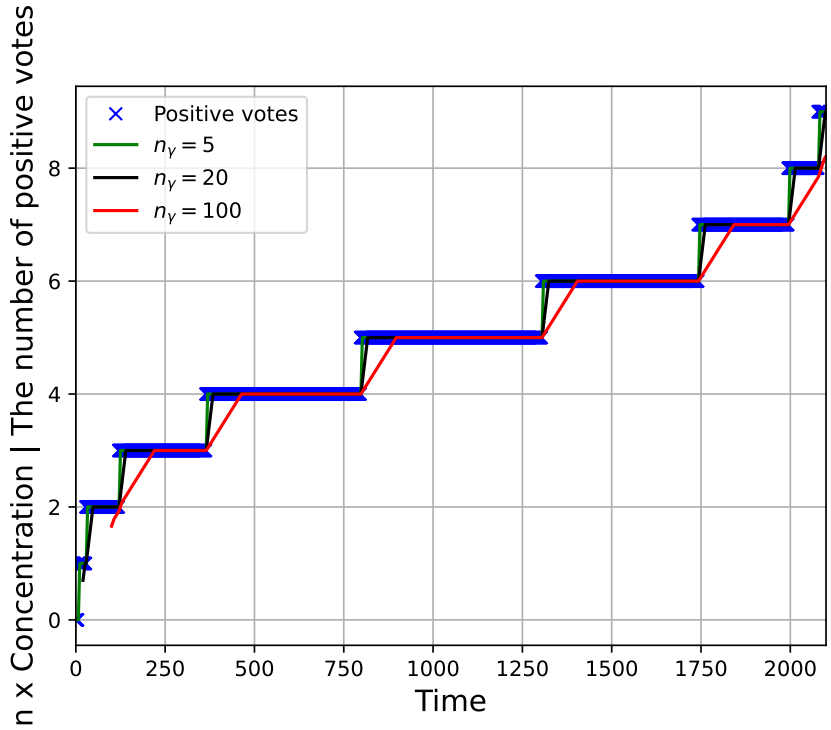


FIGURE 4.6 – Évolution de la concentration $c(t)$ lors d'une attaque continue et progressive. La dérive persistante vers $c(t) = 1.0$ reflète l'influence croissante des validateurs corrompus.

— Remplacement sélectif : identification et remplacement ciblé des validateurs présentant une déviation comportementale

— Alertes précoces : mise en place de mécanismes d'alerte dès que la dérive dépasse un seuil prédéfini

Cette analyse démontre la capacité du modèle à détecter non seulement les attaques brutales mais aussi les compromissions progressives, offrant ainsi une protection complète contre différentes stratégies adverses. La surveillance continue de la concentration $c(t)$ et l'analyse de sa dérivée temporelle constituent des outils essentiels pour maintenir la sécurité à long terme des protocoles de consensus décentralisés.

4.3 Analyse de similarité comportementale des validateurs

Dans un système de type BFT, chaque nœud peut, à un instant donné, occuper l'un des quatre états discrets définis par son rôle dans le processus de consensus et par son alignement comportemental. Soit S_i l'état d'un nœud, où $i \in \{0, 1, 2, 3\}$. Les états sont décrits comme suit (voir codage couleur dans la Figure 4.7) :

— **État S_0 : Validateur majoritaire** (en bleu dans la Figure 4.7)

$$S_0 = \{\text{Validateur} \wedge \text{Majorité}\}$$

Le nœud participe activement au protocole de consensus et s'aligne sur la décision de la majorité. Il incarne le comportement attendu dans un système tolérant aux fautes byzantines. Mathématiquement, cet état correspond à un nœud respectant toutes les spécifications du protocole tout en appartenant au quorum majoritaire.

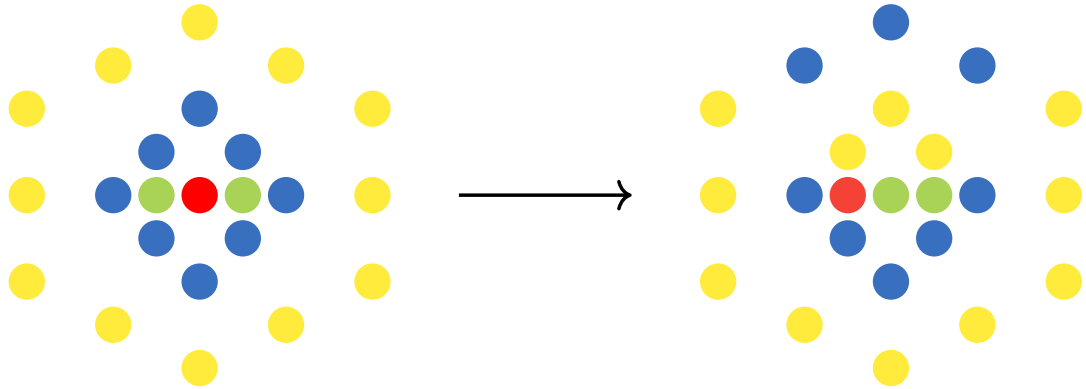


FIGURE 4.7 – Illustration des 4 états des nœud sur deux itérations.

— **État S_1 : Validateur minoritaire honnête** (en vert dans la Figure 4.7)

$$S_1 = \{\text{Validateur} \wedge \text{Minorité} \wedge \neg\text{Défectueux}\}$$

Le nœud est un validateur dont la vision diverge de celle de la majorité, le plus souvent en raison de délais de communication ou d'une information partielle. Bien qu'en minorité, il ne présente aucun comportement fautif. Cette catégorie est essentielle pour distinguer les nœuds légitimement désynchronisés des nœuds malveillants.

— **État S_2 : Validateur minoritaire fautif** (en rouge dans la Figure 4.7)

$$S_2 = \{\text{Validateur} \wedge \text{Minorité} \wedge \text{Défectueux}\}$$

Le nœud adopte une position minoritaire tout en violant les hypothèses du protocole. Cette catégorie regroupe à la fois les nœuds défaillants (pannes logicielles ou matérielles) et les participants adversariaux cherchant intentionnellement à perturber le consensus. La modélisation markovienne permet de capturer les transitions vers cet état pathologique.

— **État S_3 : Non-validateur** (en jaune dans la Figure 4.7)

$$S_3 = \{\neg\text{Validateur}\}$$

Le nœud n'intervient pas dans le processus de validation et demeure passif vis-à-vis des activités de consensus. Cet état peut être temporaire (rotation du comité de validation) ou permanent.

Ces quatre états constituent la base de la modélisation du comportement individuel des nœuds au cours du temps. Ils permettent de caractériser les dynamiques globales d'un réseau BFT en identifiant les rôles et transitions possibles entre les différents participants. La Figure 4.7 illustre cette classification à travers un schéma coloré où chaque nœud est représenté en fonction de son état. Cette représentation visuelle facilite la compréhension de la répartition et de l'évolution des rôles dans le cadre de cycles réels de consensus.

Le profil comportemental d'un nœud v sur un horizon temporel H peut être représenté par un vecteur stochastique défini comme suit :

$$P(v) = \big(P_0(v), P_1(v), P_2(v), P_3(v)\big), \quad \sum_{i=0}^{3} P_i(v) = 1 \tag{4.12}$$

Chaque composante $P_i(v)$ correspond à la probabilité empirique que le nœud v se trouve dans l'état S_i, calculée sur la base des observations historiques. Cette modélisation probabiliste permet de capturer la nature stochastique des comportements dans les systèmes distribués.

En agrégeant ces profils individuels pour l'ensemble des nœuds du réseau, on obtient une structure matricielle qui décrit la distribution des comportements à l'échelle globale. Le Tableau 4.1 présente cette organisation, où chaque ligne est associée à un état S_i et chaque colonne correspond à un nœud v. L'entrée $P_j^{(i)}$ indique ainsi la probabilité que le nœud j occupe l'état S_i.

État	**Nœud 1**	**Nœud 2**	**Nœud 3**	...	**Nœud N**
S_0	$P_1^{(0)}$	$P_2^{(0)}$	$P_3^{(0)}$	...	$P_N^{(0)}$
S_1	$P_1^{(1)}$	$P_2^{(1)}$	$P_3^{(1)}$	...	$P_N^{(1)}$
S_2	$P_1^{(2)}$	$P_2^{(2)}$	$P_3^{(2)}$	...	$P_N^{(2)}$
S_3	$P_1^{(3)}$	$P_2^{(3)}$	$P_3^{(3)}$	...	$P_N^{(3)}$

TABLEAU 4.1 – Matrice de distribution des probabilités d'état pour l'ensemble du réseau.

Cette modélisation permet de capturer la dynamique temporelle des comportements et fournit une représentation continue des stratégies adoptées par les nœuds dans un système BFT. La similarité comportementale entre nœuds peut être quantifiée à l'aide de mesures de distance entre leurs vecteurs de probabilité, permettant ainsi l'identification de groupes homogènes et la détection de comportements déviants par rapport aux normes établies du réseau.

4.3.1 Métrique de similarité

L'évaluation de la similarité comportementale entre deux nœuds u et v repose sur la comparaison de leurs distributions de probabilité respectives. Parmi les différentes mesures envisageables, la divergence de Jensen-Shannon (JSD) [Menéndez et al., 1997] constitue un choix privilégié car elle allie fondements théoriques solides et propriétés pratiques favorables à l'analyse.

La JSD est une version symétrisée et bornée de la divergence de Kullback-Leibler (KL) [Van Erven and Harremos, 2014]. Pour deux distributions $P(u)$ et $Q(v)$, elle se définit par l'équation (4.13) :

$$\mathrm{JSD}(P(u) \parallel Q(v)) = \frac{1}{2}\mathrm{KL}(P(u) \parallel M) + \frac{1}{2}\mathrm{KL}(Q(v) \parallel M) \tag{4.13}$$

où $M = \frac{1}{2}(P(u) + Q(v))$. La divergence de Kullback-Leibler, qui intervient dans cette formulation, est définie par l'équation (4.14) :

$$\mathrm{KL}(P \parallel Q) = \sum_{i=0}^{3} P_i \log_2 \left(\frac{P_i}{Q_i} \right) \tag{4.14}$$

La Figure 4.8 illustre le principe de calcul de la JSD comparé à la divergence KL classique. Le diagramme de gauche montre l'approche unidirectionnelle de la KL, mesurant la divergence de P vers Q, tandis que le diagramme de droite présente l'approche symétrique de la JSD, calculant la divergence de P et Q vers leur moyenne M. Cette représentation visuelle met en évidence le caractère équilibré de la métrique JSD.

L'intérêt de la JSD tient à plusieurs propriétés remarquables. Elle est tout d'abord symétrique, ce qui la distingue de la divergence KL classique : $\mathrm{JSD}(P \parallel Q) = \mathrm{JSD}(Q \parallel P)$. Elle est également

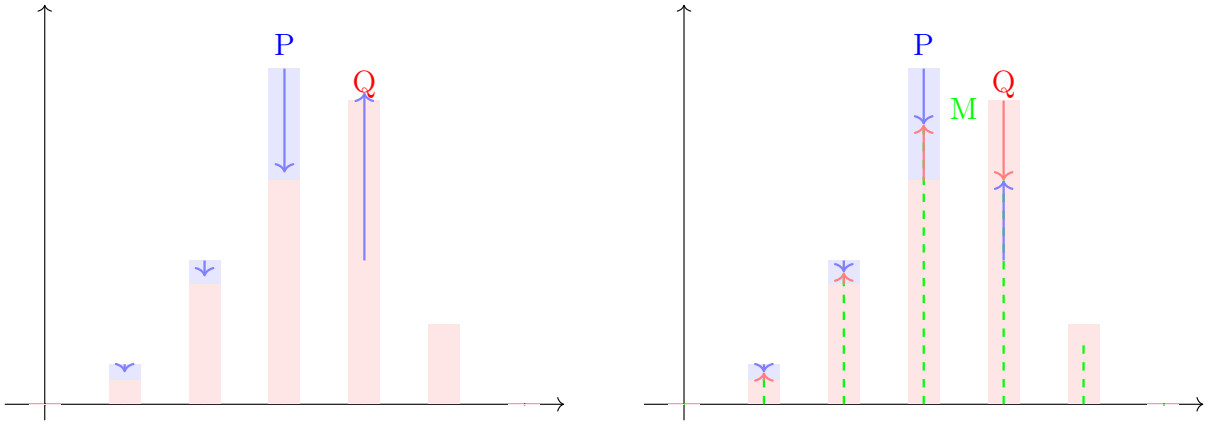


FIGURE 4.8 – Comparaison schématique des divergences : (gauche) divergence KL unidirectionnelle P→Q ; (droite) divergence JSD symétrique entre P et Q via leur moyenne M.

bornée, ses valeurs étant comprises entre 0 (distributions identiques) et 1 (distributions complètement disjointes), ce qui facilite son interprétation. En outre, elle est numériquement stable, puisqu'elle reste bien définie même lorsque certaines probabilités sont nulles, ce qui en fait une mesure robuste face aux données bruitées ou incomplètes.

La figure 4.9 présente la matrice de similarité calculée sur l'ensemble des 10 nœuds du réseau. Cette visualisation permet d'identifier les structures comportementales : les régions sombres indiquent une forte similarité entre nœuds (faible JSD), révélant des clusters comportementaux homogènes, tandis que les régions claires signalent des dissimilarités marquées, potentiellement indicatrices de comportements atypiques.

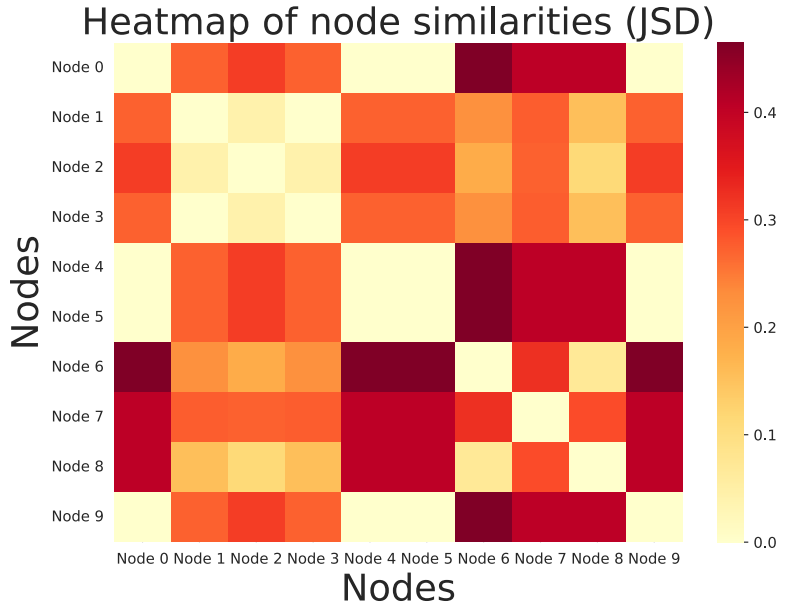


FIGURE 4.9 – Matrice de similarité fondée sur la divergence de Jensen-Shannon, illustrant les relations comportementales entre les 10 nœuds.

Ainsi, la JSD constitue une métrique particulièrement adaptée à notre cadre d'étude : elle

permet non seulement de comparer les comportements des nœuds de manière rigoureuse, mais aussi de révéler des structures au sein du réseau, comme des communautés homogènes ou des profils atypiques. Son application systématique à l'ensemble des paires de nœuds permet de construire une cartographie complète des similarités comportementales, essentielle pour la détection d'anomalies et l'identification de patterns dans les systèmes byzantins.

4.3.2 Méthodologie de clustering avancé

Afin d'identifier des structures comportementales, nous avons retenu l'algorithme HDBSCAN *Hierarchical Density-Based Spatial Clustering of Applications with Noise*) [Campello et al., 2013], une méthode de clustering hiérarchique fondée sur la densité. Contrairement aux approches classiques comme k-means, qui reposent sur des hypothèses fortes et nécessitent la définition préalable du nombre de clusters, HDBSCAN permet de révéler des structures de formes variées et complexes. Son fonctionnement repose sur la construction d'un graphe de voisinage pondéré par une mesure de densité locale, puis sur l'exploration hiérarchique de ce graphe afin d'identifier les zones stables correspondant aux clusters pertinents.

Plusieurs propriétés justifient son choix dans notre contexte. Premièrement, HDBSCAN repose sur une approche hiérarchique density-based, ce qui lui confère la capacité à capturer des relations complexes entre nœuds et à détecter des regroupements de granularité variable. Deuxièmement, il ne requiert pas la fixation a priori du nombre de clusters, l'algorithme sélectionnant automatiquement les partitions les plus stables. Troisièmement, il intègre nativement la gestion du bruit et des points aberrants, en attribuant explicitement certains nœuds au statut d'outliers. Enfin, HDBSCAN accepte des métriques de distance arbitraires : cette flexibilité est essentielle dans notre cadre, puisqu'elle permet d'exploiter directement la divergence de Jensen-Shannon comme mesure de dissimilarité entre distributions comportementales.

La méthodologie adoptée suit une séquence stricte et reproductible. Dans un premier temps, les distributions de probabilité associées aux nœuds sont extraites à partir des comportements observés. Ces distributions servent ensuite à construire une matrice de similarité, obtenue par le calcul entre toutes les paires des divergences de Jensen-Shannon. Cette matrice constitue la base de l'analyse de regroupement : appliquée à HDBSCAN, elle permet de dégager une structure hiérarchique révélant à la fois des communautés homogènes et des comportements atypiques. L'algorithme produit ainsi une cartographie robuste des interactions, en tenant compte du bruit et des variations locales de densité.

Enfin, une analyse longitudinale est menée afin d'agréger les résultats sur plusieurs rounds temporels. Cette étape vise à dépasser une simple photographie instantanée du réseau pour mettre en évidence l'évolution dynamique des comportements. Elle permet d'identifier non seulement la persistance ou la disparition de certaines communautés, mais également l'émergence progressive de nouvelles structures, offrant ainsi une vision plus fine et plus complète de la dynamique comportementale.

4.3.3 Simulations

Dans cette section, nous présentons les simulations mises en place afin d'évaluer l'évolution temporelle du comportement des nœuds dans un réseau BFT, ainsi que l'efficacité de l'algorithme de clustering HDBSCAN appliqué à la métrique de similarité fondée sur la divergence de Jensen-Shannon . L'expérimentation poursuit deux objectifs principaux :

1. analyser comment les comportements individuels des nœuds tendent à se stabiliser ou, au contraire, à diverger au fil des itérations de consensus ;

2. déterminer si ces comportements peuvent être efficacement regroupés et interprétés à l'aide de techniques de clustering non supervisées.

Nous simulons un réseau BFT composé de $N = 50$ nœuds. À chaque itération de consensus, un sous-ensemble de $n = 10$ validateurs est sélectionné aléatoirement, garantissant le respect de la règle de majorité propre au protocole. Chaque nœud est modélisé par une distribution de probabilité sur quatre états comportementaux représentatifs du protocole S_0, S_1, S_2 et S_3. Ces distributions sont générées de manière synthétique afin de refléter différents niveaux d'adhésion au protocole, allant d'un respect strict des règles à des comportements partiellement ou totalement déviants.

L'évolution de l'état des nœuds est observée sur cinq itérations consécutives, de manière à capturer à la fois des comportements stables et transitoires. Les transitions entre états sont gouvernées par des règles probabilistes, introduisant un certain degré de fluctuation afin de reproduire la variabilité aux environnements distribués. Pour chaque tour de consensus, la distribution des états de tous les nœuds est calculée et stockée, tout en vérifiant que le protocole atteint effectivement un consensus valide. Les données ainsi collectées servent de base aux mesures de divergence et à l'analyse de clustering.

Analyse des comportements des nœuds

L'analyse de la dynamique comportementale individuelle des nœuds repose sur l'étude évolutive de leurs distributions d'états au cours des itérations successives du consensus. Pour chaque nœud, nous comparons sa distribution empirique à l'itération t avec celle de l'itération suivante $t + 1$ en utilisant la divergence de Jensen-Shannon (équation 4.13). Cette approche permet de quantifier précisément l'ampleur des changements comportementaux entre tours consécutifs et d'identifier les nœuds présentant une instabilité stratégique.

La figure 4.10 présente sous forme de *heatmap* l'ensemble des divergences calculées entre itérations successives pour chaque nœud du réseau. Cette visualisation permet une analyse comparative fine de la stabilité comportementale sur l'ensemble du système. On observe notamment qu'entre les itérations 4 et 5, 32 nœuds maintiennent des comportements remarquablement stables (faibles valeurs de JSD), tandis qu'entre les itérations 5 et 6, ce nombre augmente à 40 nœuds. Cette progression indique une tendance nette vers la stabilisation comportementale globale du réseau au fil des itérations.

Pour compléter cette analyse microscopique, nous calculons la divergence moyenne entre itérations pour l'ensemble du réseau, constituant ainsi un indicateur macroscopique de la stabilité comportementale globale. La figure 4.11 montre l'évolution de cet indicateur, révélant une décroissance significative à partir de la transition entre les 5ème et 6ème itérations. Cette réduction de la divergence moyenne confirme la stabilisation comportementale du système et l'absence de variations majeures dans les stratégies des nœuds.

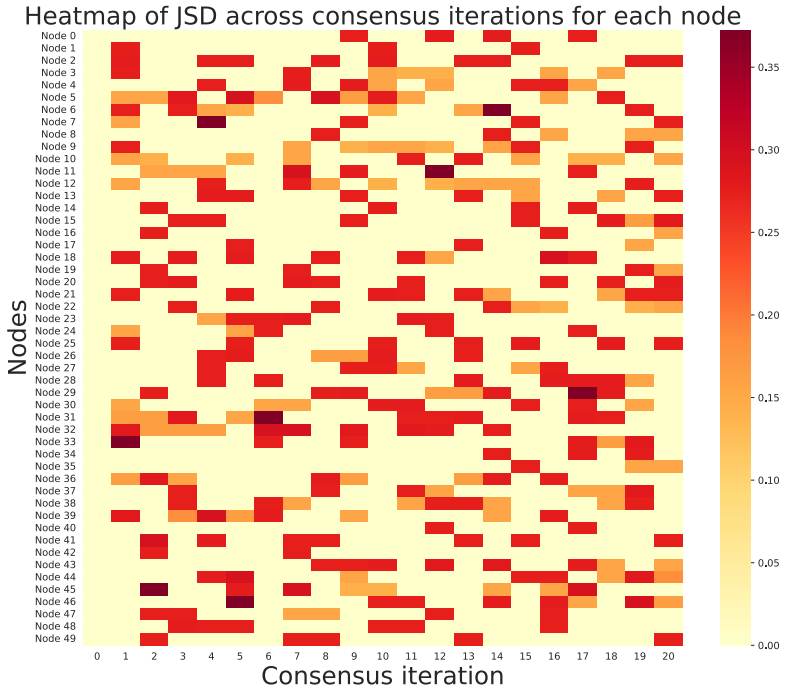


FIGURE 4.10 – Matrice des divergences de Jensen-Shannon entre itérations successives pour chaque nœud. Les valeurs faibles (jaune) indiquent une stabilité comportementale, les valeurs élevées (rouge) une forte variation.

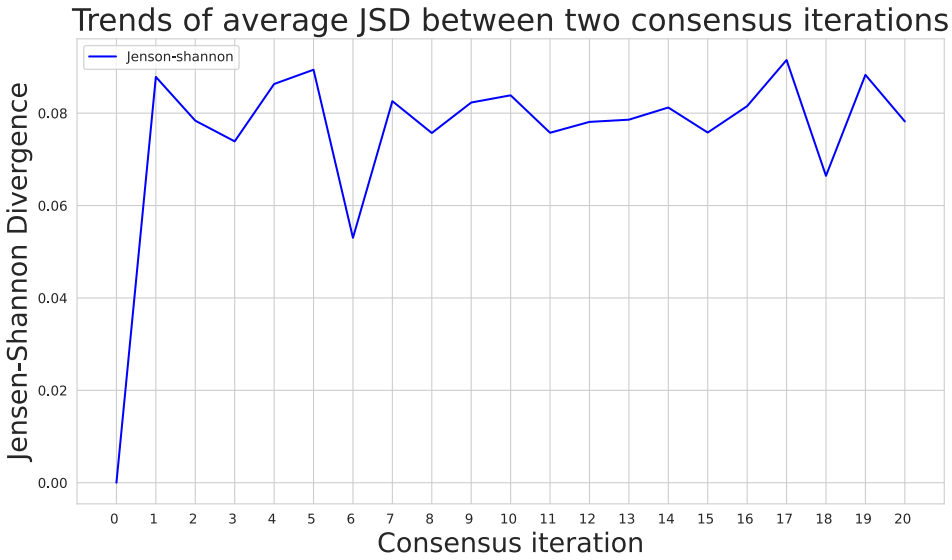


FIGURE 4.11 – Évolution de la divergence moyenne de Jensen-Shannon entre itérations successives. La décroissance observée témoigne de la stabilisation comportementale du réseau.

Résultats du clustering

La matrice de similarité issue des divergences JSD est ensuite utilisée comme entrée pour l'algorithme HDBSCAN. Celui-ci permet de regrouper automatiquement les nœuds selon leurs comportements, tout en identifiant les nœuds isolés ou atypiques.

L'application de HDBSCAN entre les itérations 4 et 5 révèle l'existence de cinq clusters distincts, regroupant au total 18 nœuds stables : 17 appartiennent au cluster 0, et 1 au cluster 1. Ce résultat corrobore l'analyse de la figure 4.10, où la majorité des nœuds présente peu ou pas de similitude notable sur cet intervalle.

En revanche, l'analyse entre les itérations 5 et 6 (figure 4.12) met en évidence une stabilité accrue : 31 nœuds conservent des distributions similaires, répartis en cinq clusters (15 dans le cluster 0, 1 dans le cluster 1, 4 dans le cluster 2, 3 dans le cluster 3, et 8 dans le cluster 4).

Ces résultats diminuent entre l'itération 6 et 7 en terme de nombre de nœuds qui partagent le même comportement pendant ces deux itérations. Cette baisse de stabilité suggère une évolution dynamique des comportements des nœuds dans le réseau.

Ces résultats de clustering confirment la tendance à la stabilisation globale du réseau, déjà suggérée par la décroissance de la divergence moyenne au voisnage de l'itération 6 (figure 4.11).

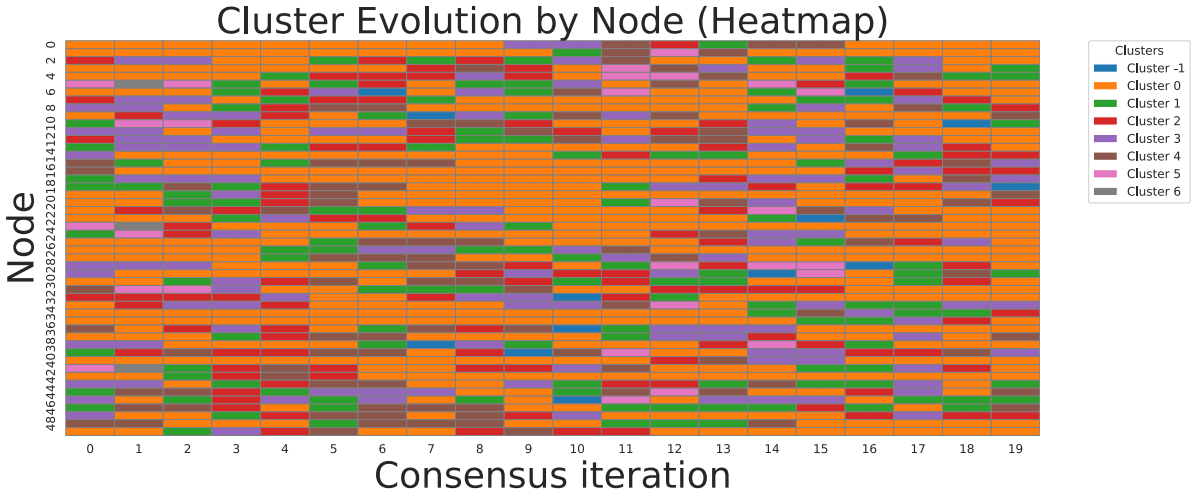


FIGURE 4.12 – Évolution de la stabilité des clusters au fil des itérations de consensus.

Enfin, il convient de souligner les implications sécuritaires de ces résultats. Des nœuds dont le comportement est trop prévisible au fil du temps peuvent devenir vulnérables à des attaques ciblées. Pour renforcer la résilience du protocole, il serait pertinent d'exclure temporairement du processus de sélection des validateurs les nœuds présentant une stabilité excessive sur plusieurs itérations. Une telle stratégie favoriserait la diversité comportementale des validateurs et réduirait le risque d'exploitation malveillante lié à la prévisibilité des comportements.

4.4 Extension de l'analyse de similarité comportementale des validateurs

L'étude des protocoles de consensus byzantins occupe une place centrale dans l'évaluation des systèmes distribués tolérants aux fautes. Cette section étend le cadre conceptuel présenté

dans la section 4.3 en enrichissant l'analyse de similarité comportementale par l'introduction de deux métriques supplémentaires, tout en conservant la même méthodologie de clustering.

Notre approche maintient l'algorithme de clustering précédemment établi mais élargit le cadre d'analyse en intégrant le coefficient de Bhattacharyya et la distance de Wasserstein comme métriques complémentaires à la divergence de Jensen-Shannon. Cette extension permet une caractérisation plus nuancée et robuste des similarités comportementales entre les validateurs.

Le coefficient de Bhattacharyya mesure le degré de chevauchement entre les distributions de probabilité. Sa sensibilité aux similarités dans les parties les moins probables des distributions en fait un complément précieux pour détecter des affinités comportementales subtiles qui pourraient échapper à d'autres métriques. La distance de Wasserstein, quant à elle, capture la distance géométrique entre les distributions en considérant le coût minimal de transformation de l'une vers l'autre, offrant une perspective métrique alternative basée sur la structure spatiale des états comportementaux.

L'objectif de cette extension triple est de mesurer la stabilité et la variabilité des comportements individuels en croisant les résultats des trois métriques pour obtenir une vision plus fiable des profils comportementaux. Nous cherchons également à identifier les regroupements naturels de nœuds à travers une analyse de clustering multi-métrique qui permet de confirmer la robustesse des structures détectées. Enfin, cette approche permet d'analyser la cohérence des similarités comportementales en comparant les résultats obtenus avec les différentes mesures, renforçant ainsi la validité des conclusions tirées.

Cette approche multi-métriques permet une validation croisée des résultats et offre une vision plus complète des dynamiques comportementales dans les réseaux byzantins. La convergence des trois métriques vers des similarités identiques renforce la confiance dans les structures comportementales détectées, tandis que leurs divergences peuvent révéler des aspects complémentaires des relations entre validateurs, tout en maintenant la cohérence méthodologique avec le cadre de clustering déjà établi.

4.4.1 Cadre conceptuel et modélisation

Comme établi à la section 4.3, le comportement d'un nœud est décrit par une distribution de probabilité sur quatre états élémentaires. Cette approche probabiliste permet de dépasser une classification stricte en intégrant des nuances et des transitions.

Un nœud v est ainsi représenté par un vecteur (voir équation 4.12), où chaque composante correspond à la probabilité empirique que le nœud occupe un état donné.

Cette représentation capture aussi bien des comportements fortement concentrés que des comportements plus dispersés.

4.4.2 Métriques probabilistes pour la comparaison

Pour analyser et comparer les comportements entre nœuds ainsi que leur évolution temporelle, il est essentiel de disposer de mesures quantitatives capables de capturer à la fois les différences globales et les similitudes structurelles entre distributions de probabilité.

Dans ce cadre, nous mobilisons trois métriques probabilistes complémentaires, chacune apportant une perspective spécifique sur les relations entre nœuds : la divergence de Jensen–Shannon, qui évalue la différence entropique et fournit un indicateur de divergence globale ; le coefficient de Bhattacharyya, qui quantifie le recouvrement entre distributions et permet de détecter des similarités ; et la distance de Wasserstein, qui mesure le coût minimal de transformation d'une distribution en une autre, en tenant compte de la distance entre états.

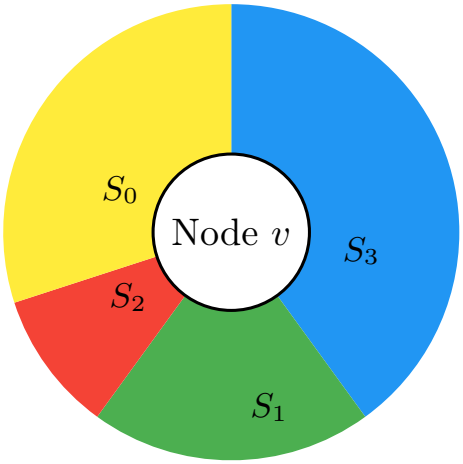


FIGURE 4.13 – Distribution de probabilité des 4 états pour un nœud v.

L'utilisation conjointe de ces trois mesures offre une vision complète de la dynamique des nœuds, permettant d'identifier à la fois des variations subtiles et des correspondances fortes dans les comportements, et constituant ainsi un socle solide pour l'analyse de la stabilité individuelle et collective dans les réseaux étudiés.

Divergence de Jensen–Shannon

La divergence de Jensen–Shannon, présentée en détail dans la sous-section 4.3.1 et formalisée par l'équation 4.13, constitue notre métrique principale. Symétrique, bornée et robuste aux probabilités nulles, elle permet de comparer efficacement des distributions empiriques et de détecter des différences marquées dans les comportements des nœuds. En particulier, cette mesure fournit un indicateur fiable de similarité entre distributions, ce qui est essentiel pour l'analyse des interactions dans les réseaux.

Coefficient de Bhattacharyya

Le coefficient de Bhattacharyya (BC) [Bhattacharyya, 1943], présenté formellement dans l'équation 4.15, constitue une mesure de similarité entre deux distributions de probabilité discrètes. Il quantifie le recouvrement global entre deux distributions en sommant, pour chaque état i, la racine carrée du produit des probabilités associées à cet état :

$$BC(P, Q) = \sum_i \sqrt{P(i) \cdot Q(i)} \tag{4.15}$$

Le BC prend ses valeurs dans l'intervalle $[0, 1]$, où 0 correspond à un absence totale de recouvrement entre les distributions, et 1 à un recouvrement parfait. Cette propriété en fait un indicateur intuitif et direct de la similarité entre distributions discrètes.

Contrairement à la divergence de Jensen–Shannon, qui se concentre sur la divergence entropique et mesure donc la différence globale en termes d'incertitude, le BC met en évidence la similarité entre distributions. Il est particulièrement utile pour détecter des comportements synchronisés ou fortement corrélés entre nœuds dans un réseau, même lorsque les probabilités individuelles diffèrent légèrement. En pratique, l'observation des barres vertes dans la figure 4.14 illustre comment chaque état contribue au recouvrement global, et donc à la valeur finale du BC.

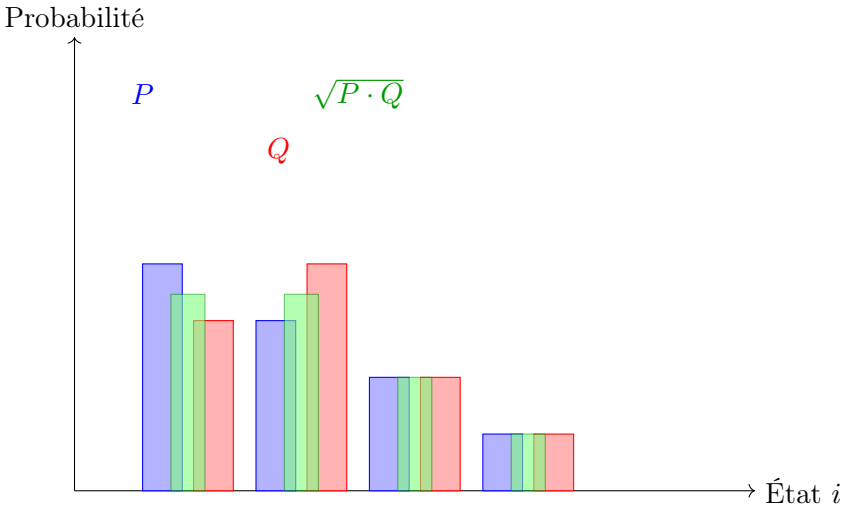


FIGURE 4.14 – Visualisation du coefficient de Bhattacharyya pour deux nœuds, chacun avec 4 états et deux distributions. Les barres vertes représentent $\sqrt{P(i) \cdot Q(i)}$ pour chaque état, et la somme de ces barres correspond à la valeur du BC.

Distance de Wasserstein

La distance de Wasserstein (ou *Earth Mover's Distance*) [Villani and Villani, 2009, Rüschendorf, 1985] mesure le coût minimal pour transformer une distribution P en une distribution Q, en tenant compte de la distance entre états. Elle est définie formellement par l'équation (4.16) :

$$W(P, Q) = \inf_{\gamma \in \Gamma(P,Q)} \sum_{i,j} \gamma(i, j) \cdot d(i, j) \tag{4.16}$$

où $\Gamma(P, Q)$ est l'ensemble des distributions jointes et $d(i, j)$ une distance de base.

Le schéma conceptuel de la figure 4.15 illustre l'idée fondamentale de cette distance pour un nœud à 4 états : les barres représentent les probabilités de chaque état pour les distributions P (bleu) et Q (rouge), tandis que les flèches vertes montrent le transport de masse nécessaire pour transformer P en Q.

Cette métrique est particulièrement adaptée pour détecter des décalages progressifs dans la distribution de probabilité et fournit une interprétation intuitive en termes de "transport de masse" entre comportements.

4.4.3 Clustering et analyse comportementale

Cette sous-section présente une méthodologie permettant de regrouper les nœuds en fonction de leurs distributions comportementales et d'évaluer la qualité des regroupements obtenus à l'aide de la distance de Wasserstein et du coefficient de Bhattacharyya. Cette approche vise à identifier des motifs collectifs pertinents tout en caractérisant la cohésion interne et la séparation entre groupes.

Étape 1 : Collecte des distributions des nœuds

À chaque itération du protocole de consensus, nous extrayons les distributions $P_1, P_2, \ldots, P_N$ décrivant l'état comportemental de chaque nœud. Ces distributions constituent la base de données brute utilisée pour l'analyse ultérieure.

Étape 2 : Construction de la matrice de JSD

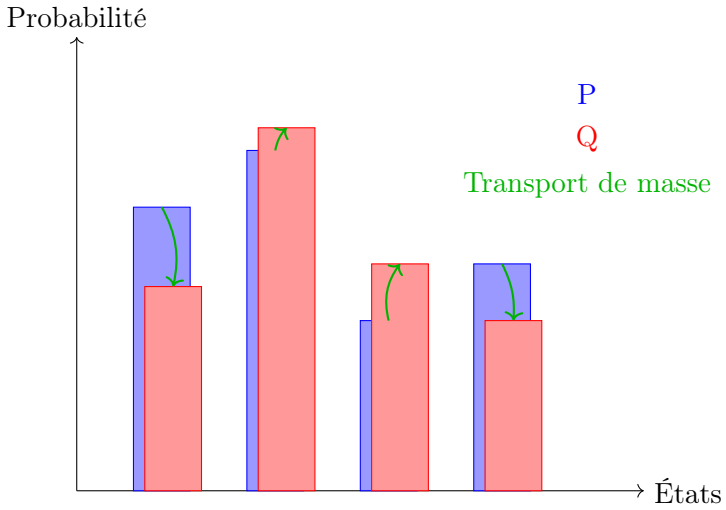


FIGURE 4.15 – Schéma conceptuel de la distance de Wasserstein pour un nœud à 4 états. Les barres représentent les probabilités de chaque état pour les distributions P (bleu) et Q (rouge), et les flèches vertes illustrent le transport de masse nécessaire pour transformer P en Q.

À partir des distributions collectées, nous calculons les divergences de Jensen–Shannon entre chaque paire de nœuds. La matrice obtenue représente la dissimilarité entre comportements et sert de support au processus de clustering. La JSD constitue un choix pertinent pour capturer les écarts entropiques entre distributions empiriques.

Étape 3 : Application de l'algorithme HDBSCAN

La matrice de JSD est ensuite utilisée comme entrée de l'algorithme HDBSCAN. Cet algorithme est retenu pour sa capacité à gérer le bruit et à détecter des clusters de densités variables. Il regroupe les nœuds présentant des comportements similaires tout en isolant ceux qui s'écartent nettement de la majorité.

Étape 4 : Évaluation des clusters *via* Wasserstein et BC

L'évaluation des regroupements repose sur deux critères complémentaires : la distance de Wasserstein et le coefficient de Bhattacharyya. Ces métriques sont mobilisées à deux niveaux :

— **Cohérence intra-cluster :** nous calculons la distance de Wasserstein moyenne et le BC moyen entre les distributions des nœuds appartenant à un même cluster. Une faible distance de Wasserstein et un BC élevé traduisent une forte homogénéité interne.

— **Séparation inter-cluster :** nous mesurons la distance de Wasserstein et le BC entre les centres (distributions moyennes) de chaque cluster. Une grande distance de Wasserstein et un faible BC indiquent une séparation nette entre groupes, gage de comportements distincts.

L'utilisation conjointe de ces métriques permet de combiner une vision entropique et géométrique de la qualité des clusters.

Étape 5 : Détection des outliers et comportements atypiques

Enfin, HDBSCAN identifie également des outliers, c'est-à-dire des nœuds dont les comportements divergent fortement de ceux des clusters détectés. Dans ce cadre, le nombre de clusters émerge directement des données, avec une taille minimale de cluster fixant le seuil d'agrégation. Les nœuds étiquetés comme bruit peuvent refléter des comportements anormaux ou instables et méritent une analyse spécifique, voire une exclusion du processus de consensus afin de préserver

la robustesse globale du système.

Cette méthodologie combine donc un clustering fondé sur la JSD et une évaluation par Wasserstein et BC, offrant un cadre robuste pour analyser le comportement des nœuds dans un protocole de consensus. Elle permet de détecter des motifs collectifs, d'évaluer la cohésion et la séparation des groupes, tout en mettant en évidence d'éventuels comportements atypiques.

4.4.4 Simulations

La configuration expérimentale adoptée s'inspire des protocoles de type BFT, où la simulation de $N = 50$ nœuds permet de capturer les dynamiques essentielles du consensus. Afin d'évaluer les performances de notre approche de regroupement, nous simulons un réseau décentralisé composé de 50 nœuds. À chaque tour, un sous-ensemble de $n = 10$ validateurs (comité) est échantillonné de manière uniforme et aléatoire parmi l'ensemble des nœuds.

Le comportement de chaque validateur est modélisé par une distribution de probabilité discrète sur l'espace d'états $\mathcal{S} = \{S_0, S_1, S_2, S_3\}$.

L'algorithme HDBSCAN est appliqué pour réaliser le clustering, en utilisant la divergence de Jensen-Shannon comme mesure de dissimilarité entre les distributions comportementales. Pour évaluer la qualité des regroupements obtenus, nous calculons la distance de Wasserstein-1 (W1) et le coefficient BC.

La distance W1 est particulièrement adaptée à notre espace d'états ordonné car elle prend en compte la structure métrique entre les états S_i. Elle s'exprime par :

$$W_1(P, Q) = \inf_{\gamma \in \Gamma(P,Q)} \sum_{i,j=0}^{3} |i - j| \cdot \gamma(S_i, S_j) \tag{4.17}$$

où $\Gamma(P, Q)$ est l'ensemble des couplages entre P et Q, et $|i - j|$ représente la distance entre les états S_i et S_j.

Ce protocole de simulation constitue un environnement contrôlé permettant d'analyser la robustesse et l'interprétabilité des résultats de clustering dans un contexte de sélection stochastique des validateurs.

Analyse individuelle des nœuds

Les trois figures présentées ci-dessous illustrent différentes mesures de similarité ou de divergence entre les tours successifs du consensus.

La Figure 4.16 montre la JSD moyenne, une mesure symétrique quantifiant la différence entre deux distributions de probabilité, et permettant d'identifier les variations des distributions de consensus d'un tour à l'autre. La Figure 4.17 présente le coefficient de Bhattacharyya moyen, qui mesure le recouvrement entre deux distributions et renseigne sur leur degré de ressemblance entre tours consécutifs. Enfin, la Figure 4.18 illustre la distance de Wasserstein moyenne, qui correspond au coût minimal de transformation d'une distribution en une autre (souvent appelée « distance du transport de terre »). Chacune de ces mesures apporte un éclairage complémentaire sur les phénomènes de convergence ou de divergence au sein du consensus.

La Figure 4.16 met en évidence un minimum local à l'itération 6, suggérant une diminution de la divergence moyenne entre les distributions comportementales des nœuds d'un tour à l'autre. Cela traduit une plus grande stabilité des comportements à ce stade, avec des variations marginales entre les itérations 5 et 6.

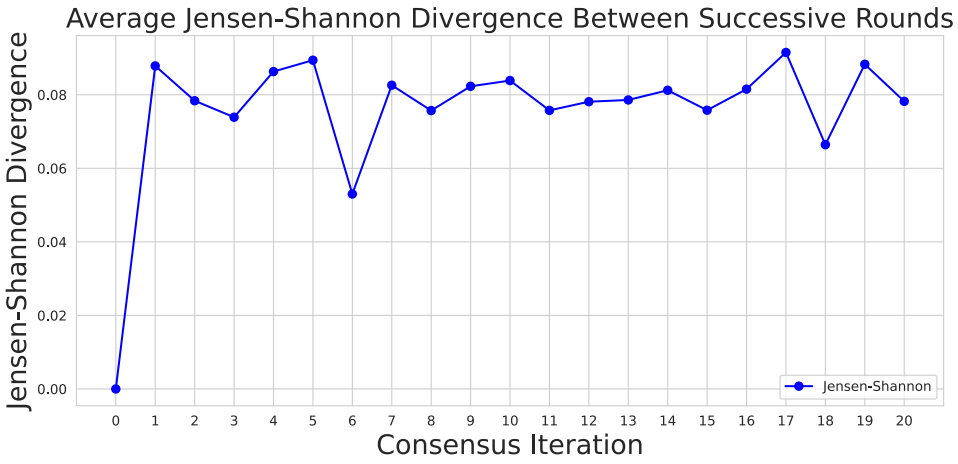


FIGURE 4.16 – JSD moyenne entre tours successifs du consensus.

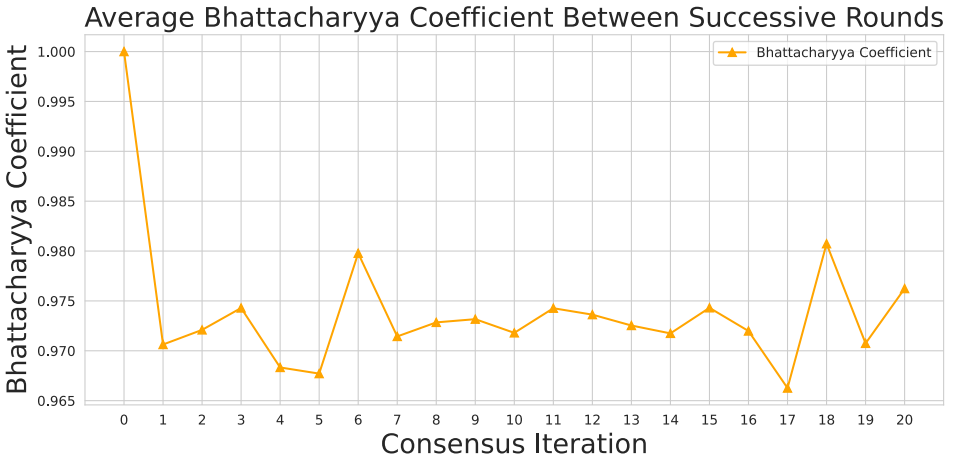


FIGURE 4.17 – Coefficient de Bhattacharyya moyen entre tours successifs du consensus.

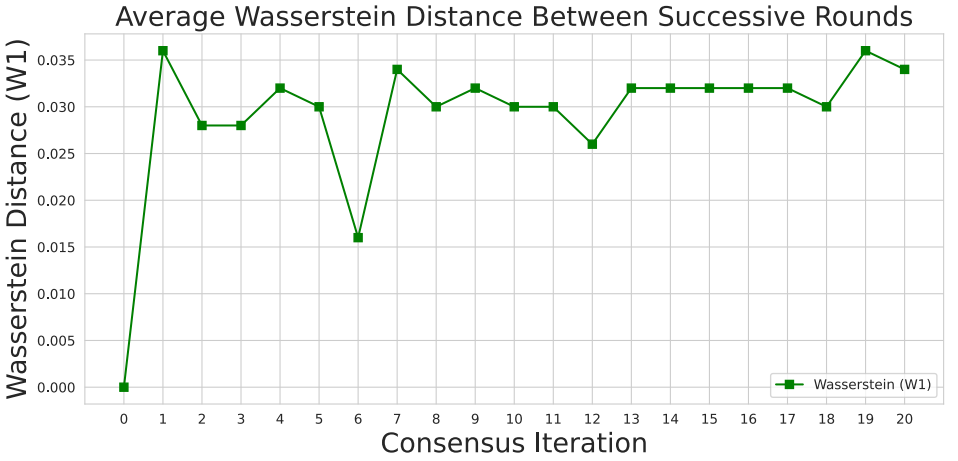


FIGURE 4.18 – Distance de Wasserstein moyenne entre tours successifs du consensus.

Cette observation est confirmée par la distance de Wasserstein (Figure 4.18), qui diminue également à l'itération 6. L'interprétation est que le « coût » requis pour transformer une distribution en une autre est réduit, indiquant un haut degré de similarité entre les comportements des nœuds à ce moment.

De même, la Figure 4.17 présente un pic à l'itération 6, révélant un recouvrement accru entre les distributions des nœuds aux itérations 5 et 6.

Ces métriques comportementales ne permettent pas seulement de caractériser la stabilité des nœuds, mais offrent également des outils pour détecter d'éventuels scénarios d'attaque. Par exemple, des fluctuations soudaines des mesures de divergence peuvent signaler un comportement anormal de certains validateurs, qu'il soit dû à une intention malveillante ou à une mauvaise configuration. De telles déviations sont susceptibles de fragiliser le consensus en introduisant de l'instabilité ou des biais. Pour y remédier, il est nécessaire de mettre en place des mécanismes de détection capables de signaler ces comportements inattendus. Dans ce cadre, les méthodes de clustering jouent un rôle clé, en regroupant les nœuds présentant des profils comportementaux similaires et en facilitant l'identification de sous-groupes potentiellement malveillants.

Détection des motifs comportementaux par clustering

Dans la Figure 4.19, le cluster étiqueté -1 correspond aux nœuds classés comme bruit par l'algorithme HDBSCAN, c'est-à-dire ceux qui ne présentent pas une similarité suffisante pour être intégrés dans un cluster significatif. L'histogramme illustre l'évolution de la taille de chaque cluster identifié au fil des itérations du consensus. On observe que le nombre de clusters varie entre 2 et 5. À l'intérieur de chaque cluster, les nœuds sont regroupés en fonction de leur similarité mesurée par la JSD. Ainsi, les nœuds appartenant au même cluster présentent de faibles valeurs de divergence, ce qui reflète des comportements similaires durant les tours de consensus.

En ce qui concerne les nœuds identifiés comme bruit (-1), les Figures 4.20 et 4.21 montrent que leur dissimilarité est particulièrement marquée. Plus précisément, le coefficient de Bhattacharyya moyen entre leurs distributions est très faible, indiquant un recouvrement quasi inexistant. De même, la distance de Wasserstein moyenne est plus élevée que pour les autres clusters, suggérant des comportements nettement distincts. La taille de ce cluster « bruit » demeure constante et faible à travers toutes les itérations, et ces nœuds ne forment jamais un regroupement cohérent. Cette absence de cohésion laisse penser que leur classification comme bruit résulte davantage des paramètres de HDBSCAN que de véritables régularités comportementales.

Les résultats observés à l'itération 6 dans les Figures 4.16, 4.18 et 4.17 trouvent un écho dans les motifs de clustering présentés aux Figures 4.20 et 4.21. Ces figures révèlent qu'à l'exception du cluster bruit (en rouge) et du cluster 2, les quatre autres clusters affichent une forte similarité interne et une cohésion marquée. Concrètement, ces clusters présentent des valeurs moyennes du coefficient de Bhattacharyya proches de 1 (fort recouvrement) et des distances de Wasserstein moyennes très faibles (faible variation intra-cluster).

La Figure 4.19 confirme que ces quatre clusters cohérents regroupent la majorité des nœuds, tandis que les clusters plus dissemblables (bruit et cluster 2) demeurent marginaux. Cela signifie qu'un grand nombre de nœuds adoptent des comportements similaires à l'itération 6, ce qui corrobore les faibles divergences et distances relevées précédemment. Ces nœuds stables tendent à former de larges regroupements internes homogènes.

Toutefois, une certaine prudence s'impose dans l'interprétation de ces résultats. Les petits clusters caractérisés par une forte dissimilarité peuvent correspondre à des nœuds au comportement atypique ou bruité, susceptibles d'influencer de manière disproportionnée les mesures globales s'ils ne sont pas isolés correctement.

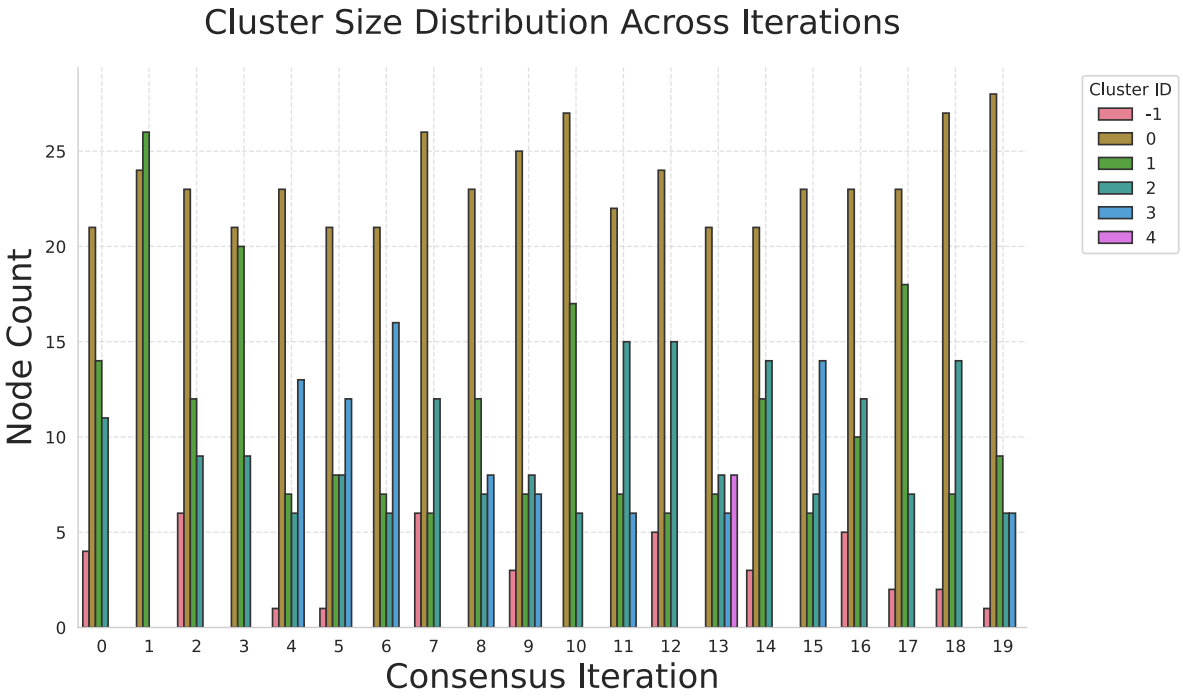


FIGURE 4.19 – Évolution de la taille des clusters au fil des itérations, illustrant la dynamique et la stabilité des regroupements de nœuds.

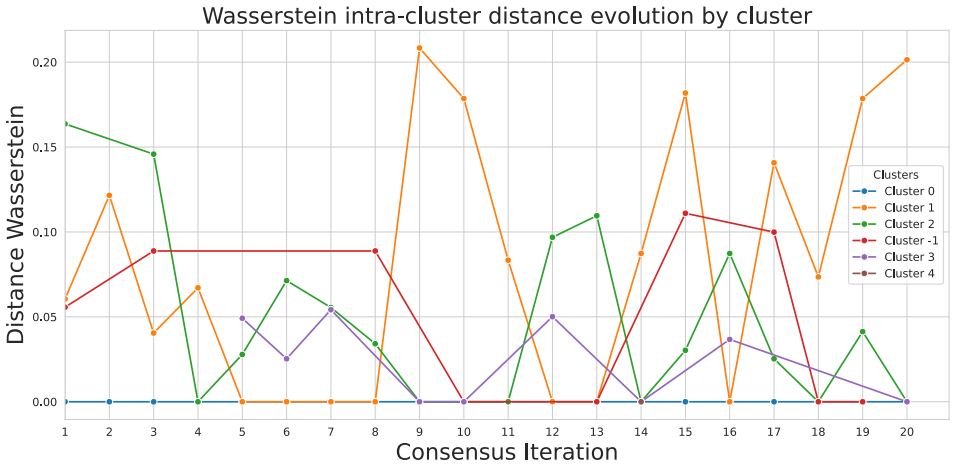


FIGURE 4.20 – Évolution de la distance de Wasserstein intra-cluster.

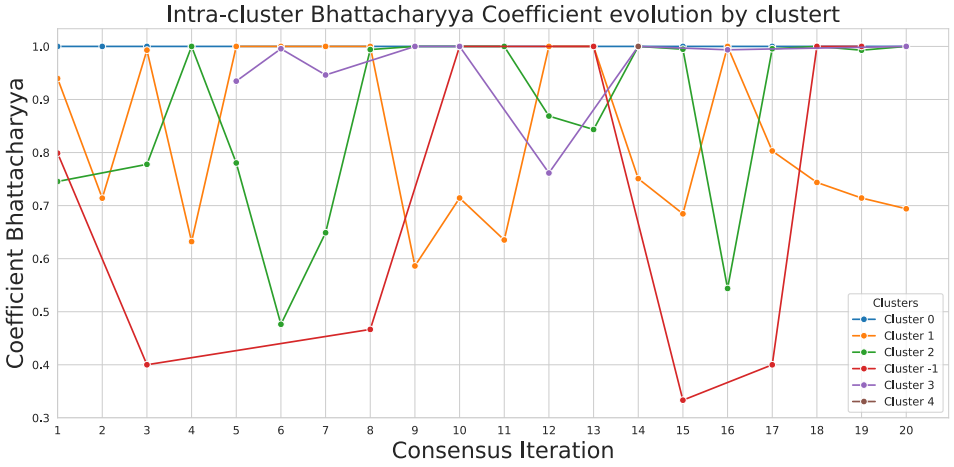


FIGURE 4.21 – Évolution du coefficient de Bhattacharyya intra-cluster.

Enfin, il convient de noter que la forte similarité observée au sein des quatre principaux clusters semble localisée autour de l'itération 6. Autrement dit, il ne s'agit pas d'un phénomène persistant sur l'ensemble des tours, mais plutôt d'une convergence transitoire. Cette stabilité temporaire, bien que significative, est probablement moins déterminante pour les dynamiques globales du système sur le long terme.

4.5 Conclusion

Ce chapitre a présenté un cadre méthodologique complet pour la modélisation comportementale et la détection des déviances dans les réseaux de validateurs BFT, articulé autour de trois contributions principales.

La première contribution, développée dans la Section 4.2, a établi les fondements d'un modèle de dynamique d'opinion probabiliste qui permet de formaliser et prédire l'évolution des comportements au sein d'un réseau de consensus. Grâce à l'approximation de champ moyen présentée dans la Sous-section 4.2.3, nous avons surmonté le problème de complexité computationnelle tout en conservant une puissance prédictive significative. Les simulations de la Sous-section 4.2.5 ont validé la capacité du modèle à détecter diverses formes de déviances, des attaques transitoires aux manipulations continues, fournissant ainsi des métriques quantitatives directement exploitables pour la gestion proactive du comité de validateurs.

La deuxième contribution, présentée dans la Section 4.3, a introduit un cadre formel pour l'analyse de similarité comportementale basée sur la théorie de l'information. La métrique de divergence de Jensen-Shannon développée dans la Sous-section 4.3.1 permet une quantification fine des similarités et différences comportementales, essentielle pour l'identification des patterns anormaux. Couplée à une méthodologie de clustering avancé, cette approche permet de révéler la structure comportementale sous-jacente du réseau.

La troisième contribution, détaillée dans la Section dédiée à l'analyse des comportements dans un réseau PBFT, a intégré modélisation théorique et analyse empirique à travers une méthodologie complète exploitant des métriques complémentaires (Wasserstein, Bhattacharyya). Les résultats des simulations menées sur un réseau de 50 nœuds ont démontré l'efficacité de cette approche pour détecter des motifs comportementaux, évaluer la stabilité du système et identifier les nœuds atypiques avec un haut degré de précision.

Collectivement, ces contributions établissent un nouveau paradigme pour l'analyse comportementale dans les réseaux byzantins. La fusion innovante entre théorie de l'information, modélisation probabiliste et clustering avancé ouvre des perspectives transformationnelles pour les systèmes distribués de nouvelle génération. Les limitations identifiées, notamment concernant l'hypothèse de champ moyen et la modélisation binaire de l'opinion, tracent des voies de recherche futures prometteuses, comme l'intégration d'états de comportement plus riches ou la prise en compte d'interactions pair-à-pair.

Les résultats présentés dans ce chapitre constituent une avancée substantielle vers des systèmes de consensus plus résilients, adaptatifs et sécurisés. Ils posent les fondements théoriques et méthodologiques pour le développement de mécanismes de détection d'anomalies et de gestion dynamique des validateurs qui pourront renforcer significativement la sécurité et la robustesse des plateformes blockchain modernes.

Conclusions et perspectives

Les travaux présentés dans cette thèse doctorale ont examiné les défis fondamentaux posés par la rigidité des mécanismes de sélection et la limitation de la modélisation comportementale au sein des protocoles de consensus tolérants aux fautes byzantines. Ces protocoles, essentiels à la sécurité et à la résilience des systèmes distribués et des blockchains, sont souvent entravés par une approche binaire et statique du comportement des validateurs. Face à ces contraintes, notre recherche s'est concentrée sur l'hypothèse qu'une optimisation dynamique de la sélection des validateurs, couplée à une analyse temporelle fine de leur comportement, pourrait améliorer significativement la robustesse et l'efficacité des systèmes BFT. L'objectif ultime de cette démarche méthodologique novatrice est de s'assurer que l'hypothèse du vote majoritaire byzantin est respectée de manière fiable face aux défaillances et aux attaques coordonnées.

Cette thèse apporte un cadre méthodologique intégré, structuré autour de contributions spécifiques visant à adresser la problématique du compromis entre sécurité et performance. Notre première contribution réside dans le développement d'un protocole de sélection centrée dynamique. Ce protocole vise à trouver un équilibre optimal entre la faible latence et la décentralisation effective, en formalisant le problème comme une recherche de clique pondérée dans un graphe de latences. Pour garantir une rotation équitable des rôles et éviter la concentration du pouvoir, nous avons introduit une métrique de distance compensée, dynamiquement ajustée par l'historique de participation du nœud. Les simulations comparatives entre la version originale et la version optimisée, qui introduit un centre sélectionné aléatoirement, ont démontré la supériorité du modèle optimisé. Ce dernier modèle a montré une réduction significative de l'indice de Gini relatif et de l'écart-type relatif, confirmant une meilleure décentralisation tout en maintenant des latences minimales pour des valeurs intermédiaires du paramètre d'ajustement.

Notre deuxième contribution majeure a établi un cadre formel de modélisation des dynamiques comportementales dans les réseaux BFT, en s'appuyant sur les concepts de la dynamique d'opinion probabiliste. Notre objectif était la recherche des nœuds potentiellement défaillants ou malveillants, afin qu'*in fine*, de futurs travaux puissent les retirer (au moins temporairement) de l'espace des validateurs. Afin de rendre le modèle applicable, nous avons surmonté la complexité exponentielle de l'équation maîtresse en employant l'Approximation de Champ Moyen, réduisant l'espace d'état à seulement des proportions de validateurs partageant une opinion commune. Cette approche permet de suivre l'évolution de la concentration des comportements. Les simulations ont validé la capacité du modèle à détecter précocement les anomalies, qu'il s'agisse de l'augmentation brusque de concentration signalant une attaque transitoire, ou de la dérive persistante lors d'une attaque continue et progressive, menaçant la résilience du consensus. Ce modèle fournit ainsi une base analytique pour le remplacement et proactif des validateurs compromis.

La troisième contribution réside dans le développement d'une méthodologie complète d'analyse de similarité comportementale et de détection des déviances. Cette méthode utilise une représentation comportementale riche sur quatre états de nœuds, capturant les nuances entre le

validateur honnête majoritaire et le validateur minoritaire fautif. Pour quantifier les différences entre les profils comportementaux des nœuds, nous avons mobilisé des métriques probabilistes avancées : la divergence de Jensen-Shannon, utilisée pour sa symétrie et ses bornes, le coefficient de Bhattacharyya, qui mesure le recouvrement structurel, et la distance de Wasserstein, évaluant le coût de transformation entre distributions. Le couplage de ces métriques avec l'algorithme de clustering HDBSCAN a permis d'identifier des patterns comportementaux collectifs et d'isoler les nœuds atypiques. L'analyse expérimentale a confirmé l'efficacité de cette approche pour détecter des motifs comportementaux stables ou fluctuants et évaluer la cohésion des groupes de nœuds.

En combinant ces trois axes de recherche, cette thèse établit les bases d'une nouvelle approche pour la conception de systèmes de consensus byzantins plus adaptatifs et résilients, en intégrant l'observation comportementale fine et des boucles de rétroaction pour l'ajustement dynamique des paramètres opérationnels.

Ces travaux ouvrent des perspectives de recherche particulièrement prometteuses. La première concerne la mise en pratique et la validation expérimentale des modèles et protocoles développés, dans des environnements réels ou simulés à grande échelle (notamment sur la Grid 5000). Nous pensons notamment à un cadre applicatif particulier, à savoir les architectures réseaux Software-Defined Networking ou même de jumeaux numériques, basées sur plusieurs contrôleurs. Pour assurer la sécurité des échange dans ces systèmes à multi-controlleurs, la blockchain, en particulier de type BFT, semble adaptée à la prise de décision distribuée. Cette étape permettra de confirmer l'efficacité opérationnelle, la robustesse et la résilience des approches proposées face aux conditions réseau réelles et aux comportements imprévisibles des validateurs.

Une deuxième direction vise à approfondir la modélisation des interactions comportementales en représentant les nœuds sous la forme d'un graphe pondéré, où les poids reflètent la force d'influence, la similarité comportementale ou la fréquence des échanges. Couplée aux modèles de dynamique d'opinion, cette représentation permettra de capturer les effets d'influence locale, de détecter des communautés comportementales et de suivre la propagation des déviances dans le réseau, offrant une représentation plus fine et réaliste des phénomènes collectifs que les approches traditionnelles.

Enfin, une troisième perspective concerne l'étude du temps de retrait et de réintégration des validateurs détectés comme déviants. La simple détection d'un comportement fautif ne suffit pas ; il est crucial de déterminer un délai de quarantaine adaptatif qui équilibre sécurité et disponibilité, et de définir les conditions d'une réintégration contrôlée, fondée sur la persistance du comportement fautif et des indicateurs de confiance dynamiques. L'étude du cycle complet détection–retrait–réintégration constitue une étape clé vers un consensus capable de s'autoréguler et de s'adapter aux comportements changeants des nœuds.

En conclusion, les fondements théoriques et méthodologiques posés par cette thèse sont destinés à contribuer à l'émergence d'une nouvelle génération de systèmes distribués, qui seront plus robustes, adaptatifs et mieux armés face aux défis sécuritaires du monde numérique de demain. Les approches développées dans ce travail doctoral représentent une avancée vers des protocoles de consensus capables de s'adapter de manière autonome aux conditions changeantes des environnements distribués réels, tout en maintenant des garanties de sécurité robustes contre les attaques sophistiquées.

Publications

Communications dans des conférences internationales avec actes

— **Rachid Guedjali**, Jean-Philippe Georges & Sylvain Kubler. Validator's Opinion Dynamic in a Practical Byzantine Fault Tolerant Network. In *IFAC MATHMOD, 11th Vienna International Conference on Mathematical Modelling*, Vienne, Autriche, février 2025. *IFAC-PapersOnLine*, 59(1) :223-228, 2025.ISSN 2405-8963. doi:10.1016/j.ifacol.2025.03.039. URL `https://www.sciencedirect.com/science/article/pii/S2405896325002563`.

— **Rachid Guedjali**, Jean-Philippe Georges & Sylvain Kubler. Measuring Consensus Stability Through Validator Behavior Patterns in Byzantine Distributed Network. In *IEEE FICLOUD, 12th International Conference on Future Internet of Things and Cloud*, Istanbul, Turquie, août 2025.

— **Rachid Guedjali**, Jean-Philippe Georges & Sylvain Kubler. Behavioral Similarity Analysis of Validators in Byzantine Distributed Control Networks. In *IFAC J3C, 1s Joint Conference on Computers, Cognition and Communication*, Padoue, Italie, septembre 2025.

Bibliographie

Michael Abd-El-Malek, Gregory R Ganger, Garth R Goodson, Michael K Reiter, and Jay J Wylie. Fault-scalable byzantine fault-tolerant services. *ACM SIGOPS Operating Systems Review*, 39 (5) :59–74, 2005.

Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Alexander Spiegelman. Solida : A blockchain protocol based on reconfigurable byzantine consensus. *arXiv preprint arXiv :1612.02916*, 2016.

Ittai Abraham, Guy Gueta, Dahlia Malkhi, and Jean-Philippe Martin. Revisiting fast practical byzantine fault tolerance : Thelma, velma, and zelma. *arXiv preprint arXiv :1801.10022*, 2018a.

Ittai Abraham, Dahlia Malkhi, and Alexander Spiegelman. Validated asynchronous byzantine agreement with optimal resilience and asymptotically optimal time and word communication. *arXiv preprint arXiv :1811.01332*, 2018b.

Ittai Abraham, TH Hubert Chan, Danny Dolev, Kartik Nayak, Rafael Pass, Ling Ren, and Elaine Shi. Communication complexity of byzantine agreement, revisited. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, pages 317–326, 2019.

Shayan E Ali, Noshina Tariq, Farrukh Aslam Khan, Muhammad Ashraf, Wadood Abdul, and Kashif Saleem. Bft-iomt : A blockchain-based trust mechanism to mitigate sybil attack using fuzzy logic in the internet of medical things. *Sensors*, 23(9) :4265, 2023.

Naif Saleh Almakhdhub, Abraham A. Clements, Mathias Payer, and Saurabh Bagchi. Benchiot : A security benchmark for the internet of things. In *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 234–246, 2019. doi:10.1109/DSN.2019.00035.

Bowen Alpern and Fred B Schneider. Defining liveness. *Information processing letters*, 21(4) : 181–185, 1985.

Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. Hyperledger fabric : a distributed operating system for permissioned blockchains. In *Proceedings of the thirteenth EuroSys conference*, pages 1–15, 2018.

Andreas M Antonopoulos. *Mastering Bitcoin : unlocking digital cryptocurrencies.* " O'Reilly Media, Inc.", 2014.

Fredy Andres Aponte-Novoa, Ana Lucila Sandoval Orozco, Ricardo Villanueva-Polanco, and Pedro Wightman. The 51 *IEEE Access*, 9 :140549–140564, 2021. doi:10.1109/ACCESS.2021.3119291.

Sanidhay Arora, Anurag Jain, and S Gujar. Ashwachain : A fast, scalable and strategy-proof committee-based blockchain protocol. In *Workshop on game theory in blockchain at wine*, volume 2020, page 9, 2020.

Adam Back, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra, Jorge Timón, and Pieter Wuille. Enabling blockchain innovations with pegged sidechains. *URL : http ://www. opensciencereview. com/papers/123/enablingblockchain-innovations-with-pegged-sidechains*, 72 :201–224, 2014.

Adam Back et al. Hashcash-a denial of service counter-measure. 2002.

Mathieu Baudet, Avery Ching, Andrey Chursin, George Danezis, François Garillot, Zekun Li, Dahlia Malkhi, Oded Naor, Dmitri Perelman, and Alberto Sonnino. State machine replication in the libra blockchain. *The Libra Assn., Tech. Rep*, 7, 2019.

Emanuele Bellini, Youssef Iraqi, and Ernesto Damiani. Blockchain-based distributed trust and reputation management systems : A survey. *IEEE Access*, 8 :21127–21151, 2020.

Marianna Belotti, Nikola Božić, Guy Pujolle, and Stefano Secci. A vademecum on blockchain technologies : When, which, and how. *IEEE Communications Surveys & Tutorials*, 21(4) : 3796–3838, 2019.

Peter Bennink and Maarten Everts. An analysis of atomic swaps on and between ethereum blockchains using smart contracts. Technical report, 2018.

Alysson Bessani, João Sousa, and Eduardo EP Alchieri. State machine replication for the masses with bft-smart. In *2014 44th Annual IEEE/IFIP international conference on dependable systems and networks*, pages 355–362. IEEE, 2014.

Anil Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distribution. *Bulletin of the Calcutta Mathematical Society*, 35 :99–110, 1943.

Eric A Brewer. Towards robust distributed systems. In *PODC*, volume 7, pages 343477–343502, 2000.

Ethan Buchman. *Tendermint : Byzantine fault tolerance in the age of blockchains*. PhD thesis, University of Guelph, 2016.

Ethan Buchman, Jae Kwon, and Zarko Milosevic. The latest gossip on bft consensus. *arXiv preprint arXiv :1807.04938*, 2018.

Conrad Burchert and Roger Wattenhofer. pichain : When a blockchain meets paxos. In *21st International Conference on Principles of Distributed Systems (OPODIS 2017)*, volume 95, page 2. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2018.

Vitalik Buterin et al. Ethereum white paper. *GitHub repository*, 1(22-23) :5–7, 2013.

Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

Adrián Carro, Raúl Toral, and Maxi San Miguel. The noisy voter model on complex networks. *Scientific reports*, 6(1) :24775, 2016.

Claudio Castellano, Santo Fortunato, and Vittorio Loreto. Statistical physics of social dynamics. *Reviews of Modern Physics*, 81(2) :591, 2009a.

Claudio Castellano, Miguel A Muñoz, and Romualdo Pastor-Satorras. Nonlinear q-voter model. *Physical Review E*, 80(4) :041129, 2009b.

Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186, 1999.

M. R. Cerioli, L. Faria, T. O. Ferreira, C. A. J. Martinhon, F. Protti, and B. Reed. Partition into cliques for cubic graphs : Planar case, complexity and approximation. *Discrete Applied Mathematics*, 156(12) :2270–2278, 2008. ISSN 0166-218X. doi:https ://-doi.org/10.1016/j.dam.2007.10.015.

Houwu Chen and Jiwu Shu. Sky : an opinion dynamics framework and model for consensus over p2p network. *power*, 13 :14, 2015.

James C. Corbett, Jeffrey Dean, Michael Epstein, Andrew Fikes, Christopher Frost, J. J. Furman, Sanjay Ghemawat, Andrey Gubarev, Christopher Heiser, Peter Hochschild, Wilson Hsieh, Sebastian Kanthak, Eugene Kogan, Hongyi Li, Alexander Lloyd, Sergey Melnik, David Mwaura, David Nagle, Sean Quinlan, Rajesh Rao, Lindsay Rolig, Yasushi Saito, Michal Szymaniak, Christopher Taylor, Ruth Wang, and Dale Woodford. Spanner : Google's globally distributed database. *ACM Trans. Comput. Syst.*, 31(3), August 2013. ISSN 0734-2071. doi:10.1145/2491245. URL `https://doi.org/10.1145/2491245`.

Evangelos Deirmentzoglou, Georgios Papakyriakopoulos, and Constantinos Patsakis. A survey on long-range attacks for proof of stake protocols. *IEEE access*, 7 :28712–28725, 2019.

Liping Deng, Huan Chen, Jing Zeng, and Liang-Jie Zhang. Research on cross-chain technology based on sidechain and hash-locking. In *International conference on edge computing*, pages 144–151. Springer, 2018.

Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In *Annual international cryptology conference*, pages 139–147. Springer, 1992.

Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. Consensus in the presence of partial synchrony. *Journal of the ACM (JACM)*, 35(2) :288–323, 1988.

Ittay Eyal. The miner's dilemma. In *2015 IEEE symposium on security and privacy*, pages 89–103. IEEE, 2015.

Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2) :374–382, 1985.

Paolo Frasca, Sophie Tarbouriech, and Luca Zaccarian. Hybrid models of opinion dynamics with opinion-dependent connectivity. *Automatica*, 100 :153–161, 2019.

Noah E Friedkin and Eugene C Johnsen. Social influence and opinions. *Journal of Mathematical Sociology*, 15(3-4) :193–206, 1990.

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The google file system. In *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, SOSP '03, pages 29–43, New York, NY, USA, 2003a. Association for Computing Machinery. ISBN 1581137575. doi:10.1145/945445.945450. URL `https://doi.org/10.1145/945445.945450`.

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The google file system. *SIGOPS Oper. Syst. Rev.*, 37(5) :29–43, October 2003b. ISSN 0163-5980. doi:10.1145/1165389.945450. URL `https://doi.org/10.1145/1165389.945450`.

Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand : Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th symposium on operating systems principles*, pages 51–68, 2017.

Seth Gilbert and Nancy Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News*, 33(2) :51–59, 2002.

Corrado Gini. *Variabilità e mutabilità : contributo allo studio delle distribuzioni e delle relazioni statistiche*. Tipogr. di P. Cuppini, 1912.

Sharon Goldberg, Jan Vcelak, Dimitrios Papadopoulos, and Leonid Reyzin. Verifiable random functions (vrfs), 2018.

Michel Grabisch and Agnieszka Rusinowska. A survey on nonstrategic models of opinion dynamics. *Games*, 11(4) :65, 2020.

Tomasz Gradowski and Andrzej Krawiecki. Pair approximation for the q-voter model with independence on multiplex networks. *Physical Review E*, 102(2) :022314, 2020.

Zhangshuang Guan, Zhiguo Wan, Yang Yang, Yan Zhou, and Butian Huang. Blockmaze : An efficient privacy-preserving account-model blockchain based on zk-snarks. *IEEE Transactions on Dependable and Secure Computing*, 19(3) :1446–1463, 2020.

Rachid Guedjali, Jean-Philippe Georges, and Sylvain Kubler. Validator's opinion dynamic in a practical byzantine fault tolerant network. *IFAC-PapersOnLine*, 59(1) :223–228, 2025.

Guy Golan Gueta, Ittai Abraham, Shelly Grossman, Dahlia Malkhi, Benny Pinkas, Michael Reiter, Dragos-Adrian Seredinschi, Orr Tamir, and Alin Tomescu. Sbft : A scalable and decentralized trust infrastructure. In *2019 49th Annual IEEE/IFIP international conference on dependable systems and networks (DSN)*, pages 568–580. IEEE, 2019.

Stuart Haber and W Scott Stornetta. How to time-stamp a digital document. In *Conference on the Theory and Application of Cryptography*, pages 437–455. Springer, 1990.

Theo Härder and Andreas Reuter. Principles of transaction-oriented database recovery. *ACM Computing Surveys*, 15(4) :287–317, 1983.

Muneeb Ul Hassan, Mubashir Husain Rehmani, and Jinjun Chen. Anomaly detection in blockchain networks : A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 25(1) : 289–318, 2022.

Rainer Hegselmann. Opinion dynamics and bounded confidence : models, analysis and simulation. *The Journal of Artificial Societies and Social Simulation*, 2015.

Rainer Hegselmann and Ulrich Krause. Opinion dynamics and bounded confidence : models, analysis and simulation. *Journal of Artificial Societies and Social Simulation*, 5(3), 2002.

Maurice Herlihy. Atomic cross-chain swaps. In *Proceedings of the 2018 ACM symposium on principles of distributed computing*, pages 245–254, 2018.

Arkadiusz Jedrzejewski and Katarzyna Sznajd-Weron. Statistical physics of opinion formation : Is it a spoof ? *Comptes Rendus Physique*, 20(4) :244–261, 2019.

Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography : principles and protocols*. Chapman and hall/CRC, 2007.

Sunny King and Scott Nadal. Ppcoin : Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August*, 19(1), 2012.

Eleftherios Kokoris Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford. Enhancing bitcoin security and performance with strong consistency via collective signing. In *25th usenix security symposium (usenix security 16)*, pages 279–296, 2016.

Yaël Kolasa, Thierry Bastogne, Jean-Philippe Georges, and Sylvain Kubler. Quality-by-design-engineered pbft consensus configuration for medical device development. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 5709–5713. IEEE, 2020.

Andrzej Krawiecki and Tomasz Gradowski. Q-voter model with independence on signed random graphs : Homogeneous approximations. *Physical Review E*, 109(1) :014302, 2024.

Avinash Lakshman and Prashant Malik. Cassandra : a decentralized structured storage system. *SIGOPS Oper. Syst. Rev.*, 44(2) :35–40, April 2010. ISSN 0163-5980. doi:10.1145/1773912.1773922. URL `https://doi.org/10.1145/1773912.1773922`.

Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM*, 21(7) :558–565, July 1978. ISSN 0001-0782. doi:10.1145/359545.359563. URL `https://doi.org/10.1145/359545.359563`.

Leslie Lamport. Paxos made simple. *ACM SIGACT News (Distributed Computing Column) 32, 4 (Whole Number 121, December 2001)*, pages 51–58, 2001.

Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3) :382–401, 1982.

Guilain Leduc, Sylvain Kubler, and Jean-Philippe Georges. Adapting the number of validator according the need in a bft chain with sabine. *IFAC-PapersOnLine*, 55(8) :64–70, 2022a. ISSN 2405-8963. doi:https ://doi.org/10.1016/j.ifacol.2022.08.011.

Guilain Leduc, Sylvain Kubler, and Jean-Philippe Georges. Sabine : Self-adaptive blockchain consensus. In *2022 9th International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 234–240. IEEE, 2022b.

Guilain Leduc, Sylvain Kubler, and Jean-Philippe Georges. A centre-based validator selection approach for a scalable bft blockchain. In *22nd IFAC World Congress*, 2023.

Chris Lesniewski-Laas and M. Frans Kaashoek. Whānau : A sybil-proof distributed hash table. In *7th USENIX Symposium on Networked Systems Design and Implementation (NSDI 10)*, San Jose, CA, April 2010. USENIX Association. URL `https://www.usenix.org/legacy/event/nsdi10/tech/full_papers/lesniewski-laas.pdf`.

Peilun Li, Guosai Wang, Xiaoqi Chen, Fan Long, and Wei Xu. Gosig : A scalable and high-performance byzantine consensus for consortium blockchains. In *Proceedings of the 11th ACM Symposium on Cloud Computing*, pages 223–237, 2020.

Lu Liu, Sicong Zhou, Huawei Huang, and Zibin Zheng. From technology to society : An overview of blockchain-based dao. *IEEE Open Journal of the Computer Society*, 2 :204–215, 2021.

Mengting Liu, F. Richard Yu, Yinglei Teng, Victor C. M. Leung, and Mei Song. Performance optimization for blockchain-enabled industrial internet of things (iiot) systems : A deep reinforcement learning approach. *IEEE Transactions on Industrial Informatics*, 15(6) :3559–3570, 2019a. doi:10.1109/TII.2019.2897805.

Mengting Liu, F. Richard Yu, Yinglei Teng, Victor C. M. Leung, and Mei Song. Performance Optimization for Blockchain-Enabled Industrial Internet of Things (IIoT) Systems : A Deep Reinforcement Learning Approach. *IEEE Transactions on Industrial Informatics*, 15(6) :3559–3570, June 2019b. ISSN 1941-0050. doi:10.1109/TII.2019.2897805. IEEE Transactions on Industrial Informatics.

Loi Luu, Viswesh Narayanan, Chaodong Zheng, Kunal Baweja, Seth Gilbert, and Prateek Saxena. A secure sharding protocol for open blockchains. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 17–30, 2016.

Hanyu Mao, Tiezheng Nie, Hao Sun, Derong Shen, and Ge Yu. A survey on cross-chain technology : Challenges, development, and prospect. *IEEE Access*, 2022.

Shirshak Raja Maskey, Shahriar Badsha, Shamik Sengupta, and Ibrahim Khalil. Alicia : Applied intelligence in blockchain based vanet : Accident validation as a case study. *Information Processing & Management*, 58(3) :102508, 2021.

María Luisa Menéndez, Julio Angel Pardo, Leandro Pardo, and María del C Pardo. The jensen-shannon divergence. *Journal of the Franklin Institute*, 334(2) :307–318, 1997.

Ralph C Merkle. A digital signature based on a conventional encryption function. In *Conference on the theory and application of cryptographic techniques*, pages 369–378. Springer, 1987.

Debasis Mohanty, Divya Anand, Hani Moaiteq Aljahdali, and Santos Gracia Villar. Blockchain interoperability : Towards a sustainable payment system. *Sustainability*, 14(2) :913, 2022.

Irinel-Constantin Morarescu and Antoine Girard. Opinion dynamics with decaying confidence : Application to community detection in graphs. *IEEE Transactions on Automatic Control*, 56 (8) :1862–1873, 2010.

Shakira Musa Baig, Muhammad Umar Javed, Ahmad Almogren, Nadeem Javaid, and Mohsin Jamil. A blockchain and stacked machine learning approach for malicious nodes' detection in internet of things. *Peer-to-Peer Networking and Applications*, 16(6) :2811–2832, 2023.

Satoshi Nakamoto. Bitcoin : A peer-to-peer electronic cash system. *Available at SSRN 3440802*, 2008.

Martin Nischwitz, Marko Esche, and Florian Tschorsch. Bernoulli meets pbft : Modeling bft protocols in the presence of dynamic failures. In *2021 16th Conference on Computer Science and Intelligence Systems (FedCSIS)*, pages 291–300. IEEE, 2021.

Hossein Noorazar. Recent advances in opinion propagation dynamics : A 2020 survey. *The European Physical Journal Plus*, 135(6) :521, 2020.

Diego Ongaro and John Ousterhout. In search of an understandable consensus algorithm. In *2014 USENIX annual technical conference (USENIX ATC 14)*, pages 305–319, 2014.

Wei Ou, Shiying Huang, Jingjing Zheng, Qionglu Zhang, Guang Zeng, and Wenbao Han. An overview on cross-chain : Mechanism, platforms, challenges and advances. *Computer Networks*, page 109378, 2022.

Yan Pang. A new consensus protocol for blockchain interoperability architecture. *IEEE Access*, 8 :153719–153730, 2020.

Antonio F Peralta, Adrián Carro, Maxi San Miguel, and Raúl Toral. Analytical and numerical study of the non-linear noisy voter model on complex networks. *Chaos : An Interdisciplinary Journal of Nonlinear Science*, 28(7), 2018.

Xiaodong Qi, Yin Yang, Zhao Zhang, Cheqing Jin, and Aoying Zhou. Linsbft : Linear-communication one-step bft protocol for public blockchains. *arXiv preprint arXiv :2007.07642*, 2020.

Ling Qian, Zhiguo Luo, Yujian Du, and Leitao Guo. Cloud computing : An overview. In *IEEE international conference on cloud computing*, pages 626–631. Springer, 2009.

Xiaofang Qiu, Zhi Qin, Wunan Wan, Jinquan Zhang, Jinliang Guo, Shibin Zhang, and Jinyue Xia. A dynamic reputation-based consensus mechanism for blockchain. *Computers, Materials & Continua*, 73(2), 2022.

Peter Robinson. Consensus for crosschain communications. *arXiv preprint arXiv :2004.09494*, 2020.

Ludger Rüschendorf. The wasserstein distance and approximation theorems. *Probability Theory and Related Fields*, 70(1) :117–129, 1985.

Roberto Saltini. Bigfoot : A robust optimal-latency bft blockchain consensus protocol with dynamic validator membership. *Computer Networks*, 204 :108632, 2022.

Fred B Schneider. Implementing fault-tolerant services using the state machine approach : A tutorial. *Acm Computing Surveys (CSUR)*, 22(4) :299–319, 1990.

Narges Shadab, Farzin Houshmand, and Mohsen Lesani. Cross-chain transactions. In *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–9. IEEE, 2020.

Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. The hadoop distributed file system. In *2010 IEEE 26th symposium on mass storage systems and technologies (MSST)*, pages 1–10. Ieee, 2010.

Matteo Signorini, Matteo Pontecorvi, Waël Kanoun, and Roberto Di Pietro. Bad : A blockchain anomaly detection solution. *IEEE Access*, 8 :173481–173490, 2020.

Demetri P Spanos, Reza Olfati-Saber, and Richard M Murray. Dynamic consensus on mobile networks. In *IFAC world congress*, pages 1–6, 2005.

William Stallings. *Cryptography and network security : principles and practice*. Pearson, Boston Munich, seventh edition edition, 2017. ISBN 978-0-13-444428-4.

Nick Szabo. Bit gold. *Recuperado de https ://nakamotoinstitute. org/bit-gold/TVer página*, pages 251–260, 2005.

Katarzyna Sznajd-Weron. Sznajd model and its applications. *arXiv preprint physics/0503239*, 2005.

Katarzyna Sznajd-Weron, M Tabiszewski, and André M Timpanaro. Phase transition in the sznajd model with independence. *Europhysics Letters*, 96(4) :48002, 2011.

Tim Van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7) :3797–3820, 2014.

Maarten Van Steen and Andrew S Tanenbaum. *Distributed systems*. Maarten van Steen Leiden, The Netherlands, 2017.

Vineeth S Varma, Irinel-Constantin Morarescu, Samson Lasaulce, and Samuel Martin. Opinion dynamics aware marketing strategies in duopolies. In *2017 IEEE 56th annual conference on decision and control (CDC)*, pages 3859–3864. IEEE, 2017.

Vineeth S Varma, Irinel-Constantin Morărescu, and Mehdi Ayouni. Analysis of opinion dynamics under binary exogenous and endogenous signals. *Nonlinear Analysis : Hybrid Systems*, 38 : 100910, 2020.

Pavel Vasin et al. Blackcoin's proof-of-stake protocol v2. *URL : https ://blackcoin. co/blackcoin-pos-protocol-v2-whitepaper. pdf*, 71 :25, 2014.

Cédric Villani and Cédric Villani. The wasserstein distances. *Optimal transport : old and new*, pages 93–111, 2009.

Longze Wang, Jing Wu, Rongfang Yuan, Delong Zhang, Jinxin Liu, Siyu Jiang, Yan Zhang, and Meicheng Li. Dynamic adaptive cross-chain trading mode for multi-microgrid joint operation. *Sensors*, 20(21) :6096, 2020.

Gavin Wood et al. Ethereum : A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014) :1–32, 2014.

Maofan Yin, Dahlia Malkhi, Michael K Reiter, Guy Golan Gueta, and Ittai Abraham. Hotstuff : Bft consensus with linearity and responsiveness. In *Proceedings of the 2019 ACM symposium on principles of distributed computing*, pages 347–356, 2019.

Haifeng Yu, Michael Kaminsky, Phillip B Gibbons, and Abraham Flaxman. Sybilguard : defending against sybil attacks via social networks. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 267–278, 2006.

Jiangshan Yu, David Kozhaya, Jeremie Decouchant, and Paulo Esteves-Verissimo. Repucoin : Your reputation is your power. *IEEE Transactions on Computers*, 68(8) :1225–1237, 2019.

Zhihao Zeng, Xiaoning Zhang, and Zixiang Xia. Intelligent blockchain-based secure routing for multidomain sdn-enabled iot networks. *Wireless Communications and Mobile Computing*, 2022 (1) :5693962, 2022.

Junlin Zhang. Wsn network node malicious intrusion detection method based on reputation score. *Journal of Cyber Security and Mobility*, 12(1) :55–76, 2025.

Yushu Zhang, Jiajia Jiang, Xuewen Dong, Liangmin Wang, and Yong Xiang. Bedcv : Blockchain-enabled decentralized consistency verification for cross-chain calculation. *IEEE Transactions on Cloud Computing*, pages 1–12, 2022. doi:10.1109/TCC.2022.3196937.

Qianwei Zhuang, Yuan Liu, Lisi Chen, and Zhengpeng Ai. Proof of reputation : A reputation-based consensus protocol for blockchain based systems. In *Proceedings of the 1st International Electronics Communication Conference*, pages 131–138, 2019.

Résumé

Cette thèse examine les défis fondamentaux liés à la sélection et à la surveillance des validateurs dans les protocoles de consensus tolérants aux fautes byzantines (BFT), qui constituent le fondement de la sécurité et de la performance des systèmes blockchain. Les approches existantes présentent des limites en matière d'adaptabilité dynamique et de détection proactive des comportements malveillants. Pour surmonter ces contraintes, nous proposons un cadre méthodologique combinant modélisation probabiliste, théorie de l'information et apprentissage automatique. Ce cadre introduit un modèle de dynamique d'opinion fondé sur une approximation de champ moyen permettant de prédire l'évolution des comportements des validateurs et de détecter précocement les anomalies, un protocole de sélection centrée dynamique optimisant le compromis entre performance réseau et décentralisation, ainsi que de nouvelles métriques de similarité comportementale basées sur la divergence de Jensen-Shannon, le coefficient de Bhattacharyya et la distance de Wasserstein. Les expérimentations sur des réseaux simulés démontrent l'efficacité de ces approches pour identifier des formes variées d'attaques, allant des déviations transitoires aux manipulations persistantes, tout en maintenant les garanties de sécurité propres au consensus BFT. Les résultats obtenus contribuent à renforcer la robustesse et l'adaptabilité des protocoles de consensus, en offrant des outils analytiques et méthodologiques pour la conception de blockchains plus fiables et performantes.

Mots-clés: Blockchain, Tolérance aux fautes byzantine, Sécurité, Consensus, Dynamique d'opinion

Abstract

This thesis investigates the fundamental challenges associated with validator selection and monitoring in Byzantine Fault-Tolerant (BFT) consensus protocols, which underpin the security and performance of blockchain systems. Existing approaches exhibit significant limitations in terms of dynamic adaptability and proactive detection of malicious behaviors. To address these shortcomings, we propose an methodological framework that combines probabilistic modeling, information theory, and machine learning. The framework introduces an opinion dynamics model based on a mean-field approximation to predict validator behavior evolution and enable early anomaly detection. It also defines a dynamically centered selection protocol that optimizes the trade-off between network performance and decentralization, together with novel behavioral similarity metrics grounded in Jensen–Shannon divergence, Bhattacharyya coefficient, and Wasserstein distance. Experiments conducted on simulated networks demonstrate the effectiveness of these approaches in identifying diverse forms of attacks—ranging from transient deviations to persistent manipulations—while preserving the security guarantees intrinsic to BFT consensus. The results contribute to enhancing the robustness and adaptability of consensus protocols, providing analytical and methodological tools for the design of more reliable and efficient blockchain systems.

Keywords: Blockchain, Byzantine Fault Tolerance, Security, Consensus, Opinion Dynamics.