

大規模センサデータストリーム収集のための位相調整を用いた負荷均等化手法の検討

川上 朋也¹ 石 芳正² 義久 智樹³ 寺西 裕一^{4,3}

概要: IoT が実現されつつあり、多種多様なセンサをもつ機器（ノード）がインターネット経由でデータを連続的に発信している。このとき、処理能力やネットワーク環境はノードごとに異なるため、ノード側が求めるデータの生成や発信の間隔も異なる。本研究では各ノードが連続的かつ定期的に発信するデータを「センサデータストリーム」、発信の間隔を「周期」と呼ぶ。我々は複数のノードが発信するセンサデータストリームを収集する仕組みとして、オーバーレイネットワークの一つであるスキップグラフを用いて複数のストリームを集約することで、各ノードのデータ送受信による負荷を軽減する手法を提案した。しかし、集約可能なストリームの数には上限が考えられ、既存手法では同じ周期のノードが多く存在する場合、特定の時刻に負荷が集中してしまう。そこで本研究では、特定の時刻への負荷集中を軽減するため、位相調整を用いた負荷均等化手法を提案する。提案手法ではセンサデータの収集周期に加えて、各ノードへ個別の時間差（位相差）を与える。提案手法はシミュレーションにより評価し、位相調整を行わない場合と比べて各ノードの瞬間的な負荷の最大値（最大瞬間負荷）が減少することを確認した。

1. はじめに

さまざまなモノがインターネットにつながる Internet of Things (IoT) [1] が実現されつつあり、カメラによる映像なども含め、多種多様なセンサをもつ機器（ノード）がインターネット経由でデータを連続的に発信している。このとき、処理能力やネットワーク環境はノードごとに異なるため、図 1 のように、ノード側が求めるデータの生成や発信の間隔も異なる。本研究では各ノードが連続的かつ定期的に発信するデータを「センサデータストリーム」、発信の間隔を「周期」と呼ぶ。IoT では膨大な数のノードやデータ、それらの利用者が存在する。そのため、高いスケーラビリティを維持しつつ、対象とする周期的なセンサデータを効率的に収集可能な仕組みが求められる。

膨大な数のデータやユーザを収容可能な高いスケーラビリティの実現手法として、分散ハッシュテーブル (Distributed Hash Table, DHT) や地理的な情報に基づくオーバーレイネットワークなどの構築手法が提案されている [2-9]。オー

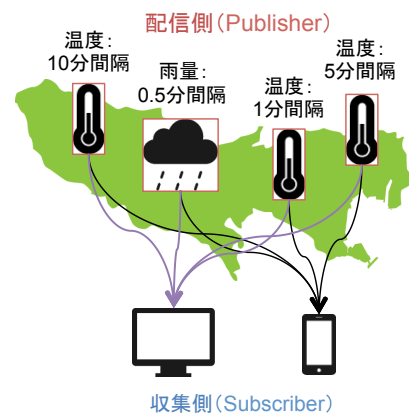


図 1 異なる間隔でのセンサデータストリーム収集

バレイネットワーク構築手法では複数のノード間に仮想リンクを構築し、データや処理を複数のノードが分担することで特定のノードやネットワークへの負荷集中を軽減する。我々は複数のノードが発信するセンサデータストリームを収集する仕組みとして、オーバーレイネットワークの一つであるスキップグラフ (Skip Graph, SG) [10] を用いた手法を提案した [11]。文献 [11] では各ノードからデータを収集する周期（収集周期）をキーとしてスキップグラフを構築し、最終的な宛先ノード（収集者）へ中継するノード上で複数のストリームを集約することで、各ノードのデータ送受信による負荷を軽減する。しかし、集約可能なストリームの数には上限が考えられ、既存手法では同じ周期の

¹ 奈良先端科学技術大学院大学
Nara Institute of Science and Technology
² 大阪大学データビリティフロンティア機構
Osaka University Institute for Datability Science
³ 大阪大学サイバーメディアセンター
Cybermedia Center, Osaka University
⁴ 国立研究開発法人情報通信研究機構
National Institute of Information and Communications Technology

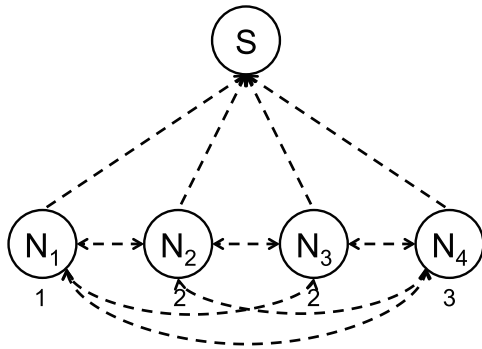


図 2 入力設定の例

ノードが多く存在する場合、特定の時刻に負荷が集中してしまう。

そこで本研究では、特定の時刻への負荷集中を軽減するため、位相調整を用いた負荷均等化手法を提案する。提案手法ではセンサデータの収集周期に加えて、各ノードへ個別の時間差（位相差）を与える。同じ周期でも位相差によって収集時刻が異なるため、集約可能な上限以上にストリームが特定の時刻へ集中する可能性を低減できる。

2. 問題設定

2.1 想定環境

本研究では、周期の異なるセンサストリーム収集における通信負荷を分散させることを目的とする。収集対象となるデータの発行ノード（Publisher）はセンサを備えており、センサデータを特定の周期で発行する。センサデータを収集したい端末（Subscriber）は該当する Publisher を探索し、データを定期的に受信する。

2.2 入力設定

センサデータの Subscriber を S 、 n 個の Publisher を N_i ($i = 1, \dots, n$) とする。また、 N_i が発行するセンサデータの周期を C_i とする。図 2 では、各ノードが Subscriber, Publisher を示しており、枝はセンサデータストリームの収集経路を示す。具体的には、アプリケーション層の通信リンクを示す。枝を点線で示したのは、収集方法によってはセンサデータストリームを収集しない枝となる可能性があるためである。上部にあるのが Subscriber S 、下部にあるのが $n = 4$ 個の Publisher N_1, \dots, N_4 である。各 Publisher 付近の数字は周期を示しており、 $C_0 = 1, C_1 = 1, C_2 = 2, C_3 = 2, C_4 = 3$ である。例えば、各 Publisher をライブカメラとすると、 N_1 が 1 秒に 1 回、 N_2 と N_3 が 2 秒に 1 回、 N_4 が 3 秒に 1 回画像を発信している場合に相当する。図 2 の例において、各 Publisher の周期（Cycle）と送信するセンサデータを表 1 に示す。

2.3 負荷の定義

Subscriber および Publisher の通信負荷は、センサデー

表 1 収集するセンサデータの例

Time	1	2	3	4	5	6	7	...
N_1 (Cycle=1)	○	○	○	○	○	○	○	...
N_2 (Cycle=2)		○		○		○		...
N_3 (Cycle=2)		○		○		○		...
N_4 (Cycle=3)			○			○		...

タストリームの受信による負荷と送信による負荷の合計で与える。受信による通信負荷は受信負荷と呼び、 N_i の受信負荷を I_i 、 S の受信負荷を I_0 で示す。送信による通信負荷は送信負荷と呼び、 N_i の送信負荷を O_i 、 S の送信負荷を O_0 で示す。

多くの場合、受信負荷および送信負荷は送受信するセンサデータストリームの単位時間あたりのセンサデータの数に比例する。 N_p が N_q ($q \neq p, p, q = 1, \dots, n$) に送信するセンサデータストリームの単位時間あたりのセンサデータの数を $R(p, q)$ で、 S が N_q から受信する数を $R(0, q)$ で表す。

S の通信負荷を L_0 、 N_i の通信負荷を L_i で示す。本研究では、単位時間あたりに 1 つのセンサデータを受信および送信する負荷を 1 と正規化し、 D_r の通信負荷 L_r を以下で与える。

$$L_r = I_r + O_r \quad (1)$$

$$I_r = \alpha \sum_{i=0}^n R(i, r) \quad (2)$$

$$O_r = \beta \sum_{i=0}^n R(r, i) \quad (3)$$

α および β はそれぞれ、1 つのセンサデータを受信および送信による負荷である。

2.4 目的関数

全ノードの総負荷 SL は以下で与えられる。

$$SL = \sum_{i=0}^n L_i \quad (4)$$

ここで、特定の時間があらかじめ設定されている場合、総負荷は各時刻における各ノードの受信負荷と送信負荷の和である。一方、各時刻における各ノードの負荷に着目すると、現実には各ノードやそのネットワーク環境は処理能力に応じて一定の負荷が許容される。そのため、電力使用におけるピークシフトのように、各ノードやそのネットワーク環境の許容量を超えないように負荷を分散させることが重要である。そこで本研究では、各ノードの瞬間的な負荷の最大値を「最大瞬間負荷」と呼び、最大瞬間負荷の最小化を目的関数とする。図 2 および表 1 の例では時刻 1~6 までの収集が以降に繰り返され、時刻 6 における Subscriber の負荷 4 が最大瞬間負荷となる。

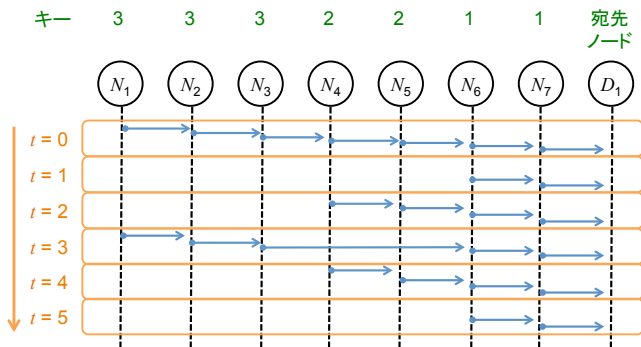


図 3 スキップグラフを用いたセンサデータストリーム収集

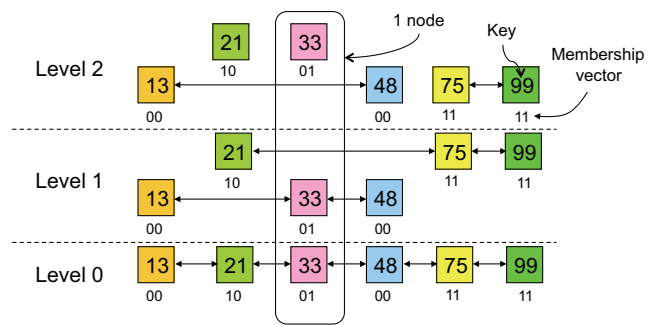


図 4 スキップグラフ

3. 提案手法

3.1 概要

我々は複数のノードが発信するセンサデータストリームを収集する仕組みとして、オーバーレイネットワークの一つであるスキップグラフ (Skip Graph, SG) [10] を用いた手法を提案した [11]. 文献 [11] では各ノードからデータを収集する周期 (収集周期) をキーとしてスキップグラフを構築し、最終的な宛先ノード (収集者) へ中継するノード上で複数のストリームを集約することで、各ノードのデータ送受信による負荷を軽減する. スキップグラフを用いたセンサデータストリーム収集を図 3 に示す. 図 3 では各ノードからほかのノードの探索はスキップグラフの階層的なリンク構造に基づき、各時刻のデータ収集経路の例を表している. また、文献 [11] ではキーを昇順か降順に並べる場合の性能評価を行い、降順に並べるほうが通信負荷が低いことを確認した. しかし、集約可能なストリームの数には上限が考えられ、既存手法では同じ周期のノードが多く存在する場合、特定の時刻に負荷が集中してしまう. そこで本研究では、特定の時刻への負荷集中を軽減するため、位相調整を用いた負荷均等化手法を提案する. 提案手法ではセンサデータの収集周期に加えて、各ノードへ個別の時間差 (位相差) を与える. 同じ周期でも位相差によって収集時刻が異なるため、集約可能な上限以上にストリームが特定の時刻へ集中する可能性を低減できる.

3.2 スキップグラフ

スキップグラフは skip list を P2P モデルに適用したオーバーレイネットワークである. スキップグラフの構造を図 4 に示す. 図 4 において、正方形はピア (ノード) のルーティングテーブルのエントリを表し、中の数字はエントリのキー値を表す. ピアはキー値の順に並んでおり、各ピア間は双方向にリンクしている. エントリの下にある数字は membership vector と呼ばれるエントリ生成時に割り当てられる整数値である. ピアは各エントリに対応しており、membership vector が一つのピアに対応している. 各エントリは、複数のレベルでエントリ間をつなぐリストに属

する. エントリが各レベルで属するリストは membership vector によって決まる. スキップグラフでは、単一キーを検索する場合、上位レベルから下位レベルへとクエリの転送先に適したピアを探索することで行われていく. これは、上位レベルでのルーティングの方が下位レベルのそれと比べて検索キーに近づく距離が大きく、より効率的に検索が行えるためである. 範囲検索を行う場合には、各ピアはクエリの届いたレベルから隣接ピアのうち検索キーを超えないピアを探す. 該当するピアが見つかった場合はそのピアにクエリを転送する. 該当するピアが見つからない場合は、1 つレベルを下げ、再び隣接ピアで該当するピアを探す. n をピア数とすると、キーの検索におけるホップ数は $O(\log n)$ であり、各ノードが持つリンク数であるルーティングテーブルの大きさは平均 $\log n$ となる.

3.3 位相調整

収集対象のノードが N_i ($i = 1, 2, \dots, 7$), 宛先ノードが D_i ($i = 1$), 選択可能な収集周期が $C_i = 1, 2, 3$ の図 3 と同じ環境において、位相差を与えたセンサデータストリーム収集の例を図 5 に示す. 図 5 では収集周期をキーとしてノードを並べ、キーの値が最も低い位置に宛先ノードを配置し、スキップグラフを構築している. また、各ノードの位相差 d はそれぞれの収集周期 C_i に基づく $0 \leq d < C_i$ の範囲で与える. 例えば図 5 で収集周期が同じ $C_3 = 3$ でも、位相差 $d = 0$ の N_3 は時刻 $t = 0, 3, 6, \dots$ で、 $d = 1$ の N_2 は時刻 $t = 1, 4, 7, \dots$ でセンサデータを発信する. 同じ周期でも位相差によって収集時刻が異なるため、集約可能な上限以上にストリームが特定の時刻へ集中する可能性を低減できる. 図 5 では、図 3 と異なる時刻に収集するデータの経路を異なる色の矢印で示している.

4. 評価

4.1 シミュレーション環境

本研究では、3 章で提案した負荷均等化手法をシミュレーションにより評価する. 本シミュレーション環境では各ノードが選択可能な収集周期を $C_i = 1, 2, \dots, 10$, 試行時刻を $t = 0, 1, \dots, 2519$ とする. なお、本来は各ノードでの

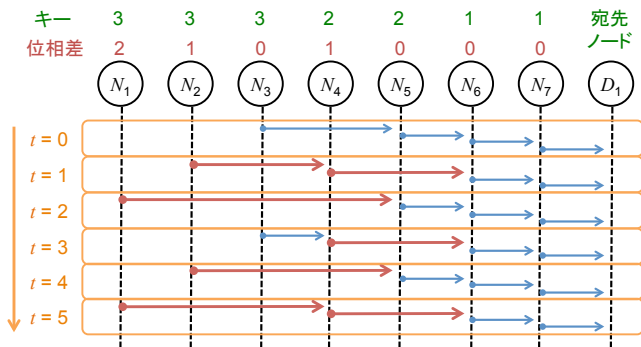


図 5 位相差を与えたセンサデータストリーム収集

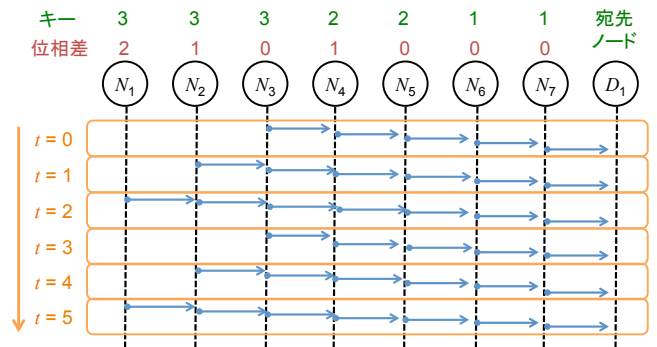


図 7 DC 方式による収集の例

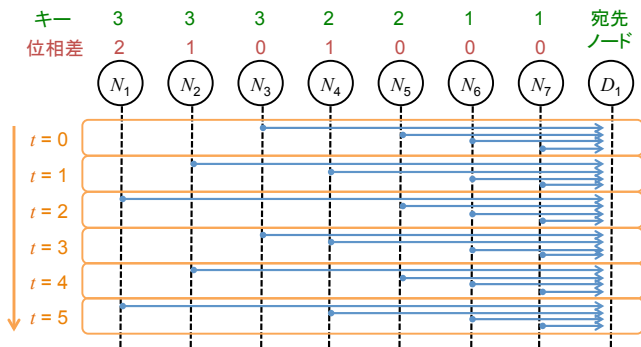


図 6 SD 方式による収集の例

処理時間やノード間の通信遅延が生じるが、本シミュレーションでは無視できるものとする。比較手法としては位相調整を行わないスキップグラフベースの手法のほか、最終的な宛先ノードへ各ノードが直接送信する方式 (Source Direct, SD) と各ノードをキーの降順で隣接ノードへ直列につなぐデ이지チェーンによる方式 (Daisy Chain, DC) を用いる。また、SD 方式と DC 方式はそれぞれ位置調整を行う場合と行わない場合で比較する。図 5 と同様に位相調整を行う環境において、SD 方式と DC 方式によるセンサデータストリーム収集の例を図 6 と図 7 にそれぞれ示す。シミュレーションはノード数と毎時刻に各ノードで集約可能な最大ストリーム数を変え、それぞれの環境と手法で 10 回ずつ行い、各試行で測定した評価項目の平均値を算出した。評価項目として、設定時間における各ノードの瞬間的な負荷の最大値 (最大瞬間負荷) と全ノードの総負荷、各配信元ノードから宛先ノードまでの平均ホップ数、最大ホップ数を測定した。

4.2 シミュレーション結果

毎時刻に各ノードで集約可能な最大ストリーム数を 10 とし、ノード数を 100~400 まで変化させた環境での最大瞬間負荷を図 8 に、全ノードの総負荷を図 9 に示す。図 8 より、宛先ノードが多くの配信元ノードからデータを同時に受信する SD 方式と比べて、位相調整を行うスキップグラフベースの提案手法は瞬間的な負荷が低い。また、DC 方式では各ノードで集約可能な最大ストリーム数が大きい

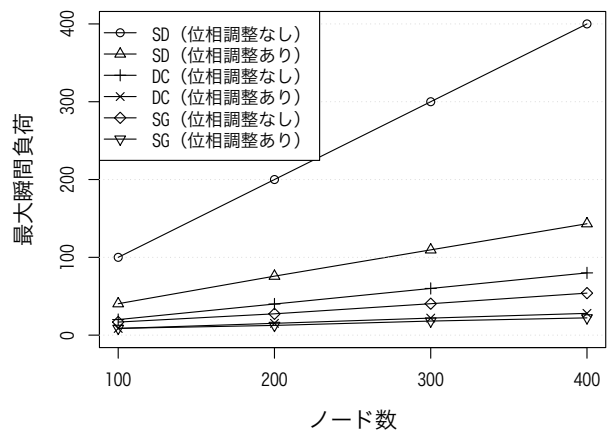


図 8 ノード数による最大瞬間負荷

ほど送受信回数が減少し、各ノードの負荷も低下するが、本シミュレーション環境では提案手法は DC 方式よりも瞬間的な負荷が低い。さらに、位相調整を行う提案手法は同じ周期の配信元ノードでもデータの送受信時刻が異なるため、同じスキップグラフベースでも、位相調整を行わない手法と比べて瞬間的な負荷が低い。一方、図 9 より、総負荷は配信元ノードが宛先ノードへデータを直接送信する SD 方式が最も低い。ただし、各ノードで集約可能な最大ストリーム数によって送受信回数が変わる DC 方式と比べて、本シミュレーション環境ではスキップグラフベースの提案手法の総負荷は低い。

同様に、毎時刻に各ノードで集約可能な最大ストリーム数を 10 とし、ノード数を 100~400 まで変化させた環境での各配信元ノードから宛先ノードまでの平均ホップ数を図 10 に、最大ホップ数を図 11 に示す。図 10 と図 11 より、配信元ノードが宛先ノードへデータを直接送信する SD 方式は前述の最大瞬間負荷が高い一方、平均ホップ数と最大ホップ数はともに 1 である。また、ノード数がホップ数に大きく影響する DC 方式と比べて、スキップグラフベースの提案手法はノード数を n とした場合に平均ホップ数を $\log n$ 程度に抑えられている。

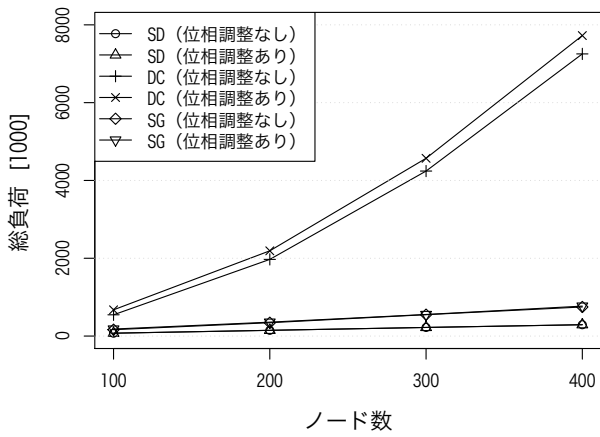


図 9 ノード数による総負荷

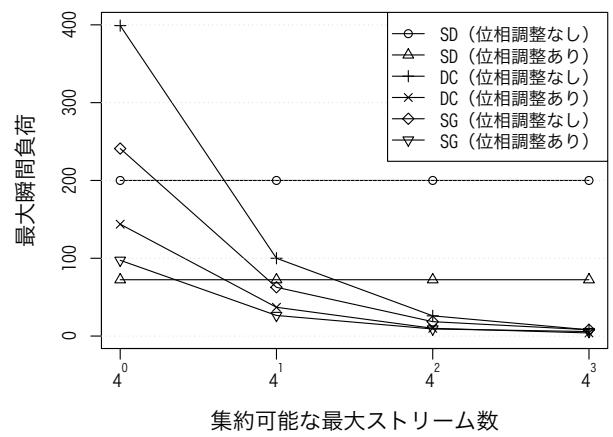


図 12 集約可能な最大ストリーム数による最大瞬間負荷

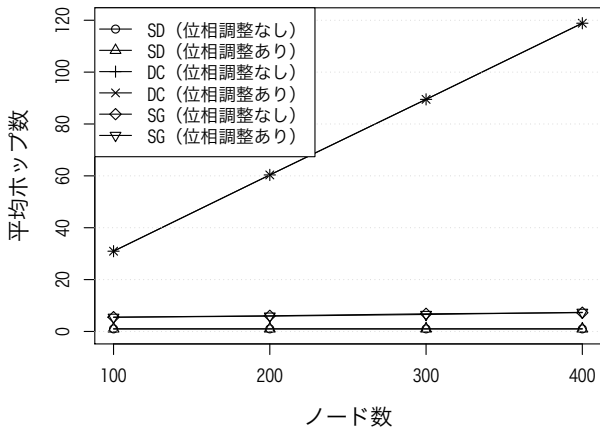


図 10 ノード数による平均ホップ数

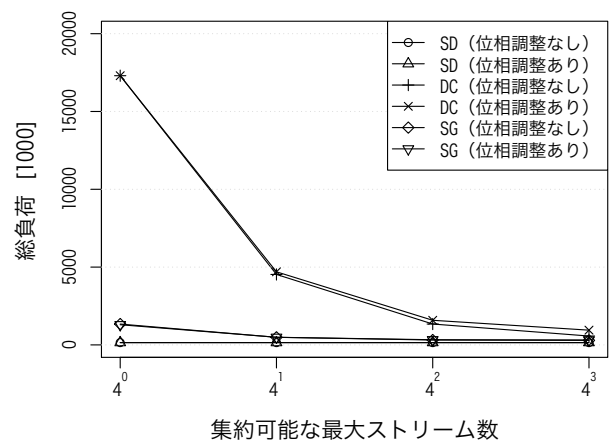


図 13 集約可能な最大ストリーム数による総負荷

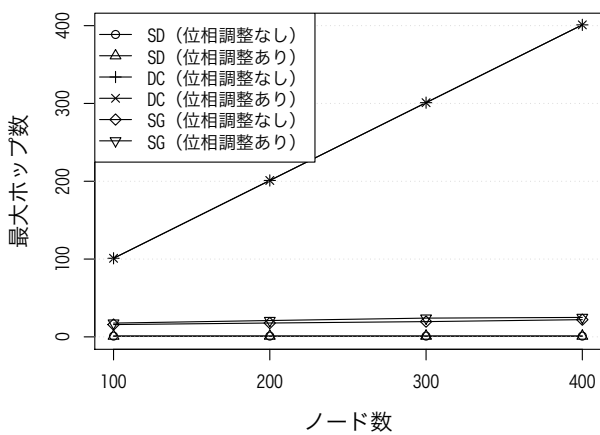


図 11 ノード数による最大ホップ数

ノード数を 200 とし、毎時刻に各ノードで集約可能な最大ストリーム数を $4^0 \sim 4^3$ まで変化させた環境での最大瞬間負荷を図 12 に、全ノードの総負荷を図 13 に示す。SD 方式では配信元ノードが宛先ノードへデータを直接送信するため、SD 方式の最大瞬間負荷は集約可能な最大ストリーム数に関係なく一定である。図 12 と図 13 より、SD 方式の最大瞬間負荷以外は最大瞬間負荷と総負荷はともに集約可能な最大ストリーム数が大きくなるに従って低下するが、各負荷を低下させるために多くのストリームの集約が必要な DC 方式と比べて、スキップグラフベースの提案手法は 4^1 の現実的な数のストリームの集約でも低い負荷を実現できている。また、各配信元ノードから宛先ノードまでのホップ数は集約可能な最大ストリーム数に関係ないため、平均ホップ数と最大ホップ数はそれぞれ図 10 と図 11 と同様である。

5. まとめ

本研究では複数のノードからのセンサデータストリーム収集において、特定の時刻への負荷集中を軽減するため、位相調整を用いた負荷均等化手法を提案した。提案手法ではセンサデータの収集周期に加えて、各ノードへ個別の時間差（位相差）を与える。同じ周期でも位相差によって収集時刻が異なるため、集約可能な上限以上にストリームが特定の時刻へ集中する可能性を低減できる。提案手法はシミュレーションにより評価し、位相調整を行わない場合と比べて各ノードの瞬間的な負荷の最大値（最大瞬間負荷）が減少することを確認した。

今後の課題としては、本研究の評価に用いたシミュレーション環境では各ノードでの処理時間やノード間の通信遅延を無視できるものとしているため、それらを考慮した環境での評価が考えられる。また、その際、宛先ノードまでの遅延とストリーム集約による負荷軽減の両方を踏まえて、各ノードが複数のストリームを集約するための待ち時間を検討する必要がある。

謝辞 本研究の一部はJSPS 科研費 16K16059, 17K00146 および公益財団法人放送文化基金の助成による成果である。

参考文献

- [1] Hodges, S., Taylor, S., Villar, N., Scott, J. and Fischer, D. B. P. T.: Prototyping Connected Devices for the Internet of Things, *IEEE Computer*, Vol. 46, No. 2, pp. 26–34 (2013).
- [2] Stoica, I., Morris, R., Liben-Nowell, D., Karger, D. R., Kaashoek, M. F., Dabek, F. and Balakrishnan, H.: Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications, *IEEE/ACM Transactions on Networking*, Vol. 11, No. 1, pp. 17–32 (2003).
- [3] Legtchenko, S., Monnet, S., Sens, P. and Muller, G.: RelaxDHT: A Churn-Resilient Replication Strategy for Peer-to-Peer Distributed Hash-Tables, *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, Vol. 7, No. 2, Article 28 (2012).
- [4] Bharambe, A. R., Agrawal, M. and Seshan, S.: Mercury: Supporting Scalable Multi-Attribute Range Queries, *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2004)*, pp. 353–366 (2004).
- [5] Tanin, E., Harwood, A. and Samet, H.: Using a Distributed Quadtree Index in Peer-to-Peer Networks, *The International Journal on Very Large Data Bases (VLDB)*, Vol. 16, No. 2, pp. 165–178 (2007).
- [6] Mondal, A., Lifu, Y. and Kitsuregawa, M.: P2PR-Tree: An R-Tree-Based Spatial Index for Peer-to-Peer Environments, *Proceedings of the International Workshop on Peer-to-Peer Computing and Databases in Conjunction with the 9th International Conference on Extending Database Technology (EDBT 2004)*, pp. 516–525 (2004).
- [7] 金子 雄, 春本 要, 福村真哉, 下條真司, 西尾章治郎: コビキタス環境における端末の位置情報に基づく P2P ネットワーク, *情報処理学会論文誌データベース*, Vol. 46, No. SIG18(TOD28), pp. 1–15 (2005).
- [8] Shu, Y., Ooi, B. C., Tan, K.-L. and Zhou, A.: Supporting Multi-Dimensional Range Queries in Peer-to-Peer Systems, *Proceedings of the 5th IEEE International Conference on Peer-to-Peer Computing (P2P 2005)*, pp. 173–180 (2005).
- [9] 大西真晶, 源元佑太, 江口隆之, 加藤宏章, 西出 亮, 上島紳一: ノード位置を用いた P2P モデルのためのドロネー図の自律分散生成アルゴリズム, *情報処理学会論文誌データベース*, Vol. 47, No. SIG4(TOD29), pp. 51–64 (2006).
- [10] Aspnes, J. and Shah, G.: Skip Graphs, *ACM Transactions on Algorithms (TALG)*, Vol. 3, No. 4 (37), pp. 1–25 (2007).
- [11] 川上朋也, 石 芳正, 義久智樹, 寺西裕一: スキップグラフを用いたスケーラブルなセンサデータストリーム収集システムの評価, *マルチメディア, 分散, 協調とモバイル (DICOMO 2016) シンポジウム論文集*, pp. 1208–1214 (2016).