一定次数のオーバレイネットワークにおける最適経路長の経路決定方法の考察

島 和之[†] 松尾浩平[†,☆] 大場充[†]
佐藤康臣[†] 清水將吾[†]

分散ハッシュ表（DHT）のためのオーバレイネットワーク，および，その上で最適な経路を求める方法を提案する．DHT は，P2P 型システムのような分散システムにおいて，リソースを効率的に検索するための仕組みである．提案するオーバレイネットワークについて確率論的な評価を行い，平均的な次数がノード数によらず一定であること，および，ノード数 n に対して経路長が $O(\log n)$ となることを示した．次数が一定であることから，ノード数が多い場合でも，ノードの参加や離脱のとき，隣接ノードを更新するための負荷が高くならず，スケーラブルなシステムを構築できると考えられる．また，従来手法の経路長は $O(\log n)$ 以上であることから，提案手法の経路長は最適であり，従来と同等以下の時間でリソースを検索できると考えられる．

A Study of Optimal Path Routing in Constant Degree Overlay Network

Kazuyuki Shima,[†] Kohei Matsuo,[†,☆] Mitsuru Ohba,[†]
Yasuomi Sato[†] and Shogo Shimizu[†]

We propose an overlay network for Distributed Hash Table (DHT) and an optimal path routing in the overlay network. DHTs are mechanisms to efficiently locate nodes that provide a particular resource in distributed systems such as peer-to-peer (P2P) systems. Probabilistic evaluation of the proposed overlay network shows the expected degree is constant not depending on the number of nodes, and the expected path length is $O(\log n)$ where the number of nodes is n. The proposed overlay network is scalable, that is, the load of updating neighbors does not become high when the number of nodes becomes large because the degree is constant. The proposed routing can locate nodes within time equals to or shorter than previous works, that is, the path length of the proposed routing is optimal because the path lengths of previous works are not less than $O(\log n)$.

1. はじめに

近年，インターネット上で不特定多数の個人同士がファイルを交換するためのシステムとして P2P 型システムが注目されている．P2P（peer-to-peer）型システムとは，水平型の分散処理システム[9] の一種であり，次の機能を持つものである．

- 不特定多数のノードが随時参加する．
- システムに参加している任意のノードが随時離脱する．
- 各ノードが持っているリソースを他のノードに提供する．
- 指定されたリソースを持つノードを検索する．

クライアント・サーバ型の場合，特定のサーバが多数のクライアントにリソースを提供するので，負荷がサーバに集中し，サーバが停止すると全クライアントがリソースを利用できなくなるという問題がある．P2P 型の場合，多数のノードが相互にリソースを提供するので，負荷が多数のノードに分散し，特定のノードが停止しても他のノードが持つリソースを利用できる．

P2P 型システムにおいて，リソースを持つノードを検索するための仕組みとして分散ハッシュ表が研究されている[1),7)]．分散ハッシュ表（Distributed Hash Table: DHT）とは，複数のノードが分担して記憶するハッシュ表であり，次の機能を持つものである．

- キーの集合を分割し，ノードに割り当てる．
- ノードの参加や離脱のとき，ノードに割り当てる

† 広島市立大学
Hiroshima City University
☆ 現在，株式会社ジャステック
Presently with JASTEC

キーの集合を更新する．
- 指定されたキーの担当ノード（そのキーを割り当てられたノード）を検索する．
- 指定されたキーの担当ノードが，そのキーに対応する値を登録または取得する．

DHT を用いてリソースを持つノードを検索するためには，あらかじめ，リソースのキーの担当ノードへ，そのリソースを持つノードのアドレスを登録しておく．リソースを持つノードを検索するときは，指定されたリソースのキーに対応する値を，そのキーの担当ノードから取得することによって，そのリソースを持つノードのアドレスを知ることができる．

DHT は，リソースの検索だけでなく，DNS[5] と同様のサービスにも応用できる．DNS は，ホスト名に対する IP アドレスを調べるサービスを提供する．DHT において，ホスト名をキーとし，IP アドレスを値としてあらかじめ登録しておけば，ホスト名に対する IP アドレスを取得できる．DNS では複数のサーバが形成する木構造においてルートとなる特別なサーバが必要であるが，DHT では特別なサーバは必要ない．DNS ではサーバ間の経路情報を手作業で管理する必要があるが，DHT では自動的に管理できる．DNS で扱うホスト名は管理上の境界（ドメイン）に従って構造化されているが，DHT で扱うホスト名には制限がない．たとえば，あるノードのリソースを別のノードに移転したとき，または，ノードが移動したとき，または，ノードが通信方法を切り替えたときなどに，DNS ではホスト名を変更する必要が生じる場合がある．しかし，DHT ではホスト名を変更する必要がなく，ユーザは同じホスト名を使ってノードにアクセスすることができる．よって，DHT を用いることによって DNS よりもシームレスなサービスを提供できる．

DHT は，キーの担当ノードを検索するために，オーバレイネットワークを構築する．DHT が構築するオーバレイネットワークは，強連結☆な有向グラフであり，任意のノードから指定されたキーの担当ノードへの経路を決定できる．キーの担当ノードを検索するノードは，そのキーを指定して検索メッセージを送信する．検索メッセージを受信したノードは，オーバレイネットワークに従って経路を決定し，隣接ノード☆☆へ検索メッセージを中継する．担当ノードが，検索メッセージを受信すると，自分のアドレスを返信する．

本論文では，平均的な次数がノード数によらず一定なオーバレイネットワーク，および，その上で経路長がなるべく短い経路を求める方法を提案する．次数（degree）とは，各ノードの隣接ノードの数である．ノードの参加や離脱のとき，隣接ノードを更新する処理を減らすためには，次数が少ないほど良い．提案するオーバレイネットワークでは，個々のノードによって次数のばらつきがあるが，次数の平均値はノード数によらず一定である．よって，ノード数が増えても，ノードの参加や離脱のとき，1 つのノードにかかる負荷は増えない．経路長（path length）とは，2 つのノード間の経路の長さである．担当ノードを検索する時間をなるべく短くするためには，経路長が短いほど良い．ただし，一般の有向グラフ上の最短経路を求める方法（e.g. ダイクストラ法）は，すべての有向辺を知る必要があるので，P2P 型システムには適さない．提案手法では，ノード数 n に対して経路長は $O(\log n)$ となる．

☆ 任意の 2 つのノード間に経路があること
☆☆ あるノードから他のノードの中継なしにメッセージを受信できるノード

2. 提案手法

2.1 仮定

本提案手法では，ノード，および，下位の通信ネットワークの性質を以下のように仮定する．
- 複数のノードは並列動作する．
- ノードは故障，停電，強制終了，省電力機能などによって，予告なく停止することがある．
- ノードはオーバレイネットワーク以外の処理も並行して行っている場合があり，それらの処理のためにオーバレイネットワークのための処理が遅延することがある．
- ノードが予告なく停止している状態と遅延している状態を他のノードは見分けることはできず，そのノードが離脱していると見なす．
- 下位の通信ネットワークはノードが離脱しても強連結を維持できる．
- 各ノードは，そのノードがアドレスを知っているノードのみへ，他のノードを中継せず，メッセージを直接に送信できる．
- メッセージの消滅，複製はなく，送信されたメッセージだけが正確に一度だけ受信される．
- 送信されたメッセージは，送信後，有限時間内に受信される．
- 受信されたメッセージは，FIFO の受信キューに保存される．
- アドレス空間はノード数に対して非常に広く，すべてあるいは無作為に選んだアドレスへメッセージを送信してノードを発見することは困難である．

2.2 キーの割当て

概念的にキーとノードを円周上に配置する．キーの位置はハッシュ関数を用いて決める．ノードの位置は擬似乱数を用いて無作為に決める．ノードが複数のとき，ノードを境界として円周を複数の円弧に分割する．各円弧について，時計回りの始点にあるノードに，終点を除く円弧上にあるキーを割り当てる．ただし，ノードが 1 つのときは，そのノードにすべてのキーを割り当てる．

円周上のキーまたはノードを数学的に表現するため，それらの位置を基点から時計回りの円弧の長さによって示す．基点とは円周上に定めた 1 点である．円周の長さは 1 とする．一般にキーは任意のデータであり，キーそのものとキーの位置とは異なるが，本論文では記述を簡潔にするため，「円周上の位置 k にあるキー」を「キー k」と略記する．また，同様に，ノードそのものとノードの位置とは異なるが，「円周上の位置 v にあるノード」を「ノード v」と略記する．

次のように定義する．

$\mathbb{R}$：実数の集合である．

$K = \{k \in \mathbb{R} | 0 \leq k < 1\}$：キーの位置の集合である．

$V \subset K$：ノードの位置の有限集合である．

$\mathbb{Z}$：整数の集合である．

$\lfloor x \rfloor = \max\{j \in \mathbb{Z} | j \leq x\}$：$x \in \mathbb{R}$ 以下の最大の整数を返す関数（床関数）である．

$\mathrm{mod}\,(x, y) = x - \lfloor x/y \rfloor y$：実数 x，$y \in \mathbb{R}$ の剰余を返す関数である．特に，$x \geq 0$ のとき，$\mathrm{mod}\,(x, 1) = x - \lfloor x \rfloor$ は x の小数部分と等しい．

区域 $A(x, r)$：円周上を基点から時計回りに長さ $x \in \mathbb{R}$ 回った位置 $\mathrm{mod}\,(x, 1)$ から時計回りの長さが $r \in \mathbb{R}$ 未満のキーまたはノードの集合である．定理 1，および，$\mathrm{mod}\,(y - x, 1) = \mathrm{mod}\,(y - \mathrm{mod}\,(x, 1), 1)$ より，$A(x, r) = \{y \in K | \mathrm{mod}\,(y - x, 1) < r\}$．

$|T(x)|$：ノードを境界として円周を円弧に分割したとき，ノード $x \in V$ が時計回りの始点にある円弧の長さである．特に，ノードが 1 つのとき，$|T(x)| = 1$．定理 2 より，$|T(x)| = 1 - \max\{\mathrm{mod}\,(x - v, 1) | v \in V\}$．

担当区域 $T(x) = A(x, |T(x)|)$：ノードを境界として円周を円弧に分割したとき，ノード $x \in V$ が時計回りの始点にある円弧であり，ノード x に割り当てるキーの集合である． □

図 1 は，区域と担当区域を示す．区域 $A(x, r)$ は，$x \in K$ から $x + r$ まで時計回りの円弧上のキーの集合を示す．$x_1, x_2, \cdots, x_5$ はノードを示す．担当区域 $T(x_3)$ は，ノード x_3 からノード x_4 まで時計回りの円弧上のキーの集合を示す．

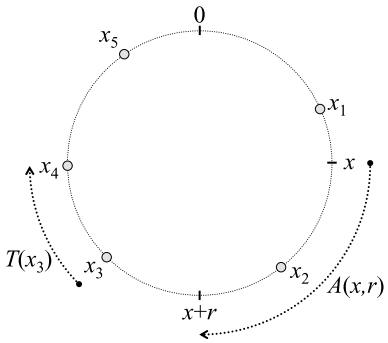

図 1 区域と担当区域
Fig. 1 Area and territory.

次の定理が成り立つ．

定理 1 キーまたはノード x，$y \in K$ について，x から y まで時計回り（あるいは，y から x まで反時計回り）の円弧の長さは $\mathrm{mod}\,(y - x, 1)$ である．

証明： x から y まで時計回りの円弧の長さを r とする．$x \leq y$ のとき，$r = y - x = \mathrm{mod}\,(y - x, 1)$．$x > y$ のとき，x から y までの長さと y から x までの長さの合計が円周の長さ 1 となることから，$r = 1 - (x - y) = 1 - x + y$．$x < 1, y \geq 0$ より，$-1 < y - x < 0$．$\lfloor y - x \rfloor = -1$ より，$\mathrm{mod}\,(y - x, 1) = y - x - \lfloor y - x \rfloor = y - x + 1$．よって，$r = \mathrm{mod}\,(y - x, 1)$． □

定理 2 任意のノード $x \in V$ について，$|T(x)| = 1 - \max\{\mathrm{mod}\,(x - v, 1) | v \in V\}$．

証明：定理 1 より，ノード v からノード x まで時計回りの円弧の長さは $\mathrm{mod}\,(x - v, 1)$ である．よって，任意のノードのうちのあるノード $y \in V$ からノード x まで時計回りの円弧の長さが最大のとき，その長さは $\mathrm{mod}\,(x - y, 1) = \max\{\mathrm{mod}\,(x - v, 1) | v \in V\}$ である．このとき，すべてのノードが y から x まで時計回りの円弧上にある，すなわち，x から y まで時計回りの円弧上に他のノードがない．よって，ノードを境界として円周を分割したとき，ノード x が時計回りの始点にある円弧が存在し，その長さは $1 - \mathrm{mod}\,(x - y, 1) = 1 - \max\{\mathrm{mod}\,(x - v, 1) | v \in V\}$． □

ノードが複数ならば，$|T(x)| = \min\{\mathrm{mod}\,(v - x, 1) | v \in V, v \neq x\}$ としても求まる．しかし，ノードが 1 つならば $|T(x)| = 1$ としなければならない．定理 2 により，場合分けせずに $|T(x)|$ を求めることができる．

2.3 オーバレイネットワーク

提案するオーバレイネットワークを有向グラフとし

て表現する．この有向グラフの頂点はノードに対応し，頂点 x から頂点 y への有向辺は「ノード x がノード y のアドレスを知っていること」を意味する．

次のように定義する．

前ノード $v_{-1}(x)$：キーまたはノード $x \in K$ から反時計回りに最初のノードである．ノードが複数のとき，$v_{-1}(x) \neq x$ である．ノードが 1 つのとき，$v_{-1}(x) = x$ である．

後ノード $v_{+1}(x)$：キーまたはノード $x \in K$ から時計回りに最初のノードである．ノードが複数のとき，$v_{+1}(x) \neq x$ である．ノードが 1 つのとき，$v_{+1}(x) = x$ である．

担当ノード $v_{\pm}(k) = v_{-1}(v_{+1}(k))$：キー $k \in K$ を割り当てられるノードである．ノードが k にない限り，k の前ノードが担当ノードである．しかし，ノードが k にある場合，そのノードが担当ノードである．そこで，k の後ノードの前ノードを担当ノードと定義する．

$E_{\pm} = \{(x, y) \in V \times V | y = v_{-1}(x) \vee y = v_{+1}(x)\}$：任意のノードからそれぞれの前ノードと後ノードへの有向辺の集合である．

子ノード：あるノードに割り当てられたキーを b 倍したキーの担当ノードである．すなわち，ノード x，$y \in V$ が $^{\exists}k \in T(x) : \mathrm{mod}\,(bk, 1) \in T(y)$ を満たすとき，ノード y をノード x の子ノードと呼ぶ．ただし，b は 2 以上の任意の整数であるが，すべてのノードにおいて同一であり，システムの実行中に変化しない定数とする．

$V_*(b, x)$：ノード $x \in V$ の子ノードの集合である．定義より，$V_*(b, x) = \{y \in V | ^{\exists}k \in T(x) : \mathrm{mod}\,(bk, 1) \in T(y)\}$ である．

$E_*(b) = \{(x, y) \in V \times V | y \in V_*(b, x)\}$：ノードからその子ノードへの有向辺の集合である． □

図 2 は，前ノード，後ノード，担当ノードを示す．ノード x_1 の前ノードは x_5，後ノードは x_2 である．キー x の前ノードは x_1，後ノードは x_2 である．ノード x_2 の前ノードは x_1 である．よって，キー x の担当ノードは x_1 である．**図 3** は，ノード $x_3 \in V$ の子ノードを示す．ノード x_3 の担当区域の長さを r とおくと，区域 $A(bx_3, br)$ を担当区域に含むノード，すなわち，ノード x_1 と x_2 がノード x_3 の子ノードとなる．

提案するオーバレイネットワークは，2 つのネットワーク $(V, E_{\pm})$ と $(V, E_*(b))$ を組み合わせたネットワーク $G = (V, E_{\pm} \cup E_*(b))$ である．ネットワーク $(V, E_*(b))$ は，ノードが多い場合でも，キーの担当ノードを効率良く検索することを目的とする．ただし，このネットワークだけでは，ノードの参加や離脱におけるキーの割当てや隣接ノードの更新が難しい．ネットワーク $(V, E_{\pm})$ は，キーの割当ての更新とネットワーク $(V, E_*(b))$ の隣接ノードの更新を目的とする．また，ネットワーク $(V, E_{\pm})$ 自体の隣接ノードの更新は容易である．ただし，ネットワーク $(V, E_{\pm})$ だけでは，ノードが多い場合，キーの担当ノードを効率良く検索することができない．

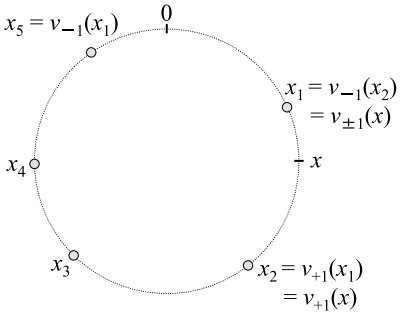

図 2 前ノード，後ノード，担当ノード
Fig. 2 The previous node, the next node and the responsible node.

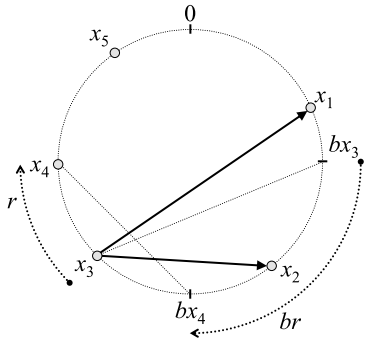

図 3 子ノード
Fig. 3 The child nodes.

以降の 2.4 節ではネットワーク $(V, E_{\pm})$ の隣接ノード（前ノードと後ノード）を更新する方法について，2.5 節ではネットワーク $(V, E_*(b))$ の隣接ノード（子ノード）を更新する方法について述べる．

2.4 前ノードと後ノードの更新

隣接ノードと通信するために，各ノードはノード表を持つ．ノード表はノード情報の集合である．ノード情報は下位の通信ネットワークにおけるノードのアドレスと円周上におけるノードの位置を含む情報である．ノード表にノード情報を追加したとき，あるいは，ノード表からノード情報を削除したとき，前ノードまたは後ノードを更新する．キーまたはノード $k \in K$ の前ノード $v_{-1}(k)$ と後ノード $v_{+1}(k)$ は，次式によっ

て求めることができる．

$$v_{-1}(k) = \begin{cases} \max\{v \in V | v < k\} \\ \qquad \Leftarrow^{\exists}\ v \in V : v < k \\ \max V \Leftarrow^{\forall}\ v \in V : v \geq k \end{cases}$$

$$v_{+1}(k) = \begin{cases} \min\{v \in V | v > k\} \\ \qquad \Leftarrow^{\exists}\ v \in V : v > k \\ \min V \Leftarrow^{\forall}\ v \in V : v \leq k \end{cases}$$

次の定理により，場合分けせずに前ノードと後ノードを選ぶことができる．

定理 3

(1) キーまたはノード $x \in K$ から時計回りに最も遠いノードは x の前ノードである．すなわち，$u \in V \wedge {}^{\forall}v \in V : \mathrm{mod}(u - x, 1) \geq \mathrm{mod}(v - x, 1) \Rightarrow u = v_{-1}(x)$.

(2) キーまたはノード $x \in K$ から反時計回りに最も遠いノードは x の後ノードである．すなわち，$u \in V \wedge {}^{\forall}v \in V : \mathrm{mod}(x - u, 1) \geq \mathrm{mod}(x - v, 1) \Rightarrow u = v_{+1}(x)$.

証明：ノード $u \in V$ が ${}^{\forall}v \in V : \mathrm{mod}(u - x, 1) \geq \mathrm{mod}(v - x, 1)$ を満たすとき，定理 1 より，x からノード u までの長さが，x から任意のノードまで時計回りの円弧の長さ以上であるので，x から u まで時計回りの円弧上にすべてのノードが存在する．このとき，x から u まで反時計回りの円弧上に他のノードが存在しないので，ノード u は x から反時計回りに最初のノード，すなわち，x の前ノード $v_{-1}(x)$ である．よって，(1) が成り立つ．(2) についても同様である． □

図 4 は，前ノードと後ノードの更新におけるノードの状態遷移を示す．ノード x は，開始状態（黒丸）から前ノード通知送信状態 S1 と受信待ち状態 S2 へ並行して遷移する．前ノード通知送信状態 S1 では，前ノード通知メッセージ $M-$ を，それぞれの後ノード x_{+1} へ，定期的に繰り返し送信する．前ノード通知メッセージには，送信ノードのノード情報を記す．受信待ち状態 S2 では，受信キューから取り出したメッセージによって遷移先を決定する．ただし，受信キューが空の間は，新たに受信したメッセージが受信キューに追加されるまで待機する．受信待ち状態 S2 において前ノード通知メッセージ $M-$ を受信キューから取り出した場合，前ノード通知受信状態 S3 へ遷移する．前ノード通知受信状態 S3 では，送信ノード x_{-1} のノード情報をノード表に追加した後，後ノード通知メッセージ $M+$ をノード x_{-1} へ返信し，受信待ち状態 S2 へ遷移する．後ノード通知メッセージには送信ノードとその前後のノードのノード情報を記す．受信待ち状態 S2 において後ノード通知メッセージ $M+$ を受信キューから取り出した場合，後ノード通知受信状態 S4 へ遷移する．後ノード通知受信状態 S4 において送信ノード x_{+1} とその前後のノードのノード情報をノード表に追加し，受信待ち状態 S2 へ遷移する．

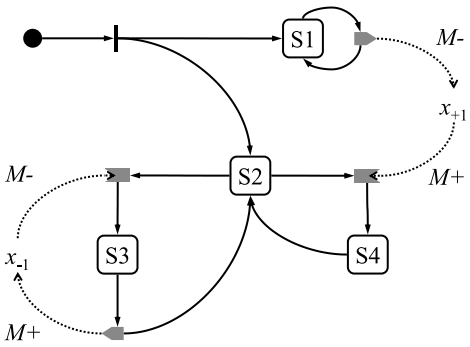

図 4 前後ノードの更新における状態遷移
Fig. 4 The state chart for updating the previous node and the next node.

前ノードから前ノード通知メッセージを最後に受信してから一定時間以上経過したとき，前ノードが離脱したと見なし，ノード表から前ノードを除き，前ノードを更新する．前ノード通知メッセージを送信してから一定時間以上経過しても後ノード通知メッセージが返信されないとき，後ノードが離脱したと見なし，ノード表から後ノードを除き，後ノードを更新する．

図 5 は，ノード z が離脱するとき，その前ノード x と後ノード y が送信するメッセージの流れを示している．このとき，次の手順に従って，x の後ノードと y の前ノードを更新する．

(1) z が離脱する前は，x は z へ，z は y へ前ノード通知メッセージを送信し，z は x へ，y は z へ後ノード通知メッセージを返信する（図 5 (a)）．

(2) ノード z が離脱した後は，x が z へ前ノード通知メッセージを送信しても z は後ノード通知メッセージを返信せず，z は y へ前ノード通知メッセージを送信しない（図 5 (b)）．

(3) x が z へ前ノード通知メッセージを送信してから一定時間以上経過しても z が応答しないとき，z が離脱していると x は見なす．

(4) y が z から前ノード通知メッセージを最後に受信してから一定時間以上経過しても z から前ノード通知メッセージを新たに受信しないとき，z が離脱していると y は見なす．

(5) z が離脱していると見なしたとき，x は後ノー

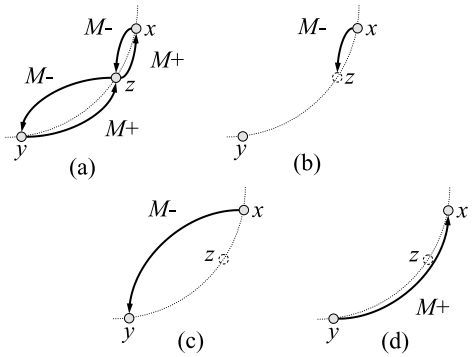

図 5 ノードの離脱におけるメッセージの流れ
Fig. 5 Message flow at leaving of a node.

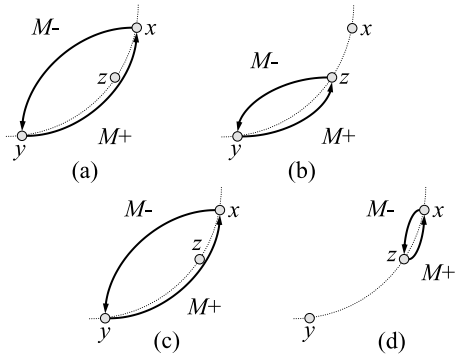

図 6 離脱したノードの再参加におけるメッセージの流れ
Fig. 6 Message flow at rejoining of a node left.

ドを y に更新し，前ノード通知メッセージを y へ送信する（図 5 (c)）．

(6) z が離脱していると y が見なす前に，x から前ノード通知メッセージを受信した場合，y は前ノードを z としたまま，後ノード通知メッセージを x へ返信するので，x は後ノードを z に更新し，手順 (2) へ戻る．

(7) z が離脱していると y が見なした後に，x から前ノード通知メッセージを受信した場合，y は前ノードを x に更新し，x へ後ノード通知メッセージを x へ返信する（図 5 (d)）．

図 6 は，前ノード x と後ノード y が離脱していると認識したノード z が実際には停止していなかった場合に再参加するときのメッセージの流れを示している．このとき，次の手順に従って，x の後ノードと y の前ノードを更新する．

(1) z が離脱していると認識した後，x は y へ前ノード通知メッセージを送信し，y は x へ後ノード通知メッセージを返信する（図 6 (a)）☆．

(2) z は離脱していると認識されていることに気づかず，前ノード通知メッセージを後ノードである y へ送信し，z から前ノード通知メッセージを受信した y は，z をノード表に追加し，前ノードを z に更新し，後ノード通知メッセージを z へ返信する（図 6 (b)）．

(3) x から前ノード通知メッセージを受信した y は，後ノード通知メッセージに前ノードとして z を記入し，x へ返信する（図 6 (c)）．

(4) y から後ノード通知メッセージを受信した x は，後ノードを z に更新する．

(5) x は z へ前ノード通知メッセージを送信し，z は x へ後ノード通知メッセージを返信する．

後ノード通知メッセージを受信したノードは，その送信ノードの前ノードと自分のノード情報を比較し，アドレスが異なり，位置が同じとき，ノードの衝突を検知し，移動（位置を変更）する．ノードの位置の桁数を大きくすることによって，ノードが衝突する確率を小さくはできるが，0 にはできない．もし一度衝突すると，どちらかが移動しない限り衝突を繰り返す．そこで，後ノードが前ノード通知メッセージを受信したタイミングが，早い方は移動せず，遅い方が移動することによって再衝突を避ける．ノード x と衝突していることをノード z が検知したとき，ノード z の移動先 z' を決める方法として次の 2 つが考えられる．

- 負荷を均等に分散させるために，ノード x に登録されているキーと値の対応の個数が等分されるように移動先 z' を定める．
- キーが一様に分布すると仮定し，ノード x からノード y までの中間を移動先 $z' = \mathrm{mod}(x + |T(x)|/2, 1)$ とする．

前者の場合，x に登録されているすべてのキーの位置を知る必要があるので，z が x へ移動先 z' を問い合わせる必要がある．また，移動先 z' を決める時点で登録されているキーの位置に偏りがあったとしても，その後，キーが一様に登録されれば，移動先 z' の偏りによって負荷の偏りが生じる恐れもある．よって，後者の方法をとり，ノード間の中間を移動先 z' とする．

2.5 子ノードの更新

図 7 は，子ノードの更新におけるノード x の状態遷移を示している．ノード x は，開始状態（黒丸）から子ノード検索送信状態 S5 と受信待ち状態 S2 へ並行して遷移する．子ノード検索送信状態 S5 では，子ノード検索メッセージ $M*$ を，時計回りに最初の子ノード $v_{\pm}(bx)$，あるいは，自分 x へ定期的に送信する．

☆ このとき，x が z へ前ノード通知メッセージを送信しないので x が離脱していると z が見なす可能性がある．しかし，x の前ノードが z へ前ノード通知メッセージを送らないので他のノードには影響しない．

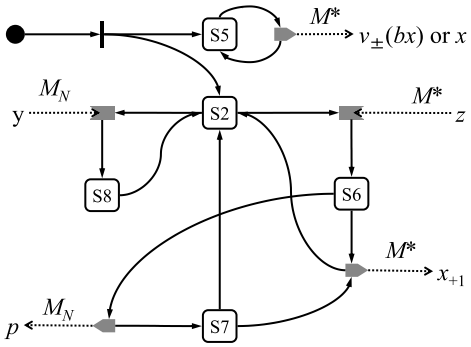

図 7 子ノードの更新における状態遷移
Fig. 7 The state chart for updating child nodes.

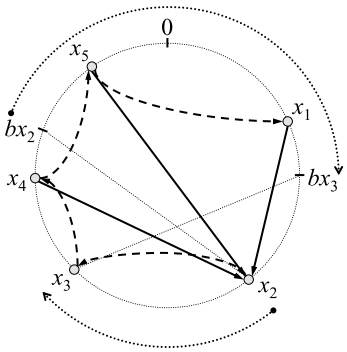

図 8 子ノードの更新におけるメッセージの流れ
Fig. 8 Message flow at updating child nodes.

子ノード検索メッセージには，親ノード p とその後ノードのノード情報を記す．子ノード検索メッセージを x よりも $v_{\pm}(bx)$ へ送信する方が子ノードを早く検索できるので，なるべく $v_{\pm}(bx)$ へ送信する．しかし，$v_{\pm}(bx)$ がノード表にない，あるいは，離脱している場合，x へ送信する．受信待ち状態 S2 は図 4 の S2 と同じ状態であり，受信キューから取り出したメッセージによって遷移先を決定する．受信待ち状態 S2 において子ノード検索メッセージ $M*$ を取り出した場合，子ノード検索受信状態 S6 へ遷移する．子ノード受信状態 S6 において x が p の子ノードならば，p へノード通知メッセージ M_N を送信し，子ノード通知状態 S7 へ遷移する．ノード通知メッセージには，送信ノードとその後ノードのノード情報を記す．子ノード受信状態 S6 において x が p の子ノードでなければ，後ノード x_{+1} へ子ノード検索メッセージ $M*$ を転送し，受信待ち状態 S2 へ遷移する．子ノード通知状態 S7 において x_{+1} が p の子ノードならば，x_{+1} へ子ノード検索メッセージ $M*$ を転送し，受信待ち状態 S2 へ遷移する．子ノード通知状態 S7 において x_{+1} が p の子ノードでなければ，受信待ち状態 S2 へ遷移する．受信待ち状態 S2 においてノード通知メッセージ M_N を取り出した場合，ノード通知受信状態 S8 へ遷移する．ノード通知受信状態 S8 では，ノード通知メッセージの送信ノードである子ノード y をノード表に追加し，子ノードとその後ノードの間のノードは離脱していると見なし，ノード表から削除し，受信待ち状態 S2 へ遷移する．子ノード検索メッセージを $v_{\pm}(bx)$ へ送信してから一定時間以上経過しても，$v_{\pm}(bx)$ からノード通知メッセージを受信しない場合，$v_{\pm}(bx)$ が離脱していると見なし，ノード表から削除する．

図 8 は，ノード x_x の子ノードの更新におけるメッセージの流れを示している．

(1) x_2 は子ノード検索メッセージを定期的に自分へ送信する．

(2) 子ノード検索メッセージを受信したノードは，自分が子ノードでなければ後ノードへ転送する（破線の矢印 $x_2 \to x_3 \to x_4$）．

(3) 子ノード検索メッセージを受信したノードは，自分が子ノードであればノード通知メッセージを親ノードへ送信する（実線の矢印 $x_4, x_5, x_1 \to x_2$）．

(4) 子ノード検索メッセージを受信したノードは，自分が子ノードであり，後ノードも子ノードであれば，子ノード検索メッセージを後ノードへ転送する（破線の矢印 $x_4 \to x_5 \to x_1$）．

2.6 担当ノードの検索

キーの担当ノードを検索するためにはネットワーク $(V, E_*(b))$ 上で担当ノード検索メッセージを担当ノードまで中継する．まず，担当ノードを検索するノードは，キーを指定して担当ノード検索メッセージを自分に送信する．担当ノードが担当ノード検索メッセージを受信すると，自分のアドレスを返信する．担当ノードではないノードが担当ノード検索メッセージを受信すると，子ノードの中から担当ノードまでの経路が短いノードを選び，メッセージを中継する．子ノードから担当ノードまでの経路の長さは，次の補題と定理より求まる．

補題 4 $^{\forall}x, r \in \mathbb{R}, ^{\forall}k \in A(bx, br), ^{\exists}k' \in A(x, r) : k = \mathrm{mod}\,(bk', 1)$.

証明：$k \in A(bx, br)$ より，$\mathrm{mod}\,(k - \mathrm{mod}\,(bx, 1), 1) < br$．$i = \lfloor bx \rfloor$，$j = \lfloor k - \mathrm{mod}(bx, 1) \rfloor$ とおくと，$\mathrm{mod}\,(k - \mathrm{mod}\,(bx, 1), 1) = k - (bx - i) - j = k - bx + i - j$．$k' = (k + i - j)/b$ とおくと，$\mathrm{mod}(k - \mathrm{mod}(bx, 1), 1) = bk' - bx < br$．$0 \le \mathrm{mod}\,(k - \mathrm{mod}\,(bx, 1), 1) < 1$ より，$0 \le k' - x < 1/b < 1$．よって，$\mathrm{mod}\,(k' - x, 1) = k' - x < r$．このとき，$k' \in A(x, r)$，$k = \mathrm{mod}\,(bk', 1)$． □

補題 5 $^{\forall}x \in V, ^{\forall}k \in A(bx, b|T(x)|) : v_{\pm}(k) \in V_*(b, x)$.

証明：補題 4 より，$k = \bmod(bk', 1)$ を満たす $k' \in A(x, |T(x)|)$ が存在する．$k' \in T(x)$ は $k = \bmod(bk', 1) \in T(v_{\pm}(k))$ を満たすので，子ノードの定義より，ノード $v_{\pm}(k)$ はノード x の子ノードである．□

定理 6 ノード $x \in V$ とキー $k \in K$ について $^{\exists}L \in \mathbb{Z}, L \geq 0 : k \in A(b^L x, b^L |T(x)|)$ ならば，ノード x からキー k の担当ノード $v_{\pm}(k)$ までの長さが L 以内の経路が存在する．

証明：以下から数学的帰納法により，任意の $L \geq 0$ について定理 6 が成立する．

(1) $L = 0$ のとき，$k \in A(x, |T(x)|)$ より，ノード x がキー k の担当ノードであり，経路の長さは 0 であるので，この定理が成り立つ．

(2) $L > 0$ のとき，$k' \in A(b^{L-1}x, b^{L-1}|T(x)|)$ ならば，ノード x からキー k' の担当ノード $v_{\pm}(k')$ までの長さが $L-1$ 以内の経路が存在すると仮定する．補題 5 より，$k \in A(b^L x, b^L |T(x)|)$ ならば，ノード $v_{\pm}(k)$ はノード $v_{\pm}(k')$ の子ノードである．よって，ノード x からノード $v_{\pm}(k)$ までの長さが L 以内の経路が存在する．□

定理 6 より，ノード $x \in V$ は，その子ノード $y \in V_*(b, x)$ から担当ノードまでの経路の長さ $L_*(b, y, k) = \min\{L \in \mathbb{Z} | L \geq 0, k \in A(b^L y, b^L |T(y)|)\}$ を求め，経路の長さが最小となる子ノードへメッセージを中継する．

図 9 は，$b = 2$ においてノード 8/64 がキー 54/64 の担当ノードを検索するときのメッセージの経路を示す．ノード 8/64 の担当区域は $T(8/64) = A(8/64, 14/64 - 8/64)$．区域 $A(2\times 8/64, 2\times 6/64) = A(16/64, 12/64)$ を担当区域に含むノード 14/64, 21/64 がノード 8/64 の子ノードである．$|T(14/64)| = 21/64 - 14/64 = 7/64$ より，

$$
\begin{aligned}
&54/64 \notin A\,(14/64, 7/64)\\
&54/64 \notin A\,(2 \times 14/64, 2 \times 7/64)\\
&\qquad = A\,(28/64, 14/64)\\
&54/64 \notin A\left(2^2 \times 14/64, 2^2 \times 7/64\right)\\
&\qquad = A\,(56/64, 28/64)\\
&54/64 \in A\left(2^3 \times 14/64, 2^3 \times 7/64\right)\\
&\qquad = A\,(48/64, 56/64)
\end{aligned}
$$

$|T(21/64)| = 32/64 - 21/64 = 11/64$ より，

$$
\begin{aligned}
&54/64 \notin A\,(21/64, 11/64)\\
&54/64 \in A\,(2 \times 21/64, 2 \times 11/64)\\
&\qquad = A\,(42/64, 22/64)
\end{aligned}
$$

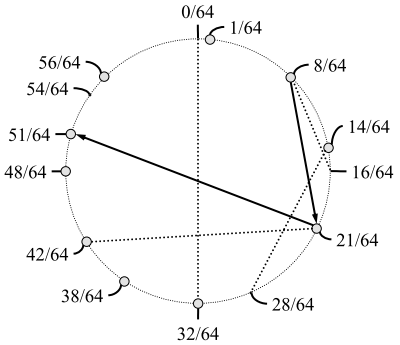

図 9 担当ノードの検索
Fig. 9 Lookup the responsible node.

$L_*(2, 14/64, 54/64) = 3$, $L_*(2, 21/64, 54/64) = 1$ より，ノード 8/64 は，担当ノードまでの経路が短いノード 21/64 を選び，検索メッセージを送信する．検索メッセージを受信したノード 21/64 も同様にして，ノード 51/64 を選び，検索メッセージを送信する．ノード 51/64 は，キー 54/64 の担当ノードであるので，検索元 8/64 へ返信する．

2.7 ノードの参加

オーバレイネットワークに参加しようとするノードは，参加済みのノードへ参加メッセージを送る．参加メッセージには，ノードの位置 $x \in K$ とアドレスを記入する．ノードの位置 x は，初回の参加では円周上から無作為に選び，2 回目以降の参加では前回参加したときの位置とする．2 回目以降では，前回までに記憶していたキーと値を再利用するためである．参加メッセージを受信したノードは，位置 x の担当ノード $x_{-1} = v_{\pm}(x)$ へ中継する．担当ノード x_{-1} が参加メッセージを受信すると，参加受付メッセージをノード x へ返信する．参加受付メッセージには，前ノード x_{-1} と後ノード $x_{+1} = v'_{+1}(x_{-1})$ のノード情報を記入する．参加受付メッセージを受信したノード x は，次の処理を行う．

- メッセージに記されたノード情報をノード表に追加する．
- 前ノードを x_{-1}，後ノードを x_{+1} とする．
- 後ノード x_{+1} へ前ノード通知メッセージを送信する．
- 子ノード検索メッセージを送信する．

前ノード通知メッセージを受信した後ノード x_{+1} は，2.4 節に示した処理を行う．

3. 確率論的評価

提案するオーバレイネットワークの平均経路長と平均次数について評価する．オーバレイネットワークの経

路長とは，あるノードが担当ノードを検索するためにメッセージを送信してから担当ノードがそのメッセージを受信するまでの経路の長さ，すなわち，その経路上の有向辺の数である．経路長が長いほど，担当ノードを検索するために要する時間が長いと考えられる．担当ノードを検索するために要する時間は，担当ノード検索メッセージがあるノードから担当ノードへ届くまでの経路上の各有向辺ごとに 2 つのノード間の通信にかかる時間，各中継ノードがメッセージを受信してから送信するまでの時間，担当ノードがメッセージを受信してから検索元のノードへアドレスを返信するまでの時間の合計となる．厳密には，オーバレイネットワーク上の 2 つのノード間の通信にかかる時間は，その下位の通信ネットワークにおいてメッセージを中継するルータの数や通信経路となった通信回線の容量などが影響するため，送信ノードと受信ノードの組合せによって異なる．また，通信ネットワークは他のシステムの通信にも使われているため，メッセージが届くまでにかかる時間は時間帯や日によっても変動する．提案するオーバレイネットワークでは，任意のノードがそのアドレスによらず無作為に位置を選ぶので，あるノードがメッセージを送信してからある隣接ノードが受信するまでの時間を確率変数 T_1 とする．また，各中継ノードがメッセージを受信してから送信するまでの時間はノードの性能や他のプロセスの負荷などによって変動するので確率変数 T_2 とする．同様に，担当ノードがメッセージを受信してから検索元のノードへアドレスを返信するまでの時間を確率変数 T_3 とする．このとき，経路長を L とおくと，検索に要する時間の期待値は $E(LT_1+(L-1)T_2+T_3) = LE(T_1)+(L-1)E(T_2)+E(T_3) = L(E(T_1)+E(T_2)) - E(T_2) + E(T_3)$ である☆．よって，そのオーダは $\mathrm{O}(L)$ であるので，経路長によって検索に要する時間を評価できる．

オーバレイネットワークの次数とは，有向グラフにおける出次数，すなわち，あるノードから出ている有向辺の数である．有向グラフには入次数もあるが，あるノードへ入ってくる有向辺は他のノードから出ている有向辺でもある．入次数と出次数を合計すると全体では重複して数えることになるため，出次数を評価し，入次数は評価しない．次数が多いほど，ノードの参加や離脱のとき，隣接ノードの更新に多くのメッセージを必要とし，処理に時間がかかると考えられる．ただし，経路長と次数はトレードオフの関係にある．極端な例として，オーバレイネットワークをすべてのノードからすべてのノードへの有向辺を持つ完全グラフとすれば，経路長は 1 であり，最短である．しかし，次数はノード数となり，隣接ノードの更新に長い時間がかかる．よって，次数が同じならば経路長がより短い，または，経路長が同じならば次数がより少ないオーバレイネットワークが望ましい．

3.1 平均次数

次数は，前ノード，後ノード，子ノードの数の合計である．全ノードが等間隔に位置する場合，1 つのノードの担当区域の長さは $1/|V|$ であり，その b 倍の区域の長さは $b/|V|$ であるので，子ノードの数は b となる．よって，直感的には，次数はノード数によらず一定の $b+2$ と考えられる．ただし，通常，全ノードは等間隔とはならない．次の定理より，平均次数がノード数によらず一定範囲 $[b+2, b+3]$ にあることを示す．

定理 7 1 ノードあたりの子ノードの数の平均を $\overline{b}$ とすると $b \le \overline{b} \le b+1$.

証明：ノード数を $n = |V|$ とする．n 個のノードを $x_1, x_2, \ldots, x_n$ とする．ノード x_j（$j \in \mathbb{Z}$，$1 \le j \le n$）の担当区域の b 倍の区域 $A(bx_j, b|T(x_j)|)$ 内に m_j 個のノードがあるとすると，$\sum_{j=1}^{n} m_j = bn$. $\bmod (bx_j, 1)$ にノードがある，または，$b|T(x_j)| \ge 1$ ならばノード x_j の子ノードの数は $|V_*(b, x_j)| = m_j$，どちらでもなければ $|V_*(b, x_j)| = m_j + 1$.

$$\sum_{j=1}^{n} m_j \le \sum_{j=1}^{n} |V_*(b, x_j)| \le \sum_{j=1}^{n} (m_j + 1)$$
$$bn \le \overline{b}n \le bn + n$$
$$b \le \overline{b} \le b+1$$

□

3.2 平均経路長

ノードから子ノードへ進むごとに到達できる区域の長さが b 倍となり，それが 1 以上となったとき，任意の担当ノードへ到達できる．全ノードが等間隔にあるとすると，ノードの担当区域の長さは $1/|V|$ である．よって，直感的には，経路長は $\lceil \log_b |V| \rceil$ 以内と考えられる☆☆．ただし，通常，全ノードは等間隔とはならない．そこで，次の補題と定理より，平均経路長が $\log_b |V| + 1/\ln b + 1$ 未満であることを示す．

補題 8 すべてのノードが一様に分布すると仮定すると，あるノードの担当区域の長さが $r \in \mathbb{R}$（$0 \le r < 1$）未満となる確率は $1-(1-r)^{|V|-1}$ である．

☆ k_1，k_2 を任意の定数，X_1，X_2 を確率変数とするとき，$E(k_1X_1 + k_2X_2) = k_1E(X_1) + k_2E(X_2)$ である[12]．

☆☆ $\lceil x \rceil = \min\{j \in \mathbb{Z} | j \ge x\}$，すなわち，$\lceil x \rceil$ は $x \in \mathbb{R}$ 以上の最小の整数を返す関数（天井関数）である．

証明：ノード $x \in V$ からノード $y \in V$，$x \neq y$ まで時計回りの円弧の長さが r 以上となる確率は $1-r$ である．ノード x 以外の任意のノード $y \in V$，$x \neq y$ について，ノード x からノード y まで時計回りの円弧の長さが r 以上であるとき，ノード x の担当区域の長さ $|T(x)|$ が r 以上となる．ノード x 以外のノードの数は $|V|-1$ であるので，$|T(x)| \geq r$ となる確率は $\Pr\{^{\forall} y \in V, x \neq y : \mathrm{mod}\,(y-x,1) \geq r\} = (1-r)^{|V|-1}$ である．すなわち，$|T(x)| < r$ となる確率は $\Pr\{^{\exists} y \in V, x \neq y : \mathrm{mod}\,(y-x,1) < r\} = 1-(1-r)^{|V|-1}$ である． □

補題 9 $L \in \mathbb{Z}, L \geq 0$ について，キー $k \in K$ の担当ノード $v_{\pm}(k)$ からキー $\mathrm{mod}\,(b^L k,1) \in K$ の担当ノード $v_{\pm}(\mathrm{mod}\,(b^L k,1))$ までの長さ L 以内の経路が存在する．

証明：キー $\mathrm{mod}\,(b^L k,1)$ の担当ノードを v_L とおく．以下から数学的帰納法により，任意の $L \in \mathbb{Z}$，$L \geq 0$ について補題 9 が成立する．

(1) $L=0$ のときは明らかである．

(2) $L \geq 1$ のとき，ノード v_0 からノード v_{L-1} までの長さ $L-1$ 以内の経路が存在すると仮定する．$x = b^{L-1}k$ とおくと，補題 5 より，キー $\mathrm{mod}\,(b^L k,1)$ の担当ノード v_L は，キー $\mathrm{mod}\,(b^{L-1}k,1)$ の担当ノード v_{L-1} の子ノードである．よって，ノード v_0 からノード v_L までの長さ L 以内の経路が存在する． □

補題 10 任意のキー $k \in K$ に対して，ノード $x \in V$ からキー k の担当ノードまでの長さが $1-\log_b |T(x)|$ 未満の経路が存在する．

証明：補題 9 より，ノード x から長さ $L \in \mathbb{Z}$，$L \geq 0$ の経路で到達できる区域は $A(b^L x, b^L |T(x)|)$ を含む．$b^L|T(x)| \geq 1$ のとき，全区域に到達できるので，$b^{L-1}|T(x)| < 1$ を満たす経路が存在する．よって，$L < 1-\log_b |T(x)|$ である． □

定理 11 すべてのノードが一様に分布すると仮定すると，任意のノードからキーの担当ノードまでの経路の長さの期待値 $\overline{L}$ は $\log_b |V| + 1/\ln b + 1$ 未満である☆．

証明：ノード数を $n = |V|$，ノードの担当区域の長さを確率変数 $\tilde{r}$ とおく．補題 8 より，$\tilde{r}$ が r 未満となる確率は $\Pr\{\tilde{r} < r\} = 1-(1-r)^{n-1}$ である．補題 10 より，$\overline{L}$ は次の不等式を満たす．

$$
\begin{aligned}
\overline{L} &< \int_0^1 (1-\log_b r)\frac{d\Pr\{\tilde{r} < r\}}{dr} dr \\
\overline{L} &< \left[(1-\log_b r)\Pr\{\tilde{r} < r\}\right]_0^1 \\
&\quad + \int_0^1 \frac{\Pr\{\tilde{r} < r\}}{r \ln b} dr \\
\overline{L} &< 1 + \lim_{r \to 0} \log_b r \left\{1-(1-r)^{n-1}\right\} \\
&\quad + \frac{1}{\ln b}\int_0^1 \sum_{j=0}^{n-2} (1-r)^j dr \\
\overline{L} &< \frac{1}{\ln b}\sum_{j=0}^{n-2}\left[-\frac{(1-r)^{j+1}}{j+1}\right]_0^1 + 1 \\
\overline{L} &< \frac{1}{\ln b}\sum_{j=1}^{n-1}\frac{1}{j} + 1 \qquad (1)
\end{aligned}
$$

$j \in \mathbb{Z}$，$j \geq 1$，$x \in \mathbb{R}$，$j-1 < x < j$ のとき，次の不等式が成り立つ．

$$
\begin{aligned}
&\int_{j-1}^{j} \frac{dx}{j} < \int_{j-1}^{j} \frac{dx}{x} \\
&\sum_{j=2}^{n}\left[\frac{x}{j}\right]_{j-1}^{j} < \sum_{j=2}^{n}\int_{j-1}^{j}\frac{dx}{x} \\
&\sum_{j=2}^{n}\frac{1}{j} < \int_1^n \frac{dx}{x} \\
&\sum_{j=1}^{n-1}\frac{1}{j} < [\ln x]_1^n + 1
\end{aligned}
$$

式 (1) より，$\overline{L} < \log_b n + 1/\ln b + 1$ である． □

b は定数であるので，平均経路長は $\mathrm{O}(\log_b |V| + 1/\ln b + 1) = \mathrm{O}(\log |V|)$ である．

4. 関連研究

代表的な DHT として，Chord [10),11)] と CAN [8)] があげられる．Chord は，ハイパーキューブ型トポロジに基づいており，次数は $\mathrm{O}(\log n)$，経路長は $\mathrm{O}(\log n)$ である．ここで，n はノード数である．CAN は，トーラス型トポロジに基づいており，次数は $\mathrm{O}(d)$，経路長は $\mathrm{O}(dn^{1/d})$ である．ここで，d は次元数である．提案手法は，Chord よりも次数が少なく，CAN よりも経路長が短い．

次数が一定で経路長が対数の DHT としては，Koorde [3)]，Distance Halving [6)]，D2B [2)]，Viceroy [4)] が提案されている．Koorde，Distance Halving，D2B は，de Bruijn グラフに基づき，Viceroy は butterfly グラフに基づいている．Viceroy は一定次数で対数経路長の最初の DHT であるが，その構築と管理が比較的複雑であり，実装が難しいといわれている [2)]．D2B は，

☆ $\ln x = \log_e x$，すなわち，$\ln x$ はネイピア数 e を底とする対数（自然対数）である．

ノードが de Bruijn グラフを形成するようにノードの識別子を変更するが，構築されたグラフが de Bruijn グラフであることを保証できない[3]．

Koorde は，m ビットのキー $\{k \in \mathbb{Z} | 0 \leq k < 2^m\}$ が de Bruijn グラフを形成するように隣接ノードを定める．このため，ノード間のグラフは de Bruijn グラフとはいえないが，キー間のグラフは de Bruijn グラフであることを保証できる．ただし，問合せ元のノードから担当ノードまでの経路長を最適にするためには，問合せ元のノードに割り当てられたキーの中から最適な出発点を選ぶ必要がある．文献 3) では，目的のキーの上位の数ビットを下位ビットとして持つキーを選ぶと説明している．ところが，Koorde は consistent hashing を用いており，複数のノードやキーが同じ識別子となる確率が無視できる程度に，識別子のビット数は十分に大きくなければならない．たとえば，ハッシュ関数として SHA-1 [13] を用いると，キーは 160 ビットである．このため，問合せ元のノードに割り当てられるキーの数が非常に多い場合があり，多数のキーの中から出発点とすべきキーを選ぶ処理に時間がかかる恐れがある．仮にノード数を 2^{20} = 約 100 万 とすると，1 ノードあたりのキーの数は 2^{140} となる．提案手法では，区域に含まれるかどうかを比較演算によって判定し，出発点を選ぶ必要はない．

Distance Halving は，de Bruijn グラフを一般化し，実数のキー $\{k \in \mathbb{R} | 0 \leq k < 1\}$ が連続グラフを形成するように隣接ノードを定める．担当ノードの検索は，2 つのフェーズからなる．第 1 フェーズでは，ランダムに選んだビットをキーの上位に付加しながら，隣接ノードをたどり，目的のキーを下位ビットとするキーの担当ノードを検索する．第 2 フェーズでは，目的のキーの上位に付加したランダムなビットを消しながら，隣接ノードを逆向きにたどり，目的のキーの担当ノードを検索する．よって，経路長は $2 \log n$ となる．提案手法では，1 フェーズで最適な経路を決定でき，経路長は $\log_b n + 1/\ln b + 1$ 未満である．よって，n が大きいとき，約半分の長さで担当ノードを検索できる．

5. おわりに

本論文では，分散ハッシュ表（DHT）のためのオーバレイネットワーク，および，その上で最適な経路を求める方法を提案した．また，提案するオーバレイネットワークについて確率論的な評価を行い，平均次数がノード数によらず一定であること，および，ノード数を n とすると平均経路長が $O(\log n)$ となることを示した．

今後の課題として，提案手法の実装やシミュレーションなどにより，従来手法と比較する必要がある．さらに，実際の P2P 型システムを構築するための技術，たとえば，キーに対する値の登録と取得，リソースの伝送，セキュリティなどを，本論文で提案したオーバレイネットワーク上でどのように実現するかが問題となる．また，本論文では，ノードの故障については，ノードの離脱としてノードが停止する故障のみを扱い，ノードが正しくない処理を行うビザンティン故障は扱わなかった．ビザンティン故障は，一般には完全に解決することはできず，複数のノードによる多数決や誤り訂正符号などによって抑制される．ハードウェアシステムやネットワークシステムの単体におけるビザンティン故障については，それぞれのシステムの誤り訂正機能によって抑制しているものとする．ハードウェアシステムやネットワークシステムの設計，または，ソフトウェアシステムの誤りによるビザンティン故障を抑制するためには，複数のハードウェアシステムやネットワークシステムを独立して設計する，または，複数のソフトウェアシステムを独立して開発する必要がある．しかし，このような設計や開発は通常のシステムでは行われないため，本論文ではビザンティン故障を扱わなかった．もし故障によって人命の危険や大きな損害が生じるクリティカルシステムを P2P 型システムとして開発する場合は，ビザンティン故障に対する対策が必要となる．

謝辞 本論文の採録に際して，査読者から貴重なご意見をいただいたことに心より感謝申しあげます．

参考文献

1) Balakrishnan, H., Kaashoek, M.F., Karger, D., Morris, R. and Stoica, I.: Looking up data in P2P systems, *CACM*, Vol.46, No.2, pp.43–48 (2003).
2) Fraigniaud, P. and Gauron, P.: D2B: A de Bruijn based content-addressable network, *Theoretical Computer Science*, Vol.355, No.1, pp.65–79 (2005).
3) Kaashoek, M.F. and Karger, D.R.: Koorde: a simple degree-optimal distributed hash table, *Proc. 2nd International Workshop on Peer-to-Peer Systems* (*IPTPS*) (Mar. 2003).
4) Malkhi, D., Naor, M. and Ratajczak, D.: Viceroy: A scalable and dynamic emulation of the butterfly, *Proc. PODC 2002* (2002).
5) Mockapetris, P.V. and Dunlap, K.J.: Development of the domain name system, *Proc. ACM SIGCOMM*, pp.123–133 (1988).
6) Naor, M. and Wieder, U.: Novel architectures

for P2P applications: the continuous-discrete approach, *15th ACM Symp. on Parallelism in Algorithms and Architectures* (*SPAA*), pp.50–59 (2003).
7) 岡下 綾，有次正義，柴田幸夫：一般化 Kautz ダイグラフに基づく DHT を用いた分散検索アルゴリズムの提案，日本データベース学会 Letters, Vol.3, No.2, pp.101–104 (2004).
8) Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Shenker, S.: A Scalable Content Addressable Network, *Proc. SIGCOMM 2001*, pp.161–172 (Aug. 2001).
9) 白鳥則郎，滝沢 誠：分散処理，丸善株式会社，(1996).
10) Stoica, I., Morris, R., Karger, D., Kaashoek, M.F. and Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications, *Proc. ACM SIGCOMM 2001*, San Diego, pp.149–160 (Aug. 2001).
11) Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M.F., Dabek, F. and Balakrishnan, H.: Chord: A scalable peer-to-peer lookup protocol for internet applications, *IEEE/ACM Trans. Networking*, Vol.11, No.1, pp.17–32 (2003).
12) 高松俊朗：数理統計学入門，学術図書出版社 (1988).
13) U.S. DoC, NIST, ITL: FIPS 180-2 Secure Hash Standard (SHS), *National Technical Information Service* (*NTIS*), 5285 Port Royal Road, Springfield, VA 22161 (Aug. 2002).

(平成 18 年 5 月 12 日受付)
(平成 18 年 11 月 2 日採録)

島 和之（正会員）

1993 年大阪大学大学院博士前期課程修了．1994 年同大学院博士課程中退．同年奈良先端科学技術大学院大学助手．2004 年より広島市立大学情報科学部助教授．博士（工学）．ソフトウェア工学の研究に従事．電子情報通信学会，IEEE Computer Society 各会員．

松尾 浩平

1983 年生．2002 年広島市立大学入学．2006 年広島市立大学卒業．同年（株）ジャステック入社．学士（工学）．P2P 型システム，分散ハッシュ表の研究に従事．

大場 充（正会員）

1949 年生．1973 年青山学院大学大学院理工学研究科修士課程修了．1974 年日本アイ・ビー・エム（株）入社．1994 年より広島市立大学情報科学部教授．分散環境における協調問題解決に関する研究等に従事．著書『ソフトウェア開発技術』（オーム社）等．日本規格協会オープンソースソフトウェア委員会委員長．IEEE，ソフトウェア技術者協会各会員．

佐藤 康臣（正会員）

1964 年生．1992 年広島大学大学院工学研究科博士課程後期単位取得退学．同年広島大学工学部助手．1994 年より広島市立大学情報科学部助手．ソフトウェア工学，インターネットの応用等に興味を持つ．工学修士．IEEE，電子情報通信学会，ソフトウェア技術者協会各会員．

清水 將吾（正会員）

2001 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了，広島市立大学情報科学部助手．博士（工学）．データベース理論の研究に従事．電子情報通信学会，日本データベース学会各会員．