

無線センサネットワークシステムにおける分散化フィルタリング機構に関する一考察

中村 達也[†] 中野 美由紀[‡] 喜連川 優[‡]

[†] 東京大学 大学院 情報理工学系研究科 電子情報学専攻 〒153-8505 東京都目黒区駒場 4-6-1

[‡] 東京大学 生産技術研究所 〒153-8505 東京都目黒区駒場 4-6-1

E-mail: [†] nakamura@tkl.iis.u-tokyo.ac.jp, [‡] {miyuki, kitsure}@tkl.iis.u-tokyo.ac.jp

あらまし 近年、デジタル情報の爆発的な増加に対し、それを支える情報化社会の IT 基盤の一つとして、無線センサネットワークシステムが着目されている。無線センサネットワークシステムでは、多量のセンサからのデータを効率的に、収集、整理、管理、利用することが必要となる。センサの技術的な進歩は急速であるが、一方で設置されるセンサ数、収集されるデータ量の爆発的な増加に対し、センサノードには未だに、電力やバンド幅、メモリなどの物理的な制限が存在している。高効率な無線センサネットワークシステムの実現を目指し、収集されたデータの Join 処理におけるセンサノードの転送量削減や、特定のノードへの過負荷を避ける効率的なルーティングなどの研究がされてきている。本研究では、無線センサネットワーク上のデータ転送量を削減するアルゴリズムとして提案された Continuous Join Filter 機構(CIF)を取りあげ、CIF のフィルタリング情報変更処理をベースステーション上からセンサノード上分散化することによって得られる転送量削減効果について考察する。

キーワード 無線センサネットワーク, フィルタリング機構, 分散化, ジョイン処理, 転送量削減

An Preliminary Investigation of Distributed Filtering System in Wireless Sensor Networks System

Tatsuya Nakamura[†] Miyuki NAKANO[‡] and Masaru KITSUREGAWA[‡]

[†] Graduate School of Information Science and Technology with a major in Information and Communication Engineering, The University of Tokyo 4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505 Japan

[‡] IIS. The University of Tokyo 4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505 Japan

E-mail: [†] nakamura@tkl.iis.u-tokyo.ac.jp, [‡] {miyuki, kitsure}@tkl.iis.u-tokyo.ac.jp

Abstract Recently, wireless sensor network systems (WSNS) are focused on as one of IT infrastructure systems to support modern information society, where digital information explosively increases day by day. An efficient WSNS are necessary to collect, arrange, manage and utilize large amount of sense data. The capability of sensor is improved rapidly, a wireless sensor node, however, still has many physical constraints such as limitation of battery power, low bandwidth, small main memory and so on while collected sensor data and number of sensors increases explosively. Therefore researchers study join processings of collected data to realize high performance WSNs systems. Recent join processing methods are to reduce transmissions and to find efficient routing path and so on. In this research, we take Continuous Join Filter system (CJF) as an algorithm for reducing data transmissions on WSNs and consider potential performance of reducing transmissions by distributing modification of filtering parameters amongst some nodes of WSNS.

Keyword WSNs, Filtering System, Distribution, Join Processing, Reducing Transmissions

1. はじめに

近年、様々な場所にセンサが設置され、分析や制御などに利用されている。無線通信機能を持つ情報通信端末の小型化、低コスト化が飛躍的に進み、無線センサネットワークの研究が多数行われている。電子タグや IC チップ、無線センサノードを多数用いることで、大規模かつ高解像度のオンライン分散計測システムを構築でき、現代社会における IT 社会基盤として期待されている。すでに、IBM の

smart planet プロジェクト[1]や Google の powermeter[2]等、実運用も図られている。これらの実用例に示されるように、無線センサネットワークから得られる大規模なデータを効率的かつ永続的に統合・管理・利用する技術が求められ、無線センサネットワーク上のデータベース技術への注目が高まっている。

しかし、無線センサノードの性能は向上してきているものの、大規模データベース等処理負荷の高いシステムは、既存の技術を単純に応用しても、ネットワークバンド幅や電力などの物理的制限が多

く、無線センサネットワークに適用可能なデータベース技術が必要となる。中でも、データベース処理で最も負荷が高く、頻繁に用いられる join 処理の研究が近年多く報告されている。各センサノードからデータベースが置かれたベースステーションに転送され、突き合わせ(マッチング)処理が行われる。しかし、単純に全データを送信すると、転送コスト(送信、受信によるノードのリソースの消費)がかかる。そこで、join 処理の突き合わせ(マッチング)対象となるデータのみを送れば、各センサノードの転送コストを抑えられる。マッチング処理となるデータをどのように選択的に転送するか(あるいは不要なデータを転送しないか)、効率的な join 処理手法の研究が多くされている。[3,4,5,6,7]

本稿では、転送コストを削減する方法として、M Stem ら[3]の CJF(Continuous Join Filter)というフィルタリング機構を取り上げ、ノードセンサ数が爆発的に増えた場合に問題となるフィルタリング条件更新に着目し、この機構を分散化させたときの join 処理における各ノードの削減転送量についての評価を行う。本稿の構成は、まず第2章で関連研究、第3章でCJF、第4章で分散フィルタリング機構、第5章で基本評価、最後にまとめと今後の課題を述べている。

2. 関連研究

センサDBのjoin処理における転送量を減らす研究が多くなされている。[4,5,6,7]

まず、H.Yu ら[4]は、Synopsis という関数を用い、各センサノード上のデータからクエリで指定されたデータストリーム情報を抽出する。抽出された情報に用いて join 処理を行い、その結果から tuple を送るべきノードを絞り込む。この手法は、対象となる tuple 属性が多くなると抽出された結果が大きくなり、適切に必要な tuple を抽出することが出来ない可能性がある。

次にM.Stem ら[5]の手法では、static な属性の join 条件を先に処理することで、送信するべきノードを減らす Base join に加え、ベースステーションにおいて、semi-join を行い、さらに対象ノードを減らして tuple の送信をする。semi-join とは join 条件に含まれる属性の値をベースステーションに送り、その属性に対してのみ join を行い、条件に合致するノードだけ、全属性を含む tuple を送るようにする操作である。

また、X.Yang ら[6]の手法では、static な attribute の value を用いて、センサノードの relation を分散化させている。分散化することで、部分的に join 処理を行えるため、転送量が抑えられている。しかしながら、static な attribute の value に依存してしまっているため、クエリに多くの active な attribute が関わると精度が落ちてしまう。また、大きな relation には不適合であることも問題になっている。

最後に、M.L.Yiu[7]らの手法では、Yang らの分散化手法で用いられていた static attribute value の代わりに、各センサノード間の距離をパラメータに用いて、relation を分散化することで、各センサノードの通信範囲内において、部分的に join 処理を行っている。

3. Continuous Join Filter (CJF)

M Stem ら[3]は Continuous Join Query に対して、Continuous Join Filter(以下 CJF)というフィルタリングアルゴリズムを提案している。この手法は、温度センサ等連続的にデータを送るノードに対し、データ転送範囲を指定したフィルタを各センサ上に設定することで、データの取捨選択を行い、余分な転送量を削減する。

CJF は、あるノードから転送されるデータの連続値が、他ノードの値と多くの場合において join しないとき、ある値幅のフィルタをノードに設けることで、センシングした値がフィルタ内であれば、ノードのデータは join に必要無いと考え、データの転送量を削減することができる。しかし、ベースステーションがフィルタリングされなかった全データを統合した際に、フィルタリングされたノードのデータに対応したデータが存在した場合、join に必要になる可能性が生じる。その際、ベースステーションはノードに再送要求を出し、データを転送させる。このように、通常の転送よりも多くの転送量が必要となってしまう処理が起きてしまうことを、collision が起きるといふ。そして、ベースステーションは、全データを統合したのち、データからフィルタリング値の調整と更新を行うことで、効率良く転送量を削減できるフィルタを維持している。

3.1 CJF 処理の流れ

操作 1. 各センサはフィルタリングされなかった場合にのみ、データをベースステーションに転送

操作 2. ベースステーションはフィルタリングされたデータが、本当に join しないかを確認。join する場合は、センサに転送命令を転送(collision)

操作 3. ベースステーションは、全センサの前データと与えられたクエリを基に、最適なフィルタを計算

操作 4. ベースステーションは、転送コストの比較を行い、フィルタを更新するかどうかを決定

CJF では1~4.の操作を繰り返している。

3.2 最適なフィルタサイズ計算

フィルタは、転送コストを最小化するように設定される。ここでいう、転送コストとは、フィルタを通過した場合の予測転送コストと、操作 2(collision)が起きた場合に必要とする予測転送コストで構成される。フィルタリングされる確率、他ノードと join する確率を用いている。

また、join するセンサ同士がフィルタリングによってデータ転送されてしまった場合、正しい join 結果を得ることができないため、フィルタは転送コストを最小に抑え、且つ、フィルタ同士の衝突をさけるための条件を加え、最適なフィルタサイズを計算している。

3.3 フィルタ更新

新しいフィルタに更新するためにも、転送量が必要となるため、変化の少ない属性を持つフィルタである場合、常に更新し続けると、かえってコストが高くなってしまいます。そのため、更新の決定の際に、

charge コストと blocking コストを考慮し、更新の可否を行う。

charge コストは、現フィルタのときと新フィルタのときの予測転送コスト差を用いている。また、最後にフィルタを更新した時間と現在の時間の時間差を加味し、古いフィルタを長時間利用する懸念を避けている。

blocking コストは、フィルタ更新をしなかった場合に、他センサの予測転送コストがどれだけ影響を受けるかを数値化したものである。他センサへの影響を考慮することにより、あるセンサの新フィルタと他センサの現フィルタが衝突し、正しい join 結果を落としてしまう可能性を取り除いている。

CJF では、フィルタの中心を前データ値を用いているため、センシングした属性値が大きく変化し続けるときには、フィルタリングされずに、余計な転送が行われるだけでなく、フィルタを頻繁に更新しなければいけない問題がある。

4. 分散化フィルタリング機構

CJF では、フィルタサイズの計算・更新のすべてをベースステーションが処理している。しかしながら、この手法だと、フィルタを更新しなければいけないセンサに対し、常にベースステーションから、新しいフィルタ情報をセンサに転送する必要が出てきてしまう。センサノード数が増えると更新再計算・更新コストは甚大なものとなる。そこで、本稿ではベースステーションで行っているフィルタ更新処理を分散化する手法について検討する。

提案する分散化フィルタリング機構では、図1のように、全センサに対し、位置情報、routing path 等を考慮して、いくつかのグループに分け、グループ毎に最もベースステーションに近いノードを中間ノードに設定する。各グループ内のセンサは、中間ノードにデータを転送することで、中間ノードは、グループ内の全センサデータを保持することができる。そのため、ベースステーションで処理されるべきフィルタサイズの計算・更新を、各中間ノードがグループ内のセンサデータを用いることで、同様の処理が可能になる。もし、中間ノードにおいて、グループ内の全センサデータを用いて、計算したフィルタサイズが、ベースステーションにおいて、全センサデータを用いて、計算されたフィルタサイズと同じであれば、ベースステーションから更新する全センサに、新しいフィルタ情報を転送する代わりに、同様のフィルタ情報を中間ノードからセンサに転送できる。このような場面では、ベースステーションは中間ノードに対して、中間ノードが計算したフィルタサイズをセンサに転送しろという命令を転送するだけで済むため、更新における転送量を削減することができると考えられる。

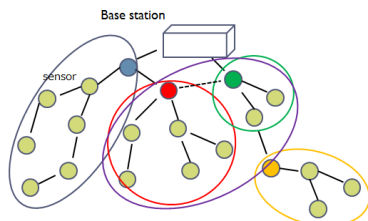


図 1: 分散化におけるグルーピング例

この分散化フィルタリング機構を適用することで、以下の二つの転送量をターゲットにできる。

4.1. フィルタ更新

これは上記のように、フィルタ更新時における新しいフィルタ情報の転送を各グループの中間ノードがベースステーションの代わりに行うことで、転送量を削減できる。

加えて、グループ内だけでなく、フィルタ更新時におけるグループ間の相関を用いることで、更なる転送量の削減が可能になる。CJF のアルゴリズムでは、あるセンサのフィルタ更新の影響で、他のセンサもフィルタを更新することになる場合がある。このような状況は、多くの場合、センサ同士、もしくはグループ同士の相関が多く見受けられる。そのため、フィルタ更新において、図1の点線のように、相関の強いグループの中間ノード同士でデータの受け渡しを可能にし、フィルタサイズ計算を行うことで、相関の強いグループを、局所的に一つのグループとしてみなせ、上記と同様に転送量を削減できると考えられる。

4.2. collision

提案機構により、collision が起きた際の、転送量も削減できる。collision とは、クエリに合致するデータを持つセンサが、フィルタリングによって、データ転送されなかった場合に、ベースステーションがセンサに対して、再送命令を転送し、データを転送させなければいけない操作である。従来の方法では、ベースステーション内にある全センサのフィルタ情報を用いて、ベースステーションが再送の可否を判断する。これに対し、分散化フィルタリング機構では、グループ内のセンサのフィルタ情報は中間ノードに保持されているため、フィルタリングされたセンサの再送の可否を、部分的に中間ノードで判断することができる。中間ノードで再送命令を転送できれば、ベースステーションから再送命令を転送する必要がなくなり、転送量を削減することができる。

5. 基本評価

この研究の目的は、効率的なフィルタリングアルゴリズムにより、転送量の削減、また、ノードが過負荷にならないような効率的なルーティングパスを提案することで、ノードのリソース使用量を抑えることにある。そこで、今回の実験では、Micro Stem らの CJF[3] のフィルタリングアルゴリズムを実装し、シミュレーションをした場合、どれほど転送量を削減出来るかのフィルタリング効果、及び分散化フィルタリング機構を適用した場合の性能評価を行う。

5.1 シミュレーション環境

5.1.1 データセット

データセットとして LUCE データセット[9]を用い、88 個のノードでネットワークを構築している。LUCE データセットはローザンヌ工科大学において、2006/11/01~2007/05/31 までの半年間の環境モニタリングデータを提供しているデータセットである。無線センサ

には mica2、mica2dot mote を利用している。気象データをセンシングした sensorscope-meteo と、センサの configuration をセンシングした sensorscope-monitor の二つのテーブルに分かれている。それぞれの持つ属性は以下の通り。

1.気象データ

「sensor ID, year, month, day, hour, minute, second, sequence number, ambient temperature, surface temperature, solar radiation, relative humidity, soil moisture, watermark, rain meter, wind speed, wind direction」の 17attributes を 5~30 秒毎にセンシングしたデータ。全 44463498 records。

2.configuration データ

「sensor ID, year, month, day, hour, minute, second, time since the epoch, sequence number, config sampling time, data sampling time, radio duty cycle, radio transmission power, radio transmission frequency, primary buffer voltage, solar panel current, global current, energy source」の 18 attributes を 5 分毎にセンシングしたデータ。4726643 records。

また、図2のように、GPS から得た各ノードの固定的な位置情報もデータセットに含まれている。

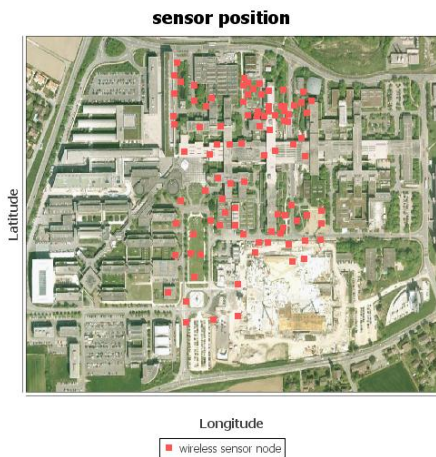


図 2: sensor position

5.1.2 ルーティンググループ

今回の実験データでは、routing 情報等がなかったため、位置情報のみを用いてグループ分けを行った。位置情報と各ノードの距離から、K-means、階層型クラスタリング(以下、階層型)、さらに数が少ないことと、ビル上に多く配置されていることから、建物ベースの視認により、位置情報に微調整を行ったもの(以下、建物ベース)の3つの方法でグループ分けを行った。また、それぞれのグループ分け手法において、比較するグループ数は4~7個としている。図3はグループ分け例として、K-means を用いて、全センサを位置情報で、6つにグループ分けした結果である。

グループを決定したのちに、全センサのルーティングを決定する。分散化フィルタリング機構では、グループ内において、最もベースステーションに近いセンサノードを中間ノードとする。そして、グループ内の他センサは、中間ノードまでのルーティングを設定する。ここで、センサの通信範囲は75m までとしている。一方、中間ノードは、ベースステーションが通信範囲の75m 以内に無い場合、よりベースステーションに近い、他グループのセンサに通信し、ベース

ステーションまで、最小ホップ数で通信できるルーティングを設定する。表1は、それぞれの手法、グループ分けでルーティング設定したときの、全センサの平均ホップ数である。

今回のシミュレーションでは比較対象を、転送量にしている。ここでいう、転送量は1ホップを1転送量としてカウントしている。

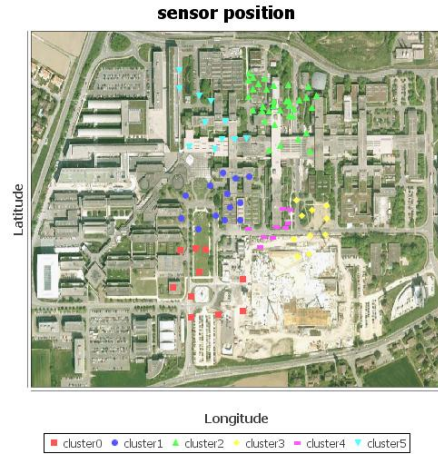


図 3: K-means(グループ数:6)

手法	グループ数	平均ホップ数
K-means	4	2.7
	5	2.83
	6	2.795
	7	2.93
階層型	4	2.75
	5	2.784
	6	2.715
	7	2.75
建物ベース	4	3.397
	5	2.67
	6	2.716
	7	2.806

表 1: 平均ホップ数比較

5.1.3 クエリと比較対象

全てのシミュレーションにおいて、以下のクエリを用いる。

```
SELECT A.*, B.*
FROM Sensors A, Sensors B
WHERE |A. humidity - B. humidity| < 2 %
AND |A. temperature - B. temperature| < 2 °C
AND A. id != B. id
SAMPLE PERIOD 30s
```

このクエリはセンサが自分以外のノードで、湿度の差が2%未満、温度差が0.3°C未満を満たすノードが存在すれば、その tuple 同士を結合する操作を30秒毎に行うためのクエリである。

今回のシミュレーションでは比較対象を、このクエリで問い合わせたときの転送量にしている。ここでいう、転送量は1ホップを1転送量としてカウントしている。

5.2 シミュレーション結果

フィルタリングアルゴリズムの性能評価を行う。

5.2.1 CJF

まず、CJF を適用した場合の、異なる3つのグループ分けにおけるフィルタリング結果を示す。

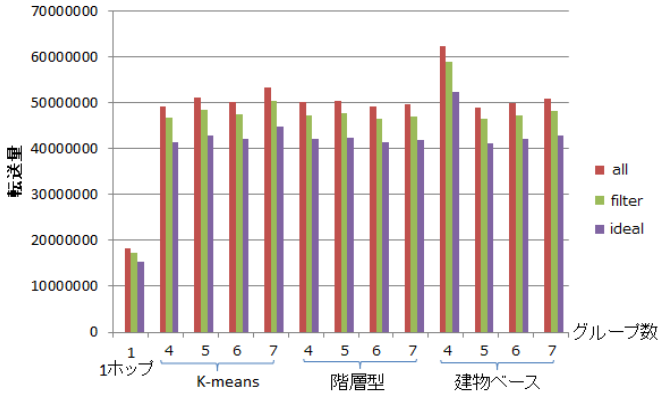


図 4: CJF 結果

図4において、allは全センサが全データを転送したときの転送量、filterはCJFを適用したときの転送量、idealはクエリに必要なセンサだけがデータを転送した場合の理想的な転送量である。allとidealの差分が、無駄になっている転送量である。そして、filterはCJFの定義より、センサのデータ転送、フィルタの更新、collisionが起こった場合の余分な転送、これらの転送量によって構成されている。また、横軸はそれぞれのグルーピング手法、グループ数を表している。

図4の結果では、参考のために全ノードを1ホップ、1グループでベースステーションと通信できる場合の結果を、左に示している。1ホップと比べると他の結果は、平均ホップ数の違いにより全体の転送量は大きくなっているものの、どのグルーピング手法、グループ数においても、1ホップと同様に、無駄な転送量の内、30%~40%を削減できているため、CJFの性能はルーティングの影響に大きく依存していないことが分かる。もちろん、より効率的なルーティングを設定することで、無駄な転送量を少なくできると考えられる。

5.2.2 分散化フィルタリング結果

次に、さらに無駄な転送量を削減するために、分散化フィルタリング機構を適用した場合の結果を示す。

分散化フィルタリング機構では、グループ内の中間ノードにおいてのフィルタ計算・更新に加え、相関が強いグループ同士は、中間ノードを通信できるようにし、ひとつのグループとして扱うことで、転送量の削減を図っている。そのため、あらかじめ複数のセンサのフィルタが同時に更新されたときに、どのグループ内、もしくは、どのグループ間で同時に更新されているかの頻度をヒストグラム化し、グループ間の相関を知る必要がある。例として、K-meansを用いて6つのグループにグルーピングした場合のヒストグラムを以下の図5に示す。

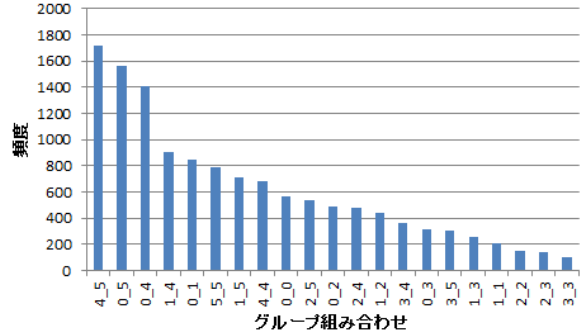


図 5: グループ間の相関(K-means:6)

図5は、縦軸が頻度、横軸がフィルタ更新に関わっているグループの組み合わせである。例えば、横軸の4_5はグループ4とグループ5のみが、フィルタ更新に関わっているときの頻度を表している。図5からも分かるように、フィルタ更新において、特定のグループ間で強い相関がある。これは、他のグルーピング手法、グループ数においても同様に表れている。

今回のシミュレーションでは、相関が強い三つのグループグループ間で互いにフィルタリング情報を共有できるように分散化フィルタリング機構を適用している。例えば、図5では、グループ0,4,5において、中間ノードが通信できるように設定する。

分散化フィルタリング機構において、フィルタ更新時に削減できる転送量を表2に示す。表2の[更新転送量]、[グループ内]、[グループ間]はそれぞれ、CJFのフィルタ更新に必要なとする転送量、提案機構を適用した際に、グループ内のみのフィルタ計算・更新で処理したときの削減量、グループ間の相関を用いて、フィルタ計算・更新で処理したときの削減量を表す。表2より、更新転送量の内、分散化フィルタリング機構を適用することで、グループ内、グループ間における処理によって、転送量の約20%強を削減することができる。今後、無線センサネットワークに用いられるセンサ数が莫大になればなるほど、分散化によって、削減できる転送量は多くなると考えられる。

手法	グループ数	更新転送量	グループ内	グループ間
K-means	4	103366	17533	5299
	5	107604	19016	5129
	6	105811	19187	4566
	7	114795	20584	5453
階層型	4	104531	18697	5405
	5	106225	18553	5190
	6	104551	17929	5199
	7	105844	17485	5217
建物ベース	4	135880	19778	5756
	5	105599	19345	5377
	6	106849	19287	5137
	7	107505	18851	5135

表 2: 分散化フィルタリング結果(フィルタ更新)

図6は、図5のグループ間の相関における、相関と削減転送量の

関係を表したものである。分散化フィルタリング機構において、利用するグループ間の相関を、頻度が高い組み合わせから、順に増やしていったときの削減転送量の変化を赤線(右軸)に表す。

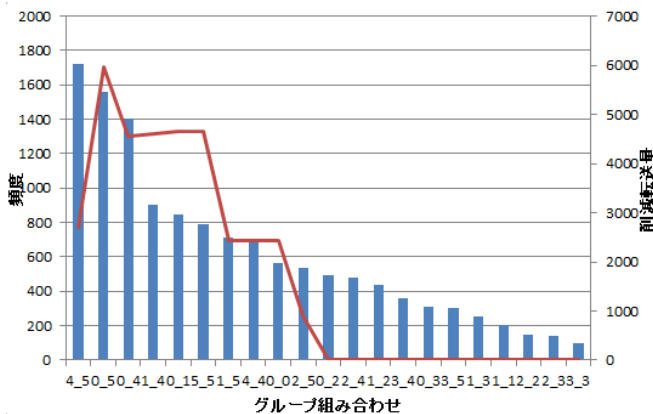


図 6: 相関と削減転送量の関係

この結果を見ると、グループ0_5の組み合わせまでを利用したときに、最も削減転送量を得られている。これは、中間ノード同士のルーティングによるものと考えられる。中間ノード間のルーティングが上手くいってないと、データの受け渡しにおいて、余計な転送量が増えてしまう。そのため、より相関を有用に用いるためには、ルーティング面の改善が必要である。

最後に、分散化フィルタリング機構を適用したことで collision が起きた際に削減できる転送量の結果を表 3 に示す。

手法	グループ数	削減量
K-means	4	88112
	5	91573
	6	47827
	7	88590
階層型	4	92283
	5	89921
	6	89550
	7	88743
建物ベース	4	112684
	5	92302
	6	92305
	7	92229

表 3: 分散化フィルタリング結果(collision)

表 3 から、分散化フィルタリング機構を用いることで、collision の際にも、約 10 万程度の転送量を削減できることが分かる。

分散化フィルタリング機構による削減量は、グルーピング手法による精度に影響される部分があるものの、グループが少なければ、少ないほど、ひとつのグループに属するセンサが多くなるため、グループ内で削減できる転送量の割合は大きくなると考えられる。しかしながら、ひとつのグループに属するセンサが多いほど、中間ノードの負担が大きくなってしまいう問題が生じる。今回の結果では、グルーピング手法、グループ数によって、削減できる転送量が大きく変化することが少ない。そのため、中間ノードにかかる負担を低減しつつ、なるべく多くの転送量を削減できるモデルを、様々な属

性、より多くのセンサを用いたデータセットを利用し、提案できれば、効率が良い無線センサネットワークが実現できると考えられる。

6. まとめ

本研究では、Micro Stem らによって提案された CJF を元に、中間ノードを用いた分散化フィルタリング機構を提案した。そして、分散化フィルタリング機構を LUCE データセットに適用して、フィルタ更新時におけるグループ内、グループ間の相関を利用した際、及び、collision が起きた際における転送量の削減を検証した。

今後の課題としては、様々なデータセット、クエリに対して、分散化フィルタリング機構の挙動を検証することが挙げられる。フィルタリングの性質上、データセットやクエリに非常に依存する特性があるため、莫大なセンサ数で構成される無線センサネットワークのデータセットなどに適用し、本機構の性能がセンサ数に対して、どのような特性を持つかを検証する必要と考えられる。また、ルーティング面など、分散化フィルタリング機構と組み合わせることで、より積極的に転送量を削減できる手法の提案が必要である。

文 献

- [1] <http://www.ibm.com/smarterplanet/us/en/>
- [2] <http://www.google.com/powermeter/about>
- [3] Micro Stern, Klemens Bohm, Erik Buchmann, Processing Continuous Join Queries in Sensor Networks: a Filtering Approach, SIGMOD 2010
- [4] Hai Yu, Ee-Peng Lim, jun Zhang, On In-network Synopsis join Processing for Sensor Networks, MDM2006
- [5] Mirco Stern, Erik Buchmann, Klemens Bohm, Towards Efficient Processing of General-Purpose Joins in Sensor Networks, ICDE2009
- [6] X. Yang, H. B. Lim, M. T. Ozsu, and K. L. Tan. In-Network Execution of Monitoring Queries in Sensor Networks. SIGMOD, 2007.
- [7] M. L. Yiu, N. Mamoulis, and S. Bakiras. Retrieval of Spatial Join Pattern Instances from Sensor Networks. SSDBM 2007.
- [8] X. Zhu, H. Gupta, and B. Tang. Join of Multiple Data Streams in Sensor Networks. IEEE TKDE, 99(1), Jan. 2009.
- [9] <http://sensorscope.epfl.ch/index.php/EnvironmentData>
- [10] Sam Madden, Joe Hellerstein, and Wei Hong, TinyDB: In-Network Query Processing in TinyOS, <http://telegraph.cs.berkeley.edu/tinydb/tinydb.pdf>
- [11] Samuel R.Madden, Michael J.Franklin, Joseph M.hellerstein and Wei Hong, TinyDB: An Acquisitional Query Processing System for Sensor Networks, ACM Transactions on Database Systems, Vol.30,No.1, March 2005, Pages 122-173
- [12] N.Jain, D.Kit, P.Mahajan, P.Yalagandula, M.Dahlin, Y.Zhang, STAR:Self-Tuning Aggregation for Scalable Monitoring, VLDB2007