

# オーバーレイネットワークにおける グループ間通信抑制手法

長尾 洋也<sup>†1</sup> 首藤 一幸<sup>†1</sup>

DHT アルゴリズムの構成手法である *LFRT* を提案する。LFRT に基づく DHT アルゴリズムは、ノードグループ情報を利用して経路表を構成し、ノードグループ間の通信を抑制する。また、LFRT に基づく DHT は、同時にサブ DHT を提供する。サブ DHT は、ノードグループごとに構成される DHT であり、通常の DHT と同等の機能を提供する。サブ DHT ごとの経路表は必要なく、ノードは通常の DHT のための単一の経路表を構築・保守すればよい。

LFRT を Chord に適用し、DHT アルゴリズム *LFRT-Chord* を構成した。実験により、LFRT-Chord がグループをまたぐ通信を抑制すること、および、サブ DHT における経路長が短く抑えられることを確認した。

## Methods of Suppression Hops between Groups in Overlay Networks

HIROYA NAGAO<sup>†1</sup> and KAZUYUKI SHUDO<sup>†1</sup>

This paper presents Labeled Flexible Routing Tables (LFRT), a methodology to construct a DHT algorithm which keeps lookup traffic within a node group and suppress communications across node groups. A DHT based on LFRT provides multiple sub-DHTs. A sub-DHT consists of all nodes in a node group and work as a complete DHT each even though a node maintains a single routing table.

We constructed LFRT-Chord by applying LFRT to Chord, a typical DHT algorithm. Results from simulations show that LFRT-Chord suppress inter-group communications and provides full-fledged sub-DHTs.

### 1. はじめに

ネットワークの利用方法の多様化に伴い、既存のネットワーク上に仮想的なネットワークを構築するオーバーレイネットワークの利用が盛んとなっている。

オーバーレイネットワークを利用例には、Peer-to-Peer ネットワークや、アプリケーション層マルチキャスト (ALM)、CDN などがある。オーバーレイネットワーク、特に Peer-to-Peer ネットワークは数百万ノードを利用したオーバーレイネットワークを想定しているため、スケラビリティや耐故障性、負荷分散に優れたネットワーク構築手法が求められている。その代表例に Distributed Hash Table (DHT) がある。DHT は、オーバーレイネットワーク上に連想配列を構成する手法であり、具体的な DHT アルゴリズムとして、Chord<sup>1)</sup> や Kademlia<sup>2)</sup>、Tapestry<sup>3)</sup>、Pastry<sup>4)</sup> など、多数のアルゴリズムが提案されてきた。

しかし、これらのアルゴリズムに基づいて構成した DHT ネットワークはアンダーレイの情報を考慮せずに構成されるため、アンダーレイから見たときに非効率なルーティングを行ってしまう。たとえばインターネット上に構成した DHT ネットワークでは、ISP 間の通信の増大や、ノード lookup 中に多くの異なる ISP を経由する必要がある問題が発生している。また、分散データストアに DHT を利用するとき、通信がネットワークスイッチを不必要に通過してしまう問題がある。

そこで、本稿では、ネットワーク上のノード群を“ノードグループ”にグループ分けした上でグループ情報を利用する DHT ルーティングアルゴリズムの構成手法 *Labeled Flexible Routing Tables (LFRT)* を提案する。グループ分けの方法は任意であり、DHT の利用者、もしくは DHT を利用するアプリケーションが設定する。たとえば、ISP 間の通信を減少させたい場合には、ISP ごとにノードグループを構成すればよい。

また、LFRT では、構成した DHT と並行して、それぞれのノードグループを一つの DHT として利用することが可能である。本稿では、前者をグローバル DHT、後者をサブ DHT と呼ぶ。先の例では、各 ISP 内のノードのみで一つのサブ DHT が構成され、ISP 内に閉じた DHT を形成することが可能となり、ISP 内に閉じたルーティングおよび担当ノード決定が行われる。

LFRT は、DHT アルゴリズムの構成手法である“柔軟な経路表”がベースとなっている。これは、柔軟な経路表のノード ID 以外の要素を経路表構成に取り込みやすいという特性がルーティングにノードグループ情報を利用するルーティングアルゴリズムの構成に適しているからである。

<sup>†1</sup> 東京工業大学 大学院情報理工学専攻 数理・計算科学専攻  
Dept. of Mathematical and Computing Sciences, Graduate School of Information Science and Engineering, Tokyo Institute of Technology

本稿では、LFRT を既存の DHT アルゴリズムである Chord に適用した DHT アルゴリズム LFRT-Chord を設計・実装し、実験を行った。その結果、Chord やラベルを考慮しない FRT-Chord と比較してグループをまたぐ通信を削減する一方で、経路長の増加が小さくとどまることを確認した。

## 2. 柔軟な経路表

### 2.1 柔軟な経路表の特徴

柔軟な経路表は、DHT アルゴリズムの構成法であり、LFRT のベースとなる手法である。柔軟な経路表では、経路表エントリの ID に関する分布に基づいて経路表を構成する。

まず、目標となる分布の満たす条件を設定する。この条件を目標分布条件と呼ぶ。柔軟な経路表では、目標分布条件を適切に定めることが、柔軟な経路表に基づく DHT アルゴリズムのルーティング効率に影響を与える。

柔軟な経路表に基づく DHT アルゴリズムは、次の“追加アルゴリズム”と“削除アルゴリズム”によって分布が目標分布条件に近づくように経路表を管理する。

#### 2.1.1 追加アルゴリズム

経路表へノードを追加する方式を経路表エントリの追加アルゴリズムと呼び、新しいノード情報を取得した際にノード ID に関わらず無条件に経路表へノードを追加する。

#### 2.1.2 削除アルゴリズム

経路表エントリ数とその最大値（経路表サイズ）を越える際にどの経路表エントリを削除するかを選択する方式のことを経路表の削除アルゴリズムと呼ぶ。

削除アルゴリズムは次の通りである。

- (1) ID 近傍ノードを削除対象からはずす。
- (2) 目標分布条件を満たす分布に近づくように削除対象ノードを選択する。

ID 近傍ノードとは、Chord で言う successor list のノードや predecessor ノードに代表される、探索の到達性を保証するための、自ノードと ID 的に近いノードのことである。

削除対象の経路表エントリを選択する際には、ID 以外の情報を利用することが可能であり、LFRT はこの特徴を応用した DHT アルゴリズム構成手法である。

#### 2.1.3 フォワーディングアルゴリズム

以上のようにして構築した経路表を利用してフォワーディング先を決定するアルゴリズムを“フォワーディングアルゴリズム”と呼ぶ。フォワーディングとは、メッセージの転送を指す。柔軟な経路表ではフォワーディングアルゴリズムを規定しない。代表的なフォワー

ディングアルゴリズムとして、経路表エントリ中から目標 ID に最も近いノードへフォワーディングする方式が考えられる。柔軟な経路表では、このような ID 情報のみを利用したフォワーディングアルゴリズム以外にも、経路表エントリごとのその他のノード情報、たとえば遅延情報などを考慮したフォワーディングアルゴリズムの利用も可能である。

### 2.2 柔軟な経路表の利点

柔軟な経路表は、経路表エントリが分布条件に近づくようにさえ管理すればよいことから、従来の経路表管理と比較して柔軟性に優れ、さまざまな利点を得られる。

#### ● ノード数に応じたゼロホップ DHT への移行

柔軟な経路表では、事前に設定した経路表サイズを超えない範囲で経路表にノードを無条件に追加するため、ネットワーク上のノード数が経路表サイズ以下のとき、経路表に全ノードを載せることが可能である。この特性から、ゼロホップ DHT を構成することが可能である。ゼロホップ DHT とは、DHT に参加しているノードから担当ノードへ到達するまでの経路長の最大値が 1 となる DHT のことである。ゼロホップ DHT を実現するためには、全ノードの経路表に全ノードが載っている必要があるが、柔軟な経路表に基づきアルゴリズムを構成することで、この必要条件を満たすことが可能である。本稿で紹介する LFRT-Chord は実際にゼロホップ DHT を実現している。

#### ● 経路表サイズの設定が可能

柔軟な経路表は、あらかじめ与えられた経路表サイズを守って経路表を構築するため、DHT アルゴリズムはメモリや死活監視の通信負荷が許す限り、自由に経路表サイズを大きくとることが出来る。また、応用に応じて経路表を小さくすることも可能である。そして、分布を管理するという特徴から、DHT を動作させたまま動的に経路表サイズを変更することが可能であり、それに伴う経路表の作り直しも不要である。

## 3. LFRT : Labeled Flexible Routing Tables

この章では、提案手法である LFRT の柔軟な経路表との差分、および LFRT の特徴について述べる。

### 3.1 LFRT の概要

LFRT は柔軟な経路表を拡張した DHT アルゴリズムの構成手法である。各ノードに設定されている“ラベル”に基づき、LFRT は同じラベルを持つノード群を“ノードグループ”と見なし、経路中のノードグループをまたぐ通信を抑制する。また、LFRT に基づいて構成した経路表を利用することでグループ内に閉じたルーティングおよび担当ノード決定を

実現可能である。これを利用することで、ネットワーク上の全ノードで構成される DHT と同一の経路表を利用してそれぞれのノードグループごとに構成される DHT を利用可能となる。このとき、ネットワーク全体の DHT をグローバル DHT、各ノードグループごとに構成される DHT をサブ DHT と呼ぶ。

### 3.2 ラベルとグループ

LFRT を利用する際は、事前に各ノードヘラベルと呼ばれる識別子を設定する。そして、同じラベルを持つノード同士を一つのノードグループとみなす。LFRT において、それぞれのノードに設定するラベルの決定方法は規定せず、自由に設定することが可能である。また一方で、それぞれのノードが事前にネットワーク上にどのようなラベルが存在するかを知っている必要はない。

### 3.3 削除アルゴリズム

LFRT は、柔軟な経路表の削除アルゴリズムのみを変更することで経路中のグループをまたぐ回数を抑える。従って、柔軟な経路表に基づいて構成した DHT アルゴリズムからの変更がわずかで済む。LFRT の削除アルゴリズムは次の通りである。

- (1) ID 近傍ノード (2.1.2) を削除対象からはずす。
- (2) 自ノードグループ ID 近傍ノードを削除対象からはずす。
- (3) 自ノードグループ ID 近傍ノードのうち、最も ID 的に自ノードに近いノードより ID 的に遠い位置の他グループノードが経路表中に存在する場合、目標分布条件を満たす分布に近づくようにその他グループノードの中から削除対象ノードを選択する。
- (4) まだ削除対象ノードが選択されていない場合、残りのノード中から目標分布条件を満たす分布に近づくように削除対象ノードを選択する。

ここで自ノードグループ ID 近傍ノードとは、自ノードグループのみで DHT が構成されていると考えた場合における ID 近傍ノードである。

### 3.4 加入アルゴリズム

新しくノードが DHT に加入する際に、ID に関して自ノードに近いノードを経路表に追加する手順を“加入アルゴリズム”と呼ぶ。たとえば、Chord の加入アルゴリズムは、自ノードの ID の担当ノードを探索することであり、それにより successor を経路表へ追加している。

LFRT では、サブ DHT を利用するために、加入アルゴリズムに注意が必要である。従来の追加アルゴリズムはグローバル DHT における加入アルゴリズムに過ぎず、加入アルゴリズムを実行した結果、グローバル DHT における ID 近傍ノードしか経路表に追加されな

い。従って、サブ DHT を利用する場合は、サブ DHT に関しても加入作業を行う必要がある。ただし、ノード ID はグローバル DHT とサブ DHT のどちらに関しても同一であるため、グローバル DHT における追加アルゴリズムとサブ DHT における追加アルゴリズムの重複は大きく、最適化が容易に考えられる。

### 3.5 フォワーディングアルゴリズム

グローバル DHT を利用する際はフォワーディングアルゴリズムに一切の変更は必要ない。一方で、サブ DHT を利用する場合は、フォワーディング先を経路表エントリのうちの自ノードグループ内ノードから選択する。

## 4. LFRT-Chord : LFRT の Chord への適用

“LFRT-Chord” は LFRT に基づいて構成した DHT アルゴリズムであると同時に、柔軟な経路表に基づいて構成した “FRT-Chord” の拡張である。この章では、LFRT-Chord の Chord に対する変更点について述べる。

### 4.1 経路表

LFRT-Chord は、LFRT に基づき successor list と finger table および predecessor node という 3 種類の経路表を一つの経路表に区別せず統合している。また、経路長の各エントリはノード ID とノードの IP アドレスとのペアと、それらに付随するラベルで構成されている。

### 4.2 担当ノード決定手法

LFRT-Chord では、担当ノードを目標 ID の successor node ではなく、predecessor ノードとしている。これにより、ゼロホップ DHT が実現される。

### 4.3 目標分布条件

LFRT-Chord は、目標分布条件として式 (3) を採用する。

$$\delta(x) = \begin{cases} 1 & (\exists e \in X, d(e) = x) \\ 0 & (\text{otherwise}) \end{cases} \quad (1)$$

$$F(x) = \frac{1}{|X|} \sum_{d=1}^{\lfloor x \rfloor} \delta(d) \quad (1 \leq x \leq 2^m - 1) \quad (2)$$

$$F(2x) - F(x) = (\text{一定}) \quad (3)$$

ただし、 $X$  は経路表エントリの ID 集合であり、 $m$  はノード ID のビット長、 $d(x)$  は自ノードから ID  $x$  までの距離を示している。

#### 4.4 追加アルゴリズム

LFRT-Chord は、自身が他のノードと通信を行った際に、通信相手を経路表へ追加する。

#### 4.5 削除アルゴリズム

LFRT-Chord では、通信の失敗時および、経路表エントリ数  $l$  が経路表サイズ  $L$  を越える場合に経路表エントリの削除を実行する。

$l = L$  の状態で経路表へノードが追加された場合、 $L + 1$  個のエントリを一時的に一つの経路表と見なし、 $L + 1$  個の経路表エントリの中から削除アルゴリズムに基づいて削除対象エントリを決定する。削除アルゴリズムは LFRT に基づく。また、経路表エントリ ID を、自ノードから近い順に  $x_0, x_1, \dots, x_{l-1}$  とし、 $x_{i+1}.pred = x_i$  とし、successor list の長さを  $s$ 、 $C(x)$  を式 (4) とする。

$$C(x) = \bar{F}(d(x)) - \bar{F}(d(x.pred)) \quad (4)$$

$$\bar{F}(x) = \frac{1}{m} \log_2 x \quad (5)$$

$Y, L, M, Z$  を ID 集合とすると、削除対象ノード  $c$  は次のように選択される。

- (1)  $Y \leftarrow X$
- (2)  $Y \leftarrow Y \setminus \{y|y \in Y, y \text{ は ID 近傍ノード}\}$
- (3)  $Y \leftarrow Y \setminus \{y|y \in Y, y \text{ は自ノードグループ ID 近傍ノード}\}$
- (4)  $L \leftarrow \{x|x \in X, x \text{ は自ノードグループ内ノード}\}$
- (5)  $M \leftarrow Y \setminus L$
- (6)  $Z \leftarrow \{z|z \in M, \exists l_0 \in L, d(l_0) < d(z)\}$
- (7) IF  $Z \neq \emptyset$  THEN  $Y \leftarrow Z$
- (8)  $c$  を  $c \in Y, C(c) = \min_{y \in Y} C(y)$  を満たすノードとする

#### 4.6 フォワーディングアルゴリズム

経路表エントリのうち、目標 ID に対する ID 的距離が最も近いノードをフォワーディング先とする。また、サブ DHT におけるフォワーディングでは、経路表エントリのグループ内ノードから、目標 ID に対する ID 的距離が最も近いノードをフォワーディング先とする。

## 5. 評価

LFRT-Chord の持つ性質を評価する。LFRT-Chord をオーバーレイ構築ツールキットである Overlay Weaver<sup>5)6)</sup> を利用して実装した。実験は、実装した LFRT-Chord アルゴリズムを利用し、1 台のマシン上でのエミュレーションにより行う。各ノードの挙動管理は、

Overlay Weaver 用のシナリオファイルを読み込ませることにより行う。利用した環境は、次の通りである。

- OS : Windows Vista SP2 32bit
- CPU : Intel Core 2 Duo E8400 3.00GHz
- メモリ : 4.00GB
- JRE 6 Update 20
- Overlay Weaver 0.9.9

### 5.1 実験 1 : グループ間通信の抑制と経路長

この実験では、LFRT-Chord がグループ間通信の抑制にどの程度寄与し、経路長とのトレードオフがどの程度に抑えられているかを検証する。

#### 5.1.1 実験内容

実験では、ノード数  $N$  を 10000 とし、1000 ノードずつ 10 のノードグループに分けたネットワークを想定する。そして、ノードグループを考慮する LFRT-Chord と考慮しない FRT-Chord とを比較した。また、参考のために Chord でも同様の実験を行った。それぞれの条件の下、通信を繰り返し、自然に経路表が構築されるのを待った後、ランダムに選択したノードから、ランダムに選択した ID への lookup を 50000 回実行した。

実行後、一回の lookup に伴う不必要なグループ間通信の回数と lookup 時の経路長を集計した。このとき、lookup を開始したノードが属するノードグループと、目標 ID の担当ノードが属するノードグループとが異なる場合には最低 1 回は通信が必要であるとみなし、異なるノードグループ間の通信回数から 1 を引いた値を不必要なグループ間通信の回数としている。

また、1 つの lookup メッセージが lookup 元のノードグループから、別のグループを経由してまた lookup 元のノードグループに 1 回以上戻ってくる経路に関しても集計した。

#### 5.1.2 実験結果と考察

グループ間通信の抑制結果は図 1 および図 2、図 3、図 4 の通りである。実験結果の図中の  $L$  は経路表サイズを表し、 $s$  は successor list の長さを表している。

図 1 は、lookup 一回につき平均で何回の不必要なグループ間通信が行われたかを示している。この図から、LFRT-Chord は、FRT-Chord と比べて、不必要なグループ間通信をおよそ 8 分の 1 まで抑制していることが分かる。

図 2 は、lookup 一回で何回の不必要なグループ間通信が行われたかを示している。この図からも不必要なグループ間通信を大幅に削減していることが分かる一方で、その最大値を

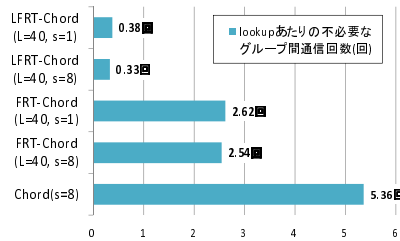


図 1 lookup あたりの不必要なグループ間通信回数  
Fig. 1 The number of unnecessary connections between groups per lookup.

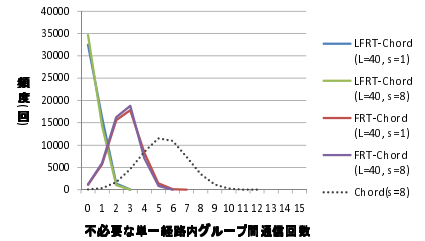


図 2 不必要な単一経路内グループ間通信回数  
Fig. 2 Histograms of unnecessary connections.

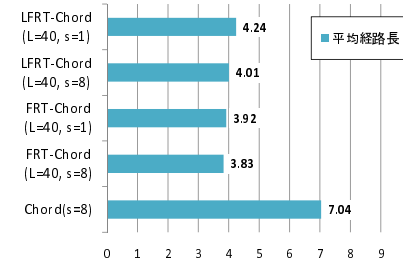


図 5 平均経路長の比較  
Fig. 5 Comparison of the number of average path lengths.

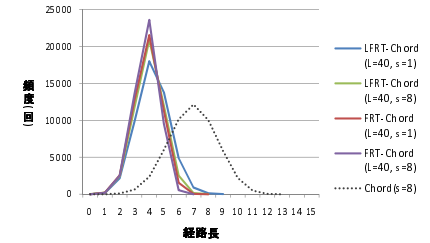


図 6 経路長の分布  
Fig. 6 Histograms of path lengths.

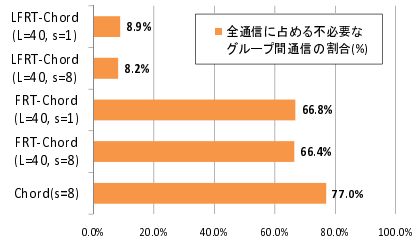


図 3 全通信に占める不必要なグループ間通信の割合  
Fig. 3 The percentage of the number of unnecessary connections between groups in all con-

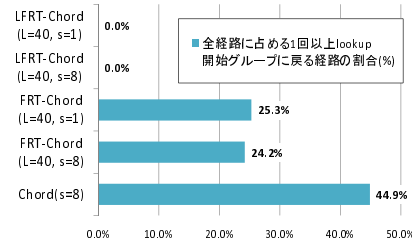


図 4 全経路に占める 1 回以上 lookup 開始グループに戻る経路の割合  
Fig. 4 The percentage of the number of paths to go back in the first group in all paths.

小さく抑えていることが分かる。

図 3 は、50000 回の lookup に伴う通信の内に占める不必要なグループ間通信の割合を集計したものである。図 1 と比較して、FRT-Chord と Chord の差が小さいことが分かる。これは、FRT-Chord と Chord の不必要なグループ間通信の回数の差が経路長の差によって生じていたことを示している。一方で、LFRT-Chord は必要な経路長のうち、より多くの経路を同一グループ内に閉じ込めていることが分かる。

図 4 は、lookup 元のノードグループに 1 回以上戻って来てしまう lookup が、全 lookup のうちのどれだけかを占めているかを示している。この図は、FRT-Chord の lookup のうちの約 4 分の 1 のメッセージが lookup を実行したグループを離れた後、再び元のグループへ戻ってきてしまっていることを示している。一方、LFRT-Chord は、グループを離れた後、

再び元のノードグループへ戻ってくるのが一切無い。これは、LFRT-Chord において経路表が十分更新されている状態において同じグループを複数回通過しないという特徴に起因している。

LFRT-Chord の経路長を集計した結果は図 5 および図 6 の通りである。

図 5 の示しているとおり、LFRT-Chord の平均経路長と FRT-Chord の平均経路長に見られる差が小さく抑えられていることが分かる。また、図 6 から、経路長の分布に大きな違いはなく、FRT-Chord の短い経路長がほぼ保たれたと考えられる。

以上の実験結果から、LFRT-Chord のグループ間通信抑制効果ははっきり現れている一方で、経路長のトレードオフが小さく抑えられており、LFRT-Chord がノードグループ情報を考慮する効率の良いアルゴリズムであるといえる。

また、LFRT-Chord では、グローバル DHT とサブ DHT の両方について successor list を経路表内部に保持する必要があるが、successor list の長さを 1 から 8 にしたとしても、経路長の短さが直ちに損なわれることはないことが分かる。

### 5.2 実験 2: サブ DHT 内 lookup の経路長について

この実験では、ノードグループに閉じた DHT であるサブ DHT を利用する場合に、同じノード数で構成された DHT と比較してどれだけ経路長に違いが発生するかを検証する。

#### 5.2.1 実験内容

実験では、サブ DHT の経路長を測定するため、10000 ノードのネットワークを 1000 ノードずつ 10 のノードグループに分け、LFRT-Chord のアルゴリズムを用いて DHT を構成した。そして、ランダムに選択したノードから、ランダムに選択した ID へノードグループ内

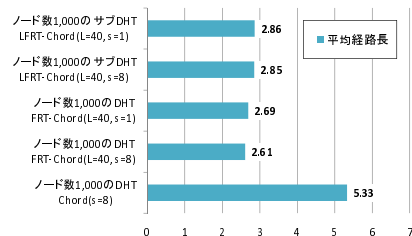


図7 平均経路長の比較

Fig. 7 Comparison of the number of average path lengths.

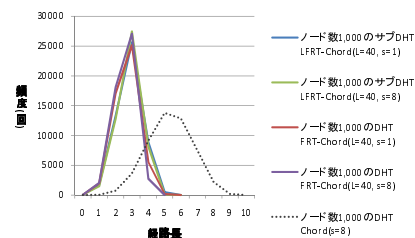


図8 経路長の分布

Fig. 8 Histograms of path lengths.

に閉じた lookup を 50000 回実行した。このとき、それぞれのサブ DHT に属するノード数は 1000 である。従って、比較対象として、全体のノード数が 1000 のネットワークを設定し、FRT-Chord を用いて DHT を構成し、50000 回の lookup を実行した。また、参考として Chord の場合でもノード数が 1000 のネットワークを設定して、50000 回の lookup を実行した。この実験でも、測定する lookup を実行する前に十分な回数の lookup を行い、経路表への追加と削除を十分に行っている。

### 5.2.2 実験結果と考察

実験結果は図 7 および図 8 の通りである。

これらの図から、LFRT-Chord が仮想的に構成したサブ DHT における lookup の平均経路長が、FRT-Chord の構成する DHT における lookup の平均経路長と比較して、わずか約 0.2 の増加にとどまっていることが分かる。また、経路長分布も非常に近く、平均経路長だけでなく、最大経路長などを含めた経路長特性の変化が小さく抑えられていることが分かる。

このことから、LFRT-Chord がグローバル DHT においてグループ間通信を抑制するとともに、サブ DHT にも利用できる経路表が構成されていることが分かった。これは、グローバル DHT とサブ DHT という 2 つの異なる DHT を一つの経路表で扱っている一方で、別々に構成する場合に近い経路長特性を実現していることが分かる。また、一つの経路表で扱うということは、複数の DHT を一つの DHT 分の経路表維持コストで実現していることを意味している。

## 6. まとめと今後の課題

本稿では、ノードグループ情報を考慮したルーティングを行う DHT アルゴリズム構成手法 LFRT を提案した。

LFRT は、柔軟な経路表の拡張性を利用した構成手法であり、あらかじめ設定したノードグループ情報を考慮した経路表構成を行うことでノードグループをまたぐ通信の抑制を行う。またそれと同時に、ノードグループごとにサブ DHT を構成し、ノードグループに閉じたルーティングや担当ノード決定を行うことが出来る。

実際に LFRT の手法を Chord に適用した DHT アルゴリズム LFRT-Chord を設計し、Overlay Weaver 上に実装した。その実装用いた実験を行い、短い経路長を維持したまま、ノードグループ間の通信をおよそ 8 分の 1 にまで抑制することを確認した。また、サブ DHT が同じノード数で構成した DHT と比較して平均経路長の増加が 0.2 程度に抑えられていることが分かった。

今後、LFRT を利用した新たな応用手法の設計および、ルーティング効率に関する定量的な証明を検討する予定である。

## 参考文献

- 1) 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 '01*, pp.149-160 (2001).
- 2) Maymounkov, P. and Mazières, D.: Kademlia: A Peer-to-Peer Information System Based on the XOR Metric, *Proc. IPTPS '02 LNCS 2429*, pp.53-65 (2002).
- 3) Zhao, B.Y., Huang, L., Stribling, J., Rhea, S.C., Joseph, A.D. and Kubiatowicz, J.D.: Tapestry: A Resilient Global-scale Overlay for Service Deployment, *IEEE Journal on Selected Areas in Communications*, Vol.22, No.1, pp.41-53 (2004).
- 4) Rowstron, A. I.T. and Druschel, P.: Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems, *Proc. IFIP/ACM Middleware 2001*, pp.329-350 (2001).
- 5) 首藤一幸: Overlay Weaver, <http://overlayweaver.sourceforge.net/>.
- 6) Shudo, K., Tanaka, Y. and Sekiguchi, S.: Overlay Weaver: An overlay construction toolkit, *Computer Communications*, Vol.31, No.2, pp.402-412 (2008).