

分散ライングラフを用いたオーバーレイネットワークの遅延評価・比較,
および改良手法について

Evaluation and Improvement on Delay for Structured Overlay Networks based on
Distributed Line Graphs

穆 雨晨 眞田 亜紀子 森田 啓義

Yuchen Mu Akiko Manada Hiroyoshi Morita

1. はじめに

構造化 P2P ネットワークは, Chord[1]や Kademlia [2]など, これまでに多くの手法が提案されてきた. 近年では, 正則グラフに基づく定数次数構造化オーバーレイネットワークについて耐障害性や探索効率に優れることが知られている[3]. 分散ライングラフ (Distributed Line Graph, DL グラフ)は Zhang ら[4]によって提案された, 正則グラフに基づくある有向グラフで, P2P システムにおける構造化オーバーレイネットワークを構築することを目的としている. DL グラフは, どのノードの出次数も一定である為, P2P ネットワークとして耐障害性やスケーラビリティに優れるといった性質を持つ[5]. しかしながら, 実環境下で DL グラフと他の分散ハッシュテーブル (Distributed Hash Table, DHT)の具体的な性能差異については研究が進んでいない.

本研究はネットワークの遅延時間に着目し, シミュレーションを通して DL グラフに基づくオーバーレイネットワークの性能を評価する. さらに遅延時間を改善する手法を提案し, その効果についても考察する.

本研究の成果は, 低遅延が求められるサービスを展開するための基盤ネットワークとしての応用が期待できる.

2. DL グラフに関する理論的背景

2.1. DL グラフの定義

DL グラフはある与えられた正則グラフ(全ての頂点の次数が等しい)を初期グラフとして以下に述べる分散ライン変換(Distributed Line Translation, DL 変換)を繰り返して得られる有向グラフの総称である. DL グラフの初期グラフの各頂点には有限アルファベット A 上の長さ 1 の文字列が割り振られている.

2.2. DL 変換の定義[4]

一般的に有向グラフ G の各頂点 $v \in V(G)$ に対し, $\Gamma_v^+(v)$ と $\Gamma_v^-(v)$ は v の出隣接頂点集合と入隣接頂点集合と置く. DL グラフ G の頂点 $v \in V(G)$ には長さ 1 以上の A 上の文字列が次の DL 変換によって 1 対 1 に対応する. 以下では頂点 v に対応する文字列を v の ID と呼ぶ, 頂点と ID を同一視して扱う. さらに, 頂点 $u = u_1u_2 \dots u_m, v = v_1v_2 \dots v_n$ ($m \geq n$) に対し, 二項演算子 \circ を $u \circ v = u_{m-n+1}v_1v_2 \dots v_n$ と定義する. 例えば, $1201 \circ 021 = 2021$.

DL 変換の手順は以下に示す:

1. DL グラフの初期グラフを頂点数 k の正則有向グラフ G_0 とおく.
2. $i \geq 0$ に対し, DL グラフ G_i とその頂点集合 $V(G_i)$ に対して DL グラフ G_{i+1} を以下の手順で求める;
 - 1) G_i の頂点集合からどの隣接頂点と比べて, 一番短い ID を持つ頂点 $v^{(i)}$ を選ぶ.
 - 2) 次式の条件を満たす頂点を作成する.

$$V(G_{i+1}) = V(G_i) \setminus \{v^{(i)}\} \cup \{u \circ v^{(i)} \mid u \in \Gamma_{v^{(i)}}^-(v^{(i)})\}$$

- 3) 次式の条件を満たす辺を作成する.

$$E(G_{i+1}) = E(G_i) \setminus \{[x, v^{(i)}] \mid x \in \Gamma_{v^{(i)}}^-(v^{(i)})\} \setminus \{[v^{(i)}, y] \mid y \in \Gamma_{v^{(i)}}^+(v^{(i)})\} \cup \{[u, u \circ v^{(i)}] \mid u \in \Gamma_{v^{(i)}}^-(v^{(i)})\} \cup \{[u \circ v^{(i)}, w] \mid u \in \Gamma_{v^{(i)}}^-(v^{(i)}), w \in \Gamma_{v^{(i)}}^+(v^{(i)})\}$$

2.3. DL グラフの性質

DL グラフは以下の性質を持つことが知られている. [4][6]

- (1) 各頂点の出次数 $\psi(v)$ (出ていく辺の数) は等しい.
- (2) 隣接する頂点同士の ID の長さの差は 1 以下
- (3) 任意の頂点 x, y について x の ID は y の ID の接尾辞でない
- (4) どの $ID = v_1v_2 \dots v_n$ においても隣接シンボルは異なる, すなわち, $v_k \neq v_{k+1}$ ($1 \leq k < n$)
- (5) 各頂点には有限アルファベット A 上の文字列 (ID) が振られており, ID に基づいて任意の頂点から別の頂点への経路を探索する
- (6) 頂点数 N の際の直径 (最大ホップ数) は $\log(N)$

また, DL 変換は局所的に行われるため, ユーザ加入・離脱に伴うコストも DHT に比べ軽減できることが見込める. これらの特徴から, オーバーレイトポロジーとして DL グラフの有用性が期待される. ID s の ID t に関するマッチングインデックスを s の接尾辞と t の接頭辞との最長一致部分列の長さとして定義した上で, 性質 5 で述べたネットワーク上の探索は次のように行う[7].

頂点 p が宛先 $t = t_1t_2 \dots$ (グラフ上存在しない場合もある) を探索することを考える. まず, p の DL グラフ上で遷移可能な隣接頂点 (隣接点) の中から t に関するマッチングインデックスが最大となる頂点 s を選択する. そして, s は s の隣接点の中からマッチングインデックスを最大化するように探索クエリの転送先を決める. マッチングインデックスを最大化するノードに到達したとき, 探索は終了する.

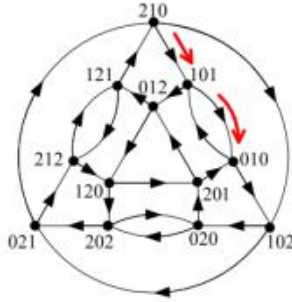


図 1 ([4]より) ルーティング : 210→010

図 1 はノード 210 からノード 010 までのルーティングを示した。探索は 210 の出隣接(101 と 102)の中から 010 に関するマッチングインデックス最大となる頂点 101 を経由して、010 に到着する。それから、010 と 010 の出隣接(101 と 102)を比べ、010 はマッチングインデックス最大となる頂点であるによって、探索は終了する。

3. 評価実験

DL グラフの性能評価をするために、実環境化での遅延シミュレーションと比較を行った。

3.1. 実験環境

CPU	Intel Core i5-8250u
OS	Windows 10
Python バージョン	3. 7. 3
OverlayWeaver バージョン	0. 10. 5

3.2. 下位ネットワークの構築

遅延シミュレーションの設定として、下位ネットワークにおけるノード間の通信遅延を設定した。ここでは、下位ネットワークをインターネットと仮定し、そのトポロジをトランジット・スタブ・モデル[8] (Transit-Stub Model, TS モデル) に基づいて決定した。TS モデルは、インターネットの上位ルータの域的な構造を表すモデルである。

今回の実験では、GT-ITM(Georgia Tech Internetwork Topology Models)[9]を使って、下位ネットワークを構築する。GT-ITM とは、TS モデルを含むいくつかのモデルに基づくネットワークトポロジを生成するツールである。

3.3. オーバレイネットワークの実装

DL グラフは上述した下位ネットワーク上のオーバレイネットワークとして Overlay Weaver[10]を用いる。

Overlay Weaver はオーバレイ構築ツールキットであり、ルーティング層の分割によって、P2P ネットワークとして代表的な分散ハッシュテーブル(DHT)アルゴリズムをたかだか数百ステップで実装することが可能である。Overlay Weaver はいくつかの DHT アルゴリズムを実装されているので、本研究のアルゴリズムとの性能と評価もしやすくなる。

今回のシミュレーションは DL グラフ(初期グラフは完全グラフ K_4 と K_5), Kademia と Chord を含め

て 4 種類の分散ハッシュテーブルから実験を行う。

3.4. 実験の流れ

まず、TS モデルに基づいて下位ネットワークを生成した。生成したネットワークのノード間の通信遅延は Dijkstra 法[11]により最短経路を求めて決定した。

生成した下位ネットワークを用いて、Overlay Weaver で DL グラフ, Kademia と Chord のオーバレイネットワークを実装した。DL グラフの初期グラフは、完全グラフ K_4 と K_5 と設定した。DL 変換を繰り返すことにより最大 10000 個のノードからなる DL グラフを生成した。

DL グラフと、これの他に用意した DHT アルゴリズムについて、それぞれ同様の下位ネットワークで実験を行い、平均ホップ数及び平均遅延を計測した。実験時のパラメータはノード数を $N=100$ 個, 500 個, 1000 個, 5000 個, 10000 個とした。また、Kademia の k -bucket サイズは、デフォルト設定としてよく利用される $k=20$ である。実験を三回行って、作成したネットワークの平均ホップ数と平均遅延を計算した平均結果を図 2, 3 に示す。

3.5. 実験結果

実験の結果を図 2, 3 に示す。横軸はノード数、縦軸は平均ホップ数 (探索中各頂点間のホップ数の平均値を「平均ホップ数」と呼ぶ) と平均遅延(ms)である。

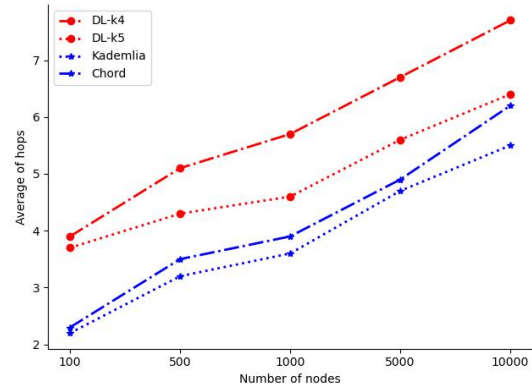


図 2. 各ネットワークの平均ホップ数

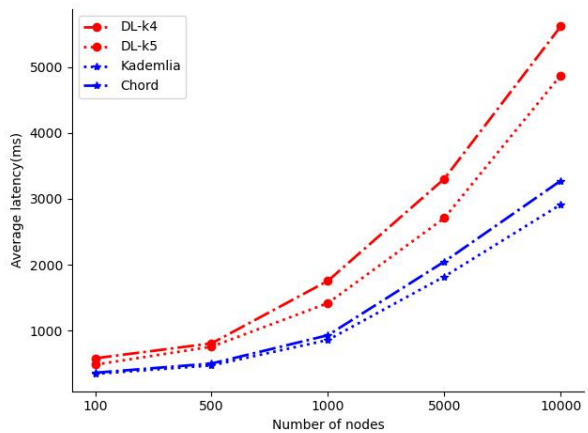


図 3. 各ネットワークの平均遅延

図 3, 4 より、ノード数が増加するに従い、平均ホップ数と平均遅延はそれぞれ増加している。また、

同じノード数のネットワーク下では、DLグラフの平均遅延は Kademia の約 1.6~1.8 倍(初期グラフ K_4)、約 1.3~1.5 倍(初期グラフ K_5)、Chord の約 1.6~1.7 倍(初期グラフ K_4)、約 1.3~1.5 倍(初期グラフ K_5)である。

この結果によって、以下の改良案を提案する。

4. 提案手法

本節では、グラフの平均ホップ数が削減されれば、ネットワークの平均通信遅延の削減に直結すること [12] に着目し、DL グラフ改良案を提案する。

4.1. 提案手法の経路表

2 節で述べたように、DL グラフの経路表は DL グラフで出隣接方向に近接する数個のノードである。提案手法では経路表を 2 ホップまで拡張し、これによって探索ホップ数を半減することが可能とする。例えば、図 1 において 210 の経路表を提案手法のように拡張する場合、新しい経路表は図 4 のようになる。



図 4. 経路表の変化

したがって、提案手法は任意の DL グラフに対して適用することができる。完全有向グラフ K_d を初期グラフとした DL グラフを提案手法に変更する場合、ノード v の出次数(経路表サイズ) $\psi(v)$ (自己ループとマルチエッジを含む)は以下となる：

$$\psi(v) = (d-1) + (d-1)^2 = d^2 - d$$

提案手法に基づく構造化オーバーレイネットワークは、DL グラフの性質をオーバーレイネットワーク上で再現することで、それに加えて探索遅延の低減が可能である。

提案手法のルーティング法は DL グラフと同じく、マッチングインデックスを最大化するように貪欲的に探索クエリの転送先を決める。しかし、異常によって経路上のあるノードが離脱とする場合(図 5 右)、直接に隣接しているマッチングインデックス最大のノードを経由して経路を続ける。

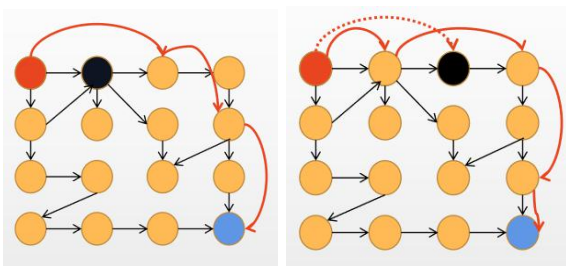


図 5. 近道を導入したのルーティング

4.2. 提案手法のメリット

このような経路表を用いると、以下のメリットがある：

1. 探索の転送回数を削減するためにショートカットとして使われる。

例えば図 5 においては、赤ノードから青ノードまでの転送回数は 6 回から 3 回まで大きく削減される。

2. 耐障害性の高さ。

出隣接が故障等によりネットワークから離脱すると到達性が失われ、探索を正常に行うことができなくなる可能性がある。そのため、提案手法は図 5(左)のように出隣接方向の 2 つ先のノードも経路表に追加すると、あるノードが異常によって離脱しても(黒いノード)、ルーティングが成立できる。それによって、グラフの堅牢性を高めている。

4.3. 実験結果

本節では、提案手法と DL グラフ、DHT に基づく Kademia と Chord を探索遅延の観点から比較する。

まず、3 節で述べたように完全グラフ K_4 と K_5 を初期グラフとして、DL グラフ改良案提案手法に基づくオーバーレイネットワークを実装した。また、提案手法と比較の為、経路表サイズが同じの DL グラフ(初期グラフは完全有向グラフ K_{13} と K_{21})の結果も示している。実験方法は 3 節と同様である。作成したネットワークの平均ホップ数と平均遅延を計算した結果を図 6, 7 に示す。

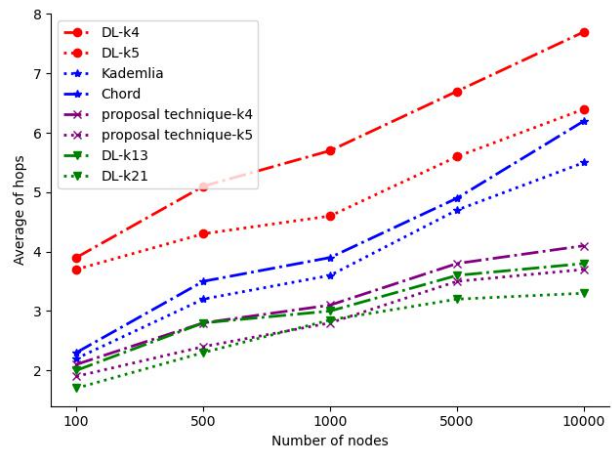


図 6. 各ネットワークの平均ホップ数

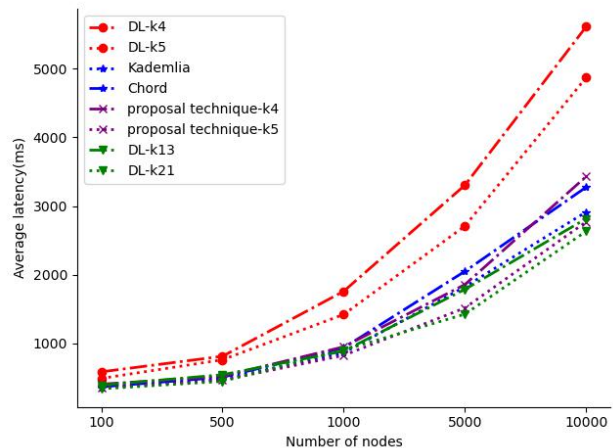


図 7. 各ネットワークの平均遅延 (ms)

5. 考察

本節では、実験結果により、提案手法の有用性に

ついて考察する.

4.1節で述べたように提案手法はDLグラフに対して経路表を2ホップまで拡大するものである. 構造化オーバーレイネットワークのトポロジーは, ネットワーク上経路表に保持するノードと無関係によって, DLグラフの性質を保持することができ(例えば, 出次数が一定), 直接DLグラフを提案手法に変更することもできる.

提案手法は経路表を拡大することによって, 経路表を維持するために必要なトラフィックは大きくなる. ただし, 図6,7より, DLグラフと比べて提案手法は平均ホップ数と平均遅延が30%~60%程度削減することが確認できる. また, 同じノード数のKademlia, Chordと比較して提案手法の探索遅延を10%~15%程度低減することを示した. 増加したトラフィックに対して, 提案手法によって得られた効果が高い.

図6,7より, 改良したDLグラフと同じ経路表サイズを持つDLグラフとの比較すると, わずかながらDLグラフの方が良い性能を示した. 例えば, ノード数5000個, 経路表サイズが12のネットワークにおいて, 提案手法(初期グラフは完全グラフ K_d)はDLグラフ(初期グラフは完全グラフ K_{13})と比べて探索遅延は約2%増加する. その原因として, 経路表を2ホップまで拡大すると, そのノード自身を含める可能性があって, 経路表サイズはDLグラフと全く等しくならぬことが考えられる. 以上の結果によって, 提案手法は初期グラフ K_d を変更せず, 完全有向グラフ K_{d^2-d+1} を初期グラフとしたDLグラフに似た平均遅延を期待できる.

6. まとめ

本研究では, DLグラフを含めたいくつかの典型的な分散ハッシュテーブルの遅延時間を比較した. その結果に基づき, DLグラフの遅延改善として2ホップまで経路表を含める改良案を提案した. その実験結果によって, 初期グラフが同じDLグラフ, KademliaとChordと比較して, 平均ホップ数と平均遅延が優れていることを示した.

今後の課題として, さらに他の分散ハッシュテーブルとの厳密な比較検討をすること, またノードが頻繁に参加・離脱を繰り返す環境下で, トポロジーを維持する状況についてシミュレーションを行うことが挙げられる. その際, 通信の混雑により, 通信遅延は変化するような状況下でのシミュレーションも考えている. また, 提案手法の耐障害性についても検討を行う.

参考文献

- [1] Stoica, Ion, et al. "Chord: A scalable peer-to-peer lookup service for internet applications. " ACM SIGCOMM Computer Communication Review 31. 4 (2001): 149-160.
- [2] Maymounkov, Petar, and David Mazieres. "Kademlia: A peer-to-peer information system based on the xor metric. " International Workshop on Peer-to-Peer Systems. Springer, Berlin, Heidelberg, 2002.
- [3] Li, Dongsheng, Xicheng Lu, and Jinshu Su. "Graph-theoretic analysis of Kautz topology and DHT schemes. " IFIP International Conference on Network and Parallel Computing. Springer, Berlin, Heidelberg, 2004.
- [4] Zhang Y, Liu L, Li D, et al. Distributed line graphs: A universal framework for building DHTs based on arbitrary constant-degree graphs[C]//2008 The 28th International Conference on Distributed Computing Systems. IEEE, 2008: 152-159.
- [5] https://ja.wikipedia.org/wiki/Peer_to_Peer
- [6] Manada A ; Morita H . On some properties of distributed line graphs[C]// International Symposium on Information Theory & Its Applications. IEEE, 2014.
- [7] 白石裕輝. ノード間の近接性を考慮した定数次数構造化オーバーレイネットワーク[J]. 2019.
- [8] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an internet network, " in Proceedings of IEEE INFOCOM '96. Conference on Computer Communications, vol. 2, pp. 594-602 vol. 2, mar 1996
- [9] <https://www.cc.gatech.edu/projects/gtitm/>
- [10] Shudo K, Tanaka Y, Sekiguchi S. Overlay weaver: An overlay construction toolkit[J]. Computer Communications, 2008, 31(2): 402-412.
- [11] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. In Numerische Mathematik, 1 (1959), S. 269 ~ 271.
- [12] 藤原一毅, et al. 小直径グラフ探索コンペ"GraphGolf" 5年間の成果. 研究報告システム・アーキテクチャ (ARC), 2019, 2019. 21: 1-9.