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

川上 朋也[1,2] 石 芳正[3,2] 義久 智樹[2] 寺西 裕一[4,2]

概要：本研究では，Skip Graph を用いた周期的なセンサデータ収集システムについて提案と評価を行う．一般的にセンサデータは一定の周期で定期的に生成され，発信者（publisher）によって発信される．一方，収集者（subscriber）は自身の処理性能やセンサデータの用途などによって，要求するセンサデータの周期が異なる．検討システムでは，構造化オーバレイネットワークの Skip Graph をセンサデータの種類（topic）と周期の組合せに基づいて構築し，複数の publisher からのセンサデータを topic や周期に基づいて集約しつつ収集する．センサデータを集約しながら転送することで，多くの publisher が存在する環境でも各ノードの瞬間的な送受信数を減らし，通信負荷の最大値を低く抑えられる．提案手法はシミュレーションにより評価し，周期をキーとして降順に並べる場合が有効であることを確認した．

Evaluation of a Scalable Collection System for Sensor Data Stream Using Skip Graphs

TOMOYA KAWAKAMI[1,2] YOSHIMASA ISHI[3,2] TOMOKI YOSHIHISA[2] YUUICHI TERANISHI[4,2]

1. はじめに

スマートフォン，センサ，ウェアラブルデバイスなど，さまざまな「モノ」がインターネットへ接続する Internet of Things (IoT) の実現に対する期待が高まっている．IoT 環境における ICT サービスでは，実世界で起きる事象 (イベント) にリアルタイムに対応するイベントドリブンなサービスとなることが期待されている．そうしたサービスを構成するためのメッセージ交換機構として，非同期かつオンデマンドでの情報配信が可能な pub/sub メッセージングが広く用いられるようになってきた．IoT におけるイベント配信機能としては，topic-base pub/sub (TBPS) が広く用いられつつある．

IoT においては，しばしばセンサデータの広域的な収集が行われる．TBPS では，メッセージは「トピック」を通じてデータ交換がなされるが，センサデータ収集を行う IoT のアプリケーションでは，センサが，センサデータの種別をトピックとして指定してデータを発行（publish）し，センサデータ収集を行うアプリケーションが，必要なセンサデータの種別に対応するトピックを購読（subscribe）する．

一方，センサから発信されるデータは，センサの性能・設定や，観測する対象に応じて異なる周期を持つことになると考えられる．例えば，「東京都の温度」というトピックで収集がなされたとき，温度センサの性能や設定によっては，1 分に 1 回，10 分に 1 回といった異なる周期で観測・発信されるデータが対象となる．また，「東京都の気象センサデータ」というトピックで異なるセンサを統合した収集を行う場合，温度センサは 10 分に 1 回，風力センサは 10 秒に 1 回などのように，観測対象によって異なる周期のデータが対象となる．本研究では，周期的に発生するこれらのセンサデータをセンサデータストリームと呼ぶ．

広域にある大量のセンサを対象とするセンサデータストリーム収集では，subscriber は大量のセンサデータを受信

[1] 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology
[2] 大阪大学サイバーメディアセンター
Cybermedia Center, Osaka University
[3] 株式会社 PIAX
PIAX Inc.
[4] 国立研究開発法人情報通信研究機構
National Institute of Information and Communications Technology

することになる．したがって，データ受信のためのネットワークプロセスの負荷や，ネットワークトラヒック量をできる限り削減する必要がある．ネットワークプロセスの負荷は，ネットワークセッション（例えば，TCP コネクション）の数，すなわち，データを送信する経路の数に応じて決まる．ネットワークトラヒック量は，ネットワークを流れるパケット数を指す．例えば，同じネットワークセッション上を 2 つの異なるメッセージを送信する必要があるとき，2 つを 1 つのメッセージにまとめて送信できれば，パケット数は削減される．したがって，データ収集にかかる経路数，メッセージ数を削減する必要がある．

ひとつのノードに対するデータ転送のネットワークセッションの数を削減し，TBPS を構成する方法として，オーバレイネットワークによる方法が多数提案されている．本研究では，Skip Graph を用いた TBPS のためのオーバレイをベースとし，大量の publisher が存在する状況のもと，周期的に発信されるデータをできる限り集約し，メッセージ数を削減したデータ収集を可能とするスケーラブルなセンサデータストリーム収集法の検討と評価を行う．

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，トピックを指定して購読する．

2.2 入力設定

センサデータの subscriber を S，n 個の publisher を N_i（$i = 1, \cdots, n$）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 1** 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，$\cdots$，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，D_4 が 3 秒に 1 回画像を発信している場合に相当する．図 1 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．

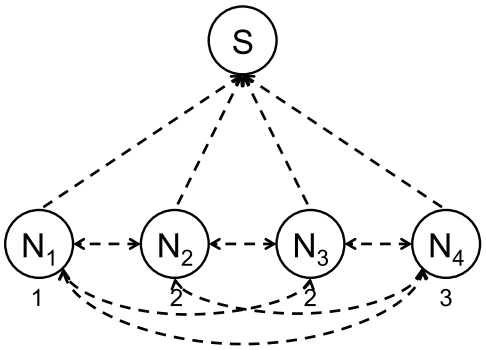

図 1 入力設定の例
Fig. 1 An example of input setting

表 1 収集するセンサデータの例
Table 1 An example of the sensor data collection

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

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（$q \neq p$，$p, q = 1, \cdots, n$）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．

$$L_r = I_r + O_r \tag{1}$$

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

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

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

システム全体の通信負荷 SL は以下で与えられる．

$$SL = \sum_{i=0}^{n} L_i \tag{4}$$

また，負荷分散の指標として，以下の Fairness Index（FI）

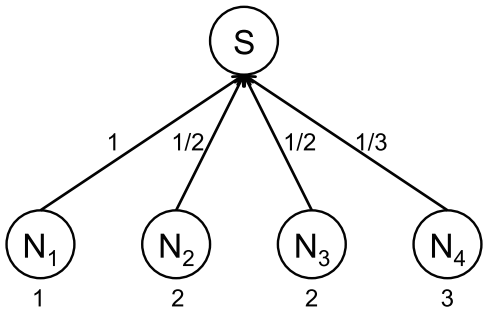

図 2　各 publisher から直接収集する場合

Fig. 2　A case of collecting the sensor data stream directly from the publishers

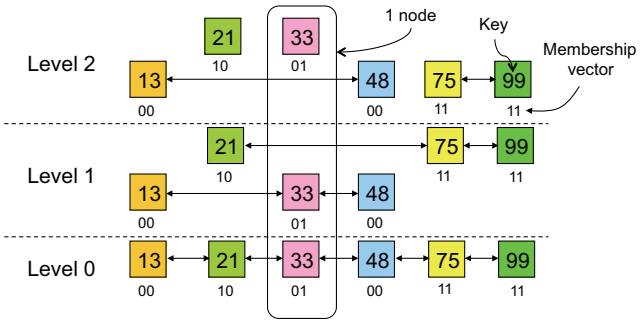

図 3　Skip Graph

Fig. 3　Skipt Graph

がよく用いられる．

$$FI = \frac{\left(\sum_{i=0}^{n} L_i\right)^2}{n \sum_{i=0}^{n} L_i^2} \tag{5}$$

$0 \leq FI \leq 1$ であり，$FI = 1$ であれば，$L_0 = \cdots = L_n$ となる．FI が 1 に近いほど，負荷が分散されていることを示す．

本研究の目的は，システム全体の通信負荷を抑えつつ，各 publisher でデータを集約しながら subscriber が収集することである．よって，目的関数は SL および $1 - FI$ となり，これらの値が最小となるように収集経路を決定する．本問題では，受信したセンサデータストリームを他の publisher へ送信できるものとし，各 publisher が送信するセンサデータを決定する．

図 1 の例において，$\alpha = \beta$ かつ publisher から直接収集する場合を**図 2** に示す．枝付近の数字は，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示す．図 1 の例では，$SL = 4.667$，$FI = 0.617$ となる．

3. Skip Graph を用いた周期的なセンサデータ収集システム

3.1 Skip Graph

本研究では，[7] と同様，Skip Graph を用いた TBPS のためのオーバレイを用いることを検討するため，以下ではまず，ベースとなる Skip Graph について概要を述べる．Skip Graph [8] 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．Skip Graph 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．Skip Graph 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ける距離が大きく，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は平均 $\log N$ となる．

3.2 PUSH による収集方法

文献 [7] は，Skip Graph をベースとした分散 TBPS 方式である．この方法は，同一トピックに対する publisher と subscriber にあたるノード群が隣接して配置される Skip Graph 構造がオーバレイとして構築される．各ノードは，トピックおよびノード識別子を，Skip Graph のキーとしてオーバレイを構築する．Skip Graph がベースであるため，publisher と subscriber の数の合計を M としたとき，各ノードが持つオーバレイのリンク数は，平均 $\log M$ となる．したがって，データ転送のネットワークセッションの数を，トピックに関するノード数に対し，対数オーダに抑えて TBPS を構成することができる．

この方法では，各センサが自律的に配信を実行する publisher となる．すなわち，データの発信側が契機となってデータが subscriber へ送信される．いわば，PUSH 型の収集方法である．

この方法では，publisher が多数存在し，それぞれが独自の周期でデータを配信する場合，同じ周期のデータであっても，異なるメッセージとして送信せざるを得ず，メッセージ数が増大してしまう．

3.3 PULL による収集方法

文献 [7] は，分散 TBPS 方式において，データの収集側が契機となって，データを収集する方法（PULL と呼

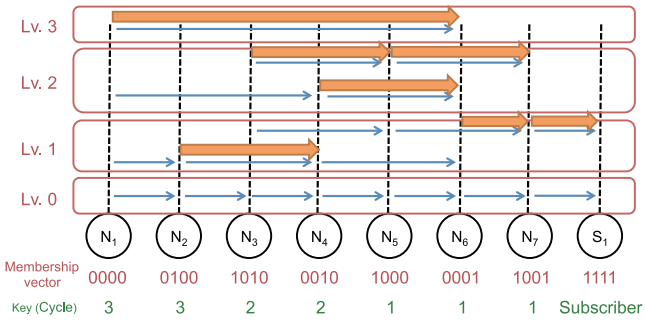

図 4　時刻 6 における転送経路

Fig. 4　Forwarding paths at time 6

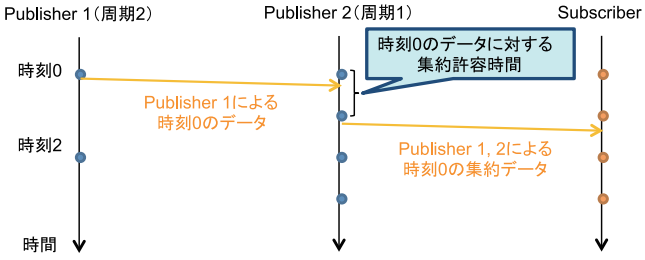

図 5　データ集約の流れ

Fig. 5　Flow of data collection

ぶ）も考えられる．たとえば，subscriber の一つが一定周期で Skip Graph 上の範囲検索を用いて同じトピックを持つ publisher を検索し，観測データを収集したのち，他の publisher に送信すれば良い．Skip Graph 上で範囲検索をスケーラブルに行う方法として SFB [9] がある．SFB では，検索条件に合致する範囲にあるノードが持つデータを，木構造で集約しながら収集する．PULL では，ノードあたりのメッ セージ数は 2 に抑えることができ，検索と収集に必要となるトラヒックの総量の平均は，$2M$ となる．

しかし，この方法では，subscriber が収集周期を決定しなければならない．すなわち，細かい周期で発信しているセンサによるセンサデータが存在するにもかかわらず収集されない状況や，収集周期が長いにもかかわらず，短い周期で収集してしまい，同一の冗長な結果が収集されてしまう状況が生じ得る．

3.4　集約しながら PUSH する方法

上記の課題を解決するセンサデータ収集手法として，各 publisher がセンサデータを PUSH で，かつ，周期ごとに集約しながら，subscriber へ転送する方法を提案する．具体的には，Skip Graph において，同じトピックで publish するノード群が，発信頻度が高いほど subscriber に近い位置に並ぶよう，キーをトピック，発信頻度（周期），ノード識別子の組とする．publisher から subscriber に対してセンサデータを送信する際，Skip Graph の階層構造を用いて，subscriber のキーを超えない最大値のキーを持つノードへのリンクを用いる．また，他のノードから受信したデータと，自ノードが発信するデータを集約し，一つのメッセージとして送信する．**図 4** は，周期をキーとして降順に並べた場合において，提案手法によるセンサデータ収集の例を示している．各レベルの細い矢印は Skip Graph のリンクを示し，太い矢印は時刻 6 におけるデータ転送経路の例である．

データ集約においては，他の publisher からのデータを待つ許容時間の設定が考えられる．一般的には次の収集時刻までなどが考えられ，**図 5** に例を示す．提案手法は，センサデータを集約しながら PUSH により転送するため，メッセージ数を削減しつつ，センサが持つ観測・配信周期でデータを集約することが可能となると考えられる．

4.　評価

本研究では，3 章の提案手法をシミュレーションにより評価した．

4.1　シミュレーション環境

本シミュレーションでは，ノード数を 100 ～ 500 台で 5 段階に変化させた．そのうち，subscriber は 1 台である．各収集対象ノードの周期は，1 ～ 10 の 10 パターンからランダムに決定した．10 パターンの周期の最小公倍数は 2520 で，時刻 0 ～ 2519 の収集を繰り返す．また，各ノードの Skip Graph における membership vector は，32 ビットでランダムに決定した．ルーティングアルゴリズムについては，各ノードは subscriber に最も近いキーのノードへ，高いレベルのリンクから使用した．本シミュレーションでは各ノードが集約可能なメッセージ数に上限を設定せず，ノード間のメッセージ送受信による遅延も生じないものとした．

比較手法については，提案手法でキーの昇順，降順に並べる場合において，それぞれ集約のあり，なしの 4 つの場合を比較した．また，単純な手法として，2 章で述べたすべてのノードが subscriber へ直接送信する Server 法，収集周期の長いノードから短いノードへ降順に集約しつつ送信し，最後のノードが subscriber へ送信する Chain 法とも比較した．Chain 法の例を**図 6** に示す．各手法において，時刻 0 ～ 2519 の最大瞬間負荷，最大負荷，総負荷 SL，Fairness Index（FI），最大ホップ数，平均ホップ数を 10 回測定し，それらの平均値を算出した．各項目の詳細は次節で述べる．

4.2　シミュレーション環境

最大瞬間負荷の結果を**図 7** に示す．最大瞬間負荷は，各ノードの単位時間における負荷のうち，最大の値である．特定の時刻に関係するすべての publisher からのメッセージを受信するため，Server 法では subscriber の瞬間負荷がノード数に比例して最大となる．提案手法においても，同

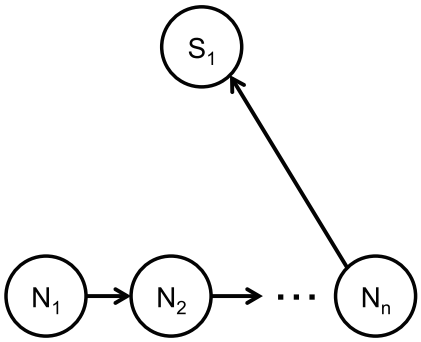

図 6　比較手法: Chain 法

Fig. 6　Comparative methods: Chain method

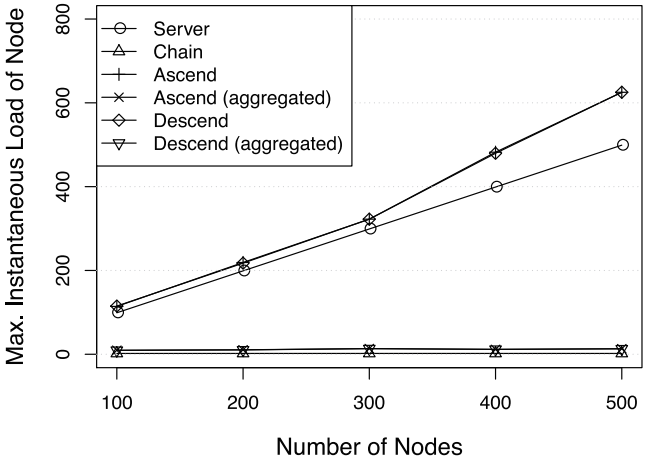

図 7　最大瞬間負荷

Fig. 7　Maximum instantaneous loads of nodes

じノードへ送信するメッセージを集約しない場合，キーの昇順，降順ともに最大瞬間負荷は高い．これは，特定の時刻に関係するすべての publisher からのメッセージを個別に送受信していることに加え，その時刻に本来関係しないノードもメッセージを中継する可能性があるためである．一方，Chain 法では各ノードが単位時間に送信，受信するメッセージ数はそれぞれ最大で 1 であるため，最大瞬間負荷は 2 となる．提案手法で同じノードへ送信するメッセージを集約する場合，各ノードが単位時間に送信するメッセージ数は最大で 1 となるため，同じノードへの冗長なメッセージ送信を抑制し，最大瞬間負荷は Chain 法の次に低い．

最大負荷の結果を**図 8** に示す．最大負荷は，各ノードの時刻 0 ～ 2519 における負荷のうち，最大の値である．最大負荷は前述の最大瞬間負荷と同様の傾向があり，Server 法，提案手法で集約しない場合の値は高く，Chain 法，提案手法で集約する場合の値は低い．

総負荷の結果を**図 9** に示す．総負荷は，全ノードの時刻 0 ～ 2519 における負荷の合計である．特定の時刻に関係する publisher からのメッセージを直接受信するため，総負荷は Server 法が最小となる．一方，提案手法で同じノードへ送信するメッセージを集約しない場合，中継ノードに

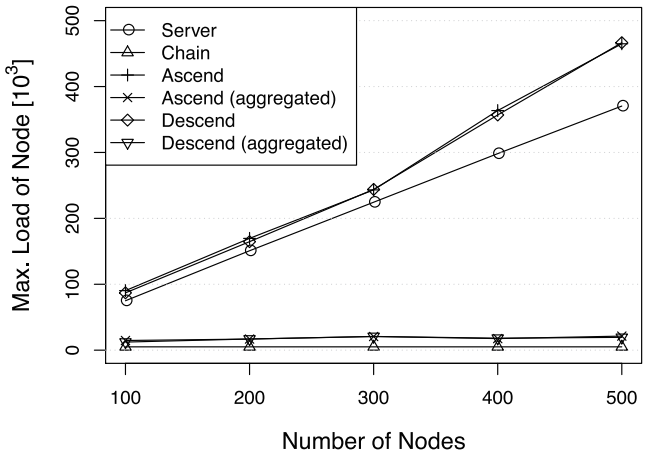

図 8　最大負荷

Fig. 8　Maximum loads of nodes

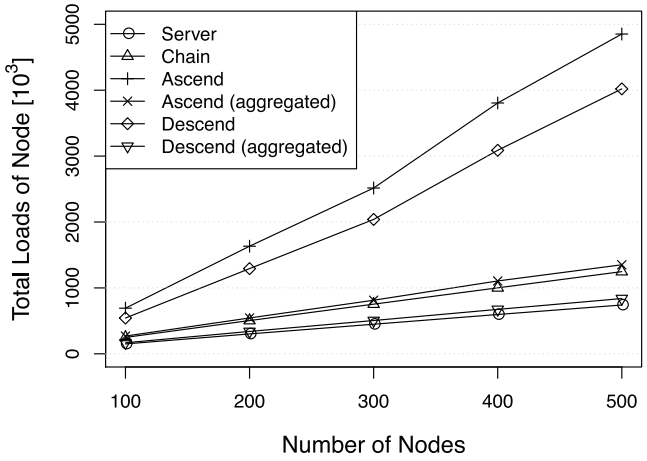

図 9　総負荷

Fig. 9　Total loads of nodes

よって送信，受信の回数が増えるため，総負荷は大きく増加する．特にキーが昇順の場合，周期が長く，その時刻に本来関係しないノードが中継する可能性が高いため，総負荷は最も高い．一方，Chain 法と提案手法で集約する場合，総負荷は低い．ただし，その時刻に本来関係しないノードが中継する可能性が高いため，Chain 法と提案手法でキーが昇順の場合は総負荷が増加する．提案手法でキーが昇順の場合の総負荷は Server 法の次に低く，その差は小さい．

Fairness Index の結果を**図 10** に示す．Fairness Index は，各ノードの時刻 0 ～ 2519 における負荷の偏りを示す．Server 法と提案手法で集約しない場合は特定のノードに負荷が集中するため，Fairness Index は低い．また，特定の時刻に関係しないノードも含めて，単位時間に送信，受信するメッセージが最大で 1 となるため，Fairness Index は Chain 法が最も高い．Fairness Index は次に提案手法でキーが昇順の場合が高いが，前述の総負荷の結果より，その時刻に本来関係しないノードのメッセージ送受信によって負荷の偏りが小さくなっている．提案手法でキーが降順の場合は昇順の場合よりも Fairness Index が低いが，冗長

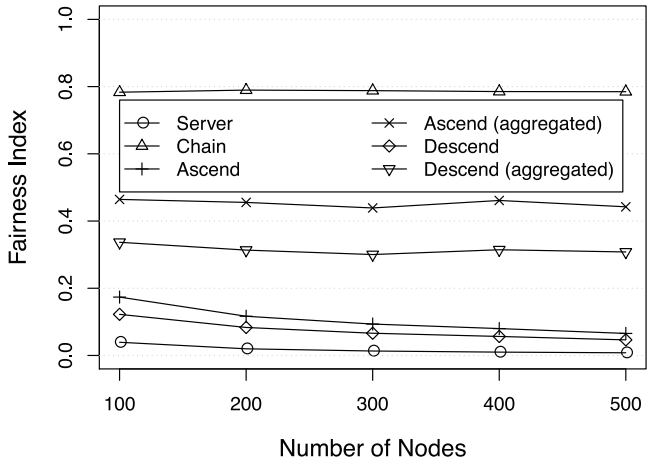

図 10　Skip Graph
Fig. 10　Skipt Graph

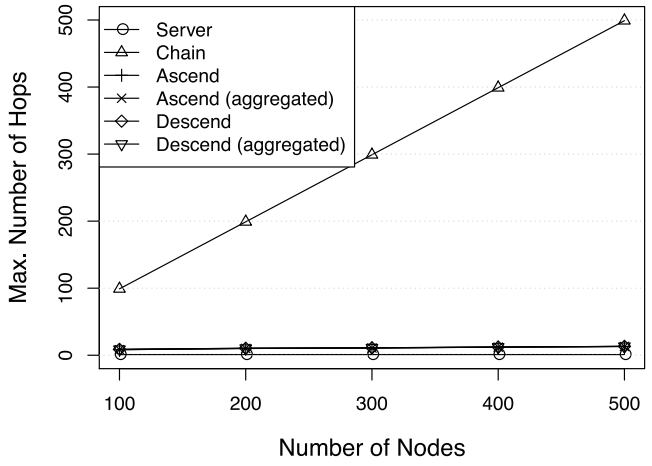

図 11　最大ホップ数
Fig. 11　Maximum number of hops

なメッセージ送受信の発生を抑制しつつ，ノード数が増えても一定の Fairness Index を維持している．

最大ホップ数の結果を**図 11** に示す．最大ホップ数は，各 publisher から subscriber までのホップ数のうち，最大の値である．Server 法では publisher から直接受信するため，最大ホップ数は 1 となる．また，周期の長いノードから短いノードへ降順に転送するため，Chain 法の最大ホップ数がノード数に比例して最も高い．他の手法では Skip Graph の階層構造のリンクを利用するためにホップ数の増加を抑制しており，500 ノードかつ提案手法で昇順の場合は約 13.1 で，降順の場合は約 13.2 である．

平均ホップ数の結果を**図 12** に示す．平均ホップ数は，各 publisher から subscriber までのホップ数の平均値である．平均ホップ数は前述の最大ホップ数と同様の傾向があり，500 ノードかつ提案手法で昇順の場合は約 6.5 で，降順の場合は約 5.4 である．

以上の結果から，各ノードの負荷やホップ数の点で，提案手法においてキーで降順に並べる場合が有効であると考えられる．

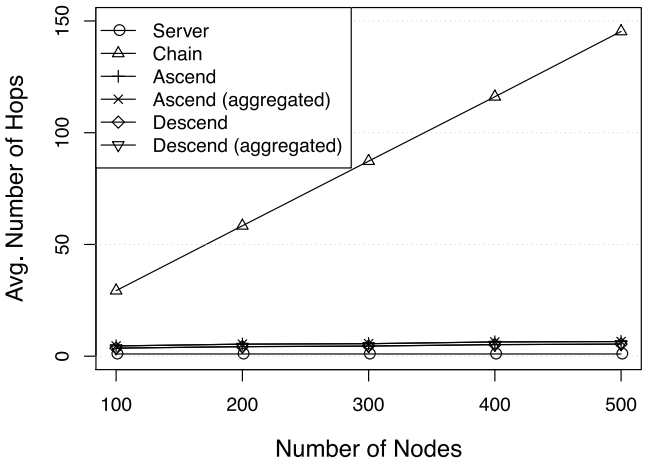

図 12　平均ホップ数
Fig. 12　Average number of hops

5. まとめ

本研究では，Skip Graph を用いたスケーラブルなセンサデータ収集システムとして，トピックと発信頻度に基づいてオーバレイネットワークを構築し，複数の publisher からのセンサデータを周期に基づいて集約しつつ収集する手法を検討した．検討手法はシミュレーションにより評価し，周期をキーとして降順に並べる場合が有効であることを確認した．

今後の課題としては，publisher からのデータを集約するために待つ許容時間の検証や，ノード間のデータ送受信に遅延が生じる環境での評価が挙げられる．

謝辞

本研究の一部は，NICT・大阪大学共同研究「大規模分散コンピューティングのための高機能ネットワークプラットフォーム技術の研究開発」，および JSPS 科研費 16K16059 の助成による成果である．

参考文献

[1] S. Hodges, et al., Prototyping Connected Devices for the Internet of Things, *IEEE Computer*, pp. 26–34, 2013.
[2] MQTT Version 3.1.1, `http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.pdf` (accessed Mar. 31, 2016).
[3] Advanced Message Queuing Protocol, `http://www.amqp.org` (accessed Mar. 31, 2016).
[4] ROS, `http://www.ros.org/` (accessed Mar. 31, 2016).
[5] M. Castro, et al., SCRIBE: A Large-scale and Decentralized Application-Level Multicast Infrastructure, *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, 2002.
[6] S.Q. Zhuang, et al., Bayeux : An Architecture for Scalable and Fault-tolerant Wide-area Data Dissemination, in *Proc. International Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2001.
[7] R. Banno, et al., Designing Overlay Networks for

Handling Exhaust Data in a Distributed Topic-based Pub/Sub Architecture, *Journal of Information Processing*, Vol.23, No.2, 2015.

[8] J. Aspnes, G. Shah, Skip Graphs, *ACM Transactions on Algorithms (TALG)*, vol. 3, no. 4, 2007.

[9] R. Banno, et al., SFB: A Scalable Method for Handling Range Queries on Skip Graphs, *IEICE Communications Express*, Vol. 4, No. 1, 2015.