DBSJ Journal, Vol.12, No.1
June 2013

モバイルアドホックネットワーク上のTop-k 検索のための複製配置およびメッセージ処理手法

Replication Strategy and Message Processing Method for Top-k Query in MANETs

佐々木 勇和♠ 原 隆浩♣ 西尾 章治郎♣

Yuya SASAKI Takahiro HARA
Shojiro NISHIO

**本稿では，モバイルアドホックネットワークにおいて，正確な結果の取得と，遅延およびオーバヘッドの削減を目的とし，Top-k 検索のための複製配置とメッセージ処理手法について検討する．まず，複製の配置率を計算し，それに基づいて複製を配置する *FReT* (topology-Free Replication for Top-k query)とよぶ複製配置法を提案する．さらに，正確な結果を取得するまで TTL を増加しながら検索を繰り返す Top-k 検索のメッセージ処理手法を提案する．
シミュレーション実験より，提案手法が高い性能を達成していることを確認した．**

**In this paper, we study a replication strategy and message processing for top-k query in mobile ad hoc networks (MANETs) in order to acquire perfect accuracy of query results with a minimal overhead and delay.
First, we propose *FReT* (topology-Free Replication for Top-k query) where each node replicates data items based on efficient replication ratio. Moreover, we propose a top-k query processing method that repeats sending a query until an exact result is acquired. We verify that our approaches achieves high performance.**

1. はじめに

Top-k 検索は，ユーザが指定する検索条件とスコアリング関数に基づいて，最も関連のある上位k個のデータを検索する．Top-k 検索は，重要なアプリケーションで頻繁に利用されるため，数多くの研究が行われている．例えば，ウェブや検索エンジンにおいて，語句の出現率やテキストの関連性の抽出，センサネットワークにおいて，はずれ値の検出，P2P ネットワークおいて，興味のあるデータの共有などがある．同様に，モバイルアドホックネットワークにおいても，Top-k 検索は有効である．モバイルアドホックネットワークは，災害地における救助活動などへの応用が期待されており，レスキュー隊員が自身のもつ端末に被災者や建物の情報を入力し，Top-k 検索によって重傷患者や損壊の激しい建物を検索することが考えられる．しかし，モバイルアドホックネットワークにおける Top-k 検索の研究は，解決すべき問題が多いにも関わらず，ほとんど行われていない．

モバイルアドホックネットワークでは，端末が自由に移動するため，ネットワークトポロジが動的に変化する．ネットワークトポロジを把握するために，頻繁にメッセージ交換すると，オーバヘッドが大きくなり，パケット衝突が発生し，検索結果の精度が低下してしまう．これを解決するためには，限られた範囲から上位k個のデータを取得し，かつその範囲を周囲の環境に従って動的に決定することが求められる．そこで，本稿では，複製配置を組み合わせた Top-k 検索手法について検討する．複製配置は，データの有用性の向上および検索範囲を小さくすることが可能で，モバイルアドホックネットワークの分野では，頻繁に研究されている．これまで，モバイルアドホックネットワークにおいて，Top-k 検索と複製配置を効果的に組み合わせた手法は提案されていない．

そこで，本稿では，モバイルアドホックネットワークにおいて，限られた狭い検索範囲から正確な結果の取得を保証する，複製配置を組み合わせた Top-k 検索のメッセージ処理手法を提案する．提案手法では，ネットワークの情報が未知であると想定するため，まず上位k個のデータ，およびネットワークの情報を取得し，これらをネットワーク内の端末に配布する．ネットワークの情報および上位k個のデータを受信した端末は，ネットワークの情報を記録し，提案する複製配置手法である FReT (topology-Free Replication for Top-k query)に従って，データを複製として配置する．FReT では，複製配置率と各端末がもつ複製の組合せを，Top-k 検索の際にアクセスする端末数が最少になるように決定する．Top-k 検索を行う端末（クエリ発行端末）は，正確な結果を取得するまで TTL を増加しながら検索を繰り返す．TTL の増加方法として，探索端末数の減少を目的とする Expanding ring 法と，探索遅延の減少を目的とする Bundling 法の二つの方法を用いる．両方の手法とも，小さい遅延およびオーバヘッドで，正確な結果の取得を保証することができる．提案手法の有効性をシミュレーション実験により評価する．

2. 想定環境と問題定義

ネットワーク内には，D個のデータが存在し，識別子$\boldsymbol{d} = \{d_1, d_2, \cdots, d_D\}$が割り当てられているものとする．本稿では，識別子の添え字をデータの順位とする（つまり，d_1は最もスコアが高いデータ，およびd_kはk番目にスコアが高いデータとする）．それぞれのデータは，特定の端末が保持しており，簡単化のためデータのサイズは全て同じとし，更新はないものとする．データのスコアは，検索条件とスコアリング関数によって決定され，検索条件は，データの種類（被災者情報など）と属性値（重症度など）を指定する．本稿では，一つの検索条件とスコアリング関数のみを使用するものとする．

ネットワーク内には，M台の端末が存在し，識別子$\boldsymbol{m} = \{m_1, m_2, \cdots, m_M\}$が割り当てられているものとする．各端末は，自由に移動し，ρ個のデータを複製として配置できるものとする．各端末は，通信半径Rの通信機器を保持しているものとし，GPS などの機器により自身の位置を正確に把握できるものとする．

クエリ発行端末は，K種類のk，$k_i\ (\rho, k_{max}](i = 1, \cdots, K)$，を$kq_i$の確率で指定する．各端末は，ある$k$が指定される確率

♠ 学生会員 大阪大学大学院情報科学研究科
sasaki.yuya@ist.osaka-u.ac.jp
♣ 正会員 大阪大学大学院情報科学研究科
{hara, nishio}@ist.osaka-u.ac.jp

を，統計情報などにより既知であるものとする．

2.1 モバイルアドホックネットワークにおける Top-k 検索の問題

クエリ発行端末は，検索条件と要求データ数(k)を指定したメッセージを送信し，ネットワーク全体で上位k個のスコアをもつデータを取得することを目的とする．Top-k 検索では，クエリ発行端末はデータの識別子を指定して検索を行わないため，正確な結果を取得できたかを判断することはできない．そのため，理想的な複製配置および結果の保証を行うためには，それぞれの端末が上位k個のデータの識別子を知る必要がある．本稿で提案する手法では，まず上位k個のスコアをもつデータを収集し，配布することにより，結果の保証および最適な複製配置を行う．本稿では，上位k個のデータを取得するために必要な平均探索端末数$An(\boldsymbol{kq})$を最小にする複製配置を行う．$An(\boldsymbol{kq})$は以下の式で計算する．

$$An(\boldsymbol{kq}) = \sum_{i=1}^{K} kq_i \cdot An(k_i). \qquad (1)$$

$An(k_i)$は，クエリ発行端末がk_iを指定した場合の平均探索端末数を示す．

2.2 Top-k 検索のための複製配置における問題

Top-k 検索では，アクセスするデータに大きな偏りがあるため，単一データアクセスのための複製配置を，Top-k 検索のための複製配置に用いることは有用とはいえない．例えば，非構造 P2P ネットワークにおける単一データアクセスのための複製配置として，最適な複製数を決定する平方根配置方式が提案されている[1]．平方根配置では，データd_iの複製配置率r_iを以下の式で決定する．

$$r_i = \frac{\sqrt{q_i}}{\sum_{j=1}^{k_{max}} \sqrt{q_j}} \text{ and } \mathrm{l} \le r_i \le \mathrm{u}. \qquad (2)$$

q_iは，d_iを指定する確率，および$\mathrm{l}\ (\geq \frac{1}{M \cdot \rho})$と$\mathrm{u}\ (\leq \frac{1}{M})$は，最小および最大の複製配置率を示す．Top-k 検索では，スコアが最も高いデータは常にアクセスされる．そのため，平方根配置方式を用いた場合，配置率に大きな偏りが発生し，データの多様性が低下する．さらに，単一データアクセスのための複製配置では，データアクセスの依存性を考慮していない．Top-k 検索では，順位が近いデータが同時にアクセスされることが多いと考えられる．

2.2 モバイルアドホックネットワークにおける複製配置の問題

モバイルアドホックネットワークでは，端末は自身と近い端末と通信するため，必要なデータがクエリ発行端末の近くに存在する場合，オーバヘッドが小さくなる．そのため，モバイルアドホックネットワークおよび無線センサネットワークの分野では，位置依存の複製配置手法が数多く提案されている．位置依存の複製配置における，検索のためのコスト$cost$は，以下の式で計算される．

$$cost = \sum_{x \in m} \sum_{i=1}^{K} kq_i \sum_{j=1}^{k_i} dist\left(x, d_j\right). \qquad (3)$$

$dist(x, d_j)$は，クエリ発行端末xとデータd_jを保持する端末との距離を示す．最小の$cost$を達成する場合，最適な複製配置となるが，NP 困難問題として知られている．さらに，モバイルアドホックネットワークでは，端末が自由に移動するため，最適な複製配置が動的に変化する．移動に伴い再配置する場合，オーバヘッドが発生するため，常に最適な配置を維持することは有用とはいえない．また，隣接端末数も動的に変化するため，上位k個のデータを取得するために検索すべき範囲を決定することが難しい．そのため，端末が移動しても，メンテナンスのためのオーバヘッドが必要ない手法，および探索範囲を動的に決定する方法が有効と考えられる．

3. 関連研究

3.1 複製配置

モバイルアドホックネットワークにおける複製配置手法について説明する．文献[3]では，三つのアルゴリズムが提案されている．それぞれのアルゴリズムは，データのアクセス率，隣接端末が保持する複製，およびネットワークトポロジを考慮して，配置する複製を決定する．文献[5]では，LACMA と呼ばれる位置依存の複製配置手法が提案されている．LACMA では，データは特定の区切られた領域（グリッド）に配置され，グリッド内を検索することによって要求データを取得できることを保証する．端末が特定のグリッドから移動する際，そのグリッドに留めるべきデータをプッシュすることにより，データを特定のグリッドに存在させる．これらのプロトコルは，基本的に一回の検索で一つのデータを検索することを想定している．しかし，Top-k 検索では，一度の検索で複数のデータの取得するため，複製データを決定することがより難しい．さらに，提案手法は，端末の移動に伴うメンテナンスコストが発生しないことも目的としている．文献[4]において，モバイルアドホックネットワークにおける Top-k 検索のための複製配置手法が提案されている．この手法では，返信データを受信および中継した端末がそのデータを複製として配置する．文献[4]では，複製配置手法のみを提案しており，複製が配置されている状況における，Top-k 検索手法については検討していない．さらに，この複製配置手法は，スコアの低いデータを必要以上に多く配置してしまう可能性があり，効率的とはいえない．

3.2 Top-k 検索処理

文献[2,8,9]において，モバイルアドホックネットワークにおける Top-k 検索処理手法が提案されている．これらの文献では，クエリ発行端末が検索メッセージをネットワーク全体に送信し，受信した端末はスコアが高いデータのみを返信する．これにより，無駄な返信データの最小化を目的としている．これらの手法では，複製を考慮していない点，ネットワーク全体へメッセージを送信する点，およびパケットロスが発生した場合，正確な結果を取得することができない点が問題となる．しかし，本稿で提案する手法においても，初期手順として，ネットワーク全体からデータを収集することが必要となるため，文献[2]を拡張した手法を用いる．文献[2]では，自身が保持するデータの一部を基準値として，検索メッセージに添付し，上位k個に入らないデータを絞り込み，返信データ数を削減する．

4. 提案手法

本章では，複製配置手法，初期データ収集と配布の方法，および Top-k 検索のメッセージ処理手法について説明する．

4.1 複製配置手法

4.1.1 複製組合せ

Top-k 検索では，順位が近いデータを同時にアクセスする可能性が高い．そのため，各端末はデータをρ位毎に区切って複製する（つまり，d_1からd_ρ，$d_{\rho+1}$から$d_{2\cdot\rho}$および$d_{\left\lfloor\frac{k_{max}}{\rho}\right\rfloor\cdot\rho-\rho+1}$から$d_{k_{max}}$）．$\rho$位毎に区切られたデータの複製を複製組合せとよび，それぞれの複製組合せの配置率をrc_i ($i=1,\cdots,\left\lceil\frac{k_{max}}{\rho}\right\rceil$)とする．FReT では，最適な複製配置となるように，複製組合せの配置率を決定する．

4.1.2 複製配置率

最適に複製配置した場合，各端末は少数の端末の探索により，上位k個のデータを取得することができる．クエリ発行端末がアクセスできる端末数は，隣接端末数とホップ数によって決まり，ホップ数および隣接端末数が多くなるにつれて，アクセスできる端末数は増加する．そこで，まず，あるホップ数hopに対して，アクセスできると予想される端末数n_h，を以下の式で決定する．

$$n_h = ann \times \sum_{j=1}^{hop} j. \qquad (4)$$

annは，平均隣接端末数を示す．この式では，端末が一様に存在すると仮定して，全ての端末が平均隣接端末数と同じ隣接端末数がいるものとし，計算している．さらに，n_h台の端末とアクセスできた場合に，上位k_i個のデータを取得できる確率，$P(k_i,n_h)$を次式で決定する．

$$P(k_i,n_h) = 1-\left(\overline{rc_1}+\cdots+\overline{rc_{\lfloor\frac{k_i}{\rho}\rfloor}}\right) + \left(\overline{rc_1+rc_2}+\cdots+rc_{\lfloor\frac{k_i}{\rho}\rfloor-1}+rc_{\lfloor\frac{k_i}{\rho}\rfloor}\right) + \cdots+(-1)^{\lfloor\frac{k_i}{\rho}\rfloor}\cdot\overline{rc_1+rc_{\lfloor\frac{k_i}{\rho}\rfloor}} \qquad (5)$$

ここで，$\overline{rc_i}$は，$(1-rc_i)^{n_h}$を示す．この式では，n_h台の端末にアクセスした際に，それぞれrc_1から$rc_{\lfloor\frac{k_i}{\rho}\rfloor}$を保持する端末にアクセスできない確率を求め，その余事象の確率を用いて，全ての複製組合せにアクセスできる確率を求めている．この際，n_hが$\left\lceil\frac{k_i}{\rho}\right\rceil$より小さい場合，$P(k_i,n_h)$は 0 とする．$n_h$が増加するほど，必要なデータを取得できる確率が増加するため，$P(k_i,n_h)$も増加する．しかし，n_hは小さいほうがよく，$P(k_i,n_h)$は大きい方がよいため，理想的なホップ数は，最小の$\frac{n_h}{P(k_i,n_h)}$を達成する値とする．加えて，$P(k_i,n_h)$は，データd_1からd_{k_i}がより多く配置されている場合に大きくなるが，小さいkにだけ着目した場合，複製の多様性が低下してしまう．そこで，全体として最小のアクセス端末数を達成するため，それぞれk_iに対してのn_h，n_{h_i}を計算する．これらの要素を考慮して，平均探索端末数$An(\boldsymbol{kq})$，を次式で決定する．

$$An(\boldsymbol{kq}) = \sum_{i=1}^{K}\left(kq_i\cdot\frac{n_{h_i}}{P(k_i,n_{h_i})}\right). \qquad (6)$$

$An(\boldsymbol{kq})$が最小となる場合，クエリ発行端末は，小さいホップ数かつ高確率で要求するデータを取得することができる．従って，FReT では，複製組合せ配置率を$An(\boldsymbol{kq})$が最小となるように決定する．

Algorithm 1 初期検索クエリ

Require: A query condition and k
Ensure: Data items with k highest scores
1: M_q broadcasts FQ
2: **if** Node, M_r receives FQ **then**
3: Stores the information on FQ
4: Sets a reply timer
5: **end if**
6: **if** M_r expires its reply timer **then**
7: **for** Replicas, d_i held by M_r that are not included $list_{qi}$ and are not overheard **do**
8: **rd** ← **rd** ∪ d_i
9: **end for**
10: Sends a reply message to the query issuer
11: **end if**
12: **if** M_r overhears a reply message **then**
13: Stores **rd**.
14: **end if**
15: **if** M_q acquires the top-k result **then**
16: Query is over
17: **else if** M_q wait a maximum reply timer **then**
18: Stores the number of nodes replied for the bundling method
19: Go to Algorithm 2
20: **end if**

4.2 初期収集および配布手法

最適な複製配置を行うためには，各端末がネットワークの情報を知る必要がある．そのため，Top-k 検索を最初に行う端末M_cは，初期メッセージを転送し，上位k_{max}個のデータ，端末数および端末の位置情報を収集する．収集のオーバヘッドをできる限り小さくするために，文献[7]で提案されている位置ベースフラッディングと，文献[2]で提案されている Top-k 検索手法を組み合わせて，収集を行う．M_cは，情報を収集後に，上位k_{max}個のデータ，および平均隣接端末数の情報を，位置ベースフラッディングを用いて全ての端末に配布する．受信した端末は，上位k_{max}個のデータの識別子，および平均隣接端末数を把握し，複製を FReT で決定した複製配置率に基づいて配置する．

4.3 メッセージ処理手法

本節では，Top-k 検索手法である，Expanding ring 法と Bundling 法について説明する．両手法とも，クエリ発行端末は，まず`初期検索クエリ'を発行し，その後上位$k$個のデータを取得するまで`再検索クエリ'を繰り返し発行する．
Expanding ring 法と Bundling 法の違いは，TTL の決定法である．

初期検索クエリと再検索クエリのメッセージ処理手順を，それぞれアルゴリズム 1 およびアルゴリズム 2 に示す．初期検索クエリFQは，クエリ発行端末の識別子，クエリの識別子，検索条件，k，およびクエリ発行端末が保持する複製のリスト($list_{qi}$)が含まれている．再検索クエリRQは，クエリ発行端末の識別子，クエリの識別子，要求するデータの識別子，送信端末の識別子(s)，送信端末の位置，および TTL が含まれている．両方のクエリに対する，返信メッセージは，クエリ発行端末の識別子，クエリの識別子，送信端末の識別子，送信端末の親端末の識別子，および返信データ($\boldsymbol{rd}$)が含まれている．それぞれの端末は，メッセージを傍受し合うことにより，返信データ数の削減を行う（アルゴリズム 1 内の 12-14

Algorithm 2 再検索クエリ

Require: Identifiers of demand data items
Ensure: Demand data items
1: M_q broadcasts RQ
2: **if** Node, M_r receives RQ **then**
3: **if** Receives first **then**
4: Stores the information on RQ
5: Decreases TTL by 1
6: **if** Has demand data items **then**
7: Sends a reply message to its parent
8: Demand data items ← demand data items - **rd**
9: **end if**
10: **if** $TTL > 0$ **and** demand data items ≠ $NULL$ **then**
11: Updates RQ
12: Sets a query timer
13: **end if**
14: **end if**
15: Updates neighbor node
16: **end if**
17: **if** M_r expires its query timer **then**
18: Calculates communication range of sender nodes
19: **if** Communication range of M_r is not covered **then**
20: Broadcasts RQ
21: Sets a reply timer
22: **end if**
23: **end if**
24: **if** M_r receives a reply message **then**
25: **for** Reply data items rd_i are not sent to parent node, and are not overheard **do**
26: **rd** ← **rd** ∪ rd_i
27: **end for**
28: **if rd** ≠ $NULL$ **then**
29: Sends a reply message to its parent
30: **end if**
31: **else if** M_r overhears a reply message **then**
32: Stores **rd**
33: **end if**
34: **if** M_q acquires the top-k result **then**
35: Query is over
36: **else if** M_q expires a timer determined based on TTL **then**
37: Updates RQ
38: M_q broadcasts RQ again
39: **end if**

行，およびアルゴリズム 2 内の 31-33 行)．

初期検索クエリと再検索クエリには，大きく 3 つの違いがある．まず，前者では，kと検索条件を指定するのに対して，後者は要求データの識別子を指定する．再検索クエリの場合，単純に必要なデータの条件を指定することができないため，識別子を添付する．また，初期検索クエリでは，クエリ発行端末が保持する複製が返信されるのを防ぐために，$list_{qi}$を添付する．次に，前者は位置ベースフラッディングを行わないため，位置情報を添付しない．最後に，後者は，探索範囲を広げるために TTL を設定するが，前者では行わない（TTL は常に 1)．これは，上位k個のデータを近くの端末から取得するため，および現在の隣接端末数をできる限り早く取得するためである．

4.3.1 Expanding ring 法

P2P ネットワークにおける，Expanding ring 法は，文献[6]で提案されており，探索端末数を最小にするため，TTL を徐々に増加させていく．提案する Expanding ring 法では，初期検索クエリの TTL を 1 に設定，および再検索クエリの TTL は，前回のクエリの TTL より 1 増加させた値とする．この方法は，返信データ数を最小にすることができ，オーバヘッド削減に効果的である．しかし，隣接端末数が少ない場合，再検索クエリの回数が多くなり，遅延が増加してしまう．

4.3.2 Bundling 法

Bundling 法では，クエリ発行端末は自身の隣接端末数を考慮して，TTL を設定する．しかし，クエリ発行端末は自身の現在の隣接端末数cnnを知らないため，初期検索クエリの TTL は 1 とする．初期検索クエリ発行後に，クエリ発行端末は，隣接端末の情報を更新できるため，最初の再検索クエリの TTL を一度に大きくする．具体的には，クエリ発行端末がk_iを指定するとき，以下の式を満たす TTL を設定する．

$$cnn \times \sum_{j=1}^{TTL} j > \frac{n_{h_i}}{P(k_i, n_{h_i})}. \qquad (7)$$

n_{h_i}および$P(k_i, n_{h_i})$は，それぞれ最適な複製配置における，式(6)の値である．この式は，高い確率で上位k_i個のデータを取得できる TTL(≥ 2)を決定する．もし，クエリ発行端末が式(7)で設定した TTL で上位k個のデータを取得できなかった場合，それ以降は 1 ずつ増加させる．Bundling 法では，TTL を一度に大きくできるため，遅延を小さくすることができるが，不必要に TTL を大きくしてしまった場合，Expanding ring 法に比べて，返信データ数は増加する．

5. 性能評価

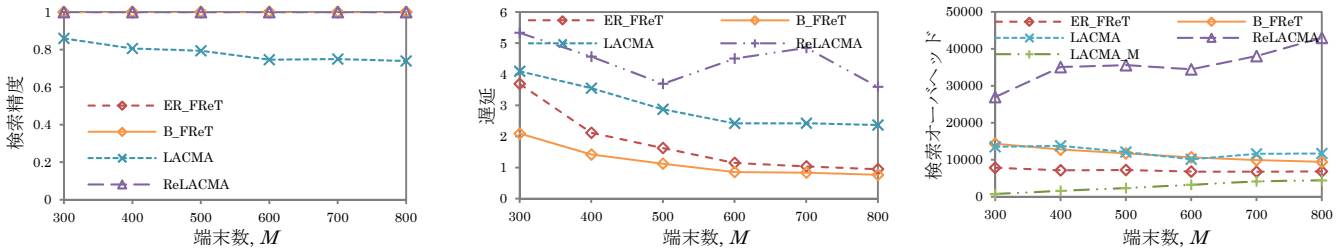
5.1 評価環境

1,000[m] ×1,000[m]の 2 次元平面上の領域にM台の端末（$m_1, \cdots, m_M$）が存在し，初期位置をランダムに決定した．各端末はランダムウォークモデルに従い，0.5 から 1[m/秒]の速度で移動する．各端末は，IEEE802.11b を使用し，伝送速度 11[Mbps]，通信伝搬距離が 100[m]程度となる送信電力でデータを送信する．各端末は，128[B]のサイズのデータをそれぞれ 100 個保持するものとし，5 個のデータを複製として配置することができる(ρ=5)．クエリ発行端末は，kとして，25，50，75，および 100 を等しい確率で指定し，30 秒間隔でクエリを発行する．以上のシミュレーション環境において，端末台数を変化させ，クエリを 300 回発生させた場合の評価値を調べた．

- 検索精度: 上位k個のデータうち，取得できたデータ数の平均割合．
- 遅延 [秒]: 検索クエリ発行後，データ取得までに経過した平均時間．
- 初期オーバヘッド [Kbytes]: 初期データ収集および配布における，初期メッセージの総バイト数．
- 検索オーバヘッド [Kbytes]: シミュレーション終了(クエリ数 300)までに発生した検索クエリおよび返信メッセージの総バイト数．

5.2 比較手法

位置依存の複製配置手法 LACMA と，提案手法である Expanding ring 法および Bundling 法を比較する（グラフの凡例; ER_FReT，および B_FReT)．



(a) 検索精度　(b) 遅延　(c) 検索オーバヘッド

図 2 Top-k 検索手法の評価
Fig. 2 Experiment for Top-k query processing

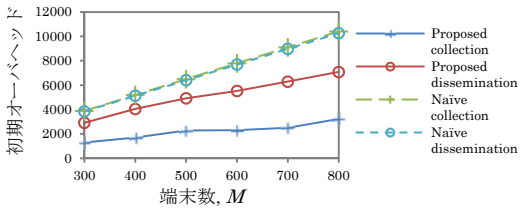


図 1　初期収集および配布手法の評価
Fig. 1 Experiment for initial procedures

- LACMA（凡例; LACMA）：データは，kのアクセス頻度と端末数によって決定されるグリッドに従って配置される．検索の際は，クエリ発行端末は，検索メッセージを自身が所属するグリッドにフラッディングし，上位k個のデータを取得する．あらかじめ設定された時間内に，上位k個のデータを取得できない場合，検索を終了する．また，LACMA では，複製を特定のグリッドに留めるため，端末は自身が現在所属するグリッドから移動する際に，留めておくべき複製をプッシュし，その複製を自身の複製領域から削除する（メンテナンスに係るオーバヘッドの凡例を LACMA_M とする）．複製を受信した端末は，自身の複製領域に空きがあれば，そのデータを複製として配置する．
- Reiterate LACMA（凡例; ReLACMA）：LACMA を用いてデータを取得後，上位k個のデータを取得できなかった場合，クエリ発行端末は提案した再検索クエリを上位k個のデータを取得できるまで繰り返す．この際，最初の再検索クエリの TTL を$\frac{grid_width\cdot\sqrt{2}}{R}$とし，その後，1 ずつ増加させる．

さらに，2 つの複製配置方式と提案手法 FReT と比較する．

- 一様配置: 全ての上位k個のデータを同じ確率で複製として配置する．
- 平方根配置: 文献[1]で提案されている平方根配置方式に従って，データを複製として配置する．

これらの複製配置方式と比較する際は，Expanding ring 法および Bundling 法を Top-k 検索処理手法として用いる（グラフの凡例; ER_uni, B_uni, ER_SQRT, および B_SQRT）．

5.3 評価結果

5.3.1 初期収集および配布

まず，初期収集および，初期配布手法の性能を，端末数Mを変化させて調べた．比較手法として，単純なフラッディングおよび自身の保持するデータと受信したデータのうち上位k個のデータを返信する単純手法（凡例; Naive collection, および Naive dissemination）を用いる．その結果を図 1 に示す．この結果から，提案した初期収集および配布手法は，単純手法より低いオーバヘッドを達成しているのがわかる．これは，不必要なデータ返信および，位置ベースフラッディングにより，メッセージ送信を抑制できているためである．初期収集の方が初期配布に比べて，オーバヘッドが小さいのは，配布では，k_{max}（つまり，100)個のデータを送信するのに対し，収集では，返信データ数がk_{max}より小さいためである．

5.3.2 Top-k 検索処理

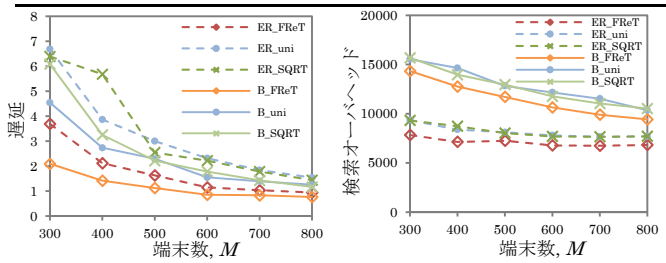
次に，Top-k 検索処理手法の性能を，端末数Mを変化させて調べた．その結果を図 2 に示す．図 2 (a)より，提案手法は正確な結果を取得できていることがわかる．一方，再検索クエリを用いない LACMA では，正確な結果を取得できてない．これは，端末の移動，端末の疎密差，およびパケットロスが原因である．

図 2 (b)より，Bundling 法が最も小さい遅延を達成していることがわかる．これは，TTL を一度に大きくすることにより，探索範囲を上位k個のデータを高い確率で取得できる範囲に一度で拡大できるためである．また，Expanding ring 法も同様に小さい遅延を達成している．これは，Bundling 法に比べ，検索クエリの送信回数は多いが，小さい探索範囲を達成できているためである．端末数が多くなると，より近い端末から上位k個のデータを取得しやすくなるため，Expanding ring 法と Bundling 法の遅延の差が小さくなる．一方，LACMA では，上位k個のデータを取得できないことが多く，最大待ち時間まで待ってしまうため，遅延が大きい．

図 2(c)より，Expanding ring 法が最も小さいオーバヘッドを達成しているのがわかる．これは，より近くの端末から上位k個のデータを取得できているためである．Bundling 法においても，正確な上位k個のデータを取得しながら，LACMA とほぼ同等のオーバヘッドを達成することができている．Reiterate LACMA は，非常に大きなオーバヘッドが発生しているのがわかる．これは，LACMA では，端末がグリッドを移動した際に，複製を削除するため，探索範囲が大きくなってしまうためである．さらに，LACMA では，端末数が増加すると，メンテナンスのオーバヘッドも増加する．

ここで，初期収集のオーバヘッドは，要求データ数kが 100 の時の Top-k 検索の 1 回分のオーバヘッドとほぼ同様と考えることができる．図 1 の初期オーバヘッドと検索オーバヘッドを比較して，検索オーバヘッドは Top-k 検索の 300 回分のオーバヘッドであるにも関わらず，非常に低く抑えられていることがわかる．これにより，複製の配置により探索範囲を小さくすることは，収集および配布にかかるオーバヘッドを考慮しても，ネットワーク全体として，オーバヘッドを大幅に抑制できていることがわかる．

この結果より，提案手法は，小さい遅延およびオーバヘッ



(a) 遅延　　(b) 検索オーバヘッド

図 3　複製配置手法の評価
Fig.3 Experiment for replication strategy

ドで，取得精度を維持できていることがわかる．さらに，FReT では，位置依存の複製配置手法より，端末の移動に対して耐性があり，高い性能を達成していることがわかる．

5.3.3 複製配置

最後に，複製配置手法の性能を，端末数Mを変化させて調べた．その結果を図 3 に示す．図 3 (a)より，FReT を用いた Bundling 法が最も小さい遅延を達成していることがわかる．また，FReT を用いた Expanding ring 法も同様に小さい遅延を達成している．これは，FReT が適切な複製の多様性を達成しているためである．一方で，平方根配置方式の性能は低くなっている．これは，複製の多様性が低くなっており，再検索クエリの送信数が多くなってしまうためである．

図 3 (b)より，FReT を用いた Expanding ring 法のオーバヘッドが最も小さい．また，FReT を用いた Bundling 法も他の複製配置手法を用いた Bundling 法より小さいオーバヘッドを達成している．ここで，一様配置手法と平方根配置方式のオーバヘッドがほぼ同様なのは，上位k個のデータを取得するための平均ホップ数がほぼ同じとなっているためである．

この結果より，FReT では，遅延およびオーバヘッドの両方の観点において，高い性能を達成していることがわかる．

6. おわりに

本稿では，モバイルアドホックネットワークにおける，Top-k 検索のための複製配置とメッセージ処理手法を提案した．提案した複製配置手法，FReT，では，複製のメンテナンスコストなしで，上位k個のデータを取得するための探索端末数を小さくすることを実現した．提案した Top-k 検索のメッセージ処理手法では，検索クエリ発行端末が TTL を増加させながら，上位k個のデータを取得するまで，検索クエリを繰り返して送信する．TTL の増加方法として，Expanding ring 法と Bundling 法を提案した．Expanding ring 法では，最少の探索端末数を達成し，Bundling 法では，より小さい遅延を達成した．それぞれの手法ともに，上位k個のデータを確実に取得することができる．シミュレーション実験によって，提案手法が高い性能を示していることを確認した．

本稿では，簡単化のため，いくつかの想定を単純化した．例えば，同じデータサイズ，同じサイズの複製領域，およびデータの更新が発生しないことなどがあげられる．しかし，これらは，実環境において必ずしも正しいとはいえない．そのため，これらの想定を外した場合の複製法およびメッセージ処理手法について，今後取り組む予定である．

[謝辞]

本研究の一部は，文部科学省研究費補助金・基盤研究 S（21220002），基盤研究 B（24300037），および特別研究員奨励費（24-293）の研究助成によるものである．ここに記して謝意を表す．

[文献]

[1] E. Cohen and S. Shenker. Replication strategies in unstructured peer-to-peer networks. In *ACM SIGCOMM Computer Communication Review*, vo. 32, no. 4, pp.177–190, 2002.
[2] R. Hagihara, M. Shinohara, T. Hara, and S. Nishio. A reduction in ad hoc networks. In *MDM*, pp.11–20. IEEE, 2009.
[3] T. Hara. Effective replica allocation in ad hoc networks for improving data accessibility. In *INFOCOM, pp.* 1568–1576, 2001.
[4] T. Hara, R. Hagihara, and S. Nishio. Data replication for top-k query processing in mobile wireless sensor networks. In *SUTC, pp.* 115–122. IEEE, 2010.
[5] S. Lee, S. Wong, K. Lee, and S. Lu. Content management in a mobile ad hoc network: Beyond opportunistic strategy. In *INFOCOM, pp.* 266–270. IEEE, 2011.
[6] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and replication in unstructured peer-to-peer networks. In *ICS*, pp. 84–95, 2002.
[7] S. Ni, Y. Tseng, Y. Chen, and J. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Mobicom*, pp. 151–162, 1999.
[8] Y. Sasaki, R. Hagihara, T. Hara, M. Shinohara, and S. Nishio. A top-k query method by estimating score distribution in mobile ad hoc networks. In *Advanced Information Networking and Applications Workshops (WAINA)*, pp. 944–949, 2010.
[9] Y. Sasaki, T. Hara, and S. Nishio. Two-phase top-k query processing in mobile ad hoc networks. In *Network-Based Information Systems (NBiS), pp.* 42–49, 2011.

佐々木 勇和　Yuya SASAKI
大阪大学大学院情報科学研究科博士後期課程在学中．2011 年大阪大学大学院情報科学研究科博士前期課程修了．モバイル環境におけるデータ検索技術に関する研究に従事．情報処理学会学生会員．

原 隆浩　Takahiro HARA
大阪大学大学院情報科学研究科准教授．1997 年大阪大学大学院工学研究科博士前期課程修了．工学博士．データベースシステム，分散処理の研究に従事．IEEE，ACM，電子情報通信学会各会員．

西尾 章治郎　Shojiro NISHIO
大阪大学大学院情報科学研究科教授，サイバーメディアセンター長．1975 年京都大学工学部卒業．1980 年同大学院工学研究科博士後期課程修了，工学博士．京都大学工学部助手等を経て，1992 年大阪大学工学部教授となり，現職に至る．文部科学省科学官，大阪大学理事・副学長等を歴任．データ工学の研究に従事．本会理事，監事を歴任し，現在，会長を務める．紫綬褒章を受章し，本会より功労賞，論文賞を受賞．