

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 およびメッセージ処理手法

佐々木勇和[†] 原 隆浩[†] 西尾章治郎[†]

[†] 大阪大学大学院情報科学研究科マルチメディア工学専攻 〒565-0871 大阪府吹田市山田丘1番5号

E-mail: †{sasaki.yuya,hara,nishio}@ist.osaka-u.ac.jp

あらまし 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、遅延およびオーバーヘッドの削減を目的とし、Top-k 検索処理と複製配置を組み合わせた手法を提案する。これまでの研究では、Top-k 検索のための最適な複製配置、および精度を維持する Top-k 検索手法は提案されていない。そこで、まず最適な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。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。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。
キーワード モバイルアドホックネットワーク, Top-k 検索, 複製配置

Yuya SASAKI[†], Takahiro HARA[†], and Shojiro NISHIO[†]

[†] Department of Multimedia Engineering, Graduate School of Information Science and Technology,

Osaka University, 1-5 Yamadaoka, Suita, Osaka, 565-0871 Japan

E-mail: †{sasaki.yuya,hara,nishio}@ist.osaka-u.ac.jp

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 ネットワークにおいて、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における、Top-k 検索処理において、考えるべき点は4つあげられる。まず、移動端末であるため、容量やバッテリーに制約があることがあげられる。2つ目として、基地局が存在しないことや、パケットロスにより、

結果の精度を保証することが難しい。3つ目として、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。最後に、頻繁にメッセージ交換すると、オーバーヘッドが大きくなり、パケット衝突が発生する。これら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、これを実現するために、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。これまでに、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。さらに、多くの複製配置手法では、移動端末が事前にネットワークの情報（領域サイズや、端末数、ネットワークトポロジなど）を把握していることを前提としている。

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 *FRtT* (topology-Free Replication for Top-k query) に従って、データを複製として配置する。FRtT 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ーヘッドで、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。

以下では、2. で想定環境と問題定義について述べ、3. で関連研究について述べる。その後、4. で提案手法について説明し、5. でシミュレーション実験の結果を示す。最後に 6. で本稿のまとめと今後の課題について述べる。

2. 想定環境と問題定義

本稿では、アドホックネットワークを構成する各端末が、自身と他の端末のもつデータに対して Top-k 検索を行う環境を想定する。クエリ発行端末は、検索条件を指定して検索クエリを発行し、ネットワーク内の上位 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。

ネットワーク内には、 D 個のデータが存在し、識別子 $\mathbf{d} = \{d_1, d_2, \dots, d_D\}$ 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 $\mathbf{m} = \{m_1, m_2, \dots, m_M\}$ 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 、 $k_i (\rho, k_{max}) (i = 1, \dots, K)$ 、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。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、 $An(\mathbf{kq})$ 、を最小にする複製配置を行う。 $An(\mathbf{kq})$ は以下の式で計算する。

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

$An(k_i)$ 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。

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

q_i は、 d_i を指定する確率、および $l (\geq \frac{1}{M \cdot \rho})$ と $u (\leq \frac{1}{M})$ は、最小および最大の複製配置率 (少なくとも一台が複製を配置する、およびいずれの端末も同じ複製を複数保持しないものとする) を示す。

Top-k 検索では、スコアが高いデータがスコアが低いデータより頻繁にアクセスされる。そのため、データ d_i の複製配置率は、データ d_{i+1} の複製配置より大きくなる。

$$r_i \geq r_{i+1}. \quad (3)$$

しかし、平方根配置方式を用いた場合、スコアが高いデータの配置率が大きくなりすぎてしまい、データの多様性が低下する。その場合、大きい k が指定された場合、遅延およびオーバーヘッドが大きくなってしまふ。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。

2.3 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、オーバー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。

$$cost = \sum_{x \in m} \sum_{i=1}^K kq_i \sum_{j=1}^{k_i} dist(x, d_j). \quad (4)$$

$dist(x, d_j)$ 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。また、隣接端末数も同様に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。この場合、複製が常に特定の領域に存在するように、各端末がその領域から移動する際に、複製を交換することで、最適な複製配置を保つことができる。しかし、メンテナンスのためのオーバーヘッドが発生し、さらに端末の密度が変化した場合、複製が存在すべき領域に留めることができない可能性がある。したがって、端末が移動しても、メンテナンスのためのオーバー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

様々なネットワーク環境において、複製配置手法、および Top-k 検索手法が提案されている。ここでは、代表的な既存手法について説明する。

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、データの可用性の向上を目的として、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[6]では、LACMA(location-aided content management architecture)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。グリッドのサイズは、データのアクセス率によって決定され、より頻繁にアクセスされるデータが配置されるグリッドは小さくなる。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。しかし、グリッド内に端末が存在しない場合、データのプッシュは成功しない。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 検索のような、ランクベースのクエリでは、一度の検索で複数のデータの取得する。提案手法では、小さい探索端末数で k 個のデータを取得することを目的としており、これらの手法より複製データの配置を決定することが難しい。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。その際、データのスコアと乱数を用いて配置する複製を決定するため、スコアが高いデータがよく配置されつつ、多様性も維持することができる。しかし、この手法では、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 検索手法は提案されていない。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、理想的な複製配置数を達成することはできない。

3.2 Top-k 検索処理

様々なネットワーク環境において、Top-k 検索手法が提案されている。ここでは、無線センサネットワークおよびモバイルアドホックネットワークにおける Top-k 検索処理手法について説明する。

無線センサネットワークにおける Top-k 検索では、通信量の削減とバッテリーの高寿命化を目的として、シンクがマルチホップ通信でデータを収集する方法が数多く提案されている。文献[12]では、各端末にフィルタを配布して、そのフィルタを通過するデータのみをシンクに送信する手法を提案している。フィルタを通過するデータは、上位 k 個のデータに影響を与えるデータとなる。文献[5]では、文献[13]で提案されている集中管理データベースにおける効率的な Top-k 検索のためのデータ構造であるドミナントグラフを用いたフィルタにより Top-k 検索を効率的に行う方法を提案している。無線センサネットワークでは、固定端末およびネットワーク内で一つのシンクを想定している点で、モバイルアドホックネットワークとは異なる。

筆者らの知る限りでは、モバイルアドホックネットワークにおける Top-k 検索処理手法は、文献[2],[9][10],[11]でのみ提案されている。文献[9]での想定環境(例えば、仮想的な通貨)は、本稿で想定する環境と本質的に違う。文献[2],[10],[11]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む。検索クエリ送信時に、基準値の更新および、クエリ発行端末を根とする木構造の親子関係を構築する。返信時は、まず子端末がいない端末が、基準値のうち最も小さい値を閾値として、閾値以上のスコアをもつデータのみ親端末へ返信する。全ての子端末から返信データを受信した端末は、同様にデータを自身の親端末へ返信する。最終的に、クエリ発行端末は、パケットロスが発生しない限り、上位 k 個のスコアをもつデータを無駄な返信データを抑制しつつ、取得することができる。

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、順位に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 $d_{2\rho}$ および $d_{\lceil \frac{k_{max}}{\rho} \rceil \cdot \rho - \rho + 1}$ から $d_{k_{max}}$ ）。 ρ 位毎に区切られたデータの複製を複製組合せと呼び、それぞれの複製組合せの配置率を $rc_i (i = 1, \dots, \lceil \frac{k_{max}}{\rho} \rceil)$ とする。FRet では、最適な複製配置となるように、複製組み合わ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。

$$n_h = ann \times \sum_{j=1}^{hop} j. \quad (5)$$

ann は、平均隣接端末数を示す。この式では、端末が一樣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あるものとし、計算している。隣接端末数が極端に多い場合、この式は正確ではないが、本稿では、それほど多い隣接端末数を想定していない。

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。

$$\begin{aligned} P(k_i, n_h) = & 1 - (\overline{rc_1} + \dots + \overline{rc_{\lceil \frac{k_i}{\rho} \rceil}}) \\ & + (\overline{rc_1} + \overline{rc_2} + \dots + \overline{rc_{\lceil \frac{k_i}{\rho} \rceil - 1}} + \overline{rc_{\lceil \frac{k_i}{\rho} \rceil}}) \\ & + \dots \\ & + (-1)^{\lceil \frac{k_i}{\rho} \rceil} \cdot \overline{rc_1} + \dots + \overline{rc_{\lceil \frac{k_i}{\rho} \rceil}}. \end{aligned} \quad (6)$$

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 $rc_{\lceil \frac{k_i}{\rho} \rceil}$ 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 $\lceil \frac{k_i}{\rho} \rceil$ 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 $\frac{n_{h_i}}{P(k_i, n_{h_i})}$ を達成する値となる。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 $An(\mathbf{k}q)$ を次式で決定する。

$$An(\mathbf{k}q) = \sum_{i=1}^K (kq_i \cdot \frac{n_{h_i}}{P(k_i, n_{h_i})}) \quad (7)$$

$An(\mathbf{k}q)$ 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、複製組み合わせ配置率を $An(\mathbf{k}q)$ が最小とな

るように決定する。

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。収集のオーバーヘッドをできる限り小さくするために、文献[8]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、収集を行う。 M_c は、情報を収集後に、上位 k_{max} 個のデータ、ネットワーク内の端末数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、ネットワーク内の端末数、および平均隣接端末数を把握し、複製を FRet に従って配置する。

初期データ収集の手順をアルゴリズム 1 に示す。初期メッセージ IM には、検索条件、 k_{max} 、送信端末の識別子 (s)、送信端末の親端末の識別子 (p)、送信端末の位置、および基準値が含まれている。返信メッセージには、送信端末の識別子、送信端末の親端末の識別子とスコア、端末の位置のリスト、閾値、および閾値以上のスコアをもつデータが含まれている。初期メッセージの転送時では、各端末は、ACK を送信することにより、親子関係を構築する。このメッセージ処理では、全ての子端末からの返信が返ってこない場合、返信を開始しない (18–20 行)。さらに、 M_c が事前に設定した時間 ($max\ wait$) 内に全ての子端末から受信できなかった場合、もう一度初期メッセージの転送を行う (23–25 行)。これにより、すべての端末から情報を収集することができる。

4.3 Top-k 検索の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 2 およびアルゴリズム 3 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 ($list_{q_i}$) 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 (\mathbf{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 2 内の 12–14 行、およびアルゴリズム 3 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。 k と検索条件を指定した方が、メッセージサイズは小さいが、再検索クエリの場合、単

Algorithm 1 初期収集方法

Require: A query condition and k_{max}
Ensure: Data items with k_{max} highest scores and positions of nodes

- 1: M_c broadcasts IM
- 2: **if** Node, M_r receives IM **then**
- 3: **if** Receives first **then**
- 4: Stores the information on IM
- 5: Sets a query timer
- 6: **end if**
- 7: Updates SS and neighbor nodes
- 8: **end if**
- 9: **if** M_r expires its query timer **then**
- 10: Calculates communication range of neighbor nodes //Proposed in [8]
- 11: **if** Communication range of M_r is not covered **then**
- 12: Broadcasts IM
- 13: Sets a reply timer
- 14: **else**
- 15: Sends a reply message to its parent
- 16: **end if**
- 17: **end if**
- 18: **if** M_r expires its reply timer and has no children **or** receives replies from its all children **then**
- 19: Sends a reply message to its parent
- 20: **end if**
- 21: **if** M_c receives replies from all its children **then**
- 22: Starts distributing the collecting information by using the location-based flooding
- 23: **else if** $max\ wait$ has passed from sending IM by M_c **then**
- 24: M_c broadcasts IM again
- 25: **end if**

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、後者は位置ベースフラディングを用いるのに対して、前者は行わない。これは、後者は、TTLが大きくなっていくため、位置ベースフラディングを使う機会があるのに対して、前者では、TTLが1であるため、使う機会がない。そのため、メッセージサイズを不要に大きくすることを防ぐために、位置情報の添付を行わない。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[7]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、単純であるが、返信データ数を最小にすることができ、オーバー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。

Algorithm 2 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: $rd \leftarrow rd \cup d_i$
- 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 3
- 20: **end if**

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。

$$cnn \times \sum_{j=1}^{TTL} j > \frac{n_{h_i}}{P(k_i, n_{h_i})}. \quad (8)$$

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(7)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(8)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

本章では、提案手法の性能評価のために行ったシミュレーション実験の結果を示す。本実験では、ネットワークシミュレータ Qualnet5.2 を用いた^(注1)。

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

(注1) : Scalable Networks: makers of QualNet and EXata, the only multi-core enabled network simulation and emulation software. [Online]. Available: <http://www.scalable-networks.com/>.

表1 メッセージサイズ

メッセージ	サイズ [bytes]
初期収集	288
初期収集返信	$32+128 \cdot i_1+16 \cdot i_2$
初期収集 (単純手法)	44
初期収集返信 (単純手法)	$28+128 \cdot i_1+16 \cdot i_2$
ACK (初期収集)	16
配布	12844
初期検索クエリ	36
再検索クエリ	$36+4 \cdot i_3$
LACMA におけるクエリ	40
返信 (全手法共通)	$24+128 \cdot i_4$
プッシュ (LACMA におけるメンテナンス)	$32+128 \cdot i_5$

Algorithm 3 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  $\leftarrow$  demand data items - rd
9:     end if
10:    if  $TTL > 0$  and demand data items  $\neq NULL$  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:     $rd \leftarrow rd \cup rd_i$ 
27:  end for
28:  if  $rd \neq NULL$  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

```

(m_1, \dots, 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 個を保持するものとし、5 個のデータを複製として配置することができる ($\rho = 5$)。

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。

- **初期オーバーヘッド [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。メッセージサイズを表 1 に示す。この表において、 i_1 は、初期収集の返信メッセージに含まれるデータ数、および i_2 は、初期収集の返信メッセージに含まれる端末の位置情報数を示す。

- **検索オーバーヘッド [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。メッセージサイズを表 1 に示す。この表において、 i_3 は、再検索クエリに含まれる要求データの識別子数、 i_4 は、返信メッセージに含まれるデータ数、および i_5 は、プッシュメッセージに含まれるデータ数を示す。

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)。

(1) LACMA (凡例; LACMA) : データは特定のグリッドに配置される。本実験では、グリッドのサイズは k のアクセス頻度と端末数によって決定する (つまり、4 種類のグリッドが存在する)。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グディングし、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必要があるため、端末が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ーヘッドの凡例を 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。

(2) 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 reiterate query 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。

(1) 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。

(2) 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. このグラフにおいて, 横軸は端末数 M を表し, 縦軸は, 初期オーバーヘッドを表す.

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ーヘッドを達成しているのがわかる. これは, まず初期収集では, 不必要なデータ返信および, メッセージ送信を抑制できているためである. 初期配布においては, 位置ベースフラッディングにより, メッセージ送信数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, オーバー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, 基準値により返信データ数が k_{max} より小さいためである. また, 単純手法における収集と配布では, 全ての端末が k_{max} 個のデータを送信するため, ほぼ同じ結果となっている.

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. これらのグラフにおいて, 横軸は端末数 M を表し, 縦軸は, 図 2 (a) では, 検索精度, 図 2 (b) では, 遅延, および図 2 (c) では, 検索オーバーヘッドをそれぞれ表す.

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, 提案手法より遅延が大きい. これは, 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つてしまうためである.

図 2 (c) より, Expanding ring 法が最も小さいオーバー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

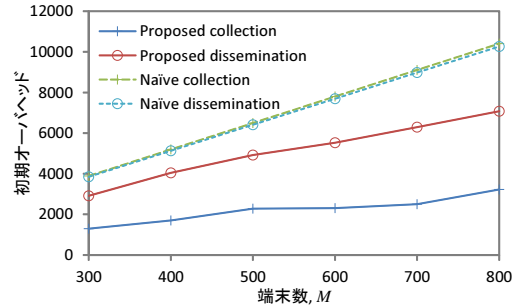


図 1 初期収集および配布手法

ぼ同等のオーバー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ー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いうためである. 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ーヘッドも増加する. ここで, 初期収集のオーバーヘッドは, 要求データ数 k が 100 の時の Top-k 検索の 1 回分のオーバーヘッドとほぼ同様と考えることができる. 図 1 の初期オーバーヘッドと検索オーバーヘッドを比較して, 検索オーバーヘッドは Top-k 検索の 300 回分のオーバー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ーヘッドを考慮しても, ネットワーク全体として, オーバー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ーヘッドで, 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. これらのグラフにおいて, 横軸は端末数 M を表し, 縦軸は, 図 3 (a) では, 遅延, および図 3 (b) では, 検索オーバーヘッドをそれぞれ表す.

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いうためである. 図 3 (b) より, FReT を用いた Expanding ring 法のオーバー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ー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ー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, 遅延およびオーバー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, 複製

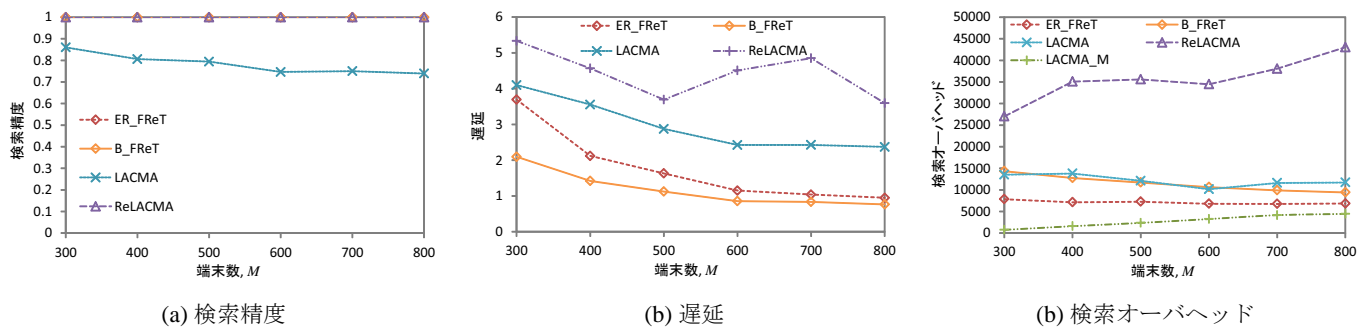


図2 Top-k 検索手法

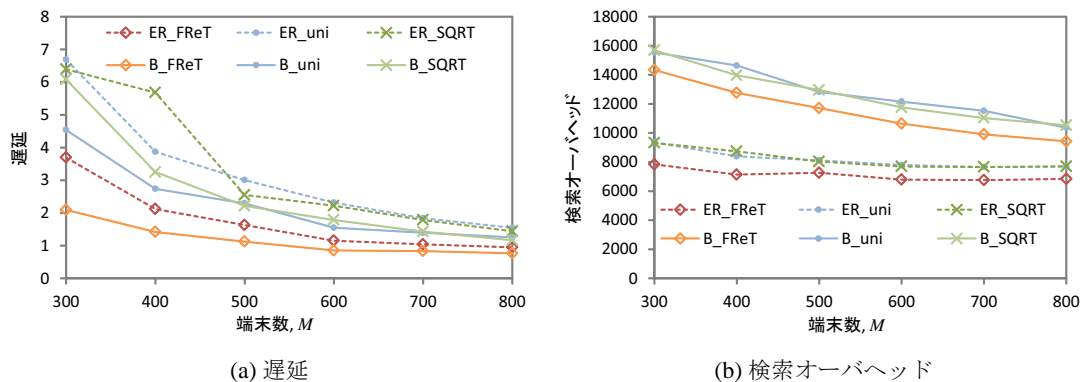


図3 複製配置手法

配置を組み合わせた Top-k 検索手法を提案した。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*, volume 32, pages 177–190. ACM, 2002.

[2] R. Hagihara, M. Shinohara, T. Hara, and S. Nishio. A message processing

method for top-k query for traffic reduction in ad hoc networks. In *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*, pages 11–20. IEEE, 2009.

[3] T. Hara. Effective replica allocation in ad hoc networks for improving data accessibility. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1568–1576. IEEE, 2001.

[4] T. Hara, R. Hagihara, and S. Nishio. Data replication for top-k query processing in mobile wireless sensor networks. In *Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), 2010 IEEE International Conference on*, pages 115–122. IEEE, 2010.

[5] H. Jiang, J. Cheng, D. Wang, C. Wang, and G. Tan. A general framework for efficient continuous multidimensional top-k query processing in sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 23(9):1668–1680, 2012.

[6] S. Lee, S. Wong, K. Lee, and S. Lu. Content management in a mobile ad hoc network: Beyond opportunistic strategy. In *INFOCOM, 2011 Proceedings IEEE*, pages 266–270. IEEE, 2011.

[7] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and replication in unstructured peer-to-peer networks. In *Proceedings of the 16th international conference on Supercomputing*, pages 84–95. ACM, 2002.

[8] S. Ni, Y. Tseng, Y. Chen, and J. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 151–162. ACM, 1999.

[9] N. Padhariya, A. Mondal, V. Goyal, R. Shankar, and S. Madria. Ecotop: an economic model for dynamic processing of top-k queries in mobile-p2p networks. In *Database Systems for Advanced Applications*, pages 251–265. Springer, 2011.

[10] 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), 2010 IEEE 24th International Conference on*, pages 944–949. IEEE, 2010.

[11] Y. Sasaki, T. Hara, and S. Nishio. Two-phase top-k query processing in mobile ad hoc networks. In *Network-Based Information Systems (NBIS), 2011 14th International Conference on*, pages 42–49. IEEE, 2011.

[12] M. Wu, J. Jianliang Xu, X. Xueyan Tang, and W.-C. Wang-Chien Lee. Top-k monitoring in wireless sensor networks. *IEEE Transactions on Knowledge and Data Engineering*, (7):962–976, 2007.

[13] L. Zou and L. Chen. Dominant graph: An efficient indexing structure to answer top-k queries. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, pages 536–545, 2008.