現実世界の条件に適応する分散ハッシュテーブル

斉藤　賢爾[†a)]　　高野　祐輝[††]

Distributed Hash Tables Adapting to the Conditions of the Real World

Kenji SAITO[†a)] and Yuuki TAKANO[††]

あらまし　今世紀最初の年に分散システムの理論的世界に登場した分散ハッシュテーブル (DHT: Distributed Hash Table) は，Key-Value 型検索 (キーに対応する値をルックアップするサービス) を規模拡大性かつ可塑性 (特に，壊れても，残った部分の変化により機能を維持できる性質) をもちつつ提供することを可能とし，その後の分散データ構造及びアルゴリズムの研究の基盤として用いられてきた．しかし，その圧倒的な関連論文の数と比較して，実用された例は極端に少ない．DHT が現実の問題に対応するためには，実社会での応用が要求する性能 (検索や経路表の維持の効率性) と機能 (範囲検索等) の条件を満たすとともに，現実に運用されているネットワークにおける様々な制約 (NAT: Network Address Translation 等) を乗り越える必要がある．本論文では，DHT がこれらの困難を克服し現実の問題の解決に寄与できるための要素技術を調査・解説する．

キーワード　分散ハッシュテーブル，DHT, P2P，規模拡大性，可塑性

1. まえがき

1.1 分散ハッシュテーブルとは何か

ハッシュテーブルは，< キー，値 > ペアを格納し，高速にルックアップするためのデータ構造及びアルゴリズムである．ハッシュテーブルにキー key を格納する際は，key のハッシュ値を計算し，その値をもとに格納先のテーブルにおけるエントリを決定する．ハッシュテーブルに用いられるテーブルは，基本的に連続したアドレスをもつメモリ空間であるため，ハッシュ値が求まるとそのエントリ自体の検索は $O(1)$ で行えることになる (実際の検索性能はハッシュ値の衝突確率による)．

分散ハッシュテーブル (DHT: Distributed Hash Table) は，ハッシュテーブルを拡張した概念であり，ネットワーク上に < キー，値 > ペアを格納し，キーに対応する値をルックアップするサービスを提供する分散データ構造及びアルゴリズムの一種である．DHT は，ノード(注1)の識別子 (IP アドレス等) とキーをハッシュ関数（典型的には暗号学的ハッシュ関数）を用いて同一の固定長ビット空間に配置し，当該空間上でキーに近接するノードに < キー，値 > ペアを格納するべく P2P (Peer-to-Peer) オーバレイネットワークを構成する手法であると一般化できる．

この手法では，ノードとキーが空間上に均一に配置されると期待できることから，均質的に負荷を分散させることが可能である．また，冗長化やデータの再配置によりチャーン (churn; ノードの頻繁な出入り) に耐性をもてるように設計されるという特徴をもつ．ノード数 N の増加に対して，一般に検索性能を $O(\log N)$ にできることから，規模拡張性をもつ分散ストレージとしての応用も期待される．

1.2 分散ハッシュテーブル小史

第 1 世代の DHT としては，リング構造のオーバレイネットワークを利用した Chord [1]，PRR [2] 構造を利用した Tapestry [3] 及び Pastry [4]，高次元

† 慶應義塾大学大学院政策・メディア研究科，藤沢市
Graduate School of Media and Governance, Keio University, 5322 Endo, Fujisawa-shi, 252–0882 Japan
†† 情報通信研究機構ネットワークセキュリティ研究所セキュリティアーキテクチャ研究室，小金井市
Security Architecture Laboratory, Network Security Research Institute, National Institute of Information and Communications Technology, 4–2–1 Nukui-Kitamachi, Koganei-shi, 184–8795 Japan
a) E-mail: ks91@sfc.wide.ad.jp

(注1)：P2P ネットワークにおけるノードをピア（対等な存在）と呼ぶことがある．しかし，本論文で解説する諸手法では，厳密な意味で対等ではない場合があるため，原則的にノードと呼び，通信で対向するノードを特に意味する場合のみピアと呼ぶ．

©一般社団法人電子情報通信学会 2013

トーラスを利用した CAN [5] (以上 2001 年)，ハッシュ値間の XOR (排他的論理和) を距離の尺度とした Kademlia [6] (2002 年) 等がある．

これらの DHT は，特に Chord, Pastry といったものを中心に，その後の分散データ構造及びアルゴリズムの研究の基盤として用いられている．Kademlia は，BitTorrent [7] の多くのクライアントでトラッカー（インデクスサーバ）機能の分散方式として採用され，また，eMule [8] で採用されるといったように，特に実用上の応用例がある．

その後に発表された DHT の例としては，バタフライグラフ (Butterfly Graph)(注2) に基づく Viceroy [9] (2002 年) や de Bruijn グラフ(注3) に基づく Koorde [10] (2003 年) 等があり，2000 年代後半になっても，後述するように新たな DHT が発表され続けている．

1.3 本論文の着眼点と構成

以上のように，今世紀最初の年に DHT が発表されて以来，新たな DHT の提案とその理論的研究は活発に行われているが，そのことと比較して，DHT が実用的に活用されている例は極端に少ない．

これは，提案されてきた多くの DHT が，現実の問題に対処するための諸性質を持ち合わせていないためと考えられる．DHT が現実の問題に対応するためには，実社会での応用が要求する性能 (検索や経路表の維持の効率性) と機能 (範囲検索等) の条件を満たすとともに，現実に運用されているネットワークにおける様々な制約 (NAT: Network Address Translation 等) を乗り越える必要がある．

そこで，本論文では，DHT にこれらの諸性質をもたせるべく行われてきた諸研究における手法を調査，整理し，解説する．

本論文は次のように構成される．**2.** では，DHT に関係する概念を整理する．**3.** では，DHT における検索や経路表の維持の効率化に向けた諸手法について述べる．**4.** では，DHT における検索機能の向上，具体的には範囲検索の実現に向けた諸手法について述べる．**5.** では，DHT を実ネットワークで運用する上での課題である，(1) 参加ノードやそれらが提供する資源の品質のばらつき，(2) 下位層のトポロジーとの乖離による性能劣化，(3) 実運用されるネットワークでの接続性上の制約，といった状況に対応できる DHT を実現するための諸手法について述べる．**6.** では本論文で解説した課題，技術，手法の関連を整理し，最後に **7.** でまとめと今後の課題を述べ総括する．

2. 概念の整理

この章では，DHT に関連する概念を整理する目的で，構造化/ 非構造化 P2P ネットワークとその DHT との関係，初期の DHT に利用されてきた構造化 P2P ネットワークの概要を解説する．

2.1 構造化/非構造化 P2P ネットワーク

2.1.1 構造化 P2P ネットワーク

構造化 P2P ネットワークは，アドレッシング構造をもつ P2P ネットワークを意味する．すなわち，各ノードにユニークな ID が割り当てられており，その ID をアドレス (=宛先) として指定して経路制御を行うための構造をもつ．各ノードは当該アドレッシング構造に基づいた経路表を保持し，あるデータを保持する担当ノードを検索するための経路制御は，その経路表に基づいて行われる．大きく分けて，構造化 P2P ネットワークには次の種類がある：

（1） $\log N$ 次数ネットワーク

ノード数に対して，各ノードに繋がるエッジの個数，すなわち次数が対数的にしか増加しない．**2.2** で述べるとおり，初期の DHT は主にこの種の P2P ネットワークを採用していた．

（2） 定次数ネットワーク

ノード数が変化しても次数が変化しない．この種の P2P ネットワークを採用する DHT については **3.3** で述べる．

（3） N 次数ネットワーク

基本的にフルメッシュで各ノードが接続される．この種の P2P ネットワークを採用する DHT については **3.1** で述べる．

また，これらのネットワークを階層化したりグループ化する試みも行われており，それらについては **3.2** で述べる．

表 1 に，以上に基づく，本論文で解説する技術の分類をまとめる．

2.1.2 非構造化 P2P ネットワーク

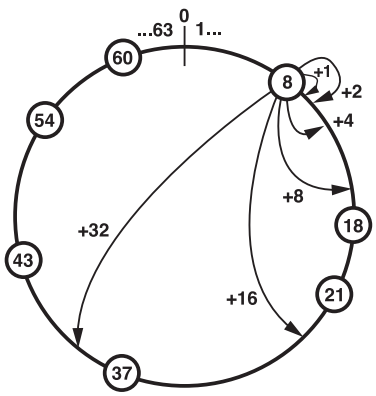
構造化 P2P ネットワークに対して，非構造化 P2P ネットワークは，特定のアドレッシング構造をもたな

(注2)：直径を r，次数を d としたとき，ノード数 N が rd^r となるグラフで，$r = 1, d = 2$ のときに蝶のような形になる．

(注3)：2^b のノードについて，あるノード a からは $b_1 = 2a \bmod 2^b$，$b_2 = 2a + 1 \bmod 2^b$ となるノードへリンクが張られるグラフ．この説明は，基数が 2 の de Bruijn グラフについてだが，基数を任意の B にすることが可能．

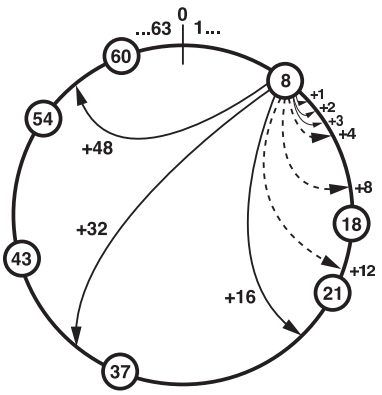
表 1　本論文で解説する技術の分類
Table 1　Classification of described technologies.

分類	技術
$\log N$ 次数ネットワーク	Chord [1], DKS [11], Chord$^{\#}$ [12], PRR [2], Tapestry [3], Pastry [4], P-Grid [13], CAN [5], Kademlia [6], BATON [14], SkipGraph [15], SkipIndex [16], SkipNet [17]
定次数ネットワーク	Symphony [18], Viceroy [9], Koorde [10], FISSIONE [19], Armada [20], ERQ [21], DK [22], SKY [23], BAKE [24]
N 次数ネットワーク	D1HT [25], 1h-Calot [26], OneHop [27], 循環経路制御 [28], EpiChord [29]
グループ化/階層化	Brocade [30], Diminished Chord [31], G-TAP [32], P2PSIP に基づく配信 [33], GTPP [34], G-Kad [35], 近接クラスタリング [36], P3ON [37], DTUN [38]



- ID 空間のビット数 $B = 6$ の例.
- ノード 8 は，隣接するノード 18，60 へのリンクの他に，矢印で示された位置から順方向に最も近いノードへのリンクをもつ.

図 1　Chord におけるリング構造
Fig. 1　Ring structure of Chord.



- ID 空間のビット数 $B = 6$，分割の基数 $k = 4$ の例.
- ノード 8 は，自身を基点に ID 空間を 4 分割して得られる最初の区間を再帰的に 4 分割していった位置に向けたリンクをもつ.

図 2　DKS によるリングの k 分割
Fig. 2　k-ary division of the ring in DKS.

い P2P ネットワークを意味する.

構造化ネットワークの性質を模倣することにより，非構造化 P2P ネットワークを採用して DHT を実現することも可能であり，[39] にて提案されているが，DHT として動作するためには，何らかのレベルにおいて特定の構造をもつことが本質的であるため，本論文では詳細を述べず，以降では構造化 P2P ネットワークのみに注目する.

2.2　DHT に利用されてきた構造化 P2P ネットワーク

この節では，特に初期の DHT に利用されてきた構造化 P2P ネットワークを，アドレッシング構造により分類し紹介する.

2.2.1　リング構造

Chord [1] は，ノードの ID 順に並んだリング状のトポロジーを形成し，各ノードは，隣接するノードへのリンクに加え，自身の ID から 2^n $(0 \leq n < B)$ だけ離れた先へのリンクをもつ (図 1)．そのため，中継の回数 (オーバレイネットワーク上でのホップ数) は $O(\log N)$ となる．ただし，ここで B は ID 空間のビット数，N はネットワークに参加しているノードの総数である.

DKS [11] (2003 年) と Chord$^{\#}$ [12] (2006 年) は，Chord の変種である．DKS では，リング上で 2 分探索を行う Chord のアルゴリズム(注4) を k 分探索できるように一般化する．(図 2)．そのため，DKS での中継回数は平均 $O(\log_k N)$ となる．Chord$^{\#}$ は Chord のアルゴリズムからハッシュ関数の利用を取り除いたものである．数値化されるキーがそのまま非均質な ID 空間を構成するが，その空間の中にノードを配置し，リンクを（確率的でなく）正確にべき乗の間隔で張るために，リモートノードがもつ経路表の情報を再帰的に利用する．例えば自ノードから +32 離れた位置へのリンクは，自己の +16 のリンクが指すリモートノードから +16 離れた位置へのリンクを取得すればよい．そのことにより，経路表の 1 エントリ当りの更新コストを $O(\log N)$ から $O(1)$ に減少させている．更に，Chord$^{\#}$ では，Chord ではできなかった範囲検索も可能としている.

(注4)：ただし，筆者らはどのような構造にも適用可能としている.

Symphony [18] (2003 年) は，スモールワールド現象(注5) [40] を応用した構造化 P2P ネットワークアルゴリズムである．Chord などでは，リンクはべき乗の間隔で張られていたが，Symphony では，リンク先はスモールワールドネットワークとなるように確率的に固定数が選ばれ，Chord と比較して少ない次数で済む．しかしながら，このアルゴリズムはネットワークに参加するノードの総数が既知であることを要求するため，実用上は，総数を予想することによりその要求に対応しなければならない．

2.2.2 PRR 構造 (プラクストン・メッシュ)

提案者である Plaxton, Rajaraman, Richa の頭文字をつなげて呼ばれる PRR [2] 構造 (1997 年；プラクストン・メッシュとも呼ばれる) に基づく方法では，ID のプレフィックス若しくはサフィックスマッチを用いてデータの転送を行う (図 3 は後者の例)．PRR 構造そのままでは，大域的知識が必要であるといったように，実用上の問題があったが，PRR 構造に基づいた，より現実的な方法がいくつか提案されている．

Tapestry [3] は PRR 構造に基づいた構造をとり，プレフィックスマッチによるデータの転送を行う．

Pastry [4] も同様に PRR 構造と，プレフィックスマッチによる中継を行うが，こちらは，ID 空間上の近隣ノードへのリンクも保有し，経路探索の最終的な局面での到達可能性と効率を高めている．

P-Grid [13] (2003 年) も，PRR 構造のようなプレフィックスマッチによる検索を行うが，こちらは ID 空間をツリー状に管理して構造化 P2P ネットワークを実現する．

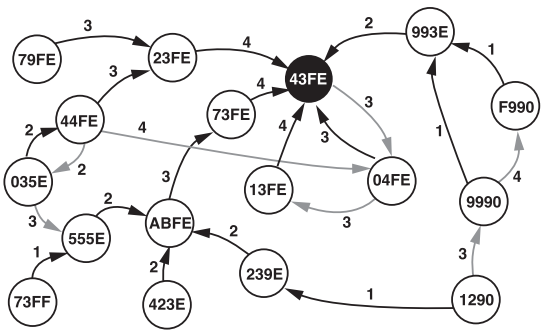
PRR，Tapestry，Pastry，P-Grid のいずれも中継回数の平均は $O(\log N)$ となる．ただし，N はネットワークに参加しているノードの総数である．

2.2.3 高次元トーラス構造

CAN [5] は，d 次元の空間をノードの ID に基づき分割していくことで構造化を行うアルゴリズムである (図 4)．そのため，CAN の各ノードには d 次元の ID が割り当てられる必要がある．これは d 個のハッシュ関数を用いて達成される．直観的には，CAN における経路制御は，基点から目的地までの直線を引き，その線分を含む区域を担当するノードをたどることで行われる．N をネットワークに参加しているノードの総数としたとき，CAN での中継回数は平均 $O(dN^{\frac{1}{d}})$ となる．

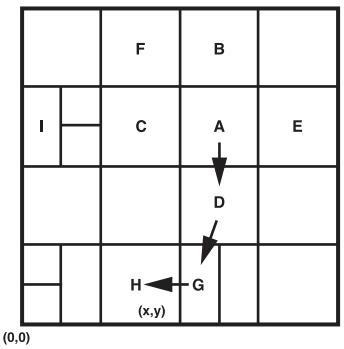
2.2.4 XOR 距離に基づく構造

Kademlia [6] は，ID を木構造とみなして構造化を行うアルゴリズムである．ID はビット列であるが，Kademlia の基本的な設計では，ID の最上位ビットから順に 0 (例えば左) か 1 (例えば右) かを選択するとみなすことで，ID 空間を平衡 2 分木として解釈する (図 5)．ほとんどの構造化 P2P ネットワークでは，ID 間の差を導出するために減算が用いられているが，Kademlia では XOR を用い，最上位ビットから順に，自ノードの ID と異なるビットをもつ ID のノードを



- ID が 16 進数で 4 桁の例.
- 矢印のラベルは，右から何桁目で ID が異なるノードへのリンクであるかを示す.
- 黒い矢印は，それぞれの場所を基点とした，ノード 43FE への経路を示す.

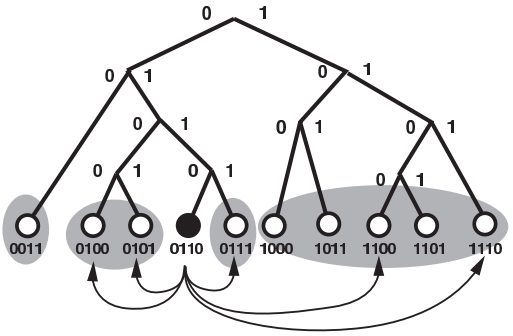
図 3 PRR 構造
Fig. 3 PRR structure.



- 2 次元トーラスの例.
- ノード A は隣接する B,C,D,E へのリンクをもつ.
- 各次元の空間は閉じており，F と H，B と G，E と I は互いへのリンクをもつ.
- 矢印はノード A から座標 (x,y) への経路を示す.

図 4 CAN における n 次元トーラス構造
Fig. 4 n-dimensional torus structure of CAN.

(注5)：知り合いを何段階かたどれば，世界の誰にでも行き着くという (実証されていない) 現象であり，「世間は狭い」ことを意味する．工学的には，スケールフリーネットワークなどで実現できる．



• 4 ビットの ID 空間で，バケット長 $k = 2$ の例．
• ノード 0110 から見た場合，灰色で囲まれたノード群が経路表の各バケットの対象となり，それぞれ 0〜2 個のリンク（矢印）をもてる．

図 5 Kademlia における XOR 距離に基づく構造
Fig. 5 Structure based on XOR distance of Kademlia.

まとめて経路表の固定長 k のバケットに置き，遠方のノードほど疎に，近隣のノードほど密にリンクをもつようにしている．この経路表を k-buckets と呼ぶ．Kademlia での中継回数は平均 $O(\log N)$ である．

2.2.5 その他の構造

BATON [14] (BAlanced Tree Overlay Network; 2005 年) は，平衡 2 分木としてトポロジーを構成するアルゴリズムである．BATON では，範囲検索に適したルーチングを行うことが可能となっており，中継回数は平均 $O(\log N)$ となる．

SkipGraph [15] (2003 年) は，SkipList(注6) [41] (1990 年) をもとにした構造化 P2P データ構造及びアルゴリズムであり，可塑性 (特に，壊れても，残った部分の変化により機能を維持できる性質) を組み込むことにより，分散システムにおいて平衡木の機能を提供する．SkipGraph は階層化された構造をもち，ノードが各階層にてどのリストに属するかはランダムに選択されたメンバシップベクタと呼ばれる値に基づいて確率的に決まる．そのため，他の多くのアルゴリズムと違いリンクも確率的に決定される．SkipGraph での中継回数は，ネットワーク上のデータ数を M とすると平均 $O(\log M)$ となり，各ノードは各データにつき平均 $O(\log M)$ のリンクを保持する必要がある．

SkipGraph の一番の特徴は，その構造とルーティング方法が，範囲検索などの，より柔軟な検索に適していることである．SkipGraph の派生として，多次元での検索を可能とした SkipIndex [16] (2004 年) がある．また，同様に SkipList に基づくものとして SkipNet [17] (2003 年) が提案されている．

3. 検索や経路表の維持の効率化

DHT では，検索にあたり，オーバレイネットワーク上のホップを必要とする．複数のノード間でのメッセージのやり取りを要する処理が行われること自体，オーバヘッドが高いことに加え，下位層のトポロジーに無関係にオーバレイのトポロジーが作られる場合，下位層での無駄な通信を引き起こしやすい．また，経路によっては，性能が著しく低いノードに全体が依存し性能が劣化するおそれもある．

一方，実社会で起きている変化としては，データセンタなどで大規模で均質的な資源が集約され利用可能になっている反面，モバイル端末の増加等によるノードの不均質性も一層進行している．加えて，現実世界と情報システムを結ぶ近年のサイバーフィジカルな応用においては，実時間に関わる要求を満たすために，性能上の最悪値を考慮した設計が求められることもあり得る．

この章では，経路表のサイズを大きくすることによってホップ数を減らす試みや，ノードの非均質性を考慮して DHT を階層化することにより性能の向上を図る試み，また，定次数化を通して経路表のサイズを抑えつつ，かつ最大ホップ数の上限を保証する試みについて解説する．

3.1 1 ホップ DHT

DHT は，チャーンに対する耐性や，規模拡張性を重視するために，基本的に経路表を小さく抑えるという発想で設計されている．経路表が小さいということは，ノードの新たな参加や離脱が起こった場合，テーブルを書き換えなければならないノードの数が低く抑えられていることを意味するからである．

しかし，DHT の設計における前提ともいえるこの条件を見直す動きも出ている．すなわち，隣接するノードからその隣接するノードへのポインタを取得し，経路表を成長させる「先読み」の手法がいくつか考案されている．

現在は，ハードウェアの進歩により，ノードのメモリや利用可能な帯域幅に対する制限が緩くなっており，数万ノードといった中規模の DHT では，実際に各ノードが他のほぼ全てのノードに対するポインタを維持することも現実的な範囲で実現可能となっている．

(注6)：階層化された連結リストにより，平衡 2 分探索木と同等の効率での探索を可能にしたデータ構造．

これは，クラウドコンピューティングにおいてDHTの技術を応用する場合のように，計算のための資源をデータセンタに集約でき，チャーンは起きないがDHTの可塑性を利用してノードの管理コストを低く抑えたい場合には，特に有用と考えられる手法である．

経路表の先読みを用いるDHTの手法においては，一貫性のあるポインタ情報をいかに高速かつ低コストで全ノードに配信するかが設計上の鍵となっている．

2004年に発表されたEpiChord [29]では，Chordのようにリング構造を用いるが，リンクをべき乗間隔で張る代わりに，検索の実行を通してリンク先を集めたキャッシュを各ノードが維持する．結果，転送が必ず1ホップになるわけではないが，検索が頻繁に行われる環境では，ほぼ1ホップで転送が行われる．

2005年に提案されたD1HT [25]はEDRA (Event Detection and Dissemination/Reporting Algorithm)によりDHTのメンバシップに関わるイベントを共有する．EDRAは，基本的にはSkipListを用いたフラッディングである．

同じく2005年に提案された1h-Calot [26]は同様にSkipListを用いてメンバシップに関わるイベントの拡散的な配信を行う．1ホップDHTを実現する1h-Calotでは数千ノード，2ホップDHTを実現する2h-Calotでは数百万ノードを維持できるとされるが，高いチャーン耐性を目指して設計されてはいない．

OneHop [27]（2004-2009年）は，初めて1ホップDHTの概念を実現したものであり，Chordのプロトコルを応用して先読みを行う手法である．OneHopでは，各ノードにてメンバシップの完全な情報を維持し，当該情報が最新のものであれば1ホップにて目的のノードに到達し，そうでなければ少数のホップ数で到達する．2009年の論文 [27]では，99%の検索において1ホップで目的のノードに到達できることが示されている．OneHopでは，高速かつ低帯域幅でメンバシップ情報をシステム全体に拡散させるために，ハッシュ値の空間をk個のスライスに分割している．各スライスは，その中央値に対応するノードをスライスリーダとして選出する．また，各スライスは更にu個のユニットに分割され，それぞれの中間点の値に対応するノードをユニットリーダとして選出する．メンバシップの変更（ノードの参加あるいは離脱）を検出したノードは，自己が属するユニットのスライスリーダにメッセージを送る (図6)．スライスリーダは，単位時間内にスライス内で生じたメンバシップの変更の通知を集約し，他のスライスリーダに通知する．各スライスリーダは，単位時間ごとに，受け取った通知を集約してスライス内のユニットリーダに送信する．ユニットリーダは，受け取った情報を通常のChordの保守メッセージに載せて近隣のノードに拡散させる．

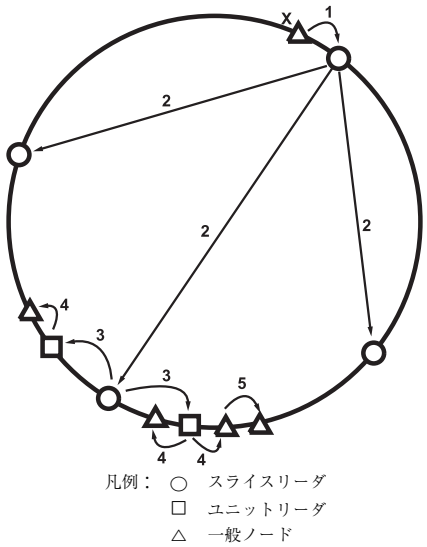


図6　OneHopにおけるメンバシップ情報の拡散手法
Fig. 6　Spreading membership data in OneHop.

OneHopは，現在，MITにて実装されているChordの拡張経路制御の選択肢の一つとして実装・提供されており，当該実装のChordの応用として作られているアプリケーションは，無修正でOneHopの恩恵を受けることができる．

2009年には，OneHop, D1HT, 及び1h-Calotの性能の比較 [42]が行われている．1,000万ノードまでに規模を拡大させた場合の3者の振舞いの検証と，データセンタ環境におけるD1HTと1h-Calotの比較を行った結果，当該論文の著者ら (D1HTの開発者でもある)は，D1HTが最もオーバヘッドが低く，また，データセンタにおける高性能コンピューティングに最も向いていると結論づけている．

3.2　グループ化/階層化DHT

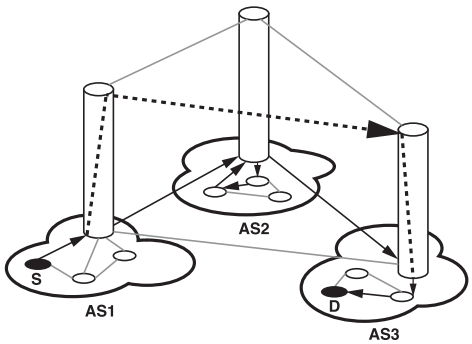
実際にオーバレイネットワークを構成するノードは，一般に非均質である．CPU性能，ストレージ資源，利用できる帯域幅等に余裕があり，また長時間ネットワークに参加しているノードと，そうでないノードが混在していることを考えると，前者のもつ資源を有効に活用するためにDHTを階層化したいと考えること

は自然な発想である．

そのような手法の先駆けとして，例えば Tapestry の研究グループでは，スーパーノードを設けるランドマーク経路制御を行う Brocade [30] (2002 年) を提案した (図 7)．

また，2004 年に提案された Diminished Chord [31] は，Chord にグループ化を採り入れたものである．Diminished Chord のリングに参加するノード群の任意の部分集合は，全体のリングにおける経路制御機構を利用することで，独自のリングを形成せずに全体に対するサブグループを形成し，当該グループ内での検索サービスを提供できる．独自のリングを形成する場合，サイズが k であるサブグループは $O(k \log k)$ のストレージ資源を消費するが，Diminished Chord では $O(k)$ しか消費しない．ただし，サブグループに属さないノードにも，当該サブグループに関する情報が格納される．

2008 年に提案された G-TAP (Grouped Tapestry) [32] は，Tapestry において，参加ノードの非均質性に基づいてオーバレイネットワークをグループに分割し，柔軟な経路制御を実現する手法である．従来の DHT における経路制御に加えて，グループ内のノードのみを経由し，グループ内のノードに到達する PC (Path-Constrained) 経路制御 (Diminished Chord はこれを実現できない) と，最終的にグループ内のノードに到達することのみを保証する DS (Destination-Specified) 経路制御を利用できる．



- 始点 S から終点 D への経路は，通常のオーバレイ経路制御では実線の矢印をたどる．
- AS ごとにランドマークとなるスーパーノード（円柱）を設け，それらを繋ぐ 2 段目のオーバレイネットワークを PRR 構造等で設ける．
- ランドマーク経路制御では，AS 間を跨ぐ経路は破線のようにショートカットされる．

図 7　ランドマーク経路制御
Fig. 7　Landmark routing.

これらの拡張経路制御を利用することにより，性能の高いノードのみを用いた計算を行ったり，安定したノードのみによりサービスを提供したり，悪意のあるノードを排除した通信，またはグループ内にプライベートな通信が可能となる．

G-TAP では，DHT 内のそれぞれのグループに対し，サブ DHT のための経路制御構造（グループ用の経路表）と，グループの発見のためのグループメンバシップ木 (GMR tree: Group Membership Rendezvous tree) を備える．

G-TAP のグループ自体は階層構造をもたないが，G-TAP を階層構造向けに最適化した H-TAP も提案されている．H-TAP は，経路局所性 (path locality) 及び経路集約性 (path convergence) を実現する．経路局所性は，二つのノードをつなぐ経路が，双方を含む最小のネットワーク上のドメインを出ないことを保証する．経路集約性は，あるキーに関わるメッセージに対して，ネットワーク上のドメインを出る経路が一つのオーバレイルータを必ず通ることを保証する．オーバレイルータは，オーバレイネットワークのレベルでネットワークのドメイン間を繋ぐノードである．トラヒックが特定のオーバレイルータを必ず通るようにすることで，広い帯域幅といった資源を有効に活用したり，トラヒックのモニタリングを容易にできる利点がある．

同じく 2008 年に提案された階層化 DHT によるマルチメディア配信サービス [33] は，IETF にて検討されている P2PSIP [43] に基づく手法である．これはスーパノードを用いるものであり，スーパノードのみが参加するオーバレイネットワークを形成して，下位の DHT を相互に接続する．ノードの識別には，プレフィックス ID とサフィックス ID から成る階層化 ID を用いる．性能に関しては，階層化された Kademlia によるシミュレーション評価が行われている．

2009 年に提案された GTPP (General Truncated Pyramid Peer-to-Peer) [34] は，トランケートされたピラミッド型，すなわち，頂上の部分が切り落とされた四角錐に形状が類似するようにオーバレイネットワークを階層化したアーキテクチャであり，複数段階の階層をサポートする DHT の階層化の例である．各階層は独自のオーバレイネットワークを形成し，各ノードは下位のネットワークに対するスーパーノードとして動作する．

2011 年に提案された G-Kad [35] は，Kademlia の

経路表である $k-$buckets が複数の異なる経路を冗長に格納できることを生かしてグループ化を行うとともに，グループの参加ノードが取得したデータの属性の履歴をもとに，当該グループにおいてデータを投機的にあらかじめ取得することにより高速化を図る手法を採用している．

3.3 定次数 DHT

DHT における最悪ケースのホップ数を「直径」と呼ぶことがある．

有向グラフ構造に基づき，一定の次数，すなわち経路表のサイズをもつ (あるいは経路表中のエントリ数の最大値が決まっている) 定次数 DHT では，経路表のサイズを小さく抑えつつ，直径の上限を保証することが可能であり，経路表の維持の効率化とともに，性能上の最悪値を考慮した設計が可能である．

そのような定次数 DHT の例として，バタフライグラフに基づく Viceroy [9] (2002 年) や de Bruijn グラフに基づく Koorde [10] (2003 年) がある．また，カウツグラフ (Kautz Graph) に基づく DHT である FISSIONE [19] (2005 年) がある．

ここで，カウツグラフについて簡単に解説する．カウツグラフは有向グラフであり，次数 d のカウツグラフは，各々d 個の外向きリンク (ノードから出る方向のエッジ) と d 個の内向きリンク (ノードに向かう方向のエッジ) をもつノードからなる．各ノードは，隣り合う数字が全て異なる $d+1$ 進数の番号（カウツ文字列）により区別され，その外向きのリンクは，自己の番号を左にシフトし，空いた桁を利用可能な数字で埋めた番号をもつノードに接続される．内向きのリンクに対してはその逆の計算を行う．例えば，次数が 3 のカウツグラフにおけるノード 132 は，外向きに番号 $320, 321, 323$ のノードとつながり，内向きに番号 $013, 213, 313$ のノードからつなげられる．

このことにより，例えば，次数が 3 でカウツ文字列長が 6 のカウツグラフでは，972 個の全ノードの中のいかなる 2 個のノードも，6 ホップ以内でつながることが保証され，また，三つの完全に異なる (一つとして同じ中継ノードをもたない) 経路をもつ．カウツグラフでは直径はカウツ文字列長に一致する．また，カウツグラフに基づく DHT では，次数は経路表のサイズを表す．

図 8 に，次数 2，カウツ文字列長 3 の場合のカウツグラフの例を示す．カウツグラフ上の基本的な経路制御は，送信元のカウツ文字列を左にシフトしながら送信先のカウツ文字列に近づけることで行える．例えば図 8 の例では，102 から 012 に向かう経路は，$102 \rightarrow 020 \rightarrow 201 \rightarrow 012$ となる．

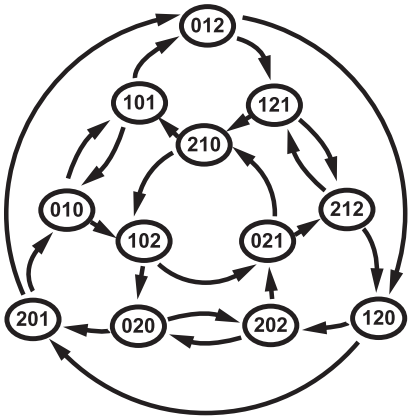


図 8 カウツグラフの例 (次数 2, カウツ文字列長 3)
Fig. 8 An example of Kautz graph.

FISSIONE ネットワークは次数 4，直径は $2\log_2 N$ であり，平均ホップ数は $\log_2 N$ 以下となる．

2008 年には，分散線グラフ (DLG: Distributed Line Graph)(注7) に基づいて任意の定次数をもつ DHT を生成する手法 [22] が提案された．この手法で生成された，N ノードが参加する外向けの次数 d の DHT では，内向きの次数は $1 \sim 2d$ であり，直径が $2(\log_d N - \log_d N_0 + D_0 + 1)$ 未満であることが保証される．ここで D_0 と N_0 はそれぞれ初期ネットワークの直径及びノード数である．この手法において，ネットワークの維持コストは $O(\log_d N)$ となる．

更にこの手法を受け，2009 年には，任意の定次数をもつ分散カウツグラフを用いた DHT である SKY [23] が提案された．SKY は，カウツグラフを用いる DHT の実用上の一つの課題を解決したものである．ハッシュアルゴリズムとしては，任意のキーを次数 2 のカウツ空間に均一にマップするカウツハッシュ(KautzHash) を，任意の次数に適用できるように拡張して用いている．

カウツグラフを実用的に用いる際の最大の問題点の一つは，次数 d とカウツ文字列長 D が決まると，カウツグラフの最大のノード数が $d^D + d^{D-1}$ に決まって

(注7)：あるグラフ G についての線グラフ $L(G)$ は，G の全てのエッジをノードとし，G で隣接するエッジに対応するノード同士を繋げたものである．L を反復的に適用することで様々なグラフを生成できるが，分散線グラフはこの手法を分散化したもの．

しまう点にある．カウツ文字列長が固定であるシステムの場合，ノード数がこの値を超える際には，全ノードの番号を付け替えるといった非現実的な対応を迫られることになる．カウツ文字列長は DHT の直径に一致するため，最小限の長さにすることが望ましく，こうした問題が起き得るが，SKY では，カウツグラフを近似する分散カウツグラフを採用し，カウツ文字列長を可変にすることでこの問題を回避している．

2008 年に提案された BAKE [24] は，DLG を用いる手法とは異なる方法で生成された，均衡カウツ木 (balanced Kautz tree) を用いた DHT である．BAKE ネットワークの直径は $\log_d N$ である．著者らは，この手法は de Bruijn グラフ等にも適用可能としている．

4. 検索機能の向上

DHT ではハッシュ値を用いるため，一般に，格納されているデータの配置はキーの順序を反映していない．そのため，現実的に頻出する，例えば地理上の特定の緯度経度内の地点といったように，ある範囲内に含まれるキーをもつ値を検索するといった応用に向かないという難点があり，そのことが DHT を現実の問題に応用する上での障壁となっていた．これに対し，ハッシュ関数を用いない分散化により範囲検索を可能にしている分散データ構造及びアルゴリズムとして，前述した SkipGraph や Chord# があるが，この章では，これと異なり，DHT 上に範囲検索を実現する手法について解説する．

4.1 プレフィックスハッシュ木を用いた範囲検索

2005 年には，DHT を下位構造とし，下位の DHT に変更を加えずに，範囲検索を含む高度な検索機能を上位モジュールとして実現した例として，Place Lab [44] にて使用された手法 [45] が発表された．Place Lab は，無線通信基地局の ID を用いた測位サービスである．Place Lab では，OpenDHT [46] (2005 年) 上にプレフィックスハッシュ木 (PHT: Prefix Hash Trees) と呼ばれる多次元範囲検索のためのデータ構造を実現することによりサービスを実装した．PHT は，2 進符号化されたキーのトライ木である．範囲検索を行う場合，範囲の最小値と最大値に対する最大の共通プレフィックスを頂点とするサブトリーに含まれるノードを並列に検索することで，当該範囲に含まれる全てのキーに対応する値を取得できる．この手法では，空間充てん曲線(注8) (この場合 Z 曲線) を利用し，多次元に分布するキーを一次元に配置することにより，多次元 (緯度及び経度) の範囲検索に対応している．

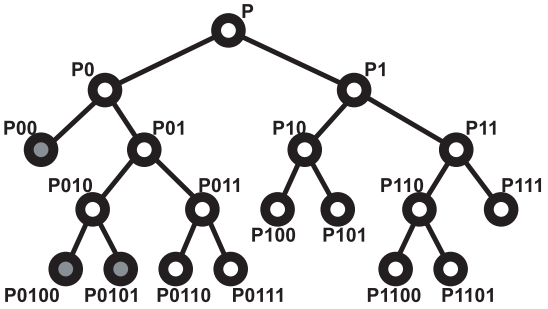
図 9 に，簡単な PHT の例を示す．PHT ノードは，特定のキーの範囲を代表し，そのラベルは，当該範囲に含まれるキーのプレフィックスとなっている．

4.2 カウツグラフを用いた範囲検索

2006 年には，DHT に基づく範囲検索手法として Armada [20] が提案された．Armada は多次元の範囲検索をサポートし，N 個のノードが参加するオーバレイネットワークで $2\log_2 N$ 以内のホップ数で結果を返すことを保証する，遅延制限付き (delay-bounded; 最大ホップ数の保証付き) の手法である．Armada はカウツグラフに基づく DHT である FISSIONE (**3.3** 参照) 上で動作する．

Armada では，実数の範囲を木構造に分割するハッシュアルゴリズムを用い，キーの順序を保存したままカウツ名前空間にマップする．その上で，FISSIONE ノードから構成され，外向きのリンクを順序づける FRT (Forward Routing Tree) を用いて，カウツ名前空間内の範囲を検索することができる．

2009 年に新たに提案された ERQ (Efficient scheme for delay-bounded Range Query) [21] は，カウツグラフに基づく DHT である DK [22] の上で並列検索と刈り込みを行うタイプの遅延制限付き手法である．



PHT ノードのラベル	Z 曲線上のキー (2 進) の例
P00	000010
P0100	010000
	010011
P0101	010101
	010110
	010111

図 9 簡単な PHT の例
Fig. 9 An example of PHT.

(注8)：n 次元の空間を覆いつくす曲線．ヒルベルト曲線や Z 曲線などがある．これを用いることで，一次元のキーを n 次元にマップできる．

ERQ では，DK の上で PHT をエミュレートする．ERQ はノード数 N と次数 d の下位 DHT において，$\log_d N(2\log_d \log_d N + 1)$ ホップ以内で検索を終了することを保証する．ERQ でも空間充てん曲線 (Z 曲線) を利用し，多次元に分布するキーを一次元に配置する．

ERQ は，次数が 4 のとき，Armada より高性能であり，処理コストも低いことが示されている．

5. 実ネットワークへの順応化

P2P はそもそも，ネットワークに接続されたコンピュータの余剰資源を効率的に利用する目的で発想されたものであり，下位層のトポロジーを考慮した上で，CPU，ストレージ，ネットワーク帯域幅といった資源を効率的に利用できるように設計されることが望ましい．

また，DHT を実用的に応用していく上では，リンク障害や，NAT やファイアウォール等の存在を前提とすると，到達性に推移性(注9)がなかったり，IP での外部からの到達性がない環境を想定する必要がある．また，悪意のあるノードや，過負荷により事実上停止しているノード等も考慮する必要もあり，様々な要因により接続性に制限がある状況を想定しなければならない．

この章では，資源の評価や近傍性の考慮の手法について解説する．また，非推移的な接続性の問題，DHT において NAT 越えを行う手法や，オーバレイネットワークの分断が起こり得る状況下で，先読みを用いて DHT を安定運用する手法を解説する．

5.1 順位付けと評判

3.2 で示したようなグループ化/階層化 DHT を利用して，実際にサービスの質を安定させるためには，参加ノードやそれらが提供する資源を各自が評価でき，不適切なノードへの転送を避けたり，必要なレベルの資源をもつノードを要求先として採用できる必要がある．

このような評価は，DHT の分散性を考えると，あらかじめ信用が付与されている第三者によるのではなく，それ自体が分散化されたアルゴリズムで行えることが望ましい．

そのような分散化した評価システムを，特に評判システムと呼ぶ．評判システムでは，あらかじめ信用が付与されていない第三者が資源やノードを評価した値，すなわち評判を利用して評価を行う．評判システムの例としては，各々のノードによる評価を，そのノードの評判により重み付けし，再帰的に計算する，(Secure) EigenTrust [47] (2003 年) 等がある．

2006 年に発表された論文 [48] では，P2P システムにおける評判システムを分類し，詳細な要求分析を行っている．この論文では，評判システムの機能を表 2 に示すように三つに分割する．論文では，関連用語を定義した上で，これらの機能を実現する際に考慮が必要な概念を整理し，要求を明らかにしている．

2009 年に提案された局所的平衡モデル [49] は，ノードのもつ資源のランク付けの計算のための数学的モデルである．ノードが自分でもつべき資源は何か，他ノードに求めるべき資源は何で，どのノードを経由して取得すべきか，自ノードが他ノードに提供すべき資源や，自ノードを経由して他ノードに転送すべき資源・サービスの品質はどの程度であるべきか，といった判断が自動的に行えることを目的としている．

この計算は，反復的にノード間の資源共有に用いられる．このモデルでは，各ノードが融通する資源の価値のバランスがとれるような調整が行われる．

5.2 近傍性の考慮

ネットワーク全体の負荷を考えると，オーバレイネットワークで近接するノードが下位のネットワークでは離れていることにより，オーバレイのホップのたびに下位層で大きなオーバヘッドがかかるような事態は避けたい．

下位ネットワークにおけるノードの近傍性の考慮は，DHT のみならず，分散システム全体における大きな課題である．

2003 年に発表された論文 [50] では，下位ネットワークの近傍性を考慮する手法を次のように分類している．

表 2 評判システムの機能
Table 2 Functionality of a reputation system.

情報収集	自己同一性の識別
	情報源
	情報の集約
	新規参入者に対するポリシー
採点と順序づけ	善い vs. 悪い振舞い
	量 vs. 質
	時間に対する依存
	選択のしきい値
	ピアの選択
応答	インセンティヴ
	罰

(注9)：ノード A–B 間と B–C 間の通信が可能である場合に，A–C 間の通信も可能である性質．

- 近接ノード選択 **(PNS: Proximity Neighbor Selection)**：経路表に置くノードを近傍性に基づいて選択．
- 近接経路選択 **(PRS: Proximity Route Selection)**：経路の次のホップを近傍性に基づいて選択．
- 近接識別子選択 **(PIS: Proximity Identifier Selection)**：近傍性に基づいた識別子空間の構成．

2008 年に提案された近接クラスタリング [36] は，近隣のノードからなるクラスタを形成するものである．単にスーパノードをルータとするのではなく，物理的に近傍なスーパノードとのオーバレイネットワークを形成する p クラスタ（物理クラスタ）と，論理的に（ハッシュ値の近い）近傍なスーパノードとの接続をもつ v クラスタ（論理クラスタ）の両方を検討し，それぞれに適した応用を分析している．近接クラスタリングは，近接ノード選択の 1 手法と分類できる．

同じく 2008 年に提案された P3ON (Proximity Based Peer-to-Peer Overlay Networks) [37] は，AS (Autonomous System) 番号とノードのハッシュ値を連結したものをノード ID とすることにより，AS ごとにノード ID が近接するようにした，近接識別子選択の一手法である．トポロジーとしては，AS ごとのリングをもつ 2 段階の階層構造をもつ．

Kademlia の経路表である k-buckets の冗長性を利用する G-Kad (**3.2**) でのグループ化は，特に近傍性を意識したものではないが，経路表に存在するノードのうち，同一グループに属するノードに対して優先的にホップするため，近傍性に基づいてグループ化を行うなら近接経路選択を行うことになる．

5.3 非推移的な接続性の問題

2005 年に発表された論文 [51] では，DHT における非推移的な接続性の問題が議論されている．

ノード A, B, C があるとして，A–B 間と B–C 間の通信は可能であるが，A–C 間の通信ができない場合，3 者間の接続性は非推移的といえる．非推移的な接続性の問題は，リンクの障害や，経路表の更新，ISP 間の問題など，様々な要因により起こり得る．

[51] の著者らは，Chord，Kademlia 及び Bamboo [52] (Pastry をもとにして 2003 年に発表された DHT 実装) のそれぞれの実装を PlanetLab [53] 上で実際に 1 年以上運用した経験から，非推移的な接続性による動作不良として，経路表に載るノードが不可視であったり，経路制御においてループが発生したり，値の取得や識別子空間の分割に失敗する例を挙げ，それぞれに対する対処方法を述べている．

[51] の著者らは，それらの対処方法は短期的なものであり，今後の DHT 設計者は，当初から非推移的な接続性の問題を考慮する必要があると述べている．

5.4 循環経路制御

3.1 にて解説した先読みは，ノード間の接続性に問題があったり，一部のノードが悪意をもって運用されている場合等での安定運用に向けた応用も可能であり，[51] の著者らもその可能性を示唆している．

循環経路制御 (CR: Cyclic Routing) [28] (2009 年) は，実環境において一般的に利用できる，先読みを用いた既存の DHT の拡張経路制御手法である．CR により拡張された Chord では，拡張されないものと比較して，5∼10%のノードが悪意をもつ場合は，$\frac{1}{2}$ の検索失敗率を，40∼50%のノードが悪意をもつ場合は，2 倍の検索成功率を観測できたとしている．

CR では，ノード s から d にメッセージを送り，ノード d から s に返信が返るまでの経路に含まれるノードのリストをサイクルと定義する．サイクルは，通常の探索メッセージにノードの情報を載せていくことにより取得できる．サイクルを知ることにより，s は経路表を先読みしてメッセージを送信できる．これにより，メッセージングの性能が向上する他，不都合なノードを迂回することが可能となる．

5.5 NAT 越え

接続性を非対称的にする NAT もまた，DHT の実運用上の問題となる．NAT を越える手法には様々なものがあるが，ここでは DHT を利用して NAT を越える手法を紹介する．

NAT には大きく分けて Cone NAT と Symmetric NAT の 2 種類がある．前者では，送信元ソースアドレスが同じであれば，宛先が異なる場合でも同じソースアドレスに変換される．一方，後者では，宛先が異なる場合には異なるソースアドレスに変換される．

Cone NAT を越えるための既存の手法の一つとして，UDP Hole Punching がある．これは，NAT 下にあるノードから対向ノードに向けて先に通信を行い，変換後のアドレスを対向ノードに通知するものであり，双方が NAT 下にある場合には，グローバルアドレスをもつ第三者ノードを待合せに利用する．

Symmetric NAT では，接続ごとに変換後のアドレスが異なるため，NAT 越えのために UDP Hole Punching は適用できず，グローバルアドレスをもつ中継サーバを用いる手法等が必要となる．

2010 年に提案された DTUN (Distributed Traver-

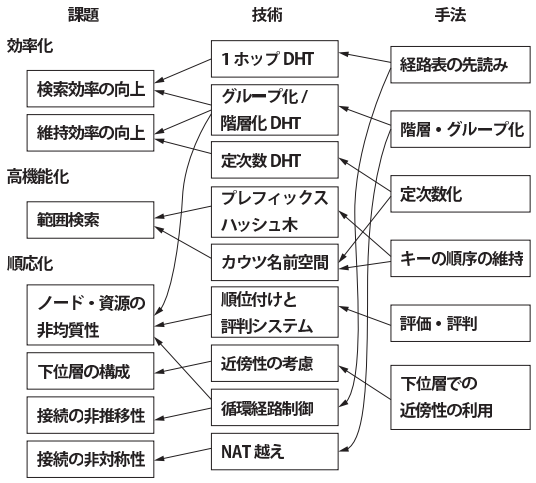


図 10　本論文で解説した課題，技術，手法の関連
Fig. 10　Relations among problems, technology and approaches described in this article.

sal of UDP through NATs) [38] という手法では，グローバルアドレスをもつノードのみからなる DTUN ネットワークを DHT として構築し，NAT 下のノードは DTUN に参加するノードを通して NAT 越えを実現する．DTUN に参加するノードは，Cone NAT 下にあるノードが UDP Hole Punching を行うための待合せノードと，Symmetric NAT 下にあるノードのための中継ノードとしての役割を兼ねる．

DTUN は，グループ化/階層化 DHT の考え方を，接続性の制約を回避する目的に適用したものといえる．

6. 既存研究の整理

図 10 に，本論文で解説した課題とその解決に向けた技術，及びそれらの技術で使われている手法との関連を示した．

本論文で解説した技術で用いられている手法は，「経路表の先読み」「階層・グループ化」「定次数化」「キーの順序の維持」「評価・評判」「下位層での近傍性の利用」に整理できる．特に「経路表の先読み」「階層・グループ化」は多くの課題の克服に役立ち，DHT を現実世界の条件に適応させる上で多様な役割を担うことができる基本的な手法であるといえる．

7. むすび

本論文では，DHT を現実の問題に対応させる各種の試みについて，2000 年代の後半以降に発表された論文を中心に，(1) 効率化 (検索効率の向上，維持効率の向上)，(2) 高機能化 (範囲検索)，(3) 順応化 (ノード・資源の非均質性，下位層の構成，接続の非推移性・非対称性への対応) といった DHT の研究に着目して調査・整理した．

第 1 世代の DHT の研究から約 10 年が経過するが，この間，DHT は分散環境の現実にもまれる中で成長し，現実的価値を増してきていると考えられる．

一方，現実の世界に目を移すと，2011 年 3 月 11 日に発生した東日本大震災により，通信回線や，通信機能をもつ計算機を動かすための電力そのものといった基盤が，多くの箇所で故障したり喪失したりするという状況が発生した．地震のみならず，気候変動の影響などによる災害での被害の甚大化も想定される現在，そうした状況下においても人々の生活を支え得る分散システムを設計することは，我々にとっての現実的な課題となっている．

例えば，電力の不足による輪番停電が実施されたり，あるいは突発的に停電が起きるといった場合に，DHT を構成する，まとまった単位のノードが一気にシステムから離脱するということが起き得る．そうした場合にも耐え得るレベルの可塑性を実現し，検証していくことは，直近の課題の一つといえるだろう．

本論文が，そうした課題に取り組むための一助になれば幸いである．

謝辞　本サーベイは，文部科学省の科学技術戦略推進費による「気候変動に対応した新たな社会の創出に向けた社会システムの改革プログラム」で，慶應義塾大学が実施する「グリーン社会 ICT ライフインフラプロジェクト」の一環として実施されました．

文　献

[1] I. Stoica, R. Morris, M.F.K. David Karger, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for internet applications,” Proc. ACM SIGCOMM, pp.149–160, Aug. 2001.

[2] C.G. Plaxton, R. Rajaraman, and A.W. Richa, “Accessing nearby copies of replicated objects in a distributed environment,” Proc. ACM SPAA, pp.311–320, June 1997.

[3] B.Y. Zhao, L. Huang, J. Stribling, S.C. Rhea, A.D. Joseph, and J. Kubiatowicz, “Tapestry: A resilient global-scale overlay for service deployment,” IEEE J. Sel. Areas Commun., vol.22, no.1, pp.41–53, 2004.

[4] A. Rowstron and P. Druschel, “Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems,” Proc. 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001), pp.329–350, Nov. 2001.

[5] S. Ratnasamy, P. Francis, R.K.M. Handley, and S. Shenker, “A scalable content-addressable network,” Proc. ACM SIGCOMM, pp.161–172, Aug. 2001.

[6] P. Maymounkov and D. Mazières, “Kademlia: A peer-to-peer information system based on the XOR metric,” 1st International Workshop on Peer-to-Peer Systems (IPTPS) (Springer LNCS 2429), pp.53–65, March 2002.

[7] B. Cohen, “Incentives build robustness in BitTorrent,” The First Workshop on Economics of Peer-to-Peer Systems, May 2003.

[8] eMule Team, “emule-project.net,” 2002. Available electronically at http://www.emule-project.net/

[9] D. Malkhi, M. Naor, and D. Ratajczak, “Viceroy: A scalable and dynamic emulation of the butterfly,” 21st Annual Symposium on Principles of Distributed Computing, pp.183–192, July 2002.

[10] F. Kaashoek and D.R. Karger, “Koorde: A simple degree-optimal distributed hash table,” 2nd International Workshop on Peer-to-Peer Systems (IPTPS), pp.98–107, Feb. 2003.

[11] L.O. Alima, S. El-Ansary, P. Brand, and S. Haridi, “Dks (n, k, f): A family of low communication, scalable and fault-tolerant infrastructures for P2P applications,” CCGRID ’03, pp.344–350, 2003.

[12] T. Schutt, F. Schintke, and A. Reinefeld, “Structured overlay without consistent hashing: Empirical results,” CCGRID ’06, p.8, 2006.

[13] K. Aberer, P. Cudré-Mauroux, A. Datta, Z. Despotovic, M. Hauswirth, M. Punceva, and R. Schmidt, “P-grid: A self-organizing structured p2p system,” SIG-MOD Record, vol.32, no.3, pp.29–33, 2003.

[14] H.V. Jagadish, B.C. Ooi, and Q.H. Vu, “Baton: A balanced tree structure for peer-to-peer networks,” VLDB, pp.661–672, 2005.

[15] J. Aspnes and G. Shah, “Skip graphs,” Proc. Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp.384–393, 2003.

[16] C. Zhang and A. Krishnamurthy, “Skipindex: Towards a scalable peer-to-peer index service for high dimensional data,” Technical Report, Princeton University, 2004.

[17] N.J.A. Harvey, M.B. Jones, S. Saroiu, M. Theimer, and A. Wolman, “Skipnet: A scalable overlay network with practical locality properties,” USENIX Symposium on Internet Technologies and Systems, p.9, 2003.

[18] G.S. Manku, M. Bawa, and P. Raghavan, “Symphony: Distributed hashing in a small world,” USENIX Symposium on Internet Technologies and Systems, p.10, 2003.

[19] D. Li, X. Lu, and J. Wu, “FISSIONE: A scalable constant degree and low congestion DHT scheme based on Kautz graphs,” INFOCOM, pp.1677–1688, 2005.

[20] D. Li, X. Lu, B. Wang, J. Su, J. Cao, K.C.C. Chan, and H. vaLeong, “Delay-bounded range queries in DHT-based peer-to-peer systems,” Proc. 26th IEEE Intl. Conference on Distributed Computing Systems (ICDCS ’06), p.64, 2006.

[21] Y. Zhang, L. Liu, D. Li, F. Liu, and X. Lu, “DHT-based range query processing for web service discovery,” International Conference on Web Services, pp.477–484, 2009.

[22] Y. Zhang, L. Liu, D. Li, and X. Lu, “Distributed line graphs: A universal framework for building DHTs based on arbitrary constant-degree graphs,” 28th International Conference on Distributed Computing Systems, pp.152–159, 2008.

[23] Y. Zhang, X. Lu, and D. Li, “SKY: Efficient peer-to-peer networks based on distributed Kautz graphs,” Science in China Series F: Information Sciences, vol.52, no.4, pp.588–601, 2009.

[24] D. Guo, Y. Liu, and X.-Y. Li, “BAKE: A balanced Kautz tree structure for peer-to-peer networks,” INFOCOM, pp.2450–2457, 2008.

[25] L.R. Monnerat and Cláudio L. Amorim, “D1HT: A distributed one hop hash table,” The 20th IEEE Intl. Parallel and Distributed Processing Symposium, p.10, 2005.

[26] C. Tang, M.J. Buco, R.N. Chang, S. Dwarkadas, L.Z. Luan, E. So, and C. Ward, “Low traffic overlay networks with large routing tables,” ACM SIGMETRICS Performance Evaluation Review, vol.33, no.1, pp.14–25, 2005.

[27] P. Fonseca, R. Rodrigues, A. Gupta, and B. Liskov, “Full-information lookups for peer-to-peer overlays,” IEEE Trans. Parallel Distrib. Syst., vol.20, no.9, pp.1339–1351, 2009.

[28] D. Korzun, B. Nechaev, and A. Gurtov, “Cyclic routing: Generalizing look-ahead in peer-to-peer networks,” ACS/IEEE Intl. Conference on Computer Systems and Applications, pp.697–704, 2009.

[29] B. Leong, B. Liskov, and E.D. Demaine, “EpiChord: Parallelizing the Chord lookup algorithm with reactive routing state management,” Proc. 12th IEEE International Conference on Networks (ICON 2004), pp.270–276, 2004.

[30] B. Zhao, Y. Duan, L. Huang, A. Joseph, and J. Kubiatowicz, “Brocade: Landmark routing on overlay networks,” 1st International Workshop on Peer-to-Peer Systems (IPTPS), pp.34–44, March 2002.

[31] D.R. Karger and M. Ruhl, “Diminished Chord: A protocol for heterogeneous subgroup formation in peer-to-peer networks,” 3rd International Workshop on Peer-to-Peer Systems (IPTPS), pp.288–297, 2004.

[32] Y. Zhang, D. Li, L. Chen, and X. Lu, “Flexible routing in grouped DHTs,” Proc. 2008 Eighth International Conference on Peer-to-Peer Computing, pp.109–118, 2008.

[33] I. Martinez-Yelmo, A. Bikfalvi, C. Guerrero, R. Cuevas, and A. dreasMauthe, “Enabling global multimedia distributed services based on hierarchical DHT overlay networks,” Proc. 2008 The Second International Conference on Next Generation Mobile Applications, Services, and Technologies, pp.543–549, 2008.
[34] Z. Ou, E. Harjula, T. Koskela, and M. Ylianttila, “General truncated pyramid peer-to-peer architecture over structured DHT networks,” Mobile Networks and Applications, vol.15, no.5, pp.729–749, 2009.
[35] 黒宮佑介，構造化 p2p オーバーレイネットワークにおけるオブジェクトの属性を用いた高速な選択的データ配送，Master’s thesis，慶應義塾大学大学院政策・メディア研究科，March 2011.
[36] H. Shen and C.-Z. Xu, “Hash-based proximity clustering for efficient load balancing in heterogeneous DHT networks,” J. Parallel Distrib. Comput., vol.68, no.5, pp.686–702, 2008.
[37] K. Park, S. Pack, and T. Kwon, “Proximity based peer-to-peer overlay networks (P3ON) with load distribution,” ICOIN 2007 Revised Selected Papers, pp.234–243, 2008.
[38] 高野祐輝，井上朋哉，知念賢一，篠田陽一，“NAT 問題フリーな DHT を実現するライブラリ libcage の設計と実装,” コンピュータソフトウェア，vol.27, no.4, pp.58–76, 2010.
[39] K.P.N. Puttaswamy and B.Y. Zhao, “A case for unstructured distributed hash tables,” Proc. IEEE Global Internet Symposium, pp.7–12, 2007.
[40] J. Travers and S. Milgram, “An experimental study of the small world problem,” Sociometry, vol.32, pp.425–443, 1969.
[41] W. Pugh, “Skip lists: A probabilistic alternative to balanced trees,” Commun. ACM, vol.33, no.6, pp.668–676, 1990.
[42] L.R. Monnerat and Cláudio L. Amorim, “Peer-to-peer single hop distributed hash tables,” GLOBECOM, pp.1–8, 2009.
[43] D.A. Bryan, P. Matthews, E. Shim, D. Willis, and S. Dawkins, “Concepts and terminology for peer to peer SIP,” Internet-Draft, July 2008.
[44] Intel Corporation, “Place Lab,” 2006. Available electronically at http://ils.intel-research.net/place-lab
[45] Y. Chawathe, S. Ramabhadran, S. Ratnasamy, A. LaMarca, S. Shenker, and J. Hellerstein, “A case study in building layered DHT applications,” Proc. ACM SIGCOMM, pp.97–108, 2005.
[46] S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu, “OpenDHT: A public DHT service and its uses,” Proc. ACM SIGCOMM, pp.73–84, 2005.
[47] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina, “The EigenTrust algorithm for reputation management in P2P networks,” Proc. Twelfth International World Wide Web Conference, pp.640–651, 2003.
[48] S. Marti and H. Garcia-Molina, “Taxonomy of trust: Categorizing P2P reputation systems,” Comput. Netw., vol.50, no.4, pp.472–484, 2006.
[49] D. Korzun and A. Gurtov, “A local equilibrium model for P2P resource ranking,” ACM SIGMETRICS Performance Evaluation Review, vol.37, no.2, pp.27–29, 2009.
[50] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica, “The impact of DHT routing geometry on resilience and proximity,” Proc. ACM SIGCOMM, pp.381–394, 2003.
[51] M.J. Freedman, K. Lakshminarayanan, S. Rhea, and I. Stoica, “Non-transitive connectivity and DHTS,” Proc. 2nd conference on Real, Large Distributed Systems - Volume 2 (WORLDS ’05), pp.55–60, 2005.
[52] S. Rhea, “The bamboo DHT,” as of 2005. Available electronically at http://bamboo-dht.org/
[53] The Trustees of Princeton University, “Planetlab – an open platform for developing, deploying, and accessing planetary-scale services,” Available electronically at http://www.planet-lab.org/

（平成 24 年 9 月 7 日受付，25 年 1 月 14 日再受付）

斉藤　賢爾　（正員）

1993 コーネル大学大学院工学修士課程了．2006 慶應義塾大学より博士号取得．博士（政策・メディア）．現在，慶應義塾大学大学院政策・メディア研究科特任講師としてインターネットと社会の諸問題に関する研究に従事．

高野　祐輝

2005 北陸先端科学技術大学院大学情報科学研究科博士前期課程了．2011 北陸先端科学技術大学院大学より博士号取得．博士（情報科学）．現在，情報通信研究機構ネットワークセキュリティ研究所セキュリティアーキテクチャ研究室研究員．