The University of Osaka
Institutional Knowledge Archive

Title	モバイルアドホックネットワークにおけるTop-k検索に関する研究
Author(s)	佐々木, 勇和
Citation	大阪大学, 2014, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/34569
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

モバイルアドホックネットワークにおける
Top-k 検索に関する研究

2014 年 1 月

佐 々 木　勇 和

モバイルアドホックネットワークにおける
Top-k 検索に関する研究

提出先　　大阪大学大学院情報科学研究科

提出年月　2014 年 1 月

佐 々 木　勇 和

関連発表論文

1. 学会論文誌発表論文

1. Sasaki, Y., Hara, T., and Nishio, S., “Top-k Query Processing for Replicated Data in Mobile Peer to Peer Networks,” International Journal of Systems and Software (採録済).

2. Komai, Y., Sasaki, Y., Hara, T., and Nishio, S., “KNN Query Processing Methods in Mobile Ad Hoc Networks,” IEEE Transaction on Mobile Computing (採録済).

3. Goto, K., Sasaki, Y., Hara, T., and Nishio, S., “Data Gathering Using Mobile Agents for Reducing Traffic in Dense Mobile Wireless Sensor Networks,” Mobile Information Systems (採録済).

4. 天方 大地, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “アドホックネットワークにおける Top-k 検索のためのルーティング手法,” 情報処理学会論文誌, Vol. 54, No. 8, pp. 2036–2047 (2013 年 8 月).

5. 佐々木 勇和, 原 隆浩, 西尾 章治郎, “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,” データベース学会論文誌, Vol. 12, No. 1, pp. 115–120 (2013 年 6 月).

6. 後藤 啓介, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “密なモバイルセンサネットワークにおける地理的粒度を考慮したデータ収集のためのエージェント制御方式,” 情報処理学会論文誌, Vol. 53, No. 2, pp. 741–753 (2012 年 2 月).

7. 藤井 俊充, 加治 充, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “モバイルアドホックネットワークにおける ack-carry 方式を用いた情報配布手法,” 情報処理学会論文誌, Vol. 53, No. 1, pp. 286–297 (2012 年 1 月).

8. 佐々木 勇和, 原 隆浩, 西尾 章治郎, “アドホックネットワーク上のスコア分布推定による Top-k 検索手法,” 情報処理学会論文誌, Vol. 52, No. 1, pp. 175–186 (2011 年 1 月).

9. Hara, T., Hagihara, R., Sasaki, Y., Shinohara, M., and Nishio, S., “Efficient Top-K Query Processing in Mobile Ad Hoc Networks,” International Journal of Next-Generation Computing, Vol. 1, No. 2, pp. 193–210 (Nov. 2010).

2. 国際会議発表論文

1. Goto, K., Sasaki, Y., Hara, T., and Nishio, S., “A Mobile Agents Control Scheme for Multiple Sinks in Dense Mobile Wireless Sensor Networks,” Proceedings of International Conference on Mobiquitous, CD-ROM (Dec. 2013).

2. Komai, Y., Sasaki, Y., Hara, T., and Nishio, S., “Processing k Nearest Neighbor Queries for Location-Dependent Data in MANETs,” Proceedings of International Conference on Database and Expert Systems Applications (DEXA), pp. 213–227 (Aug. 2013).

3. Amagata, D., Sasaki, Y., Hara, T., and Nishio, S., “A Robust Routing Method for Top-K Queries in Mobile Ad Hoc Networks,” Proceedings of IEEE International Conference on Mobile Data Management (MDM), pp. 250–255 (June 2013).

4. Sasaki, Y., Lee, W.-C., Hara, T., and Nishio, S., “On Alleviating Beacon Overhead in Routing Protocols for Urban Vanets,” Proceedings of IEEE International Conference on Mobile Data Management (MDM), pp. 66–76 (June 2013).

5. Amagata, D., Sasaki, Y., Hara, T., and Nishio, S., “A Routing Method for Top-K Query Processing in Mobile Ad Hoc Networks,” Proceedings of IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 161–168 (Mar. 2013).

6. Komai, Y., Sasaki, Y., Hara, T., and Nishio, S., “Searching K-Nearest Neighbor Nodes Based on Node Density in Ad Hoc Networks,” Proceedings of International Conference on Advances in Mobile Computing & Multimedia (MoMM), pp. 199–207 (Dec. 2012).

7. Goto, K., Sasaki, Y., Hara, T., and Nishio, S., “Data Gathering Using Mobile Agents in Dense Mobile Wireless Sensor Networks,” Proceedings of International Conference on Advances in Mobile Computing & Multimedia (MoMM), pp. 58–65 (Dec. 2011).

8. Sasaki, Y., Hara, T., and Nishio, S., “Two-Phase Top-K Query Processing in Mobile Ad Hoc Networks,” Proceedings of International Conference on Network-Based Information Systems (NBiS), pp. 42–49 (Sep. 2011).

9. Fujii, T., Kaji, M., Sasaki, Y., Hara, T., and Nishio, S., “A Flooding Control Method with Ack-carry for Location-Based Information Dissemination in Mobile Ad Hoc Networks,” Proceedings of International Symposium on Applications and the Internet (SAINT), pp. 128–135 (July 2011).

10. Komai, Y., Sasaki, Y., Hara, T., and Nishio, S., “A kNN Query Processing Method in Mobile Ad Hoc Networks,” Proceedings of International Conference on Mobile Data Management (MDM), pp. 287–288 (June 2011).

11. Sasaki, Y., Hagihara, R., Hara, T., and Nishio, S., “A Top-K Query Method by Estimating Score Distribution in Mobile Ad Hoc Networks,” International Workshop on Data Management for Wireless and Pervasive Communications (DMWPC), pp. 944–949 (Apr. 2010).

3. 国内会議発表論文（査読付）

1. 佐々木 勇和, Wang-Chien Lee, 原 隆浩, 西尾 章治郎, “位置依存 Top-k 検索のための効率的なインデックス構造の提案,” 情報処理学会マルチメディア通信

と分散処理ワークショップ (DPSWS) 論文集, pp. 220–227 (2013 年 12 月).

2. 天方 大地, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “アドホックネットワークにおける多次元データ Top-k 検索手法,” 情報処理学会マルチメディア通信と分散処理ワークショップ (DPSWS) 論文集, pp. 212–219 (2013 年 12 月).

3. 駒井 友香, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “アドホックネットワークにおけるデータの複製を考慮した k 最近傍データ検索手法,” 情報処理学会マルチメディア，分散，協調とモバイル (DICOMO) シンポジウム論文集, pp. 553–560 (2013 年 7 月).

4. 天方 大地, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “モバイルアドホックネットワークにおけるクラスタを用いた Top-k 検索のためのルーティング手法,” 情報処理学会マルチメディア，分散，協調とモバイル (DICOMO) シンポジウム論文集, pp. 561–568 (2013 年 7 月).

5. 津田 琢士, 駒井 友香, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “アドホックネットワークにおけるデータ差替え攻撃を考慮した Top-k 検索手法および攻撃端末特定手法,” 情報処理学会マルチメディア，分散，協調とモバイル (DICOMO) シンポジウム論文集, pp. 569-576 (2013 年 7 月).

6. 駒井 友香, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “アドホックネットワークにおける k 最近傍データ検索手法,” 情報処理学会マルチメディア通信と分散処理ワークショップ (DPSWS) 論文集, pp. 17–25 (2012 年 10 月).

7. 天方 大地, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “アドホックネットワークにおける Top-k 検索のためのルーティング手法の拡張,” 情報処理学会マルチメディア通信と分散処理ワークショップ (DPSWS) 論文集, pp. 9–16 (2012 年 10 月).

8. 佐々木 勇和, Wang-Chien Lee, 原 隆浩, 西尾 章治郎, “都市部の VANET におけるビーコン制御を用いたルーティングプロトコル,” 情報処理学会マルチメディア通信と分散処理ワークショップ (DPSWS) 論文集, pp. 26–33 (2012 年 10 月).

9. 天方 大地, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “アドホックネットワーク上の効率的な Top-k 検索のためのルーティング手法,” 情報処理学会マルチメディア，分散，協調とモバイル (DICOMO) シンポジウム論文集, pp. 1236–1244 (2012 年 7 月).

10. 松尾 和哉, 後藤 啓介, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “密なモバイルセンサネットワークにおける データ値の地理的分布を考慮したデータ収集方式,” 情報処理学会マルチメディア，分散，協調とモバイル (DICOMO) シンポジウム論文集, pp. 2257–2265 (2012 年 7 月).

11. 藤井 俊充, 加治 充, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “アドホックネットワーク上の周辺情報配布における情報受信による経路変更を考慮したフラッディング制御について,” 情報処理学会マルチメディア，分散，協調とモバイル (DICOMO) シンポジウム論文集, pp. 271–278 (2011 年 7 月).

12. 佐々木 勇和, 原 隆浩, 西尾 章治郎, “アドホックネットワークにおけるデータの複製配置を考慮した Top-k 検索手法,” 情報処理学会マルチメディア通信と分散処理ワークショップ (DPSWS) 論文集, pp. 1–6 (2010 年 10 月).

13. 佐々木 勇和, 原 隆浩, 西尾 章治郎, “アドホックネットワークにおけるデータ数と値を考慮した Top-k 検索,” 情報処理学会マルチメディア，分散，協調とモバイル (DICOMO) シンポジウム論文集, pp. 1121–1128 (2010 年 7 月).

14. 藤井 俊充, 加治 充, 佐々木 勇和, 萩原 亮, 原 隆浩, 西尾 章治郎, “アドホックネットワークにおける周辺情報配布のための ack-carry 方式を用いたフラッディング制御について,” 情報処理学会マルチメディア，分散，協調とモバイル (DICOMO) シンポジウム論文集, pp. 1114–1120 (2010 年 7 月).

15. 佐々木 勇和, 萩原 亮, 原 隆浩, 篠原 昌子, 西尾 章治郎, “アドホックネットワークにおけるヒストグラムを用いた Top-k 検索手法,” 情報処理学会マルチメディア通信と分散処理ワークショップ (DPSWS) 論文集, pp. 13–18 (2009 年 10 月).

4. その他の研究会等発表論文

1. 佐々木 勇和, 原 隆浩, 西尾 章治郎, “モバイルアドホックネットワークにおける Top-k 検索のための位置情報を用いた複製配置手法,” 情報科学技術フォーラム (FIT), pp. 383–386 (2013 年 9 月).

2. 後藤 啓介, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “密なモバイルセンサネットワークにおける複数シンクを考慮したデータ収集方式について,” 電子情報通信学会研究報告 (データ工学研究会 DE2013-7), pp. 19–24 (2013 年 7 月).

3. 佐々木 勇和, 原 隆浩, 西尾 章治郎, “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,” データ工学と情報マネジメントに関するフォーラム (DEIM), CD-ROM (2013 年 3 月).

4. 後藤 啓介, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “密なモバイルセンサネットワークにおける複数シンクのデータ収集条件を考慮したエージェント制御方式,” データ工学と情報マネジメントに関するフォーラム (DEIM), CD-ROM (2012 年 3 月).

5. 駒井 友香, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “アドホックネットワークにおける端末の疎密を考慮した k 最近傍端末の検索手法,” データ工学と情報マネジメントに関するフォーラム (DEIM), CD-ROM (2012 年 3 月).

6. 藤井 俊充, 加治 充, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “疎なアドホックネットワークにおける固定端末を用いたフラッディング制御について,” 情報処理学会研究報告 (マルチメディア通信と分散処理研究会 2012-DPS-150), pp. 1–8 (2012 年 2 月).

7. 後藤 啓介, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “密なモバイルセンサネットワークにおけるエージェントを用いたデータ収集のためのエージェント再配置方式について,” 電子情報通信学会研究報告 (データ工学研究会 DE2011-8), pp. 7–12 (2011 年 8 月).

8. 佐々木 勇和, 原 隆浩, 西尾 章治郎, “アドホックネットワークにおける 2 フェーズ Top-k 検索手法,” データ工学と情報マネジメントに関するフォーラム (DEIM), CD-ROM (2011 年 3 月).

9. 駒井 友香, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “アドホックネットワークにおける k 最近傍端末の検索手法,” データ工学と情報マネジメントに関するフォーラム (DEIM), CD-ROM (2011 年 3 月).

10. 後藤 啓介, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “密なモバイルセンサネットワークにおけるエージェントを用いたデータ収集方式の性能評価,” データ工学と情報マネジメントに関するフォーラム (DEIM), CD-ROM (2011 年 3 月).

11. 後藤 啓介, 佐々木 勇和, 原 隆浩, 西尾 章治郎, “密なモバイルセンサネットワークにおけるエージェントを用いたデータ収集方式,” 平成 22 年度 情報処理学会関西支部 支部大会 講演論文集, CD-ROM (2010 年 9 月).

12. 佐々木 勇和, 萩原 亮, 原 隆浩, 西尾 章治郎, “アドホックネットワークにおけるヒストグラムと確率密度関数を用いた Top-k 検索手法,” 第 2 回データ工学と情報マネジメントに関するフォーラム (DEIM), CD-ROM (2010 年 2 月/3 月).

以上

内容梗概

近年，自律分散的にネットワークを構築できるモバイルアドホックネットワークへの関心が高まっている．モバイルアドホックネットワークは，災害時の救助活動などにおける情報共有やセンサネットワークでの情報収集への応用が期待されている．一方，情報量の増加と多様化に伴い，膨大なデータの中から必要なデータのみを効率的に取得するための検索技術が求められている．その一つとして，指定した検索条件に関連する上位 k 個のデータを検索する Top-k 検索が幅広く研究されている．もともとは，集中管理システムにおいて，計算量を削減するための研究が一般的であったが，近年では，Peer to Peer(P2P) ネットワークやセンサネットワークなどの分散環境における研究が盛んに行われている．Top-k 検索は，モバイルアドホックネットワークにおいても，端末間の情報共有による迅速な救助活動の支援など，様々な状況で有効である．しかし，モバイルアドホックネットワークにおける，Top-k 検索に関する研究はほとんど行われていない．

そこで，本研究では，モバイルアドホックネットワークにおいて，効率的な Top-k 検索処理の実現を目的とする．モバイルアドホックネットワークでは，パケットロスや端末の電力消費を抑制する観点から，通信量の削減は必要不可欠である．そのため，モバイルアドホックネットワークにおける Top-k 検索では，通信量の削減をしつつ，検索精度を維持することが重要な課題となる．通信量はデータの返信数に大きく依存するため，不必要な返信データを削減することが必要である．さらに，端末が自由に移動するという特徴から，ネットワークトポロジが動的に変化するため，無線通信リンクの切断やネットワーク分断に対応することも必要となる．一方，モバイルアドホックネットワークでは，ネットワーク分断に対応するためや，より効率的にデータを取得するために，複製配置が頻繁に研究されている．複製配置は，他の端末が保持するデータを自身の記憶領域に保持するが，一般的に記憶領域に制限があるため，どのデータを複製として配置するかが課題となる．Top-k 検索のデータアクセスの特徴を考慮した複製配置手法，および，効率的に Top-k 検索処理と複製配置を組み合わせた手法が必要となる．

本論文は，5 章から構成され，各章の内容は次の通りである．まず，第 1 章にお

いて，序論として研究の背景と目的について述べる．

第 2 章では，ネットワーク内に存在するデータのスコアが何らかの分布に従うものと想定し，データ分布を推定することにより，返信データ数を削減する Top-k 検索手法を提案する．この手法では，各端末は，自身が保持するデータのスコアをヒストグラム化し，クエリメッセージに添付する．このヒストグラムから，データ分布および，k 位のデータのスコアを推定し，この推定値以上のスコアをもつデータのみを返信する．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し，その有効性について検証する．

第 3 章では，各端末に複製が配置されている環境において，重複した返信データを防ぐ Top-k 検索手法を提案する．この手法では，既存手法を拡張し，複製データを考慮して返信データを絞込む．クエリメッセージにデータの識別子を添付することにより，自身よりクエリ発行端末に近い端末が保持するデータを把握する．また，端末が複製を配置している環境では，一部の端末からのみで上位 k 個のデータを取得できる可能性がある．その場合，全ての端末にメッセージを送信すると無駄な通信が発生してしまう．そのため，各端末は，上位 k 個のデータを既に取得可能かを自律的に判断することにより，クエリメッセージの転送を停止し，データ返信を開始する．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し，その有効性について検証する．

第 4 章では，Top-k 検索のための複製配置およびメッセージ処理手法を提案する．この手法では，まず，ネットワーク全体のデータおよび端末の位置情報を取得し，上位 k 個のデータを把握する．その後，データのアクセス頻度および端末の位置情報より，最適な複製配置頻度を決定し，順位が近いデータをまとめて各端末に配置する．これにより，上位 k 個のデータを取得するために，必要な端末数を減少させる．さらに，検索範囲を徐々に広げながら，上位 k 個のデータを取得するまで繰り返しメッセージを送信する手法も提案する．この方法は，近くの端末から必要なデータを取得することを実現する．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し，その有効性について検証する．

最後に第 5 章では，本論文の成果を要約したのち，今後の研究課題について述べ，本論文のまとめとする．

目 次

第 1 章　序章 **1**
1.1　研究背景 1
1.2　研究動機 3
1.2.1　通信量の削減と検索精度の維持 3
1.2.2　ネットワークトポロジの変化への対応 3
1.2.3　複製配置と Top-k 検索の効率的な組合せ 5
1.3　研究内容 6
1.4　本論文の構成 8

第 2 章　データ分布推定による Top-k 検索手法 **11**
2.1　まえがき 11
2.2　関連研究 12
2.2.1　固定ネットワークにおける Top-k 検索手法 12
2.2.2　非構造型 Peer to Peer ネットワークにおける Top-k 検索手法 . 12
2.2.3　センサネットワークにおける Top-k 検索手法 14
2.2.4　モバイルアドホックネットワークにおける Top-k 検索手法 . 14
2.3　想定環境 15
2.4　提案手法 16
2.4.1　ヒストグラムの作成 16
2.4.2　メッセージの処理手順 16
2.4.3　スコア分布と k 番目のスコアの推定 22
2.4.4　k 番目のスコアの補正 25
2.5　性能評価 26
2.5.1　シミュレーション環境 26

2.5.2 データ要求数 k の影響 27
2.5.3 データサイズの影響 29
2.5.4 Zipf 分布の影響 30
2.5.5 議論 31
2.6 むすび 32

第 3 章 複製配置を考慮した Top-k 検索手法 35
3.1 まえがき 35
3.2 関連研究 36
3.3 想定環境 37
3.4 先行研究 37
3.4.1 Top-k 検索手法 38
3.4.2 複製配置手法 39
3.5 提案手法 40
3.5.1 基準値の設定 41
3.5.2 検索クエリの転送 41
3.5.3 クエリ応答の返信 44
3.6 性能評価 45
3.6.1 シミュレーション環境 45
3.6.2 データ更新が発生しない環境における評価結果 46
3.6.3 データ更新が発生する環境における評価結果 52
3.7 むすび 55

第 4 章 Top-k 検索のための複製配置およびメッセージ処理手法 57
4.1 まえがき 57
4.2 想定環境と問題定義 58
4.2.1 想定環境 58
4.2.2 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 59
4.2.3 Top-k 検索のための複製配置における問題 59
4.2.4 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 . 60

4.3 関連研究 61
4.3.1 非構造型 Peer to Peer ネットワークにおける複製配置方式 . . 61
4.3.2 モバイルアドホックネットワークにおける複製配置 62
4.4 提案手法 64
4.4.1 複製配置手法 64
4.4.2 初期収集および配布手法 66
4.4.3 Top-k 検索のメッセージ処理手法 67
4.5 性能評価 72
4.5.1 シミュレーション環境 72
4.5.2 比較手法 73
4.5.3 評価結果 75
4.6 むすび 83

第 5 章 結論 85
5.1 本論文のまとめ 85
5.2 検討課題 87

謝辞 89

第1章　序章

1.1　研究背景

近年，IEEE802.11 [25] や Bluetooth [64]，Zigbee [74] などをはじめとする無線通信技術の発展と，PDA(Personal Digital Assistant) や携帯電話などの計算機の小型化や高性能化に伴い，ルータ機能をもつ移動端末のみで一時的な無線ネットワークを形成するモバイルアドホックネットワークへの関心が高まっている [27,36,42,46,51,54]. モバイルアドホックネットワークでは，端末が自身の無線通信範囲内に存在する端末と通信する場合，基地局を介さずに通信できる．一方，自身の無線通信範囲内に存在しない端末と通信する場合，これらの端末の間に存在する端末がパケットを中継してマルチホップ通信を行う．このように，アドホックネットワークは既存の通信インフラを必要とせずに，端末同士で自律分散的にネットワークを構築できるため，災害時の救助活動および山岳での遺跡発掘調査などにおける情報共有やセンサネットワークでの情報収集への応用が期待されている．

一方，情報量の増加と多様化に伴い，膨大なデータの中から必要なデータのみを効率的に取得する検索技術が求められている．その一つとして，指定した検索条件に関連する上位 k 個のデータを検索する Top-k 検索が幅広く研究されている．Top-k 検索では，端末が何らかの値（スコア）によって順序付けられたデータの上位 k 個以内のものを検索する [4, 8, 11, 13, 24, 26, 33, 62, 70]．もともとは，集中管理システムにおいて，計算量を削減するための研究が一般的であったが，近年では，P2P ネットワークやセンサネットワークなどの分散環境において研究が盛んに行われている [2, 35, 40, 65]．モバイルアドホックネットワークに Top-k 検索を適応することにより，端末間の情報共有による迅速な救助活動の支援などを実現することができる．例えば，災害地において，各端末（ユーザ）に限られた資源（救急車

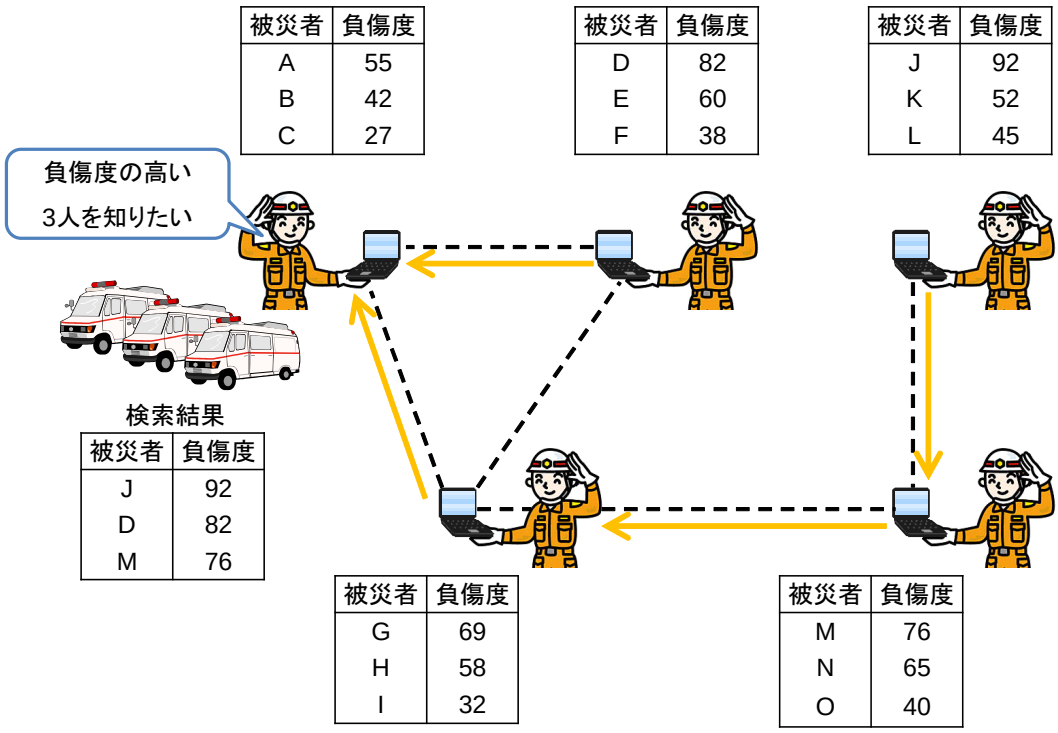


図 1.1: アドホックネットワークにおける Top-k 検索の例

や薬，人員など）を割り当てる場合，資源の必要性が高い端末を探すことや，関連性の高い情報のみ（負傷度が大きい被害者の情報）を収集することなど，様々な状況で Top-k 検索は有効に働くと考える．図 1.1 を用いて，具体例を説明する．緊急災害時の救助活動において，救助隊員が被災者の負傷度の情報を管理し，救助隊員の持つ端末同士でネットワークを構築している様子を示している．このような環境において，左上の救助隊員が限られた数の救急車で，負傷度の高い被災者を優先的に搬送する場合，Top-k 検索を用いることにより，左上の救助隊員が負傷度の高い数人（この例では 3 人）の被災者を検索することができる．

モバイルアドホックネットワークにおいて，Top-k 検索が必要となる機会が多いと考えられるにも関わらず，ほとんど研究されていないのが現状である．そのため，本論文では，モバイルアドホックネットワークにおける，効率的な Top-k 検索手法を提案することを目的とする．

1.2　研究動機

1.2.1　通信量の削減と検索精度の維持

モバイルアドホックネットワークでは，端末は無線通信により，メッセージを送信するため，通信量（トラヒック）が増加すると，パケットロスが発生してしまう．モバイルアドホックネットワークにおいて Top-k 検索を実行する場合，Top-k 検索を実行する端末（クエリ発行端末とよぶ）は，まず検索クエリメッセージをネットワーク内に存在する端末に送信し，クエリメッセージを受信した端末が自身が保持するデータを返信することにより，データを取得する．しかし，各端末が保持する全てのデータを返信すると，ネットワーク内のトラヒックが増加し，パケットロスが発生する原因となってしまう．パケットロスが発生した場合，上位 k 個に入る必要なデータをクエリ発行端末が取得できず，検索精度が低下してしまう．トラヒックを減少させるために，返信データ数を削減することが重要である．しかし，不用意に返信データ数を削減すると，必要なデータも返信しない可能性がある．そのため，上位 k 個に入らない無駄なデータの返信によるトラヒックを削減しつつ，上位 k 個のデータに入る必要なデータを返信することが求められる．各端末は，他の端末が保持するデータに関する情報をもたないため，検索クエリにどのような情報を添付して，メッセージを送信するかが重要となる．

図 1.1 では，クエリ発行端末は，負傷度が大きい上位 3 人のデータを要求している．この例では，データ J，D，および M のみが必要なデータとなるため，その他のデータの返信は，無駄なトラヒックとなる．

1.2.2　ネットワークトポロジの変化への対応

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が変化すると，隣接する端末とのリンクが切断または他の端末と新たにリンク接続する．データの返信をする場合に，クエリメッセージ受信時には接続していたリンクの切断が発生すると，クエリメッセージ送信端末にデータを転送できなくなる可能性がある．そのため，

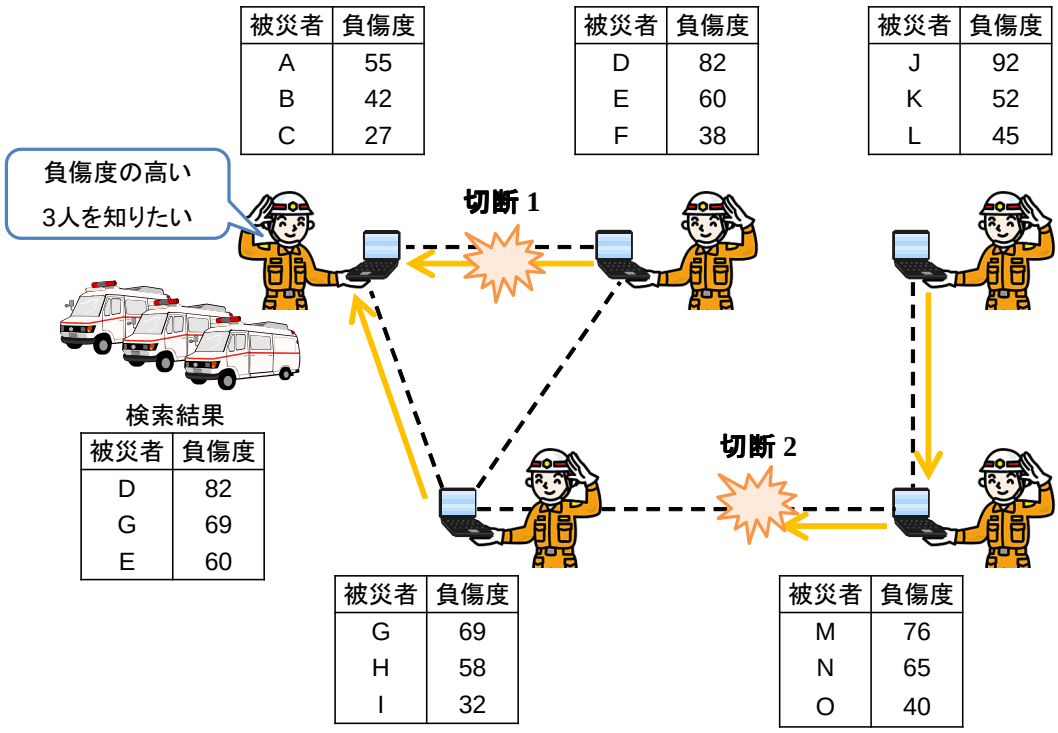


図 1.2: ネットワーク分断の例

できる限りリンク切断が発生する前に検索処理を終わらせること，および，それでもリンク切断が発生した場合は新たな経路を探す処理が必要となる．

さらに，ネットワークが複数のグループに分断してしまうことも考えられる．この場合，他のグループに所属する端末へのアクセスができず，必要なデータを取得できない可能性がある．これは，後述する複製配置によって解決する．

図 1.1 の例において，リンク切断が発生した場合の影響を図 1.2 を用いて説明する．リンク切断 1 が発生した場合では，中央上の端末がデータを左上のクエリ発行端末に直接送信することができないため，左下の端末を経由して，送信する必要がある．一方，リンク切断 2 が発生した場合，ネットワークが分断してしまうため，クエリ発行端末は右側のグループに所属する端末のデータを取得することができず，検索精度が低下する．

1.2.3　複製配置と Top-k 検索の効率的な組合せ

他の端末が保持するデータを自身の記憶領域に保持する複製配置により，より効率的な Top-k 検索を実現することができる．例えば，頻繁にアクセスされるデータを多くの端末が保持することにより，自身が再度クエリを発行する場合や，近くの他の端末がクエリを発行する場合に，効率的に必要なデータを取得できる．モバイルアドホックネットワークでは，ネットワーク分断が発生するため，全ての端末にアクセスしなくても，正確な検索結果の取得を可能にする複製配置は極めて有効である．しかし，複数の端末が同じデータを保持すると，同じデータが複数返信されることが考えられる．これは，トラヒック増加の原因となり，パケットロスを引き起こす可能性があるため，複製配置を考慮したメッセージ処理が必要となる．

また，一方で，各端末がどのデータを複製として保持するかが課題となる．Top-k 検索では，上位のデータが頻繁にアクセスされるため，多くの端末が上位のデータを保持する可能性が高い．しかし，これは複製データの多様性の低下の原因となるため，Top-k 検索のための複製配置手法が必要となる．

例えば，図 1.2 において，上位 3 個に入るデータ J および M を左側のグループの端末が複製として保持することにより，リンク切断 2 によりネットワークが分断しても，検索結果の精度を保つことができる．しかし，ネットワーク分断が発生していない場合では，J および M を保持する複数の端末がそれらのデータを重複して返信し，無駄なトラヒックが発生する．

ここで，各端末がデータを複製として配置する場合，データの更新が問題となる．データの更新が不定期かつ頻繁に発生する場合，複製の更新が頻繁に必要となるため，複製の管理に大きなトラヒックが発生してしまう．このような場合，複製配置は効果的ではない．一方，データの更新がない，もしくは定期的かつ更新頻度が少ない場合，長期間において複製が有効であるため，複製配置が効果的である．複製配置が効果的かそうでないかは，想定する環境によって異なるため，本研究では，複製配置を想定する環境と想定しない環境の両方について考える．

1.3　研究内容

本論文では，トラヒックを削減しつつ，検索精度を維持することを目的として，モバイルアドホックネットワークにおける Top-k 検索処理手法について議論する．具体的には，それぞれの目的を実現する三つの研究課題に取り組む．

- データ分布推定による Top-k 検索手法

  各端末は，他の端末がもつデータについての情報をもたないため，上位 k 個に入るデータを把握できず，無駄な返信データが増加してしまう．ここで，データのスコアは何らかの分布（正規分布や Zipf 分布）に従っている可能性がある．例えば，健康状態がよくない被災者を把握するために，血圧が高い上位の被災者の情報を取得するといった場合，血圧は正規分布に従っているということを利用可能である．そこで，データが何らかの分布（正規分布や Zipf 分布）に従うものと想定し，一部のデータのスコアの情報からスコア分布を推定し，返信データを削減する Top-k 検索手法を提案する．この手法では，データのスコアからヒストグラムを構築し，クエリメッセージに添付する．このヒストグラムから k 位のデータのスコアを推定し，そのスコア以上のスコアをもつデータのみを返信する．

- 複製配置を考慮した Top-k 検索手法

  文献 [32] において，検索実行時に，各端末が返信データを複製として，配置する手法が提案されている．この手法を含めて，一般的に，複製配置では，頻繁にアクセスされるデータの複製を配置する．Top-k 検索では，スコアが大きいデータが頻繁にアクセスされるため，各端末がスコアが大きいデータを複製として保持することになる．文献 [21] では，Top-k 検索において，データのスコアを考慮し，配置する複製を決定する手法が提案されている．しかし，この手法では，多くの端末がスコアが大きいデータを配置するため，同じデータが重複して複数返信されてしまう可能性が高い．

  そこで，重複する返信データを防ぐ Top-k 検索手法を提案する．この手法では，クエリメッセージに自身が保持するデータの識別子を添付することによ

り，自身よりクエリ発行端末に近い端末が保持するデータを把握する．さらに，全ての端末にメッセージを送信しなくても，一部の端末からのみで必要なデータを取得できる可能性がある．そのため，提案手法では，これまでにクエリメッセージを受信した端末のみから既に上位 k 個のデータを取得できるかを各端末が自律的に判断する．取得できると判断した場合，クエリメッセージの転送を停止し，データの返信をすぐに開始する．これにより，トラヒック，および検索時間の削減が可能である．

- Top-k 検索のための複製配置およびメッセージ処理手法

  モバイルアドホックネットワークにおける Top-k 検索では，近くの端末から上位 k 個のデータを取得することが理想的である．複製配置では，頻繁にアクセスされるデータを多くの端末が保持した方が，探索しなければならない端末数を減少させることができるのが一般的である．Top-k 検索では，頻繁にアクセスされるデータは，上位のスコアを保持するデータとなる．このとき，指定したホップ数内（つまり，限られた狭い範囲）に存在する端末から正確なデータを取得することを目標とするが，アクセスできる端末数は限られている．そのため，上位のデータを保持しすぎた場合，配置している複製に多様性がなくなり，近くの端末から全ての必要な上位 k 個のデータを取得できない可能性がある．そこで，近くの端末から全ての必要なデータを取得できるように，複製の配置率を決定する必要がある．しかし，端末が移動すると，アクセス可能な端末が変化するため，周囲の状況に合わせて，検索の範囲を動的に決定することも必要とされる．

  本研究では，複製配置手法とメッセージ処理手法を同時に考えることにより，近くの端末からデータを取得することと周囲の状況に合わせて検索範囲を動的に決定することを実現する．まず，複製配置においては，データの多様性を維持しつつ，データのアクセス頻度を考慮して複製の配置頻度を決定する．さらに，Top-k 検索では，順位が近いデータは，同時にアクセスされる可能性が高いという特徴がある．このとき，ある端末が 1 位と 2 位のデータをまとめて保持していれば，この端末にのみアクセスできれば上位 2 位のデータ

を取得することができる．つまり，近い順位のデータをまとめて保持すれば，アクセスしなければいけない端末数を減少させることができる．そこで，各端末が保持するデータの複製の組合せ，および複製データの多様性を保つ複製の配置頻度を決定する複製配置手法を提案する．また，近くの端末から必要なデータを取得するまで，検索範囲を広げながら，繰り返しメッセージを送信するメッセージ処理手法も提案する．

それぞれ三つの研究課題は，Top-k 検索を実行する上で異なる環境を想定している．つまり，各端末が複製を保持しているのか，保持していない（複製を配置できない）のか，さらに，複製を保持している場合においても，各端末が保持する複製を Top-k 検索のみのために制御できるのか（複製配置と Top-k 検索が連携しているか），否か，という環境の違いである．それぞれの研究において，データ分布推定による Top-k 検索手法（2 章）は複製を保持していない環境，複製配置を考慮した Top-k 検索手法（3 章）は何らかの複製配置手法により各端末が複製を配置を保持しているがそれを制御できない環境，および，Top-k 検索のための複製配置およびメッセージ処理手法（4 章）は複製配置と Top-k 検索が連携している環境を想定している．3 章の想定環境は，Top-k 検索だけではなくほかのアプリケーションでもデータを利用し，複製を作成する環境などに該当する．なお，本論文では，2 章の Top-k 検索手法を 3 章の環境にも適用可能であるが，簡単化のため，より単純な検索手法を想定している．また，4 章の環境では，複製配置を制御できるという性質上，2 章および 3 章とは根本的に異なる Top-k 検索手法を提案する．

1.4　本論文の構成

本論文は，5 章から構成され，本章以降の内容は次の通りである．

第 2 章では，ネットワーク内に存在するデータのスコアが何らかの分布に従うものと想定した上で，データ分布を推定し，返信データ数を削減する Top-k 検索手法を提案する．この手法では，各端末は，自身が保持するデータのスコアをヒストグラム化し，クエリメッセージに添付する．このヒストグラムから，データ分布，および k 位のデータのスコアを推定し，この推定値以上のスコアをもつデータの

みを返信する．データ分布および k 位のデータのスコアの推定方法として，二つの方法を提案する．一つ目の方法は，ヒストグラム内のデータ数を用いた方法であり，ヒストグラムの各階級のデータ数を基にネットワーク内のデータ数を考慮して，データ分布を推定する．各階級内のデータ数を大きいスコアの階級から数え，k 個超えた階級の下限値を k 番目のスコアと推定する．二つ目の方法は，確率密度関数を用いた方法であり，ヒストグラムから分布の形状を決定する要素を計算し，データ分布を推定する．その後，推定したデータ分布に対応する確率密度関数を用いて，k 番目のスコアを推定する．また，推定した k 番目のスコアが実際の k 番目のスコアより，大きくなると必要なデータが返信されないため，推定値を効果的に小さくさせるセーフティマージンにより，返信漏れを防ぐ．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し，その有効性について検証する．

第 3 章では，データの複製が配置されている環境において，重複した返信データを防ぐ Top-k 検索手法を提案する．この手法では，文献 [18] で提案された Top-k 検索手法を拡張し，複製データを考慮して返信データを絞込む．クエリメッセージにデータの識別子を添付することにより，自身よりクエリ発行端末に近い端末が保持するデータを把握する．さらに，各端末は，上位 k 個のデータが既に取得可能かを自律的に判断し，クエリメッセージの転送を停止し，データ返信を開始する．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し，その有効性について検証する．

第 4 章では，Top-k 検索のための複製配置およびメッセージ処理手法を提案する．この手法では，初期処理として，ネットワーク全体のデータおよび端末の位置情報を取得し，上位 k 個のデータを把握する．その後，データのアクセス頻度および端末の位置情報より，最適な複製配置頻度を決定し，順位が近いデータをまとめて各端末に配置する．これにより，上位 k 個のデータを取得するために，必要な端末数を減少させる．さらに，近くの端末から必要なデータを取得可能なメッセージ処理手法も提案する．このメッセージ処理手法では，検索範囲を徐々に広げながら，上位 k 個のデータを取得するまで繰り返しメッセージを送信する．検索範囲の増加方法として，Expanding ring 法と Bundling 法を提案する．Expanding

ring 法では，返信データ数を削減するために，1 ホップずつ徐々に検索範囲を広げる．一方，Bundling 法では，隣接端末数を考慮して，一度に検索範囲を広げることにより，遅延を抑制する．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し，その有効性について検証する．

最後に第 5 章では，本論文の成果を要約したのち，今後の研究課題について述べ，本論文のまとめとする．

第 2 章は文献 [45, 48, 50, 53] で公表した結果に，第 3 章は文献 [47, 55] で公表した結果に，第 4 章は文献 [49, 52] で公表した結果に基づき論述する．

第2章　データ分布推定によるTop-k検索手法

2.1　まえがき

モバイルアドホックネットワークにおいて，Top-k 検索を実現する単純な方法として，メッセージを受信した端末がそのメッセージを隣接端末にブロードキャストする処理を繰り返すフラッディングという方法により，まずクエリ発行端末が検索クエリをネットワーク全体の端末に送信し，この検索クエリを受信した端末が自身のもつデータの中からスコアの高いものを固定数返信する方法が考えられる．各端末の返信するデータの数が多い場合，検索クエリを発行した端末は，ネットワーク全体の上位 k 個のデータを取得できる可能性が高い．しかし，検索結果に入らないデータまで返信されるため，不要なトラヒックが発生し，パケットロスが頻繁に発生する可能性がある．一方，各端末の返信データ数が少ない場合，検索結果に入らないデータが返信される可能性は低くなるが，検索結果に入るデータが返信されず，検索結果の精度が低下する．また，モバイルアドホックネットワークでは，端末の移動によりネットワークトポロジが動的に変化する．Top-k 検索の実行中に端末間のリンクが切断された場合，クエリ発行端末が検索結果に含まれるデータを取得できず，検索結果の精度が低下する可能性がある．

そこで本章では，モバイルアドホックネットワークにおいて，トラヒックの削減と検索結果の精度の低下の抑止を目的とする Top-k 検索手法を提案する．提案手法では，各端末が自身のもつデータのスコアからヒストグラムを作成し，ネットワーク全体の k 番目のスコアを推定する．このとき，k 番目のスコアを正確に推定するために，各端末は検索クエリやクエリ応答にヒストグラムを添付して送信する．次に，各端末は推定した k 番目のスコア以上のスコアをもつデータのみを

返信することで，できる限り検索結果に入るデータのみを返信する．また，返信先の端末とのリンクが切断した端末は，他の隣接端末にクエリ応答を返信することで，検索結果の精度の低下を抑止する．

以下では，2.2 節で関連研究を紹介する．2.3 節で本論文の想定環境を示し，2.4 節で提案手法について述べる．2.5 節でシミュレーション実験の結果を示す．最後に 2.6 節で本章のまとめと今後の課題について述べる．

2.2　関連研究

2.2.1　固定ネットワークにおける Top-k 検索手法

固定ネットワークを想定した分散データベースの分野では，これまでに Top-k 検索に関する研究が数多く行われている [3, 12, 13, 16, 37]．Fagin は，文献 [12] において，閾値アルゴリズム [13, 16, 37] を拡張し，スコアの組合せによる順序付けによって画像検索を行う手法を提案している．これらの研究では，スコアをさまざまな要素の組合せにより求めることを想定しており，データと検索クエリによって一意に算出することを想定している本研究と異なる．また，Babcock らは，文献 [3] において，ネットワークの統計情報などのように連続的にデータが発生する環境において，そのデータの値が上位 k 個以内のものを求める手法を提案している．この研究は，検索クエリを用いない点で本研究と異なる．

2.2.2　非構造型 Peer to Peer ネットワークにおける Top-k 検索手法

非構造型検索トポロジを用いた P2P ネットワークの分野において，Top-k 検索に関する研究が盛んに行われている [1, 5, 28, 29, 34, 63]．非構造型検索トポロジでは，検索クエリは，クエリ発行端末から各端末に，フラッディングを用いて転送される．Kalnis らは，文献 [28, 29] において，端末がクエリ応答の要求数 k を含めた検索クエリをフラッディングし，これを受信した各端末が自身のデータを順位付けした後，上位のスコアをもつ k 個のデータを返信するシステムを提案している．この研究では，検索結果に入らない不要なデータの返信によるトラヒックが多く，

通信帯域が限られているモバイルアドホックネットワークで用いた場合，パケット衝突による転送遅延の増大やスループットの低下が発生してしまうため，有効ではない．

また，Balke らは，文献 [5] において，1 回の検索クエリの送信で 1 個のデータを取得し，この処理を k 個の検索結果が取得できるまで繰り返す Top-k 検索手法を提案している．具体的には，最初の検索クエリによって，各端末から最上位（1 番目）のデータのスコアを返信してもらい，その中で最もスコアの高いデータを取得する．これにより，ネットワーク内で 1 番目のスコアをもつデータを取得したことになる．次に，1 番目の検索結果をもつ端末に対して，その端末の 2 番目のスコアをもつデータを要求し，他の端末がもつ 1 番目のスコアと比較することで，ネットワーク内の 2 番目のスコアをもつデータを取得する．この処理を k 番目のスコアをもつデータを取得するまで繰り返す．この手法では，返信するデータの個数は少ないものの，検索結果の取得に時間がかかるため，ネットワークトポロジが動的に変化するモバイルアドホックネットワークでは有効ではない．さらに，Akbarinia らは，文献 [1] において，検索クエリとクエリ応答を適切な隣接端末に送信することでトラヒックを削減する手法を提案している．また，この手法では，端末がクエリ応答を返信するときに，隣接端末が存在しない場合，検索クエリを発行した端末にクエリ応答を直接返信する．この研究では，各端末が自身の隣接端末を把握している環境を想定している点，任意の端末間で直接通信が可能なネットワークを想定している点で本研究と異なる．

松波らは，文献 [34, 63] において，P2P コンテンツ検索システムにおける Top-k 検索手法を提案している．文献 [34] では，まず，検索クエリの受信時に各端末が上位 k 個以内のスコアをもつデータのみを返信する Fixed-k query 手法を提案している．この手法では，各端末が他端末からのクエリ応答を中継する際，自身およびこれまでに中継したクエリ応答に含まれる返信データを考慮して，上位 k 個以内に入る可能性のあるデータのみを返信し，それ以外は破棄する．これにより，クエリ応答に伴うトラヒックをある程度は削減できるが，基本的に各端末が k 個のデータを返信するため，その効果は十分ではない．また，Fixed-k query 手法を改良し，すべての隣接端末からのクエリ応答を受信するまで待機し，その段階で上位 k 個

以内に入るスコアをもつデータのみを返信することでクエリ応答によるトラヒックを削減する Delayed Fixed-k query(DF) 手法を提案している．さらに，文献 [63] では，文献 [34] の方式を拡張し，検索クエリの転送時に，自身までの転送経路上の端末が保持する上位 k 個以内のデータのスコアを転送する Delayed Fixed-k and Score Propagating query(DFSP) 手法を提案している．これらの手法においても，経路毎に k 個のデータが返信されるため，トラヒックの削減効果はそれほど大きくない．

2.2.3　センサネットワークにおける Top-k 検索手法

センサネットワークの分野では，Top-k 検索によって消費電力を削減させ，ネットワーク全体の生存時間を長くすることを目的とした研究がいくつか行われている [59,60,68,69,72]．Wu らは，文献 [68,69] において，フィルタリングされたセンシング情報に対して Top-k 検索を行うことで，センシング対象領域の異常などを検出する手法を提案している．また，Yazti らは，文献 [72] において，センシングデータの特徴を考慮して，送信すべきデータの上限値および下限値を求めることで，Top-k 検索を行った結果，上位 k 個以内に入る可能性のあるデータのみを転送する手法を提案している．この研究は，センシングデータの値に対して Top-k 検索を行っており，検索条件によって端末が保持する（ある程度大きな）データがスコア付けされる本研究とは，想定するデータのサイズが異なる．Silberstein らは，文献 [59,60] において，過去のセンシング情報から，予測できる情報の送信を抑制することで，消費電力を削減する Top-k 検索手法を提案している．これらの研究は，過去の情報から現在の情報を予測できる環境を想定している点で本研究と異なる．

2.2.4　モバイルアドホックネットワークにおける Top-k 検索手法

これまでに筆者の研究グループは，文献 [17,18] において，モバイルアドホックネットワークにおけるトラヒックの削減と検索結果の精度の維持を実現する Top-k 検索手法を提案した. この手法では，各端末が検索結果に含まれるデータのスコアを推測し，その一部を基準値として用いることで，検索クエリの転送時およびク

エリ応答の返信時に検索結果の候補となるデータを絞り込む．この手法では，上位 k 個に入らないことが確実なデータの返信を防ぐことはできるが，検索結果の絞り込みの効果はそれほど大きくない．一方，本章の提案手法は，中継端末のもつデータのスコア分布からネットワーク全体の k 番目のスコアを推定するため，絞り込みの効果が大きい．しかし，推定の精度が低い場合，上位 k 個のデータが返信されない可能性がある．

2.3 想定環境

本章では，モバイル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．Top-k 検索を行う端末は，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．

ネットワーク内には n 個のデータが存在し，各々が特定の端末に保持されている．簡単化のため，全てのデータのサイズは等しいものとする．データのスコアは，検索条件とデータの属性値から決定し，何らかのスコアリング関数を用いて算出される．また，スコアは特定のスコア分布（例えば正規分布）に従っているものとする．データは，周期的に更新され，これに伴い，データのスコアも更新される．

ネットワーク内には，m 個の端末（識別子：$M_1, M_2, ..., M_m$）が存在し，各々が自由に移動する．各端末は，ネットワーク内のデータ数 n，ネットワーク内のデータのスコアが従う分布の種類，およびスコアの下限および上限を把握しているものとする．また，各端末のもつデータ数はネットワーク全体のデータ数（n）に比べて少ないため，各端末のもつデータのスコア分布は，ネットワーク全体のデータのスコア分布に従っているとは限らない．各端末は，データの複製を作成しないものとする．

2.4 提案手法

本章では，トラヒックを削減しつつ，検索結果の精度の低下を抑止する Top-k 検索手法を提案する．以下では，まず各端末が作成するヒストグラムについて説明する．次に，検索クエリの転送時，クエリ応答の返信時，およびリンク切断の検出時におけるメッセージの処理手順について説明する．最後に，クエリ応答を返信する端末が，ヒストグラムを用いてスコア分布と k 番目のスコアを推定する方法，および推定した k 番目のスコアを補正する方法について説明する．

2.4.1 ヒストグラムの作成

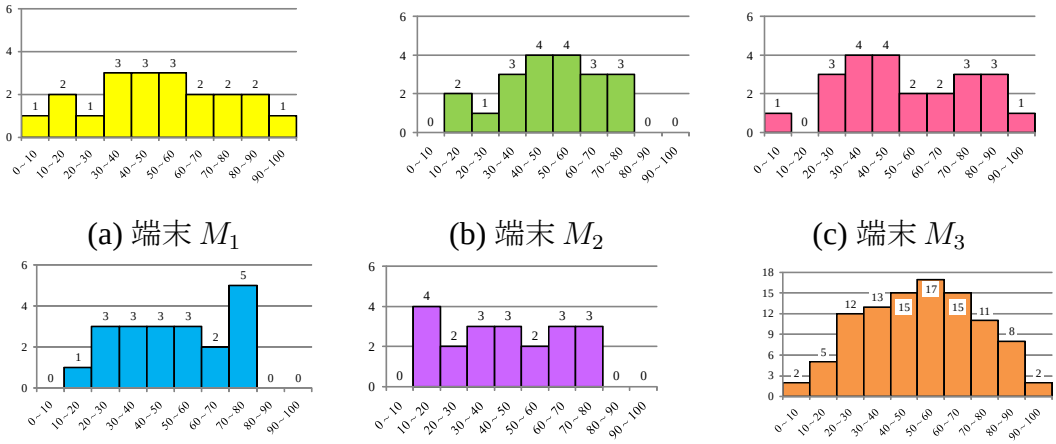
各端末は，自身のもつデータのスコアを検索条件に基づいて算出し，ヒストグラムを作成する．ヒストグラムは，階級と呼ばれる適当な大きさの区間ごとに，値がその区間に含まれるスコアの数を数えたものであり，スコアの分布状況を表すのに用いられる．提案手法では，端末 M_i のもつデータの中で，そのスコアが階級 c_j $(1 \leq j \leq C)$ に含まれる個数を数えたものを M_i のヒストグラム H_i とする．ただし，その範囲は $[MIN + \frac{(j-1)(MAX-MIN)}{C}, MIN + \frac{j(MAX-MIN)}{C})$ となる．

例えば，ネットワーク内のデータのスコアが，平均 50，分散 20 の正規分布に従い，スコアのとりうる値の範囲が [0,100)，階級数 $C = 10$ の場合に，ネットワーク内の端末 M_1 から M_5 が作成したヒストグラム H_i $(1 \leq i \leq 5)$ およびネットワーク全体のヒストグラムを図 2.1 に示す．

2.4.2 メッセージの処理手順

検索クエリの転送時

提案手法では，ヒストグラムに含まれるスコア数が多いほど，そのスコア分布はネットワーク内のデータのスコア分布に近くなるため，端末はスコア分布と k 番目のスコアをより正確に推定できる．そこで，各端末は，ヒストグラムを検索クエリメッセージに添付して送信し，検索クエリメッセージを中継する端末がヒ



(a) 端末 M_1 (b) 端末 M_2 (c) 端末 M_3

(d) 端末 M_4 (e) 端末 M_5 (f) 全端末

図 2.1: 各端末の作成したヒストグラムとネットワーク全体のヒストグラム

ストグラムを更新しながら転送する．以下では，検索クエリを発行した端末 M_p と検索クエリメッセージを受信した端末 M_q の動作について説明する．

1. 端末 M_p は検索条件，および検索での要求データ数 k を指定する．また，検索条件から自身のもつデータのスコアを算出し，2.4.1 項の方法に従って，自身のヒストグラム H_p を作成する．

2. 端末 M_p は，全ての隣接する端末に検索クエリメッセージを送信する．このメッセージには，検索クエリ発行端末 M_p の識別子，検索クエリの識別子，要求データ数 k，検索条件，経路端末リスト，およびクエリヒストグラム QH が含まれる．ここで，経路端末リストは，クエリ発行端末から自身までの経路上に存在する端末の識別子が含まれ，クエリヒストグラムは，これらの経路上の端末のヒストグラムを統合したものが含まれる．この場合，経路端末リストには M_p のみが含まれ，クエリヒストグラム QH は M_p のヒストグラム H_p となる．

3. 検索クエリを受信した端末 M_q は，それが初めて受信したものであれば，経路端末リストの末尾に格納されている端末を自身の親とし，経路端末リストを記録する．手順 4. へ進む．

受信した検索クエリメッセージが，既に受信したものであれば，手順 5. へ進む．

4. 端末 M_q は，手順 1. と同様に検索条件から自身のヒストグラム H_q を作成し，検索クエリメッセージに含まれるクエリヒストグラム QH に H_q を統合する．また，自身の識別子 M_q を経路端末リストに追加する．M_q は，自身の隣接端末に検索クエリメッセージを送信し，手順 3. に戻る．

5. 検索クエリメッセージを再受信した端末 M_q は，経路端末リストの末尾の端末を親でない隣接端末とし，経路端末リストに含まれる端末数から 1 減らした数をクエリ発行端末からその端末までのホップ数として記録する．また，経路端末リストの末尾から 2 番目の端末が自身の場合，経路端末リストの末尾の端末を自身の子とする．

検索クエリの転送では，各端末がクエリ発行端末から自身までの経路上に存在する端末のヒストグラムを統合したクエリヒストグラムを検索クエリメッセージに添付することで，ヒストグラムに含まれるスコア数を増加させる．また，各端末は経路端末リストにより，検索クエリ発行端末を根とする木構造における自身の親と子，クエリ発行端末から自身までの経路，および親以外の隣接端末のクエリ発行端末からのホップ数を把握できる．

図 2.2 を用いて，M_1 が Top-k 検索を行った場合の検索クエリの転送例を説明する．吹出しは各端末が検索クエリに添付したクエリヒストグラムを示し，破線の矢印は端末間の親子関係を示す．検索クエリには，クエリ発行端末からの経路上の端末のヒストグラムを統合したものが添付されているが，統合の状態をわかりやすく示すため，端末 M_1，M_2，M_3，M_4，および M_5 のヒストグラムに相当する部分をそれぞれ黄色，緑色，桃色，青色，および紫色で表す．なお，各端末のヒストグラムは図 2.1 で表されるものとする．

クエリ応答の返信時

上記で述べたように，提案手法では，ヒストグラムに含まれるスコア数が多いほど，スコア分布や k 番目のスコアをより正確に推定できるため，各端末はヒス

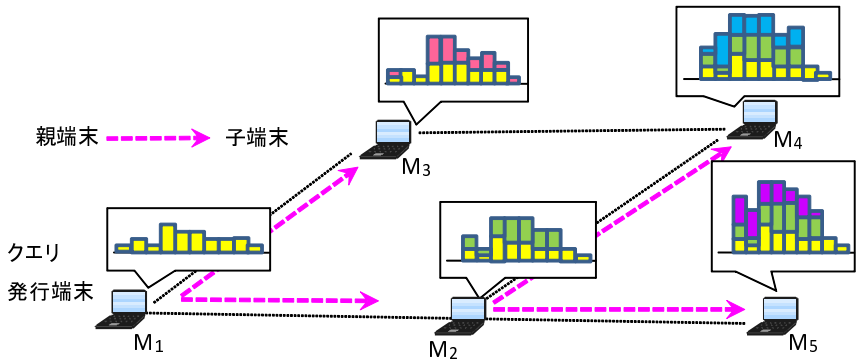


図 2.2: 検索クエリの転送例

トグラムをクエリ応答メッセージにも添付して送信する．また，不要なデータの返信を抑止するため，クエリ応答を中継する端末は，受信した情報から返信の必要がないデータを検出した場合，そのデータを返信しない．以下では，端末 M_p が検索クエリを転送した後，各端末がクエリ応答を返信する動作について説明する．

1. 自身の子がいない端末 M_r は，自身の送信した検索クエリメッセージに添付したクエリヒストグラムを用いて，スコア分布と k 番目のスコアの推定（2.4.3 項），および k 番目のスコアの補正（2.4.4 項）を行う．その後 M_r は，クエリ応答メッセージを自身の親に送信する．このメッセージには，検索クエリ発行端末 M_p の識別子，検索クエリの識別子，返信リスト，および応答ヒストグラム RH が含まれる．ここで，返信リストは自身のもつデータの中で，推定した k 番目のスコア以上のスコアとそのスコアをもつデータを示す．また，応答ヒストグラムは，自身および自身の子孫である端末のヒストグラムを統合したものが含まれ，この場合 M_r のヒストグラム H_r となる．

2. 全ての子からクエリ応答メッセージを受信するか，検索クエリメッセージを送信してから一定時間経過した端末 M_s は，受信した全てのクエリ応答メッセージに含まれる応答ヒストグラム，および検索クエリメッセージに添付したクエリヒストグラムを統合する．このヒストグラムは，クエリ発行端末から M_s までの経路上の端末，および端末 M_s の全ての子孫端末のヒストグラムを統合したものである．端末 M_s は，この統合したヒストグラムを用いて，

スコア分布と k 番目の推定（2.4.3 項），および k 番目のスコアの補正（2.4.4 項）を行う．手順 3. へ進む．

3. 端末 M_s は，クエリ応答メッセージを作成し，自身の親に送信する．ここで，返信リストは，受信した全てのクエリ応答メッセージに含まれる返信リストおよび自身のもつデータの中で，推定した k 番目のスコア以上のスコアとそのスコアをもつデータが含まれる．ただし，該当するデータ数が要求データ数 k より大きい場合，上位 k 個のスコアとそのスコアをもつデータのみが含まれる．また，応答ヒストグラムは，自身のヒストグラム H_s と受信した全てのクエリ応答メッセージに含まれる応答ヒストグラムを統合したものが含まれる．

4. クエリ応答の返信後，別の子からクエリ応答メッセージを受信した端末 M_s は，手順 2. と同様にヒストグラムを統合し，再びスコア分布と k 番目のスコアの推定（2.4.3 項），および k 番目のスコアの補正（2.4.4 項）を行う．その後，端末 M_s は，手順 3. と同様にクエリ応答メッセージを作成し，自身の親に送信する．ただし，データの重複を防ぐため，既に返信したデータとそのスコアは返信リストから削除する．

クエリ応答の返信では，推定した k 番目のスコア以上のスコアをもつデータのみを返信する．また，中継端末で検索結果に入らないと判断できたデータは返信しない．さらに，自身および自身の子孫である端末のヒストグラムを統合した応答ヒストグラムをクエリ応答メッセージに添付することで，クエリ発行端末に近い端末ほど，推定に用いるヒストグラムに含まれるスコア数が増加し，k 番目のスコアをより正確に推定できる．これらにより，各端末は検索結果に入るデータを返信しつつ，検索結果に入らないデータの返信を抑止でき，検索精度を低下させずにトラヒックを削減できる．

図 2.2 および図 2.3 を用いて，クエリ応答の返信例を説明する．図 2.3 において，吹出しは各端末がクエリ応答に添付した応答ヒストグラムを示し，矢印はクエリ応答の流れを示す．例えば，M_4 は，図 2.2 に示すクエリヒストグラム（M_1[黄色], M_2[緑色]，M_4[青色] のヒストグラムを統合したもの）を用いて，k 番目のスコア

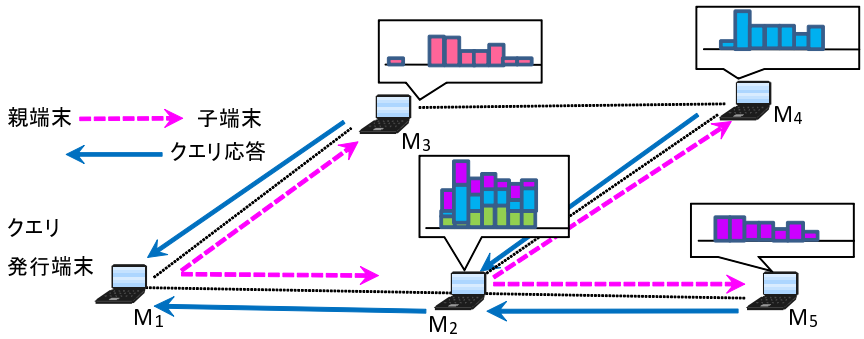


図 2.3: クエリ応答の返信例

を推定し，応答ヒストグラム（M_4[青色] のヒストグラム）と推定した k 番目のスコア以上のスコアをもつデータを，親である M_2 に返信する．

リンク切断の検出時

モバイルアドホックネットワークでは，端末の移動によりネットワークトポロジが動的に変化する．親とのリンクが切断された端末は，クエリ応答メッセージを返信できないため，検索結果の精度が低下する．そこで，提案手法では，リンク切断を検出した端末は，他の隣接端末にクエリ応答を送信する．以下では，親とのリンク切断を検出した端末 M_t が，他の隣接端末にクエリ応答を返信する動作について説明する．

1. 端末 M_t は，親以外の隣接端末にクエリ応答メッセージを転送する．ここで，複数の端末と隣接している場合，トラヒックや遅延の増分をできる限り小さくするため，隣接端末のうちクエリ発行端末からのホップ数が最小のものに転送する．

   どの端末とも隣接していない場合，クエリ応答メッセージを破棄する．

2. 自身の子以外の端末からクエリ応答メッセージを受信した端末 M_u は，端末 M_t がクエリ発行端末から自身までの経路上に存在しない場合，受信したメッセージを子からのメッセージと同様に扱い，クエリ応答返信の手順に従って親に返信する．一方，端末 M_t がクエリ発行端末から自身までの経路上に存

在する場合（端末 M_t が端末 M_u の親端末である場合も含む），M_u が親に返信したメッセージは端末 M_t まで返信されてしまう．そこで，この場合，端末 M_u は，受信したクエリ応答メッセージを親以外のいずれかの隣接端末に送信する．親以外の端末と隣接していない場合，クエリ応答メッセージを破棄する．

リンク切断の検出では，端末が親以外の隣接端末にクエリ応答を返信することで，検索結果をできる限りクエリ発行端末まで返信し，検索精度の低下を抑止できる．ただし，切断を検出した端末が孤立している場合や親からメッセージを受信した端末が他の端末と隣接していない場合，クエリ応答がクエリ発行端末まで返信されないため，検索精度が低下してしまう．

図 2.3 を用いて，リンク切断時の処理例を説明する．例えば，$M_2 - M_4$ 間のリンクが切断した場合，M_4 はクエリ発行端末からのホップ数が最小の M_3 にクエリ応答メッセージを送信する．一方，$M_1 - M_2$ 間のリンクが切断した場合，端末 M_2 は M_4 にクエリ応答メッセージを送信する．これを受信した M_4 は，M_2 がクエリ発行端末から自身までの経路上に存在するため，親以外の隣接端末（ここでは M_3）にクエリ応答メッセージを送信する．

2.4.3　スコア分布と k 番目のスコアの推定

提案手法では，各端末はクエリ応答を返信する時，クエリ発行端末から自身までの経路上の端末，および自身の子孫の端末のヒストグラムを統合した推定用ヒストグラム EH を用いて，ネットワーク内のデータのスコア分布および k 番目のスコアを推定する．以下ではまず，ヒストグラム内のスコア数を用いて推定する方法を説明する．次に，確率密度関数を用いて推定する方法を説明する．

ヒストグラム内のスコア数による推定

ヒストグラム内のスコア数による推定では，各端末は推定用ヒストグラム EH がネットワーク内のデータのスコア分布と等しいと考え，スコア分布と k 番目の

スコアを推定する．具体的には，推定を行う端末は，以下の式を満たす最大の自然数 a $(\leq C)$ を求める．

$$\sum_{l=a}^{C} \frac{n}{n_{EH}} \cdot e_l \geq k, \tag{2.1}$$

$$n_{EH} = \sum_{l=1}^{C} e_l. \tag{2.2}$$

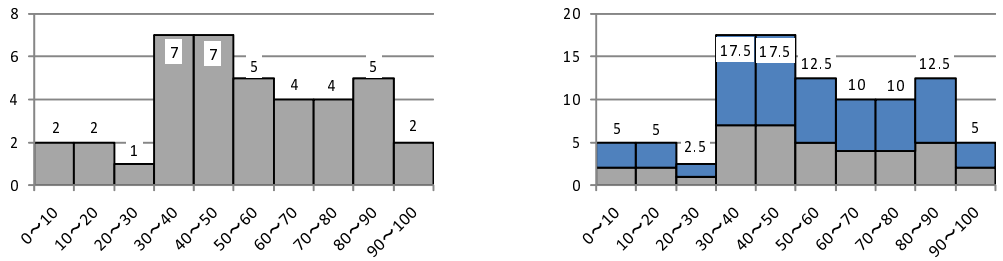
ここで，n はネットワーク内のデータ数，k は要求データ数，C は推定用ヒストグラム EH の階級数，e_j $(1 \leq j \leq C)$ は階級 c_j に含まれるスコアの数を示す．つまり，n_{EH} は推定用ヒストグラムに含まれるスコア数を示し，式 (2.1) は推定用ヒストグラム内のスコア数とネットワーク内のデータ数の比に基づいて，推定用ヒストグラムの各階級に含まれるスコア数を増やしたヒストグラムを，ネットワーク内のデータのスコア分布として推定していることを示す．このとき a は，階級 c_a 以上の階級に含まれるスコア数の合計が k 以上となる最大の整数を示す．次に端末は，以下の式を用いて，k 番目のスコア S_k を求める．

$$S_k = MIN + \frac{a(MAX - MIN)}{C}. \tag{2.3}$$

ここで，MIN と MAX は，スコアのとりうる値の最小値と最大値を示し，S_k は階級 c_a の下限値を示す．

図 2.4(a) のヒストグラムを用いて，ヒストグラム内のスコア数による推定を行う例を説明する．この図は，M_3 が 2.4.2 項のメッセージ処理において作成した推定用ヒストグラムを示す．式 (2.2) より $n_{EH} = 40$ となるため，$n = 100$ とすると，M_3 はヒストグラムの各階級に含まれるデータ数を 2.5 倍したものを，ネットワーク内のデータのスコア分布と推定する（図 2.4(b)）．$k = 10$，$c = 10$ とすると，$a = 9$ において，

$$\sum_{l=9}^{10} \frac{100}{40} \cdot e_l = \frac{100}{40}(2+5) \geq 10.$$



(a) 推定に用いるヒストグラム　　(b) ヒストグラム内のスコア数により推定されたヒストグラム

図 2.4: ヒストグラムを用いたスコア分布推定

と k を超えるため，M_3 は k 番目のスコア S_k を階級 c_9 の下限値 80 と推定する．

確率密度関数による推定

確率密度関数による推定では，各端末は，ネットワーク内のデータのスコア分布の種類を把握していることを利用して，スコア分布と k 番目のスコアを推定する．具体的には，推定を行う端末は，推定用ヒストグラムの各階級の中央値が，その階級に含まれるスコア数ずつ存在すると仮定し，ネットワーク内のデータのスコア分布の確率密度関数 $f(x)$ におけるパラメータを求める．次に端末は，確率密度関数 $f(x)$ の上側累積確率 $Q(x)$ が，以下の式を満たす x を求める．

$$Q(x) = \int_x^\infty f(t)dt = \frac{k}{n}. \tag{2.4}$$

ここで，x は上側累積確率が $Q(x)$ となるパーセント点を示す．式 (2.4) により，x は，上側累積確率がネットワーク内のデータに対する上位 k 個のデータの割合となるスコアを表すため，これを k 番目のスコア S_k と推定する．

図 2.4(a) のヒストグラムを用いて，確率密度関数による推定を行う例を説明する．ネットワーク内のデータのスコア分布が正規分布に従う場合，確率密度関数 $f(x)$ は，

$$f(x) = \frac{1}{\sqrt{2\pi}} exp(-\frac{(x-\mu)^2}{2\sigma^2}) \tag{2.5}$$

となる．ヒストグラムより平均 μ と標準偏差 σ は，それぞれ $\mu = 53.8$，$\sigma = 22.1$ となるため，$n = 100$，$k = 10$ として，式 (2.4) を満たす $x(= S_k)$ を求めると，$S_k = 82.1$ となる．

2.4.4 k 番目のスコアの補正

提案手法では，各端末は，2.4.3 項の方法に従って推定した k 番目のスコア以上のスコアをもつデータを返信するため，推定した k 番目のスコアが実際と異なる場合，性能が低下してしまう．特に，端末が k 番目のスコアを実際より高く推定した場合，実際の k 番目のスコアから推定した k 番目のスコアまでのスコアをもつデータは，検索結果に入るにも関わらず返信されないため，検索精度が低下する．

そこで提案手法では，セーフティマージンを用いてスコアの補正を行う．具体的には，推定した k 番目のスコアをセーフティマージンの大きさだけ減算した値に補正する．ここで，ネットワーク内のデータ数に対して推定用ヒストグラムに含まれるスコア数が少ない場合，推定した k 番目のスコアが実際と異なる可能性，および実際のスコアとの差が大きい可能性が高い．そのため，セーフティマージンは，ネットワーク内のデータ数が多く，ヒストグラム内に含まれるスコア数が少ないほど，大きくなるように設定する．具体的には，各端末は，以下の式を用いてセーフティマージンの大きさ $margin$ を決定する．

$$margin = \alpha \cdot \frac{n}{n_{EH}}. \tag{2.6}$$

ここで α は，事前に設定される正の定数であり，セーフティマージンに対する重み係数を示す．次に端末は，以下の式を用いて，k 番目のスコアを補正する．

$$S_k{}' = S_k - margin. \tag{2.7}$$

ここで，S_k は 2.4.3 項で推定した k 番目のスコア，$S_k{}'$ は補正後の k 番目のスコアを表す．つまり，ネットワーク内のデータ数が多く，ヒストグラム内のスコア数が

少ないほど，セーフティマージンは大きくなり，k 番目の推定スコアはその分低くなる．

2.5　性能評価

本節では，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 Qualnet4.0 [56] を用いた．

2.5.1　シミュレーション環境

1,000[m]×1,000[m] の 2 次元平面上の領域に 50 台の端末（$M_1, \ldots, M_{50}$）が存在する．各端末はランダムウェイポイント [6] に従い，0.5 [m/秒] の速度で移動する．停止時間は 60[秒] とした．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．ネットワーク内には，s [KB] のサイズのデータが 5,000 個存在し，各端末はそれぞれ 100 個のデータを保持するものとした．また，2.5.4 項を除いてネットワーク全体のスコア分布は正規分布に従うものとし，スコアの幅は [80, 320) とした．

各端末は，1,000 から 2,000 [秒] の間隔で Top-k 検索クエリを発行する．Top-k 検索手法には，提案手法において，スコア分布をヒストグラム内のスコア数により推定する場合と確率密度関数により推定する場合を用い，$C = 20$ とした．ここで，文献 [18] が提案している手法（従来手法と呼ぶ）において，基準値が 20 の場合を，比較手法として用いた．さらに，各端末が自身のもつデータの中からスコアの高い $R = k, \lceil k/2 \rceil, \lceil k/50 \rceil$ 個のデータを固定数返信する手法も，比較手法として用いた．例えば，$R = \lceil k/50 \rceil$ の場合は，端末数が 50 台であるため，クエリ発行端末は，k が 50 以下のときは 50 個のデータ，k が 50 以上のときは k 個のデータを受信する．また，全ての手法において，最大再送回数は 1 回，および各端末の最大待ち時間は (60 − 5× ホップ数) 秒（0 以下となる場合は 5 秒）とした．

表 2.1 に，シミュレーション実験に用いるパラメータをまとめる（太字はデフォルト値とする）．以上のシミュレーション環境において，各端末の初期位置をラン

表 2.1: パラメータ

パラメータ	値
k（要求データ数）	1,5,10,50,**100**,200,300
s（データサイズ）	0.01,0.1,0.5,**1**,3,5,10

表 2.2: メッセージサイズ

メッセージ	サイズ [B]
検索クエリ（提案手法および従来手法）	44+i_1
検索クエリ（固定数返信）	4+i_1
クエリ応答（提案手法）	22+$s \cdot i_2$
クエリ応答（従来手法）	3+$s \cdot i_2$
クエリ応答（固定数返信）	2+$s \cdot i_2$

ダムに決定し，7,200[秒] を経過させたときの以下の評価値を調べた．

- 検索結果の精度

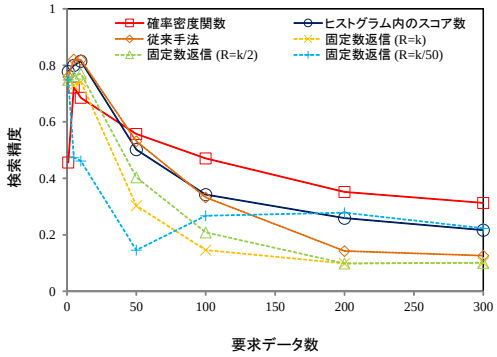
  上位 k 個のデータの中で，検索クエリの発行後 60[秒] の間に取得できたデータの数の割合．

- トラヒック

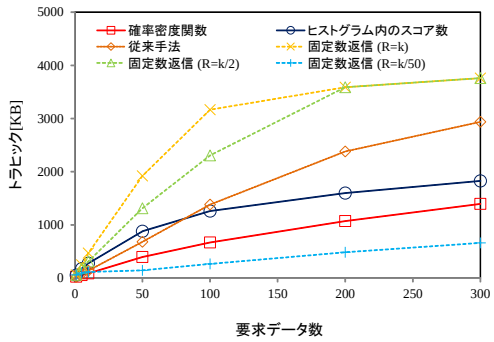
  シミュレーション時間内に Top-k 検索によって送信された検索クエリメッセージ，およびクエリ応答メッセージの Top-k 検索毎の平均バイト数．各メッセージのサイズを表 2.2 に示す．この表において，i_1 は検索クエリに含まれる経路端末リスト内の端末数，および i_2 は返信メッセージに含まれるデータ数を示す．

2.5.2 データ要求数 k の影響

要求データ数 k を変化させた時の提案手法の性能を調べた．その結果を図 2.5 および図 2.6 に示す．図中の各グラフにおいて,「確率密度関数」,「ヒストグラム内のスコア数」は，それぞれ提案手法においてスコア分布を確率密度関数を用いて推定する場合，およびヒストグラム内のスコア数により推定する場合を示す．また，

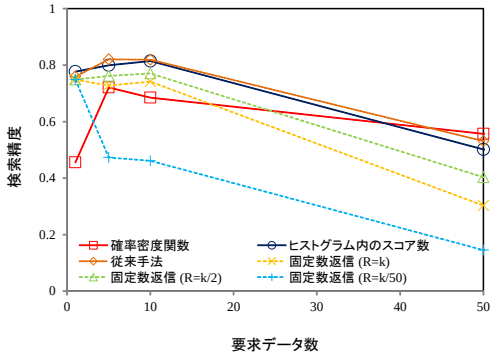


(a) 検索精度

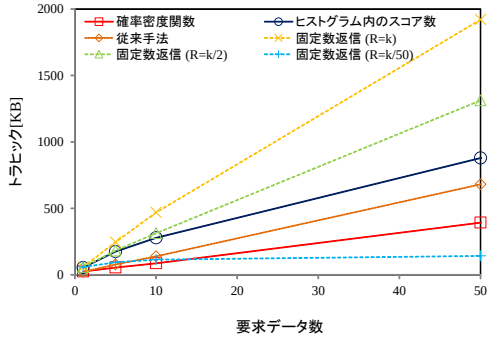


(b) トラヒック

図 2.5: 要求データ数 k の影響



(a) 検索精度



(b) トラヒック

図 2.6: 要求データ数 $k(\leq 50)$ の影響

「従来手法」は，文献 [18] で提案されている手法,「固定数返信 ($R = x$)」は，各端末が自身のもつデータの中からスコアの高い $R(= x)$ 個のデータを返信する手法を示す．これらのグラフにおいて，横軸は要求データ数 k を表している．縦軸は，図 2.5(a) および図 2.6(a) では検索精度，図 2.5(b) および図 2.6(b) ではトラヒックを表している．ここで，図 2.6 は，図 2.5 において k が 50 以下の結果を拡大したものである．

この結果から，要求データ数が小さい範囲では，提案手法においてヒストグラム内のスコア数を用いる場合と文献 [18] の手法，固定数返信 ($R = k, k/2$) が精度が高いことがわかる．ヒストグラム内のスコア数を用いた場合，k 番目のスコアを小さく推定することが多いため，要求データ数が少ない場合においても，上位

k 個のデータの返信漏れが少なく，検索精度が高い．従来手法および固定数返信 ($R = k,\ R = k/2$) では，全端末から必要なデータが返信されるため，検索精度が非常に高い．一方，確率密度関数を用いる場合，要求データ数が非常に小さい範囲では，k 番目のスコアを高く推定してしまう可能性が高く，多くの端末がデータを 1 つも返信できず，検索精度は非常に低くなる．

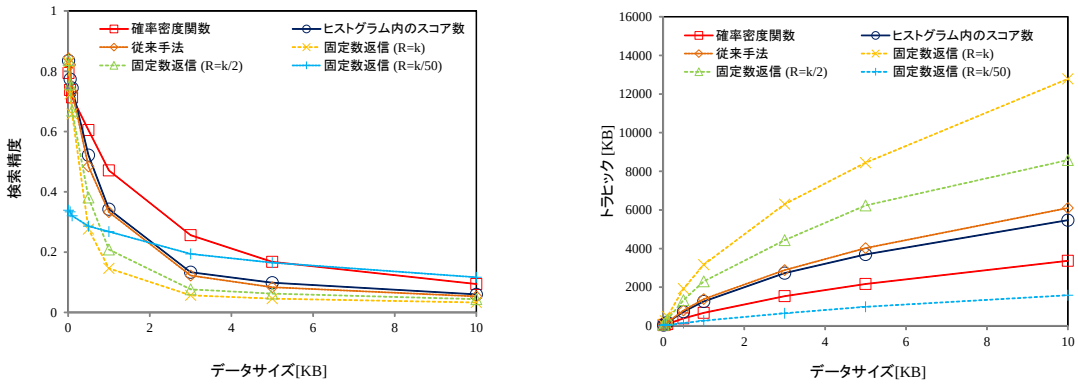
要求データ数が多くなると，全ての手法において，返信するデータが多くなり，パケットロスが増加するため，検索精度が低下する．文献 [18] の手法では，要求データ数が 100 より大きい場合において，ヒストグラム内のスコア数を用いる手法より検索精度が低くなっている．これは，絞り込みが甘いため，返信データ数が多く，パケットロスがより多く発生しているためである．要求データ数が大きい範囲では，提案手法において確率密度関数を用いる場合が比較手法より精度が高いことがわかる．これは，k 番目のデータをより詳細に推定することにより，無駄な返信データ数が少なく，パケットロスを他の手法より抑えることができるためである．

ここで，固定数返信 ($R = k/50$) において，要求データ数 $k = 50$ の場合，検索精度が低いのは，各端末が上位 50 個に入るデータを複数もっていても，1 つのデータしか返信しないためである．

2.5.3 データサイズの影響

データサイズ s を変化させた時の提案手法の性能を調べた．その結果を図 2.7 に示す．これらのグラフにおいて，横軸はデータサイズ s を表している．縦軸は，図 2.7(a) では検索精度，図 2.7(b) ではトラヒックを表している．

この結果から，データサイズが大きくなると，いずれの手法も，検索精度は低くなることがわかる．これは，データサイズが大きいと，各端末が返信するクエリ応答メッセージのサイズが大きくなり，パケットが衝突する機会が増加するからである．精度の低下の原因として，バッファ溢れも考えられるが，今回のシミュレーション実験では発生していない．データサイズが小さい範囲では，提案手法の検索精度は，文献 [18] の手法や固定数返信 ($R = k$) の場合とほぼ同じである．



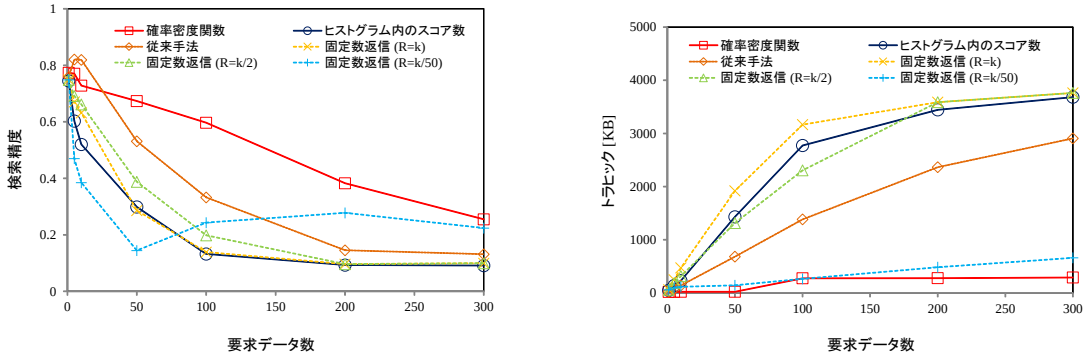
(a) 検索精度　　(b) トラヒック

図 2.7: データサイズ s の影響

データサイズが大きい範囲では，提案手法の確率密度関数を用いる場合，もしくは固定数返信 $(R = k/50)$ の検索精度が高い．これは，提案手法でヒストグラム内のスコア数を用いる場合や文献 [18] の手法，固定数返信 $(R = k)$ では，返信するデータが多い分，トラヒックが大きくなり，パケットが衝突する機会が多いからである．提案手法の確率密度関数を用いた場合では，スコアが高いデータのみを返信し，返信データ数が抑制されているため，比較手法よりもトラヒックが小さく，検索精度が高い．一方，ヒストグラム内のスコア数を用いる場合，推定した k 番目のスコアを小さく推定する傾向があるため，確率密度を用いた場合よりもトラヒックが大きくなりやすいが，固定数返信 $(R = k/2, k)$ よりトラヒックは抑えられている．

2.5.4　Zipf 分布の影響

次に，データのスコアが正規分布ではなく，Zipf 分布 [75] に従う場合のデータ要求数 k の影響を調べた．Zipf 分布は，分布に大きな偏りがあり，ここでは，スコアが極端に大きいデータが少数存在し，スコアが小さいデータが多く存在するよう設定した．そのため，返信すべき必要なデータと不必要なデータをスコアから判断しやすいという特徴ある．ここで，Zipf 分布における，データのスコアの幅は [0, 2,000) とした．その結果を図 2.8 に示す．これらのグラフにおいて，横軸



(a) 検索精度　　(b) トラヒック

図 2.8: Zipf 分布における要求データ数 k の影響

は要求データ数 k を表す．縦軸は，図 2.8 (a) では検索結果の精度，図 2.8 (b) ではトラヒックをそれぞれ表す．

この結果から，確率密度関数を用いた場合，トラヒックが小さく抑えられ，検索精度も高いことがわかる．一方，ヒストグラム内のスコア数を用いた場合は，推定の際に k 番目のスコアを非常に高く，もしくは非常に小さく推定してしまうことが多い．これは，Zipf 分布の場合，スコアに大きな偏りがあるためである．その結果，トラヒックが大きく，かつ検索精度も低くなっている．文献 [18] の手法では，データ要求数 k が大きくなるにつれて，絞り込みが甘くなるため，トラヒックが大きくなり，検索精度も下がる．固定数返信の場合，分布により大きな差異が発生しないため，正規分布とほぼ同じ結果になっている．提案手法において確率密度関数を用いる場合，比較手法よりもトラヒックが小さく，検索精度が非常に高くなっている．これは，Zipf 分布の形状を決定する要素（つまり，形状変数）を計算し，データ分布を推定することにより，スコアが小さいデータの返信を抑制することができるためである．これにより，トラヒックが他の手法より，非常に小さく，パケットロスの頻度を小さくすることができる．

2.5.5 議論

Top-k 検索を使用する環境およびユーザの意図により，指定する k の大きさは変化する．例えば，救急車や薬品などの物理的な資源を割り当てる場合では，その

資源の量により k が変わる．救急車などの場合には，k は小さく，薬品などの場合は大きく設定される．一方で，ネットワークを構築している領域の全体的な状況を把握したい場合などは，大きい k が指定されることが考えられる．負傷度の高い上位の被害者の情報を広く取得することで，どの程度の医療従事者や医療施設が必要かを把握することができる．そのため，評価においては，k をある程度の広い範囲内で変化させて評価した．その結果では，k の大きさに関わらず，提案手法が有効であることを確認できた．

しかし，k が大きい場合では，提案手法においても，検索の精度が 40% 程度と高くない．さらに，高精度を達成したい場合では，推定した k 番目のスコアおよび既に取得したデータの識別子をクエリメッセージに添付して，検索を再実行するなどの方法が有効である．その場合においても，単純手法に比べ，1 度目の精度が提案手法の方が高いため，再検索するデータ数が少なくなり，トラヒックを抑えることができる．

2.6　むすび

本章では，モバイルアドホックネットワークにおいて，トラヒックの削減と検索結果の精度の低下の抑止を目的とする Top-k 検索手法を提案した．提案手法では，各端末が検索クエリやクエリ応答にヒストグラムを添付して送信することで，ネットワーク全体のデータのスコア分布と k 番目のスコアを推定する．各端末は推定した k 番目のスコア以上のスコアをもつデータを返信することで，できる限り検索結果に入るデータのみを返信する．また，返信先の端末とのリンクが切断した端末は，他の隣接端末にクエリ応答を転送することで，検索精度の低下を抑止する．

シミュレーション実験の結果から，提案手法は，スコア分布と k 番目のスコアを推定して，検索結果に入るデータのみを返信することで，トラヒックを削減しつつ，検索精度の低下を抑止できることを確認した．

提案手法では，簡単化のため，ネットワーク内のデータ数，ネットワーク内のデータのスコアが従う分布の種類，およびスコアのとりうる値の範囲が既知と想

定した．しかし実環境では，これらは時々刻々と変化すると考えられるため，各端末が予め把握することは困難な場合がある．そこで今後は，各端末が取得したデータやそのスコアのログを記録し，定期的に統計をとるなど，上記の値を動的に推定する必要がある．

第3章　複製配置を考慮したTop-k検索手法

3.1　まえがき

モバイルアドホックネットワークでは，端末の移動によりネットワークトポロジが動的に変化し，ネットワークが分断される可能性がある．この場合，分断された一方のネットワークに属する端末は，他方のネットワークに属する端末のもつデータにアクセスできなくなる．そのため，検索精度が低下してしまう可能性がある．ネットワーク分断時の検索精度を向上させるアプローチとして，各端末が他の端末の保持するオリジナルデータの複製を作成し，ネットワークが分断してもデータにアクセスできるようにすることが有効である．

これまでに筆者の研究グループでは，文献 [18] において，モバイルアドホックネットワーク上のトラヒックの削減と検索精度の維持を目的とする Top-k 検索手法，および文献 [21] において，検索実行時に，各端末がスコアが高い返信データを複製として配置する，Top-k 検索のための複製配置手法を提案した．ここで，文献 [21] の手法により複製を配置した場合，多くの端末が，スコアの高いデータの複製を保持する可能性が高い．このような状況で， 文献 [18] の手法で Top-k 検索を行うと，同じデータ（複製）が複数回返信されたり，クエリ発行端末から遠い端末からのデータが返信により，大きなトラヒックが発生してしまう．

そこで本章では，文献 [18] の手法を拡張し，モバイルアドホックネットワークにおけるデータの複製配置を考慮した Top-k 検索手法を提案する．提案手法では，同じデータが複数回返信されることを抑止するために，各端末が自身のもつデータのうち上位 k 個に入る可能性のあるデータの識別子をクエリメッセージに添付する．これにより，クエリの伝搬経路上で，自身よりクエリ発行端末に近い端末が

保持するデータを把握することができる．データの返信時に各端末は自身よりクエリ発行端末に近い端末が保持しているデータを返信しない．また，異なる経路上を伝搬したクエリメッセージを受信したとき，その経路上に同じデータを保持する端末が存在した場合，その端末の方がクエリ発行端末に近ければ，そのデータを返信しない．これにより，遠くの端末からのデータの返信や，重複したデータの返信を抑制することができる．

さらに，全ての端末にメッセージを送信しなくても，一部の端末からのみで必要なデータを取得できる可能性がある．そのため，提案手法では，各端末は既に上位 k 個のデータが取得できるかを自律的に判断し，クエリメッセージの転送を停止し，データの返信をすぐに開始する．これにより，トラヒック，および検索時間の削減が可能である．

以下では，3.2 節で関連研究を紹介する．3.3 節で想定環境について述べ，3.4 節で先行研究を紹介する．3.5 節で提案手法について述べ，3.6 節でシミュレーション実験の結果を示す．最後に 3.7 節で本章のまとめと今後の課題について述べる．

3.2　関連研究

分散環境において，複製を用いた Top-k 検索手法がいくつか提案されている [43, 66, 73]．文献 [73] では，分散データベース環境において，BRANCA と呼ばれる手法を提案している．BRANCA では，サーバが検索結果をキャッシュとして保持し，検索時にはキャッシュ内のデータをまず探索する．キャッシュ内に必要な（上位 k 個の）データが存在する場合，検索を終了し，存在しない場合のみ，他のサーバにアクセスし上位 k 個のデータを取得する．文献 [43] では，分散データベース環境において，ARTO と呼ばれる手法を提案している．ARTO では，BRANCA と同様の処理を行うが，取得できなかったデータのみを他のサーバにアクセスし，取得するため，BRANCA より検索データ数が少なくなる．端末が検索結果をキャッシュすることは，モバイルアドホックネットワークにおいても，有用といえるが，キャッシュ可能なデータ数が固定サーバよりも，かなり小さいことが一般的である．そのため，残りのデータを取得するための，検索クエリのメッセージ処理が

重要な課題となる．

文献 [66] では，非構造 P2P ネットワークにおいて，スーパーピアを用いて，効率的な Top-k 検索を実現する SPEERTO と呼ばれる手法が提案されている．SPEERTO では，スーパーピアが自身の子端末のデータを複製として保持し，スーパーピア間の通信のみで Top-k 検索処理を行う．しかし，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，スーパーピアの選出は非常に難しい．また，スーパーピア間で直接通信できるとは，限らないため，検索範囲が非常に広くなってしまう可能性がある．

3.3 想定環境

基本的に 2 章と同様に，モバイルアドホックネットワークを構成する端末が，他の端末が保持するデータに対して，Top-k 検索を行う環境を想定する．

ネットワーク内には m 個の端末（識別子：$M_1, M_2, ..., M_m$）が存在し，各々が自由に移動する．各端末は c 個のオリジナルデータを保持している．また，自身のオリジナルデータを格納する記憶領域以外に，データ ρ 個分の複製を配置する記憶領域を保持している．簡単化のため，すべてのデータや複製のサイズは等しいものとする．

時間の経過に伴ってデータが更新されることでスコアが変化する．この際，簡単化のため，すべてのデータおよびそのスコアはある一定の周期 T で変化するものとする．なお，各データが更新されるタイミングは一致するとは限らないものとする．

3.4 先行研究

これまでに筆者のグループでは，先行研究として，アドホックネットワーク上の Top-k 検索手法 [18]，および Top-k 検索のための複製配置手法 [21] を提案した．本章では，これらの先行研究について概説する．

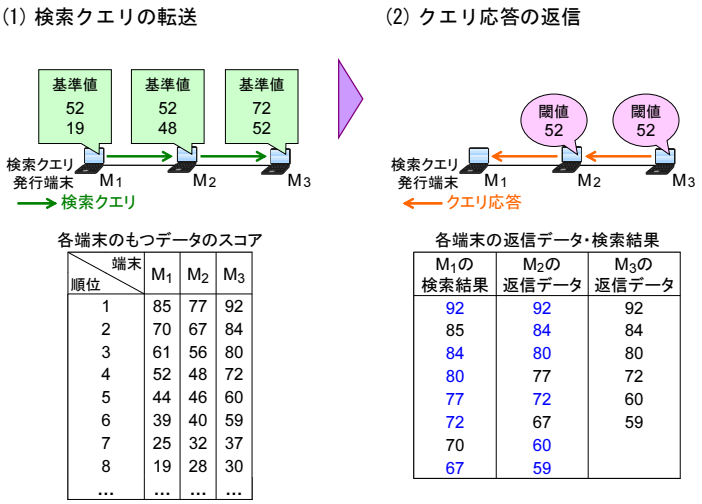
3.4.1　Top-k 検索手法

文献 [18] の提案手法では，アドホックネットワーク上のトラヒックの削減と検索精度の維持を目的としている．この手法では，検索結果の候補となるデータを絞り込むため，検索クエリを発行した端末が N 個の基準値を添付して隣接端末に送信する．ここで，基準値は検索クエリの発行端末が保持するデータの上位 $\frac{k}{N}$ 個ごとのスコアに設定されるため，i 個目の基準値は，それぞれ $\frac{k}{N}i(1 \leq i \leq N)$ 個のデータを取得できることを保証する．

次に，検索クエリを受信した各端末は，各自の保持するデータのスコアに基づいて基準値を逐次更新する．具体的には，基準値を，受信した検索クエリに添付された基準値と，自身が保持するデータの上位 $\frac{k}{N}$ 個ごとのスコア（計 $2N$ 個）の中で，上位 N 個以内のスコアに更新する．また，検索クエリにメッセージ送信端末とその親端末の識別子を添付することで，検索クエリを発行した端末を根とする木構造における自身の親と子，検索クエリの発行端末から自身までの経路，および親以外の隣接端末の検索クエリの発行端末からのホップ数を把握する．

検索クエリの転送時に更新した基準値を用いて，各端末は検索結果の候補となるデータを絞り込み，クエリ応答の返信では，この候補となったデータのみを自身の親へ返信する．具体的には，N 個目の基準値が k 位のスコアに対応しているため，このスコア以上のデータをクエリ応答として返信する．さらに，このスコアを閾値として，クエリ応答に付加する．クエリ応答を中継する端末が，受信した返信データと閾値から，自身のもつデータのうち，検索結果に入る可能性のあるもののみを返信し，それに伴い，クエリ応答に付加する閾値を更新する．また，端末がリンク切断を検出した場合，他の隣接端末を経由する返信経路を構築し，構築した経路に沿ってクエリ応答を送信する．

図 3.1 を用いて，文献 [18] の Top-k 検索手法の処理例を説明する．この図は，M_1 が 2 個の基準値（$N = 2$）を用いて上位 8 個以内のデータを検索する例（$k = 8$）を示している．図中の (1) における表は各端末の保持するデータのスコアを，(2) における表は各端末の返信データと，検索クエリの発行端末が取得した検索結果を示している．なお，(2) における表中の青字は，検索結果または返信データ内で，自身が保持していないデータのスコアを表している．また，(1) における四角吹出し



順位 \ 端末	M_1	M_2	M_3
1	85	77	92
2	70	67	84
3	61	56	80
4	52	48	72
5	44	46	60
6	39	40	59
7	25	32	37
8	19	28	30
...	...	...	...

M_1の検索結果	M_2の返信データ	M_3の返信データ
92	92	92
85	84	84
84	80	80
80	77	72
77	72	60
72	67	59
70	60	
67	59	

図 3.1: Top-k 検索の処理例

は各端末の基準値を，(2) における丸吹出しは各端末の閾値を表している．例えば，M_2 は，M_1 から受信した検索クエリメッセージに含まれる基準値リスト 52，19 と自身の $\frac{k}{N}$ 個ごとのスコア 48，28 の中から上位 2 個のスコア 52，48 を M_2 の基準値とし，検索クエリの転送を続ける．検索クエリの転送後，子がいない M_3 は，2 番目の基準値 52 を閾値とし，52 以上のスコアとそのデータを M_2 に返信する．M_2 は自身の閾値を，M_3 から受信したメッセージに含まれる閾値 52 と自身の基準値 48 の大きい方に更新し，受信した返信リストと自身の保持するデータの中で，スコアが閾値 52 以上のデータとそのスコアを M_1 に返信する．

3.4.2 複製配置手法

文献 [21] の複製配置手法では，文献 [32] で提案されたパス複製法に従って，各端末が自身のクエリ応答を親に返信する際に，返信データの複製を配置する．このアプローチは，複製の配置によって追加のトラヒックが発生しないため，通信帯域が限られているアドホックネットワークにおいて有効であると考えられる．ここで，一般的に端末の記憶領域には制限があり，端末が作成できる複製の数は限られている場合が多い．このとき，すべての端末が単純にスコアの高いデータの複製を優先して配置すると，端末間で保持する複製の種類が類似してしまう．相

互接続している端末間で複製の重複が多いと，これらの端末全体で保持できる複製が少なくなる．その結果，ネットワークが分断した場合に，検索結果に入るべきデータを取得できない可能性が高くなる．そこで，文献 [21] の手法では，高順位のデータを優先しつつ，配置するデータの種類を多様化するように配置する．具体的には，各データのスコアと乱数からなる評価値を定義し，その評価値が大きいデータの複製を配置する．

また，データの更新によるスコアの変化が発生する環境では，データの有効期限を考慮せずに複製を配置すると，多くの端末が有効期限までの残り時間の短い複製を配置してしまう可能性がある．その結果，多くの複製が短時間のうちに無効になってしまうため，複製配置の効果が小さくなる．そこで，文献 [21] の手法では，データの更新によるスコアの変化が発生する環境では，スコアが高く有効期間が長いデータの複製を優先的に配置する．

3.5　提案手法

本章で提案する Top-k 検索手法は，基本的には文献 [18] と同様の手順により Top-k 検索を実行する．しかし，文献 [18] の手法では，複製を考慮せずに基準値の決定や返信を行っているため，3.1 節で述べたように，同じデータが複数回返信されたり，クエリ発行端末から遠い端末によるデータ返信が行われたりすることで，大きなトラヒックが発生してしまう．

そこで提案手法では，文献 [18] の手法を拡張して，複製の存在を考慮して基準値および返信データを決定する．具体的には，各端末が自身のもつデータのうち上位 k 個に入る可能性のあるデータの識別子をクエリメッセージに添付する．これにより，クエリの伝搬経路上で，自身よりクエリ発行端末に近い端末が保持するデータを把握することができる．データの返信時に各端末は自身よりクエリ発行端末に近い端末が保持しているデータを返信しない．また，異なる経路上を伝搬したクエリメッセージを受信したとき，その経路上に同じデータを保持する端末が存在した場合，その端末の方がクエリ発行端末に近ければ，そのデータを返信しない．これにより，遠くの端末からのデータの返信や，重複したデータの返

信を抑制することができる．

さらに，全ての端末にメッセージを送信しなくても，一部の端末からのみで必要なデータを取得できる可能性がある．そのため，提案手法では，各端末は既に上位 k 個のデータが取得できるかを自律的に判断し，クエリメッセージの転送を停止し，データの返信をすぐに開始する．提案手法の詳細を以下に示す．

3.5.1 基準値の設定

提案手法では，文献 [18] の手法と同様に，N 個のスコアを基準値として設定する．ここで，端末 M_i における N 個の基準値 $B(i,j)(1 \leq j \leq N)$ はそれぞれ，検索クエリを発行した端末が，M_i までの経路上の端末から $\frac{k}{N}j(1 \leq j \leq N)$ 個のデータ取得できることを保証するスコアとなる．

各端末 M_i は，自身が保持するデータ（複製）に対して，基準値を以下のように設定する．

$$S(i, \frac{k}{N}j) \quad (1 \leq j \leq N) \tag{3.1}$$

ここで，$S(x,y)$ は，端末 M_x が，自身が保持する全データ（複製）のスコアを降順に並べたときの y 番目のスコアを示す．さらに，各端末は，受信した検索クエリメッセージに添付された基準値と自身がもつデータの基準値のうち，上位 N 個を自身の基準値 $B(i,j)$ とする．なお，クエリ発行端末の基準値（初期値）は，その端末がもつデータの上位 $\frac{k}{N}$ 個ごとのスコアとなる．

3.5.2 検索クエリの転送

提案手法では，検索結果の候補となるデータを絞り込むため，各端末が自身の基準値を検索クエリに添付して隣接端末に送信する．また，これを受信した端末が基準値を更新することで，絞込みの効率を向上させる．以下では，端末 M_p が Top-k 検索クエリを発行する際の各端末の動作について説明する．

1. クエリ発行端末 M_p は検索条件，および要求データ数 k を指定する．さらに，3.5.1 項の方法に従って，基準値の初期値を設定する．

2. 端末 M_p は，隣接する端末に検索クエリメッセージを送信する．このメッセージには，検索クエリ発行端末 M_p の識別子，要求データ数 k，検索条件，端末 M_p の基準値リスト $B(p,j)(1 \leq j \leq N)$，N 番目の基準値 $B(p,N)$ 以上のスコアをもつデータの識別子リスト，ホップ数が含まれる．ホップ数は，クエリ発行端末までのホップ数を示し，この場合 0 となる．また，クエリ応答の返信先を把握するため，メッセージ送信端末とその親端末の識別子が含まれる．この場合，メッセージ送信端末は M_p となり，親端末には何も設定されない．

3. 検索クエリメッセージを初めて受信した端末 M_q は，メッセージ送信端末を自身の親とし，ホップ数とメッセージ送信端末の組，およびデータの識別子リストを記録する．受信したデータの識別子リストを調べ，自身が同じ識別子のデータを所持している場合，そのデータは基準値の計算には用いない（所持していないものとして扱う）．端末 M_q は，手順 1. と同様に，自身のもつデータ，および受信したクエリメッセージに含まれる基準値リストから，3.5.1 項の方法に従って，自身の基準値 $B(q,j)(1 \leq j \leq N)$ を決定する．

4. 端末 M_q は，メッセージ送信端末と親端末の識別子をそれぞれ M_q と自身の親端末の識別子とし，基準値リストを自身の基準値 $B(q,j)(1 \leq j \leq N)$ とする．データの識別子リストは，受信したクエリメッセージ中のデータの識別子リストに対して，自身が所持し，かつ，受信したリストに含まれないデータのうち，N 番目の基準値 $B(q,N)$ 以上のスコアをもつデータ識別子を追加したものとする．また，ホップ数を 1 増やし，これらの情報を含めた検索クエリメッセージを自身の隣接端末に送信する．

   データの識別子リストに変化がある場合，端末 M_q は，隣接端末にメッセージを送信し，手順 3. に戻る．一方，データの識別子リストに変化がない場合（つまり，M_q がデータの識別子リストに加えるデータがない場合），M_q はクエリの転送を中止（この処理を‘クエリ停止’と呼ぶ）し，クエリ応答の返信を開始する．

5. 同じ検索クエリメッセージを再び受信した端末 M_r は，メッセージ送信端末

の親端末が自身の場合，メッセージ送信端末を自身の子とする．一方，メッセージ送信端末が自身の子でない場合，手順3. と同様に，メッセージに含まれるホップ数とメッセージ送信端末の組を記録する．受信した端末と自身の端末のクエリ発行端末からのホップ数を比較して，自身の方が大きい場合，または，受信した端末と自身の端末のクエリ発行端末からのホップ数が同じ，かつ，送信端末の受信した端末の識別子を比較して自身の方が小さい場合，データの識別子リストを記録する．さらに，受信したデータの識別子リストを調べ，自身が同じ識別子を所持している場合，そのデータを所持していないものとして，3.5.1 項に従って，自身の基準値 $B(r, j)(1 \leq j \leq N)$ を再計算する．これは，複数の経路上から同じデータが返信されるのを防ぐためである．まず，ホップ数を比較することにより，クエリ発行端末により近い端末が，データを返信することを可能としている．ここで，送信端末と受信端末は基本的に通信範囲内にいるので，互いが送信したメッセージを相手が受信できていることを把握できる．そのため，上記の手順により，両方がデータ返信を取りやめることはなく，一方（より遠い端末）のみが返信を取りやめることを可能とする．

検索クエリの転送において，基準値を更新することで検索結果の候補となるデータの絞込みができる．さらに，クエリの伝搬経路上において，自身よりクエリ発行端末に近い端末のデータの保持状況を把握できるため，無駄なデータ返信を防ぐことができる．また，各端末は，検索クエリメッセージに含まれるメッセージ送信端末とその親端末の識別子により，検索クエリ発行端末を根とする木構造における自身の親と子を把握できる．ホップ数とメッセージ送信端末の組から，親子関係ではない隣接端末，自身，および検索クエリ発行端末の位置関係を把握できる．加えて，各端末が適切なタイミングでクエリメッセージの送信を停止できるため，トラヒックおよび検索時間を削減することができる．

3.5.3　クエリ応答の返信

3.5.2 項の手順により，各端末は検索結果の候補となるデータを絞り込めるため，クエリ応答では，この候補となったデータのみを返信する．また，クエリ応答を中継する各端末は，受信した情報から，自身の返信予定のデータの中で返信の必要がないデータを検出した場合，その返信を行わない．以下では，3.5.2 項の手順の後，各端末がクエリ応答を返信する動作について説明する．

1. 自身の子がいない端末 M_s は，クエリ応答メッセージを自身の親に返信する．このメッセージには，検索クエリ発行端末 M_p，閾値，および返信リストが含まれる．閾値は推測した k 位のスコア $B(s, N)$ である．返信リストには，自身が所持し，かつ，受信した識別リストに含まれないデータのうち，閾値以上のスコアをもつものが含まれる．

2. クエリ応答メッセージを受信した端末 M_t は，自身の閾値を受信したメッセージに含まれる閾値と $B(t, N)$ の大きい方に更新する．そして，メッセージに含まれる返信リストから閾値未満のデータを削除する．また，自身が所持し，かつ，クエリ転送時に受信したデータの識別子リスト，および，返信リストに含まれないデータのうち，閾値以上のスコアをもつものを返信リストに追加する．返信リストに含まれるデータ数が k より大きい場合，閾値を返信リストに含まれるデータの k 番目のスコアに更新し，閾値未満のデータを返信リストから削除する．

   すべての子からのクエリ応答を受信した端末は，自身の親に更新したクエリ応答メッセージを送信する．

クエリ応答の返信では，閾値を用いて，検索結果に入らないと推測されたデータを返信しないことで，トラヒックを削減する．また，データの識別子リストを参照し，自身よりクエリ発行端末に近い端末がデータを返信したり，他の経路上の端末が返信するデータを返信しないことで，さらにトラヒックを削減する．

3.6 性能評価

本節では，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 Qualnet4.0 [56] を用いた．

3.6.1 シミュレーション環境

600[m]×600[m] の 2 次元平面上の領域に，50 台の端末（識別子 : $M_1, M_2, ..., M_{50}$）が存在する．各端末はランダムウェイポイント [6] に従い，v[m/秒] の一定の速度で移動する．また，各端末は IEEE 802.11b を使用し，伝送速度 11[Mbps]，通信伝播距離が 100[m] 程度となる送信電力でデータを送信する．各端末は 1[KB] のサイズのデータ（オリジナル）を 100 個保持するものとし，各端末のもつデータのスコアは 0 から 100 の範囲の一様乱数に従う．また，各端末は ρ 個のデータを複製として自身の記憶領域に保持できる．

複製の配置は，文献 [21] の手法に基づいて決定する．ここで，文献 [21] の複製配置手法では，データ更新の有無により，複製の配置方法が異なる．データ更新がある場合，更新間隔も考慮に入れて配置する複製を決定する．3.6.2 項では，データ更新がないものとし，3.6.3 項では，各データの更新間隔は 2,000 秒とし，各データが最初に更新されるタイミングはランダムに決定した．

表 3.2 に，シミュレーション実験に用いるパラメータをまとめる（太字はデフォルト値とする）．以上のシミュレーション環境において，各端末は，端末 M_1 から端末 M_{50} まで，順番に 1 回ずつ N 個の基準値を用いて，上位 k 個以内のデータを要求する検索クエリを 60[秒] 間隔で発行する．Top-k 検索手法には，提案手法，文献 [18] の手法（従来手法），および各端末が k 個のデータを返信する単純手法を用い，以下の三つの性能指標について評価した．また，全ての手法において，最大再送回数は 20 回，および各端末の最大待ち時間は $60/(\text{ホップ数}+1)$ 秒とした．本章では，クエリ停止の有効性を確認するため，検索時間についても評価する．

- 検索精度
  ネットワーク全体の上位 k 個以内のスコアをもつデータのうち，検索クエリの発行端末が取得したデータ数の割合の全試行における平均．

表 3.1: メッセージサイズ

メッセージ	サイズ [B]
検索クエリ（提案手法）	4 $+3N+i_1$
検索クエリ（従来手法）	4 $+3N$
検索クエリ（固定数返信）	4
クエリ応答（提案手法および従来手法）	3$+1000\cdot i_2$
クエリ応答（固定数返信）	2$+1000\cdot i_2$

表 3.2: パラメータ

パラメータ	値
ρ（配置可能複製数）	0,20,40,**50**,60,80,100
v（移動速度）	0,1,2,**3**,4,5
k（要求データ数）	20,40,**50**,60,80,100
N（基準値数）	5,**10**,15,20,25,30,35,40,45,50

- トラヒック
  1 つのクエリに対して，ネットワーク全体で送信されたメッセージ，およびデータの合計サイズの全試行における平均．各メッセージのサイズを表 3.1 に示す．この表において，i_1 は検索クエリに含まれるデータの識別子リスト内の識別子数，および i_2 は返信メッセージに含まれるデータ数を示す．

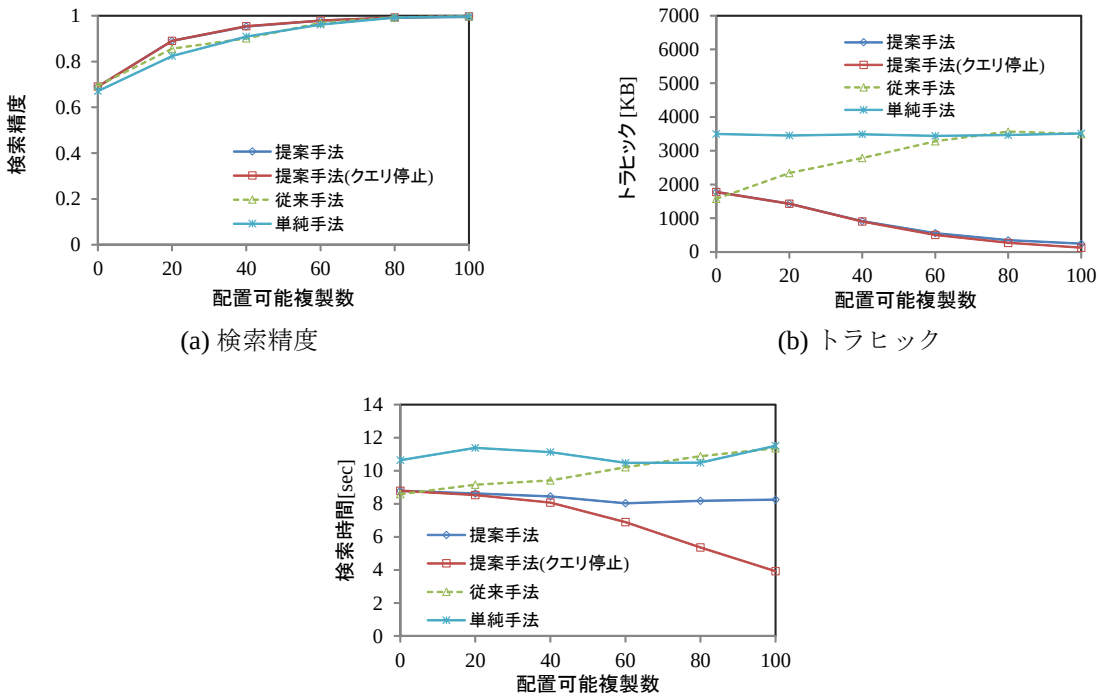
- 検索時間
  クエリ発行端末がクエリを送信してから，検索結果を取得するまでの時間．

3.6.2　データ更新が発生しない環境における評価結果

まず，データの更新が発生しない環境における，シミュレーション結果を示す．

配置可能複製数の影響

配置可能複製数 ρ を変化させた時の提案手法の性能を調べた．その結果を図 3.2 に示す．これらのグラフにおいて，横軸は配置可能複製数 ρ を表している．縦軸



(a) 検索精度

(b) トラヒック

(c) 検索時間

図 3.2: データ更新が発生しない環境における配置可能複製数の影響

は，図 3.2(a) では検索精度，図 3.2(b) ではトラヒック，および図 3.2(c) では検索時間を表している．また，図中の各グラフにおいて,「提案手法」はクエリ停止を用いない場合の提案手法,「提案手法+クエリ停止」はクエリ停止を用いる場合の提案手法,「従来手法」は文献 [18] で提案されている手法，および「単純手法」は各端末が k 個返信する手法を示す．

図 3.2(a) から，$\rho = 0$ のとき，全ての手法でほぼ同じ検索精度となっているのがわかる．ρ が増加するにつれて，全ての手法で検索精度が増加する．これは，複製が配置されることによってアクセス可能なデータが増加するためである．全ての ρ において，提案手法の検索精度が最も高い．これは，複製がより多く配置された場合に，返信データ数を抑制でき，トラヒックが小さいため，パケットロスが他の手法に比べて少ないからである．従来手法では，配置されている複製が増加する毎に返信データが多くなるため，トラヒックが増加する．そのため，提案手法よりパケットロスが若干多い．単純手法は，ρ が小さい場合，提案手法および従来

手法より検索精度が低い．これは，絞込みを行わないために，返信データ数が多いからである．

図 3.2(b) の結果より，先ほども述べたが，提案手法では，配置されている複製数が増加する毎に返信データ数が少なくなるため，トラヒックが減少する．従来手法では，配置されている複製数が増加する毎に返信データ数が多くなるため，トラヒックが増加する．単純手法は，返信データ数が複製に関係ないため，トラヒックが一定である．

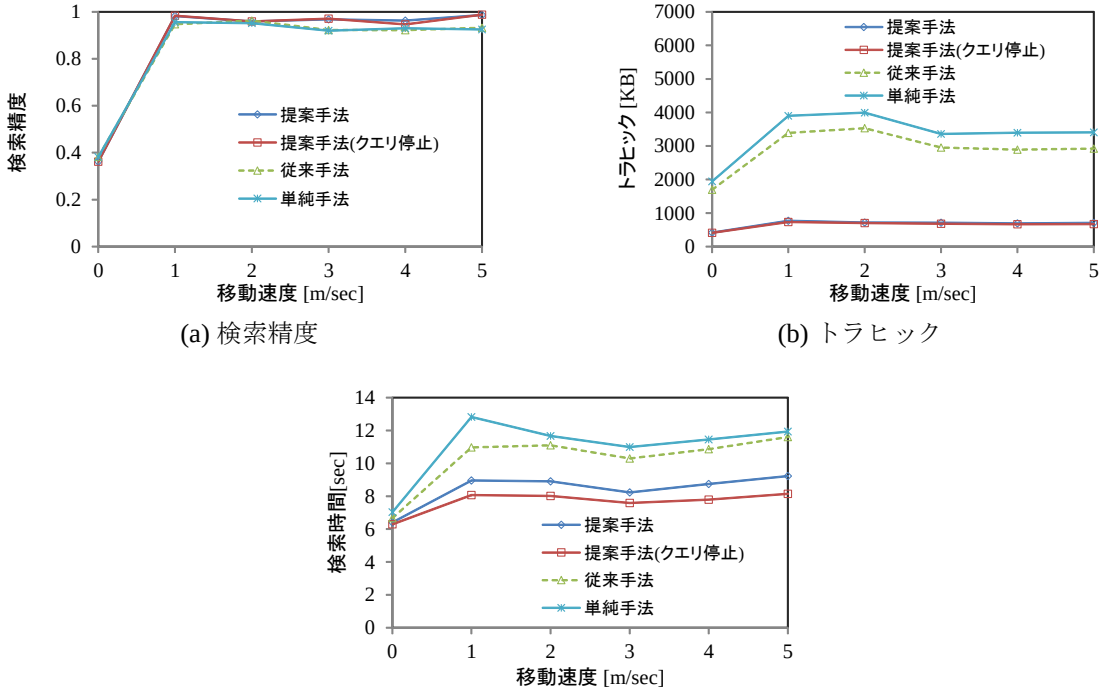
図 3.2(c) の結果より，検索時間はトラヒックと相関があることがわかる．トラヒックが増加するとデータ送信に時間がかかるため，検索時間も増加する．ただし，提案手法は，複製数が増加するにつれて，トラヒックが減少するのに対し，検索時間は一定である．この結果から，1500[KB] よりトラヒックを減少させても，検索時間が変かわらないことがわかる．クエリ停止を用いた提案手法では，検索時間が減少している．これは，複製がより多く配置されるとクエリ停止を行う機会が多くなるため，検索クエリのホップ数を減少させることができるからである．

移動速度の影響

次に，端末の移動速度 v の影響を調べた．その結果を図 3.3 に示す．これらのグラフにおいて，横軸は端末の移動速度 v を表している．縦軸は，図 3.3(a) では検索精度，図 3.3(b) ではトラヒック，および図 3.3(c) では検索時間を表している．

図 3.3(a) より，$v = 0$ のとき（つまり，端末が停止しているとき），全ての手法において検索精度が低いことがわかる．これは，初期状態でネットワーク分断が発生しており，シミュレーション終了までアクセス不可能なデータが存在するためである．端末が移動すると，ネットワークトポロジが変化するため，複製が広い範囲に配置され，ネットワーク分断が発生したとしても，検索精度を維持することができる．提案手法が他の手法よりも，検索精度が高いのは，低トラヒックで検索でき，パケットロスが少ないため，かつ検索時間が小さく，リンク切断の発生頻度が小さいためである．

図 3.3(b) より，提案手法は他の手法よりトラヒックが小さいことがわかる．これは，上記で説明したように，返信データ数が少ないためである．$v = 1$ および



(a) 検索精度

(b) トラヒック

(c) 検索時間

図 3.3: データ更新が発生しない環境における端末の移動速度の影響

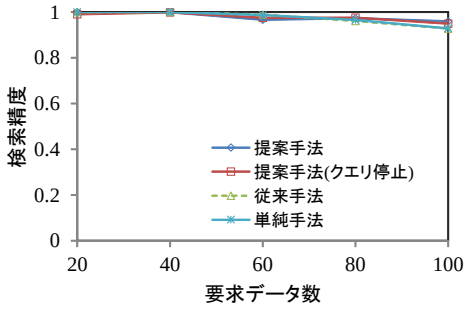
$v = 2$ のときに，単純手法のトラヒックが大きい．これは，$v = 0$ のときは，ネットワーク分断により返信データ数が少ない，および v が 2 より大きい場合では，リンク切断により返信データ数が減少するためである．

図 3.3(c) より，提案手法の検索時間が他の手法よりも小さい．これは，図 3.2(c) と同様の理由によるものである．$v = 0$ のとき，リンク切断が発生せず，全ての子端末から高い確率でクエリ応答を受信できるため，検索時間は短くなる．

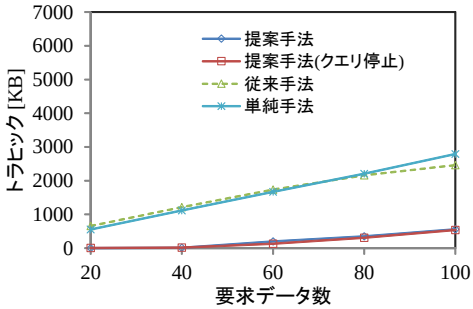
要求データ数の影響

次に，要求データ数 k の影響を調べた．その結果を図 3.4 に示す．これらのグラフにおいて，横軸は端末の要求データ数 k を表している．縦軸は，図 3.4(a) では検索精度，図 3.4(b) ではトラヒック，および図 3.4(c) では検索時間を表している．

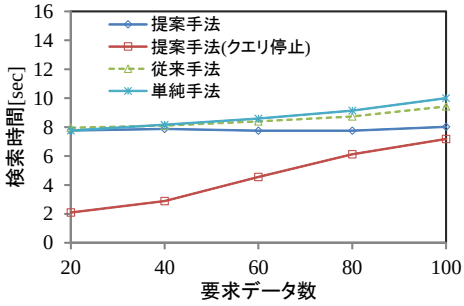
図 3.4(a) より，k が小さいとき，全ての手法の精度が高い．これは，各端末が自



(a) 検索精度



(b) トラヒック



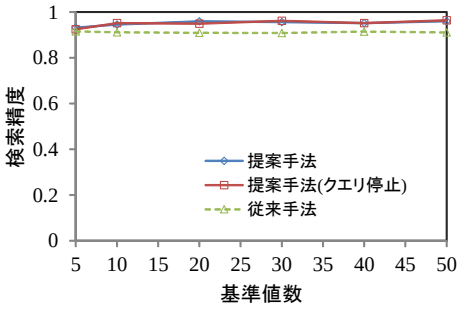
(c) 検索時間

図 3.4: データ更新が発生しない環境における要求データ数の影響

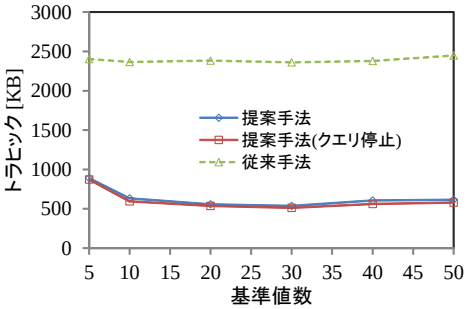
身の複製から上位 k 個のデータを取得できる可能性が高いためである．k が増加すると，自身の複製から上位 k 個を取得できない機会が増加することと，パケットロスにより，検索精度が低下する．

図 3.4(b) より，k が増加すると，全ての手法でトラヒックが増加する．これは，返信データ数が増加するためである．k が配置可能な複製数より小さい場合，クエリ発行端末は，自身の複製より検索結果を取得できることが多いため，提案手法のトラヒックはほぼ発生しない．

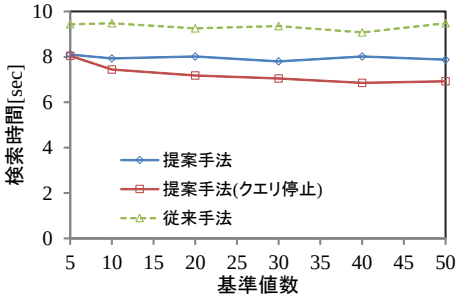
図 3.4(c) より，k が増加すると，全ての手法の検索時間は基本的に増加する．クエリ停止を用いた提案手法では，k が小さいとき，検索時間が他の手法に比べ非常に小さい．これは，他の手法では，検索クエリをネットワーク全体に送信するのに対し，クエリ停止により，クエリの送信を早めに停止できるためである．



(a) 検索精度



(b) トラヒック



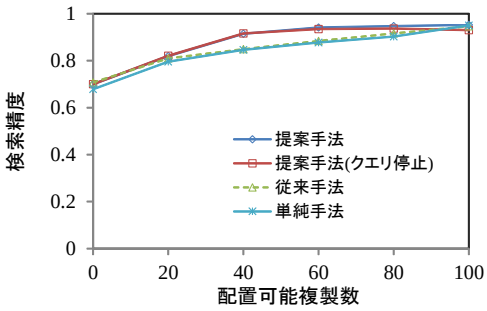
(c) 検索時間

図 3.5: データ更新が発生しない環境における基準値数の影響

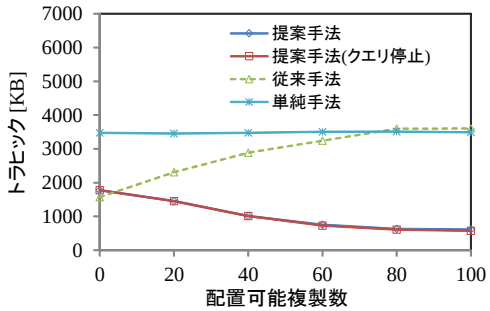
基準値数の影響

次に，基準値数 N の影響を調べた．その結果を図 3.5 に示す．これらのグラフにおいて，横軸は基準値数 N を表している．縦軸は，図 3.5(a) では検索精度，図 3.5(b) ではトラヒック，および図 3.5(c) では検索時間を表している．

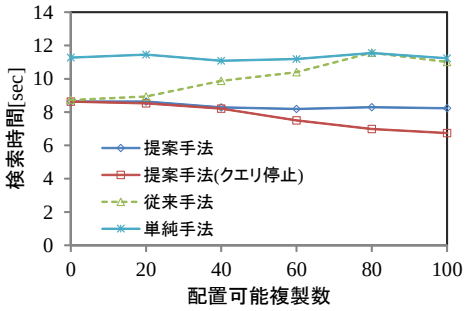
図 3.5 より，それぞれの手法の性能は基準値数にそれほど依存しないことがわかる．トラヒックでは，最初減少し，その後，徐々に増加するのがわかる．基準値数が小さいと，検索クエリのサイズが小さくなるが，返信データ数が増加する．一方，基準値数が非常に大きいと，検索クエリのサイズが大きくなるが，返信データ数がそれほど変化しない．その結果，基準値数が小さい場合と大きい場合にトラヒックが大きくなる．最適な基準値数は，要求データ数や，端末数，データサイズに依存する．クエリ停止を用いた提案手法では，基準値数が増加すると，若干検索時間が減少する．これは，返信データの候補の絞込み精度が上がり，クエリ停止が頻繁に発生するようになるためである．



(a) 検索精度



(b) トラヒック



(c) 検索時間

図 3.6: データ更新が定期的に発生する環境における配置可能複製数の影響

3.6.3　データ更新が発生する環境における評価結果

次に，データの更新が定期的に発生する環境における，シミュレーション結果を示す．

配置可能複製数の影響

配置可能複製数 ρ を変化させた時の提案手法の性能を調べた．その結果を図 3.6 に示す．これらのグラフにおいて，横軸は配置可能複製数 ρ を表している．縦軸は，図 3.6(a) では検索精度，図 3.6(b) ではトラヒック，および図 3.6(c) では検索時間を表している．

図 3.6(a) および (b) から，データが発生しない環境と似た特性を表していることがわかる．図 3.2(a) と比較して，複製の有効期限が切れてしまうため，検索精度が低くなっている．提案手法におけるトラヒックは，図 3.2(b) より大きくなって

いる，一方，従来手法では，小さくなっている．これは，提案手法では，有効な複製数が多い方が，遠くからの端末からのデータの返信および重複したデータの返信を抑制できるため，トラヒックが減少し，一方，従来手法では，重複した返信により，トラヒックが増加するためである．

図 3.6(c) から，提案手法における検索時間が図 3.2(c) より大きいことがわかる．これは，有効な複製数が減少するためである．特に，クエリ停止が有効に働かないことが多くなる．

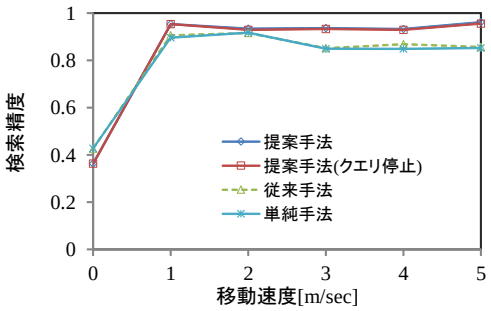
移動速度の影響

次に，端末の移動速度 v の影響を調べた．その結果を図 3.7 に示す．これらのグラフにおいて，横軸は端末の移動速度 v を表している．縦軸は，図 3.7(a) では検索精度，図 3.7(b) ではトラヒック，および図 3.7(c) では検索時間を表している．この結果から，図 3.7 は，図 3.3 と同様の傾向を示していることがわかる．

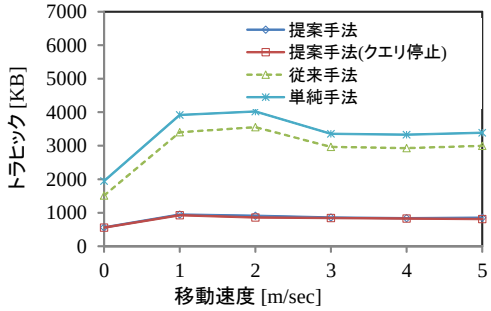
要求データ数の影響

次に，要求データ数 k の影響を調べた．その結果を図 3.8 に示す．これらのグラフにおいて，横軸は端末の要求データ数 k を表している．縦軸は，図 3.8(a) では検索精度，図 3.8(b) ではトラヒック，および図 3.8(c) では検索時間を表している．

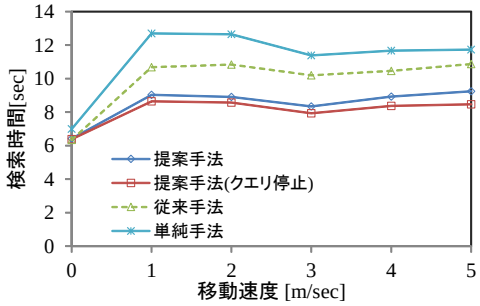
図 3.8(a) より，k が増加すると，クエリ停止を用いた提案手法を除いて，検索精度が減少する．k が小さいとき，クエリ停止を用いた提案手法の精度が低い．これは，上位のデータが削除された場合，ネットワーク全体を探す必要があるが，クエリ停止を用いた提案手法では，複製にスコアが高い（上位 k 個に入るとは限らない）データが配置されていることにより，検索クエリを不適切なタイミングで停止することが多くなるためである．k が増加すると，クエリ停止を用いた提案手法の検索精度が高くなる．これは，一部の端末のみで上位 k 個のデータを取得できる確率が大きくなることにより，検索クエリを適切なタイミングで停止できる機会が多くなるためである．k が大きいと，提案手法は，他の手法より高い検索精度を達成することができる．



(a) 検索精度



(b) トラヒック



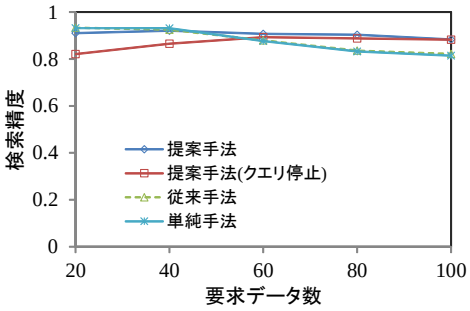
(c) 検索時間

図 3.7: データ更新が定期的に発生する環境における端末の移動速度の影響

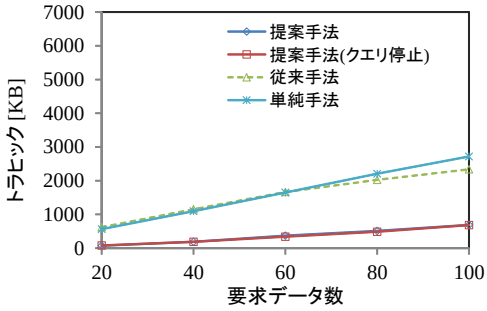
図 3.8(b) および (c) は，図 3.4(b) および (c) と基本的に同じ傾向を示している．k が小さいとき，クエリ停止を用いた提案手法の検索時間が，図 3.4(c) における値よりも大きい．これは，データ更新が発生する環境では，配置する複製を決定する際，更新時間を考慮するため，更新が発生しない環境より複製の多様性が増加し，クエリ停止をする機会が減るためである．

基準値数の影響

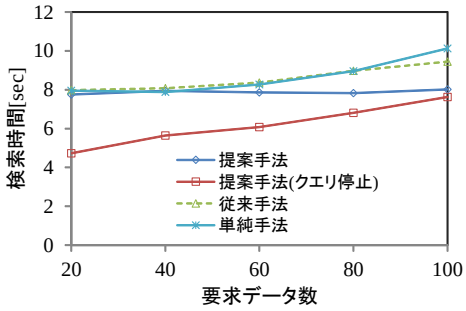
次に，基準値数 N の影響を調べた．その結果を図 3.9 に示す．これらのグラフにおいて，横軸は基準値数 N を表している．縦軸は，図 3.9(a) では検索精度，図 3.9(b) ではトラヒック，および図 3.9(c) では検索時間を表している．この結果から，図 3.9 は，図 3.5 と同様の傾向を示していることがわかる．



(a) 検索精度



(b) トラヒック



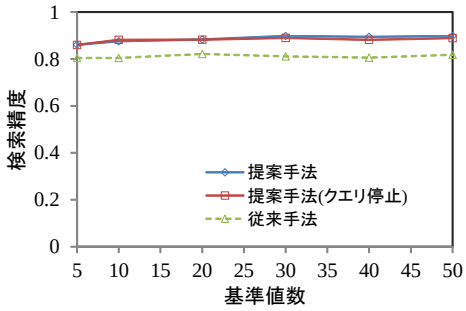
(c) 検索時間

図 3.8: データ更新が定期的に発生する環境における要求データ数の影響

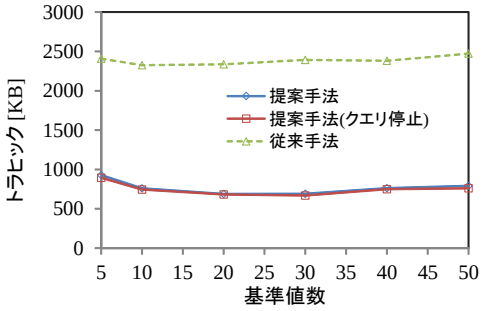
3.7 むすび

本章では，モバイルアドホックネットワークにおいて，配置されている複製を考慮した Top-k 検索手法を提案した．提案手法では，検索クエリメッセージに各端末のデータ保持情報を添付することにより，同じデータが複数回返信されたり，クエリ発行端末から遠い端末によりデータが返信されることを回避することにより，トラヒックを削減できる．また，一部の端末からのみで上位 k 個の全てのデータを既に収集可能であると判断した場合，クエリの伝搬を停止することにより，さらなるトラヒックの削減および遅延の削減が可能である．シミュレーション実験の結果から，提案手法により同じデータが複数回返信されることや長い経路上を返信されることを防げるため，トラヒックが削減するとともに，検索精度が向上することを確認した．

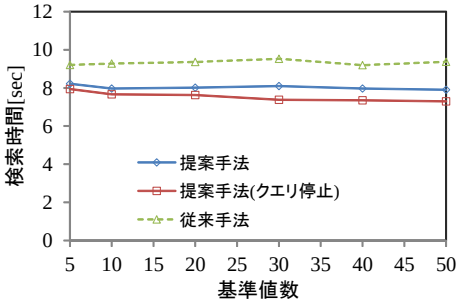
提案手法は，返信データ数を削減できるという長所の反面，複製を配置する機



(a) 検索精度



(b) トラヒック



(c) 検索時間

図 3.9: データ更新が定期的に発生する環境における基準値数の影響

会が少なくなる．そのため，ネットワーク分断が発生した場合に，必要な複製が十分に配置されておらず，検索精度が低くなってしまう可能性がある．そこで，今後はトラヒックの若干の増加は許容し，複製配置の機会を効果的に増やすためのデータ返信の方法を検討する予定である．

第4章　Top-k検索のための複製配置およびメッセージ処理手法

4.1　まえがき

モバイルアドホックネットワークにおけるTop-k検索処理では，限られた範囲の端末から上位k個のデータを取得することが理想的である．端末の位置が固定の場合，ある一定の範囲でデータを取得できるように複製を配置することが可能であるが，モバイルアドホックネットワークでは，端末が移動するため，検索の範囲を周囲の環境に従って動的に決定することが必要である．検索の範囲を決定する場合，各端末が複製の配置率を把握していれば，何台の端末を探索すれば，必要なデータを高確率で取得できるかを計算することができる．そこで，本章では，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，データの有用性を向上し，検索範囲を小さくすることが可能であるため，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．しかし，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，本章では，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．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，小さい遅延およびトラヒックで，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，4.2 節で想定環境と問題定義について述べ，4.3 節で関連研究について述べる．その後，4.4 節で提案手法について説明し，4.5 節でシミュレーション実験の結果を示す．最後に 4.6 節で本章のまとめと今後の課題について述べる．

4.2 想定環境と問題定義

4.2.1 想定環境

2 章および 3 章と同様に，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．

ネットワーク内には，D 個のデータが存在し，識別子 $\mathbf{d} = \{d_1, d_2, \cdots, d_D\}$ が割り当てられているものとする．本章では，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．

ネットワーク内には，m 台の端末が存在し，識別子 $\mathbf{M} = \{M_1, M_2, \cdots, M_m\}$ 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 (ρ, k_{max}] ($i = 1, \cdots, K$)，を kq_i の確率で指定する．各端末は，ある k が指定される確率を，統計情報などにより既知であるものとする．

4.2.2 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検索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．本章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．

本章では，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，$An(\mathbf{kq})$，を最少にする複製配置を行う．$An(\mathbf{kq})$ は以下の式で計算する．

$$An(\mathbf{kq}) = \sum_{i=1}^{K} kq_i \cdot An(k_i). \tag{4.1}$$

$An(k_i)$ 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．

4.2.3 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 [10]．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．

$$r_i = \frac{\sqrt{q_i}}{\sum_{j=1}^{k_{max}} \sqrt{q_j}} \text{ and } l \leq r_i \leq u. \tag{4.2}$$

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．

$$r_i \geq r_{i+1}. \tag{4.3}$$

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，遅延およびトラヒックが大きくなってしまう．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．

4.2.4 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，トラヒックが小さくなる．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．

$$cost = \Sigma_{x \in M} \Sigma_{i=1}^{K} k q_i \Sigma_{j=1}^{k_i} dist(x, d_j). \tag{4.4}$$

$dist(x, d_j)$ 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トラヒックが発生し，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トラヒックが必要ない手法，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．

4.3 関連研究

4.3.1 非構造型 Peer to Peer ネットワークにおける複製配置方式

非構造型 P2P ネットワークの分野では，各端末にデータの複製を配置することで，検索効率を向上させることを目的とした研究が数多く行われている [10, 32]. Lv らは，文献 [32] において，文献 [10] で提案されている平方根配置モデルの実現を目的とした複製配置方式として，オーナー複製法，パス複製法，ランダム複製法に関する考察を行っている．ここで，平方根配置モデルとは，各データのアクセス頻度に比例する数の複製をネットワーク全体に配置したモデルのことを示し，文献 [7] において最も検索効率が良い複製配置モデルであることが証明されている．これらの複製配置方式は，検索クエリがヒットした場合に複製を作成する．また，各端末において，複製を作成できる上限数を超えてしまった場合は，ランダム，FIFO(First In First Out)，もしくは LRU(Least Recently Used) に従って，保持する複製の中から 1 つを選択して削除する．

- オーナー複製法 : 検索クエリの発行端末がデータを受信した際に，その複製を配置する．
- パス複製法 : 検索クエリの発行端末から，要求データを保持する端末までに，検索クエリが経由したパス上にあるすべての端末にその複製を配置する．この手法は，オーナー複製法よりも平方根配置に近い複製数を実現できる．
- ランダム複製法 : 検索クエリの発行端末が要求するデータを保持する端末を発見するまでに検索クエリが経由したパス上にある端末の数と同じ数の複製を，検索クエリが伝播したすべての端末の中から，ランダムに選んで配置する．ネットワーク上に配置される複製の数はパス複製法と同じになる．

これらの研究では，データの識別子を指定して 1 つのデータにのみアクセスする検索を想定しているのに対し，本研究では Top-k 検索を想定している点で異なる．

4.3.2 モバイルアドホックネットワークにおける複製配置

モバイルアドホックネットワークの分野においても，複製配置やキャッシングに関する研究が数多く行われている [7, 9, 20, 22, 23, 31, 39, 44, 57, 58, 67, 71].

まず，モバイルアドホックネットワークにおける協調キャッシング手法について説明する．文献 [7, 31, 44, 71] では，端末がインターネットに接続可能な環境を想定したキャッシング方法を提案している．文献 [44] では，端末の消費電力の削減を目的としたキャッシング方法を提案している．この方法では，ルーティングプロトコルに ZRP(Zone Routing Protocol) [41] を利用し，端末がキャッシュしているデータを要求した端末と自身が要求したデータをキャッシュしていた端末の情報を保持する．また，文献 [71] では，効率的なデータアクセスを目的として，キャッシングデータ方式，キャッシングパス方式，およびハイブリッドキャッシュ方式を提案している．キャッシングデータ方式では，今後のデータ要求に用いるために，中継端末がデータをキャッシュする．キャッシングパス方式では，中継端末は，データをキャッシュするのではなく，データを保持する端末までの経路を記憶する．ハイブリッドキャッシュ方式では，上記 2 つの方式を組合せ，データのサイズが小さい場合，データをキャッシュし，一方，データのサイズが大きい場合，経路をキャッシュする．これらの研究はインターネットに接続可能な端末であるアクセスポイントが存在することを想定しており，アクセスポイントに接続できればすべての情報を取得できる．本研究ではアクセスポイントのような特別な端末は存在しないモバイルアドホックネットワークを想定している点で異なる．

文献 [61] では，記憶領域の制限を考慮して，ネットワーク全体で検索コストを減少させる，キャッシュ配置アルゴリズムを提案している．文献 [14, 15] では，隣接端末が保持するキャッシュの多様性を保つことを目的として，Hamlet と呼ばれる手法が提案されている．隣接端末が多様なデータをキャッシュすることにより，近くの端末からのみデータを取得できる可能性があがり，ネットワーク全体へのフラッディングを抑制することができる．文献 [15] では，各端末が，クエリメッセージと返信メッセージから，自律的にキャッシュするデータを決定する手法を提案しており，文献 [14] では，さらにデータのサイズも考慮している．これらの方法は 1 回の検索クエリで 1 つのデータを検索することを想定しているため，返信

データは常に 1 つである．そのため，周辺の端末が必要なデータを保持していないのを把握することができる．これにより，キャッシングするデータの多様性を保つことが比較的簡単である．一方，Top-k 検索では，複数の返信データを受信するため，どのデータをキャッシングすれば多様性を保つことができるかの判断が難しい．

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 [19] 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 [30] 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も上記の協調キャッシング手法と同様に，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．提案手法では，少ない探索端末数で k 個のデータを取得することを目的としているため，これらの手法を用いて複製データの配置を決定することは難しい．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大きな特徴である．

文献 [21] 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

う可能性や極端にスコアが大きいデータを全ての端末が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．

4.4 提案手法

本節では，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．

4.4.1 複製配置手法

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 $d_{\lceil\frac{k_{max}}{\rho}\rceil\cdot\rho-\rho+1}$ 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 $rc_i(i=1,\cdots,\lceil\frac{k_{max}}{\rho}\rceil)$ 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．

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．

$$n_h = ann \times \sum_{j=1}^{hop} j. \tag{4.5}$$

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．隣

接端末数が極端に多い場合，この式は正確ではないが，本研究では，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．

$$\begin{aligned} P(k_i, n_h) &= 1 - (\overline{rc_1} + \cdots + \overline{rc_{\lceil \frac{k_i}{\rho} \rceil}}) \\ &\quad + (\overline{rc_1 + rc_2} + \cdots + \overline{rc_{\lceil \frac{k_i}{\rho} \rceil - 1} + rc_{\lceil \frac{k_i}{\rho} \rceil}}) \\ &\quad + \cdots \\ &\quad + (-1)^{\lceil \frac{k_i}{\rho} \rceil} \cdot \overline{rc_1 + \cdots + rc_{\lceil \frac{k_i}{\rho} \rceil}}. \end{aligned} \tag{4.6}$$

ここで，$\overline{x}$ は，$(1-x)^{n_h}$ 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，全体として最少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 $An(\mathbf{kq})$，を次式で決定する．

$$An(\mathbf{kq}) = \sum_{i=1}^{K} (kq_i \cdot \frac{n_{h_i}}{P(k_i, n_{h_i})}) \tag{4.7}$$

$An(\mathbf{kq})$ 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，複製組み合わせ配置率を $An(\mathbf{kq})$ が最小となるように決定する．

アルゴリズム 1 に $An(\mathbf{kq})$ を決定する擬似コードを示す．アルゴリズム内における，δ は，rc の刻み幅を表すパラメータである．δ が小さければ小さいほど，正確な値になるが，計算量が大きくなる．与えられた rc において，$An(k_i)$ が最も小さ

Algorithm 1 FReT アルゴリズム

Input: ρ, ann, kq and δ
Output: rc
1: $rc_{num} = \lceil \frac{k_{max}}{\rho} \rceil$
2: $rc_i = \frac{1.0}{rc_{num}} (i = 1, \cdots, rc_{num})$
3: **while not** all patterns of rc are done **do**
4: **for** $i = 1, \cdots, K$ **do**
5: **for** $hop = 1, \cdots$ **do**
6: $n_{h_i} = ann \cdot \sum_{j=1}^{hop} j$
7: $An(k_i)_{hop} = \frac{n_{h_i}}{P(k_i, n_{h_i})}$
8: **if** $hop \neq 1$ **and** $An(k_i)_{hop} \geq An(k_i)_{hop-1}$ **then**
9: $An(k_i) = An(k_i)_{hop-1}$ and **break**
10: **end if**
11: **end for**
12: **end for**
13: $An(\mathbf{kq}) = \sum_{i=1}^{K} kq_i \cdot An(k_i)$
14: **if** $minAn > An(\mathbf{kq})$ **then**
15: $minAn = An(\mathbf{kq})$, and record rc
16: **end if**
17: Calculate next rc based on δ
18: **end while**

くなる hop を全ての i に対して求め，$An(\mathbf{kq})$ を求める（4–13 行）．全てのパターンの rc（ただし，$rc_i \geq rc_{i+1}$）に対して計算し，最小値を求める（17 行）．最終的に，最小の $An(\mathbf{kq})$ となる，rc を最適な複製配置率として，返信する．

4.4.2 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．収集のトラヒックをできる限り小さくするために，既存手法を用いて Top-k 検索を行う．例えば，文献 [38] で提案されている位置ベースフラッディングや，文献 [18] で提案されている Top-k 検索手法を組み合わせて，収集を行うことが考えられる．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 2 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．

4.4.3 Top-k 検索のメッセージ処理手法

本項では，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 3 およびアルゴリズム 4 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 (**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 3 内の 12–14 行，およびアルゴリズム 4 内の 30–32 行）．

Algorithm 2 初期収集方法

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

```
M_c broadcasts IM
if Node, M_r receives IM then
    if Receives first then
        Stores the information on IM
        Sets a query timer
    end if
    Updates SS and neighbor nodes
end if
if M_r expires its query timer then
    Calculates communication range of neighbor nodes //Proposed in [8]
    if Communication range of M_r is not covered then
        Broadcasts IM
        Sets a reply timer
    else
        Sends a reply message to its parent
    end if
end if
if M_r expires its reply timer and has no children or receives replies from its all
children then
    Sends a reply message to its parent
end if
if M_c receives replies from all its children then
    Starts distributing the collecting information by using the location-based flooding
else if max wait has passed from sending IM by M_c then
    M_c broadcasts IM again
end if
```

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ッ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，位置ベースフラッ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

Algorithm 3 初期検索クエリ

Input: A query condition and k
Output: 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: $\mathbf{rd} \leftarrow \mathbf{rd} \cup d_i$
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 $\mathbf{rd}$.
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 4
20: **end if**

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．

Expanding ring 法

P2Pネットワークにおける Expanding ring 法は，文献 [32] 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，トラヒック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．

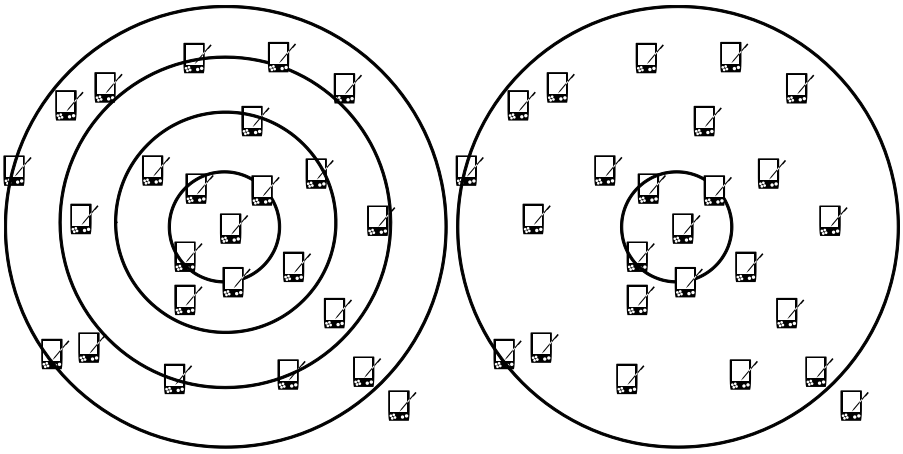
図4.1(a)に，Expanding ring 法におけるTTLの増加のイメージを示す．図内の円

Algorithm 4 再検索クエリ

Input: Identifiers of demand data items
Output: 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:     end if
22: end if
23: if M_r receives a reply message then
24:     for Reply data items rd_i are not sent to parent node, and are not overheard do
25:         rd ← rd ∪ rd_i
26:     end for
27:     if rd ≠ NULL then
28:         Sends a reply message to its parent
29:     end if
30: else if M_r overhears a reply message then
31:     Stores rd
32: end if
33: if M_q acquires the top-k result then
34:     Query is over
35: else if M_q expires a timer determined based on TTL then
36:     Updates RQ
37:     M_q broadcasts RQ again
38: end if
```

が TTL を表し，徐々に検索範囲を広がっていく．



(a) Expanding ring 法 (b) Bundling 法

図 4.1: Expanding ring 法と Bundling 法の TTL の違い

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．

$$cnn \times \sum_{j=1}^{TTL} j > \frac{n_{h_i}}{P(k_i, n_{h_i})}. \tag{4.8}$$

n_{h_i} および $P(k_i, n_{h_i})$ は，それぞれ最適な複製配置をにおける，式 (4.7) 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($\geq$ 2) を決定する．もし，クエリ発行端末が式 (4.8) 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．

図 4.1(b) に，Bundling 法における TTL の増加のイメージを示す．最初の再検索

クエリの TTL を一度に大きく広げている．

4.5 性能評価

本節では，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 [56] を用いた．

4.5.1 シミュレーション環境

1,000[m]×1,000[m] の 2 次元平面上の領域に m 台の端末（$M_1, \ldots, M_m$）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 v[m/秒] の速度で移動する（$v = 0$ のときは，停止する）．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，s[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 ($\rho = 5$).

端末は I 秒毎に Top-k 検索クエリを発行する．その際，2 種類のアクセスモデル（一様分布および Zipf-like 分布）に基づいて，k を 25，50，75，および 100 の中から指定する．一様分布では，全ての k を等確率 (25%) で指定し，Zipf-like 分布では，25，50，75，および 100 をそれぞれ 80%，10%，5%，および 5% で指定する．

表 4.1 に，シミュレーション実験に用いるパラメータをまとめる（太字はデフォルト値とする）．以上のシミュレーション環境において，クエリを 300 回発生させた場合（シミュレーション時間が $300 \times I$）の評価値を調べた．本章では，トラヒックとして，初期収集に発生するものと検索時に発生するものの 2 つを調べる．また，Expanding ring 法と Bundling 法の有効性を評価するため，検索時間も調べる．

- 検索精度
  ネットワーク全体の上位 k 個以内のスコアをもつデータのうち，検索クエリの発行端末が取得したデータ数の割合の全試行における平均.

- 検索時間
  クエリ発行端末がクエリを送信してから，検索結果を取得するまでの時間.

表 4.1: パラメータ

パラメータ	値
m（端末台数）	300,400,**500**,600,700,800
v（最大速度）	0,**1**,2,3,4,5
s（データサイズ）	32,**128**,256,512,1024
I（クエリ発行間隔）	5,10,20,**30**,60

- 初期トラヒック
  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．各メッセージのサイズを表 4.2 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含まれる端末の位置情報数を示す.

- 検索トラヒック
  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． 各メッセージのサイズを表 4.2 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.

4.5.2 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

表 4.2: メッセージサイズ

メッセージ	サイズ [B]
初期収集	288
初期収集返信	32+$s \cdot i_1 + 16 \cdot i_2$
初期収集 (単純手法)	44
初期収集返信 (単純手法)	28+$s \cdot i_1 + 16 \cdot i_2$
ACK (初期収集)	16
配布	44+$s \cdot 100$
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+$s \cdot i_4$
プッシュ (LACMA におけるメンテナンス)	32+$s \cdot i_5$

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トラヒックの凡例を 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．

提案手法を含む，全ての手法において，再送は行わず，各端末は 0〜0.5 秒待った後，返信を開始するものとした．

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 [10] 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）．

4.5.3 評価結果

初期収集および配布

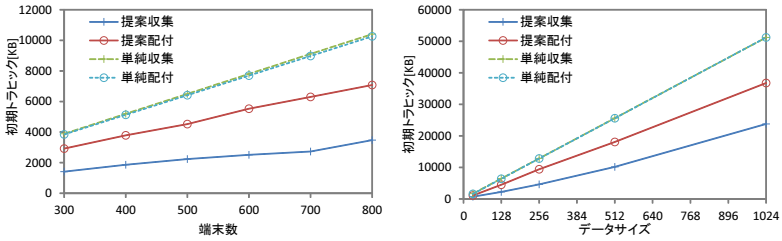
まず，初期収集および，初期配布手法の性能を端末数 m およびデータサイズ s を変化させて調べた．文献 [38] で提案されている位置ベースフラッディングおよび，文献 [18] で提案されている Top-k 検索手法を組み合わせて用いたものを「提案収集」と呼ぶ．文献 [38] で提案されている位置ベースフラッディングにより，収集した情報を全ての端末に配布するものを「提案配付」と呼ぶ．比較手法として，単純なフラッディングおよび自身の保持するデータと受信したデータのうち上位 k 個のデータを返信する単純手法（凡例; 単純収集，および単純配付）を用いる．

実験結果を図 4.2 に示す．このグラフにおいて，図 4.2(a) および (b) では，それぞれ端末数 m およびデータサイズ s を変化させた場合の，初期トラヒックを示す．この結果から，提案した初期収集および配布手法は，単純手法より低いトラヒック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，トラヒックが小さいのは，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．

初期収集における結果は，複製配置を用いない環境において，全ての端末にアクセスし，上位 100 個のデータを取得する Top-k 検索の結果と同様となる．

Top-k 検索処理

次に，Top-k 検索処理手法の性能調べた．



(a) Impacts of the number of nodes (b) Impacts of data size

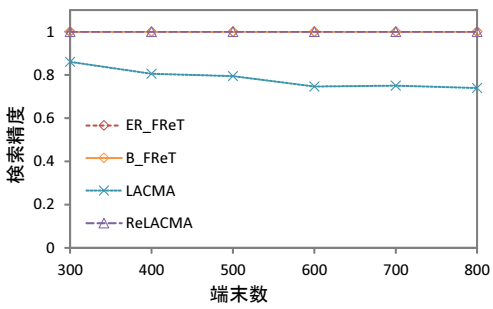
図 4.2: 初期収集および初期配付

端末数の影響．まず，端末数を変化させた場合の提案手法の影響を調べた．その結果を図 4.3 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 4.3 (a) では，検索精度，図 4.3 (b) では，検索時間，および図 4.3 (c) では，検索トラヒックをそれぞれ表す．

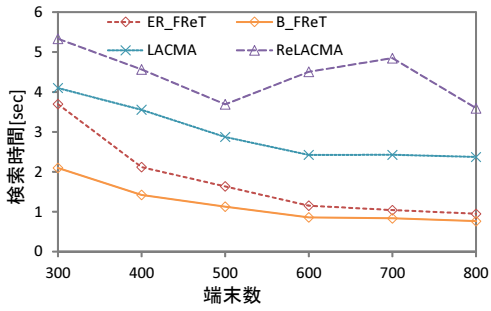
図 4.3 (a) より，提案手法は正確な結果を取得できていることがわかる．一方，再検索クエリを用いない LACMA では，正確な結果を取得できてない．これは，端末の移動，端末の疎密差，およびパケットロスが原因である．

図 4.3 (b) より，Bundling 法が最も小さい検索時間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 法も同様に小さい検索時間を達成している．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 法の検索時間の差が小さくなる．一方，LACMA では，提案手法より検索時間が大きい．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ってしまうためである．

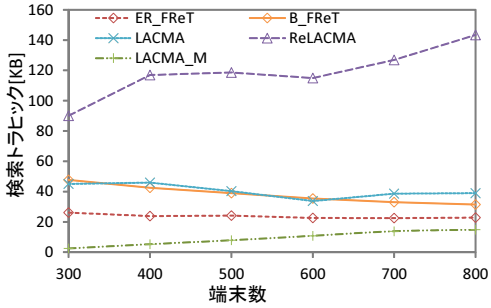
図 4.3 (c) より，Expanding ring 法が最も小さいトラヒックを達成しているのがわかる．これは，より近くの端末から上位 k 個のデータを取得できているためである．Bundling 法においても，正確な上位 k 個のデータを取得しながら，LACMA とほぼ同等のトラヒックを達成することができている．Reiterate LACMA は，非常に大きなトラヒックが発生しているのがわかる．これは，LACMA では，端末が



(a) 検索精度



(b) 検索時間



(c) 検索トラヒック

図 4.3: 一様アクセスにおける端末数の影響

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，メンテナンスのトラヒックも増加する．ここで，初期収集のトラヒックは，要求データ数 k が 100 の時の Top-k 検索の 1 回分のトラヒックとほぼ同様と考えることができる．図 4.2 の初期トラヒックと検索トラヒックを比較して，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トラヒックを考慮しても，ネットワーク全体として，トラヒック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，小さい検索時間およびトラヒックで，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．

端末速度の影響．端末の移動速度を変化させた場合の提案手法の影響を調べた．その結果を図 4.4 に示す．これらのグラフにおいて，横軸は端末の移動速度 v を表し，

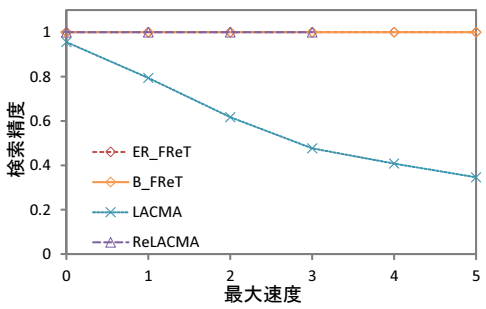
縦軸は，図 4.4 (a) では，検索精度，図 4.4 (b) では，検索時間，および図 4.4 (c) では，検索トラヒックをそれぞれ表す．

図 4.4(a) より，提案手法は，端末の移動速度が増加しても，正確な検索結果を取得していることがわかる．一方，LACMA は，速度が大きいと，多くの端末がグリッドから移動してしまうため，精度が低下していくことがわかる．

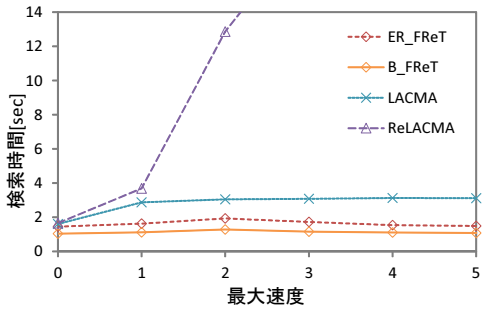
図 4.4(b) より，提案手法はLACMA よりいい性能を示している．これは，LACMA では，クエリ発行端末が正確な検索結果を取得できず，最大待ち時間まで待ってしまうためである．ReLACMA は，検索範囲が非常に広くなってしまうため，かなり大きな検索時間が発生してしまう．

図 4.4(c) より，提案手法の検索トラヒックは，速度に依存しないことがわかる．端末が停止しているとき ($v = 0$)，LACMA と Expanding ring 法の検索トラヒックが最小となる．LACMA では，地理的に等しく（つまり，理想的に）配置されているが，探索範囲がグリッドに限られてしまう．これにより，近接している端末でも，異なるグリッドに所属する場合，その端末からデータを取得することができなくなってしまう．一方，Expanding ring 法では，近くの少ない端末からの取得を可能としているため，トラヒックが小さくなる．結果として，データ取得のアプローチは異なるが，検索トラヒックはほぼ等しくなる．端末が移動すると，配置された複製の位置が変化するため，LACMA では，検索トラヒックが増加する．さらに，メンテナンスのためのトラヒックも増加する．特に，端末の速度が大きい場合，メンテナンスのためのトラヒックが検索トラヒックよりも大きくなってしまう．これより，LACMA は端末の移動に対する耐性が低いことがわかる．一方，提案手法は，端末の移動にロバストであるとこがわかる．

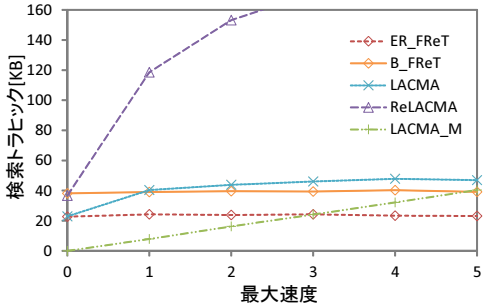
クエリ発行間隔の影響．クエリ発行間隔を変化させた場合の提案手法の影響を調べた．その結果を図 4.5 に示す．これらのグラフにおいて，横軸はクエリ発行間隔 I を表し，縦軸は，図 4.5 (a) では，検索精度，図 4.5 (b) では，検索時間，および図 4.5 (c) では，検索トラヒックをそれぞれ表す．図 4.5 より，提案手法は，I に関わらず，高い性能を維持していることがわかる．一方，LACMA では，I が大きくなるにつれて，性能が低下している．これは，クエリ発行間隔が長くなると，シミュレーション時間が長くなるため，グリッドから移動する端末が増加するため



(a) 検索精度



(b) 検索時間



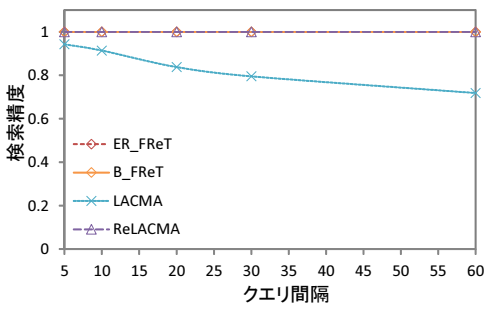
(c) 検索トラヒック

図 4.4: 一様アクセスにおける最大速度の影響

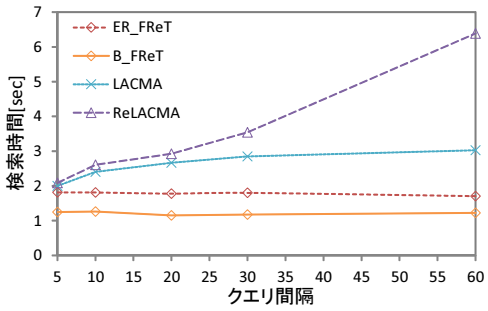
である．結果として，配置された複製を特定のグリッドに留まらせることが難しくなる．

データサイズの影響．データサイズを変化させた場合の提案手法の影響を調べた．その結果を図 4.6 に示す．これらのグラフにおいて，横軸はデータサイズ s を表し，縦軸は，図 4.6 (a) では，検索精度，図 4.6 (b) では，検索時間，および図 4.6 (c) では，検索トラヒックをそれぞれ表す．図 4.6(a) より，データサイズが増加すると，LACMA では精度が低下することがわかる．これは，データサイズ増加により，パケットロスが頻繁に発生するためである．

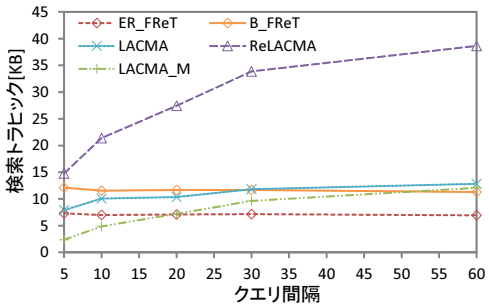
図 4.6(b) より，データサイズ増加に伴い，Bundling 法と ReLACMA の検索時間が大きくなることがわかる．これは，これらの手法では，再検索クエリのデータ数が多く，頻繁にパケットロスが発生してしまうためである．一方，Expanding ring 法では，検索時間はそれほど大きくならない．これは，Expanding ring 法は，返信データ数を小さくすることができ，パケットロスの発生頻度が小さいためである．



(a) 検索精度



(b) 検索時間



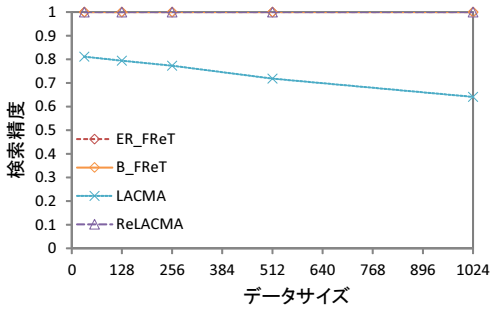
(c) 検索トラヒック

図 4.5: 一様アクセスにおけるクエリ発行間隔の影響

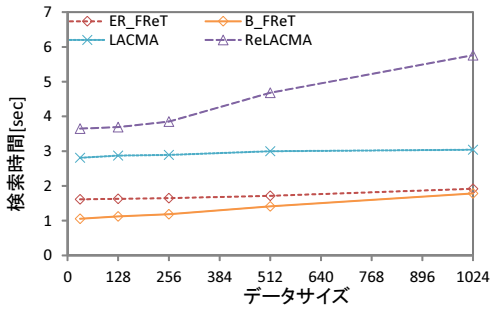
図 4.6(c) より，データサイズが増加すると，検索トラヒックも同様に増加する．特に，Expanding ring 法では，Bundling 法より，返信データ数が少ないため，トラヒックの増加が小さい．

アクセスモデル（Zipf-like 分布）の影響．端末数を変化させて，アクセスモデルが Zipf-like 分布に従う場合の提案手法の影響を調べた．その結果を図 4.7 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 4.7 (a) では，検索精度，図 4.7 (b) では，検索時間，および図 4.7 (c) では，検索トラヒックをそれぞれ表す．図 4.7(a) より，アクセスモデルが変化しても，提案手法は正確な検索結果を取得できていることがわかる．一方，LACMA では，アクセスモデルが一様分布に従う場合よりも，検索精度が低下している．これは，1 位から 25 位までのデータを留めるためのグリッドサイズが小さくなり，グリッドから多くの端末が移動してしまうためである．

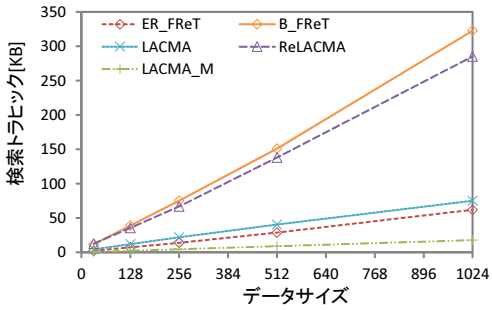
図 4.7(b) より，Expanding ring 法と Bundling 法の検索時間の差が小さくなってい



(a) 検索精度



(b) 検索時間

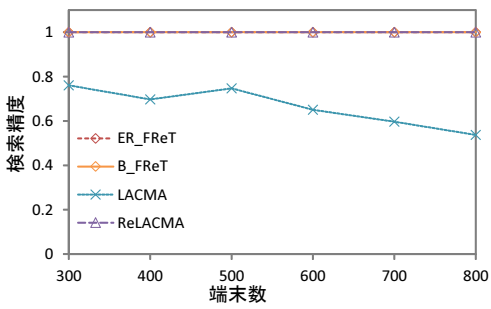


(c) 検索トラヒック

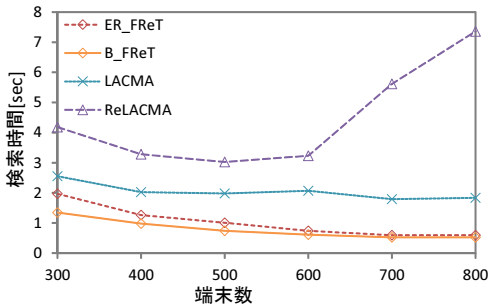
図 4.6: 一様アクセスにおけるデータサイズの影響

るのがわかる．これは，Zipf-like 分布では，k が小さいクエリが頻繁に発行されるため，1 ホップで必要なデータを取得できる可能性が高くなるためである．ReLACMA では，端末数が多い場合において，検索時間が大きくなっている．これは，端末数増加に伴い，グリッドサイズが小さくなり，端末がグリッドから頻繁に移動してしまうためである．

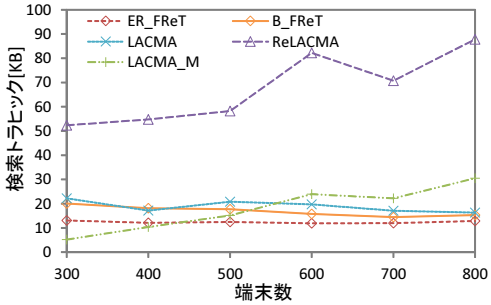
図 4.7(c) より，検索トラヒックが一様分布の場合に比べて，小さくなっているのがわかる．これは，小さい k が発行される確率が高いためである．提案手法における検索範囲が LACMA で設定したグリッドより小さくなることが多いため，LACMA より検索オーバーヘッドが小さい．また，LACMA では，グリッド間の端末の移動が多くなり，メンテナンスのためのトラヒックが大きく発生してしまう．これより，LACMA は，データのアクセスに大きな偏りがある場合に，有効に働かないことがわかる．提案手法は，データのアクセス分布に依らず，高い性能を維持することができる．



(a) 検索精度



(b) 検索時間



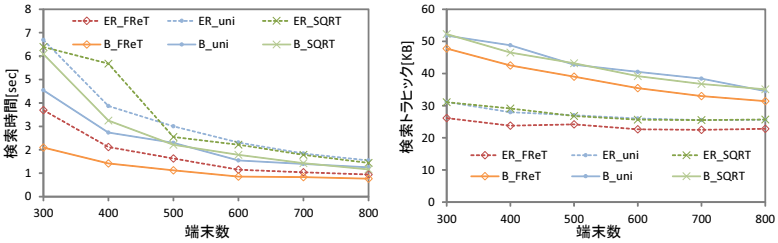
(c) 検索トラヒック

図 4.7: Zipf-like アクセスにおける端末数の影響

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．アクセスモデルが一様分布に従う場合の，結果を図 4.8 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 4.8 (a) では，検索時間，および図 4.8 (b) では，検索トラヒックをそれぞれ表す．

図 4.8 (a) より，FReT を用いた Bundling 法が最も小さい検索時間を達成していることがわかる．また，FReT を用いた Expanding ring 法も同様に小さい検索時間を達成している．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 4.8 (b) より，FReT を用いた Expanding ring 法のトラヒックが最も小さい．また，FReT を用いた Bundling 法も他の複製配置手法を用いた Bundling 法より小さいトラヒックを達成している．ここで，一様配置手法と平方根配置方式のトラヒックがほぼ同



(a) 検索時間　　(b) 検索トラヒック

図 4.8: 一様アクセスにおける複製配置手法の影響

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．

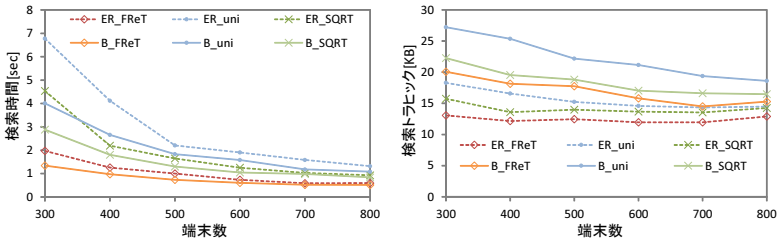
次に，アクセスモデルが Zipf-like 分布に従う場合の結果を図 4.9 に示す．これらのグラフにおいて，横軸は端末数 m を表し，縦軸は，図 4.9 (a) では，検索時間，および図 4.9 (b) では，検索トラヒックをそれぞれ表す．図 4.9 (a) より，FReT が最も小さい検索時間を達成していることがわかる．一様配置方式は，Zipf-like 分布では有効に働いていない．これは，スコアが低いデータ (25 位から 100 位のデータ) がアクセスされる可能性が低いにも関わらず，多く配置しすぎてしまい，再検索クエリが増加してしまうためである．

図 4.9 (b) より，FReT における Expanding ring 法の検索トラヒックが最も小さく，FReT における Bundling 法も同様に小さい検索トラヒックを達成できている．FReT では，隣接端末数を考慮して，複製配置率を決定しているため，小さいホップ数でデータを取得できる可能性が高くなっている．一方，平方根配置方式では，スコアが大きいデータを配置しすぎてしまうため，多くの重複したデータ返信が発生してしまう．

この結果より，FReT では，検索時間およびトラヒックの両方の観点において，どのようなアクセスモデルにおいても，高い性能を達成していることがわかる．

4.6 むすび

本章では，モバイルアドホックネットワークにおける，複製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



(a) 検索時間 (b) 検索トラヒック

図 4.9: Zipf-like アクセスにおける複製配置手法の影響

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，より小さい検索時間を達成した．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．

本章では，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配置手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．

第5章　結論

5.1　本論文のまとめ

本論文では, モバイルアドホックネットワークにおける Top-k 検索処理手法について議論した．まず，第 1 章では，モバイルアドホックネットワークにおいて，通信量を削減し，ネットワークトポロジに対応すること，および複製配置の利用の重要性について述べた．また，本論文における研究目的について述べ，本研究の位置づけを明らかにした．

第 2 章では，データ分布を推定することにより，返信データ数を削減する Top-k 検索手法を提案した．返信データ数を削減するには，正確な k 番目のスコアを求める必要があるが，検索クエリメッセージ転送の際は，一部の端末しか経由しないため，全ての端末が保持するデータの情報を取得することは困難である．そのため，一部の端末のデータからネットワーク全体の端末が保持するデータのスコアを推定することが求められる．本研究では，データのスコアがあるデータ分布に従っているものと想定し，このデータ分布の推定により，k 番目のスコアを求めることを試みた．具体的には，検索クエリメッセージにデータのスコアから作成されるヒストグラムを添付し，各端末はこのヒストグラムからネットワーク全体のデータが従う分布を推定する．分布の推定では，ヒストグラム内のスコア数を用いる方法，および確率密度関数を用いる方法を用いた．ヒストグラム内のスコア数を用いる方法では，ヒストグラムのそれぞれの値を，メッセージを経由した端末数とネットワーク全体の端末数から求める係数倍する．また，確率密度関数を用いる方法では，分布の概形を決定する要素（正規分布では，平均と分散）を計算する．推定した分布から k 番目のスコアを求め，そのスコア以上のスコアをもつデータのみを返信する．また，推定の誤差により，検索精度の低下を防ぐた

めに，推定した k 番目のスコアを補正するセーフティマージンも提案した．さらに，ネットワークトポロジの変化に伴う端末間のリンクが切断した場合，データを迂回路を用いて返信する．この方法により，従来の方法より，返信データ数を削減しつつ，検索精度を維持することが可能となった．

第 3 章では，データの複製を考慮して，Top-k 検索を実行する手法を提案した．複製配置手法として，既存研究である，データのスコアに応じて返信データを複製として配置するパス複製法を用いた．この既存研究の複製配置手法では，スコアが大きいデータが多くの端末に配置されてしまうため，重複した返信が多くなるという問題点があった．そこで，提案手法では，検索クエリメッセージにデータの識別子のリストを添付することにより，重複するデータの返信を抑制する．また，データの複製配置により，全ての端末にアクセスしなくても，正確な検索結果を取得できる可能性があるため，自律的に検索クエリメッセージの送信を停止し，データ返信を開始する方法も提案した．これにより，重複したデータの返信を抑制しつつ，検索結果取得までの遅延の削減も実現した．

第 4 章では，Top-k 検索のための効率的な複製配置手法およびメッセージ処理手法を提案した．既存の複製配置手法では，Top-k 検索にとって効率的な複製配置を実現することは難しい．そこで，各端末が保持するデータの組合せとデータを複製として配置する確率を決定する効率的な複製配置手法 FReT を提案した．FReT では，まず初期動作として，全ての端末の位置およびデータを収集する．これにより，上位のデータおよび平均隣接端末台数を知ることができる．そこから，上位 k 個のデータを取得するために必要な端末台数を最小化するよう，データの配置率を決定する．上位 k 個のデータを配布し，各端末はデータの配置率に従い，データを配置する．このとき，各端末は近い順位のデータをまとめて配置する．FReT により，上位 k 個のデータを近くの少ない端末から取得することが可能となる．一方，検索メッセージの転送処理方法として，Expanding ring 法と Bundling 法を提案した．Expanding ring 法では，徐々に検索範囲を広げていくことにより，アクセス端末数を最小とすることができる．Bundling 法では，検索範囲を隣接端末数に応じて，一度に広げることにより，検索遅延を削減することができる．提案した FReT，およびメッセージ処理手法は，既存の複製配置手法およびメッセージ処理

手法よりも，高い性能を達成した．

5.2 検討課題

本節では，モバイルアドホックネットワークにおける Top-k 検索のためのメッセージ処理および複製配置に関する今後の研究課題について述べる．

まず，本研究における Top-k 検索では，主に返信データ数を削減することを目的とした．データサイズが大きい場合，返信データの削減により，通信量を大きく削減することができるが，データサイズが小さい場合では，検索クエリの送信のトラヒック削減も重要となってくる．本研究の全ての提案手法では，検索クエリの送信方法は，基本的にフラッディングを用いるため，効率的とはいえない．フラッディングでは，上位 k 個のデータを取得するために必要のない端末にまでメッセージが送信されてしまう．そこで，Top-k 検索に適したルーティングやクラスタリングを用いることによる，効率的なメッセージ転送が求められる．例えば，ルーティングでは，上位 k 個のデータを保持する端末への経路を記憶することにより，無駄なクエリ転送を抑制することができる．一方，クラスタリングでは，クラスタ間で通信することにより，少数の端末のみにより，広い範囲に検索クエリを転送することができる．ルーティングにおける経路の管理や，クラスタの構築方法などの課題があるが，効率的な検索クエリの転送を行うことは，トラヒック削減において，非常に有効である．

本研究では，ネットワーク内に存在する端末は，全て同じ性能を有していると想定した．しかし，実際には，バッテリ残量や，配置可能な複製の数，端末の移動頻度などが異なる端末が混在している．そこで，端末の性能が不均一な環境における，効率的な複製配置を検討することは有益である．例えば，単純に，バッテリ残量が少ない端末には，アクセス頻度が小さいデータを複製として保持させることや，配置可能な複製数が多い端末に優先的にメッセージを送信することなどが考えられる．また，頻繁に移動する端末が各端末に複製を配付させることや，データ更新に関する情報を配布させ，複製の一貫性を保つことも可能である．

謝辞

本研究全般に関して，懇切なる御指導と惜しみない御助言を頂きました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 西尾章治郎教授に謹んで御礼申し上げます.

本研究を推進するにあたり，直接の御指導，御助言，御討論を頂きました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 原隆浩准教授に衷心より感謝申し上げます.

本論文をまとめるにあたり，大変有益な御指導と御助言を多数賜りました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 下條真司教授，細田耕教授に心より感謝申し上げます.

講義，学生生活を通じて，学問に取り組む姿勢をご教授頂きました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 藤原融教授，薦田憲久教授に厚く感謝申し上げます.

本研究において，ともに研究を進め，直接の御助言，御協力，御討論を頂いたペンシルバニア州立大学 Wang-Chien Lee 准教授に深く御礼申し上げます.

本研究において，多大なる御助言，御協力，御支援を頂きました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 寺西裕一准教授，大阪大学大学院工学研究科 春本要准教授，大阪大学サイバーメディアセンター 義久智樹准教授，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 神崎映光助教に深謝致します.

本研究において，ともに研究を進め，多大なる御協力を頂いた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 後藤啓介氏，駒井友香氏，天方大地氏，Kamalas Udomlamlert 氏，津田琢士氏， 大阪大学工学部電子情報工学科 Nguyen Duong Ho 氏に深く御礼申し上げます.

本研究を進める上で惜しみない御助言，御協力，研究活動を進めるにあたって

の多大なる御支援を頂いた松下電器産業株式会社 加治充氏，株式会社富士通研究所 篠原昌子博士，日本 IBM 株式会社 萩原亮氏，SCSK 株式会社 藤井俊充氏に感謝の意を表します.

本研究を進めるにあたり，多くの御討論や御助言を頂きました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 西尾研究室の諸氏に心より感謝申し上げます.

最後に，私のこれまでの人生，そして研究生活を送る上で，暖かい支援と理解を頂いた両親をはじめとする家族に心から感謝致します.

参考文献

[1] Akbarinia, R., Pacitti, E., and Valduriez, P.: Reducing network traffic in unstructured P2P systems using top-k queries, *IEEE Transaction on Distributed and Parallel Databases*, Vol. 19, No. 2, pp. 67–86 (2006).

[2] Akbarinia, R., Pacitti, E., and Valduriez, P.: Best position algorithms for top-k queries, in *Proceedings of International Conference on Very Large Data Base (VLDB)*, pp. 495–506 (2007).

[3] Babcock, B., and Olston, C.: Distributed top-k monitoring, in *Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 28–39 (2003).

[4] Balke, W.-T., and Kießling, W.: Optimizing multi-feature queries for image databases, in *Proceedings of International Conference on Very Large Data Base (VLDB)*, pp. 10–14 (2000).

[5] Balke, W.-T., Nejdl, W., Siberski, W., and Thaden, U.: Progressive distributed top-k retrieval in peer-to-peer networks, in *Proceedings of International Conference on Data Engineering (ICDE)*, pp. 174–185 (2005).

[6] Camp, T., Boleng, J., and Davies, V.: A survey of mobility models for ad hoc network research, *Wireless Communications and Mobile Computing*, Vol. 2, No. 5, pp. 483–502 (2002).

[7] Cao, G., Yin, L., and Das, C. R.: Cooperative cache-based data access in ad hoc networks, *IEEE Computer*, Vol. 37, No. 2, pp. 32–39 (2004).

[8] Chaudhuri, S., and Gravano, L.: Evaluating top-k selection queries, in *Proceedings of International Conference on Very Large Data Base (VLDB)*, Vol. 99, pp. 397–410 (1999).

[9] Chen, K., and Nahrstedt, K.: An integrated data lookup and replication scheme in mobile ad hoc networks, in *Proceedings of International Symposium on the Convergence of Information Technologies and Communications (ITCom)*, pp. 1–8 (2001).

[10] Cohen, E., and Shenker, S.: Replication strategies in unstructured peer-to-peer networks, in *Proceedings of ACM SIGCOMM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, Vol. 32, pp. 177–190 (2002).

[11] Das, G., Gunopulos, D., Koudas, N., and Tsirogiannis, D.: Answering top-k queries using views, in *Proceedings of International Conference on Very Large Data Base (VLDB)*, pp. 451–462 (2006).

[12] Fagin, R.: Combining fuzzy information from multiple systems, *Journal of Computer and System Sciences*, Vol. 58, No. 1, pp. 83–99 (1999).

[13] Fagin, R., Lotem, A., and Naor, M.: Optimal aggregation algorithms for middleware, *Journal of Computer and System Sciences*, Vol. 66, No. 4, pp. 614–656 (2003).

[14] Fiore, M., Casetti, C., and Chiasserini, C.: Caching strategies based on information density estimation in wireless ad hoc networks, *IEEE Transaction on Vehicular Technology*, Vol. 60, No. 5, pp. 2194–2208 (2011).

[15] Fiore, M., Mininni, F., Casetti, C., and Chiasserini, C.: To cache or not to cache?, in *Proceedings of International Conference on Computer Communications (INFOCOM)*, pp. 235–243 (2009).

[16] Güntzer, U., Balke, W.-T., and Kiesling, W.: Optimizing multi-feature queries for image databases, in *Proceedings of International Conference on Very Large Data Base (VLDB)*, pp. 419–428 (2000).

[17] 萩原亮, 篠原昌子, 原隆浩, 西尾章治郎 : アドホックネットワークにおける Top-k 検索のためのメッセージ処理手法, 情報処理学会論文誌, Vol. 51, No. 1, pp. 152–162 (2010).

[18] Hagihara, R., Shinohara, M., Hara, T., and Nishio, S.: A message processing method for top-k query for traffic reduction in ad hoc networks, in *Proceedings of International Conference on Mobile Data Management (MDM)*, pp. 11–20 (2009).

[19] Hara, T.: Effective replica allocation in ad hoc networks for improving data accessibility, in *Proceedings of International Conference on Computer Communications (INFOCOM)*, pp. 1568–1576 (2001).

[20] Hara, T.: Replica allocation methods in ad hoc networks with data update, *ACM/Kluwer Journal on Mobile Networks and Applications (MONET)*, Vol. 8, No. 4, pp. 343–354 (2003).

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

[22] Hara, T., Loh, Y.-H., and Nishio, S.: Replica allocation in ad hoc networks considering the stability of radio links, *IPSJ Jornal*, Vol. 44, No. 9, pp. 2308–2319 (2003).

[23] Hayashi, H., Hara, T., and Nishio, S.: Updated data dissemination methods for updating old replicas in ad hoc networks, *ACM/Splinger Personal and Ubiquitous Computing*, Vol. 9, No. 5, pp. 273–283 (2005).

[24] Hristidis, V., Koudas, N., and Papakonstantinou, Y.: PREFER: a system for the efficient execution of multi-parametric ranked queries, *ACM SIGMOD Record*, Vol. 30, No. 2, pp. 259–270 (2001).

[25] IEEE 802.11, The Working Group for Wireless LANs
: *http://grouper.ieee.org/groups/802/11/*.

[26] Ilyas, I. F., Beskales, G., and Soliman, M. A.: A survey of top-k query processing techniques in relational database systems, *ACM Computing Survey*, Vol. 40, No. 4, p. 11 (2008).

[27] Johnson, D. B.: Routing in ad hoc networks of mobile hosts, in *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, pp. 158–163 (1994).

[28] Kalnis, P., Ng, W. S., Ooi, B. C., and Tan, K.-L.: Answering similarity queries in peer-to-peer networks, in *Proceedings of International World Wide Web Conference (WWW)*, pp. 482–483 (2004).

[29] Kalnis, P., Ng, W. S., Ooi, B. C., and Tan, K.-L.: Answering similarity queries in peer-to-peer networks, *Information Systems*, Vol. 31, No. 1, pp. 57–72 (2006).

[30] Lee, S., Wong, S., Lee, K., and Lu, S.: Content management in a mobile ad hoc network: beyond opportunistic strategy, in *Proceedings of International Conference on Computer Communications (INFOCOM)*, pp. 266–270 (2011).

[31] Lim, S., Lee, W.-C., Cao, G., and Das, C.: A novel caching scheme for internet based mobile ad hoc networks, in *Proceedings of International Conference on Computer Communications and Networks (ICCCN)*, pp. 38–43 (2003).

[32] Lv, Q., Cao, P., Cohen, E., Li, K., and Shenker, S.: Search and replication in unstructured peer-to-peer networks, in *Proceedigns of International Conference on Supercomputing*, pp. 53–65 (2002).

[33] Marian, A., Bruno, N., and Gravano, L.: Evaluating top-k queries over web-accessible databases, *ACM Transactions on Database Systems*, Vol. 29, No. 2, pp. 319–362 (2004).

[34] 松波秀和, 寺田努, 西尾章治郎 : P2P 型コンテンツ検索システムのための効率的な Top-k 検索手法, 情報処理学会論文誌, Vol. 47, No. 9, pp. 2850–2859 (2006).

[35] Michel, S., Triantafillou, P., and Weikum, G.: KLEE: a framework for distributed top-k query algorithms, in *Proceedings of International Conference on Very Large Data Bases (VLDB)*, pp. 637–648VLDB Endowment (2005).

[36] Mobile Ad-Hoc Networks (manet) Charter
: *http://www.ietf.org/html.charters/manet-charter.html/*.

[37] Nepal, S., and Ramakrishna, M.: Query processing issues in image (multimedia) databases, in *Proceedings of International Conference on Data Engineering (ICDE)*, pp. 22–29 (1999).

[38] Ni, S., Tseng, Y., Chen, Y., and Sheu, J.: The broadcast storm problem in a mobile ad hoc network, in *Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pp. 151–162 (1999).

[39] Nuggehalli, P., Srinivasan, V., and Chiasserini, C.-F.: Energy-efficient caching strategies in ad hoc wireless networks, in *Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 25–34 (2003).

[40] Padhariya, N., Mondal, A., Goyal, V., Shankar, R., and Madria, S.: EcoTop: an economic model for dynamic processing of top-k queries in mobile-P2P networks, in *Proceedings of International Conference on Database Systems for Advanced Applications (DASFAA)*, pp. 251–265 (2011).

[41] Pearlman, M., and Haas, Z.: Determining the optimal configuration for the zone routing protocol, *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, pp. 1395–1414 (1999).

[42] Perkins, C. E., and Ooyer, E. M.: Ad-hoc on-demand distance vector routing, in *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, pp. 90–100 (1999).

[43] Ryeng, N. H., Vlachou, A., Doulkeridis, C., and Nørvåg, K.: Efficient distributed top-k query processing with caching, in *Proceedings of International Conference on Database Systems for Advanced Applications (DASFAA)*, pp. 280–295 (2011).

[44] Sailhan, F., and Issarny, V.: Cooperative caching in ad hoc networks, in *Proceedings of International Conference on Mobile Data Management (MDM)*, pp. 13–28 (2003).

[45] 佐々木勇和, 萩原亮, 原隆浩, 篠原昌子, 西尾章治郎：アドホックネットワークにおけるヒストグラムを用いた Top-k 検索手法, 情報処理学会マルチメディア通信と分散処理ワークショップ (DPSWS) 論文集, pp. 13–18 (2009).

[46] 佐々木勇和, 原隆浩, 西尾章治郎 : アドホックネットワークにおける 2 フェーズ Top-k 検索手法, データ工学と情報マネジメントに関するフォーラム (DEIM), CD-ROM (2010).

[47] 佐々木勇和, 原隆浩, 西尾章治郎：アドホックネットワークにおけるデータの複製配置を考慮した Top-k 検索手法, 情報処理学会マルチメディア通信と分散処理ワークショップ (DPSWS) 論文集, pp. 1–6 (2010).

[48] 佐々木勇和, 原隆浩, 西尾章治郎：アドホックネットワークにおけるデータ数と値を考慮した Top-k 検索, 情報処理学会マルチメディア，分散，協調とモバイル (DICOMO) シンポジウム論文集, pp. 1121–1128 (2010).

[49] 佐々木勇和, 原隆浩, 西尾章治郎：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, データ工学と情報マネジメントに関するフォーラム (DEIM), CD-ROM (2010).

[50] 佐々木勇和, 原隆浩, 西尾章治郎 : アドホックネットワーク上のスコア分布推定による Top-k 検索手法, 情報処理学会論文誌, Vol. 52, No. 1, pp. 175–186 (2011).

[51] 佐々木勇和, 原隆浩, 西尾章治郎：モバイルアドホックネットワークにおける Top-k 検索のための位置情報を用いた複製配置手法, FIT2013 第 12 回情報科学技術フォーラム, pp. 383–386 (2013).

[52] 佐々木勇和, 原隆浩, 西尾章治郎：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, データベース学会論文誌, Vol. 12, No. 1, pp. 115–120 (2013).

[53] Sasaki, Y., Hagihara, R., Hara, T., Shinohara, M., and Nishio, S.: A top-k query method by estimating score distribution in mobile ad hoc networks, in *Proceedings of International Workshop on Data Management for Wireless and Pervasive Communications (DMWPC)*, pp. 944–949 (2010).

[54] Sasaki, Y., Hara, T., and Nishio, S.: Two-phase top-k query processing in mobile ad hoc networks, in *Proceedings of International Conference on Network-Based Information Systems (NBiS)*, pp. 42–49 (2011).

[55] Sasaki, Y., Hara, T., and Nishio, S.: Top-k query processing for replicated data in mobile peer to peer networks, *Systems and Software* (To appear).

[56] Scalable Network Technologies: Creators of QualNet Network Simulator Software: *http://www.scalable-networks.com*.

[57] 篠原昌子, 林秀樹, 原隆浩, 西尾章治郎：アドホックネットワークにおける消費電力を考慮したデータ複製配置方法, 情報処理学会論文誌, Vol. 47, No. 1, pp. 15–27 (2006).

[58] 篠原昌子, 原隆浩, 西尾章治郎：モバイルアドホックネットワークにおける消費電力を考慮したデータ複製へのアクセス方式, 情報処理学会論文誌, Vol. 48, No. 12, pp. 3931–3948 (2007).

[59] Silberstein, A., Braynard, R., Ellis, C., Munagala, K., and Yang, J.: A sampling-based approach to optimizing top-k queries in sensor networks, in *Proceedings of International Conference on Data Engineering (ICDE)*, p. 68 (2006).

[60] Silberstein, A., Munagala, K., and Yang, J.: Energy-efficient monitoring of extreme values in sensor networks, in *Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 169–180 (2006).

[61] Tang, B., Gupta, H., and Das, S.: Benefit-based data caching in ad hoc networks, *IEEE Transaction on Mobile Computing*, Vol. 7, No. 3, pp. 289–304 (2008).

[62] Tao, Y., Hristidis, V., Papadias, D., and Papakonstantinou, Y.: Branch-and-bound processing of ranked queries, *Information Systems*, Vol. 32, No. 3, pp. 424–445 (2007).

[63] 寺田努, 松波秀和, 西尾章治郎 : P2P 型コンテンツ検索システムにおけるコンテンツ分布を考慮した Top-k 検索処理手法, 情報処理学会論文誌, Vol. SIG14(TOD 35), No. 48, pp. 1–10 (2007).

[64] The Official Bluetooth Wireless Info Site: *http://www.bluetooth.com/*.

[65] Vlachou, A., Doulkeridis, C., and Nørvåg, K.: Distributed top-k query processing by exploiting skyline summaries, *IEEE Transaction on Distributed and Parallel Databases*, Vol. 30, No. 3-4, pp. 239–271 (2012).

[66] Vlachou, A., Doulkeridis, C., Nørvåg, K., and Vazirgiannis, M.: On efficient top-k query processing in highly distributed environments, in *Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 753–764 (2008).

[67] Wang, K., and Li, B.: Efficient and guaranteed service coverage in partitionable mobile ad hoc networks, in *Proceedings of International Conference on Computer Communications (INFOCOM)*, pp. 1089–1098 (2002).

[68] Wu, M., Xu, J., Tang, X., and Lee, W.-C.: Monitoring top-k query in wireless sensor networks, in *Proceedings of International Conference on Data Engineering (ICDE)*, p. 143 (2006).

[69] Wu, M., Xu, J., Tang, X., and Lee, W.-C.: Top-k monitoring in wireless sensor networks, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 7, No. 19, pp. 962–976 (2006).

[70] Xie, M., Lakshmanan, L. V., and Wood, P. T.: Efficient top-k query answering using cached views, in *Proceedings of International Conference on Extending Database Technology (EDBT)*, pp. 489–500 (2013).

[71] Yin, L., and Cao, G.: Supporting cooperative caching in ad hoc networks, in *Proceedings of International Conference on Computer Communications (INFOCOM)*, pp. 2537–2547 (2004).

[72] Zeinalipour-Yazti, D., Andreou, P., Chrysanthis, P. K., and Samaras, G.: MINT views: materialized in-network top-k views in sensor networks, in *Proceedings of International Conference on Mobile Data Management (MDM)*, pp. 182–189 (2007).

[73] Zhao, K., Tao, Y., and Zhou, S.: Efficient top-k processing in large-scaled distributed environments, *Data & Knowledge Engineering*, Vol. 63, No. 2, pp. 315–335 (2007).

[74] ZigBee Alliance: *http://www.zigbee.org/*.

[75] Zipf, G. K.: Human behavior and the principle of least effort., *addison-wesley* (1949).