修士論文

ブロックチェーンにおける ストレージを考慮した DHT 負荷分散クラスタリング

17892511　金子勇大

指導教員　朝香 卓也

2019 年 1 月

首都大学東京大学院 システムデザイン研究科
システムデザイン専攻
経営システムデザイン学域

目次

図目次

第 1 章

序論

1.1 研究背景

近年，ハッシュ関数等の暗号技術やピアツーピア (P2P) ネットワーク技術の発展に伴い，P2P ネットワーク上で暗号化された通貨として取引を行う P2P 電子マネーシステムが普及してきている．そして，金融の IT 化を推し進める Fintech への関心の高まりや Internet of Things (IoT) への技術的な応用に対する期待から，その基幹技術であるブロックチェーンに関する研究が注目されるようになった [1] -[3]．一般に，最も代表的な P2P 電子マネーシステムとしてビットコインと呼ばれるシステムが存在するが，これは 2009 年にサトシナカモトと呼ばれる人物が発表した論文 [4] を元に作られたもので，この論文がブロックチェーンの発端であり，これが P2P 電子マネーシステムの根幹となっている．P2P 電子マネーシステムには，Paypal やポイントカードのような電子マネーと呼ばれるシステムとは大きな特徴の違いがあり，それは中央サーバや金融機関等の管理者が存在しないという性質をもつことである．それまでのインターネット上の商取引は，電子取引を処理するための信用できる第三者機関としての金融機関が必要不可欠で，第三者機関を通さずに通信チャンネル経由での支払いを可能とするメカニズムは存在していなかった．ビットコインでは，P2P 分散型タイム・サーバやハッシュ関数，そして Proof of Work と呼ばれる承認アルゴリズム等を組み合わせたブロックチェーンを用いることによって，非構造のピュア P2P ネットワーク上において時系列取引のコンピュータ的証明を作成することで，改ざんや二重使用に対する問題の解決策を提案している．そして，これらの仕組みによって信用できる二者間での第三者機関を必要としない直接取引が可能となっている．また，ブロックチェーンは第三者の承認を必要とせず，その価値を保証できる性質から，P2P マネーとしての利用だけではなく，医療，スマートシティ，IoT 等の多くの分野での活用が期待されている [5] -[8]．

P2P ネットワークとは，サーバとクライアント両方の振る舞いが可能とされる端末（ピア）から構成されたネットワークシステムである [9] -[11]．ネットワークに参加しているピアは

オーバレイという論理ネットワークを介して他のピアとの通信を行う．システム運用の観点から，従来のクライアント・サーバ型のような集中管理方式では，システムの管理団体の経営が立ち行かなることや，システム障害が単一障害点になることでシステムが停止する懸念がある．一方，P2P ネットワークでは集中して通信を行うサーバを持たないため，非中央集権型の管理が可能で，故障の影響もシステム全体に波及しずらいため，システム運用が継続しやすいという利点がある．従って，第三者機関のいらない非中央集権型である管理や，また，頑健性のあるシステム運用を行うために，P2P 電子マネーシステムでは P2P ネットワークが適用されている．また，P2P ネットワークにおいて主流の用途である「各ピアに分散的にデータを配置する環境で，あるピアが P2P ネットワークからデータの探索を行う」ための分散ファイル共有システムではなく，P2P 電子マネーシステムでは，「更新されていく共有データを全ピアが保持し，ピア同士で改ざん等の不正を検知し合う」という不正検知システムとして P2P ネットワークが機能している．

1.2　研究目的

ビットコインでは，ブロックチェーンにおけるブロックにアルゴリズム上の制約条件を設けているため，現在，ネットワーク全体での取引 (トランザクション) の処理能力は約 7TPS (Transactions Per Second) となっている．ブロックチェーンの制約条件とは，主にブロックサイズとブロック生成間隔のことである．ブロックとはブロックチェーンにおける一定のまとめられたトランザクションデータの集まりのことであり，ブロックサイズはブロックのデータサイズ及びその上限のことを指している．現在，ビットコインではブロックサイズが 1MB となっており，このブロックサイズの上限を解放することでトランザクション処理能力を向上することができる．また，ブロック生成間隔とは，1 つ前のブロックのが生成されてから新しいブロックを生成するまでの間隔の目安であり，現在ビットコインは 10 分が基準となっている．この間隔を短くし，時間あたりのブロックの生成個数を増やすことでトランザクション処理能力を向上することができる．しかしながら，これらの制約条件の緩和は，処理能力は向上する代わりに，全ての参加ノードへのストレージ負荷やネットワーク負荷が増大することにより，個人単位の参加ノードが離脱してしまう．そして，大規模マイナーやウォレットサーバ等の法人単位の参加ノードのみが生存し，ネットワーク維持のためプロセスが集中化してしまい, 中央集権化を招く恐れがある．また，クレジットカード等の一般的なトランザクション処理システムに比べ，ビットコインの処理能力が著しく低いことからトランザクション処理能力に関するスケーラビリティの議論は長年行われており，これらの制約条件を考慮したり，違うネットワークレイヤでトランザクションを処理したりするような様々なスケーラビリティに関する研究とプロトコルの提案が行われている．しかしながら，従来研究ではブロックデータの伝播遅延に着目した研究は多く存在しているが，ネットワーク負荷やデータ送信の効率性に着目した

研究は少ない．そのため，P2P ネットワークのアーキテクチャに着目した提案手法はほとんど存在しておらず，ノード数に関するスケーラビリティはあまり考慮されていない．

そこで，本稿では特定ピアに負荷が集中せずに大規模なコンテンツ探索を実現する DHT (Distributed Hash Table) のうち，代表的なアルゴリズムの 1 つである Kademlia をブロックチェーンに応用し，効率的に P2P ネットワーク内へのブロックチェーンのデータのブロードキャスティングを行う手法と，ブロックチェーンのデータを DHT のネットワーク内で分散配備させる手法を組み合わせることで，非中央集権性を保ったまま，ストレージとネットワーク負荷を分散させる手法の提案を行う．

本稿では，2 章でビットコインにおけるブロックチェーンの技術的な概要やトランザクション処理能力に関するスケーラビリティの従来研究，3 章で P2P ネットワークの概要，DHT を利用したブロードキャスティングの従来研究について説明する．4 章では提案手法について詳細な説明を行い，5 章では計算機シミュレーションによって提案手法の有効性についての評価を行う．最後に，6 章ではまとめと今後の課題について述べる．

第 2 章

ブロックチェーン技術

本稿では，2.1 節で記述するビットコインの仕様を，従来手法としている．従来手法であるビットコインでは，非構造のピュア型 P2P ネットワーク上で稼働するソフトウェアであり，執筆時における稼働ノード数は約 9000 ノードあり，ブロック生成感覚の基準は 10 分， ブロックサイズ上限は 1MB となっている．2.2 節では，スケーラビリティの改善プロトコルやそれらに付随する問題点について説明する．本稿における提案手法ではブロックサイズの上限の引き上げやブロック生成間隔の短縮を前提としており，これらに付随するブロックデータの伝播遅延やフォークの発生頻度の増加等の問題点を考慮した従来研究について説明する．また，提案手法ではノードのストレージに着目しているため，ノードのストレージと非中央集権性に着目した従来研究について説明する．

2.1　ビットコイン及びブロックチェーン概要

文献 [12] や文献 [13]，図 2.1 に示されているビットコインとその基盤技術の概要について述べる．また，各ノードは他ノードから接続リクエストがあった時にそれを拒否しないため，近接ノード数は 2.1 のようになっていると推測されている [12]．ビットコインにおいて，各ユーザーは管理する秘密キーから生成されるウォレット及びビットコインアドレスを保持し，これを自身の銀行口座のように扱うことができる．しかしながら，そこに第三者の介入なく，P2P ネットワークの参加ノードの独立したプロセスによって送金や着金が承認される．具体的には，送金ノードは特定の他ノードのアドレスに対してトランザクションと呼ばれる取引記録を生成し，それがネットワーク全体に伝播される．そしてマイニングと呼ばれる承認作業を経てトランザクションの集合体であるブロックを生成し，そのブロックがネットワーク全体によって検証されることによって送金が成り立っている．執筆時では，ビットコインを扱う取引所によって為替取引が可能であり，また，通信販売における決済やその他支払いも可能となっている．

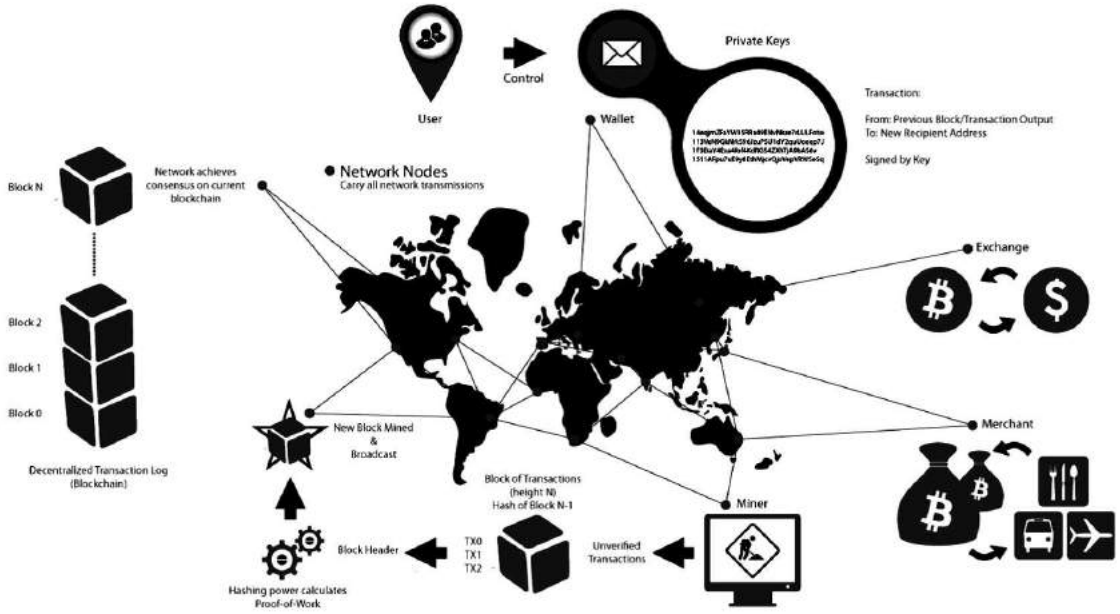


図 2.1. ビットコインネットワークの概要図 [12].

表 2.1. ビットコインネットワークの主な仕様.

項目	仕様
ネットワーク	非構造型ピュア P2P ネットワーク
執筆時における稼働ノード数	9840 [14]
マイニングに用いるハッシュ関数	SHA-256
ノードの最低接続数	8 [23]
ノードの平均接続数	32 [23]
ブロック生成間隔基準	10 分
ブロックサイズ上限	1 MB

本節では，P2P 電子マネーシステムを運用するためのビットコインと呼ばれる最も代表的な P2P ソフトウェアの仕様について述べる．また，執筆時におけるビットコインの主な仕様を表 2.1 に示す．

本節では，ブロックチェーンについての重要な概念とともに，ビットコインにおける実際の各ノードにおける独立したプロセスについて説明し，次に，マイニングとフォークについて説明する．

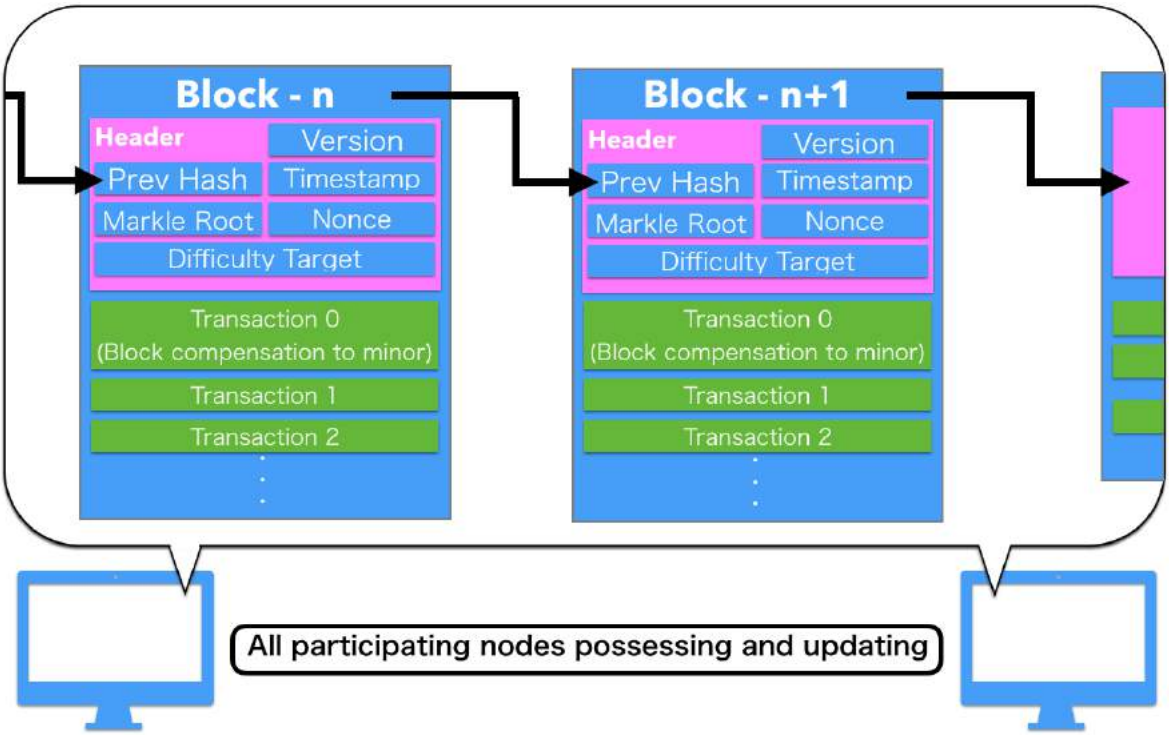


図 2.2. ブロックチェーンのブロック構成.

2.1.1　トランザクション伝播プロセス

はじめに，トランザクションの生成と伝播のプロセスについて説明する．従来手法では，参加ノード間の送金を行う際，送金ノードがトランザクションを生成し，それを非構造型 pureP2P ネットワーク全体に伝播させる．この時点では，トランザクションは無効であり，送金は達成されていない．ここでは，トランザクションを生成し参加ノード全体へそのトランザクションを伝播させるプロセスのことをトランザクション伝播プロセスと呼ぶことにする．トランザクション伝播プロセスの手順を以下に示す．

(step1)　送金ノードがトランザクションを生成し，近接ノードへ送信
(step2)　受信ノードは自身のブロックチェーンを参照し，不正がなく有効なトランザクションであることを検証
(step3)　そのノードは自身の近接ノードへ検証済みのトランザクションを送信
(step4)　(step2)，(step3) を繰り返し，ネットワーク全体へトランザクションを伝播

2.1.2 ブロック伝播プロセス

次に，ブロックの生成と伝播についてのプロセスについて説明する．伝播されたトランザクションはマイニングノードと呼ばれるトランザクションの承認処理を行うノードによって複数のトランザクションまとめられ，ブロックが生成される．ここで，ブロックの構成を図 2.2 に示す．ブロックのヘッダーには，P2P ソフトウェアであるビットコインのバージョンを示す Version，前のブロック全体のハッシュ値を示す Prev Hash，ネットワークにおける時系列の示すための Timestamp，トランザクションを格納し整理するためのインデックステーブルである Markle Root，マイニングにおいて用いられる Nonce と，Difficulty Target によって構成されている．図 2.2 のように，P2P ソフトウェアの初期稼働時に生成される 0 番目のブロックから，1 つ前のブロック全体のハッシュ値を参照し次のブロックへ鎖のように連なるように積み上げられている構造であること，また，参加ノード全員で同じ積み上げられたブロックチェーン共有し更新することが，ブロックチェーンの重要な考え方の一つとなっている．

過去の全てのブロックチェーンを持ったノードによって，不正なトランザクションがないか検証された後，トランザクションは集積されブロックが生成される．そのブロックがブロックチェーンへの付加される時には，マイニングと呼ばれる承認作業を行わなければならず，マイニングが終わると，新規ブロックハッシュ値が伝播され各ノードで独立的に検証が行われる．

生成されたブロックを伝播するプロセスを以下に示す．

(step1) マイニングノードがマイニングを行い，トランザクションを集積しブロックを生成
(step2) マイニングノードから生成された新しいブロックを近接ノードへ送信
(step3) 受信ノードは自身のもつブロックチェーンを参照に，不正なトランザクションがないかどうかの検証
(step4) そのノードは，その検証済みのブロックを自身の近接ノードへ送信し，また，自身のブロックチェーンへ付加
(step5) (step3)，(step4) を繰り返し，ネットワーク全体へブロックを伝播

また，追加される新規ブロックには，ブロックを生成したマイナーに報酬として供給される送金元アドレスが存在せず，着金先のみの指定されたトランザクションが組み込まれており，これが，ビットコインネットワークにおける紙幣発行の役割を担っている．

2.1.3 マイニング

マイニングとは，条件を満たすまでナンスの値 (図 2.2 における Nonce) を求めハッシュ計算を繰り返すことであり，実質的に改ざんを防いでいる分散合意についての重要な概念であ

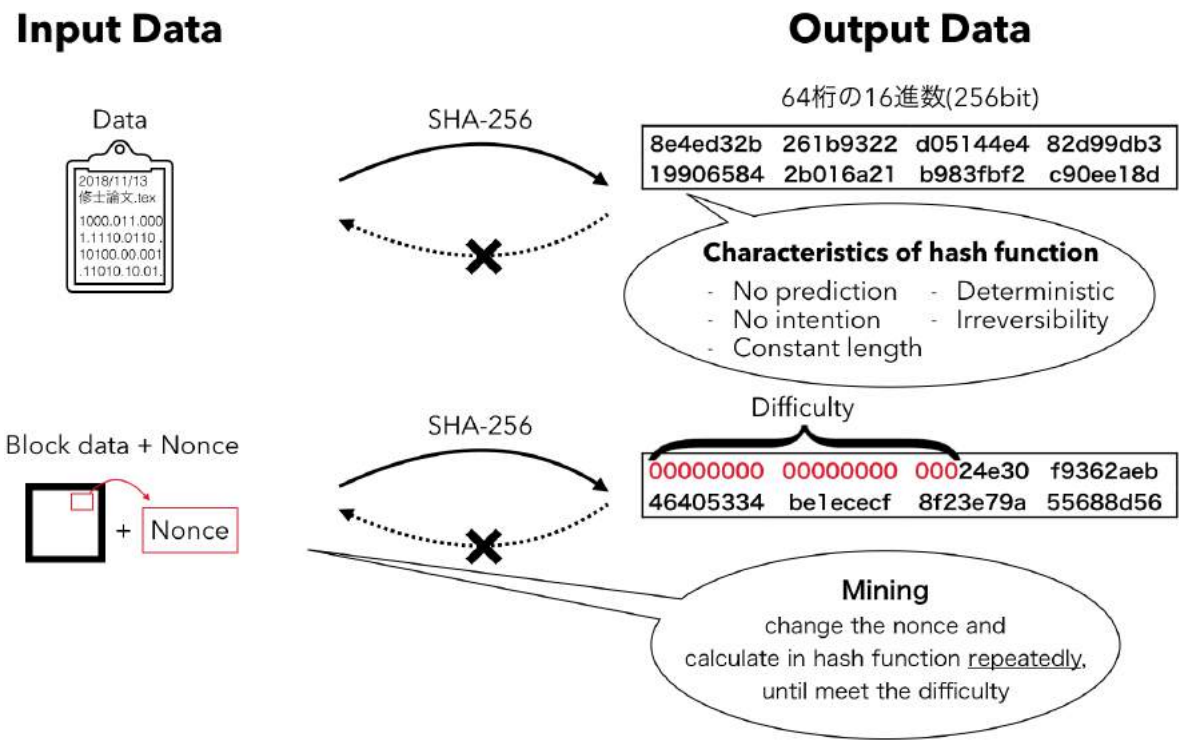


図 2.3. ハッシュ関数とマイニングの概要図.

る．ここで，ハッシュ関数とマイニングの概要を図 2.3 に示す．ビットコインのマイニングでは，ハッシュ関数 SHA-256 を用いる．このハッシュ関数は出力結果を前もって予測できず，特定のハッシュ値を作り出すようなパターンを作り出すようなことも決めれず，入力に対する出力は常に一定であるという性質を持つ．また，出力されるハッシュ値の値は常に一定長であり，ハッシュ値から入力データへ戻せないという不可逆性を持つ．ナンスとは，ハッシュ計算を行う前に，任意に設定する 32 bit のブロックのヘッダーフィールドの値である．図 2.2 で全てのトランザクションデータ他の全てのヘッダのフィールドが埋められたときに，ナンスの初期値を 0 としてマイニングのプロセスを開始することができる．SHA-256 のようなハッシュ関数では，意図的なハッシュ値を作り出すことができないため，「ナンスを変え，それを含めたブロックデータ全体を入力値としてハッシュ関数によって出力値を算出する」という作業を繰り返し行い，出力値が Difficulty の条件を満たすまで試行を繰り返さなければならない．ここでは，出力値が 000001 以下というような 0 の桁数を表すための Difficulty という条件値が設けられている．この Difficulty は 10 分を目安で計算が終わるように，一定の間隔で自動調整される．また，Difficulty が増えるほど，ハッシュ計算の試行回数が増えるため，ハッシュレート (一秒間でのハッシュ計算の試行回数) が高い GPU や ASIC を使用している計算機が有利とされている．

そして，入力値に対する出力値は常に一定なので，新規ブロックが伝播された各ノードは，

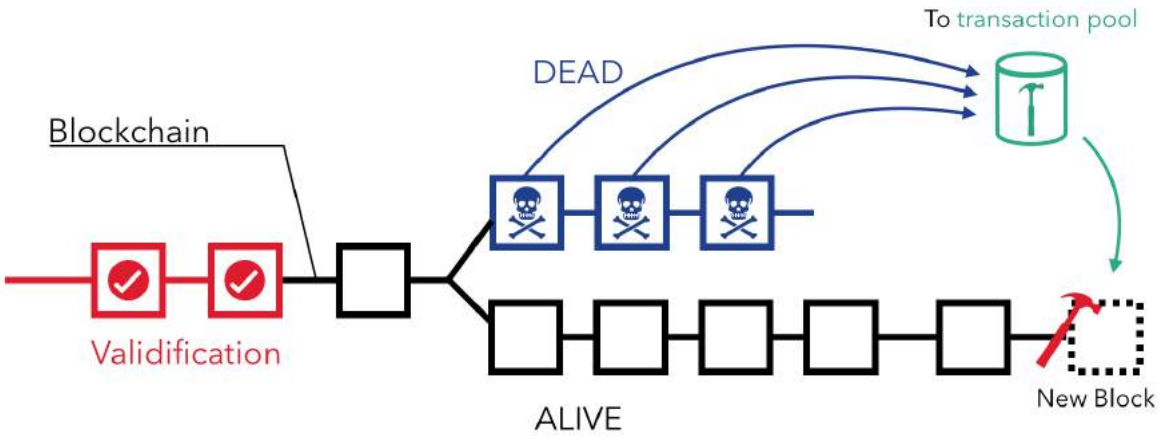


図 2.4. ブロックチェーンにおけるフォークの概念図.

Difficulty の条件を満たすかどうか容易に検証を行うことができる．これらの承認作業のアルゴリズムが，Proof of Work(PoW) と呼ばれている．

ブロックを作成できるのは，最も早く Difficulty の条件値を満たすナンスを見つけネットワーク全体に伝播できたノードのみであり，ビットコインのブロックチェーンでは，10 分間隔を目安にその競争を何度も繰り返している．実際にトランザクションデータの改ざんを行おうとして，特定のトランザクションに改変を加えた場合，そのトランザクションが含まれているブロックの入力値が変化してしまうため，ハッシュ関数にかけた出力値は全く違うものになってしまう．さらにそのブロックのハッシュ値を格納する後続ブロックのヘッダーフィールドにも変更が加えられてしまうことが連鎖的に起こるため，結果として改ざんを行ったブロックから現在のブロックまで，全てのブロック情報が書き換えられ，各々のブロックにおけるブロックハッシュ値は全て Difficulty 条件値を満たさなくなる．再度，改ざんしたブロックから現在までのブロックまでマイニングを繰り返し，条件を満たせば，改ざんは可能であるが，そのためには，現在までのブロックのマイニングの計算作業を再度行うことに加え，現在マイニングされているブロック作成者にも選ばれる必要があるため，ネットワーク全体の計算リソースのうち，大半を占めていなければ，改ざんは達成できない．ビットコインが分権化を考慮した設計になっている以上，計算機リソースを占領することは実質不可能であり，これが，改ざん等の不正が発生しない根拠となっている．

2.1.4　フォーク

ブロックチェーンでは，ヘッダーに格納する前ブロックのブロックハッシュ値は 1 つとなっているため，ネットワークに認められるチェーンはただ一つであり，これをメインチェーンと呼ぶ．マイニングではその都度，最も早く PoW を解いたノードがブロック生成者となる権利

を得ることができるが，ネットワーク全体に伝播される前に，条件を満たした他のブロック生成者がブロックを伝播する状況も考えられる．こうしたブロックチェーンの分岐はフォークと呼ばれている．ここでフォークの概要を図 2.4 に示す．フォークは想定された状況でありブロックチェーンでは，時間経過とともに一つに収束するアルゴリズムが組まれている．そのアルゴリズムでは，原則として「最も長いチェーンが真」とされている．長いとは，詳細には全てのブロックの Difficulty の集積値が大きいことを指しているが，後続するブロックの個数が多いこととほぼ同義の意味を持つ．メインチェーンに選ばれなかったチェーンはトランザクションデータに解体され，マイナーがマイニングするトランザクションを選出するためメモリにプールされる．この時，少数派のチェーンの後続ブロックを生成して後にメインチェーンとして認証されずに解体されることを考慮すると，マイナーにとって多数派のチェーンに後続するブロックの生成を行うインセンティブが働く．そしてネットワークの各ノードは一時的に同じブロック高のブロックを保持し，後続されるブロックが多いチェーンを選択することで，メインチェーン以外のブロックは解体され 1 つに収束される設計となっている．

これに加え，マイニングの不正が行われる場合のコストも考慮すると，あるブロックにおける安全性は時間経過とともに高まる．しかし，これは確率的に不正が行われる可能性が極めて小さくなることを指しており，ブロックチェーンでは決済におけるファイナリティは達成されない．そのため取引所などビットコインを通貨として扱う団体は，現在のブロックからいくつ前のブロックを安全なブロックとみなすどうかを任意に設定している．

2.2　ブロックチェーンの技術的課題

1 章で述べたように，ブロックチェーンではそのアルゴリズムの制約条件からトランザクション処理能力のスケーラビリティが問題となっている．従来研究では，セキュリティや分散性等のブロックチェーン特有の性質を保ちながら，トランザクション処理能力を向上させるための研究が多く存在する [15] - [20]．これらは，ビットコインのブロックチェーンのアーキテクチャから細部の修正を施したプロトコルをはじめ，分散合意プロトコルを変更しそれに伴い適宜修正を加えたもの，ブロックを生成せずトランザクションをチェーンとして繋ぐプロトコル，ブロックチェーンの取引処理を別のレイヤで行うプロトコル，など多種多様な提案がなされている．しかしながら，これらをセキュリティや分散性等のブロックチェーンに係る様々な要素を加味した上で評価できる指標は定まっておらず，現在において事実として認められるのは，ビットコインの培ってきた経験則的な安全性のみである．そのため，多くのブロックチェーンにおけるネットワークやセキュリティ等の様々な観点による分析や解析が求められており，そうした研究も進められている [23] - [28]．また，それらの研究の中には，ブロックチェーンプロトコルの技術的な仕様の新たに見つかった脆弱性を指摘するものも存在する [28] - [30]．

本節では，トランザクション処理能力のスケーラビリティを改善するプロトコル提案の従来研究について説明する．文献 [15] では，フォークされたブロックを条件付きでメインチェーンとみなすことでマイニングの効率性やブロックの伝播遅延を少なくなる手法を提案している．また，文献 [16] では，ブロックチェーンのネットワークを分断するエクリプス攻撃 [29] を防ぎ，ブロック伝播遅延を少なくなるようなルーティングを実現させる手法を提案している．そして文献 [17] では，既存のブロックチェーンの構造とは違い，有向非巡回グラフやマルコフ連鎖を応用することでトランザクション単体をチェーンとしてつなぐ手法を提案しており，自立分散で稼働する膨大センサー等を利用したマイクロペイメントを想定した技術として注目されている．以降は，オフチェーンと呼ばれるブロックチェーン外の取引を行うプロトコルについて説明し，次に本稿に関連のあるビックブロックについて説明し，最後に本稿に関連のあるノードのストレージサイズと非中央集権性について説明する．

2.2.1 オフチェーンプロトコル

ブロックチェーンのアルゴリズム外でトランザクション処理を行う手法はオフチェーンプロトコルと呼ばれ，ブロックチェーンのアルゴリズムに制限されることが少なく，自由度が高いため，設計次第ではスケーラビリティに多大な影響を与える分野として注目されている．例えば，サイドチェーン [19] と呼ばれるプロトコルでは，メインチェーンとは別に副次的なチェーンを作り特定のブロックにおけるブロック情報のハッシュ値をメインチェーンに記載することで，メインチェーンの耐改ざん性等のメリットを享受できるような仕組みとなっている．このようにオフチェーンプロトコルでは，ネットワーク層より上層でアプリケーションが機能しているように，ブロックチェーンのメインチェーンを下層のレイヤとして扱うことで，上層で様々なプロトコルを動かそうとする傾向がある．そのため，オフチェーンプロトコルはセカンドレイヤプロトコルとも呼ばれている．

Lightning Network

セカンドレイヤでのトランザクション処理を行う手法のうち最も代表的なものに Lightning Network[18] というものがある．Lightning Network では，ブロックチェーンの外で取引を行うために当事者ノードの 2 者間での複数による署名 (マルチシグ) を行い，セカンドレイヤにマイクロペイメントチャネルを設ける．マイクロペイメントチャネルを開設するトランザクションと閉鎖するトランザクションのみを，メインチェーンに記載することで，ネットワーク全体へブロードキャストする必要がなくなるため，手数料不要の取引が可能となる．

Lightning Nework とは，分散性や安全性を考慮した上で，マイクロペイメント・チャネルを用いて第三者経由でオフチェーンでのトランザクション取引を行うプロトコルのことで，第三者が資金を持ち逃げされないようにするため，Hashed Time-Lock Contatct(HTLC) 取引

という仕組みを導入している．また，Lightning Network のルーティングプロトコルとして flare[20] を提案している．Lightning Network の提案はスケーラビリティ問題が懸念され始めた頃から存在したが，これを導入するためにはトランザクション展性，相対的なタイムロックトランザクションの作成という，プロトコル上の課題が存在したが，近年におけるビットコインプロトコルの拡張により，トランザクション展性を解決する Segwit，相対タイムロックを刷る新しい OP コード (OP CSV) が導入されたため，課題が解決され，再び注目を浴びている．

2.2.2　ビックブロック

一方，ブロックチェーン自体のアルゴリズムに修正を加えることで，スケーラビリティを解決しようとするものは，オンチェーンプロトコルにおけるソリューションである．オンチェーンプロトコルを検討する際，一般に「ブロックチェーンのトリレンマ」と呼ばれる状態を特に考慮する必要がある．これは，ブロックチェーンにおいて非中央集権性，スケーラビリティ，安全性の 3 つの特性は同時に達成できないことを指す．例えば，ビットコインでは，執筆当時で 7TPS しか持っていないことから，スケーラビリティが低い代わりに残り二つに重きを置いていることが分かる．TPS とは，Transactions Per Second の略称であり，「1 秒間あたりにどれだけのトランザクションの処理が可能か」を表す単位であり，クレジット決済システム等の金融システムにおける重要な指標である．ブロックサイズ上限の制限の緩和は最も代表的なスケーラビリティ改善手法であり，ビックブロックと呼ばれている．ビックブロックでは，1 つのブロックのトランザクションを追加することでスケーラビリティを向上させようとすることを提案している．文献 [27] では，ブロックサイズが増大した際のネットワーク分析を行っており，トランザクションをより多くブロックに加えることでスケーラビリティを向上させることができるが，ブロック伝播遅延を引き起こし，不正の成功確率が増加する懸念がなされている．また，ネットワークのノードの運用するためのハードウェア要件が厳しくなることで，要件を満たせないノードがネットワークを脱退し，非中央集権性が保たれなくなる懸念がなされている．このことから，ビックブロックはスケーラビリティに重きを置くことで，安全性，非中央集権性が低くなる可能性が指摘されている．ブロック伝播遅延については，文献 [23] によるとブロック伝播遅延時間はブロックデータ転送時間とブロックの検証時間の合計値で示されており，ブロックサイズが 20kB 以下のときにはラウンドトリップ遅延が発生し，20kB 以上のときには遅延時間はブロックサイズに比例してで 80ms/kB の割合で増加することが報告されている．

Bitcoin-NG

文献 [21] では，Bitcoin-NG と呼ばれるプロトコルを提案している．これはブロックサイズ上限を引き上げた際の，ブロック伝播の遅延を解消しマイナー間の競争をフェアにすることを

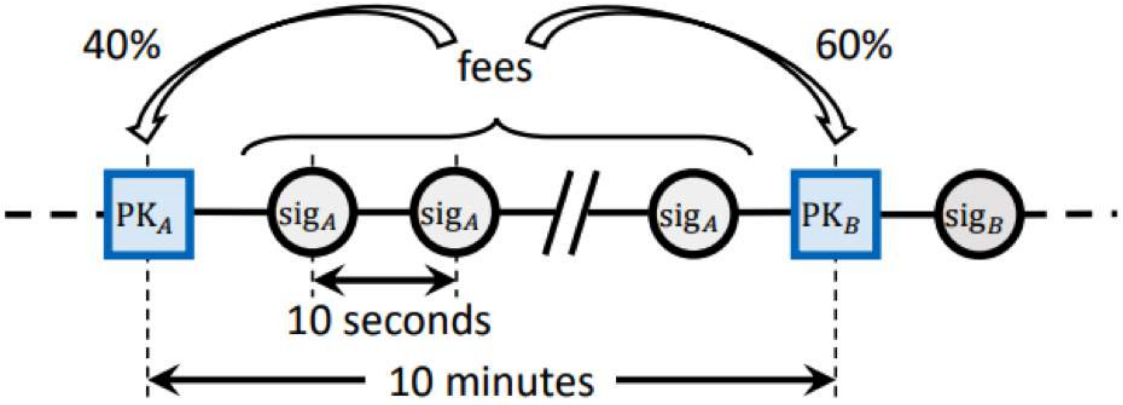


図 2.5. Bitcoin-NG におけるチェーン構造の概要図 [21].

目的としている．現在のマイニングは，マイニングプールと呼ばれる手法が主に用いられており，これは運営団体がマイナー集団を形成し，集団でマイニングを行い得られる報酬を山分け形式参加マイナーに分配する手法である．こうすることで，報酬の得られる機会を均一化できるため，マイニングプール参加することでキャッシュフローのリスクを抑えるメリットがある．しかしながら，大規模なマイニングプールを形成することは非中央集権性を損ね，運営団体が行う不正の成功確率が増加する懸念がなされている．この場合の不正は，対立するプールに非積極的マイナーを潜入させる Block withholding や見つけたブロックを直ちに伝送しないことで自信を有利にさせる Selfish Mining 等，様々な攻撃方法の可能性が指摘されている [28]．また，Selfish Mining に関しては，ブロック伝播遅延が大きいほど，不正の成功確率が増えることも報告されている [28].

Bitcoin-NG では，マイニングの公平性，マイニングの効率や有用性，フォークの収束時間の減少を達成するために，ブロック遅延の構成要素であるブロックデータ転送時間やブロックの検証時間がほとんど必要ないプロトコルを提案している．手法の概要を図 2.5 に示す．図中の PKa は，あるマイナー A の生成したブロックにマイナー A の秘密鍵から生成した $PublicKey$ をブロックヘッダーに付加することを示している．生成したブロックはキーブロックと呼ばれ，ブロックチェーンの新規発行分のトランザクション以外のトランザクションの付加されていないブロックである．生成者 A はマイクロブロックと呼ばれる，あらかじめ定められたサイズの生成者 A の署名が付加されているブロックであり，通常のブロック同様，前ブロックのハッシュ値が後続ブロックに格納されている．キーブロックの生成者は 10 秒ごとにマイクロブロックを生成する権利を与えられ，トランザクション手数料は後続のキーブロック生成者と 4 : 6 の割合で分配されることにより，攻撃に対する安全性を確保している．Bitcoin-NG では，トランザクションの入っていないキーブロックのみをマイニングの対象にすることで，マイニングの公平性や効率，有用性を向上させており，キーブロックのデータサ

イズが小さくて済むことによって，ブロック遅延の問題を解決している．

2.2.3　ノードの参加要件と非中央集権性

ビックブロックではブロックサイズが増加し，ノードのストレージやネットワーク負荷を圧迫し，ハードウェア要件を満たさなくなったノードが脱退することで，ネットワーク全体でのノード数が減少することが懸念されている．そして，ノード数が減少し計算機等の資本を持つ団体が有利となってしまうことで，非中央集権性を低下する可能性が指摘されている．

Proof of Activity

文献 [22] では，Proof of Activity (PoA) と呼ばれるプロトコルを提案している．PoA では，ビットコインにおけるマイニング報酬が仕様としてデフレーションを起こすことを考慮している．マイニングが半減期を繰り返し，ブロック報酬が存在しなくなったとき，マイナーの利益になるのはブロック内のトランザクション手数料のみである．その状況では，マイナーはどんなに少ない手数料でもブロックに追加することになり，ネットワークにトランザクションが認証されるまでの時間が減少する．その結果，トランザクションを生成するノードは手数料を上げる必要がなくなるため，手数料は減少傾向になる．そして，マイナーは減少傾向の手数料のみで採算を立てる必要があるため，マイニングから脱退するマイナーが増え，ハッシュレートが下がっていく．このことから，ブロック生成するインセンティブがなくなり，どんなに少ない手数料でもブロックに追加することは，ネットワーク全体の安全性を低下させることになる．これは一般に，経済学においてコモンズの悲劇と呼ばれる現象である．

さらに，ビットコインの仕様上，フルノードを立てるインセンティブが少ないことから，先述したように，ビックブロックを採用した場合ネットワークのノード数は減少する．このときマイナーは手数料の高いトランザクションをネットワークへブロードキャストしないことで，そのマイナーで手数料を確保する戦略が取る可能性がある．そのため，高い手数料を設定しても，そのトランザクションが承認が遅延する可能性がある．こうした将来起こり得る問題を解決するために，PoA ではフルノードを立てることで，各ノードにインセンティブを与えるプロトコルを提案している．PoA でも，Bitcoin-NG と同様にマイニングノードはトランザクションの含まれていないブロックを生成する．そして，ブロックヘッダのハッシュを擬似乱数として扱い，フルノードの中から複数のノードを抽出する．抽出されたノードは自身の秘密鍵での署名をブロックヘッダに付加しそのブロックをネットワークに伝播する．これを繰り返すことによって，ブロックに含まれる署名が増えていき，規定の数の署名が得られたとき，その抽出されたノードは同様の署名を行いトランザクションをブロックへ追加し，それを伝播する．こうすることで，マイニングノードと抽出されたノードの間で，トランザクション手数料を分配することができる．抽出方法に関して，指定以上の金額のトランザクションをデポジットする

必要がありその要件を満たしたノードから抽出される．これは，ネットワーク内での通貨の保持量によってブロック生成者を確率的に選択する Proof of Stake と呼ばれる承認アルゴリズムを応用したものである．

第 3 章

P2P ネットワーク

3 章 1 節では，ブロックチェーンの構成要素である P2P ネットワークの概要について説明する．提案手法では，DHT として Kademlia を応用した P2P ネットワークアーキテクチャを提案しているため，Kademlia の概要と Kademlia を応用したブロードキャストについての従来研究について説明する．提案手法におけるデータの送信手法では，これらの従来研究を応用したものを考案している．

3.1　ピアツーピア (Peer-to-Peer) ネットワーク

ピアツーピア (Peer-to-Peer) ネットワークは，P2P とも呼ばれ，TCP/IP より上位のアプリケーション層で構築されるオーバレイネットワーク (論理ネットワーク) の一種である．一般的に通信ネットワークと言った場合「ゲートウェイネットワーク」のように，物理的なゲートウェイで接続されたネットワークを指し，TCP/IP 以下のレイヤでネットワークの基本的な形が規定される．これに対し，オーバレイネットワークは下位レイヤの伝送方式の違いに関係なく，既存のネットワークを横断するネットワークとして仮想的に構築されるもので，その構成を容易に変更することができ，また，多様な特性を持つネットワークを同時に構築することができるといった特徴を持つ．P2P ネットワークでは，サーバ・クライアント型のネットワークのように，サービスの提供者と利用者を明確に区別せず，各ノードがサーバとクライアント両方の機能を持つピア (Peer) としてネットワークを構成することで，互いにサービスを提供し合う．また，サービスを提供するピアがネットワーク上で分散するため，ユーザが要求するサービスの提供ピアを探索する機構が必要となる．P2P ネットワークを用いたネットワークサービスの形態は，ピアの探索機構によって，サーバ・クライアント型のネットワークを融合させたハイブリッド型 P2P ネットワークと，完全な分散環境であるピュア型 P2P ネットワークに分類される．

ハイブリッド型 P2P ネットワークは，サービス提供が可能なピアを探索するためにサーバ

を用いる．ハイブリッド型 P2P システムを適用したアプリケーションとしては，Napster[36]，OpenNap[37] などがある [38]．これらの P2P ネットワークでは，探索サービスを提供するサーバが，ネットワークに参加している全ピアの識別子 (IP アドレスなど) や，それらのピアが提供可能なサービスを，インデックス情報として一括して管理する．ピアは所望のサービスを提供できるピアを探索する場合，サーバに問合せを行い，サービスを提供するピアの情報を得たあとは，ピア同士の直接通信によってサービスの提供を受ける．このようにハイブリッド型 P2P システムでは，サーバへの問合せのみで，容易にサービスの提供可能なピアを発見できる．しかし，限られたサーバのみに探索の機能が集中しているため，サーバが故障もしくは停止した場合，システム全体が停止してしまう．また，完全な分散環境ではないため，ピア数が増大した場合にサーバにデータの管理および探索処理が集中する．

ピュア型 P2P ネットワークは，サービス提供の中心となるサーバを用いないシステムである．ピュア型 P2P システムを適用したアプリケーションとして，Winny[39] や Share[40]，Gnutella[41]，[42] 等が挙げられる．サーバレスのネットワークであるため，クエリによって探索する際，目的のサービスを持ったピアが現れるまでネットワーク内を探索する．つまり，単純な分散探索において，目的のサービスを持ったピアが探索に掛からない限り他のピアを長時間探索し続ける．クエリの探索によってサービスを提供するピアの情報を得たあとは，ピア同士の直接通信によってサービスの提供を受ける．更に，ネットワーク内にサーバが無いことで各ピアのアカウント管理などによるセキュリティ管理が難しいため，悪意あるピアの参加には対応しにくいという問題がある．その反面，各ピアの把握が困難なことから匿名性を保証するという利点がある．一方，サーバ・クライアント型やハイブリッド型において，限られたサーバのみに探索の機能が集中することによってサーバが故障・停止した場合，システム全体が停止してしまうため，サービスの提供側はサーバの維持や運用に膨大なコストを投資しなければならない．つまり，サーバの存在そのものがシステム運用においてボトルネックになり兼ねる．しかし，サーバが存在しないピュア型はコンテンツの管理や探索処理が各ピアに依存することで負荷分散が可能となり，単一障害点を持たない．つまり，ピアの一つが故障・停止してもシステム運用においては影響が無く，サービスの提供側は維持費や運用費に掛けるコストを抑えることが可能である．ピュア型 P2P ネットワークにおけるネットワークトポロジーは，ピア間のリンクに特定の論理的規則性を持たせた構造 (Structured) 型と特定の論理的規則性を持たせない非構造 (Unstructured) の二種類に分類される．

上記 2 つの P2P は互いにトレードオフの関係になっている．例えば，ピュア型 P2P はサーバのようなネットワーク全体を把握する管理者の役割を果たすノードが存在しないため，探索処理やデータ管理の負荷は各ピアに分散されるが，各々に関係のないデータの探索処理であっても中継ノードとしての役割を担う．一方，ハイブリッド型 P2P では，インデックス・サーバがデータ管理や探索処理を行うため，各ピアは他のピアの探索に関わる必要はない．しかし，代わりにインデックス・サーバへの負荷が集中してしまう．そこでこれら 2 種類の P2P

ネットワークの欠点を補い，かつ利点を生かしたアーキテクチャとしてスーパーノード型 P2P ネットワークが存在する．スーパーノード型 P2P システムを適用したアプリケーションとして Skype[43] や KaZaA[44] 等が挙げられる．スーパーノード型 P2P において，スーパーノード (また，はスーパーピア) と呼ばれる特別なピア群がデータ探索用クラスタを形成し，その群がサーバのような役割を果たす．スーパーノードはサービスの提供側が予め設けているわけではなく，ネットワーク内の一般のノードの中から性能や帯域等の条件を満たすものが任意で選ばれる．一度スーパーノードが稼働すると以降のネットワーク構成はスーパーノード群が管理することになり，ハイブリッド型の特徴である効率的な情報管理が可能となる．スーパーノードの数はネットワーク内のピア数に対して一定の割合になるように動的に変化する．つまり，スーパーノードは固定的ではないため，スーパーノード 1 つが故障・停止しても，数ある他のピアの中から再び選ばれるのでスケーラビリティが保証されている．また，スーパーノード群がデータの探索処理を担うため，他のピアは自身と関係のないデータの探索処理が開始されてもクエリを中継することはない．よって，ピュア型 P2P のような探索クエリによる帯域圧迫も起こらないシステムである．しかし，スーパーノード型 P2P において，スーパーノードの算出の方法やスーパーノード群でのデータの管理の方法等が複雑なこと，ネットワーク内のノード数がある程度の規模にならないとシステムとして安定した QoS を提供できないことから，上記 2 つのシステムに比べ実装が難しい．上記 3 つのピアの探索機構を簡潔にまとめた図表を図 3.1 に示す．以下の項では，ピュア型の P2P ネットワークにおける非構造型と構造型の従来研究について説明する．

3.1.1　非構造型ピュア P2P ネットワーク

非構造型トポロジーを用いた P2P ネットワークは，DHT のように，特段の論理的規則を持たないネットワークであり，明確な方針に従って決定された探索トポロジーは存在しない．そのため，クエリ伝播のためのネットワークトポロジーと，データ配置は独立している．非構造トポロジーを用いた P2P ネットワークでは，クエリを伝播すべき隣接ノードが定まらない．そのため，ネットワークを構成することや管理することは比較的容易に実現できるものの，コンテンツの探索にはランダムウォーク探索やフラッディング探索を行うため遅延が大きくなり必ずしも効率的ではない．代表的な技術として，Gnutella や BitTorrent[32]，Freenet[33]-[35] 等が挙げられる．所望のデータを保持するノードにクエリが伝播されると，クエリが送られてきたノードを逆順にたどって，クエリを発行したノードにデータを送信する．このため，トラヒックや探索にかかる遅延が大きい．しかし，任意の論理ネットワークを構築できるため，システムを運用しやすいといった利点がある．以下に，非構造型 P2P ネットワークにおいて代表的な Guntella について述べる．

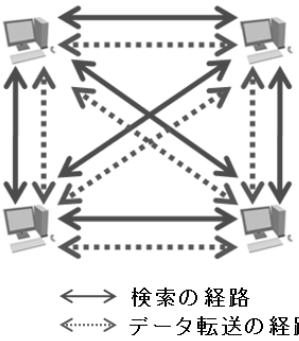
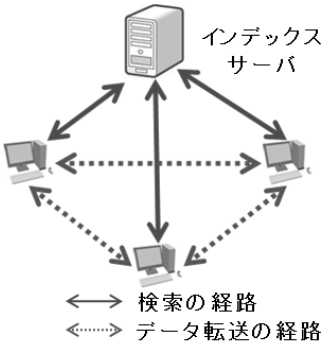
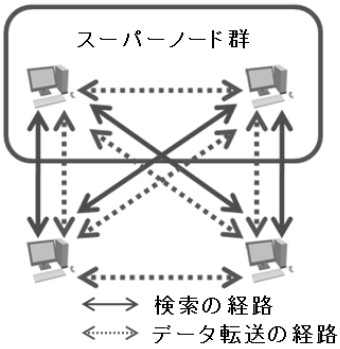
	ピュア型	ハイブリッド型	スーパーノード型
構成			
ｱﾌﾟﾘ	Winny・Gnutella	Napster・Open Nap	Skype・KaZaA
特徴	・データの管理・検索はピアが行う ・自律分散システム ・クエリは他のピアを中継する	・検索処理はサーバが行う ・データの所在はサーバが管理 ・データ転送はピア間で行う	・高性能のノードがサーバのようにデータの管理・検索を行う ・スーパーノードは複数存在する
利点	・拡張性が高い ・耐障害性が高い ・サーバの維持コストが不要	・システムの管理・制御が可能 ・検索における遅延が少ない ・検索時のトラヒックの増大はない	・ピュア型・ハイブリッド型の両方の利点を持つ
欠点	・長期間探索する可能性 ・セキュリティ管理が難しい ・クエリによるトラヒック量の増大	・耐障害性が低い ・拡張性が低い ・サーバの維持コストが掛かる	・スーパーノードの決定が困難 ・ある程度のノード数が必要

図 3.1. ピアの探索機構の分類表 [31].

Gnutella

Gnutella はファイル共有を目的とした非構造型 P2P のプロトコルの 1 つである．そのため，サーバを介さずに他のノードと通信し，ファイル名による探索とダウンロードを行う．Gnutella において，各ノードは GnutellaNet と呼ばれる独自の論理ネットワークに自律的に参加することになり，そのネットワーク上で 4 つ程度の他のノードとリンクを確立する．そして，それらのノードも同様に，他のノードとリンクを確立することによって再帰的な構造のネットワークを構成する．Gnutella は基本的には相手ホストを探索するためのフレームワークであり，拡張して他の用途に応用することが可能である．また，Gnutella において，ネットワークを維持するための制御パケットのやり取りは全てそのネットワーク上で行われる．その制御パケットは以下の 5 種類がある．

- PING：他のノードを発見するためのパケット
- PONG：PING に対する応答のためのパケットであり，PING を受信したノードのア

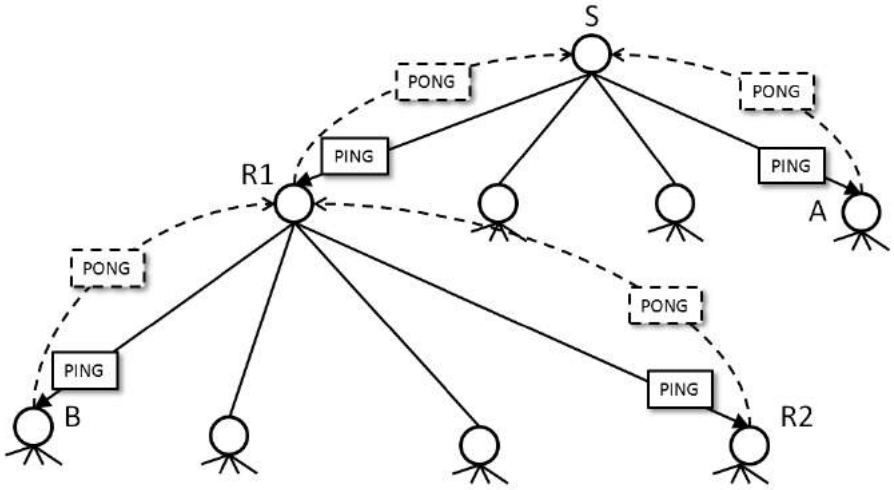


図 3.2. Gnutella の構成と情報取得手順の概要図 [31].

ドレスや利用可能なリソース情報等を含む

- Query： 探索文字列を含んだ探索要求のパケット．ノード間を予め指定された回数 ($TTL(TimeToLive)$) 分だけ転送
- QueryHit：Query に対して探索要求に該当したデータを保有しているノードから送信される応答パケット．データサイズやデータの保存先の URL 等の情報が含む
- PUSH：データを送信するノード側がファイアウォール内にいる場合のデータを送り込むための要求パケット

周囲の情報を取得する際の手順を図 3.2 を用いて説明する．まず，あるノード S は自身とリンクを確立している他のノード A や R1 へ PING をフラッディング (ブロードキャスト) する．PING を受信した他のノード A や R1 は送信元のノード S へ PONG を送信することで応答する．更に，PONG を送信したノード R1 において，自身とリンクを確立している他のノード B や R2 に対して PING の送信元のノード S の識別子を格納した PING をフラッディングする．送信元ノードから送信された PING には予め TTL が指定されているため，ノードを経由する度に PING の TTL を減算する．結果，TTL が 0 になるまで PING は周囲のノードへ伝搬されることになる．一方，PONG は PING が経由してきたノードを逆順に辿っていくことで PING の送信元のノードへ到着する．これによって，PING の送信元のノード S は周囲の他のノードの情報を収集することが可能となる．また，探索処理を行う際の手順を図 3.3 を用いて説明する．あるノード S は探索要求をフラッディングする．ノード S は自身とリンクを確立している他のノード A や R1 の全てに Query を送信する．Query を受信したノード A や R1 は，まず，送信された Query が以前に送信されたものであるか調べる．同一ノードからの同一の Query の重複を避けるため，その Query が以前に送信されたものであれば，その

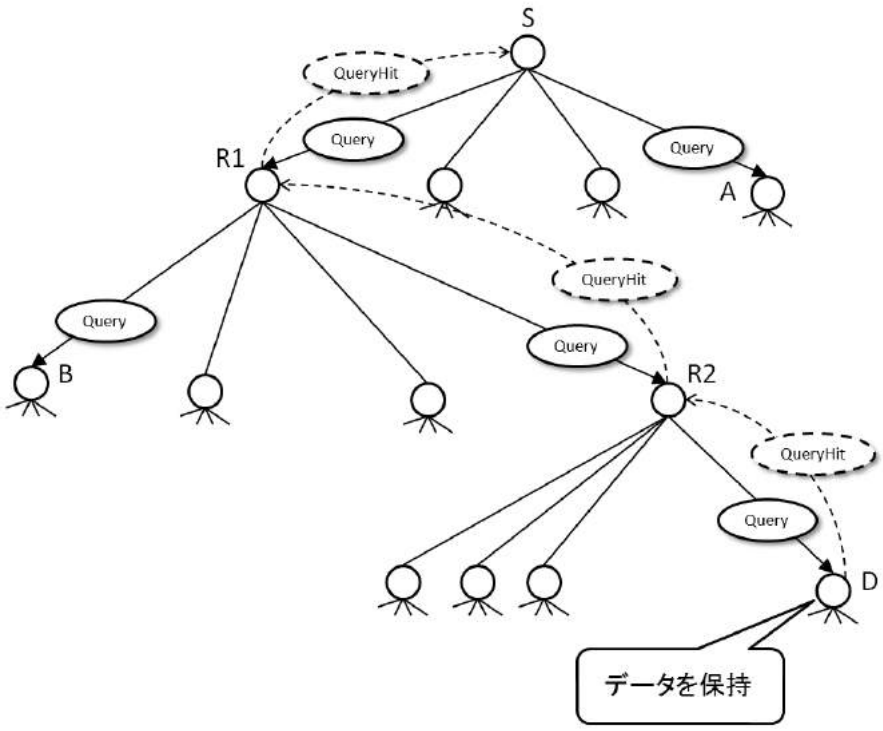


図 3.3. Gnutella の探索処理の概要図 [31].

ノードはその Query を棄却する．その Query が以前に送信されたものでなければ，自身が探索要求対象のデータを保持しているかを確認し，Query を受信したノード A や R1 が探索要求対象のデータを保持していない場合，QueryHit は送信しない．その変わりに，ノード R1 において，自身とリンクを確立している他のノード B や R2 の全てに Query を送信する．同様のことを繰り返していき，ノード D において，ノード D が探索要求対象のデータを保持している場合，Query の送信元ノード S へ QueryHit を送信することで回答する．この時，送信された QueryHit は，PONG 同様，Query が経由してきたノード R1 や R2 を逆順に辿ることによって探索要求元のノードへ送信される．また，Query には，PING 同様，TTL が設定されており，ノードは自身がリンクを確立している Query を送信してきたノード以外の他のノードへ Query に設定された TTL を減算して送信する．以下，Query は再帰的に周囲のノードへ伝搬されていき，最終的に Query の TTL が 0 になった時点で探索は終了する．

3.1.2　構造型ピュア P2P ネットワーク

構造型トポロジーを用いた P2P ネットワークは Distributed Hash Table(DHT) などを用いている．DHT はネットワークトポロジーの決定やデータ探索に必要なキーは位置を厳密に定義するシステムである．構造型トポロジーを用いた P2P ネットワークとしては，

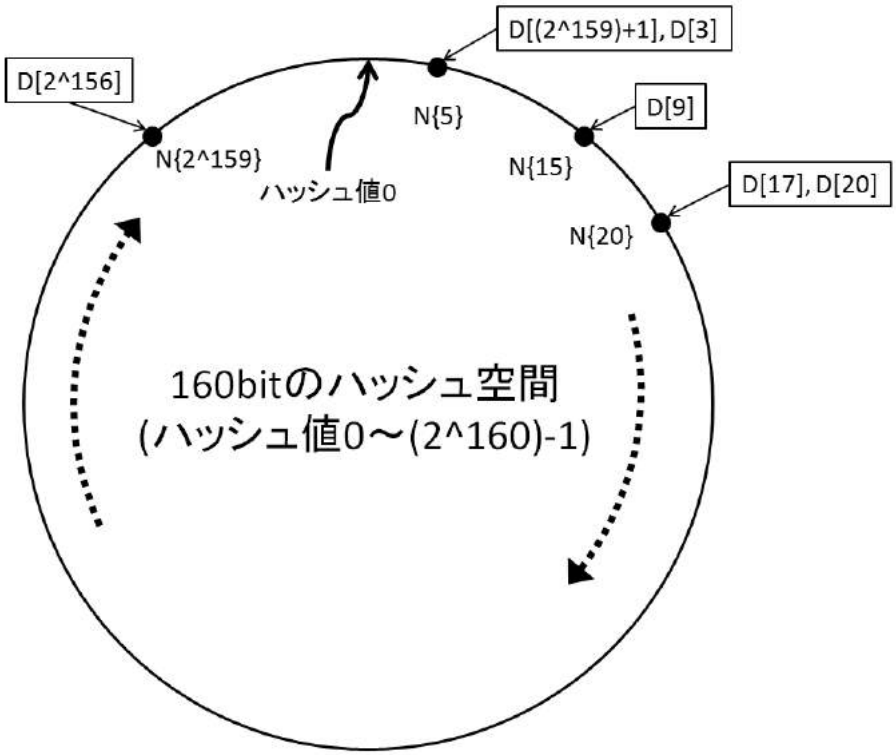
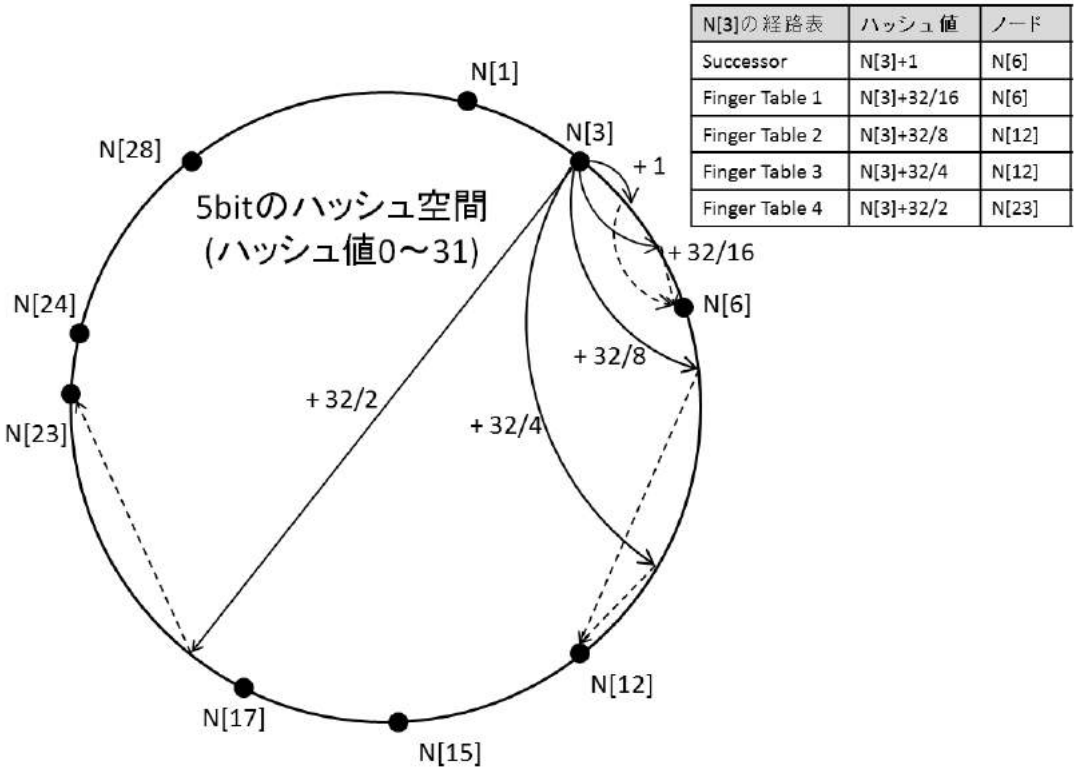


図 3.4. Chord の構成の概要図 [31].

Kademlia[46]，CAN[47]，Pastry[48]，Chord[49]，Tapestry[50] などがある．また，これらの DHT 上でサービスを提供することを想定した研究も行われている．構造型トポロジーを用いた P2P ネットワークにおいて，ピア個別情報から固定長の疑似乱数を生成するハッシュ関数と呼ばれる演算手法を適用してハッシュ値を算出し，それらをハッシュ空間と呼ばれる論理空間の一点に対応付ける．また，ピアが保持する各データのキー情報も同様にデータ名などをもとにハッシュ値を求め，ハッシュ空間内でもっとも近いピアにそのキー情報を格納する．データを探索する際には，探索対象のデータのハッシュ値を算出することで，そのハッシュ値に最も近いピアにクエリを転送することが可能になり，要求するデータのキー情報を得ることができる．このようにネットワークのトポロジーやクエリの転送方法を明確に定義することにより，探索を効率化している．また，DHT に参加するノード数 N に対して，どのノードから探索を開始しても，全てのノードに $O(N)$ オーダよりも良いオーダ (Chord 等の代表的なアルゴリズムでは $O(log_2(N))$) により到達可能なことが保証されている．以下に，構造型トポロジーを用いた P2P ネットワークにおいて代表的な Chord と CAN について述べる．



N[3]の経路表	ハッシュ値	ノード
Successor	N[3]+1	N[6]
Finger Table 1	N[3]+32/16	N[6]
Finger Table 2	N[3]+32/8	N[12]
Finger Table 3	N[3]+32/4	N[12]
Finger Table 4	N[3]+32/2	N[23]

図 3.5. Chord の経路表構成の概要図 [31].

Chord

Chrod は，探索対象となるネットワーク上の各データ，およびこれらを分散管理する各ノードは，一般的に 160bit の識別子の独自の識別子が与えられている．160bit の識別子として，ノードの IP アドレスやデータファイル名を基に，Secure Hash Algorithm 1 (SHA-1) 等のハッシュ関数から得たハッシュ値を用いる．ハッシュ値は図 3.4 の様に円状のハッシュ空間に割り当てられることによって構成される．この時各データは，ノード同様，自信に割り当てられたハッシュ関数を基に，ハッシュ空間上のハッシュ値から正の方向 (時計回り) に見て，次に現れるハッシュ値を持つノードに管理されられる．ノードの数を p，ハッシュ値を a，データの数を q，ハッシュ値を b とした場合，各ノードを ID-N[a1]，ID-N[a2]，$\cdots$，ID-N[ap] とし，各データを ID-D[b1]，ID-D[b2]，$\cdots$，ID-D[bq] とすると，ID-N[a1] のノードは，ハッシュ空間の ID-N[ap] $< x \leq 2^{160} - 1$ および $0 \leq x \leq$ ID-N[a1] の範囲内のハッシュ値が付与されたデータを管理する．以下同様に ID-N[a2] のノードは ID-N[a1]$< x \leq$ID-N[a2] のハッシュ値が付与されたデータを管理する (図 3.4).

各ノードは自身が管理するデータ情報を自身の持つ専用のテーブルに保持する．テーブルは

データ毎のハッシュ値や IP アドレス，ポート番号等から構成されたエントリ情報から構成される．Chord において，ネットワーク上のデータを多数のノードが分担して各々が管理している．そのため，自身が担当する以外のデータに対する情報を取得するためにはノード同士が互いに通信する必要がある．つまり，各ノードは複数の他のノードの IP アドレスやポート番号等のノード自身の情報を経路表として持つ必要がある．経路表は，ハッシュ空間上で自身のハッシュ値から見て正の方向 (時計回り) に最初に現れるノードの情報を「Successor」として保持している．更に，自身のハッシュ値から見て負の方向 (反時計回り) に最初に現れるノードの情報を「Predecessor」として保持している．経路表にはこれ以外に，自身のハッシュ値から見て正の方向に現れる複数のノードの情報を「Successor List」また，は「Finger Table」として保持している．Successor List は，自身のハッシュ値から見て正の方向に最初に現れる m 台 (ハッシュ空間が 2^m の時) のノードの情報を持つリストである．一方，Finger Table は，自身のハッシュ値を X とした時，$X + 2^m (0 \leq m < 160)$ のハッシュ値を担当領域とするノードの情報を持つリストである．つまり，自身のハッシュ値からハッシュ空間の大きさの $1/2$，$1/4$，$1/8$，$\cdots 1/2^m$ 先のハッシュ値を管理する他のノードの情報を持つことになる．具体的に 5bit のハッシュ空間から構成される図 3.5 で説明すると，ID-N[3] において，ID-N[3] からハッシュ空間の大きさの 1/2 先のハッシュ値は 18 を示す．そして，ハッシュ空間において，18 のハッシュ値を担当領域とするノードは ID-N[23] であることから，ID-N[3] の経路表の 1/2 先のノードの部分に ID-N[23] が格納される．同様に，ハッシュ空間の大きさの 1/4，1/8 先のノードの情報には ID-N[12] が，1/16 先のノードの情報には ID-N[6] が格納される．Finger Table を用いた場合，ネットワークを構成するノードの数を N とすると，任意のノードに $O(log_2(N))$ のホップ数 (オーダまた，は Path Length) で到達が可能である．そのため，Successor List と比較して，スケーラビリティに優れた効率的な探索が出来ることから，一般的には経路表は Successor と Predecessor，そして Finger Table で構成される．

Chord のデータ探索処理において，ネットワーク上のあるデータに対して探索処理をあるノードから要求されたノードは，該当データのファイル名等を基にハッシュ値を求める．算出されたハッシュ値が，ハッシュ空間内において，自身の管理する範囲内に該当する場合は，自身のテーブルを参照し，要求元のノードへ回答する．一方，算出したハッシュ値が自身の管理するハッシュ空間内に存在しない場合，経路表を参照して探索要求メッセージを他のノードへ送信する．送信先の決定の手順として，まず，探索処理対象のハッシュ値が自身のハッシュ値と Successor のハッシュ値の間に位置する場合，つまり，自身のハッシュ値から見てハッシュ空間上で正の方向に，探索対象のハッシュ値，Successor のハッシュ値，の順に並んでいる場合，送信先として Successor を選択する．一方，探索対象のハッシュ値が自身のハッシュ値と Successor のハッシュ値の間存在しない場合，経路表に記載されたノードのうち，探索対象のハッシュ値から見てハッシュ空間上で正の方向に向かって最も近くに存在するノードを送信先として選択する．この処理を繰り返すことによって最終的に探索要求は探索対象のハッシュ値

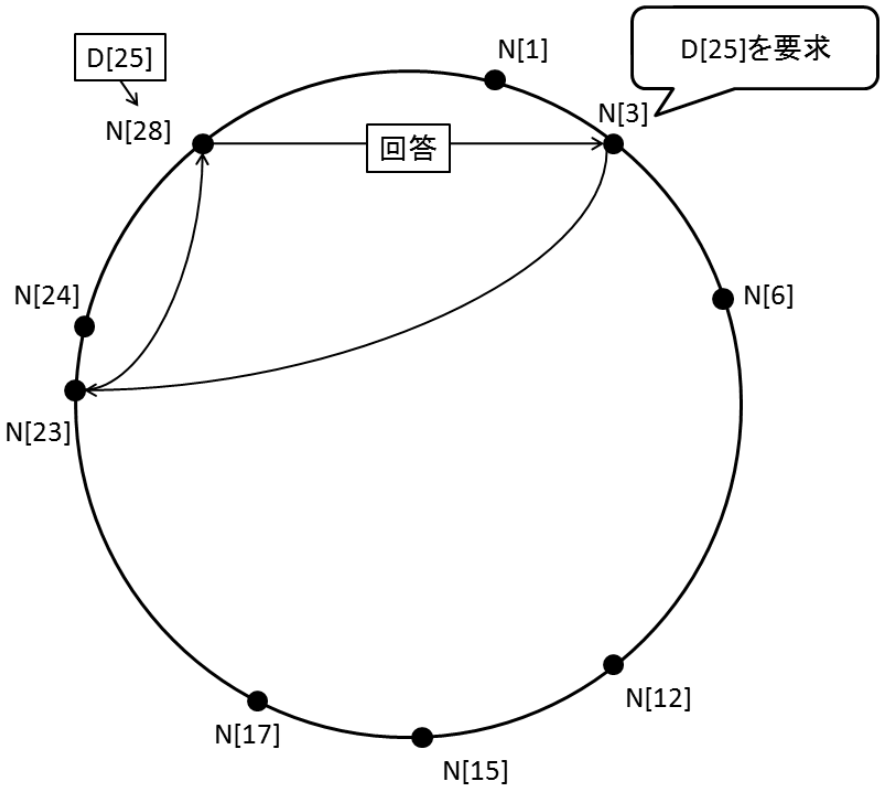


図 3.6. Chord の探索処理の概要図 [31].

を管理するノードに到達し，要求元ノードはそのノードから回答を得ることで探索処理は完了する．探索処理を図 3.5 を基に図 3.6 で説明すると，ノード ID-N[3] がデータ ID-D[25] を探索する場合，ID-N[3] の経路表において，探索要求対象のデータのハッシュ値 25 は経路表内のどのノードのハッシュ値よりも大きい．そのため，ID-N[3] は自身の経路表の中から正の方向へ向かってハッシュ値 25 に最も近い位置に存在するノード ID-N[23] へ探索要求を送信する．この時，ID-N[23] の経路表内の Finger Table において，ハッシュ空間の大きさの 1/2 先の情報を管理するノードは ID-N[12]，1/4 先は ID-N[1]，1/8 と 1/16 先は ID-N[28] を示している．ID-N[23] は自身の経路表の中から正の方向へ向かってハッシュ値 25 に最も近い位置に存在するノードはハッシュ空間の大きさの 1/16 先のハッシュ値を担当領域とする ID-N[28] である．そして，ノード ID-N[28] はデータ ID-D[25] を保持しているため，ID-N[23] から探索要求を受信した ID-N[28] は ID-N[3] へと回答することで探索処理が完了する．

CAN

CAN において，ネットワーク上に分散している各データはインデックスによって管理されている．そして，それらのインデックスはシステムを構成するノードに分散的に管理されてい

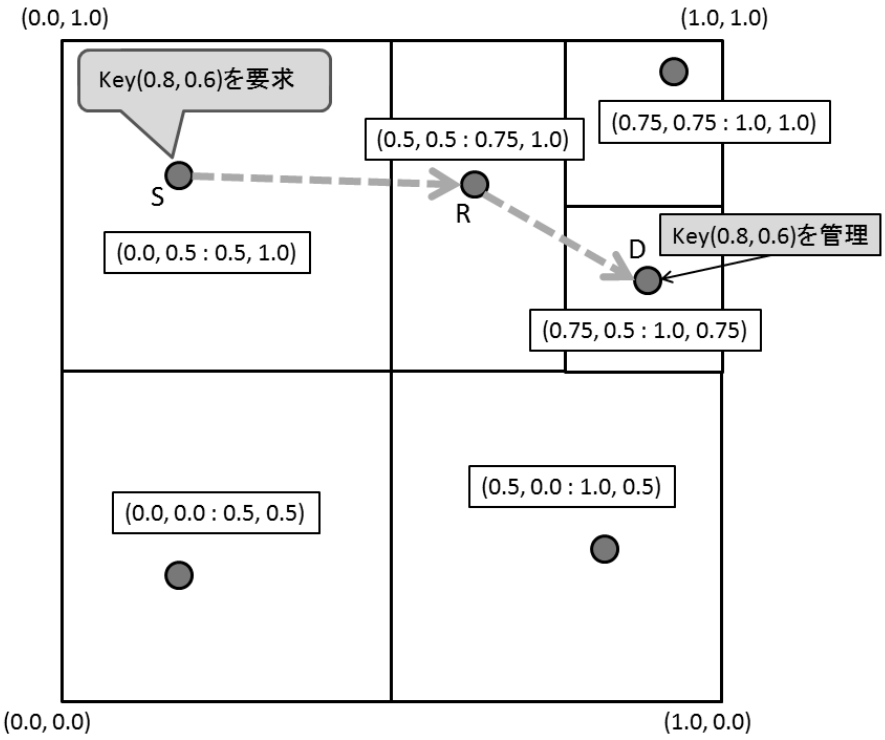


図 3.7. CAN の構成と探索処理の概要図 [31].

る．つまり，各データには予め関連のある key が設定されており，key は適切なハッシュ関数用いて，Chord のハッシュ空間同様，d 次元直交空間 (N 次元トーラス) 上の該当する座標に付与される．座標空間をゾーンと呼ぶ複数の空間に分割し，各ピアが一つのゾーンを管理する．そして，各ゾーンを管理するノードは自身のいるゾーンに該当する key や自身の IP アドレス等の識別子，隣接するゾーンを管理する全てのノードの識別子を保持している．そのため，CAN を利用する場合，ネットワークを構成するノードの数を N，次元数を d とすると，任意のノードに $O(d * N^{1/d})$ のホップ数 (オーダまた，は Path Length) で到達が可能である．探索処理を行う際は，要求元のノードが求める key から目的の座標をハッシュ関数から求め，自身のゾーンと隣接するゾーンの中で目的の座標に最も近いゾーンを管理するノードへ探索要求を送信する．探索要求を受信したノードは同様に目的の座標に向けて探索要求を送信していき，最終的に目的座標を管理するノードへ送信される．目的座標を管理するノードは要求元のノードに対して与えられた key に関する探索結果である自身のアドレスを要求元へ回答する．探索処理の様子を具体的に 2 次元座標空間から構成される図 3.7 を用いて説明する．図において，(x, y) は空間内の座標，$(x1, y1 : x2, y2)$ は，$(x1, y1)$，$(x2, y2)$ を結ぶ線分を対角線とするゾーンを示す．ここで (0.0, 0.5 : 0.5, 1.0) のゾーンを担当するノード S が key(0.8, 0.6) に対応づけられたデータを探索する場合，自身の隣接ノードの中で，key(0.8, 0.6) に最も近いゾーン (0.5, 0.5 : 0.75, 1.0) を管理しているノード R に探索要求を送信する．同様の操作を，

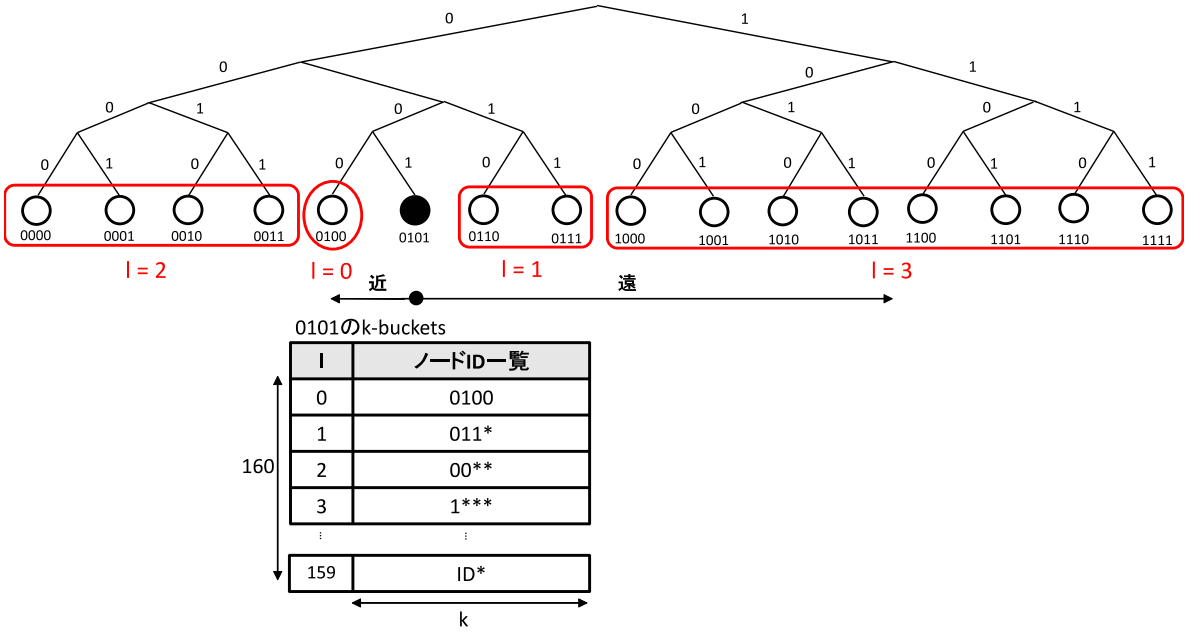


図 3.8. Kademlia 構成の概要図.

座標 (0.8, 0.6) を含むゾーンを管理しているノード D に探索要求が到達するまで繰り返すことにより，対象となるデータを発見することで探索処理を完了する．

3.2 Kademlia と DHT ブロードキャスト

DHT は本来，ピュア型 P2P ネットワークにおいて探索の計算量を効率化するために考案されたものであるが，その論理的なネットワークトポロジーを応用することで，効率的なブロードキャストに応用する研究が行われている [51] - [55]．本稿では，DHT の一つである Kademlia を応用したプロトコルを提案するため，本節では Kademlia の詳細と，Kademlia の応用的なブロードキャストに関する従来研究について説明する．

3.2.1 Kademlia

Kademlia のノード空間はノード ID が 160bit 長で表現されるノードの集合から成る．ノード ID はノードの IP アドレスをもとに，SHA-1 のハッシュ関数から得たハッシュ値を用いる．コンテンツも同様にそのデータファイル名などをキーとしてハッシュ関数より 160bit 長に変換され，キーに近い k 個のノードに格納される．

Kademlia の特徴の一つとして，ノード間の距離をノード ID の排他的論理和演算 (XOR) を用いて決定することである．XOR 距離によると，同じ値を持つ二つのノード ID の距離は

0 となる．また，二つのノード ID 値を 2 進数で表現した時に，先頭からプレフィックス一致長が長いほどノードは近接であるとする．図 3.8 は Kademlia のノード空間を図示した二分木で，説明の簡略化のためノード ID は 4bit としている．

各ノードは，自身の ID との XOR 距離に対して反比例した濃度で隣接ノード情報を格納する k-buckets と呼ばれる経路情報を持つ．この k-buckets は，ノード間距離の最上位ビット毎に最大 160 個のバッファで構成され，各バッファ内にそのノードが知る他ノード情報を k 個格納しておく．他ノード情報は，他ノードからの接続メッセージや，探索時に取得した情報などを基にバッファへの挿入を行う．ただし，バッファに余裕がない場合には，バッファ内の先頭にあるノードに生存確認メッセージを送り，生存が確認されないとき先頭ノードを削除し，新規ノードをバッファの末尾に挿入する．生存が確認された場合，そのノードはバッファの末尾に移動する．これは P2P システム全般の「長寿命のノードは今後も長寿命である」という経験則より，安定した探索が行えるよう，ネットワークの安定性を優先したアルゴリズムである．このため，Kademlia ネットワークはノードの参加，退出 (churn) に対して堅牢であるとされている．

Kademlia において，ネットワークを維持するための制御パケットのやり取りは全てそのネットワーク上で行われる．その制御パケットは以下の 4 種類がある．

- PING : 対象ノードの生存確認
- STORE(Key, Value) : 対象ノードにキーと値のペアを保持される
- FIND_NODE(ID) : 対象ノードから ID に近いノードを k 個取得
- FIND_VALUE(Key) : 対象ノードからキーの値を取得, 値がない場合は, FIND_NODE と同様

新しいノードが Kademila に参加するには少なくとも 1 つのノード ID を知っておく必要がある．この既知のノードが Kademila へ参加するためのブートストラップノードとなる．ノード参加は次のようにして行われる．

(step1)　ブートストラップへ FIND_NODE(0000) を送信
(step2)　参加するノードの ID に近いノードをブートストラップノードから k 個取得
(step3)　それらの k 個のノードを自身のバケットに登録
(step4)　それら k 個のノードに対して FIND_NODE(0000) を送信

(step4) の処理はバケットリフレッシュと呼ばれるもので，各バケットの範囲に含まれるランダムな ID をバケット内のノードに対して探索させる．コンタクトを受け取ったノードは問い合わせ元を自分のバケットに登録する．これにより多数のノードにコンタクトさせてそれぞれのノードの k-buckets に自分自身を登録してもらう．バケットのリフレッシュは，さらに別の目的で使用される．それは，ノードの新規参加や離脱に対して k-buckets を最新の状態に保つ

ためである．

ノードの離脱時には特にやるべきことはない．これも Kademlia の大きな特徴である．他の形式のシステムの例を挙げると，経路表方式の DHT を採用している Chord システムでは双方向リンクの修正，DHT の補正，コンテンツの移動等の処理が必要となり管理コストが増大する．

Kademlia でのキーと値の探索は次ようにして行われる．(step2) は Remote Procedure Call(RPC) で実行される処理で，選択した k 個のノードに対して目標 ID に近い α 個のノードに問い合わせる．ノード探索の期待値は $O(logN)$ となる．

(step1) 自身の k-buckets よりキーに近いノードを k 個選択 (近い順・RTT 順)
(step2) 選んだノード k 個にの先頭から α 個づつ FIND_VALUE(Key) を送信
(step3) α 個の問い合わせ結果に，キーとペアの値があるなら，値を問い合わせ元へ返して終了，なければキーに最も近い値を問い合わせ元へ返信
(step4) (step2) で得たノード一覧と (step3) で得たノード一覧をマージ& ソート (近い順・RTT 順) して (step2) へ

3.2.2 Kademlia の応用的ブロードキャスト

文献 [52] では，Kademlia ネットワークで応用させる手法の提案とそのネットワーク分析を行っている．ここで説明される Kademlia ネットワークのブロードキャストの方法は，主 2 つに分かれる．一つ目は，バケットベースの手法である．まず，送信ノードはメッセージをを自身のバケット内の接続ノードに送信し，受信ノードは自身の領域の範囲内のバケット内の接続ノードにメッセージを転送する．そのときメッセージには，すでに受信した領域内のノードの識別子が含まれていおり，ループを回避するために再度それらには送信されない．安定した状態では，ID を持つノードは最低 1 つの接続先を確保していることに加え，受信ノードはより近い距離のバケット内の全ての既存ノードを知る必要があるため，100% のメッセージ到達率を確保することができる．この手法は，ノードがブロードキャストメッセージを送信する全ての領域が受信者の完全なバケットの数にマッピングされ，送信者よりも近い接続先である場合にのみ，説明されたアルゴリズムが機能する．二つ目は，一般的なプレフィックスベースの手法である．この手法では送信ノードの ID から XOR 距離を元に，ブロードキャストツリーを生成しそれを元に，メッセージを転送していく．この手法では，そのブロードキャストツリーのバランスに偏りが発生してしまう点である．三つ目は，空間分割の手法である．これは，k-ary 探索法 [56] を元にしており，各ノードが自身の担当する ID 区間にある接続ノードに対してメッセージを転送する手法である．この手法は，Chord のように，より送信ノードに近い ID がより多くの情報を持っているという前提条件のため，XOR 距離で構成されている

Kademlia では全て当てはまらず，特定の条件下のみで機能する．

第 4 章

提案手法

4.1 概要

ビットコインでは，アルゴリズムの制約条件によりトランザクション処理能力が低いというスケーラビリティの問題が存在し，これを解決するために多くの研究でスケーラビリティに関する研究と新たなプロトコルの提案がされている．ブロックチェーンでは，安全性，スケーラビリティ，そして非中央集権性が同時に達成することが難しく，このことはブロックチェーンにおける研究では重要な課題となっている．スケーラビリティ改善案の一つであるビックブロックは，ブロックの伝播遅延を引き起こしチェーンのフォークを発生しやすくすること，また，ネットワークのノードにより多くのストレージ負荷がかかるためハードウェア要件を満たさないノードが離脱してしまうことから，安全性と非中央集権性を損なう可能性が指摘されている．

そこで，本稿では，特定ピアに負荷が集中せずに大規模なコンテンツ探索を実現する DHT (Distributed Hash Table) のうち，代表的なアルゴリズムの 1 つである Kademlia をブロックチェーンに応用し，効率的に P2P ネットワーク内へのブロックチェーンのデータのブロードキャスティングを行う手法と，ブロックチェーンのデータを DHT のネットワーク内で分散配備させる手法を組み合わせることで，非中央集権性を保ったまま，ストレージ負荷とネットワーク負荷を分散させる手法の提案を行う．

提案手法のネットワークトポロジーの概観図 4.1 に示す．この図は後述する 2 種類のノードが，それぞれ違うネットワークに参加しており，その二つのネットワーク間がそれぞれのノードの接続によって，繋がっていることを表している．マイニングノードが非構造のピュア P2P ネットワーク，データノードが DHT として Kademlia を応用したネットワークによって構成されている．

次に，提案手法のシステムの概要図を 4.2 に示す．提案手法では，まず DHT ネットワークのデータノード間で発生したトランザクションをマイニングノードへと送信するトランザク

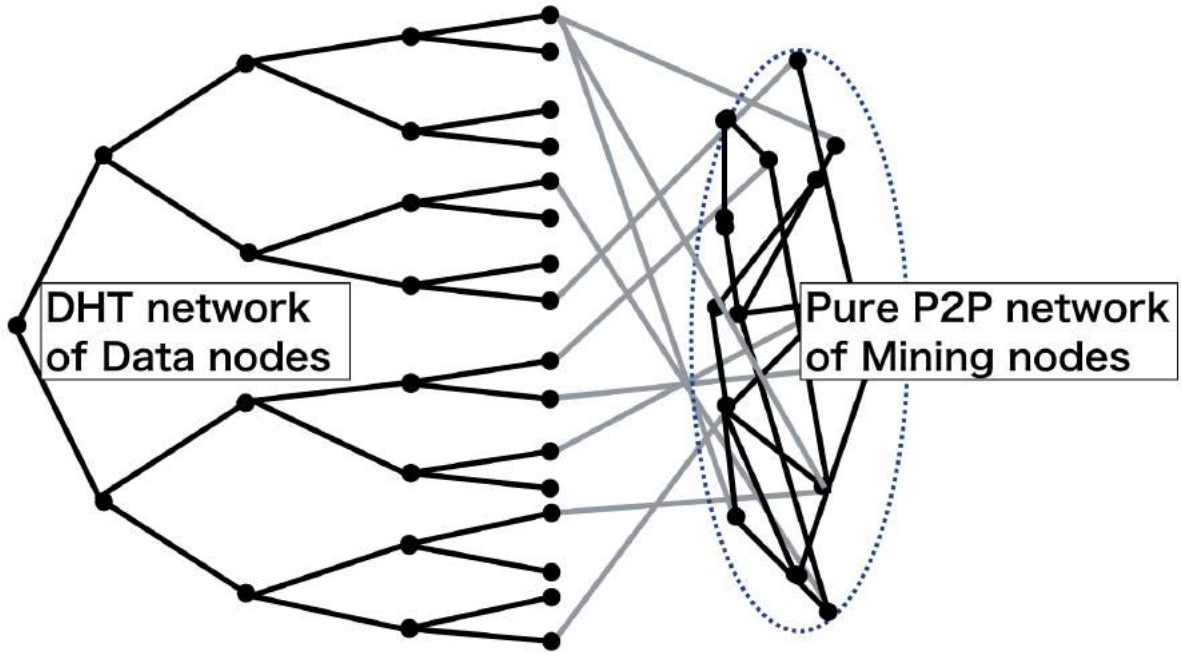


図 4.1. 提案手法ネットワークトポロジー.

ション伝播プロセスを一定回数行い，ブロック生成する準備が行う．ブロックを生成すると，ピュア P2P ネットワークのマイニングノードからデータノードへとブロックデータが送信されるブロック伝播プロセスを行う．データノードで構成される DHT ネットワークにおいて，トランザクション伝播の対象範囲は全てのデータノードであり，ブロック伝播の対象範囲はブロックによって決まる担当クラスタに所属するデータノードのみである．また，データノード間での全てのデータ送信では，後述するブロードキャスティング手法を用いて効率化を図っている．さらに，トランザクションやブロックを転送するときには，必ず他のクラスタへブロックチェーンの参照を行なって不正がないかどうかの検証が事前に行われる．このように，提案手法ではマイニングのための計算作業をマイニングノードが行い，ブロックチェーンデータの自身のストレージへの保管をブロックデータ保有ノード (データノード) が行うことにより機能分離を図っている．

提案手法の DHT ネットワークについて，Kademlia は本来 160 bit の ID 空間から DHT ネットワークが構成されるが，提案手法では 256 bit の ID 空間としており，ハッシュ関数として SHA-256 を用いる．また，提案手法におけるネットワークへのノードの参加離脱や k-bucket 更新のプロセスは Kademlia と同様とする．しかしながら，提案手法では Kademlia のネットワークを用いてデータのブロードキャストを行うため，本来の Kademlia の用途であるクエリによるコンテンツ探索は行わず，以後の節で後述するトランザクション伝播プロセス及びブロック伝播プロセスに従う．

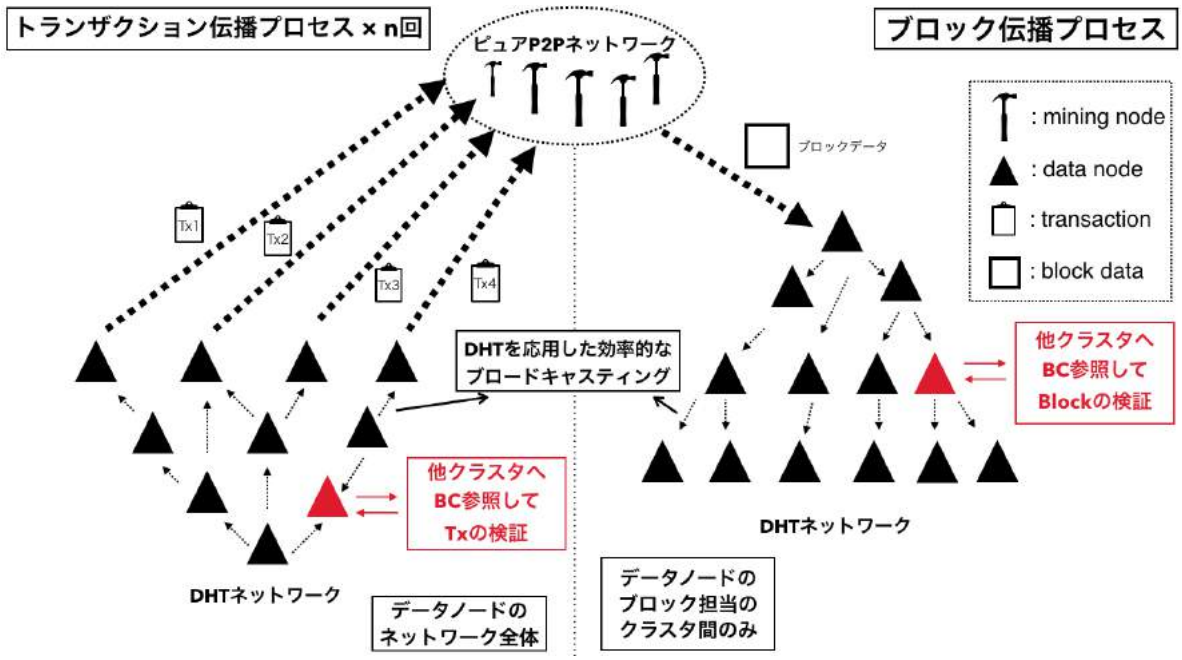


図 4.2. 提案手法システムの概要図.

提案手法では，ブロックチェーンデータ容量の増加速度が増加した場合も，ストレージのハードウェア要件を抑えることが期待できる．また，DHT ネットワーク内で効率的なブロードキャスティングを行うことで，ネットワーク負荷を抑えることができる．この 2 つの利点は，非中央集権性を保つことを前提にしており，一般のネットワークの参加者が離脱せずにスケーラビリティを改善するシステムとなっている．

4.2 節では，提案手法におけるネットワークトポロジーとノードの役割について，4.3 節では，考案したデータの送信手法について，4.4 節では，トランザクションの伝播の手順について，4.5 節では，ブロックの伝播の手順について，4.6 節では，4.4 節と 4.5 節で述べた伝播プロセスの動作例について述べ，最後に，4.7 節では，提案手法の利点と欠点について述べる．

4.2　トポロジーとノードの役割

提案手法の説明のための便宜上のネットワークトポロジの図 4.3 に示す．左の概観図では，図 4.1 をさらに簡略化した構成を表している．右の上面図では，各データノードがクラスタによって，担当するブロックチェーンデータを分割して保有していることを表している．4.4, 4.5 節では伝播プロセスの手順では，この図を用いて詳細なプロセスについて説明を行う．

また，提案手法のために用いる図 4.3，図 4.4，図 4.5，図 4.8，図 4.9 において，クラスタ数を 4 とし，それぞれのクラスタを A，B，…，D と定義している．

次に，ノードの役割について説明する．データノードでは，トランザクションの生成とその

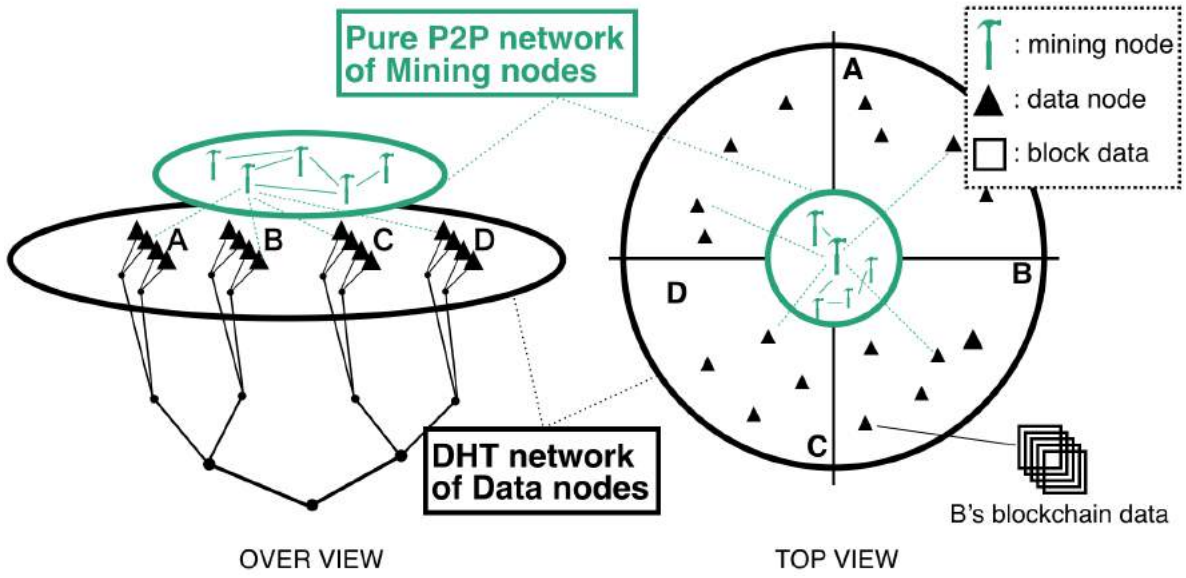


図 4.3. 便宜上の提案手法ネットワークトポロジー.

伝播を行い，マイニングノードはそのトランザクションを受け取り，ブロックを生成している．そして，マイニングノードから送信されたブロックデータをデータノード間で転送している．どちらのノードでも，伝播の際，保有していないデータを持つクラスタへブロックチェーンの参照を行い，送信するデータの検証を行なっている．

4.2.1　データノード

データノードは各々，自身の IP アドレスをハッシュ変換した 256 bit のノード ID と，ノード ID を格納するノードリスト (バケット) を持っており，ノード ID のプレフィックスから，自分の配属されるクラスタが割り当てられる．また，データノードは，図 4.1，図 4.3 のようにマイニングノードの非構造ピュア P2P ネットワークに接続先を持っている．図 4.4 は，ID 空間が 5 bit，ノード数が 32，クラスタ数が 4，プレフィックス桁数が 2 のときのデータノードのネットワークにおける ID ツリーを表している．プレフィックスはノード ID からクラスタを判別するために用いられ，ID やプレフィックスは 2 進数のビットで構成されるため，プレフィックスが n 桁のとき，クラスタ数は 2^n で表される．ノード ID が $\langle 00110 \rangle$ の場合，プレフィックスの 2 桁は $\langle 00 \rangle$ となっているため，図 4.4 における A クラスタへ配属される．図 4.4 において，ノード ID $\langle 00110 \rangle$ をもつノードが所有するバケットを図 4.5 に示す．これはノード ID $\langle 00110 \rangle$ のもつ接続先ノードのリストを表しており，図 4.5 における K-bucket の項目で示される各バケットには，最大 k 個の ID とその ID を持つ接続ノードの IP アドレスが格納されている．Kademlia と同様に，リストを持つノードのノード ID を基準とする XOR

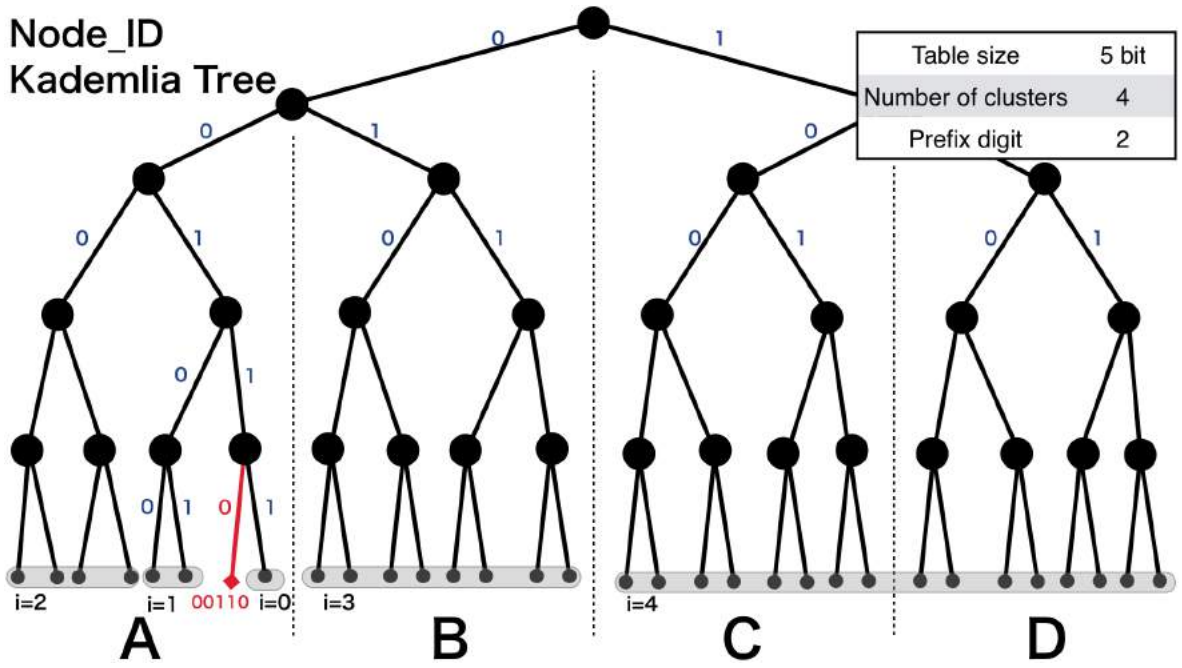


図 4.4. データノードにおける 5 bit のノード ID ツリー.

XOR distance	i	Cluster group	K-bucket (Node list)
1 ($2^0 \leqq XOR < 2^1$)	0	A	00111
2〜3 ($2^1 \leqq XOR < 2^2$)	1	A	00100, 00101
4〜7 ($2^2 \leqq XOR < 2^3$)	2	A	00000, 00001, 00010, 00011
8〜15 ($2^3 \leqq XOR < 2^4$)	3	B	01XXX, 010XX, …
16〜31 ($2^4 \leqq XOR < 2^5$)	4	C	10XXX, 100XX, …
16〜31 ($2^4 \leqq XOR < 2^5$)	4	D	11XXX, 110XX, …

Prefix

Store up to K node IDs in each bucket

Differences from k-bucket of original Kademlia (Not splitting buckets with i)

図 4.5. ノード ID ⟨00110⟩ のもつノードリスト.

距離を $2^i \leqq XOR < 2^{i+1}$ の範囲に区分けし，その範囲内のノード ID から最大 k 個ランダムに選択したものを 1 つのバケットとしている．しかしながら，提案手法では同じ i の値を持つバケットであっても，指定された桁数のプレフィックスが異なる場合，異なるバケットとして区切られるため，1 つのバケット内に存在するプレフィックスは常に 1 種類となっている．そのため，クラスタ C とクラスタ D は i が同値であるが，プレフィックスがクラスタ C を ⟨10⟩ と ⟨11⟩ という 2 種類存在するため，異なるバケットとして扱っている．

データノードの役割は，ブロックチェーンデータをブロックごとに分割し，それを保有することである．マイニングノードで生成されるブロックには判別 ID が付加されており，転送されてきたブロックの判別 ID のプレフィックスと自身のノード ID のプレフィックスが一致するとき，そのブロックデータを保有する．

トランザクションやブロックの転送方法については後述するトランザクションやブロックの伝播プロセスに従う．

4.2.2　マイニングノード

マイニングノードの役割は，データノードから伝播されてくるトランザクションを一定数集め，有効なブロックを生成することである．接続するデータノードは各クラスタから各々最低 1 つ以上の接続先を持ち，データノードからの接続リクエストは拒否しない．マイニングの方法は，ビットコインの従来手法と同様とする．マイニングによって条件を満たすナンスを発見されたブロックは，ブロックハッシュ値とともに，それをさらに SHA - 256 によってハッシュ変換し得られる格納先を判別するための ID (判別 ID) が生成される．そして接続しているデータノードから，判別 ID と同値のプレフィックスを持つノード ID を持つノードを選択しブロックデータを送信する．

4.2.3　トランザクション及びブロックの検証作業

データノードでは，ビットコインの従来手法のように全てのブロックを 1 つのノードで保有しないため，トランザクションやブロックの検証作業として他クラスタのノードにブロックチェーンを参照するプロセス (以下，BC 参照) が必要となる．これはブロックチェーンを持っていないマイニングノードも同様に必要なプロセスである．データノードとマイニングノードの全てのノードがトランザクションやブロックを転送するとき，必ず BC 参照を行う必要があり，転送されるデータは他のクラスタのノードから，検証を受けることで有効となる．これは，従来手法には存在しないプロセスのため，5 章のシミュレーション評価では，これがネットワーク負荷に影響を与える可能性を考慮して，シミュレーションを行っている．

4.2.4　クラスタ

本稿におけるクラスタとは，DHT ネットワークに所属するデータノードを複数に分けたノード群のことを指す．それぞれのノード群には，ブロックに付随する割当 ID によって，保有するノード群が特定される．クラスタ数は，提案手法の性質上，2^n 個に分けられ，特定の条件を満たす場合に，n の値が増加する．特定の条件は，ノード数やブロックチェーンデータサイズ等の条件が考えられるが，本稿では特に定めておらず，一般のネットワーク参加者

の CPU やストレージ等のハードウェア要件が満たせなくなる場合にクラスタ数を増加させることが望ましい．クラスタの識別子は，データノードの項で述べたように，ノード ID のプレフィックスで決められる．クラスタ数を増加させる場合，プレフィックスを 1 桁増やすことで，クラスタ数は倍増する．

4.3 DHT を応用したブロードキャスティング手法

ここでは，全てのデータノードへデータをブロードキャストを行う手法について述べる．

5 章におけるシミュレーションでは，ここで述べる MFFF 手法，RRL 手法に加え，ノードの持つバケットに存在する全てのノードへ送信するフラッディングを提案手法として用いている．

4.3.1 MFFF 手法

これは，XOR 距離から構成された ID ツリーを持つ Kademlia のトポロジーを応用し，送信する最初のノードのノード ID からより XOR 距離が遠いノード ID を持つノードへ，データの転送を行う手法である．トランザクション伝播ではトランザクションを生成し送信するデータノードを最初のノードと定義する．また，ブロック伝播において，マイニングノードはノード ID を持たないため，ブロックデータを生成したマイニングノードから送信されたデータノードを DHT ネットワーク内での最初のノードと定義する．本稿では，この方式を MFFF (More Far From the First node) 手法と呼ぶ.

まず，送信を行う最初のデータノードは，送信するデータに自身のノード ID (以下，$firstID$) を付加し，自身の持つノードリストに存在する全てのノードに最初の送信を行う．このとき，その受信ノードの自身のノード ID (以下，$myID$) と $firstID$ の XOR 距離は，$myID \oplus firstID$ で表される．同様に，受信ノード自身のノードリストに存在する n 個の接続ノードのノード ID(以下，$yourD$) と $firstID$ との XOR 距離は，$yourID \oplus firstID$ と表される．受信ノードはデータの転送を行う際，以下の条件式を満たした場合にのみ，転送を行うことが可能となる．

$$\text{条件式}：myID \oplus firstID < yourID \oplus firstID$$

受信ノード自身のノードリストの接続ノードへ送信する際，この条件式を満たしたデータノードにのみ転送を行う．以上の手法で，データノードのネットワーク全体の ID ツリーにおいて，最初の送信ノードの ID からより遠いノードにのみ送信を行うことができる．

図 4.6 の例では，トランザクション伝播の場合の送信を例に動作説明する．最初の送信ノードである ID[00110] から ID[10100] へ送信した後，受信した ID[10100] では，転送先である ID[01111]，ID[11001] にそれぞれ送信するかどうかの判断を行う．ID[01111] の場合，XOR

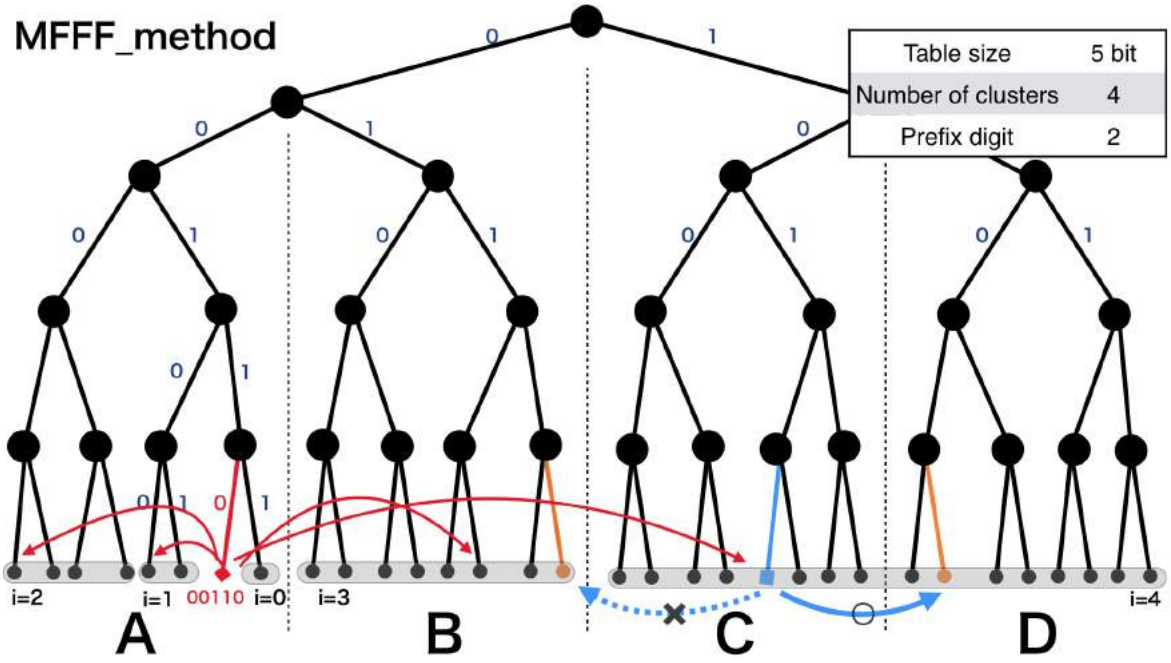


図 4.6. MFFF 手法の例.

距離の比較は $00110 \oplus 01111 < 00110 \oplus 10100$ となるため，最初の送信ノードからの XOR 距離は ID[01111] の方が，ID[10100] より近いことが分かる．従って，条件式を満たさないため，ID[10100] では，ID[01111] へ送信を行わない．同様に，ID[11001] の場合，XOR 距離の比較は $00110 \oplus 11001 > 00110 \oplus 10100$ となり，これは条件式を満たすため，ID[11001] へ送信を行うことができる．このように，より XOR 距離がより遠くのノードへのみ送信を行うことで，送信の効率化を図っている．

ブロック伝播の場合は，このアルゴリズムが同じクラスタ内で同様に動作する．

4.3.2　RRL 手法

MFFF と同様に，XOR 距離の論理的な構成を応用した手法について述べる．これは経由ノードが増えるごとに，XOR 距離によって範囲を決める条件を増やすことで，送信できる範囲を再帰的に制限する手法である．本稿では，この方式を RRL (Recursive Range Limit) 手法と呼ぶ．まず，送信を行う最初のデータノードは，送信するデータに自身のノード ID (以下，$firstID$) を付加し，自身の持つノードリストに存在する全てのノードに最初の送信を行う．同様にデータを受信したノードは自身のノード ID (以下，$secondID$) を付加し転送先ノードへ送信していく．

最初のノードによる送信が終わったとき，その受信ノードでは各々 $2^i < firstID \oplus secondID < 2^{i+1}$ をとりうる i が一意に決まる．secondID をもつノード自身のノードリス

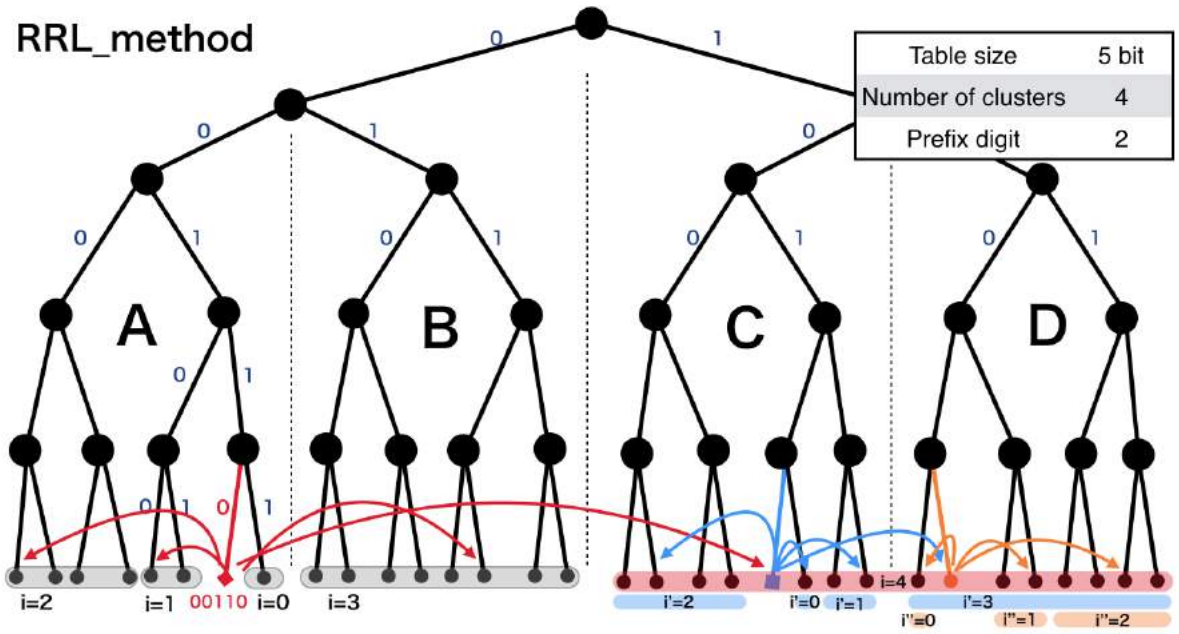


図 4.7. RRL 手法の例.

トに存在する n 個の接続ノードのノード ID(以下，$yourID$) をもつとき，$secondID$ をもつノードが転送可能となる XOR 距離の範囲の条件式は以下のようになる．

$$\text{条件式 } 1 : 2^i \leq secondID \oplus yourID < 2^{i+1}$$

次に転送されるデータノードの ID(以下，$thirdID$) の持つデータには，$firstID$ と $secondID$ の情報が付加されている．同様に，この時，$2^j <= firstID \oplus secondID < 2^{j+1}$ をとりうる j が一意に決まる．同様に，$thirdID$ をもつノード自身のノードリストに，n 個の接続ノードのノード ID(以下，$yourID'$) があるとき，$thirdID$ をもつノードが転送可能となる XOR 距離の範囲の条件式は以下のようになる．

$$\text{条件式 } 1 : 2^i <= secondID \oplus yourID' < 2^{i+1}$$

かつ

$$\text{条件式 } 2 : 2^j <= thirdID \oplus yourID' < 2^{j+1}$$

このように，経由ノードの ID をデータに付加し，範囲制限を再帰的に増やすことで転送可能な範囲が狭まる．この手法は 2^i ごとにバケットをもつ Kademlia では，各バケットに 1 つ以上の ID が存在するとき，到達率が 100% になる送信手法となっている．

図 4.7 の例でも同様に，トランザクション伝播の場合の送信を例に動作説明する．最初の送信ノードである ID[00110] から ID[10100] へ送信した後，受信した ID[10100] では，送信できる範囲が $2^4 < 00110 \oplus 10101 < 2^5$ つまり，最初の送信ノードである ID[00110] の持つ $i = 4$

のバケットに限定される．次に，ID[10100] から，ID[11001] に転送した場合，ID[11001] の送信できる範囲は，$2^4 < 00110 \oplus 10101 < 2^5$ かつ $2^3 <= 10100 \oplus 11001 < 2^4$ つまり，先ほどの範囲から，さらに限定された最初の送信ノードである ID[00110] の持つ $i = 4$ のバケットかつ ID[10100] の持つ $i' = 3$ のバケットに限定される．

ブロック伝播の場合は，このアルゴリズムが同じクラスタ内で同様に動作する．

4.4　トランザクション伝播プロセス

あるクラスタのデータノードから送信されるトランザクションが，そのクラスタ内のデータノード間のみで転送を行なった場合，ネットワークトポロジーの偏りによって一部のマイニングノードにしか転送されない可能性がある．また，先述したように，マイナーのインセンティブを考慮したとき手数料を高く設定したトランザクションがマイニングノードのネットワーク内で転送されない可能性も指摘されている [22]．そのため，提案手法では他のクラスタも含めた全てのデータノードへトランザクションの転送を行なっている．

トランザクション伝播のプロセス手順

トランザクション伝播プロセスの手順を以下に示す．

(step a)　あるデータノードがトランザクションを生成し，そのノードは自身のノードリストの全てのノードにトランザクションを送信

(step b)　受信ノードは，自身の所属しているクラスタ以外のクラスタの接続ノードに BC 参照を行い，トランザクションの有効性を検証

(step c)　その受信ノードは自身のノードリストから適用するブロードキャスト手法に応じたノード ID を持つノードへその検証済みトランザクションを転送し，また，自身の持つマイニングノードへそのトランザクションを転送

(step d)　(step2) - (step3) を繰り返し，マイニングノードを含めたネットワーク全体にトランザックションを伝播

(step e)　マイニングノードは転送されてきたトランザクションに BC 参照を行い，有効性が検証された後，生成するブロックへそのトランザクションを追加

4.5　ブロック伝播プロセス

データノードでは，割り当てられたブロックデータを保有するが，その割り当て方は判別 ID のプレフィックスによって決められる．ノード ID のプレフィックスによって，保有すべ

きノードがブロックごとに特定されているため，マイニングノードは自身の持つノードリストから同じノード ID のプレフィックスを持つデータノードにのみブロックデータを送信する．データノードも同様に，自身の持つノードリストの中から，同じノード ID のプレフィックスを持つノードにのみブロックデータが転送する．以上を踏まえて，ブロック伝播プロセスの手順を以下に示す．

(step f) マイニングノードがマイニングを行い，トランザクションを集積しブロックを生成し，そのブロックから割り当てるクラスタを決めるための判別 ID を生成

(step g) マイニングノード自身の持つノードリストから，判別 ID とプレフィックスの等しいノード ID を持つノードへブロックデータを送信し，また，そのノードの所属する以外のクラスタの接続ノードへブロックヘッダを送信

(step h) ブロックデータを受信したデータノードは，BC 参照を行いブロックの有効性を検証

(step i) そのデータノードは，(step g) と同様に検証済みのブロックデータを，自身の持つノードリストから，判別 ID とプレフィックスの等しいノード ID を持ち，適用するブロードキャスト手法に応じたノード ID を持つノードへブロックデータを送信し，自身の持つブロックチェーンにそのブロックデータを付加して保有

(step j) (step i) - (step j) を繰り返し，データノードの中から，そのブロックデータを担当するクラスタ全体へブロックデータを伝播

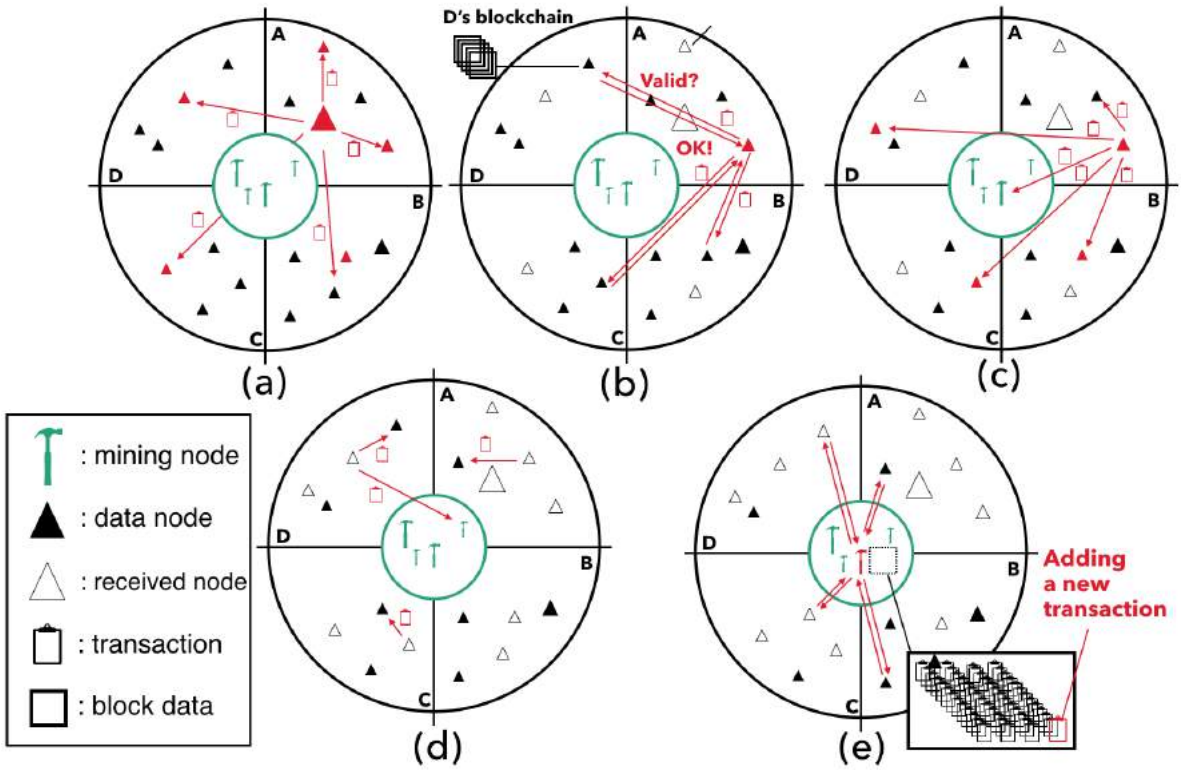


図 4.8. トランザクション伝播プロセスの手順.

4.6　動作例

ここでは，クラスタ数が 4，プレフィックスの桁数が 2，ID 空間が 5 bit のときの動作例について述べる．プレフィックスの桁数は 2 桁であるため，⟨00⟩，⟨01⟩，⟨10⟩，⟨11⟩，と表されるクラスタ A，B，C，D と定義する．

トランザクション伝播プロセス手順を図 4.8 に，ブロック伝播プロセス手順を図 4.9 に示す．

(step a) では，ノード ID のプレフィックス ⟨00⟩ を持ち，クラスタ A に配属されているノードが新しいトランザクションを生成し，自身の持つノードリストのノードへ送信する．(step b) では，そのトランザクションの受信ノードは，接続ノードの中からクラスタ B，C，D のノードに BC 参照を行い，トランザクションの有効性を検証する．(step c) では，その受信ノードは，自身のノードリストの全てのデータノードに対して，条件式を満たすかどうかを確認し，満たした場合にのみそのノードに検証済みトランザクションを転送し，自身の接続しているマイニングノードへもトランザクションを転送している．(step d) では，(step b) - (step c) を繰り返すことで，マイニングノードを含めた全てのノードへトランザクションを伝播する．(step e) では，トランザクションを受け取ったマイニングノードが，接続しているクラス

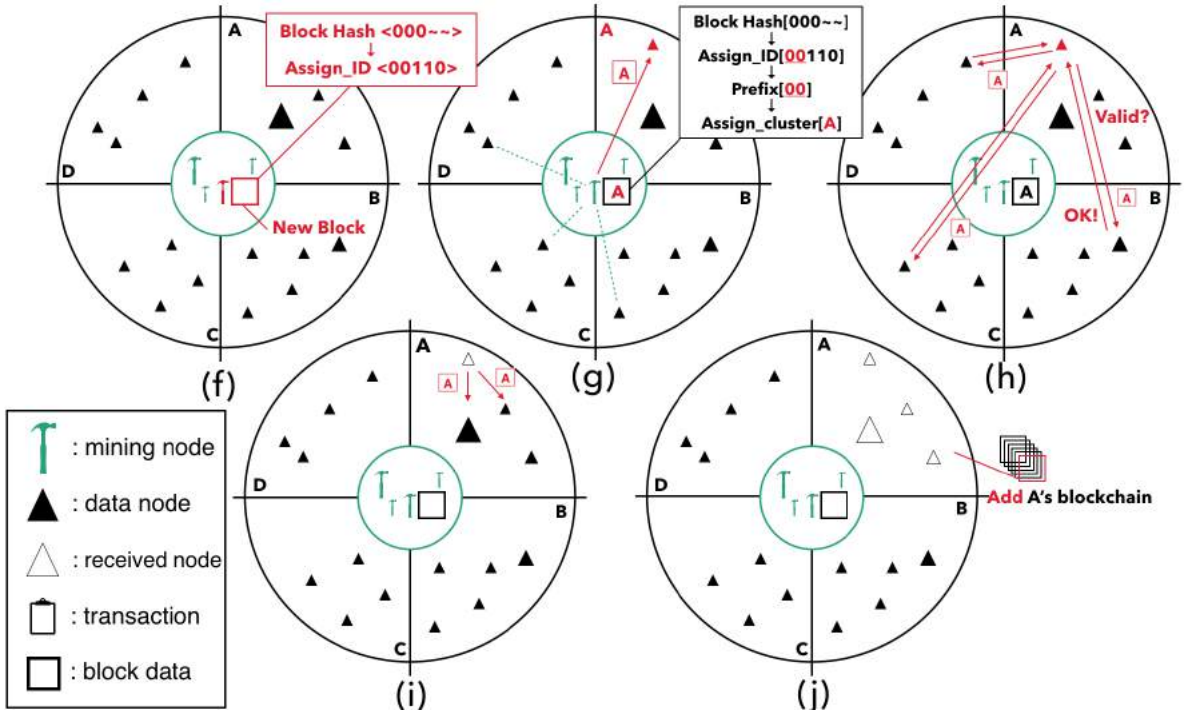


図 4.9. ブロック伝播プロセスの手順.

タ A，B，C，D に BC 参照を行い，トランザクションの有効性を検証した後，自身の生成するブロックへそのトランザクションを追加する．

(step f) では，あるマイニングノードが生成したブロックのハッシュ値を再び，ハッシュ計算を行うことで判別 ID⟨00110⟩ が生成された．(step g) では，生成したブロックの判別 ID⟨00110⟩ は，プレフィックス ⟨00⟩ を持つため，そのブロックは A クラスタへ伝播する．マイニングノードは自身の接続ノードからクラスタ A のノードへブロックデータを送信する．このとき，クラスタ B，C，D の接続ノードへブロックヘッダを送信する．(step h) では，ブロックデータを受信したクラスタ A のノードは，自身の接続ノードの中からクラスタ B，C，D のノードへ BC 参照を行う．(step i) では，ブロックデータを受信したクラスタ A のノードは自身の接続ノードの中から，クラスタ A に所属するノードにのみ転送を行う．(step j) では，クラスタ A 内にブロックデータが伝播されることで，クラスタ A のノードが全て新しく生成されたブロックを自身のクラスタのノード保有するブロックチェーンに付加している．

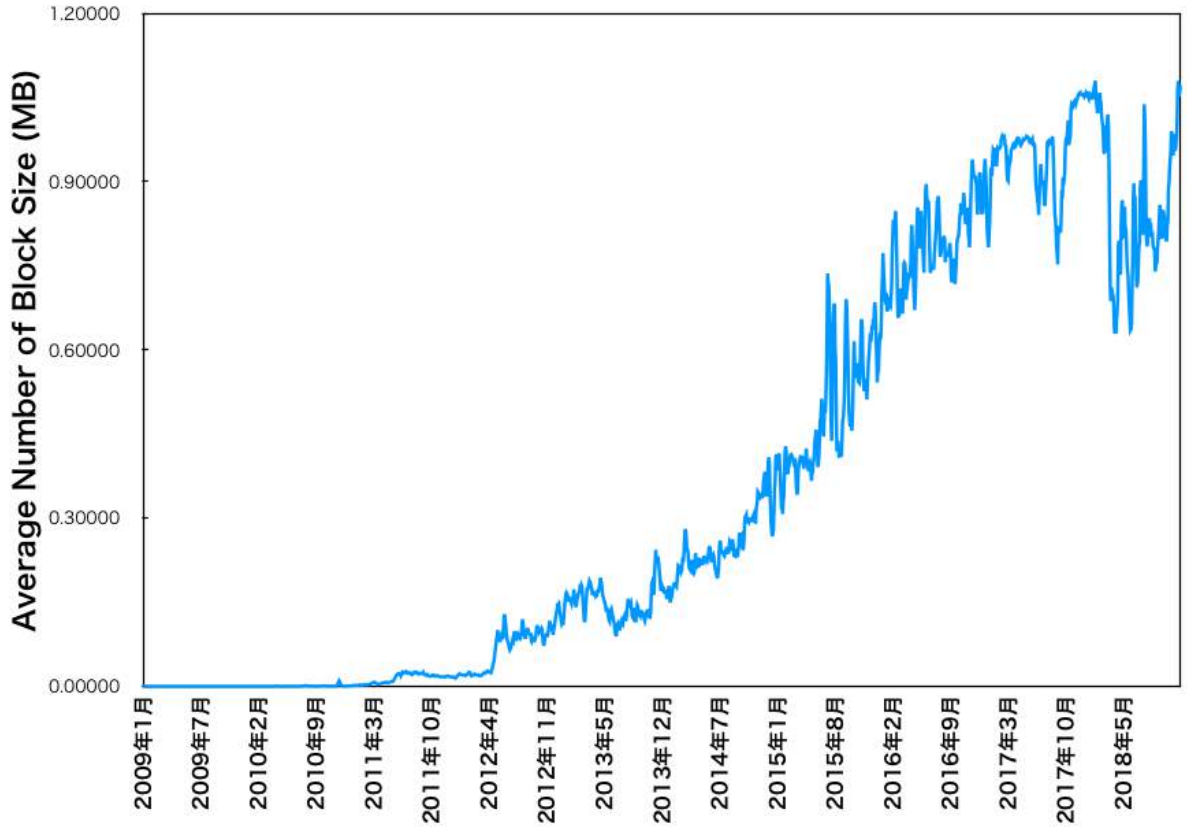


図 4.10. ビットコインにおける 24 時間平均のブロックのデータ容量の推移 [57].

4.7　提案手法の利点と欠点

4.7.1　提案手法の利点

図 4.10 は，ビットコインが 2009 年にデプロイされてから現在までのブロックチェーン全体のデータ容量の実測値を示しており，図 4.11 は同様に 24 時間平均したブロックのデータ容量の実測値を表している [57]．ビットコインの仕様上，ブロックサイズ上限は 1MB となっており，図 4.10 における 2018 年 1 月及び 2018 年 11 月には，ブロックのデータ容量が 1MB に到達している．このことから早急にビッグブロック等のスケーラビリティの改善を行う必要があることがわかる．また，図 4.11 を見ると，ブロックチェーンのデータ容量は 2018 年 11 月において約 190GB に到達している．ビックブロックによってスケーラビリティの改善を行なった場合，ブロックサイズ上限をどこまで引き上げるかに比例してブロックチェーンのデータ容量の増加速度が増加することが推測される．ビックブロックがネットワークに与える影響は，文献 [27] に詳細に述べられている．以上のことから，一般的な PC のストレージサイズの定義にもよるが，ビックブロックを施行するかどうかに関わらず，今後，ストレージのハー

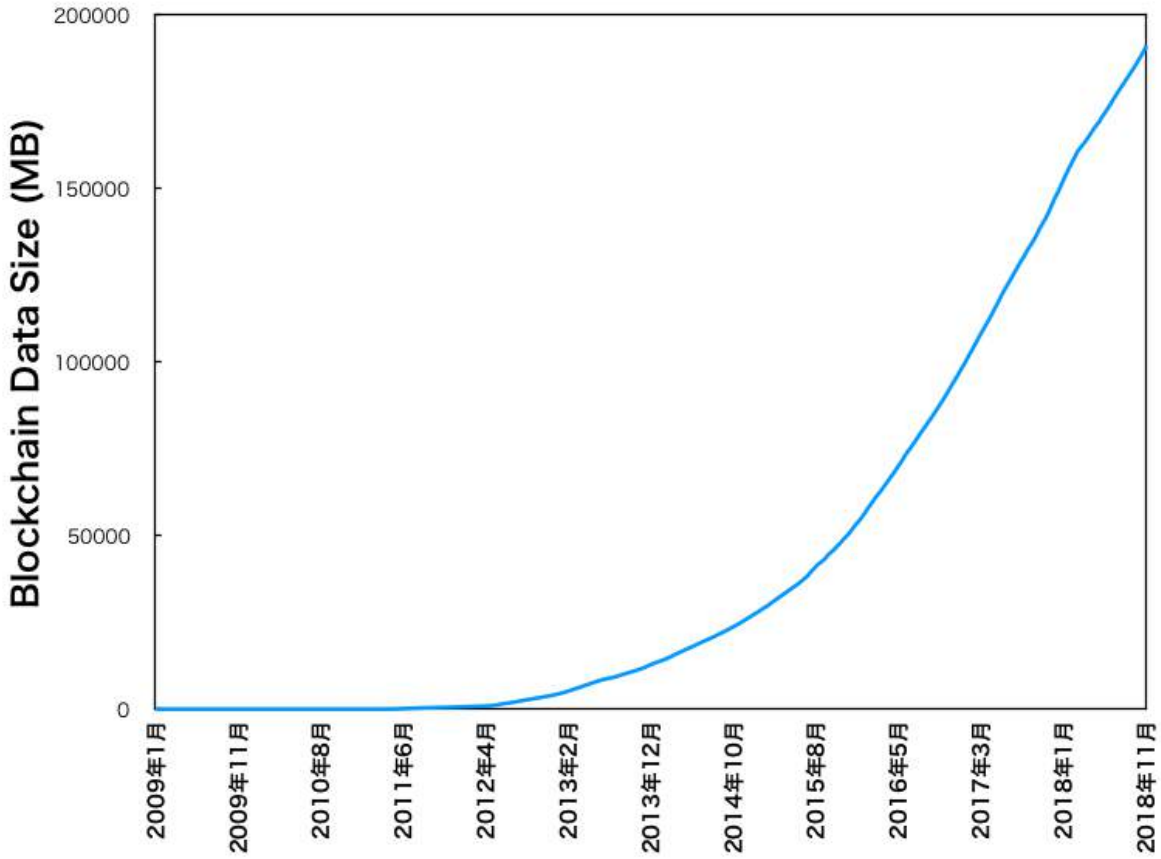


図 4.11. ビットコインにおけるブロックチェーンのデータ容量の推移 [57].

ドウェア要件を満たせずに離脱してしまうノードが増え，非中央集権性を損なう可能性が指摘されている．提案手法では，ブロックチェーンを構成するブロックの総数をプレフィックスの桁数によって設定したクラスタ数で分配するため，1 データノードの保有するブロックチェーンのデータ容量はブロックの総数をクラスタ数で割ったものとして推測される．また，ブロックの判別 ID をもとに配属するクラスタを特定するため，ハッシュ関数の特性上どこかのクラスタにブロックが他のクラスタよりも多く割り当てられる可能性は長期的に少ないと考えられる．以上のことから，提案手法ではビックブロックによってブロックチェーンデータ容量の増加速度が増加した場合にも，ストレージのハードウェア要件を抑えることが期待できる．

次に，提案手法では，DHT ネットワークを用いることによって，効率的にブロードキャスティングを行えるため，ネットワーク負荷を減らすことができる．提案手法ではブロードキャスティングの効率化のために DHT ネットワークを用いているが，ネットワーク維持の観点と ID ツリーの応用可能性から Kademlia を採用している．Kademlia では，他の DHT に比べノードのネットワークへの参加退出に対してネットワーク維持のためのメッセージを必要としていないため，維持コストが低い．また，ノード ID ツリーは，文献 [22] のように，ブロック

チェーンのデータを持つノードに対してインセンティブを与えるようなプロトコルを考案するとき，そのデータノードを特定する必要があるため，ID ツリーを用いることによって応用できる可能性があることが Kademlia を用いる利点として挙げられる．

最後に，これらストレージコストの改善とデータ送信の効率性の 2 点を第三者介入や，P2P ネットワークにおける中間管理者 (スーパーノード) なしに，達成できる．そのため，ビックブロックやブロック生成間隔の短縮等のスケーラビリティの改善を行なった場合でも，一般のネットワーク参加者が不利にならずに，ブロックチェーンの重要な観点である非中央集権性を保ったままスケーラビリティを向上できることが提案手法の最も大きな利点となっている．

4.7.2　提案手法の欠点

一方，提案手法では BC 参照のプロセスを頻繁に行う必要があり，これがブロードキャストの効率性を低減させる可能性がある．また，全てのノードが 1 つのブロックチェーンを共有，更新しているビットコインに比べ，他のクラスタのノードを信頼しなければならない点で，安全性が低下する可能性がある．加えて，各々のクラスタにはノードが一時的に極少数になってしまう状況にならないような十分数のノードが存在していることを前提条件としている．あるクラスタ内のノードが全て停止してしまうような状況では，BC 参照を行うことができず，ブロックチェーンの一部が消失してしまうことになる．このような観点から，提案手法は安全性を下げることによって非中央集権性とスケーラビリティを考慮している手法だと言える．また，欠点として，従来手法のシンプルな設計にすることで安全性を高めているビットコインに比べ，システムが複雑で実装が難しいという点が挙げられる．トランザクション処理システムでは，様々なネットワーク攻撃方法の考案やシステム上のバグを利用等の不正を働くことによって，直接的に利益をあげることができるため，不正を働くインセンティブは強い．そのため実装にはより多くの検証や，不正対策を行いシステムの信頼性をあげる必要がある．

第 5 章

シミュレーション評価

5.1　評価項目とシミュレーションモデル

本章では，提案方式の有効性をシミュレーションによって評価を行う．シミュレーションはトランザクションの伝播プロセスとブロックの伝播プロセスの 2 つのシミュレーションモデルで行う．本来の提案手法の ID 空間は 256 bit であるが，シミュレーションではデータノード数に相当する数の ID が存在し，ID ツリーの全ての ID がネットワークのデータノードに付与されているものとする．また，これらのシミュレーションには，離散事象イベントシミュレーションを用いる．まず，最初のデータ送信ノードによるイベントが発生し，そのデータの受信ノードによるイベントが全て行われた後，さらにそのイベントで受信したノードによるイベントへと移行する．本章では，物理ネットワークを考慮せず，オーバレイネットワーク上のみのシミュレーションを行う．そのため，トラヒック量やデータの転送時間は全てのノード間で同じ値をとるものとして仮定し，また，輻輳やトラヒック遅延の発生等は考慮しない．

比較評価のための従来手法は全て 2 章 1 節で述べた非構造型のピュア P2P ネットワーク上で稼働するビットコインの仕様を元にしたモデルを用いる．従来手法モデルでは，非構造型のピュア P2P ネットワーク上で，ランダムネットワークモデルを用いる．提案手法と同様に離散事象のシミュレーションモデルを用い，送信手法はフラッディング手法となっている．従来手法におけるフラッディングとは，各ノードが持っている全ての近接ノードへ送信する送信手法である．ビットコインでは，ノード ID が存在しないが，従来手法ではネットワーク全てのノードに ID を付加し，近接ノード数分の接続をネットワーク全体からランダムに選択している．従来手法では，クラスタやバケットの概念が存在しないため，近接ノード数等のパラメータを提案手法との比較のために提案手法と同値に調整したものを用いる．

提案手法において，フラッディング，4 章 3 節で述べた MFFF 手法，RRL 手法を比較評価に用いる．提案手法におけるフラッディングとは，各ノードの所有する k-buckets に入っている全てのバケットのノードへ送信を行う送信手法である．シミュレーションにおけるパラメー

タを表 5.1 に示す．ノード数は 8192 とし，ID 空間は 13 bit で形成される．そのため，提案手法におけるノードリストのバケット数は 13 となり，接続ノード数はノードリストの各バケットに格納される最大の個数 k のによって決められるため，K=1 のとき 13，K=2 のとき 25，K=3 のとき 36 となる．ブロック伝播プロセスのシミュレーションでは，マイニングノードから最初のブロックの伝播を行うときの各クラスタへの接続ノード数は 1 としている．

今回のシミュレーションにおいて，発生するメッセージについて述べる．クエリメッセージは，転送先ノードが送信しようとしているデータを保有しているかどうかを確認するメッセージであり，転送先ノードはそのデータを持っているかどうかの応答メッセージを返す．トランザクションやブロックのデータ送信時のメッセージには，データを保有していない応答をしたノードへデータ送信を行うメッセージと BC 参照メッセージの 2 種類ある．データ送信メッセージは，データを保有していない応答メッセージを返したノードに対して，データを送信するメッセージである．また，BC 参照メッセージにおいては，自分のクラスタ内の保有するブロックチェーンを確認し，検証を行なった後，有効かどうかの応答メッセージを返す．提案手法におけるブロック伝播シミュレーションではこれらに加えて，ブロックヘッダを保有しているかどうかを確認するメッセージとその応答メッセージと，保有していない応答のメッセージを返したノードへブロックヘッダを送信するメッセージが発生する．総メッセージ数の評価では，これらのメッセージの合計値を計測している．

評価項目は，総メッセージ数，到達率，冗長率，平均ホップ数，データ送信メッセージ数である．到達率とは，メッセージの受信ノード数がネットワーク全体の総ノード数で割った指標である．また，メッセージ冗長率とは，ネットワーク全体のクエリメッセージとその応答メッセージにおいて，冗長であったメッセージ数を BC 参照メッセージとその応答メッセージの総数で割った指標である．また，平均ホップ数は，全ての受信ノードの持つ，自身へのホップ数の総ノード数で割った平均値である．2 つのシミュレーションモデルでは，メッセージの到達率が 100% に満たない場合を除いて，全てのノードの伝播を終えたとき，これらの指標を算出している．

パラメータ	値
ノード数	8192
ID 空間	2^{13}
クラスタ数	2 - 16
ノードリストにおける接続ノード数	13 - 57
K: k-buckets の各バケットの格納できる最大ノード数	1 - 5
マイニングノードの持つ各クラスタへの接続数	1

表 5.1. シミュレーション評価におけるパラメータ.

5.2　提案手法のトランザクション伝播シミュレーション評価

本節では，トランザクション伝播のシミュレーション結果について述べる．

5.2.1　リンク数別評価

図 5.1 はリンク数別の総メッセージ数の評価である．ここでのリンク数とは各ノードの k-buckets の全てのバケットに入っている近接ノードへの総リンク数のことである．また，従来手法におけるリンク数は，各ノードが持っている近接ノード数への総リンク数とした．比較対象は，提案手法におけるフラッディング (Pro1_flooding)，MFFF 手法 (Pro2 _MFFF)，RRL 手法 (Pro3_RRL) と，従来手法におけるフラッディング (Con_flooding) である．また，提案手法では，実線をクラスタ数 4，波線をクラスタ数 2 としている．従来手法，提案手法に関わらずフラッディングでは，リンク数が増えると総メッセージ数が他の手法に比べ増加傾向にあることが確認できる．続いて，MFFF 手法，RRL 手法という順で，総メッセージ数の増加傾向が低いことが確認できる．また，提案手法においてクラスタ数の増加に伴い，総メッセージ数が増加することが確認できるが，これは，データ検証プロセスの際，提案手法のみに発生する BC 参照メッセージが増加するためである．

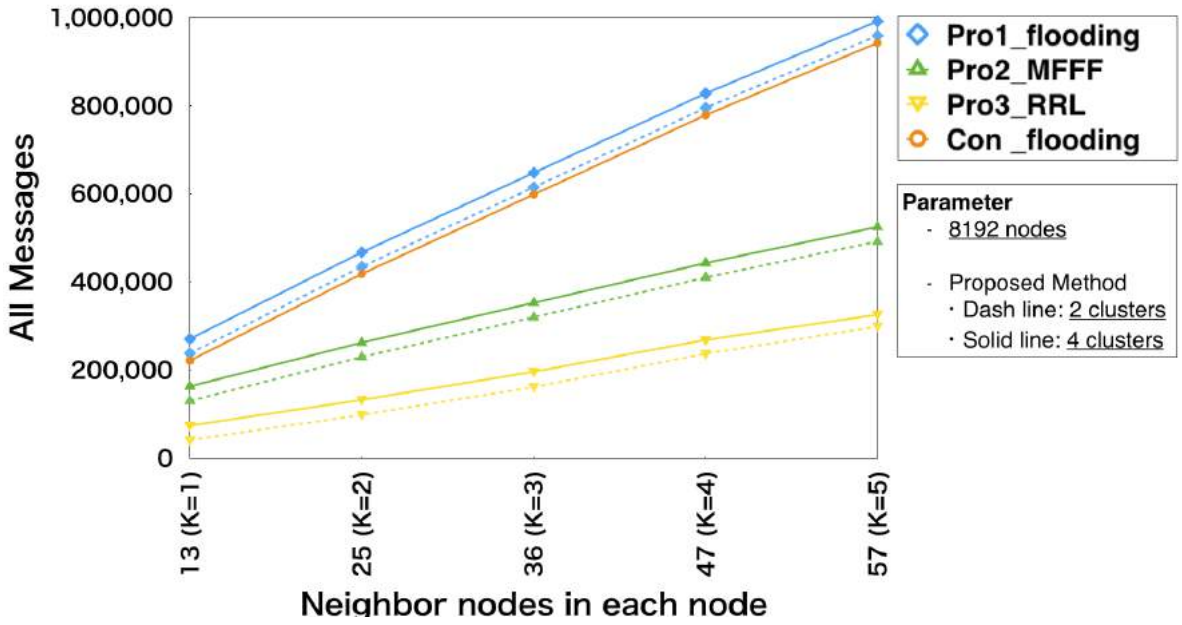


図 5.1. トランザクション伝播におけるリンク数別総メッセージ数評価.

図 5.2 はリンク数別の冗長率，また，図 5.3 はリンク数別の平均ホップ数の評価である．トランザクション伝播において，冗長率および平均ホップ数はクラスタ数の影響がほとんどないことが確認できる．これは，提案手法において，トランザクションの転送先が全てのデータノードとしているであるためである．冗長率は，フラッディング，MFFF 手法，RRL 手法の順で，減少していることが確認できる．従来手法と提案手法において，冗長率において差は見られなかった．また，RRL 手法では，接続ノード数が 13 (K=1) とき，冗長率が 0% であることが確認できる．これは，RRL 手法における送信可能な範囲に各バケットに存在する 1 つの近接ノードが対応し，重複を起こらないためである．K=2 のときでは，冗長率は 78% となり，各バケットに存在する 2 つの近接ノードによって RRL 手法における送信可能な範囲内で重複が発生することが分かる．

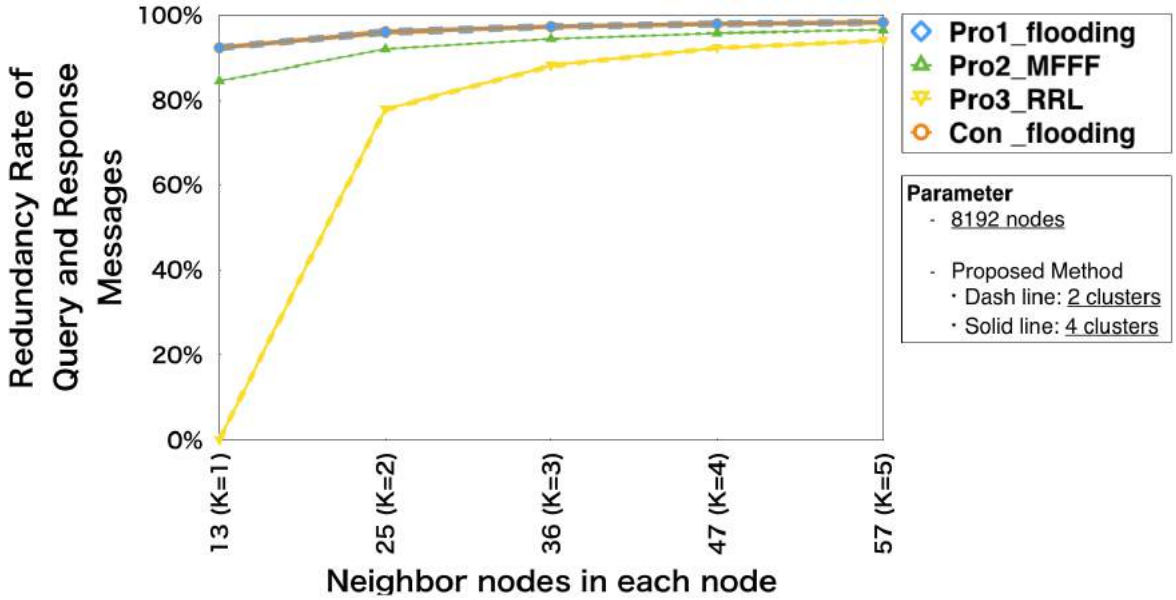


図 5.2. トランザクション伝播におけるリンク数別冗長率評価.

一方，平均ホップ数の評価では，近接ノード数が 13 (K=1) のとき，RRL 手法では 6.5 となり，このように RRL 手法は他の手法に比べ，平均ホップ数が増加してしまうことが分かる．ここから，RRL 手法の論理的な送信手法を用いることにより，冗長率と平均ホップ数がトレードオフの関係になっていることが推測できる．また，近接ノードの増加に伴い，従来手法の平均ホップ数に収束していることも確認できる．ここから，従来手法のフラッディングがホップ数の評価において最短のホップ数となることが分かる．

5.2.2　ホップ数別評価

トランザクションの過程を測定するために，ホップ数別の評価を行なった．ホップ数別評価では，8192 ノード，近接ノード数が 13 (K=1)，クラスタ数が 4 のパラメータで評価を行なっている．評価項目は，クエリ要求メッセージとその応答メッセージの冗長率と到達率である．図 5.4 に冗長率，図 5.5 に到達率を示す．提案手法は，従来手法に比べ 1 ホップ早くメッセージの重複が発生し始めている．これは，提案手法における最初の送信ノードからの XOR 距離が近い範囲内では，転送ノードからの距離が近いバケットにおいて重複が起こりやすいためで

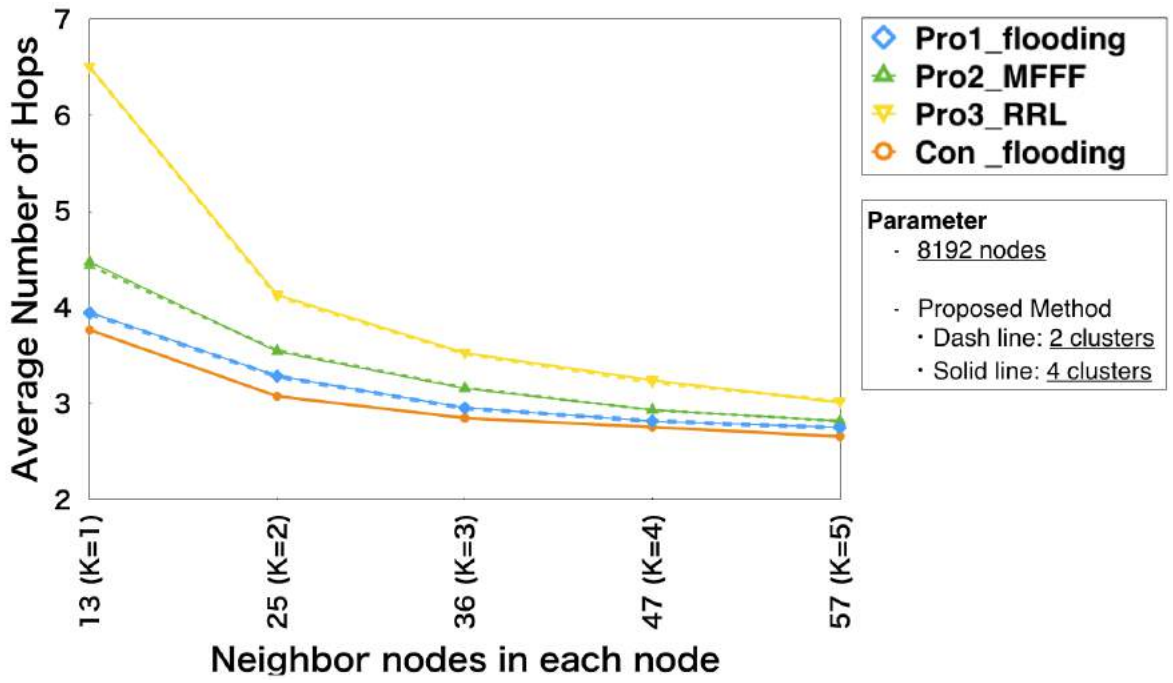


図 5.3. トランザクション伝播におけるリンク数別平均ホップ数評価.

ある．また，フラッディングについて比較すると，従来手法は，論理的な構成を持たないため，ある程度メッセージが到達している状態から，重複が発生しやすくなることが分かる．一方，提案手法では，遠い XOR 距離のノードに対しては，バケットが広範囲になっており，これが重複を発生しずらくさせている．

到達率は，RRL 手法が最も緩やかに増加することが分かる．これは論理的な構成から，転送するノードを効率よく発見できる代わりに 1 ホップでの送信回数が減り，より XOR 距離に転送するまでに時間がかかるためである．MFFF 手法では，唯一到達率が 100% にならないことが確認された．これは，より遠くの XOR 距離に転送するため一度通り過ぎた距離のノードには転送されないことがあるためである．

5.2.3　クラスタ数別評価

図 5.6 にクラスタ数別のデータ送信メッセージ数評価を示す．データ送信メッセージとはトランザクションやブロックのデータを送信するメッセージのことで，4 章 1 節で述べたように提案手法において，BC 参照の際に他のクラスタのノードへ転送されてきたデータを送信するメッセージと，要求メッセージがきたノードへデータを送信するメッセージの 2 種類ある．クラスタ数評価において，提案手法の比較対象として，データ送信メッセージ (Data sending_proposed) とその構成要素である BC 参照メッセージ (BC reference) とトランザク

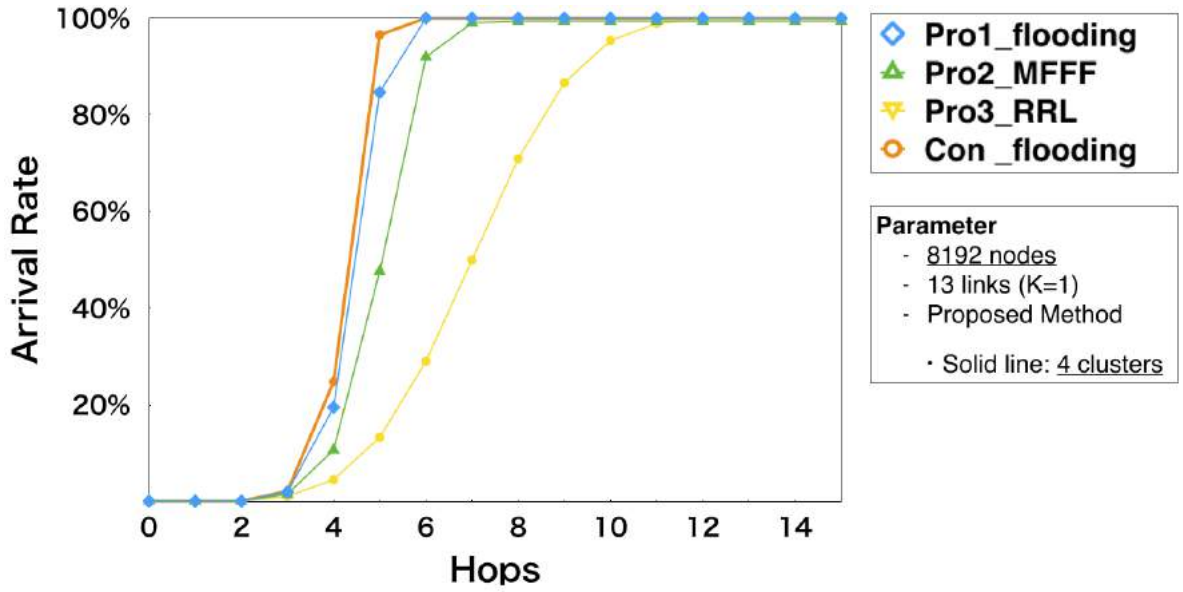


図 5.4. トランザクション伝播におけるホップ数別冗長率評価.

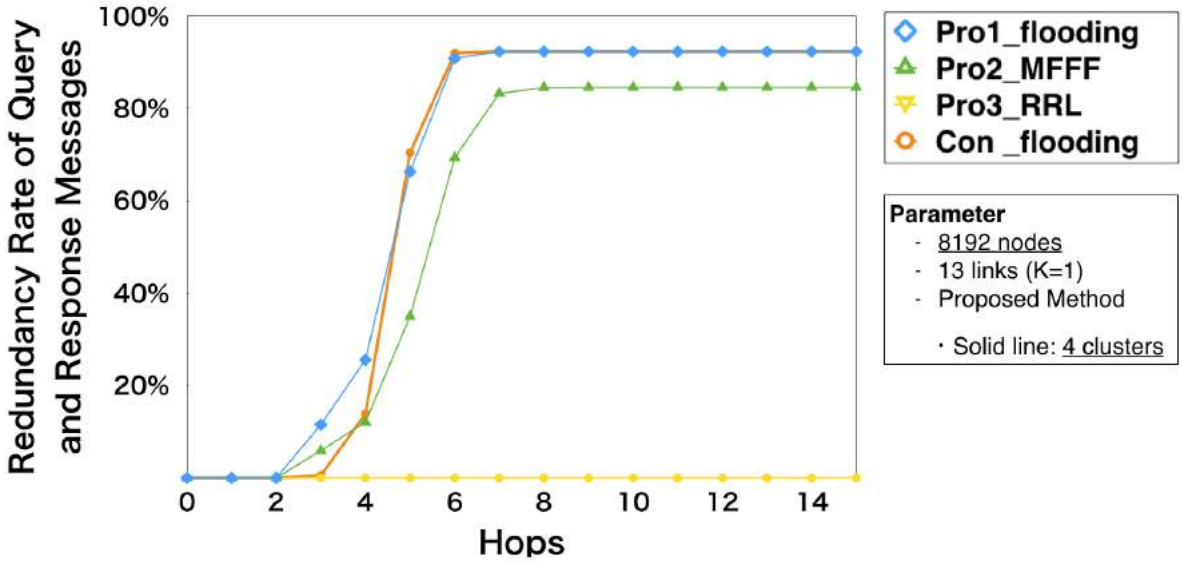


図 5.5. トランザクション伝播におけるホップ数別到達率評価.

ション送信メッセージ (Tx sending) に分けて，評価を行なっている．また，従来手法での比較対象は，データ送信メッセージ (Data sending_conventional) のみである．また，この評価では，トランザクション伝播において送信手法に影響がないため，RRL 手法のみを用いている．

図 5.6 から，提案手法におけるデータ送信メッセージの大半を BC 参照が占めていることが分かる．また，トランザクションを転送するときは要求するノードにしか転送しないため，トランザクションの送信メッセージ数は，ノード数分しか発生しないため，一定となっている．

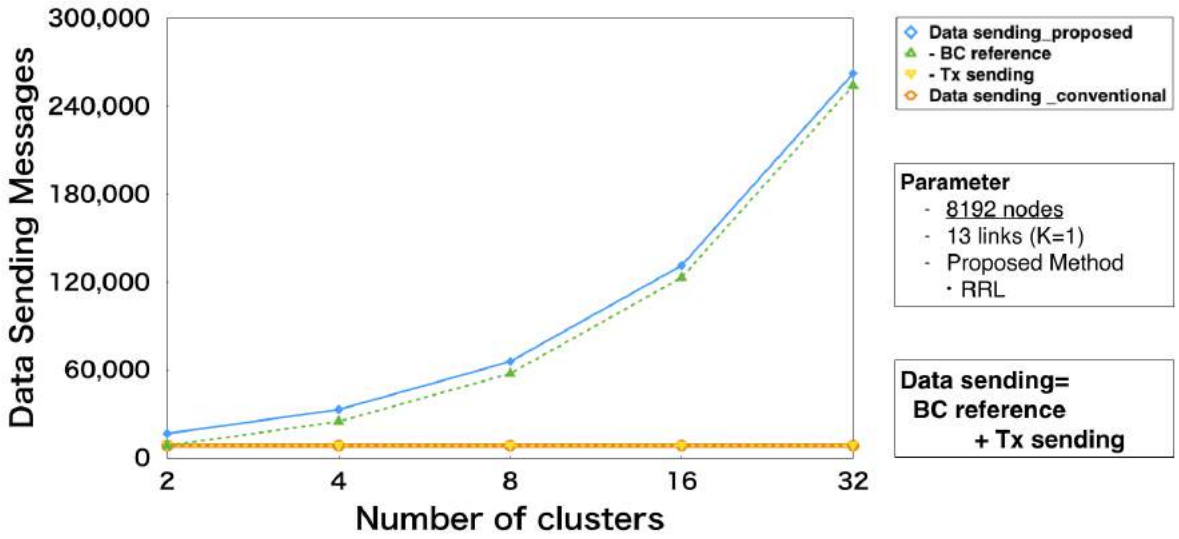


図 5.6. トランザクション伝播におけるクラスタ数別データ送信メッセージ数評価.

5.3　提案手法のブロック伝播シミュレーション評価

この節では，ブロック伝播のシミュレーション結果について述べる．

5.3.1　リンク数別評価

ここでは，リンク数別評価について述べる．MFFF 手法では，リンク数に関わらず到達率が平均 40 - 60% であったため，リンク数別評価において正確性に欠けるデータとなっている．

図 5.7 は，リンク数別の総メッセージ数評価である．総メッセージ数において，全ての提案手法が，従来手法より低い．また，トランザクション伝播とは異なり，クラスタ数が増えると，減少する傾向であることが確認できる．これは，ブロックを伝播する範囲が特定のクラスタのみなので，クラスタ数が増えるにつれていて，BC 参照メッセージが少なくなるためである．さらに，RRL 手法では，減少傾向の差は他の手法に比べて少ないことが確認できる．これは，重複を抑えている送信手法なので，ブロックの要求メッセージや応答メッセージの発生がより少なくなるためである．

図 5.9 にリンク数別の冗長率評価を示す．MFFF 手法を除けば，トランザクションと同様の結果になっている．

図 5.9 に，リンク数別の平均ホップ数評価を示す．リンク数別の平均ホップ数はトランザクション伝播と同様の傾向を示している．しかし，クラスタ数が増えると，トランザクション伝播とは異なり，平均ホップ数が減少すること確認される．これは，ネットワーク全体で，転送

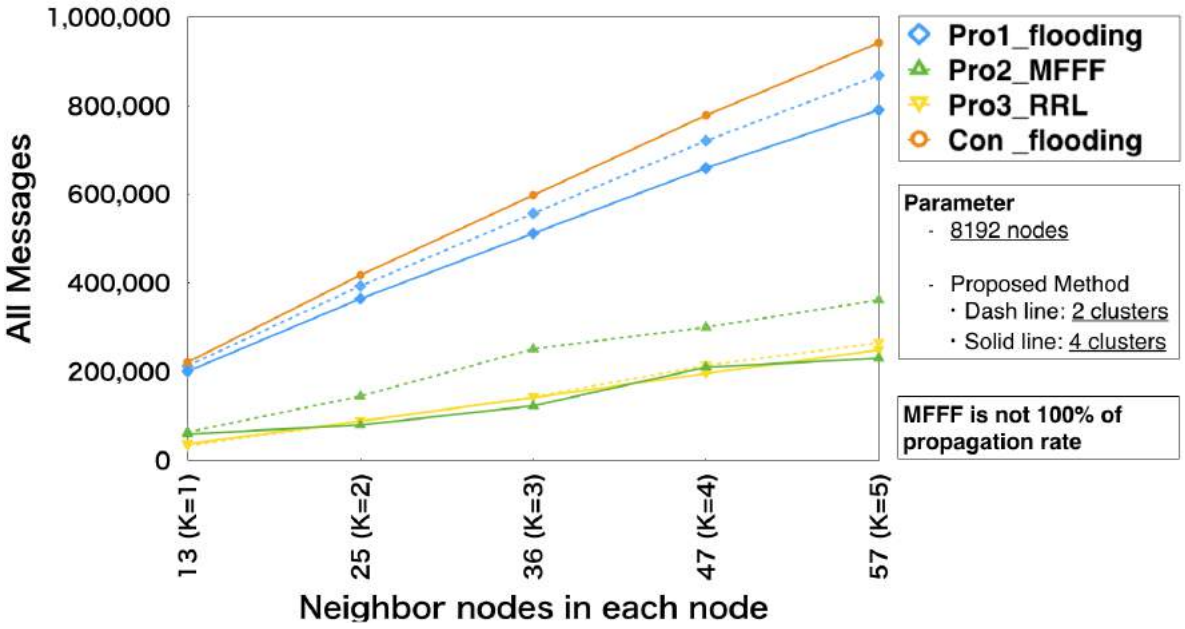


図 5.7. ブロック伝播におけるリンク数別総メッセージ数評価.

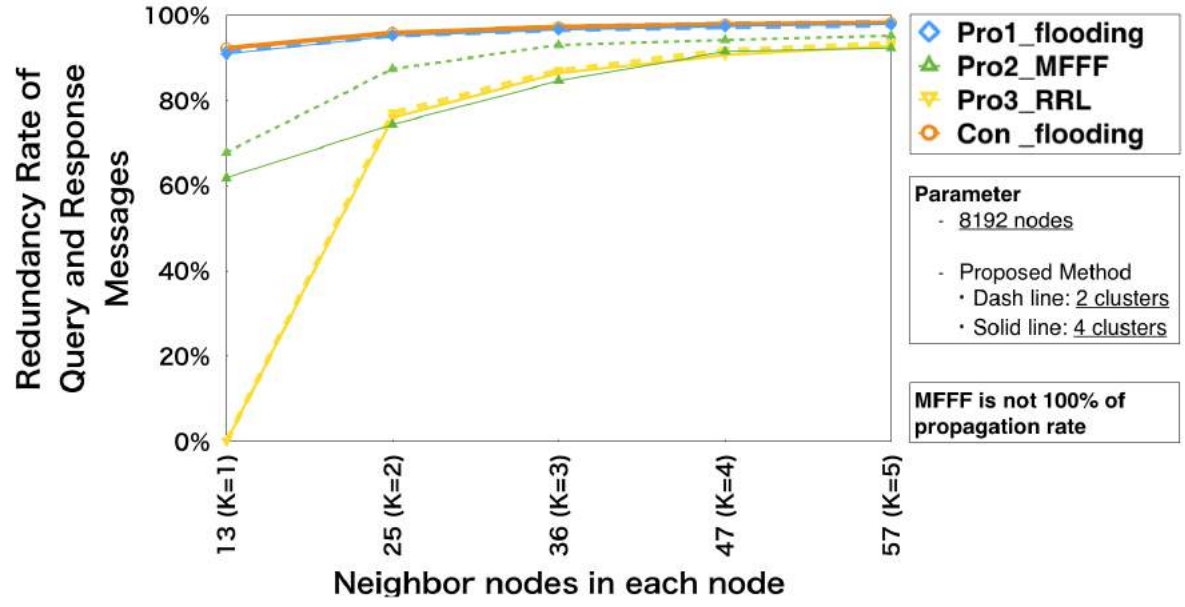


図 5.8. ブロック伝播におけるリンク数別冗長率評価.

すべきノードの数が特定クラスタのノード数のみであるためである.

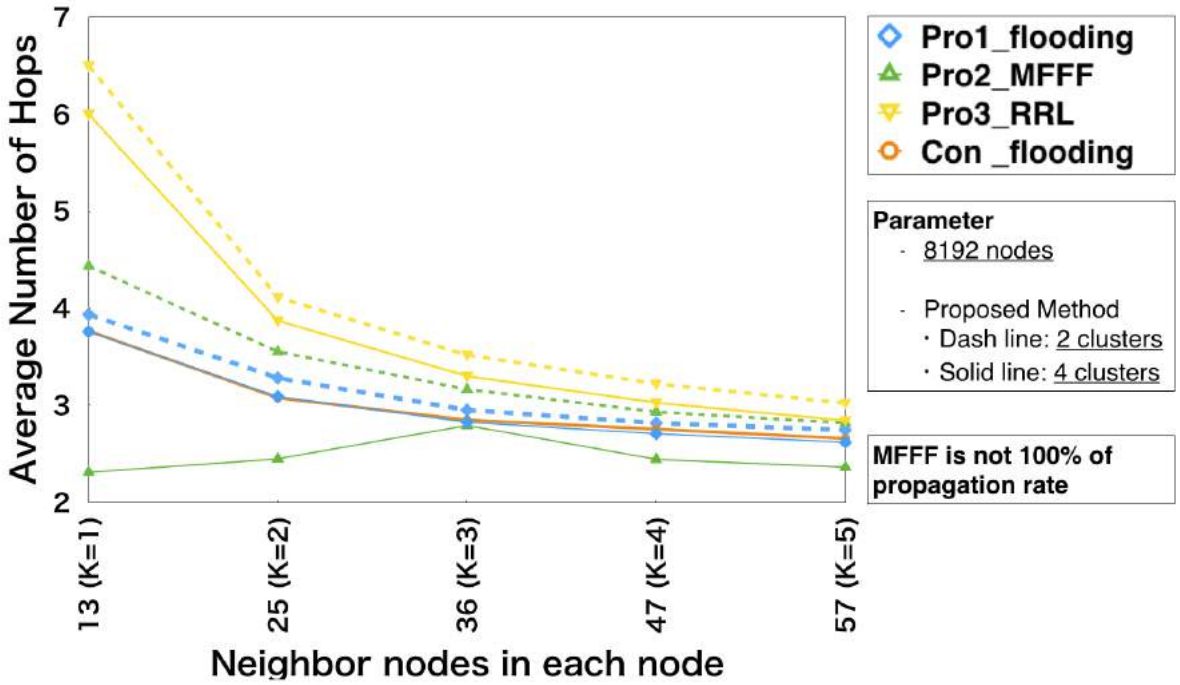


図 5.9. ブロック伝播におけるリンク数別平均ホップ数評価.

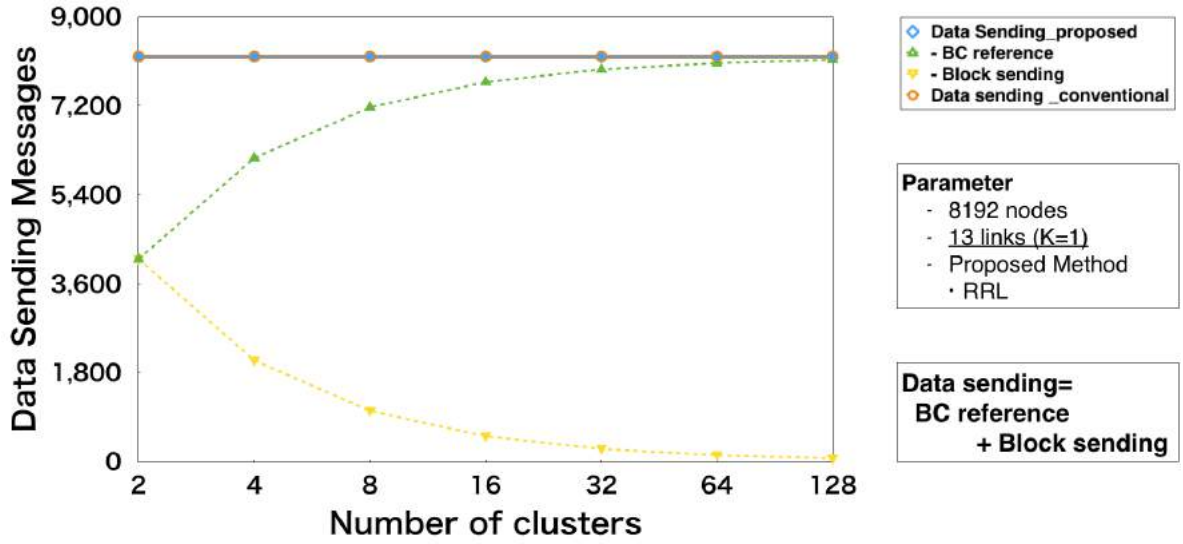


図 5.10. ブロック伝播におけるクラスタ数別データ送信メッセージ数評価.

5.3.2　クラスタ数別評価

図 5.10 では，トランザクション伝播と同様に，データ送信メッセージ数を評価している．データ送信メッセージ数の評価では，送信手法による影響はなく，ここでは，RRL 手法を用

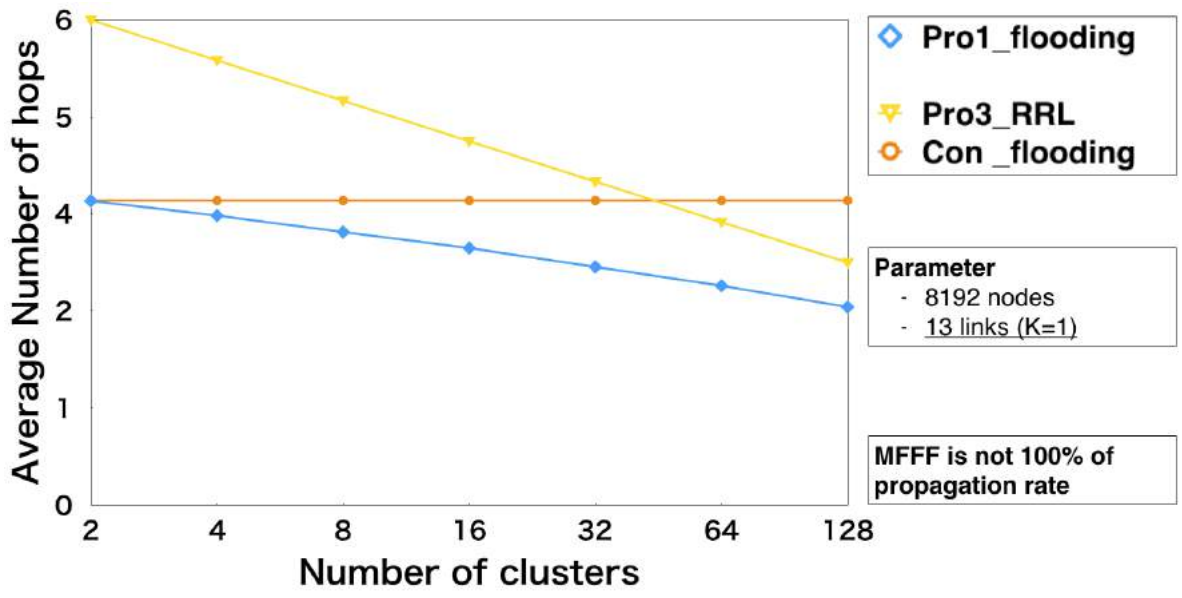


図 5.11. ブロック伝播におけるクラスタ数別平均ホップ数評価.

いている．ブロック伝播では，クラスタ数が増えるにつれて，データ送信を行うべきノードが減っていく分，BC 参照先のクラスタ数が増えるため一定の値となっている．データ送信メッセージは従来手法，提案手法ともにノード数の値を取っている．

K=1,2 のときのクラスタ別評価

RRL 手法では，冗長率，平均ホップ数において K=1 から K=2 にするときの変化率の幅が大きいことを考慮して，2 つのリンク数の場合でのクラスタ別評価を行う．また，MFFF 手法ではブロック伝播において，到達率が 100% にならないことから評価項目へ入れていない．

図 5.11，図 5.12 は K=1,2 のときのクラスタ数別の平均ホップ数評価である．K=1 で，クラスタ数が 64 に増加するとき，また，K=2 で，クラスタ数が 16 に増加するとき，RRL 手法は従来手法の平均ホップ数を下回る．これは，リンク数別評価と同様に，これは，ネットワーク全体で，転送すべきノードの数が特定クラスタのノード数のみであるためである．また，提案手法のフラッディングと RRL 手法を比較すると，平均ホップ数の減少率は K=1 の方が，K=2 のときよりも高いことが確認できる．これは，リンク数が多いほど，平均ホップ数が減少する傾向があるため，減少率が低下した．

図 5.13，図 5.14 は K=1,2 のときのクラスタ数別の冗長率評価である．クラスタ数を増やすにつれて，提案手法の冗長率が減少することが確認できる．K=1 の K=2 のときでは，RRL 手法の冗長率に大きな差がある．ここから，提案手法におけるバケットが 2 つ以上になった時に重複が急激に発生することが分かる．

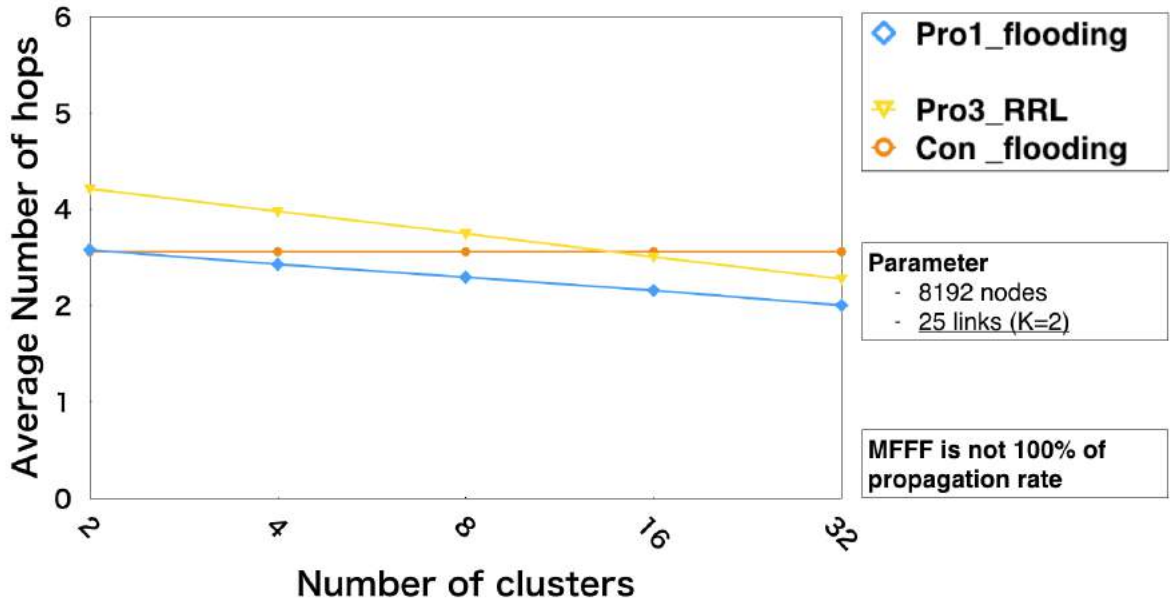


図 5.12. ブロック伝播におけるクラスタ数別平均ホップ数評価 2.

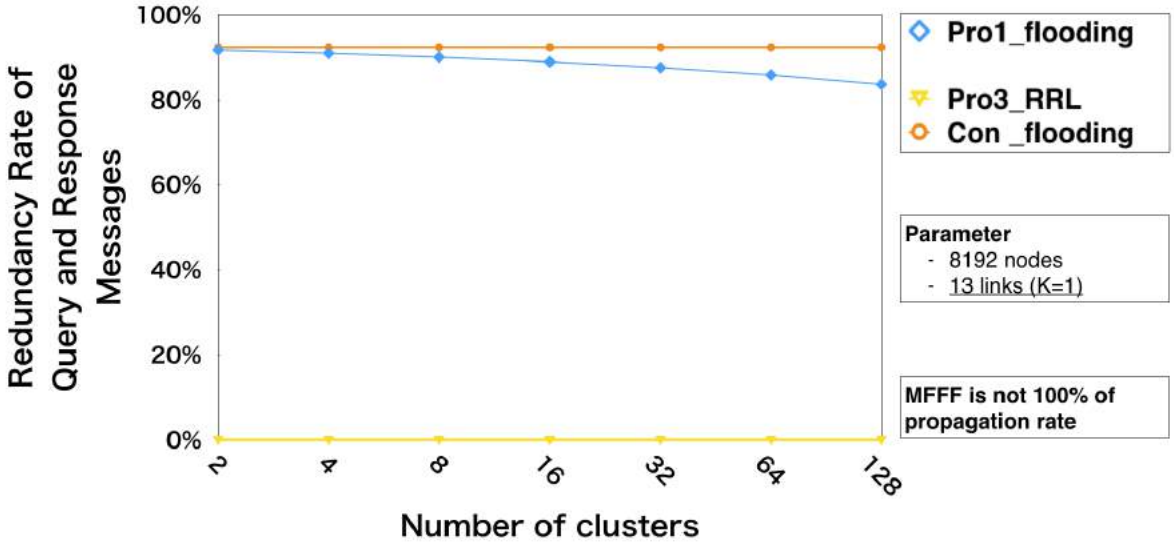


図 5.13. ブロック伝播におけるクラスタ数別冗長率評価.

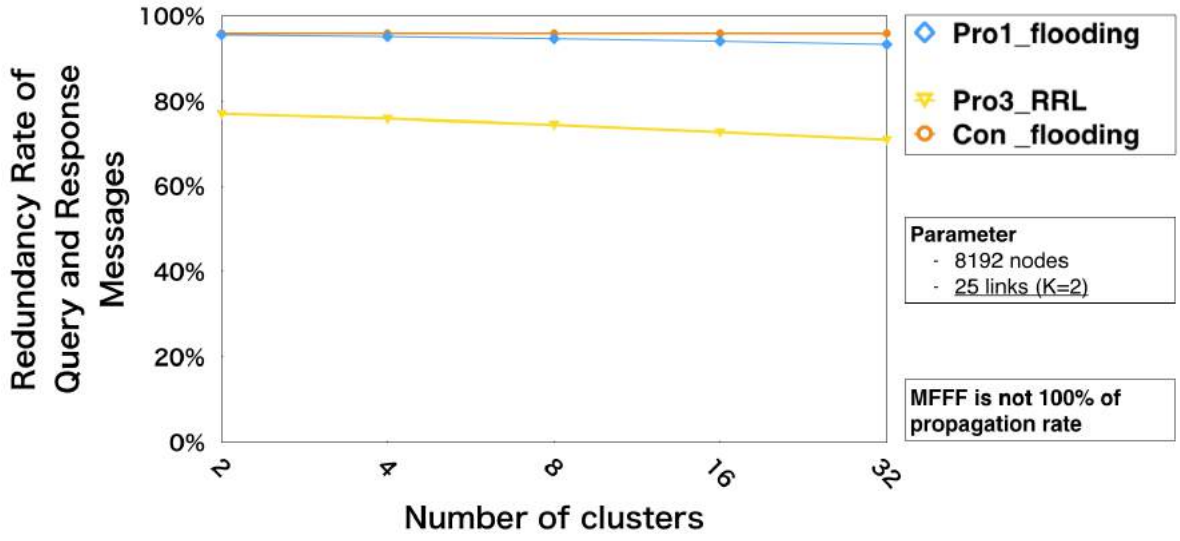


図 5.14. ブロック伝播におけるクラスタ数別冗長率評価 2.

第 6 章

結論

6.1　まとめ

現在，ビットコインではブロックサイズが 1MB となっており，このブロックサイズの上限を解放することでトランザクション処理能力を向上することができる．また，ブロック生成間隔を短くし，時間あたりのブロックの生成個数を増やすことでトランザクション処理能力を向上することができる．しかしながら，これらの制約条件の緩和は，処理能力は向上する代わりに，全ての参加ノードへのストレージ負荷やネットワーク負荷が増大することにより，個人単位の参加ノードが離脱してしまう．そして大規模マイナーやウォレットサーバ等の法人単位の参加ノードのみが生存し，ネットワーク維持のためプロセスが集中化してしまい, 中央集権化を招く恐れがある．本稿では，ブロックチェーンデータを保有するノードとマイニングを行うノードで役割分担を行い，ブロックチェーンを分散配備させ，DHT の一つである Kademlia による効率的なブロードキャストを行う手法を提案した．

また，トランザクション伝播とブロック伝播のシミュレーションを用いて，メッセージ数，要求応答メッセージに対する冗長率，平均ホップ数等のシミュレーション評価を行なった．トランザクション伝播において RRL 手法では，総メッセージ数，冗長率が最も低く，また，接続ノード数が 13 (K=1) のとき，冗長率が 0% になることを確認した．さらに，2 クラスタ，4 クラスタの場合の評価を行うことによって，クラスタ増加によって総メッセージ数が増加傾向にあることが分かった．反対に，RRL 手法の平均ホップ数は最も多く，また，クラスタ数の増加に伴い，BC 参照メッセージが指数関数的に増加することを確認した．トランザクション伝播の目的は，マイニングノードへ偏りなくトランザクションを転送することであり，伝播遅延の影響はあまり考慮しなくてよいが，トランザクションデータの送信メッセージ数の増加や，平均ホップ数の増加がシステム全体に与える影響を考慮しなければならないことがわかった．リンク数別評価では，総メッセージ数や平均ホップ数はクラスタ増加によって，トランザクション伝播とは異なり，減少傾向にあることが分かった．まず，クラスタ数別評価では，デー

タ送信メッセージ数がクラスタ数の増加に伴い，BC 参照先のクラスタ数が増え，伝播すべきクラスタのノード数が減ることによって一定となることを確認した．また，クラスタ数の増加に伴い冗長率，平均ホップ数の減少傾向にあることが確認された．接続ノード数 13 (K=1) において，クラスタ数が 64 のとき RRL 手法が従来手法より平均ホップ数を下回り，接続ノード数 25 (K=2) において，クラスタ数が 16 のとき RRL 手法が従来手法より平均ホップ数を下回ることが分かった．以上の点から，データの要求応答メッセージにおける冗長率や総メッセージ数等のネットワーク負荷，ブロック伝播における特定のクラスタ数以上のときの平均ホップ数において有用性が確認できた．

6.2 今後の課題

提案手法の改善点として，DHT ネットワークにおける効率的なブロードキャスティング手法と伝播プロセスが挙げられる．DHT ネットワークにおける効率的なブロードキャスティング手法に関する研究は多く存在し，提案手法における RRL 手法で K=2 よりも大きい時であっても，手法の改善次第で冗長率を下げられる可能性がある．また，従来研究との比較が行えていないため，今後行う必要がある．トランザクション伝播プロセスでは，マイニングノードへ偏りなくトランザクションを伝播することが必要であったため，データノード全てにトランザクションを伝播する方法をとった．しかし，これが冗長なデータ送信メッセージの増加に影響し，ネットワーク負荷を下げる可能性が高いことが分かった．そのため，今後は，セキュリティの観点から，マイニングノードが不正を起こせないようにするためのトランザクション伝播の範囲を特定する必要がある．さらに，本稿ではセキュリティの観点においての評価がなされていないため，様々な攻撃に対する耐性や改ざん可能性等のセキュリティの評価を行う必要がある．

また，文献 [22] のように，マイニング報酬が枯渇することで，セキュリティに影響する可能性が指摘されており，参加ノードがブロックチェーンデータを保有すること自体にインセンティブを与える設計が注目されている．本稿の提案手法では，ノード ID を付加しているため，ブロックチェーンデータの保有にインセンティブを与える応用が考えられるため，このような設計を考慮したシステムの考案できる可能性がある．

本稿におけるシミュレーションでは，ホップ数の指標を元に評価を行なっているため，トラヒックについて考慮しておらず，伝播遅延についての評価結果が得られていない．特に，ブロックの伝播遅延の発生は，フォークの発生頻度を増加することでネットワークのセキュリティに大きな影響を与えるため，今後は，実装に向けて物理的なネットワークトポロジーも含めた，より現実的なシステム全体のシミュレーション評価が必要である．また，提案手法の DHT ネットワークにおける ID 空間に全てノードが参加している状況を前提として，シミュレーションを行なっているため，256 bit の ID 空間では，あるバケット内の入るノードの数が

極端に少なくなる場合が考えられる．そのため，今後は ID 空間をより，現実的なサイズに変えて評価を行う必要がある．

謝辞

本論文の作成にあたり日頃よりご指導頂きました朝香卓也教授，西辻崇助教にこの場を借りて深くお礼申し上げます．また，本研究を進めるにあたり多くの場面で助言を頂いた研究室の皆様に深くお礼申し上げます．

参考文献

[1] I. Lee, “Fintech: ecosystem and business models, ” U- and E- Service Science and Technology, pp. 57-62, 2016.

[2] NTT Data Management Institute and Y. Kenzo, “How is finTech used and where to head?,” http://www.keieiken.co.jp/pub/yamamoto/column/column_150901.html, (Access in the 2018-Sep-26), 2015.

[3] T. Nomakuchi, “A case study on fintech in japan based on keystone strategy,” 2018 Portland International Conference on Management of Engineering and Technology (PICMET), Honolulu, HI, USA, pp. 1-5, Aug. 2018.

[4] S. Nakamoto, “Bitcoin: a peer-to-peer electronic cash system,” www.bitcoin.jp, 2009. (accessed 2018-11-8)

[5] K. Christidis and M. Devetsikiotis, “Blockchains and smart contracts for the internet of things,” in IEEE Access, vol. 4, pp. 2292-2303, 2016.

[6] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, “Medrec: using blockchain for medical data access and permission management,” 2016 2nd International Conference on Open and Big Data, 22 Sep. 2016.

[7] K. Biswas and V. Muthukkumarasamy, “Securing smart cities using blockchain technology,”in 2016 IEEE 18th International Conference on High Performance Computing and Communications (HPCC/SmartCity/DSS), pp. 1392-1393, Dec 2016.

[8] Y. Zhang and J. Wen, “An IoT electric business model based on the protocol Of bitcoin,” in 2015 18th International Conference on Intelligence in Next Generation Networks, pp. 184-191, Feb. 2015.

[9] D. Schoder and K. Fischbach, “Peer-to-peer prospects,” Communications of the ACM, vol.46, pp.27-29, 2003.

[10] S. Androutsellis-Theotokis and D. Spinellis, “A survey of peer-to-peer content distribution technologies,” ACM Computing Surveys, vol.36, no.4 pp.335-371, Dec. 2004.

[11] J. Risson and T. Moors, “Survey of research towards robust peer-to-peer networks:

search methods," Computer Networks: The International Journal of Computer and Telecommunications Networking, vol.50, no.17, pp.3485-3521, Dec. 2006.

[12] アンドレアス･M･アントノプロス著, 今井崇也･鳩貝淳一郎訳 (2016),「Mastering Bitcoin ビットコインとブロックチェーン　暗号通貨を支える技術」,NTT 出版

[13] 赤羽喜治・愛敬真生編著,(2016)「ブロックチェーン　仕組みと理論　サンプルで学ぶ Fintech のコア技術」, リックテレコム

[14] https://coin.dance/nodes (accessed 2018-11-8)

[15] Y. Sompolinsky, and A Zohar, "Accelerating bitcoin's transaction processing fast money grows on trees, not chains," https://eprint.iacr.org/2013/881.pdf, 2013 (accessed 2018-10-29)

[16] U.Klarman, S.Basu, A.Kuzmanovic, and E.Sirer, "BloXroute: a scalable trustless blockchain distribution network whitepaper," http://www.blocroute.com, Mar. 2018. (accessed 2018-10-29)

[17] S. Popov, "The tangle", https://untangled.world/iota-whitepaper-tangle/ (accessed 2018-10-29)

[18] J. Poon and T. Dryja, "The bitcoin lightning network: scalable Off-chain instant payments," https://lightning.network/lightning-network-paper.pdf, Jan 14, 2016. (accessed 2018-10-29)

[19] A. Back, M.Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A.Miller, A. Poelstra, J. Timon, and P. Wuille, "Enabling blockchain innovations with pegged," https://blockstream.com/sidechains.pdf, Sep. 2014. (accessed 2018-10-29)

[20] P. Prihodko, S.Zhigulin, M. Sahno, and A.ostrovskiy, "Flare: an approach to routing in lightning network White Paper,"
https://bitfury.com/content/downloads/whitepaper_flare_an_approach
_to_routing_in_lightning_network_7_7_2016.pdf,
Jul. 2016. (accessed 2018-10-29)

[21] I. Eyal, A. E. Gencer, E. G. Sirer, and R. van Renesse, "Bitcoin-NG: a scalable blockchain protocol.'' http://arxiv.org/abs/1510.02037, Oct. 2015. (accessed 2018-10-29)

[22] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: extending bitcoin's proof of Work via proof of stake ," ACM SIGMETRICS Performance Evaluation Review, v.42 n.3, Dec. 2014.

[23] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," IEEE P2P 2013 Proceedings, Trento, pp. 1-10, Sep. 2013.

[24] A. Miller, and R.Jansen, "Shadow-bitcoin: scalable simulation via direct execution

of multi-threaded applications," In Proceedings of the 8th USENIX Conference on Cyber Security Experimentation and Test (CSET'15), Aug. 2015.

[25] A.Miller, J. Litton, A. Pachulski, N.Gupta, D.Levin, N.Spring, and B. Bhattacharjee, "Discovering bitcoin's public topology and influential nodes," https://www.cs.umd.edu/projects/coinscope/coinscope.pdf, 2015. (accessed 2018-11-8)

[26] L. Luu, V. Narayanan, C. Zheng, K.Baweja, S.Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16), Oct. 2016.

[27] BitFury Group, "Block size increase," http://bitfury.com/content/5-white-papers-research/block-size-1.1.1.pdf, Sep 06, 2015 (accessed 2018-10-29)

[28] M. Conti, S. K. E, C. Lal and S. Ruj, "A survey on security and privacy issues of bitcoin," in IEEE Communications Surveys and Tutorials, May. 2018.

[29] E. Heilman, A. Kendler, and A. Zohar, "Eclipse attacks on bitcoin's peer-to-peer network," In the Proceedings of the 24th USENIX Security Symposium, p.p.129-144, Aug. 2015.

[30] Y. Marcus, E. Heliman, and S. Goldberg, "Low-resource eclipse attacks on ethereum's peer-to-peer network," IACR Cryptology ePrint Archive, https://www.cs.bu.edu/ goldbe/projects/eclipseEth.pdf, Feb. 2018. (accessed 2018-10-29)

[31] 阿部達成，朝香卓也，"P2P センサーネットワークにおける適応的負荷分散法," 首都大学東京, 修士論文 pp.4 - 28, Feb.2016.

[32] B. Cohen, "Incentives build robustness in bittorrent, " Economics of Peer-to-Peer Systems, Brekeley, USA, May. 2003, http://bittorrent.com/.

[33] Freenet, http://de-co.info/freenet/, (accessed 2016-12-19).

[34] I. Clarke, O. Sandberg, B. Wiley and T. W. Hong, 'Ffreenet: a distributed anonymous information storage and retrieval system, " ICSI Workshop on Designing Privacy Enhancing Technologies, vol.2009, pp.46-66, July. 2001.

[35] I. Clarke, S. G. Miller, O. Sandberg and T. W. Hong, "Protecting free expression online with freenet, " IEEE Internet Computing, vol.6 no.1, pp.40-49, Jan/Feb. 2002.

[36] Napster, http://napster.com/, (accessed 2016-12-10).

[37] OpenNap, http://opennap.sourceforge.net/, (accessed 2016-12-10).

[38] S. Saroiu, K. P. Gummadi and S. D. Gribble, "Measuring and analyzing the characteristics of napster and gnutella hosts, " Multimedia Systems, vol.9, no.2, pp.170-184, Aug. 2003.

[39] 金子 勇「Winny の技術」 ASCII

[40] Wikipedia(online), available from
http://ja.wikipedia.org/wiki/Share, (accessed 2016-12-19).

[41] Gnutella, http://www.gnutella.com/, (accessed 2016-12-19).

[42] E. Adar, B. Huberman, “Free riding on gnutella, ” First Monday, vol.5-10, Oct. 2000.

[43] Skype, http://www.skype.com/, (accessed 2016-12-19).

[44] KaZaA, http://www.kazaa.com/, (accessed 2016-12-19).

[45] M. Sekine and K. Sezaki, “Adaptive sensor data management for frequencies of data update and query access in peer-to-peer network,” Proc. IEICE Trans. On Comm. Vol. J92-B, pp.400-412, 2009.

[46] P. Maymounkov, D. Mazieres, “Kademlia: a peer-to-peer information system based on the xor metric, ” Lecture Notes in Computer Science, vol.2429, pp.53-65, 2002.

[47] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, “A scalable content-addressable network,” ACM SIGCOMM, vol.31, no.4, pp.161-172, Oct. 2001.

[48] A. Rowstron and P. Druschel, “Pastry: scalable, decentralized object Location, and routing for large-scale peer-to-peer systems,” Lecture Notes in Computer Science, vol.2218, pp.329-350, Nov. 2001.

[49] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishman, “Chord: a scalable peer-to-peer lookup service for internet applications,” ACM SIGCOMM, vol.31, no.4, pp.149-160, Oct. 2001.

[50] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, “Tapestry: a resilient global-scale overlay for service deployment,” IEEE Journal on Selected Areas in Communications, vol.22, no.1, pp.41-53, Jan. 2004.

[51] M. Wahlisch, T. C. Schmidt and G. Wittenburg, “Broadcasting in prefix space: P2P data dissemination with predictable performance,” 2009 Fourth International Conference on Internet and Web Applications and Services, Venice, pp. 74-83, May.2009.

[52] A. D. Peris, J. M. Hern’ndez and E. Huedo, “Evaluation of the broadcast operation in kademlia,” 2012 IEEE 14th International Conference on High Performance Computing and Communication and 2012 IEEE 9th International Conference on Embedded Software and Systems, Liverpool, pp. 756-763, Jun. 2012.

[53] Z. Czirkos and G. Hosszu, “Enhancing the kademlia P2P network,” Periodica Polytechnica Electrical Engineering. 54. 87. 10.3311/pp.ee.2010-3-4.01, 2010

[54] Z. Czirkos and G. Hosszu, “P2P based instrusion detection,” Infocommunica tions Journal LXIV I, 3-10, 2009.

[55] S. E. Ansary, L. O. Alima, P. Brand, S. Haridi, “Efficient broadcast in structured

P2P networks" Kaashoek M.F., Stoica I. (eds) Peer-to-Peer Systems II. IPTPS 2003. Lecture Notes in Computer Science, vol 2735, 2003.

[56] F. Petrini and M. Vanneschi, "K-ary n-trees: high performance networks for massively parallel architectures," Proceedings 11th International Parallel Processing Symposium, Genva, Switzerland, pp. 87-93, 1997.

[57] https://www.blockchain.com/ja/charts (accessed 2018-11-21)