

大規模センサデータストリーム収集における データ集約のための待ち時間設定方式の検討

川上 朋也¹ 義久 智樹² 松本 哲² 寺西 裕一^{3,2}

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、多くのデータを集約するためには各中継ノード上で待ち時間が必要となり、不適切な待ち時間の設定によって、集約前に次のノードへデータが転送されたり、最終的な宛先ノードまでの遅延などが生じる。そこで本論文では、各ノードの最大待ち時間が目安として与えられるものとし、その最大待ち時間以内で、各ノードがデータを効率的に集約可能な待ち時間設定方式を提案する。提案手法はシミュレーションにより評価し、各ノードへ設定した待ち時間によってデータ収集時の負荷や宛先ノードまでの遅延が変化することを確認した。

1. はじめに

Internet of Things (IoT) [1] ではさまざまなモノがインターネットにつながり、カメラによる映像などを含め、多種多様なセンサをもつ機器（ノード）がデータを連続的に発信する。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 [2–10]。

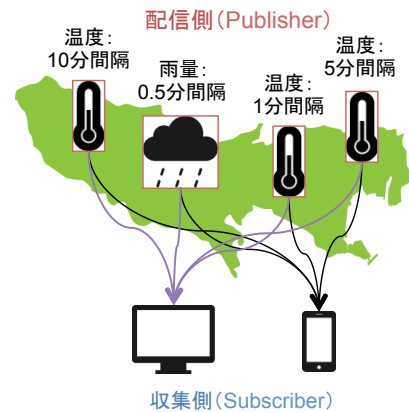


図 1 異なる間隔でのセンサデータストリーム収集

オーバー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) [11–15] を用いた手法を提案した [16]。文献 [16] 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）までの中継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。また、センサデー

¹ 福井大学大学院工学研究科
Graduate School of Engineering, University of Fukui
² 大阪大学サイバーメディアセンター
Cybermedia Center, Osaka University
³ 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

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)			○			○		...

タの収集周期に加えて、個別の時間差（位相差）を各ノードへ与えることで、特定の時刻への負荷集中を軽減する。しかし、多くのデータを集約するためには各中継ノード上で待ち時間が必要となり、不適切な待ち時間の設定によって、集約前に次のノードへデータが転送されたり、最終的な宛先ノードまでの遅延などが生じる。

そこで本論文では、各ノードの最大待ち時間が目安として与えられるものとし、その最大待ち時間以内で、各ノードがデータを効率的に集約可能な待ち時間設定方式を提案する。提案手法ではキー空間上の位置に基づき、各ノードに個別の待ち時間を設定し、宛先ノードから遠いノードほど、早い時刻に次のノードへデータを転送する。

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 $R(p, q)$ 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 つのセンサデータの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。

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、オーバレイネットワークの一つ

であるスキップグラフ (Skip Graph, SG) を用いた手法を提案した [16]. 文献 [16] 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. また、センサデータの収集周期に加えて、個別の時間差 (位相差) を各ノードへ与えることで、特定の時刻への負荷集中を軽減する. しかし、多くのデータを集約するためには各中継ノード上で待ち時間が必要となり、不適切な待ち時間の設定によって、集約前に次のノードへデータが転送されたり、最終的な宛先ノードまでの遅延などが生じる. そこで本論文では、各ノードの最大待ち時間が目安として与えられるものとし、その最大待ち時間以内で、各ノードがデータを効率的に集約可能な待ち時間設定方式を提案する. 提案手法ではキー空間上の位置に基づき、各ノードに個別の待ち時間を設定し、宛先ノードから遠いノードほど、早い時刻に次のノードへデータを送信する.

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レイネットワークである. スキップグラフの構造を図 3 に示す. 図 3 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

文献 [16] 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した. 文献 [16] で

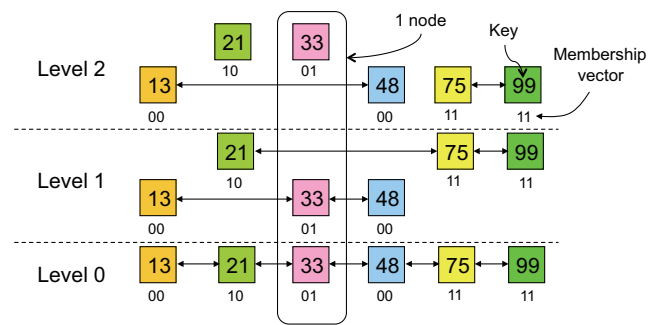


図 3 スキップグラフ

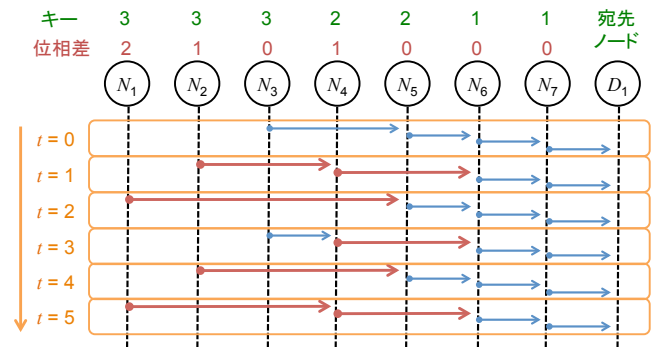


図 4 位相差を与えたセンサデータストリーム収集

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.

収集対象のノードが N_i ($i = 1, 2, \dots, 7$), 宛先ノードが D_1 ($i = 1$), 選択可能な収集周期が $C_i = 1, 2, 3$ の環境において、位相差を与えたセンサデータストリーム収集の例を図 4 に示す. 図 4 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. 図 4 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、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 4 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.

3.4 待ち時間の設定

文献 [16] および図 4 の例に示す収集方式では、収集周期が長くキー空間上の位置が左側のノードからデータを先に転送すれば、宛先ノードまでの中継ノードで効率的にデータを集約できる. そのため、キー空間上の位置が宛先ノードに近いほど、自身のデータを転送するまでの待ち時間を長く設定することで、他ノードからのデータを集約する可

Algorithm 1: 各ノードの待ち時間の決定

Input: 選択可能な収集周期の配列 C , 自身の収集周期 c , 自身の位相差 d , 最大待ち時間 w_{\max}

Output: 自身の待ち時間

```

1  $p = 0$  // 周期と位相差による自身の位置
2 for  $i$  in  $C$  do
3   if  $c = C_i$  then
4     break
5    $p = p + C_i$ 
6  $p = p + d$ 
7  $p_{\max} = 0$  // 周期と位相差による最大値
8 for  $i$  in  $C$  do
9    $p_{\max} = p_{\max} + C_i$ 
10 return  $w_{\max}(1 - p/p_{\max})$ 

```

能性が高められる。一方、スキップグラフなどの自律分散型のオーバレイネットワークでは、自身がリンクを直接構築していないノードの状況を把握するためにはコストがかかり、特にネットワークの規模に依存する。そこで提案手法では、各ノードが自身の収集周期と位相差に基づいてキー空間上の位置を大まかに推定し、自身のデータ転送までの待ち時間を設定する。待ち時間の決定においては、宛先ノードの左隣に位置し、最も右側の配信元ノードの待ち時間を最大待ち時間 w_{\max} として、各ノードがあらかじめ把握しているものとする。各ノードは自身のキー空間上での推定位置と最大待ち時間に基づいて、宛先ノードから遠いノードほど早い時刻に次のノードへデータを転送するよう、個別の待ち時間を設定する。

提案手法における待ち時間決定の流れをアルゴリズム 1 に示す。1 行目から 6 行目では、自身の収集周期と位相差から、最大待ち時間のノードまでの距離を求めている。また、6 行目から 9 行目では、選択可能な収集周期から、キー空間上の距離の最大値を求めている。10 行目では自身の位置とキー空間上の距離の最大値から自身の相対的な位置を求め、相対的な位置が遠いほど待ち時間が短くなるよう、自身の待ち時間を設定する。

4. 評価

本論文では位相調整を用いたスキップグラフベースの手法において、3 章で提案した待ち時間設定方式をシミュレーションにより評価する。

4.1 シミュレーション環境

シミュレーション環境を表 2 に示す。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。配信元ノード (Publisher) の数は 250, 500, 750, 1000 で変化させ、データを集約しつつ、1 台の宛先ノード (Subscriber) へ転送する。各配信

表 2 シミュレーション環境

項目	値
選択可能な収集周期	1, 2, ..., 10
収集周期の決定方法	ランダム
宛先ノード数	1
配信元ノード数	250, 500, 750, 1000
平均通信遅延	0.005, 0.01, 0.02, 0.04
最大待ち時間	0 (待ち時間なし), 0.5, 0.75, 1.0
集約可能なストリーム数	10
試行回数	10
評価指標	最大瞬間負荷, 総負荷 平均到達遅延, 最大到達遅延

元ノードの収集周期はランダムに決定する。各ノードの通信遅延は平均を $0.005 \times 2^0, 0.005 \times 2^1, 0.005 \times 2^2, 0.005 \times 2^3$ で変化させ、標準偏差 0.001 の正規分布で与える。毎時刻に各ノードで集約可能な最大ストリーム数は 10 とする。

シミュレーションはノード数と平均通信遅延を変えつつ、最大待ち時間を 0 (待ち時間なし), 0.5, 0.75, 1.0 とした場合の結果を比較する。また、シミュレーションはそれぞれの環境で 10 回ずつ行い、各試行で測定した評価指標の平均値を算出する。変化させる項目以外の既定値はノード数を 500, 平均通信遅延が 0.01 とする。評価指標として、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到達遅延、最大到達遅延を測定した。

4.2 ノード数による結果

各ノードの平均通信遅延を 0.01 とし、ノード数を 250 から 1000 まで変化させた環境での最大瞬間負荷を図 5 に、全ノードの総負荷を図 6 に示す。図 5 より、最大待ち時間が長いほどデータの集約される可能性が高いため、最大瞬間負荷は軽減される。ただし、本シミュレーション環境ではノード数が 1000, 最大待ち時間が 1.0 の場合でも、最大待ち時間が 0 (待ち時間なしで、データを集約しない) と比べて約 9% の減少となっている。図 6 の総負荷は減少率が最大瞬間負荷よりも高く、待ち時間なしの場合と比べて、減少率は最大で約 30% となっている。

同様に、各ノードの平均通信遅延を 0.01 とし、ノード数を 250 から 1000 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到達遅延を図 7 に、最大到達遅延を図 8 に示す。図 7 より、最大待ち時間が 1.0 の場合は負荷が最も低い一方、平均到達遅延は 1.0 を超え、次の周期時刻へ割り込んでいる。ただし、待ち時間なしの場合は到達遅延の増加が約 0.1, 待ち時間ありの場合は最大待ち時間から約 0.05 の増加であり、許容する到達遅延によっては影響が小さいとみなせる。また、提案手法は平均ホップ数を $\log n$ 程度に抑えられるスキップグラフを用いているため、本シミュレーション程度のノード数の増加では平均到

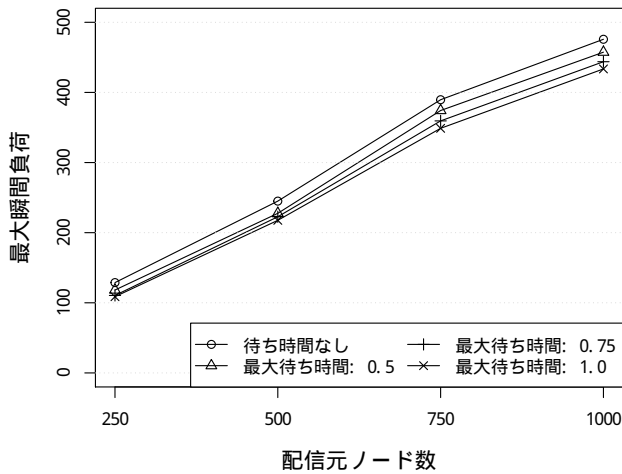


図 5 ノード数による最大瞬間負荷

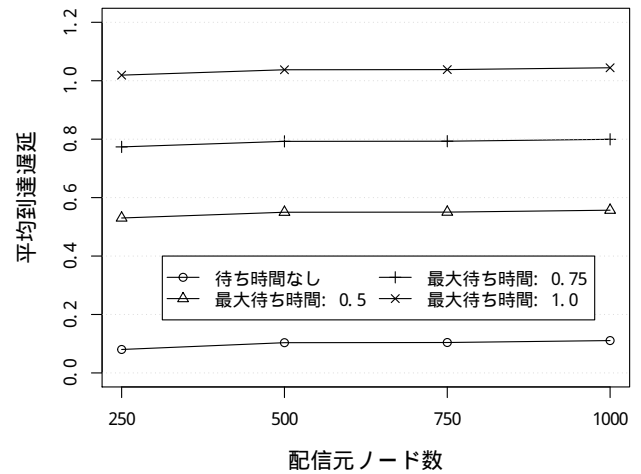


図 7 ノード数による平均到達遅延

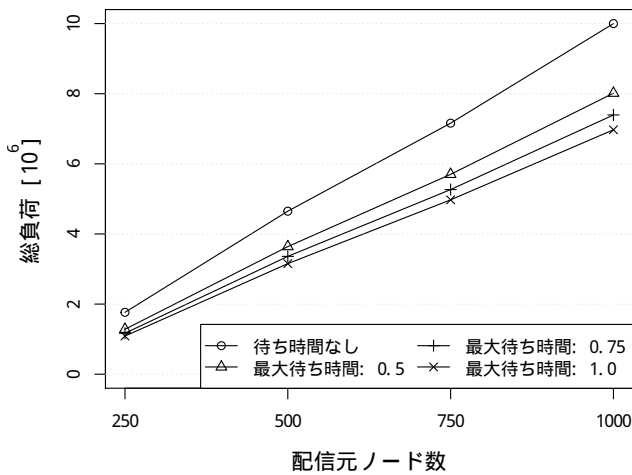


図 6 ノード数による総負荷

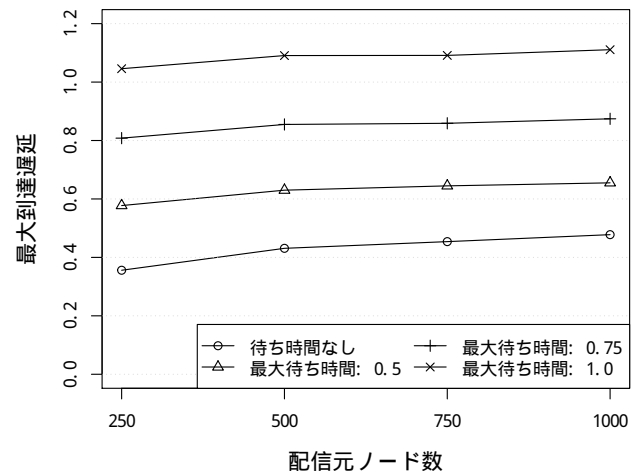


図 8 ノード数による最大到達遅延

達遅延への影響は小さい。一方、図 8 より、最大到達遅延はノード数による影響が平均到達遅延と比べて大きい。増加率は最大待ち時間が短いほど大きく、ノード数が 1000、待ち時間なしの場合、ノード数が 250 の場合と比べて最大到達遅延は約 34%増加している。

4.3 平均通信遅延による結果

ノード数を 500 とし、各ノードの平均通信遅延を 0.005 から 0.04 まで変化させた環境での最大瞬間負荷を図 9 に、全ノードの総負荷を図 10 に示す。図 9 より、平均通信遅延が大きいほどデータの集約される可能性が低いため、最大瞬間負荷は増加する。図 10 の総負荷も同様であるが、増加率は最大瞬間負荷よりも高く、平均通信遅延が 0.04、最大待ち時間が 1.0 の場合、平均通信遅延が 1/8 の 0.005 の場合と比べて総負荷は約 43%の増加している。

同様に、ノード数を 500 とし、各ノードの平均通信遅延を 0.005 から 0.04 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到達遅延を図 11 に、最大到達遅延を図 12 に示す。ノード数による結果と同様、最大待

ち時間が 1.0 の場合は平均到達遅延が 1.0 を超え、次の周期時刻へ割り込んでいる。また、平均通信遅延は平均到達遅延と最大到達遅延に影響し、特に平均通信遅延が 0.04、待ち時間なしの場合、平均通信遅延が 1/8 の 0.005 の場合と比べて最大到達遅延は約 4 倍に増加している。

5. まとめ

本論文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、各ノードがデータを効率的に集約可能な待ち時間設定方式を提案した。提案手法ではキー空間上の位置に基づき、各ノードに個別の待ち時間を設定し、宛先ノードから遠いノードほど、早い時刻に次のノードへデータを転送する。提案手法はシミュレーションにより評価し、各ノードへ設定した待ち時間によってデータ収集時の負荷や宛先ノードまでの遅延が変化することを確認した。

今後の課題としては、提案手法ではキー空間上の相対的な位置に比例して待ち時間を決定していたが、異なる待ち時間の決定方式の検証が考えられる。また、本論文の評価では各ノードの収集周期はランダムに、平均通信遅延を正

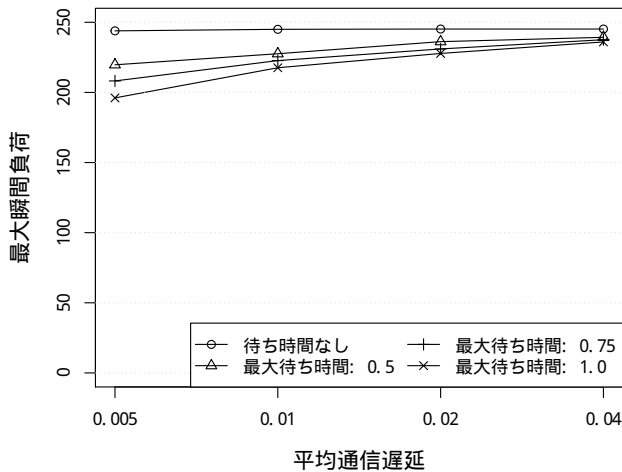


図 9 平均通信遅延による最大瞬間負荷

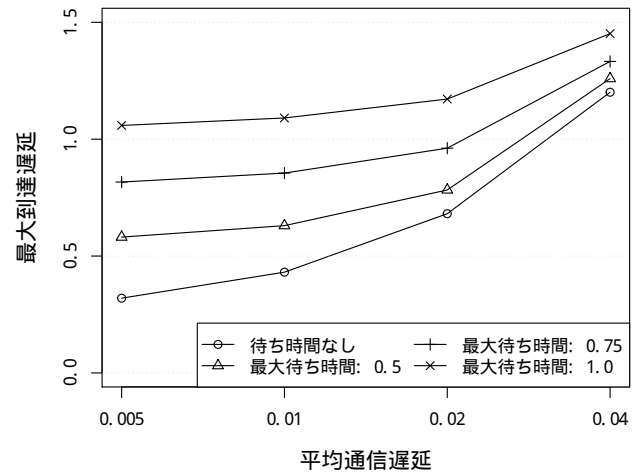


図 12 平均通信遅延による最大到達遅延

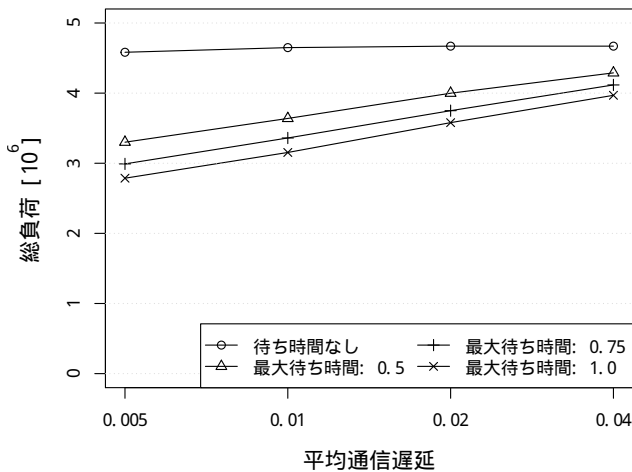


図 10 平均通信遅延による総負荷

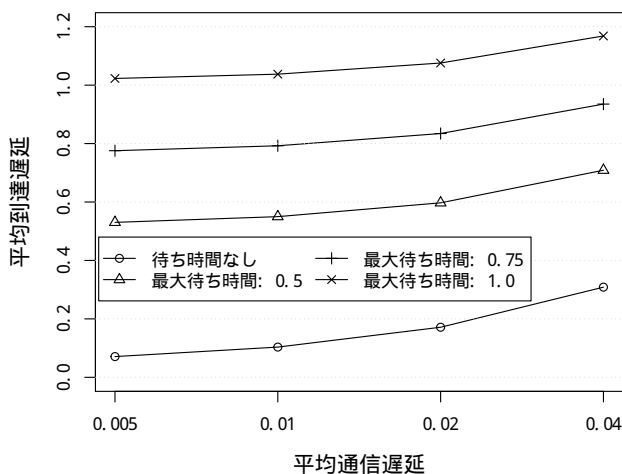


図 11 平均通信遅延による平均到達遅延

規分布に設定したが、Zipf 分布などの異なる環境での評価も挙げられる。

謝辞 本研究の一部は G-7 奨学財団研究開発助成事業による成果である。

参考文献

- [1] Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H. and Zhao, W.: A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications, *IEEE Internet of Things Journal*, Vol. 4, No. 5, pp. 1125–1142 (2017).
- [2] 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).
- [3] 呉 承彦, 安倍広多, 石橋勇人, 松浦敏雄: Chord#における経路表の維持管理コスト削減手法の提案とその評価, *情報処理学会論文誌*, Vol. 53, No. 12, pp. 2752–2761 (2012).
- [4] Tatarave, S. K., Tripathy, S. and Ghosh, R. K.: V-Chord: An Efficient File Sharing on LTE/GSM Network, *Proceedings of the 19th International Conference on Distributed Computing and Networking (ICDCN 2018)* (2018).
- [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] Gu, Y. and Boukerche, A.: HD Tree: A Novel Data Structure to Support Multi-Dimensional Range Query for P2P Networks, *Journal of Parallel and Distributed Computing*, Vol. 71, No. 8, pp. 1111–1124 (2011).
- [7] 金子 雄, 春本 要, 福村真哉, 下條真司, 西尾章治郎: ユ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] 小西佑治, 寺西裕一, 竹内 亨, 春本 要, 下條真司, 西尾章治郎: センサ観測値分布の概要把握を可能とする階層化ドロネーオーバーレイネットワーク構築手法, *情報処理学会論文誌*, Vol. 51, No. 2, pp. 624–634 (2010).
- [10] Ohnishi, M., Inoue, M. and Harai, H.: Incremental Distributed Construction Method of Delaunay Overlay Network on Detour Overlay Paths, *Journal of Information Processing (JIP)*, Vol. 21, No. 2, pp. 216–224 (2013).

- [11] Aspnes, J. and Shah, G.: Skip Graphs, *ACM Transactions on Algorithms (TALG)*, Vol. 3, No. 4 (37), pp. 1–25 (2007).
- [12] 小西佑治, 吉田 幹, 竹内 亨, 寺西裕一, 春本 要, 下條真司: 単一ノードに複数キーを保持可能とする Skip Graph 拡張, *情報処理学会論文誌*, Vol. 49, No. 9, pp. 3223–3233 (2008).
- [13] Banno, R., Fujino, T., Takeuchi, S. and Takemoto, M.: SFB: A Scalable Method for Handling Range Queries on Skip Graphs, *IEICE Communications Express*, Vol. 4 (2015), No. 1, pp. 14–19 (2015).
- [14] Boshrooyeh, S. T. and Ozkasap, O.: Guard: Secure Routing in Skip Graph, *Proceedings of the 2017 IFIP Networking Conference (IFIP Networking) and Workshops* (2017).
- [15] Aoki, Y., Ohnishi, M. and Shudo, K.: Ballistic Skip Graph: A Skip Graph-Style Constant-Degree Structured Overlay, *Proceedings of the 23rd IEEE Symposium on Computers and Communications (ISCC 2018)* (2018).
- [16] 川上朋也, 石 芳正, 義久智樹, 寺西裕一: 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, *マルチメディア, 分散, 協調とモバイル (DICOMO2018) シンポジウム論文集*, pp. 760–765 (2018).