

収集周期の異なるセンサデータストリームのための 上限ホップ数を設けたP2P型配信システムの実現と評価

石 芳正^{1,a)} 川上 朋也^{1,2,b)} 義久 智樹^{1,c)} 寺西 裕一^{1,3,d)}

受付日 2015年6月1日, 採録日 2015年11月5日

概要: 我々の研究グループでは, センサの観測データが連続的に流れるセンサデータストリームの配信に際し, 複数の配信先がそれぞれ異なる周期のセンサデータを要求する環境を想定し, P2P 技術により通信負荷を分散する P2P 型センサデータストリーム配信システムを研究してきた. これまで配信系全体の負荷の公平性のみに着目した LCF 法, LLF 法を提案してきたが, それらの手法では同一周期の配信先が多数存在する場合に配信時のホップ数が大きくなり, 配信遅延が増大するという問題があった. この問題に対し, 配信ホップ数に上限を設ける LLF-H 法を提案し, シミュレーションによる評価を行ってきたが, 実機における処理のオーバーヘッドや, 実ネットワークにおける遅延の影響は考慮されていない. 本研究では, LLF-H 法におけるホップ数の制限が配信遅延や公平性に与える影響について PIAX テストベッドが提供している実機試験環境上で評価を行い, 上限ホップ数を制限した場合であっても既存手法の LLF 法と比較して負荷の公平性を大きく損ねることなく配信遅延時間を抑制できることを確認した.

キーワード: センサデータストリーム, 配信周期, 分散処理, P2P

Development and Evaluation of A P2P Streaming System with Hop Limit for Sensor Data Stream with Different Collection Cycles

YOSHIMASA ISHI^{1,a)} TOMOYA KAWAKAMI^{1,2,b)} TOMOKI YOSHIHISA^{1,c)} YUUCHI TERANISHI^{1,3,d)}

Received: June 1, 2015, Accepted: November 5, 2015

Abstract: Our research team proposed some sensor data streaming methods based on peer-to-peer techniques named LCF, LLF and LLF-H for distributing communication loads when delivering sensor data streams with different data collection cycles. The LCF method and the LLF method mainly focus on the fairness of communication load. However, when many distribution nodes have a same collection cycle, the number of hops to distribute becomes large, and the transmission delay increases by those techniques. To suppress the delay time, we proposed the LLF-H method which add a upper limit to the number of hops on delivering, and evaluated the LLF-H method by the simulation. In this paper, we evaluate the behavior of the transmission delay and the fairness on the LLF-H method with many destination nodes which require same collection cycle, by using the PIAX testbed. From the evaluation result, we confirmed that the transmission delay could be suppressed, without spoiling the fairness of load greatly as compared with the existing LLF method, even if the maximum number of hops was limited to several hops.

Keywords: Sensor data stream, Delivery cycle, Distributed processing, P2P

¹ 大阪大学サイバーメディアセンター
Cybermedia Center, Osaka University, Ibaraki, Osaka 567-0047, Japan
² 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, Ikoma, Nara 630-0192, Japan
³ 国立研究開発法人情報通信研究機構
National Institute of Information and Communications Technology, Koganei, Tokyo 184-8795, Japan

1. はじめに

近年, センサデバイスの小型化・低廉化・高機能化にとも

a) ishi.yoshimasa@ime.cmc.osaka-u.ac.jp
b) kawakami@is.naist.jp
c) yoshihisa@cmc.osaka-u.ac.jp
d) teranisi@nict.go.jp

ない、それらセンサデバイスを用いた環境観測機器が容易に入手可能となっている。また、IoT (Internet of Things) や M2M (Machine to Machine) といった、小型デバイスの環境中への導入とネットワークを介したモノどうしの相互連係動作についての研究開発もさかんに行われており、それら小型デバイスが持つセンサデバイスが観測した様々な観測データ、センサデータを相互に利活用するサービスの実現が期待されている。こうしたサービスでは、センサデバイスより得られた環境状態の変化や観測値を用いて、リアルタイム・準リアルタイムで状況を判断することによって、状況に応じたサービスが実現される。

センサデバイスは、多くの場合センサデバイス固有の周期により、一定間隔でセンサデバイス周囲の環境が観測され、その観測結果がセンサデータとして周期的に出力される。たとえば気象センサの場合は数秒から数分に1回、カメラデバイスの場合では、1秒間に30枚から60枚程度の画像が得られる。これらのセンサデバイスから逐次送出される一連のセンサデータは、センサデータストリームと呼ばれ、センサデータストリームを直接監視したり解析処理したりすることにより、リアルタイム・準リアルタイムでの監視業務や環境モニタリングなどに利用できる。また、1台のセンサデバイスから得られるセンサデータストリームは、ネットワークを介して様々な利用先へと送信することにより、1つのセンサデータストリームを複数のサービス、たとえば気象状況を用いた緊急避難通知サービスや大気汚染状況の記録・監視サービスといった多岐にわたるサービスへの展開が期待できる。同時に、1つのセンサデバイスを複数のサービスや利用者で共用することで、センサデータストリームの利用者らはそれぞれ個別に観測網を構築する必要がなくなる。これにより、センサ機器の重複設置を避け、効率の良い観測網を構築することができる。このようなセンサデータストリームをネットワークを介して複数の利用先に送り届けることはセンサデータストリーム配信と呼ばれている。

一般的なデータストリーム配信においては、これまでビデオストリームの配信を中心として、配信時の通信負荷を分散させる様々な手法が研究されている [1], [2], [3], [4], [5], [6], [7], [8]。これらの既存研究では、あるデータストリームを複数の利用先に配信する場合には、データを受信した配信先がさらに他の配信先に再送信することで、データストリームの配信元に集中していた通信負荷を分散させる手法が主となっており、すべての配信先に同一のデータストリームを配信する場合に有効に機能する。しかしながら、センサデータストリーム配信においては、配信先により要求するセンサデータの周期が異なる場合、すなわち時間方向で異なる解像度のセンサデータが要求される場合が想定される。たとえば街頭に設置されている屋外カメラが出力する映像ストリームを用いて動体検出を行う場合、自動車

を検出対象とする場合にはその移動速度が速いことから、短いフレーム周期のセンサデータストリームが必要となる。その一方で、検出対象が人物の場合は車と比べて移動速度が遅いため、自動車を対象とする場合と比較して長いフレーム周期のセンサデータストリームであってもよい。同時に、長いフレーム周期のセンサデータストリームを用いることで、必要となるネットワーク帯域や受信処理時の消費電力を削減可能となる。このように、センサデータストリーム配信においては、同じセンサデバイスが収集したセンサデータであっても、その用途や配信先、状況に応じて収集周期の異なるセンサデータストリームが要求される。これに対して、既存のビデオストリーム配信技術を用いて同一のセンサデータストリームをすべての配信先に配信しつつ、各配信先において不要なセンサデータを破棄する手法も考えられるが、不要なセンサデータが配信されることにより、ネットワーク上に利用されることがない無駄なセンサデータが流れることとなり、ネットワーク帯域や配信先の処理能力が無駄に消費されることとなる。

我々の研究グループでは、配信先がそれぞれ異なる配信周期を要求する場合において、配信先が必要としていないセンサデータの配信を避けつつ、配信元・配信先（双方をまとめてノードと表記する）の送受信負荷を分散させるセンサデータストリーム配信経路を構築する手法として、LCF 法 (Longest Cycle First; 最長周期優先) と LLF 法 (Lowest Load First; 最小負荷優先) を提案している。先行研究においては、シミュレーション評価 [9], [10] と、それぞれの手法を実装のうえ、国立研究開発法人情報通信研究機構 (NICT) により運用されている PIAX テストベッド*1を用いた実機上での挙動評価 [11] を行っている。これらの評価においては、LCF 法・LLF 法では未考慮であった、あるセンサデータが配信元から配信先に届くまでに要する配信遅延時間の問題、特に同一の配信周期の配信先が数多く存在する場合において、すべての配信先に到達するまでに要するホップ数と配信遅延時間が増大してしまうという問題が判明した。これに対し、新たに上限ホップ数を設けることにより、配信元や配信先の公平性を維持しつつホップ数や配信遅延時間を抑制することを目指した LLF-H (Lowest Load First considering Hops) 法を提案している [12]。

この先行研究においては、ノードが送受信するメッセージ数をもとにした公平性の変化や、ホップ数制限を加えた場合での各ノードへの所要ホップ数の差異についてシミュレーション評価が行われており、ホップ数を制限した場合であっても適切な制限値とすることで最大ホップ数を抑制しつつ十分な負荷分散が可能であることが示されている。しかしながら、シミュレーション評価では、実環境において生じる処理のオーバヘッドや、ネットワークにお

*1 PIAX テストベッド <https://piax.jgn-x.jp/>

ける遅延の影響を反映できておらず、実システムとして構築した場合のシステムの挙動は未知数となっている。たとえば、上限ホップ数を小さく設定した場合、処理のオーバーヘッドが影響し、想定に反して遅延が大きくなる可能性が考えられる。また、センサデータのサイズが大きい場合、ネットワークの帯域に応じた遅延が生じることが考えられる。LLF-H 法では、ノード間の公平性と配信遅延時間の間でトレードオフを上限ホップ数の設定により実現できるが、適切な上限ホップ数を選択するためには LLF-H 法を実装したうえで、実環境上での挙動を知る必要がある。

そこで本研究では、LLF-H 法を実環境で動作する P2P 型のセンサデータストリーム配信システムとして実装し、通信路や端末の特性の影響を受ける実環境においてシミュレーションと同様の有効性が得られるか、また、実環境におけるシステムがどのような挙動を示すか NICT が提供している PIAX テストベッドを用いて確認する。

2. センサデータストリーム配信手法

2.1 関連研究

1 つのデータソースから得られるデータを、複数の配信先に逐次配信するデータストリーム配信手法として、これまで主にビデオストリームを対象とした様々な配信手法が提案されてきている。これらの手法を、その実現階層により大きく分けると、ネットワーク層で同一のデータを複数の配信先に配信する IP マルチキャストと、アプリケーション層において同様の機能を実現する Application Layer Multicast (ALM) に分けられる。IP マルチキャストにおいては、ネットワーク層において、データの複製や配信経路の構築を行うため、ネットワークリソースの利用効率が良いが IP マルチキャストに対応したネットワーク機器が不可欠となる。これに対して ALM では、データの複製や配信経路の構築をネットワーク末端の各ノードで行うため、ネットワーク層においてはユニキャストのパケットを扱うのみとなる。これにより、ネットワーク機器の IP マルチキャストへの対応の有無に関係なく、アプリケーションをインストールするだけで利用できるため、既設のネットワーク環境に左右されず導入が容易という利点がある。

ALM を実現する代表的な手法として、データストリームの配信元 1 つと複数の配信先の間でマルチキャスト木と呼ばれる、データストリームの上流・下流が明確に分かるツリー状の配信経路を構築する手法があげられる。たとえば、文献 [1] では、複数の配信元ノードからの遅延や各ノードのリンク数（次数）の制約を満たしつつ、マルチキャスト木を構築する手法が提案されている。負荷分散を考慮した手法として、ZIGZAG 法では、複数のノードをまとめてクラスタグループとし、それらクラスタグループ間でマルチキャスト木を構築し、木における各層に含まれるクラスタグループ数を平衡化することで負荷分散を行っている

[2]。また、一部の端末間の通信遅延のみを考慮することで、ZIGZAG 法と比べて高い負荷分散を実現した LAC (Locality Aware Clustering) [3] や、物理的なネットワーク構造の情報を取得し、ノード間の通信遅延時間を用いて特定の端末への負荷集中を回避する MSMT/MBST 法 [4] などがあげられる。

一方で、配信先ノード間でデータストリームやその一部分を相互にやりとりすることで配信を行う Peer-to-Peer 型の配信手法も存在する。文献 [5] では、BitTorrent の技法を取り入れた配信手法において、データストリームを分割した小要素に対して、送受信優先度を調整することによりデータストリームの途切れを抑制する手法が提案されている。また、Scribe [6] では、分散ハッシュテーブルの一種である Pastry [13] の探索経路を逆順にたどることで Pastry の P2P ネットワーク上にマルチキャスト木を構築する。SplitStream [7] は、複数の Scribe マルチキャスト木を使うことで、ノード間の負荷分散を行っている。

しかしながら、これらの手法はすべて配信元ノードが送出したデータストリームを、そのまま各配信先ノードに配信する手法であるため、センサデータストリーム配信に生じる各々の配信先ノードによって異なる周期のセンサデータストリームが要求される場合では、その周期ごとに対応する配信ネットワークや配信木を構築しなければならない。配信元ノードは、それらの独立した配信木それぞれにセンサデータストリームを送信することとなるため、同一のセンサデータを複数回送信しなければならない場合も生じる。同時に、ネットワーク上には重複したセンサデータが流れることとなり、配信元ノードの負荷の増加とネットワーク帯域を無駄に消費することとなる。既存のデータストリーム配信手法により、同一のセンサデータストリームをすべての配信先に配信し、各配信先において不必要なセンサデータを破棄する手法も考えられるが、配信先によっては不必要なセンサデータが配信されるため、それにより消費されるネットワーク帯域や配信先の処理能力が無駄となる。

以上より、既存のデータストリーム配信手法では、異なる周期を扱うセンサデータストリーム配信において、ノード間において負荷集中を避けつつ、不必要なセンサデータの配信によるネットワークリソースの無駄な消費を避けるという 2 つの要件を満たすことは難しい。

2.2 配信モデル

図 1 に、我々のグループが想定しているセンサデータストリーム配信のモデルを示す。図中 S は、センサがつながったサーバを示しており、センサから周期的に観測値を取得している。以下、このサーバを配信元ノードと呼ぶ。また、センサから得られた観測値はセンサデータと呼ぶ。配信元ノードは、センサごとに存在し、センサと 1 対 1 で

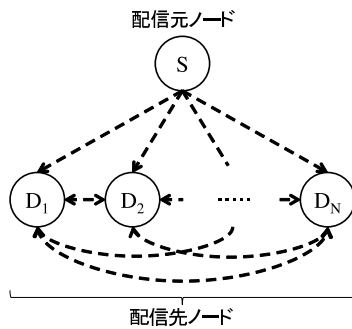


図 1 想定モデル

Fig. 1 Assumed model.

表 1 収集するセンサデータの例

Table 1 An example of sensor data collection.

Time	0	1	2	3	4	5	6	...
D_1 (Cycle=1)	○	○	○	○	○	○	○	...
D_2 (Cycle=1)	○	○	○	○	○	○	○	...
D_3 (Cycle=1)	○	○	○	○	○	○	○	...
D_4 (Cycle=3)	○			○			○	...
D_5 (Cycle=3)	○			○			○	...
D_6 (Cycle=3)	○			○			○	...

対応している。 $D_1 \sim D_N$ は、 N 台のセンサデータ受信端末を示しており、これら受信端末を配信先ノードと呼ぶ。配信先ノードどうしは相互にセンサデータを受け渡すことができる。以下、これら受信端末を配信先ノードと呼ぶ。これらの配信元ノードと配信先ノードは、オーバーレイネットワークにより接続され、相互に探索可能な状態とする。

表 1 は、配信先ノード $D_1 \sim D_6$ が収集するセンサデータの収集周期 (Cycle) と、受信するセンサデータを ○ で示している。 D_1 から D_3 は、収集周期が 1 であり、配信元ノードが収集したすべてのセンサデータを受け取る。同様に D_4 から D_6 は、時刻 3 ごとにセンサデータを受け取る。

収集周期は各配信先ノードが保持しており、センサデータストリームの配信を開始する前に配信先ノードが配信元ノードに自身の収集周期を通知する、あるいは配信元ノードが自身のセンサデータを必要としている配信先ノードから収集周期を収集する。これらは、ノード間を接続しているオーバーレイネットワークを介して対象のノードを探索することにより実現する。

2.3 通信負荷モデル

多くの場合、各ノードの通信負荷は送受信するセンサデータストリームにおける単位時間あたりのセンサデータの数に比例すると考えられる。ここで、ノード D_x がノード D_y に送信する単位時間あたりのセンサデータ数を $R(x, y)$ と表すと、ノード D_r の単位時間あたり負荷 L_r は式 (1) となる。 α は、センサデータを 1 つ受信した場合に生じる負荷、 β は、センサデータを 1 つ送信した場合に生じる負荷を示し、前項はノード D_r が他のノードからセンサデータ

ストリームを受信することによる単位時間あたり負荷、後項はノード D_r が他のノードにセンサデータストリームを送信することによる単位時間あたり負荷となる。なお、 D_0 は S を表すものとし、 N は配信先ノードの数とする。

$$L_r = \alpha \sum_{i=0}^N R(i, r) + \beta \sum_{i=0}^N R(r, i) \quad (1)$$

2.4 基本的なアイデア

基本的な方針として、配信先ノードが受信したセンサデータストリームに含まれるセンサデータのうち、他の配信先ノードが要求する周期に合致するセンサデータをそれらの配信先ノードに再配信することにより通信負荷を分散させる。たとえば、表 1 では、時刻 0、時刻 3、時刻 6 のセンサデータはすべての配信先ノードが要求しており、ある配信先ノードが他の配信先ノードにセンサデータを再配信することで、サーバが直接配信することなくそれらのセンサデータを必要としている配信先ノードに配信できる。同様に時刻 1、時刻 2、時刻 4、時刻 5 のセンサデータについては収集周期 1 の配信先ノード間で相互に再配信が可能である。サーバから直接配信する場合、通信負荷がサーバに集中するが、この例のように、センサデータを受信した配信先ノードが他の配信先ノードに再配信することで、通信負荷の分散が期待できる。

2.5 配信経路構築手法

ここでは、LLF 法と LLF-H 法の経路構築手順について述べる。

2.5.1 LLF 法

LLF 法は、各ノードの収集周期から各ノードに加わる通信負荷を 2.3 節の負荷モデルにより推定し、ある時刻のセンサデータを受け取る際には、同時刻のセンサデータを受信しているノードの中から負荷の推定値が最小となるノードを選択し、そのノードからセンサデータを受け取ることで通信負荷を分散する。負荷を推測することにより LLF 法では、LCF 法より均等に負荷が分散されることが期待できる。配信経路を求める際、その計算量を抑制するため、再送信の方向は長周期の配信先ノードからより短周期の配信先ノード方向のみとする。

LLF 法では、各配信先ノードが要求する収集周期に基づいて配信経路を構築していく。あるセンサデータを配信する際に、配信先ノードがそのセンサデータを受信するパターンは、配信先ノードの収集周期の最小公倍数周期で繰り返される。たとえば、表 1 の場合、配信先ノードの収集周期の最小公倍数の 3 周期で同パターンの配信が繰り返されることとなる。このため、この繰り返される配信パターンを 1 配信周期として、この配信周期内の各周期での配信経路を構築していくこととなる。まず最初に、各ノードが 1 配信周期中に受信しなければならないセンサデータによ

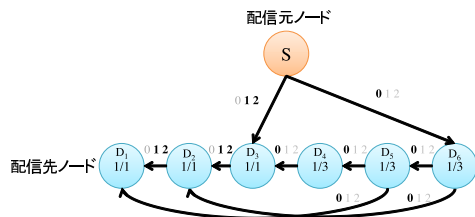


図 2 LLF 法による配信経路

Fig. 2 The delivery route in the LLF method.

る受信負荷を求める。センサデータ 1 個あたりの送信負荷を 1, 受信負荷を 1 とした場合 ($\alpha = \beta = 1$), 各ノードに加わる負荷は 3, 3, 3, 1, 1, 1 となる。各周期において, 最長周期の配信先ノードは, 配信元ノードからセンサデータを受け取り, 2 番目に長い周期の配信先ノードに再配信することとなるため, 次にそれらによる送信負荷を求める。たとえば, 周期 0 では D_6 が配信元ノードからセンサデータを受け取り, D_5 に再配信する。周期 1, 2 では D_3 が配信元ノードからセンサデータを受け取り, D_2 に再配信する。これらの最長周期の配信先ノードに生じる送信負荷を加算すると, 各配信先ノードの負荷は 3, 3, 5, 1, 1, 2 となる。最後に, 各周期においてセンサデータの取得元が決定していない配信先ノードの取得元を決定してゆく。たとえば, ここまでの手順で, 周期 0 では D_6 は配信元ノードから受け取る, D_5 は D_6 から受け取ることを決定している。 D_4 は, この時点で経路が決定している負荷が最小のノードを選択し, 周期 0 のセンサデータを受け取る。これは D_5 となる。この再配信による送信負荷を加算すると, 各配信先ノードの負荷は 3, 3, 5, 1, 2, 2 となる。同様に D_3 は D_4 から受け取り, 各配信先ノードの負荷は 3, 3, 5, 2, 2, 2 となる。これを D_2 , D_1 についても行い, 周期 0 の経路が決定される。この時点での, 各配信先ノードの負荷は 3, 3, 5, 2, 3, 3 となる。以降, 残りの周期 1, 2 についても負荷が最小のノードを選択することで経路を決定し, 表 1 の場合における LLF 法による配信経路は図 2 のとおりとなる。各配信先ノードに加わる負荷は 1 配信周期の期間で 3, 5, 5, 2, 3, 3 となる。

2.5.2 LLF-H 法

LLF-H 法では, センサデータの再配信を許すホップ数に上限を設けることで, 配信遅延の抑制を目指す。具体的には, LLF 法において負荷が最小のノードを選択する過程において, 候補となる配信先ノードから再配信を受けた場合に事前に与えられたホップ数の上限を超えないか判定を行う。ホップ数の上限を超える場合は, 次の候補で同じ判定を行い, 条件を満たす配信先ノードがない場合は各周期で最長の周期を持つノードから受け取る。

LLF 法での例示と同様に, 配信先が要求する周期が表 1 の場合を例として配信経路の構築手順を考える。配信元ノードを起点としたホップ数の上限を 3, $\alpha = \beta = 1$ とし

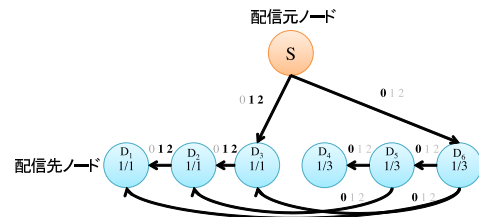


図 3 LLF-H 法による配信経路

Fig. 3 The delivery route in the LLF-H method.

た場合, 各ノード D_1 から D_6 に加わる受信負荷は LLF 法と同様に 3, 3, 3, 1, 1, 1 となる。

次に, 各周期において, 最長周期の配信先ノードが, 2 番目に長い周期の配信先ノードに再配信する経路を候補としてあげる。周期 0 では D_6 より D_5 に再配信し, 周期 1, 2 では D_3 より D_2 に再配信する経路が候補となり, それぞれの周期において, ホップ数は 2 となり, ホップ数上限には達しないため, これらの経路を採用する。これらの最長周期の配信先ノードに生じる送信負荷を加算することで, 各配信先ノードの負荷は 3, 3, 5, 1, 1, 2 となる。

最後に, 各周期においてセンサデータの取得元が決定していない配信先ノードの取得元を決定してゆく。周期 0 では, ここまでの手順で D_6 は配信元ノードより受け取り, D_5 は D_6 から受け取ることを決定しており, 各配信先ノードへのホップ数は, —, —, —, —, 2, 1 となっている。“—”はホップ数未決定を意味する。 D_4 は, この時点で経路が決定している配信先ノード D_5 と D_6 のどちらかより, ホップ数が上限を超えない負荷が最小のノードを選択し, 周期 0 のセンサデータを受け取る。ホップ数では, D_5 と D_6 のどちらから受信してもホップ数上限は超えないため, 負荷が小さい D_5 より受信することとなる。これにより, D_5 に再配信による送信負荷を加算すると, 各配信先ノードの負荷は 3, 3, 5, 1, 2, 2 となる。また, ホップ数については配信元ノードから D_4 までのホップ数は 3 となることから, —, —, —, 3, 2, 1 となる。続く D_3 は, この時点で負荷が最小となっている D_4 から受信しようとするが, その場合, ホップ数が 4 となりホップ数上限を超えるため, 次に負荷が低い D_6 を選択する。これにより, 各配信先ノードの負荷は 3, 3, 5, 1, 2, 3, ホップ数は, —, —, 2, 3, 2, 1 となる。 D_2 は, 同様の手順で D_5 を選択し, 各配信先ノードの負荷は 3, 3, 5, 1, 3, 3, ホップ数は, —, 3, 2, 3, 2, 1 となる。最後に D_1 は D_6 を選択し, 各配信先ノードの負荷は 3, 3, 5, 1, 3, 4, ホップ数は 2, 3, 2, 3, 2, 1 となる。以上で, 周期 0 における全配信先ノードへの配信経路が決定される。

残りの周期 1, 2 についても, 同手順により経路を構築する。具体的には, D_1 は周期 1, 2 のそれぞれで, D_2 より受信することとなり, LLF-H 法による配信経路は, 図 3 のとおりとなる。この配信経路を用いることで, 1 配信周期中に各配信先ノードに加わる負荷は 3, 5, 5, 1, 3, 4 となる。

3. PIAX を用いたセンサデータストリーム配信システムの設計と実装

3.1 PIAX の概要

PIAX [14] は, NICT や大阪大学を中心として研究開発が進められている Java ベースのプラットフォームミドルウェアで, オープンソースソフトウェアとして公開されている*2. PIAX は, P2P オーバレイネットワークとモバイルエージェントの機能が統合された環境を提供する.

PIAX では, 主要な機能をピア (PIAX では, ノード上で動作し, P2P 通信を行う 1 つのプロセスをピアと呼ぶ) 上で動作するエージェントと呼ばれるソフトウェアモジュールにより実装し, それらエージェント間の連携動作によりシステム全体の動作を実現する. ピア間は, PIAX がサポートする各種オーバレイネットワークにより相互に接続されており, 標準では範囲探索を実現する SkipGraph [15] や, 完全一致型探索を行う DOLR [16], 緯度経度を基準とした地理的探索を行う LL-Net [17] をサポートしている. エージェントは, これらオーバレイネットワークの探索能力を利用して, 他ピア上のエージェントを探索・発見し, 発見した後はエージェント間でのメッセージングやエージェントのピア間移動によりシステムの動作を実装する.

PIAX では, エージェント間のメッセージングをエージェントが持つメソッドを呼び出すことで実現している. エージェント間メッセージング機能として, 個別のエージェントにメッセージを伝える 1 対 1 呼び出し (call) と, ある条件に合致するエージェントすべてにメッセージを伝える発見型呼び出し (discovery call) の 2 種類のメッセージング機構を備えている. 1 対 1 呼び出しでは, 各ピアに割り当てられたピア ID と, 各エージェントに割り当てられたエージェント ID を用いて個別のエージェントを指定し, そのエージェントが持つメソッドを実行することでメッセージングを行う. 発見型呼び出しでは, エージェントに与えた任意の属性値を探索対象として DOLR や LLNet などのオーバレイネットワークが持つ探索機能により, その探索条件に合致するエージェントを発見し, 同時にそれらエージェントのメソッドを実行することで 1 対多のメッセージングを実現している. 発見型メッセージング機能を用いることで, エージェントがどのピア上に配置されているかを事前に把握することなく目的とするエージェントにメッセージを通知したり処理を要求することができる.

本研究においては, 配信元ノードと配信先ノードは事前には互いにその接続先情報を持たず, 必要なときに必要なノードが接続する環境を想定しているため, これらの PIAX の発見型呼び出し機能や 1 対 1 呼び出し機能を用いて相互発見や通信を行う.

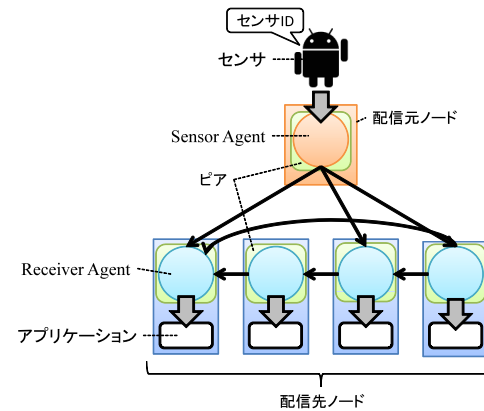


図 4 センサデータストリーム配信システムの構成

Fig. 4 Structure of sensor data stream delivery system.

3.2 センサデータストリーム配信システムの構成

図 4 は, センサデータストリーム配信システムの構成を示している. 1 つのセンサが接続された 1 台の配信元ノードと, その配信元ノードから配信を受ける 1 台以上の配信先ノードにより, 図 4 にある 1 つのセンサデータストリーム配信グループを構成する. 配信元ノードに接続されるセンサには, 文字列からなるユニークなセンサ ID が割り振られているものとする. 各ノード上ではそれぞれ 1 つの PIAX ピアを動作させ, オーバレイネットワークで接続することで相互に通信可能な状態となっている. 配信元ノードのピア上には, センサに対応する SensorAgent を, 配信先ノードのピア上には, ReceiverAgent を配置する. 図中の SensorAgent と ReceiverAgent 間を結ぶ矢印は, エージェント間通信によるセンサデータ配信経路の一例を表している.

3.3 動作シーケンス

本研究では, 初期段階において, 配信元ノードと配信先ノードは互いの存在を知らず, 接続情報を持たない状態から, 必要に応じて接続し, 配信を行う利用形態を想定している. この動作を実現するためには, 配信元ノードが配信先ノードを発見するか, 逆に配信先ノードが配信元ノードを発見する必要がある. 前者は, 街頭に設置された環境センサのように恒常的に設置されていたセンサに対して, 配信先ノードが任意の時点でセンサデータの配信を要求する場合に相当し, 後者では逆に, 移動型のセンサやユーザ端末に内蔵されたカメラのように, 不定期にセンサストリームを生成するセンサに対して配信先ノードが待ち受ける Publish - Subscribe モデルに相当する. これらは, 配信元ノードと配信先ノードの一方に他方を発見させることで実現できるが, ここでは, 配信先ノードが配信元ノードを探索・発見する場合の動作シーケンスを示す.

3.3.1 各ノードの初期設定

まず, 配信元ノードに接続されているセンサのセンサ

*2 PIAX: P2P Interactive Agent eXtensions - <http://www.piax.org/>

ID をその配信元ノードに与え、SensorAgent に設定する。配信先ノードには、センサデータストリーム配信を受けようとするセンサのセンサ ID と要求する収集周期を与え、それらのパラメータを ReceiverAgent に設定する。配信元ノードは、配信先ノードの探索により発見される必要があるため、配信元ノード上の SensorAgent は、センサ ID を DOLR ネットワークに探索対象キーとして登録する。DOLR ネットワークは、PIAX が標準サポートしており、探索条件として与えられた文字列や数値に完全一致する属性値を持つエージェントを発見できるオーバーレイネットワークである。これにより、他エージェントよりセンサ ID を探索キーとして DOLR ネットワークを探索することで、センサ ID に対応する SensorAgent を発見できる状態となる。

3.3.2 ノードの発見と配信経路の作成

配信先ノード上の ReceiverAgent は、自身の識別情報 (pid; ピア ID, aid; エージェント ID) と希望する収集周期 (cycle) を、センサデータストリームを受信しようとしているセンサに対応する SensorAgent に通知する (pre-registerDestination) (図 5)。この通知には、PIAX の発見型呼び出し機能 (discovery call; dc) を使用する。ここでの発見型呼び出しでは、センサ ID を探索対象キーとして DOLR ネットワーク上での探索が行われ、該当するセンサ ID を持つ SensorAgent のみにメソッド呼び出しメッセージが届けられる。これにより、ReceiverAgent は SensorAgent のピア ID やエージェント ID を事前に知ることなく、センサ ID のみで目的の SensorAgent に要求を通知することができる。

次に、SensorAgent は ReceiverAgent から通知を受けた収集周期をもとに、LLF-H 法により配信経路表 (route) を構築する。ここで作成される配信経路表は、1 配信周期内の各周期において配信先・再配信先となる ReceiverAgent のピア ID とエージェント ID の組を記した表であり、SensorAgent と各 ReceiverAgent それぞれ専用の配信経路表となる。構築された各配信経路表は、SensorAgent 用については SensorAgent 自身が保持し、各 ReceiverAgent 用の配信経路表は、エージェントに対する 1 対 1 呼び出し機能 (call; c) を使用し、個々の ReceiverAgent にそれぞれ対応する配信経路表を与える (setRoute)。

ReceiverAgent は、SensorAgent から受け取った配信経路表を「次のセンサデータストリームの配信経路情報」として保持する。配信経路表の作成時には、ストリーム ID (streamid) という配信経路ごとにユニークな ID を生成し、配信経路表とあわせて ReceiverAgent に配布している。ストリーム ID は、配信されるセンサデータストリームと配信経路表を対応づけるために設けている。ストリーム ID が無い場合、センサデータストリーム配送中に配信先ノードの出入りや配信周期の変更が生じた際に、新たな配信経

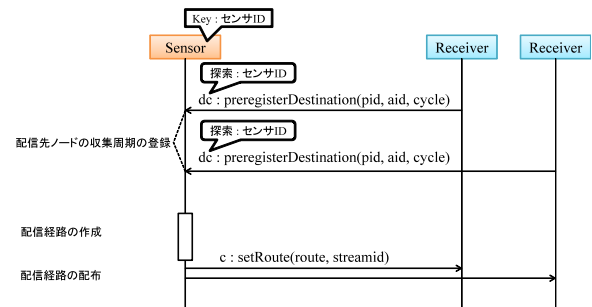


図 5 配信先ノードが発見する場合の配信経路作成シーケンス

Fig. 5 Sequence of creating and dispatching a delivery route with discovering by ReceiverAgents.

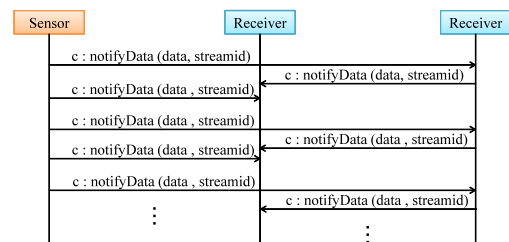


図 6 配信シーケンス

Fig. 6 Sequence of delivering a sensor data stream.

路表が構築・配布されることとなり、それまで配信されていたセンサデータストリームが予期せぬ配信先ノードに再配信されたり、届くべき配信先ノードに配信されないという問題が生じる。

3.3.3 センサデータストリーム配信

SensorAgent は、ReceiverAgent への配信経路表の配布が終わると先に作成した配信経路表に従い、エージェントに対する 1 対 1 呼び出し機能を用いて、各 ReceiverAgent へのセンサデータの配信 (notifyData) を始める (図 6)。配信されるセンサデータには、SensorAgent でストリーム ID とシーケンス番号を付与する。シーケンス番号は、SensorAgent 内のカウンタより求め、送出されるセンサデータごとに連続した番号が付与される。

他のエージェントからセンサデータを受け取った ReceiverAgent は、現在使用している配信経路表が持つストリーム ID とセンサデータに付与されたストリーム ID を比較し、合致した場合はその配信経路表とセンサデータに付与されたシーケンス番号より再配信先の ReceiverAgent を決定する。そして、得られた再配信先の ReceiverAgent に対して 1 対 1 呼び出し機能によりセンサデータを再配信する。受信したセンサデータのストリーム ID と現在使用している配信経路表のストリーム ID が合致しない場合は、保持していた「次のセンサデータストリームの配信経路情報」のストリーム ID と比較する。ここでストリーム ID が合致した場合は、新たな配信経路表による配信であると判断し、それまで使用していた配信経路表を破棄し、保持

していた配信経路表に更新し、その配信経路表に従いセンサデータを再配信する。保持していた次配信経路表のストリーム ID とも合致しない場合は、そのセンサデータを無効データとして破棄する。

4. PIAX テストベッドを用いた評価

PIAX を用いて LLF-H 法を実装したセンサデータストリーム配信システムを PIAX テストベッドを用いて評価を行った。PIAX テストベッドを利用することで、シミュレーションでは検証が困難な CPU 負荷や通信遅延の存在する環境下での配信システムの挙動を検証する。

4.1 PIAX テストベッドの概要

PIAX テストベッドは、PIAX エージェントを用いたシステムの動作検証や性能評価を容易に行うためのテストベッドとして NICT により整備・運営されているサービスであり、同機構の新世代ネットワーク技術の研究開発テストベッド JGN-X の一部として提供されている。

PIAX テストベッドでは、計 492 台^{*3}の仮想サーバ上に PIAX エージェントプログラムの実行基盤が用意されており、ユーザは、ブラウザを用いてテストベッドのポータルサイトからログインし、作成した PIAX エージェントプログラムのアップロードや仮想サーバ上で動作させるピア数の設定、エージェントプログラムへの実行指示、実行ログの収集指示などの操作を行う。実行基盤を構成する仮想サーバは、国内 5 拠点（北海道札幌市、東京都小金井市、東京都大手町、京都府精華町、大阪府吹田市）に配置されており、各仮想サーバは JGN-X L2 サービスにより構成された同一セグメントの実験用ネットワークに接続されている。各仮想サーバ間の通信において、同一拠点内では 1 ms 未満、異なる拠点との通信では約 2 ms から 26 ms の通信遅延が生じ、通信遅延の影響を考慮した広域分散システムの検証にも利用できる。

4.2 単一収集周期における配信遅延時間とノード間公平性の評価

LLF-H 法においてパラメータとして与える上限ホップ数と配信先ノード数が配信遅延時間とノード間の公平性に与える影響を検証するために評価を行った。

4.2.1 配信環境と評価指標

評価実験を行う配信環境は、配信元ノード 1 ノードと配信先ノード 100 ノードとする。これらのノードにはノード間の通信遅延時間の差異を極力排除するため、すべて京都府精華町に配置された仮想サーバを使用し、各ノードに対してそれぞれ個別の仮想サーバを割り当てる。負荷推定に用いるパラメータ α と β は、送受信の負荷が同一であると

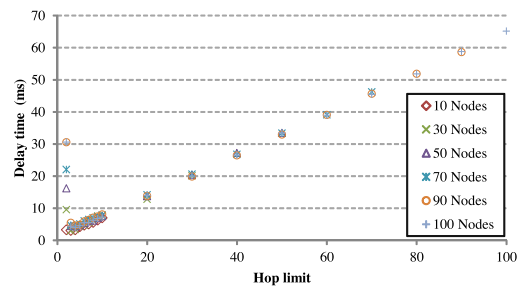


図 7 配信遅延時間の変化

Fig. 7 Average delay time.

仮定し、 $\alpha = \beta = 1$ とした。LLF 法において問題となる収集周期が同じノードが多数存在する場合を想定し、全配信先ノードに同一の収集周期を与える。この評価においては収集周期として 1 を与え、配信元ノードが送出する全データを受信する形とした。また、これによりすべてのセンサデータが各配信先ノードまで同じ経路を用いて配信されることとなる。センサデータには、実際の観測値の代用として 0 パディングされた 1,024 バイトのダミーデータを用い、このセンサデータを 50 ms 間隔で 5 分間配信元ノードから送出することで計測を行う。LLF-H 法により、配信経路を構築する際に指定する上限ホップ数は 2 から 10 まで 1 刻みとし、以降 100 ホップまでは 10 刻みとした。また、配信先ノードの数による影響を評価するため、配信先ノード数が 10, 30, 50, 70, 90, 100 ノードの状態それぞれ評価を行った。

配信先ノードが複数となるため、あるセンサデータが配信元ノードから送信され、各配信先ノードに到達するまでに要する配信遅延時間やその平均値も各配信先ノードごとに存在することとなるが、ここでは配信遅延時間の評価値として、計測期間における配信遅延時間の平均値が最大となる配信先ノードの値を用いる。これは、あるセンサデータがすべての配信先ノードに行きわたるまでに必要となる平均時間を表すこととなる。

公平性の評価においては、通信フローの公平性評価などに広く用いられる Jain's fairness index（以下 Fairness Index と表記する）[18] により評価する。本研究では、配信先ノード数を N 、各配信先ノードの負荷を L_i 、配信元ノードの負荷を L_0 として、式 (2) により Fairness Index を求める。この Fairness Index は、 $\frac{1}{N+1}$ から 1 の間の値となり、1 に近いほどより公平な状態であるといえる。

$$FairnessIndex = \frac{\left(\sum_{r=0}^N L_r\right)^2}{(N+1) \cdot \sum_{r=0}^N L_r^2} \quad (2)$$

4.2.2 配信遅延時間

図 7 は、横軸をホップ数の上限、縦軸を配信遅延時間としてプロットしたグラフを示している。また、図 8 は図 7 より先頭 10 ホップまでの部分を切り出し、拡大したグラフとなっている。

*3 2014 年 3 月時点

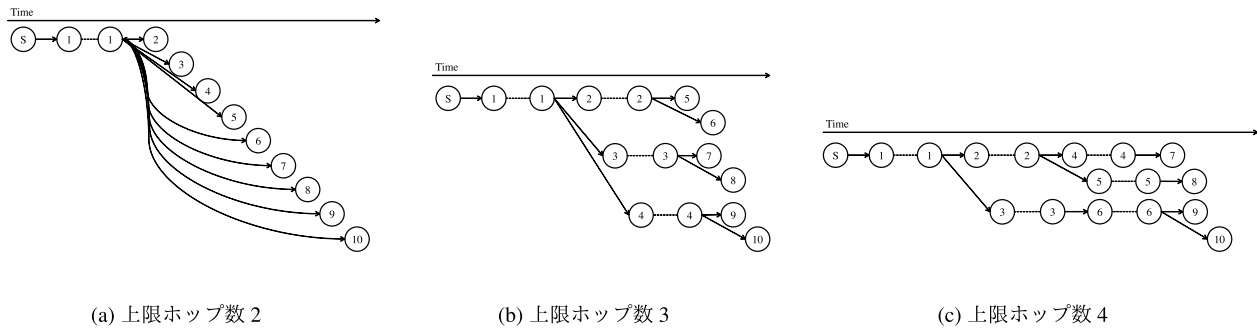


図 9 10 ノード時の配信順序

Fig. 9 Delivering sequence in case of 10 nodes.

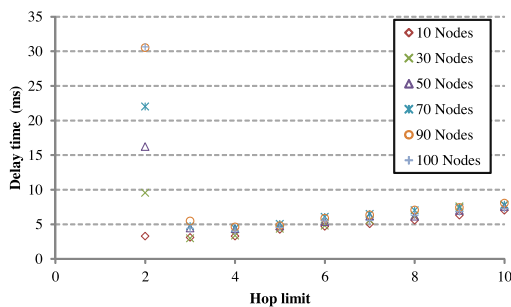


図 8 配信遅延時間の変化 (2～10 ホップ)

Fig. 8 Average delay time (2 to 10 hops).

上限ホップ数を 2 ホップに制限した場合、配信元ノードからセンサデータを直接受信した 1 ノードが他の全配信先ノードに順次再配信するスター状の経路となることから、ノード数にほぼ比例して配信遅延時間が増加することとなっている。また、90 ノードと 100 ノードでの計測の結果、ほぼ同じ配信遅延時間となっているが、これは 100 ノードの場合では再送信するセンサデータ数がより多いことから再送信している 1 ノードにおいて Java の JIT による最適化が進行し、再配信に要する時間が短縮されたためと推測している。

次に、上限ホップ数が配信先ノード数以上になる場合、LLF 法とほぼ同値となる。これは、その配信経路が LLF 法と同経路となることによる。LLF 法では、経路構築において負荷が最も低いノードに再配信させる経路を構築していくため、すでに再配信を行っており、受信負荷と送信負荷が加算された配信先ノードより、受信負荷のみとなっている配信先ノードが次の再配信ノードに選択され、結果として配信経路がチェーン状となる。これにより、末端部の配信先ノード以外の配信先ノードの負荷は送信メッセージ数 1 受信メッセージ数 1 と等しくなり、負荷は均等化されるが、末端部の配信先ノードまでの配信に要するホップ数は配信先ノード数に等しくなる。

上限ホップ数を 2 ホップより制限を緩和すると、ノード数が 10, 30 の場合では、上限ホップ数が 3 の場合に配信遅延時間が最小となり、それ以降のノード数 50 から 100 の

場合では上限ホップ数が 4 の場合に最小となり、3～5 ms の配信遅延時間で全配信先ノードにセンサデータが届いていることが分かる。配信遅延時間の最小部より右側では、上限ホップ数の制限を緩和すると、その上限ホップ数にほぼ比例して増加することが分かる (図 7)。

このように配信遅延時間に最小点が生じる理由として、配信先ノード数が 10 ノードの場合を例として取り上げる。図 9 は、配信先ノード数が 10 ノードの場合におけるセンサデータの配信順序を示している。各円はノードを示しており、円内が S は配信元ノードを表し、円内が数字のノードはそれぞれ配信先ノードを表している。横軸は時間経過を示しており、ノード間を結ぶ矢印はノード間でのセンサデータの送受信による時間経過、同一番号ノード間を結ぶ破線はそのノード内での再配信のためのオーバーヘッドによる時間経過を表している。まず、上限ホップ数が 2 の場合 (図 9(a)) では、先に記したように、配信元ノードからセンサデータを受け取った配信先ノードが他の全配信先ノードに再送信することとなるため、図 8 にあるように配信先ノード数に比例した配信遅延時間となる。また、上限ホップ数が 3 の場合 (図 9(b)) と 4 の場合 (図 9(c)) を比較すると、上限ホップ数が 3 の場合では配信先ノード $1 \rightarrow 4 \rightarrow 10$ が最も時間を要する経路となっており、上限ホップ数が 4 の場合では配信先ノード $1 \rightarrow 3 \rightarrow 6 \rightarrow 10$ が最も時間を要する経路となっている。センサデータの送受信による時間経過を t 、再配信のためのオーバーヘッドによる時間経過を d とすると、配信先ノード 1 を基準として、上限ホップ数が 3 の場合では $3t + d + 2t = 5t + d$ 、上限ホップ数が 4 の場合では $2t + d + t + d + 2t = 5t + 2d$ となり、より多くの時間を要することが分かる。これ以降では、ホップによる遅延 ($d + t$) の累積が優勢となるため、ホップ数に比例して配信遅延時間が増加すると考えられる。配信先ノード数が 50 以降における配信遅延時間に最小点があるのは、配信先ノード数が多い場合において、上限ホップ数を抑制すると再配信先 (図 9 における分岐先) が増加し、これらの再配信先への順次再配信による遅延の累積がホップによる遅延に優越するため

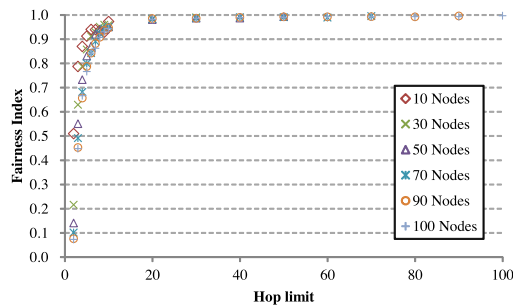


図 10 CPU 負荷による Fairness Index

Fig. 10 Fairness Index by CPU load.

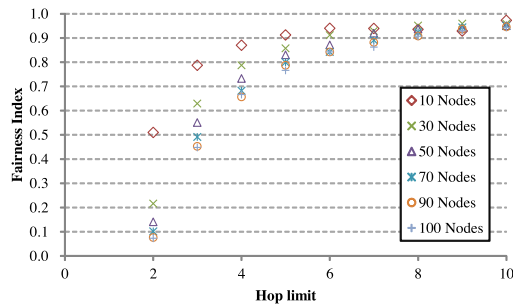


図 11 CPU 負荷による Fairness Index (2~10 ホップ)

Fig. 11 Fairness Index by CPU load (2 to 10 hops).

であると考えられる。たとえば配信先ノード数 50、上限ホップ数 4 の場合では、この再配信先が最大で 4 ノードとなっているが、上限ホップ数 3 の場合では、この再配信先が最大で 7 ノードとなる。これにより、ホップによる遅延の増加を上回る再配信先への順次再配信による遅延が生じており、上限ホップ数 3 の場合において上限ホップ数 4 の場合より配信遅延時間が大きくなる要因となっている。

4.2.3 Fairness Index

計測期間中の平均 CPU 負荷 (CPU 利用率) から求めた Fairness Index の変化を図 10 と図 11 に示す。図 11 は図 10 より先頭 10 ホップまでの部分を切り出し、拡大したグラフとなっている。

図 10 から読み取れるように、上限ホップ数の緩和に対して Fairness Index は早期に立ち上がり、以降の変化は少なくなっている。0.9 を閾値とすると、10 ノード時には 5 ホップ、30 ノード時には 6 ホップ、50 ノード時には 7 ホップ、それ以降は 8 ホップにおいて閾値を超えておりほぼ公平といえる状態となっている。

図 9 から分かるとおり、上限ホップ数が小さい場合では、ホップ数の制限に対応するため、特定の配信先ノードがその他の配信先ノードに再配信する経路となり、それら特定の配信先ノードはそれ以外の配信先ノードと比較してより多くのセンサデータを送信することとなる。同時に、再配信に参加せず受信のみとなる配信先ノードも多く生じるため、それら特定の配信先ノードに負荷が偏ることとなる。特に、配信先ノード数が多い場合においては、上限ホップ数が同じであっても再配信先となる配信先ノード数が多く

なるため、負荷の偏りは大きく現れることとなる。配信先ノード数が多い場合において、Fairness Index の立ち上がりが遅れるのは、この負荷の偏りの特性のためである。

上限ホップ数を緩和を進めると、ホップ数の制限に余裕ができることから再配信に参加できるノードが増え、LLF 法の手順に従い負荷が低い配信先ノードから優先的に再配信を行うようになる。これは配信経路においては、複数の再配信先を持つ配信先ノードが減少し、チェーン状となる部分が増加する形で現れる。チェーン状になった配信経路上の配信先ノードでは送信センサデータ数と受信センサデータ数が等しくなり、各ノードが扱うセンサデータ数も等しくなることから、負荷の均等化が進むこととなる。特に、上限ホップ数が配信先ノード数の $\frac{1}{2}$ を超えると、配信経路は 2 本のチェーン状の経路となり、上限ホップ数をそれ以上大きくしても、その緩和に合わせてチェーン間で配信先ノードが移動するのみとなり、全体で見ると配信先ノードが送受信するセンサデータ数は変化しなくなる。これによって Fairness Index が一定値となる区画が、上限ホップ数が配信先ノード数より 1 少ない点まで続く。上限ホップ数が配信先ノード数以上となった場合、配信経路は LLF 法と同経路の完全なチェーン状となるため、LLF 法での Fairness Index と同値となる。

4.3 複数の収集周期における配信遅延時間とノード間公平性の評価

LLF-H 法が想定している異なった収集周期を持つ配信先ノードに対して配信する場合において、ホップ数の制限が配信遅延時間とノード間の公平性に与える影響について検証を行った。各配信先ノードが要求する収集周期は、配信先ノードが個々に収集周期を決める環境を想定していることから、Zipf 分布による割当てを行った。Zipf 分布によって、Web などにおける特定のコンテンツへのアクセス頻度と同様の偏りを生じさせることができる。また、記録や解析のためには、より短周期のストリームが好まれると仮定し、Zipf 分布の各順位に対応する収集周期として 1 から 6 までは割り当てる。配信先ノードは全体で 48 ノードとし、各周期のノード数の偏りが小さい場合と大きい場合の挙動の差異を確認するため、Zipf 分布のパラメータ s には 0.5 と 2.5 を与える。各収集周期に対して割り当てる配信先ノード数は $s = 0.5$ と $s = 2.5$ のそれぞれの場合において図 12 のとおりとなる。配信周期やセンサデータのサイズなどの条件は、4.2 節の評価と同条件とし、各センサデータの配信遅延時間と配信に要したホップ数、配信期間中の各ノードの CPU 負荷を計測した。

図 13 と図 14 は、それぞれ $s = 0.5$ とした場合と、 $s = 2.5$ の場合における各センサデータの配信に要した配信遅延時間と、各配信遅延時間に対応するセンサデータ数をプロットした度数分布図を示している。センサデータが配

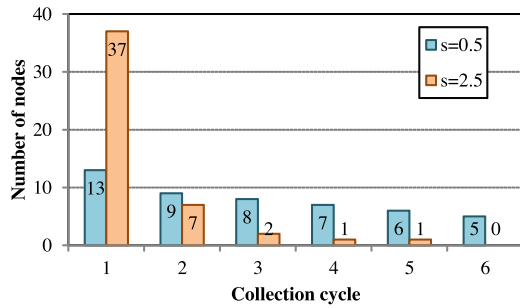


図 12 Zipf 分布に基づく収集周期割当て

Fig. 12 Node assignment based on a Zipf distribution.

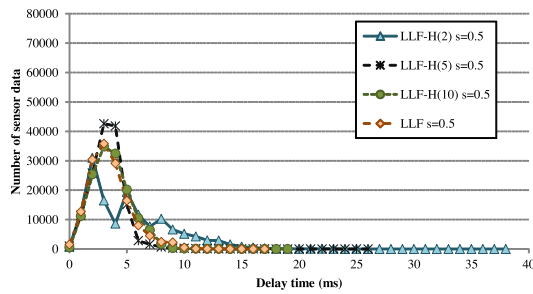


図 13 配信遅延時間の度数分布 ($s = 0.5$)

Fig. 13 Frequency distribution of delay time ($s = 0.5$).

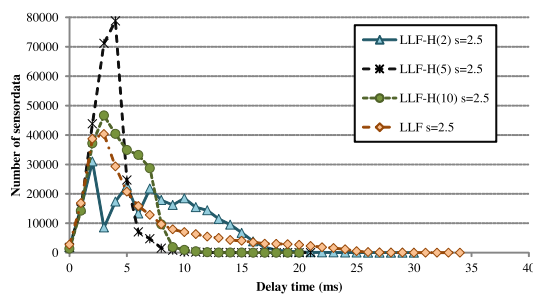


図 14 配信遅延時間の度数分布 ($s = 2.5$)

Fig. 14 Frequency distribution of delay time ($s = 2.5$).

信元ノードから配信先ノードに受信されるまでに要した配信遅延時間を横軸に、各配信遅延時間に対応するセンサデータの数縦軸としてプロットしている。また、図 15 と図 16 は、横軸を各センサデータの配信に要したホップ数とし、縦軸をそれに対応するセンサデータ数としてプロットした度数分布図となっている。各図の凡例において、LLF-H 法の後続く括弧内の数値は LLF-H 法での上限ホップ数を示している。

まず、配信遅延時間の度数分布 (図 13, 図 14) を見ると、LLF-H 法において上限ホップ数を 2 とした場合では、その他の上限ホップ数を指定した場合や LLF 法の場合と比較して $s = 0.5$, 2.5 の双方において、配信に 10ms 前後を要したセンサデータが多くなっている。それに対して 0~5ms の配信遅延時間を持つセンサデータは他の場合と比較して少なくなっており、全体として、他の手法と比較して高遅延なセンサデータの割合が多くなっていることが分かる。このとき、ホップ数の度数分布 (図 15, 図 16) を参

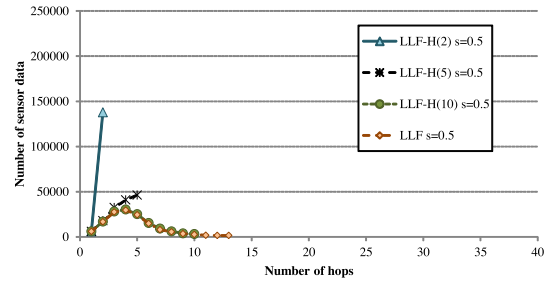


図 15 ホップ数の度数分布 ($s = 0.5$)

Fig. 15 Frequency distribution of hops ($s = 0.5$).

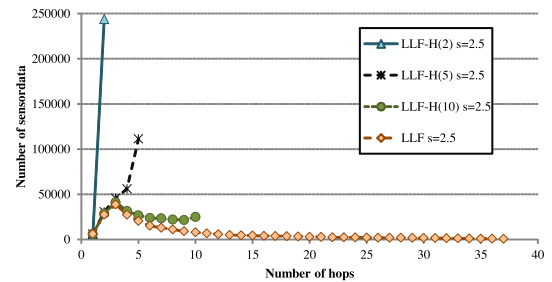


図 16 ホップ数の度数分布 ($s = 2.5$)

Fig. 16 Frequency distribution of hops ($s = 2.5$).

照すると、上限ホップ数を 2 とした場合では、ほとんどのセンサデータがホップ数 2 で配信されている。4.2 節での評価のとおり、上限ホップ数を 2 ホップに制限した場合、配信元ノードから直接センサデータを受け取ったノードが他のノードに順次配信することから、複数ノードにより再配信される他の場合と比較して、高遅延なセンサデータの割合が多くなると考えられる。これは、 $s = 0.5$ の場合と $s = 2.5$ の場合の比較において、配信周期 1 の配信先ノード数が多い $s = 2.5$ の場合において、配信遅延時間が 10~15ms となったセンサデータが多いことから読み取れる。

上限ホップ数 5 の場合では、 $s = 0.5$, 2.5 の双方ともに配信遅延時間が 3~4ms であるセンサデータが最も多く、5ms 以上を要したセンサデータは他の評価と比較して少なく保たれている。これに対して、上限ホップ数 10 とした場合では、主に $s = 2.5$ の場合において顕著であるが、上限ホップ数を 5 の場合に 3~4ms に現れていたピークが低くなり、遅延時間が 5~7ms となったセンサデータが増えており、比較的高遅延なセンサデータが増えているといえる。図 15, 図 16 より、上限ホップ数 5 の場合と上限ホップ数 10 の場合におけるホップ数の分布を比較すると、ホップ数 1~3 であったセンサデータ数はほぼ同数であるが、ホップ数 4, 5 において、上限ホップ数 5 の場合が上限ホップ数 10 の場合を大きく上回っていることが分かる。これは、負荷分散の観点ではより大きいホップ数を要する経路が望ましいが、ホップ数の制限により、その制限を上回るセンサデータの配信経路が抑制され、その結果、センサデータが制限値以下の経路に集中することによる。逆にホップ数の制限が緩和されると負荷分散が優先され、より

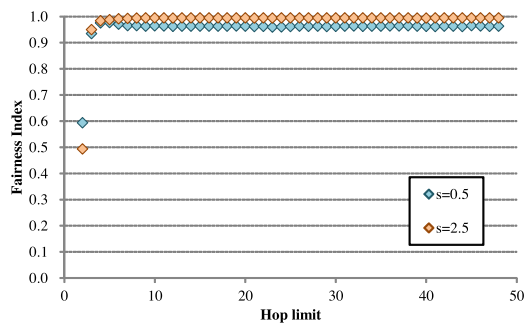


図 17 CPU 負荷による Fairness Index

Fig. 17 Fairness Index by CPU load.

大きいホップ数を要する経路が選択される。上限ホップ数を 10 とした場合において、上限ホップ数を 5 とした場合と比較して、配信遅延時間が 5～7ms のセンサデータが増加するのは、それらのセンサデータは、制限が緩和されることで選択できるようになったホップ数 6～10 を要する経路で配信されたと考えられる。

LLF 法に注目すると、 $s = 2.5$ とした場合では、10～24ms においてそれぞれ約 7,000～1,000 個のセンサデータが含まれており、配信される総センサデータの約 20% が配信に 10ms 以上を要していることが分かる。ホップ数の分布 (図 15, 図 16) では、 $s = 2.5$ の場合では最大で 37 ホップを要しており、これは収集周期 1 の 37 配信先ノードがチェーン状の配信経路を構成したためと考えられ、LLF 法ではノード数の偏りが大きい場合において配信遅延が大きくなりやすいことが分かる。

次に Fairness Index を計測期間中の平均 CPU 負荷 (CPU 利用率) から求めた結果を図 17 に示す。この図から読み取れるように、上限ホップ数が 3 の場合で 0.9 を超えており、複数周期の場合においても Fairness Index は上限ホップ数の緩和に対して早期に立ち上がることが分かる。

4.4 センサデータサイズによる配信遅延時間の評価

センサデータサイズが配信遅延時間に与える影響を検証するため、配信するセンサデータのサイズと上限ホップ数を変化させて計測評価を行った。

センサデータには 0 パディングされたダミーデータを用い、1 KiB から 128 KiB まで 2 の累乗単位 of データサイズを与える。配信経路を構築する際に指定する上限ホップ数は 2 から 10 まで 1 刻み、以降は 20, 30 とする。配信先ノードは 30 ノードとし、その他の条件は、4.2 節の評価時と同条件で計測を行う。

図 18 は、横軸をセンサデータサイズ、縦軸を配信遅延時間として計測結果をプロットしたグラフとなっている。図中の凡例は LLF-H 法での上限ホップ数を示している。グラフより、センサデータサイズの変化にほぼ比例して配信遅延時間が増加していることが分かる。また、上限ホップ数を 3 とした場合に配信遅延時間が最小であり、これは

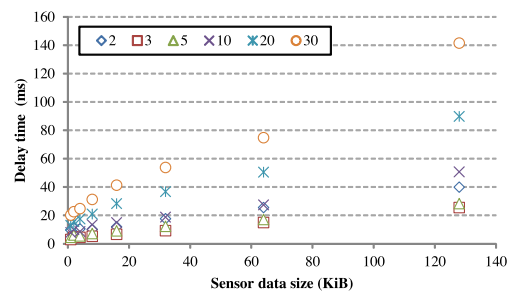


図 18 センサデータサイズによる配信遅延時間の変化

Fig. 18 Average delay time by increasing the size of sensor data.

4.2 節での評価と同理由によるが、この評価の範囲におけるセンサデータサイズの変化の影響は生じないことが分かる。

4.5 考察

4.2 節, 4.3 節での評価において、上限ホップ数を 2 とした場合では、配信元ノードから直接センサデータを受け取ったノードが他の全ノードに再配信することから、1 ノードに負担が集中し、遅延、公平性の両面において不適当な状態といえる。その一方で、上限ホップ数を 3 以上に緩和していくと配信遅延時間は大きく減少し、最小値を経た後に、上限ホップ数に比例して増加することから、配信遅延時間は配信先ノード間のホップ数からの影響が大きく、最小点を超えた後にはある配信先ノードが他の配信先ノードに順次再配信することにより生じる遅延時間はホップ間で生じる遅延時間に埋没し、影響を与えなくなると考えられる。

LLF 法においては、公平性の面ではそれ以前に提案した LCF 法と比較して良好となる手法であったが、同一収集周期の配信先ノードが多数存在する場合においては、センサデータの配信経路がチェーン状になることからホップ数が増え、配信遅延時間が増加する点が問題であった。LLF-H 法では、上限ホップ数を配信先ノード数とした場合において LLF 法と同経路となることから、配信遅延時間の最悪値と Fairness Index の最良値は LLF 法を用いた場合と同等となる。逆に、配信遅延時間の最良値は上限ホップ数 3 以降の近傍に現れることから、LLF-H 法ではその間の上限ホップ数を選択することで公平性と配信遅延時間の間でトレードオフの選択を行うことができる。

また、本研究における 4.2 節, 4.3 節の評価より、公平性は、上限ホップ数の緩和に対して素早く改善し、LLF 法を用いた場合の公平性に近づく。このため、上限ホップ数を数ホップ程度に制限した場合であっても Fairness Index は 0.9 程度と高い公平性を示すことから、配信遅延時間を抑制しつつ公平性も確保した配信が実現できると考えられる。

5. まとめ

本研究では、センサデータストリーム配信手法の 1 つで

ある LLF-H 法を実装し, PIAX テストベッドを用いた実機環境での配信遅延・公平性の挙動について評価を行った. その結果, LLF-H 法では配信遅延時間は上限ホップ数の制限に単純比例するのではなく, 上限ホップ数を 2 とした場合には高遅延を示し, 上限ホップ数を 3 とした場合以降に最小値を示し, それ以降は上限ホップ数に比例して増加していくことが判明した. また, 公平性については先行研究におけるシミュレーション評価と同様に, LLF-H 法では上限ホップ数を数ホップ程度に制限した場合であっても負荷の公平性を大きく損なわれることはなかった. 以上の評価結果より, LLF-H 法では, 適切な上限ホップ数を指定することにより, 公平性を保ちつつ配信遅延時間を抑制しうることが確認できた.

今後は, 本計測で得られた結果をもととして, 配信遅延時間のモデル化を進め遅延時間の事前推測, ノード間遅延時間が不均一な環境への適用, 検証を行っていく.

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」による成果である. 検証環境として JGN-X 上の PIAX テストベッド (プロジェクト番号 JGNX-A12005) を利用した.

参考文献

- [1] 池田和史, Thilmee, M., 廣森聡仁, 山口弘純, 東野輝夫: 安定したストリーム配信を実現するオーバーレイマルチキャストプロトコルの設計と PlanetLab 上での実証実験, 情報処理学会論文誌, Vol.50, No.6, pp.1549-1560 (2009).
- [2] Tran, D.A., Hua, K.A. and Do, T.: ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming, *Proc. 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, Vol.2, pp.1283-1292 (2003).
- [3] Silawarawet, K. and Nupairoj, N.: Locality-Aware Clustering Application Level Multicast for Live Streaming Services on the Internet, *Journal of Information Science and Engineering*, Vol.27, No.1, pp.319-336 (2011).
- [4] Jin, X., Yiu, W.-P.K., Chan, S.-H.G. and Wang, Y.: On Maximizing Tree Bandwidth for Topology-Aware Peer-to-Peer Streaming, *IEEE Trans. Multimedia*, Vol.9, No.8, pp.1580-1592 (2007).
- [5] 坂下 卓, 義久智樹, 原 隆浩, 西尾章治郎: ストリーミング環境における分割データの重要度を考慮した視聴中止端末数削減手法, 情報処理学会論文誌, Vol.52, No.11, pp.3008-3017 (2011).
- [6] Castro, M., Druschel, P., Kermarrec, A.-M. and Rowstron, A.I.: SCRIBE: A large-scale and decentralized application-level multicast infrastructure, *IEEE Journal on Selected Areas in Communications*, Vol.20, No.8, pp.1489-1499 (2002).
- [7] Castro, M., Druschel, P., Kermarrec, A.-M., Nandi, A., Rowstron, A. and Singh, A.: SplitStream: High-bandwidth multicast in cooperative environments, *ACM SIGOPS Operating Systems Review*, Vol.37, No.5, pp.298-313, ACM (2003).
- [8] Magharei, N. and Rejaie, R.: PRIME: peer-to-peer receiver-driven mesh-based streaming, *IEEE/ACM Trans. Networking (TON)*, Vol.17, No.4, pp.1052-1065 (2009).
- [9] Kawakami, T., Ishi, Y., Yoshihisa, T. and Teranishi, Y.: A Delivery Method considering Communication Loads for Sensor Data Stream with Different Collection Cycles, *Proc. 28th ACM Symposium on Applied Computing (SAC 2013)*, pp.611-618 (2013).
- [10] Kawakami, T., Ishi, Y., Yoshihisa, T. and Teranishi, Y.: A P2P Delivery Method for Sensor Data Stream Based on Load Estimation from Collection Cycles, *Proc. 4th IEEE International Workshop on Enablers for Ubiquitous Computing and Smart Services (EUCASS 2013)* in conjunction with The 37th Annual International Computer Software & Applications Conference (COMPSAC 2013), pp.289-294 (2013).
- [11] 石 芳正, 川上朋也, 義久智樹, 寺西裕一: 収集周期の異なるセンサデータストリームのための P2P 型配信システムとその評価, 情報処理学会論文誌, Vol.55, No.2, pp.707-720 (2014).
- [12] Kawakami, T., Ishi, Y., Yoshihisa, T. and Teranishi, Y.: A P2P-Based Sensor Data Stream Delivery Method to Accommodate Heterogeneous Cycles, *Journal of Information Processing*, Vol.22, No.3, pp.455-463 (2014).
- [13] Rowstron, A. and Druschel, P.: Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems, pp.329-350 (2001).
- [14] 吉田 幹, 奥田 剛, 寺西裕一, 春本 要, 下條真司: マルチオーバーレイと分散エージェントの機構を統合した P2P プラットフォーム PIAX, 情報処理学会論文誌, Vol.49, No.1, pp.402-413 (2008).
- [15] Aspnes, J. and Shah, G.: Skip Graphs, *ACM Trans. Algorithms*, Vol.3, No.4, p.37 (2007).
- [16] Hildrum, K., Kubiawicz, J.D., Rao, S. and Zhao, B.Y.: Distributed object location in a dynamic network, *Proc. 14th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA '02)*, pp.41-52, ACM (2002).
- [17] 金子 雄, 春本 要, 福田真哉, 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.18, pp.1-15 (オンライン), 入手先 (<http://ci.nii.ac.jp/naid/110002977724/>) (2005).
- [18] Jain, R., Chiu, D.-M. and Hawe, W.: A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems, *DEC Research Report TR-301* (1984).



石 芳正 (正会員)

2004 年京都工芸大学工芸学部電子情報工学科卒業. 2006 年大阪大学大学院情報科学研究科博士前期課程修了. 同年同大学サイバーメディアセンター特任研究員. 2008 年同大学院情報科学研究科特任研究員. 2012 年同大学サイバーメディアセンター特任研究員となり, 現在に至る. 修士 (情報科学). ユビキタス応用システム等の研究開発に従事.



川上 朋也 (正会員)

2005 年近畿大学理工学部経営工学科卒業。2007 年大阪大学大学院情報科学研究科博士前期課程修了。同年同研究科マルチメディア工学専攻特任研究員。2012 年同大学サイバーメディアセンター特任研究員。2013 年神戸大学大学院工学研究科学術推進研究員。2014 年大阪大学サイバーメディアセンター特任研究員を経て、2015 年より奈良先端科学技術大学院大学情報科学研究科助教および大阪大学サイバーメディアセンター招へい研究員となり、現在に至る。情報科学博士 (2013 年 3 月, 大阪大学)。分散処理システム, P2P ネットワークに関する研究に従事。IEEE 会員。



義久 智樹 (正会員)

2002 年大阪大学工学部電子情報エネルギー工学科卒業。2003 年同大学院情報科学研究科マルチメディア工学専攻博士前期課程を修了し、2005 年同専攻博士後期課程修了。博士 (情報科学)。2005 年京都大学学術情報メディアセンター助教。2008 年大阪大学サイバーメディアセンター講師を経て、2009 年より同准教授となり、現在に至る。この間、カリフォルニア大学アーバイン校客員研究員。センサネットワークおよびインターネット放送に興味を持つ。電子情報通信学会, IEEE 各会員。



寺西 裕一 (正会員)

1993 年大阪大学基礎工学部情報工学科卒業。1995 年同大学院基礎工学研究科博士前期課程修了。同年日本電信電話株式会社入社。2005 年大阪大学サイバーメディアセンター講師。2007 年同大学院情報科学研究科准教授。2008 年より情報通信研究機構専攻研究員, 招へい専門員を兼任。2011 年より情報通信研究機構研究マネージャおよび大阪大学サイバーメディアセンター招へい准教授, 現在に至る。分散システム, オーバレイネットワーク, センサネットワークおよびその応用システムに関する研究開発に従事。博士 (工学) (2004 年 3 月, 大阪大学)。IEEE 会員。