

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

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

受付日 2013年5月13日, 採録日 2013年10月9日

概要: ライブカメラや環境モニタリングなどの用途によるセンサの普及にともない, 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対する注目が高まっている. その中でも, 複数の配信先がそれぞれ異なる周期によるセンサデータ収集を要求する環境において, Peer-to-Peer 技術を用いて配信先の通信負荷を分散する Peer-to-Peer 型センサデータストリーム配信手法として LCF 法や LLF 法が提案されている. しかしながら, これらの提案では主にシミュレーションに基づく評価のみが行われており, 実環境における有効性は未知数であった. 本稿では, 当該手法に基づくセンサデータストリーム配信システムの設計と実装について述べ, 情報通信研究機構が提供している PIAX テストベッドを用いた実機評価について記す.

キーワード: センサデータストリーム配信, Peer-to-Peer

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

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

Received: May 13, 2013, Accepted: October 9, 2013

Abstract: Due to the prevalence of sensors such as security cameras or environmental sensors, sensor data stream delivery which means delivering the sensor data every cyclic collection attracts great attention. For sensor data stream delivery, various methods to distribute communication loads in the case of delivering the same sensor data streams to multiple clients have been studied. Some methods which are considered in the case of delivering the sensor data stream that have different data collection cycles are proposed. Although these methods did not evaluate the system performance of distribution delay, jitter, etc. In this paper, we evaluated the Peer-to-Peer streaming system using the PIAX test bed which NICT provides.

Keywords: sensor data stream distribution, Peer-to-Peer

1. はじめに

近年, センサデバイスの高機能化・小型化・低廉化にともない, ライブカメラや気象センサといった環境観測機器の普及が進んでおり, 様々な利用者がセンサデバイスから得られるセンサデータを共有・活用するアプリケーションが現実的になりつつある. これらのセンサでは, 多くの場合, 周期的にセンサデータが収集され, 収集されるたびにセンサデータ利用者に配信されることとなる. たとえば, ライブカメラでは, 1 秒間に 30 枚から 60 枚の画像がライブカメラから送り出され, それらを受け取ったセンサデー

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 神戸大学大学院工学研究科
Graduate School of Engineering, Kobe University, Kobe, Hyogo 657-8501, Japan
³ 独立行政法人情報通信研究機構
National Institute of Information and Communications Technology, Koganei, Tokyo 184-8795, Japan
^{a)} ishi.yoshimasa@ime.cmc.osaka-u.ac.jp
^{b)} kawakami.tomoya@eedept.kobe-u.ac.jp
^{c)} yoshihisa@cmc.osaka-u.ac.jp
^{d)} teranisi@nict.go.jp

タ利用者が視聴や解析処理を行ったり、蓄積や再配信などの二次サービスを展開したりすることとなる。センサから周期的に送出される一連のセンサデータは、一般にセンサデータストリームと呼ばれており、センサデータを必要としている利用者に逐次配信することにより映像配信や環境モニタリングなどに利用するセンサデータストリーム配信への注目が高まっている。

センサデータストリーム配信では、1台のセンサから得られるセンサデータを多数の利用者で共有することが一般的であり、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]。これらの研究では、複数の端末に同じセンサデータストリームを配信する場合に、センサデータを受信した端末が他の端末に再配信することで、センサデータの配信元に集中していた通信負荷を分散させている。これらの研究は、すべての端末においてセンサデータの収集周期が等しい均質なセンサデータストリームを対象としているが、次の例のように、同じセンサが収集したセンサデータに対して周期の異なるセンサデータストリームを配信する場合も考えられる。

- 短い周期でのデータ取得により個人が特定される可能性があるセンサデータの場合、信頼済みのサービス提供者と、その他のサービス提供者との間で異なる周期でセンサデータを提供する場合。
- 温度センサの値から将来の気温を複数の機関が予測している場合、各機関が持つ端末の処理速度に応じて収集周期を決める。
- 映像ストリームを解析し、対象物が被写体として映っているかどうかによって、存在を検出したい場合、検出対象とする物（人・車など）の移動速度に応じて収集周期を決める。

収集周期が異なる場合であっても、収集周期の最も短いセンサデータストリームをすべての端末に配信し、各端末で必要とする周期のセンサデータのみを使用し、他は無視することで所望の収集周期を再現できる。この場合、既存手法を適用できるが、端末が必要としないセンサデータまで配信することとなり、端末に無用な通信負荷が生じることとなる。一方、収集周期を考慮してセンサデータを再配信することで、無駄な通信を抑制しつつ負荷を分散する Peer-to-Peer 型のセンサデータストリーム配信手法として LCF 法 [7] と LLF 法 [8] が提案されている。しかしながら、これらの文献ではシミュレーションによるメッセージ数を基にした評価のみが行われており、実環境において配信性能などに影響を与えるセンサデータのサイズや通信路の特性が配信に与える影響は未知数となっている。

本研究では、LCF 法と LLF 法に基づき実環境で動作する Peer-to-Peer 型のセンサデータストリーム配信システムを構築し、当該手法の適用範囲と実環境での有効性を確認するため、情報通信研究機構が提供している PIAX テス

トベッドにおいて、提案システムの挙動を調査し、評価を行う。

2. センサデータストリーム配信手法

ここでは、収集周期を考慮したセンサデータ配信手法として提案されている LCF 法と LLF 法が対象としている環境とそれぞれの手法について述べる。

2.1 配信モデル

図 1 に LCF 法、LLF 法が想定している配信モデルを示す。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収集している。以下、このサーバを配信元ノードと呼ぶ。配信元ノードは、センサごとに存在し、センサと 1 対 1 の対応となる。また、センサから取得した観測値はセンサデータと呼ぶ。 D_x は、センサデータの受信端末を示しており、 D_x 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。LCF 法、LLF 法では、センサごとにそれぞれの配信経路を構築する。複数の異なるセンサから生じるセンサデータを配信する場合は、図 1 の構造をそれぞれのセンサごとに構築する。

表 1 は、配信先ノード $D_1 \sim D_4$ 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 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。 D_2 と D_3 は、収集周期が 2 であり、時刻 2 ごとにセンサデータを受け取る。同様に D_4 は、時刻 3 ごとにセンサデータを受け取る。これらの収集周期は、各配信先ノードがそれぞれに保持する。

配信元ノードは自身のセンサデータを受信しようとする

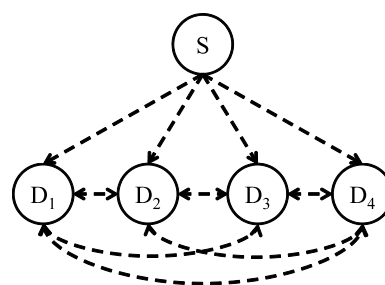


図 1 想定モデル

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=2)	○		○		○		○	...
D_3 (Cycle=2)	○		○		○		○	...
D_4 (Cycle=3)	○			○			○	...

配信先ノードの収集周期をもとに配信経路を決定し、各配信先ノードに通知するとともに、センサデータストリームの配信を始める。これにより、配信先ノードは希望する周期のセンサデータを配信元ノードまたは他の配信先ノードから受け取る。

2.2 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。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 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負荷、送信負荷 O_r は式 (1) となる。なお、 D_0 は S を表すものとし、 α と β は、それぞれ受信処理と送信処理の重み係数、 N は配信先ノードの数とする。

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

同様に、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負荷、受信負荷 I_r は式 (2) となる。

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

以上より、ノード D_r の通信負荷 L_r は、送信負荷 O_r と受信負荷 I_r の合計となる (式 (3))。

$$L_r = O_r + I_r \quad (3)$$

2.3 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する。たとえば、表 1 では、時刻 0 と時刻 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 2 と時刻 4 のセンサデータについては収集周期 1 の配信先ノードと収集周期 2 の配信先ノード、時刻 3 については収集周期 1 の配信先ノードと収集周期 3 の配信先ノードが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。

ノード間の負荷分散の指標としては、公平性の評価として一般的に用いられる Jain's fairness index を採用している (式 (4))。配信元ノードの負荷を L_0 、配信先ノードの

負荷を L_i で示している。FI は $\frac{1}{N}$ から 1 の間の値をとり、1 に近いほど公平な状態であるといえる。

$$FI = \frac{\left(\sum_{r=0}^N L_r\right)^2}{N \cdot \sum_{r=0}^N L_r^2} \quad (4)$$

2.4 LCF 法

想定環境において、収集周期の短い配信先ノードは、単位時間あたりの受信するセンサデータの数が多く、他の収集周期のセンサデータが含まれる確率が高い。たとえば、収集周期 2 の配信先ノードが受信するセンサデータには、その倍数である収集周期 4, 6... のセンサデータが含まれる。このため、収集周期の短い配信先ノード D_s が、収集周期がより長い配信先ノード D_l よりも優先的に再配信する場合、多くの配信先ノードがその再配信先となり、配信先ノード D_s に通信負荷が集中する可能性が高くなる。また、同時に単位時間あたりにより多くのセンサデータを扱うことから、収集周期の短い配信先ノードは再配信を行わなくても負荷が高い状態であると考えられる。

逆に、収集周期の長い配信先ノードは単位時間あたりに扱うセンサデータが少ないことから、相対的に負荷は低く、再配信先の候補となる配信先ノードも少ない。LCF (Longest Cycle First; 最長周期優先) 法では、収集周期の長い配信先ノードが優先的に再配信することで負荷分散する。配信先ノード D_1, D_2, D_3, D_4 の収集周期はそれぞれ 1, 2, 2, 3 であるため、最小公倍数の 6 が全体の周期となる。この全体の周期内の各周期での配信経路を求め、全体の周期 6 で繰り返し各配信経路を用いることとなる。周期 0 では、その周期のセンサデータを受け取る配信先ノードのうち、最長の周期を持つ D_4 は配信元ノードからセンサデータを受け取る。 D_4 は、周期 0 のセンサデータを必要とする D_3 と D_1 に再配信する。 D_3 は、同周期である D_2 に再配信する。 D_4 は 3 の倍数となる時刻のセンサデータしか配信できないため、 D_1 や D_3 は残りの周期のセンサデータを他のノードから受信することとなり、通信負荷の分散が期待できる。以降、残りの周期 1~5 についても経路を決定し、表 1 の場合における LCF 法での配信経路は図 2 のとおりとなる。このように、LCF 法では各配信先ノードが要求する収集周期のみに着目して配信経路を構築しており、比較的容易に実装できる。

2.5 LLF 法

LCF 法では、各配信先ノードが要求する収集周期のみに着目しており、受信や再配信により生じる負荷については経路作成に利用していない。このため、必ずしも負荷が適切に分配されない場合が生じる。たとえば、長周期の配信先ノードでは、他の配信先ノードへ多く送信するため、同じ時刻に再配信先が多い場合、送信負荷が増大する可能性

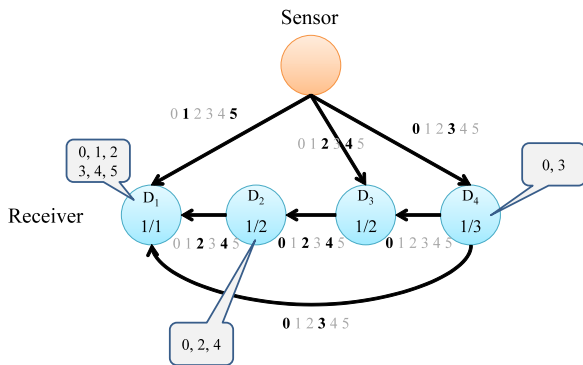


図 2 LCF 法による配信経路

Fig. 2 The delivery route in the LCF method.

がある。また、同一の収集周期を持つノードによるグループが複数ある場合、LCF 法ではグループ間をまたぐ再配信はグループ内の特定のノードのみが行うこととなり、負荷に偏りが生じる。これに対し、LLF (Lowest Load First ; 最小負荷優先) 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 2.2 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で推定された負荷が最小のノードから受け取ることで通信負荷を分散させる。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。

表 1 の場合、配信先ノード D_1 , D_2 , D_3 , D_4 の収集周期はそれぞれ 1, 2, 2, 3 であるため、最小公倍数の 6 が全体の周期となる。この全体の周期内の各周期での配信経路を求め、全体の周期 6 で繰り返し各配信経路を用いることとなる。センサデータ 1 個あたりの送信負荷を 1, 受信負荷を 1 とする。まず、各ノードが受信しなければならない周期のセンサデータによる受信負荷を求めると、6, 3, 3, 2 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_4 が配信元ノードからセンサデータを受け取り、 D_3 に再配信する。周期 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_1 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 6, 3, 5, 4 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_4 は配信元ノードから受け取る、 D_3 は D_4 から受け取る決定している。 D_2 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_4 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 6, 3, 5, 5 となる。同様に D_1 は D_2 から受け取り、各配信先ノードの負荷は 6, 4, 5, 5 となる。以降、残りの周期 1~5 につい

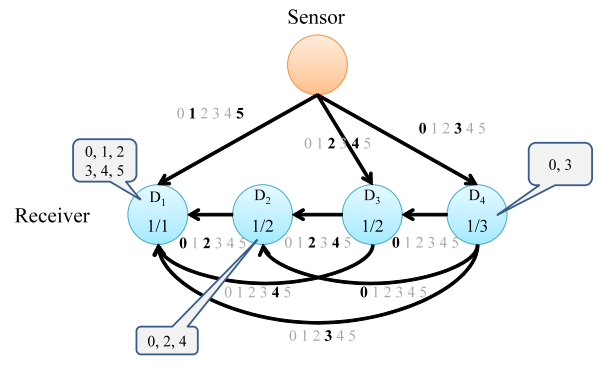


図 3 LLF 法による配信経路

Fig. 3 The delivery route in the LLF method.

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 3 のとおりとなる。

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

本研究では、配信元ノードと配信先ノード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。このため、オーバーレイネットワークを介した探索機能を持つ PIAX を用いてシステムを構築する。

3.1 PIAX の概要

PIAX [9] は、NICT や大阪大学を中心に研究開発されている Peer-to-Peer 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プラットフォームミドルウェアである。PIAX は、Java により実装されており、オープンソースソフトウェアとして公開されている*1。

PIAX では、主要な機能をピア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する。ピア間では、PIAX がサポートする各種オーバーレイネットワークにより相互に接続され、オーバーレイネットワークを用いたエージェントの探索やエージェント間通信、エージェントのピア間移動によりシステム全体の動作を実現する。ピアやエージェントは、それぞれユニークな ID、ピア ID とエージェント ID を持ち、これら ID により通信相手を識別する。たとえば、異なるピア上に置かれたエージェントのメソッドをリモート呼び出しする場合、call(ピア ID, エージェント ID, メソッド名, 引数...) とする。

また、PIAX は複数のオーバーレイネットワークの同時運用を実現しており、非構造型のフラッディングネットワークのほか、地理的探索に有用な LL-Net [10]、完全一致型探索を行う DOLR [11]、範囲探索を行う SkipGraph [12] など

*1 PIAX: P2P Interactive Agent eXtensions - <http://www.pi-ax.org/>

の構造型オーバーレイネットワークを標準でサポートしている。これにより、エージェントが持っている位置情報を対象に探索する場合は LL-Net, エージェントに付けられた任意の ID で探索する場合は DOLR といったように、エージェントを探索する際にはその探索条件に応じたオーバーレイネットワークを利用することで、効率良く目的とするエージェント群を発見できる。加えて、このとき発見されたエージェント群に対し、発見と同時に処理要求を伝え、その結果を 1 度に収集することが可能な発見型メッセージング (discovery messaging) 機能を実現しており、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したりすることができる。

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、配信先ノードでは PIAX のピアを動作させ、オーバー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。また、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。

配信元ノードのピア上には、センサに対応する SensorAgent を置く。SensorAgent は、以下の機能を持つ。

- センサからのセンサデータの収集
- 配信先ノードの希望収集周期に応じた配信経路の作成と配布
- センサデータストリームの配信

配信先ノードのピア上には、ReceiverAgent を配置する。ReceiverAgent は、以下の機能を持つ。

- 自身の収集周期の配信元ノードへのリクエスト
- 他ノードからのセンサデータストリームの受信

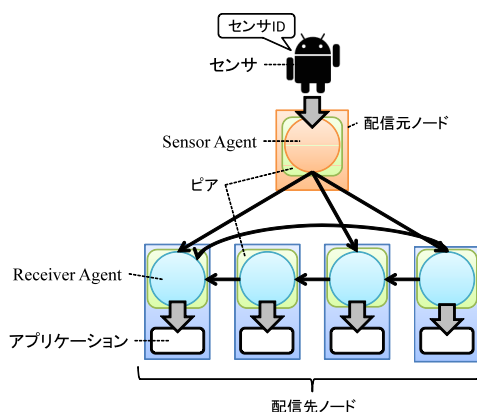


図 4 センサデータストリーム配信システムの構成

Fig. 4 Structure of sensor data stream delivery system.

- 配信元ノードから与えられた配信経路に従ったセンサデータストリームの再配信
- アプリケーションへのセンサデータの受け渡し

SensorAgent と ReceiverAgent 間を結ぶ図中の矢印は、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

3.3.1 各ノードの初期設定

本研究では、配信元ノードと配信先ノードは互いの通信先を知らない状態から、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。このため、配信に先立ちそれぞれを発見するために PIAX が標準サポートしている DOLR ネットワークを利用する。PIAX の DOLR ネットワークは、探索条件として与えられたキーに完全一致するキーを持つエージェントを発見することができるオーバーレイネットワークであり、SkipGraph の一部機能を用いて実装されている。これにより、登録されているキー数 N に対し $O(\log N)$ のホップ数で探索対象のキーを持つエージェントを発見し、メッセージを届けることができる。また、詳しくは 3.3.2 項で述べるが、SensorAgent が ReceiverAgent を探索する場合と、逆に ReceiverAgent が SensorAgent を探索する場合が存在するため、それぞれが DOLR ネットワークに探索対象キーを登録し、探索対象となる必要がある。

まず、配信元ノードに接続されているセンサのセンサ ID をその配信元ノードに設定する。センサ ID は SensorAgent に渡され、DOLR ネットワークに探索対象キーとして登録される。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。

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 と収集周期を設定する。ReceiverAgent も SensorAgent と同様にセンサ ID を識別子とするが、センサ ID をそのまま DOLR ネットワークに登録した場合、同じセンサ ID に対応する SensorAgent への探索要求を受け取ってしまうこととなる。これは不必要なトラヒックとなるだけでなく、不要な応答を回避する処理が必要となるため、ReceiverAgent ではプレフィックスを付与したセンサ ID を登録する。ReceiverAgent を探索する際には、プレフィックスを付与したセンサ ID を探索キーとして探索を行うことで、ReceiverAgent のみを探索し、同時に SensorAgent への探索と分離する。プレフィックスには、事前に定義したシステム全体で共通の文字列を用いる。

SensorAgent や ReceiverAgent は、センサ ID により識別、探索できるため、1つの DOLR ネットワーク上で複数のセンサデータストリーム配信グループを構築可能となる。

3.3.2 ノードの発見と配信経路の作成

本システムでは、配信元ノードから配信先ノードを発見する場合と、逆に配信先ノードから配信元ノードを発見する場合を想定して構築している。これらは、それぞれ SensorAgent が ReceiverAgent を探索する、逆に ReceiverAgent が SensorAgent を探索することで実現できる。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を想定している。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が待ち受ける Publish - Subscribe モデル類似の状況を想定している。

図 5 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している。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 (preregisterDestination)。この通知には、PIAX の発見型メッセージング機能の 1 つである発見型メソッド呼び出し機能 (discoveryCall; dc) を使用する。ここでの発見型メソッド呼び出しでは、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、LCF 法または LLF 法により配信経路 (route) を作成し、センサデータストリームをリクエストしてきた ReceiverAgent を含め、SensorAgent が配信するセンサデータストリームを受け取る全 ReceiverAgent に配布する (setRoute)。ここでは、エージェントに対する個別メソッド呼び出し機能 (call; c) を使用し、個々

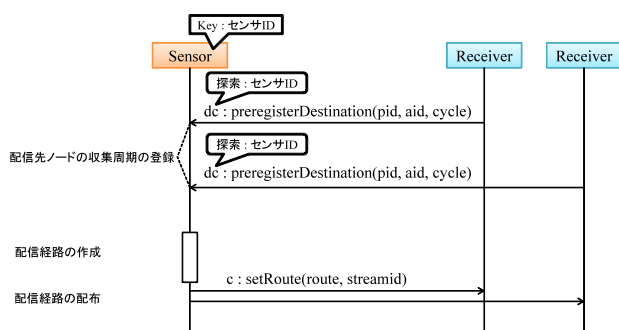


図 5 配信先ノードが配信元ノードを発見する場合の配信経路作成シーケンス

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

の ReceiverAgent にそれぞれ対応する配信経路を与える。ReceiverAgent は、SensorAgent から受け取った配信経路を「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。配信経路の作成時には、ストリーム ID (streamid) という ID を生成し、配信経路と併せて ReceiverAgent に配布している。ストリーム ID は、配信経路ごとにユニークとなる ID で、配信されるセンサデータストリームと配信経路を対応づけるために設けている。ストリーム ID がない場合、センサデータストリーム配送中に新たな配信経路が配布されると即座に新しい配信経路を使用することとなり、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かったりするという問題が生じる。

図 6 は、配信元ノードから配信先ノードを探索・発見する場合の動作シーケンスを示している。配信元ノード上の SensorAgent は、自身のセンサ ID にプレフィックスを付与したキーを探索条件として発見型メソッド呼び出しを行う (getDeliveryInfo)。この発見型メソッド呼び出しにより、SensorAgent は自身からのセンサデータストリームを受信しようとしている ReceiverAgent を発見し、それらの ReceiverAgent からそれぞれの識別情報 (pid, aid) と希望する収集周期 (cycle) を収集する。これ以降は、先に記した場合と同様に、配信経路の作成、全 ReceiverAgent への配布を行う。

3.3.3 センサデータストリーム配信

SensorAgent は、ReceiverAgent への配信経路の配布が終わると先に作成した配信経路に従い、ReceiverAgent にセンサデータの配信 (notifyData) を始める (図 7)。配信されるセンサデータにはストリーム ID が付与されており、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が持つストリーム ID と比較し、合致した場合はその配信経路に記された転送先の ReceiverAgent にセンサデータを再配信する。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

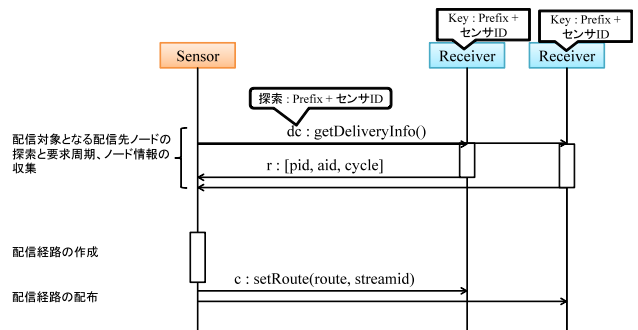


図 6 配信元ノードが配信先ノードを発見する場合の配信経路作成シーケンス

Fig. 6 Sequence of creating and dispatching a delivery route with discovering by a SensorAgent.

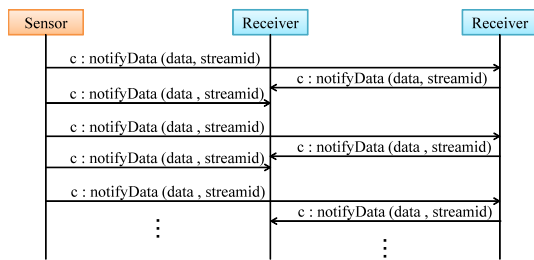


図 7 配信シーケンス

Fig. 7 Sequence of delivering a sensor data stream.

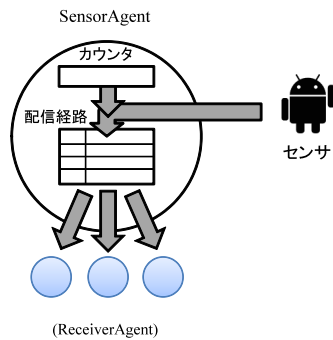


図 8 SensorAgent

Fig. 8 SensorAgent.

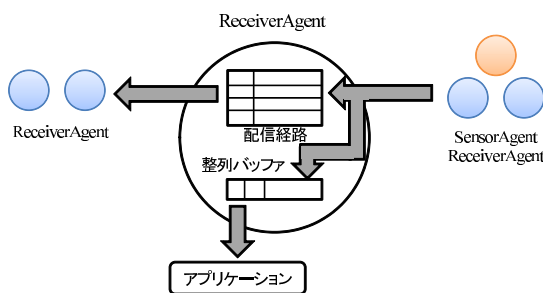


図 9 ReceiverAgent

Fig. 9 ReceiverAgent.

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を破棄し，保持していた配信経路に更新し，その配信経路に従いセンサデータを再配信する．保持していた配信経路のストリーム ID と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．

3.4 エージェントの実装

SensorAgent 内での処理を図 8 に示す．まず SensorAgent は，センサからセンサデータを定期的取得し，カウンタが発行する一連のシーケンス番号を付与する．次に，シーケンス番号と配信経路から，配信先の ReceiverAgent を決定し，その ReceiverAgent にセンサデータを配信する．配信経路は収集周期と配信先の対応表となり，シーケンス番号と収集周期を比較することで配信先を決定できる．

図 9 は ReceiverAgent 内での処理を示している．ReceiverAgent は，他のエージェントからセンサデータを受け

取ると，付与されているシーケンス番号をもとに配信経路を参照し，再配信先がある場合は，その配信先の ReceiverAgent にセンサデータを再配信する．同時に，受け取ったセンサデータを整列バッファに格納する．整列バッファは，次にアプリケーションに受け渡すべきセンサデータのシーケンス番号を保持しており，そのシーケンス番号を持つセンサデータが投入されるまで，それ以外のセンサデータを保持する．これにより，センサデータが異なる配信経路を通ることによる受信順序の入れ替わりに対処する．整列バッファによって順序関係が整ったセンサデータから順に，アプリケーションにセンサデータを受け渡す．

4. センサデータストリーム配信システムの評価

情報通信研究機構が提供している PIAX テストベッドを用い，実装したセンサデータストリーム配信システムの評価を行った．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計測困難な 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^{*2}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，同機構が整備・運営している新世代ネットワーク技術の研究開発テストベッドの JGN-X の一部として提供されている．

PIAX テストベッドでは，JGN-X の仮想マシン環境上に PIAX 実行基盤が用意されており，ユーザにはそれらの仮想マシンが複数台割り当てられる．2012 年 6 月現在，PIAX テストベッドは JGN-X 札幌 AP，NICT 大手町，NICT 小金井，NICT けいはんな，大阪大学の各拠点におかれた計 492 台の仮想マシンによって構成されており，各仮想マシンは JGN-X L2 サービスにより相互に接続されている．

ユーザは，ブラウザを用いてポータルサイトからログインし，作成したエージェントプログラムをパッケージ化したものをアップロード，ピアの割当て，エージェントプログラムの実行指示などの操作を行う．実行結果は，ログファイルを介して収集するほか，リアルタイムで実行状況を確認することもできる．

4.2 評価環境の構成

評価を行うノード構成は，配信に携わるノードとして配信元ノード 1 ノードと配信先ノード 10 ノード (1~10) の全 11 ノードを用いる．さらに，各配信ノードへのパラメータ設定や動作指示といった評価時の動作制御を行う制御

^{*2} PIAX テストベッド <https://piax.jgn-x.jp/>

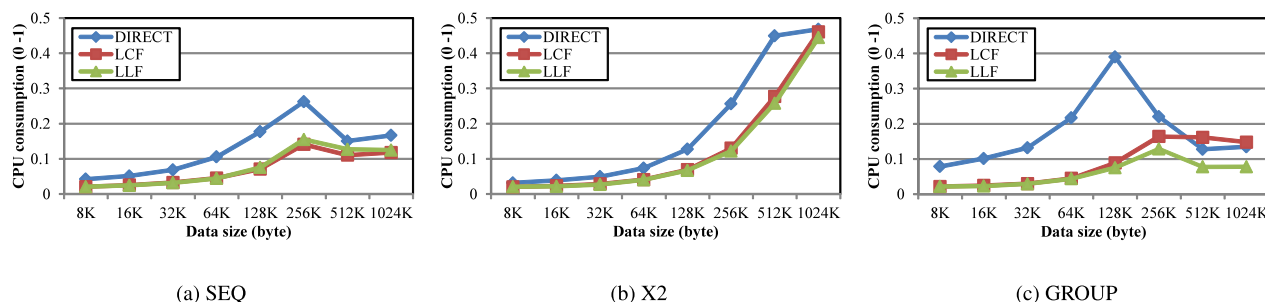


図 10 配信元ノードの CPU 負荷
Fig. 10 CPU consumption of the sensor data source.

表 2 評価に用いたサーバの割当て
Table 2 Assigned node for evaluation.

Node	0	1	2	3	4	5	6	7	8	9	10
Region	K	K	K	K	K	K	O	S	S	O	S

表 3 評価に用いた収集周期の割当て

Table 3 Assigned collection cycle for evaluation.

Node	1	2	3	4	5	6	7	8	9	10
SEQ	1	2	3	4	5	6	7	8	9	10
X2	1	2	4	8	16	32	64	128	256	512
GROUP	1	1	1	1	2	2	2	3	3	3

ノードを加えた構成とした。PIAX テストベッド上のサーバへの割当ては、制御ノード、配信元ノード、配信先ノード 1~5 はけいはんな、配信先ノード 6, 9 は大手町、配信先ノード 7, 8, 10 は札幌に設置されている仮想マシンを利用した (表 2)。表中、ノード 0 は配信元ノードを示しており、他はそれぞれ配信先ノード 1~10 を示している。

各配信先ノードが要求する収集周期として、次の 3 種のパターンを与えた。具体的な収集周期は表 3 のとおりとなる。

SEQ

1~10 の収集周期を順に割り当て

X2

前のノードの 2 倍の収集周期を割り当て

GROUP

同一の収集周期を持つノードからなるグループを構成するよう割り当て

評価に用いる配信経路は、DIRECT, LCF, LLF の 3 種を用いる。DIRECT は、配信元ノードから各配信先ノードへそれぞれの周期に応じたセンサデータを直接配信した場合、LCF は LCF 法、LLF は LLF 法により構築した配信経路を用いた結果となる。LLF 法での負荷推定に用いるパラメータ α と β には、送受信の負荷が同一であると仮定し、 $\alpha = \beta = 1$ とした。

4.3 ノード負荷の評価

4.3.1 CPU 負荷

LCF 法と LLF 法を用いることによる負荷の変化を確認するため、CPU 負荷の評価を行った。ここでのセンサデータには、実際の観測値の代用として 0 パディングされたダミーデータを用いる。このセンサデータを 25 ms 間隔で設定した送信予定時刻に配信元ノードに与え、各手法により配信先ノードに配信する。配信元ノードの配信処理に 25 ms 以上要し、次の送信予定時刻を超過した場合は、即座に次のセンサデータを配信先ノードに与える。配信期間は 300 秒とし、配信期間中の各ノードの CPU 負荷を計測する。ダミーデータは 8KB から 1,024KB まで 2 の乗数単位のサイズで与え、それぞれで計測を行う。実装したセンサデータストリーム配信システムは Java を用いて構築しており、JIT コンパイルの影響で処理時間に差異が生じる可能性がある。この影響を抑制するため、各計測に先立ちダミー実行を行っている。

図 10 (a)~ 図 10 (c) は、配信元ノードの CPU 負荷を示している。横軸はセンサデータのサイズ、縦軸は CPU 負荷としている。CPU 負荷は、配信期間中に評価プロセスに割り当てられた CPU 時間と配信期間との比より求めており、無負荷の場合は 0、全負荷の場合は 1 となる。図 11 (a)~ 図 11 (c) は、配信元ノードでのセンサデータの送信達成率を示している。送信達成率とは評価期間中に送信されるべきセンサデータ送信数に対して実際に送信されたセンサデータ数の割合を意味している。この評価では、センサデータを 25 ms 間隔で 300 秒間配信するため、すべて送信し終えた場合は 12,000 個のセンサデータが配信されることとなる。たとえば、評価期間の 300 秒間に 9,000 個しか送信できなかった場合、送信達成率は 75% となる。

各収集周期割当てとも、センサデータサイズの増加に合わせて CPU 負荷が増加した後に、あるデータサイズを境に CPU 負荷が減少している。この CPU 負荷が減少している範囲の送信達成率 (図 11 (a)~ 図 11 (c)) をそれぞれ参照すると、センサデータが送信しきれておらず、漏れが生じていることが分かる。この送信漏れは、配信元ノードにおいて、1 個のセンサデータの送信処理が 25 ms 以内に終了せ

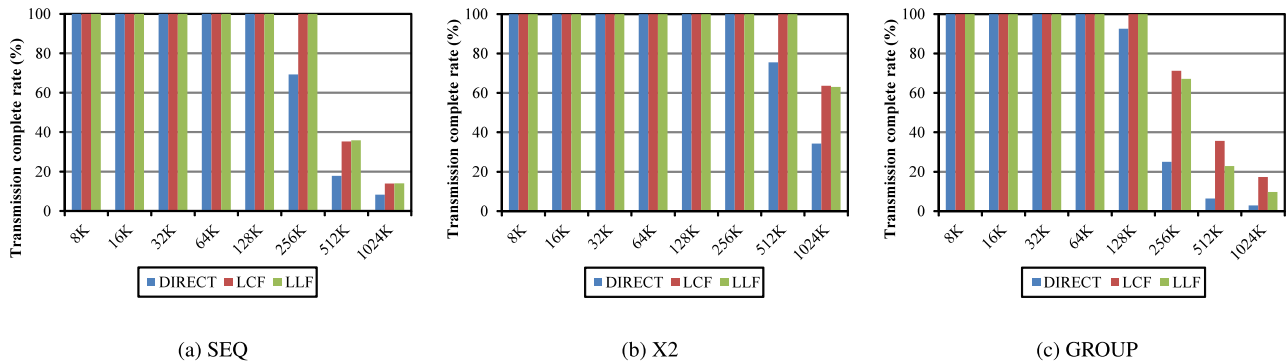


図 11 送信達成率

Fig. 11 Transmission complete rate.

ず、以降の送信予定時刻に対する遅れが累積することにより生じている。これは、現実装および PIAX テストベッド上のピア間の通信性能の上限（同一データセンタ内で 1 分間に約 2 GB）によるものと考えられる。加えて、PIAX テストベッドが稼動している各データセンタ間の RTT は、けいはんな・札幌間 26 ms、けいはんな・大手町間 10 ms、大手町・札幌間 17 ms となっており、配信経路がデータセンタをまたぐ場合は実効通信帯域が減少することから送信処理により時間を要すると考えられる。この送信処理中、配信システムのプロセスは OS が管理するプロトコルスタックのバッファが空くまで I/O 待ち状態となることから、配信システムのプロセスの CPU 負荷は減少したものと考えられる。

図 10(a) と図 10(b) では、LCF、LLF の負荷の推移がほぼ同等となっている。これは収集周期割当て SEQ と X2 では、それぞれの手法で構築される配信元ノードの送信先が同一となるため、収集周期割当てにより負荷の大小は異なるが、手法による違いは現れていない。

収集周期割当て X2（図 10(b)、図 11(b)）では、収集周期割当て SEQ と比較して、総配信センサデータに対して収集周期が相対的に短い配信先ノード 1～5 へ配信するセンサデータが占める割合が多くなる。遠隔地のノードが含まれる配信先ノード 6～10 への配信頻度が少なくなるため、LCF、LLF では 512 KB まで配信できており、それにとともに CPU 負荷も上昇している。

収集周期割当て GROUP（図 10(c)、図 11(c)）では、他の割当ての結果と異なり、LCF と LLF では 256 KB 以降、それぞれの方式で計測結果に差異が生じている。これは LCF 法では、配信元ノードの配信先が配信先ノード 1, 5, 8 となっており、それぞれけいはんな・けいはんな・札幌に配置されていることから RTT による送信遅延の影響が LLF 法の場合と比較して少ないためと考えられる。一方、LLF 法では、配信先ノード 4, 7, 10 が配信先となっており、それぞれけいはんな・札幌・札幌に配置されていることから、配信されるセンサデータの $\frac{2}{3}$ が札幌に配信され、結果として RTT による遅延の影響を受ける周期が多くな

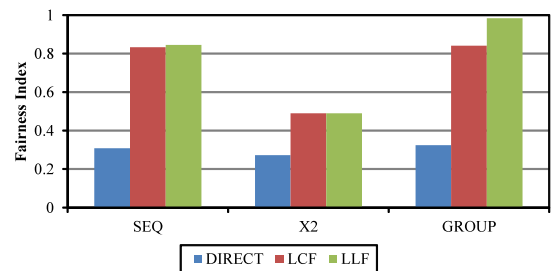


図 12 送受信センサデータ数による Fairness Index

Fig. 12 Fairness Index by the number of sensor data.

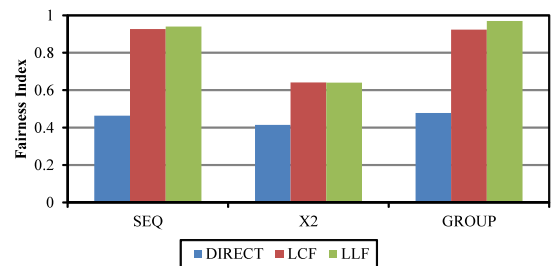


図 13 CPU 負荷による Fairness Index

Fig. 13 Fairness Index by CPU consumption.

るためである。

4.3.2 Fairness Index

図 12 と図 13 は 4.3.1 項での計測結果から得た Fairness Index の値を示している。それぞれ、センサデータサイズ 8 KB 時の送受信センサデータ数とセンサデータサイズ 32 KB 時の CPU 負荷から求めた値となっている。送信漏れがない状態で送信センサデータ数から得られる Fairness Index は、シミュレーションでの結果とほぼ同等になると考えられる。

4.4 配信遅延・ジッタ

通信路による影響を確認するため、配送遅延とジッタの評価を行った。けいはんなのノードのみで配信路を構成した場合と、遠隔地のノードを含む場合で評価を行い、ノード間の通信遅延の違いによる影響をみる。センサデータは 100 ms 間隔で配信し、実際の観測値の代用として 0 パディ

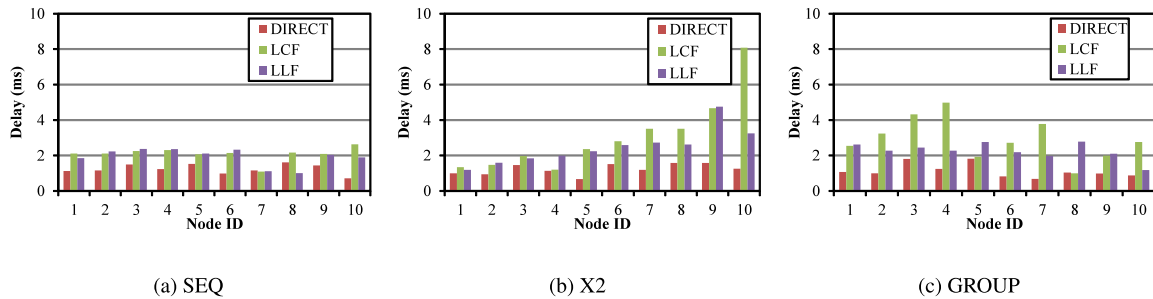


図 14 平均配信遅延時間 (けいはんな)

Fig. 14 Average delay of delivery (Keihanna region).

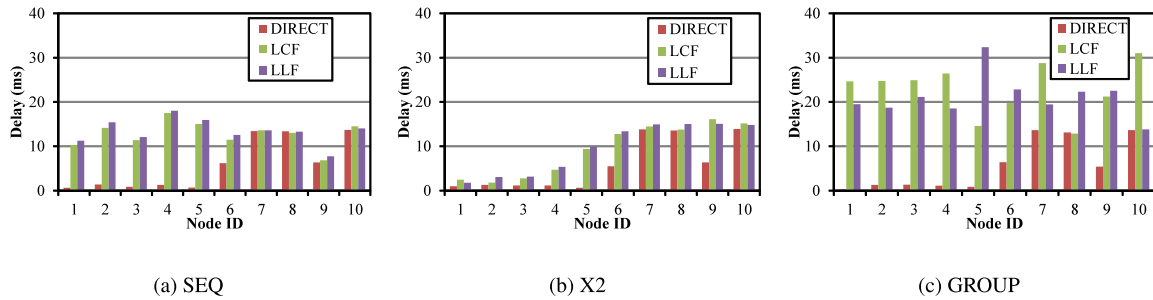


図 15 平均配信遅延時間 (3 拠点)

Fig. 15 Average delay of delivery (3 regions).

ングされた 1,024 バイトのダミーデータを用いる。このセンサデータを 10 分間配信することで計測を行う。

4.4.1 配信遅延時間

ここでは、あるセンサデータが配信元ノードにより配信され始める直前から、配信先ノード上のアプリケーションに受け渡されるまでに要する時間を配信遅延時間とする。配信先ノードでは、再配信が必要なセンサデータを受け取った場合、再配信を行った後にアプリケーションにセンサデータを受け渡す。

図 14 (a)～図 14 (c) は、けいはんなに設置されたノードのみで配信経路を構築した場合における各配信先ノードの平均配信遅延時間を示している。横軸は配信先ノード番号、縦軸は遅延時間 (ms) を示している。各ノード間の通信遅延時間は約 1 ms となっている。また、各図はそれぞれ、SEQ, X2, GROUP で割り当てた収集周期での遅延時間となっている。

DIRECT では配信先ノードによる再配信がないため、配信元ノードと各配信先ノード間の通信遅延の影響のみを受けており、収集周期割当てパターンの違いによる差異はほぼ見られない。LCF, LLF では再配信が行われる分、遅延時間が増加している。収集周期割当てパターン X2 での LCF (図 14 (b)) では、配信先ノード 10 は配信元ノードから受け取ったセンサデータを、他の全配信先ノードに再配信することから、再配信処理に時間を要し、LLF での同ノードより多く遅延することとなっている。また、収集周期割当てパターン GROUP での LCF (図 14 (c)) では、同一の収集周期のノード間はすべてのセンサデータがチェー

ン状に再配信されるため、その経路に沿って遅延時間が累積している。

図 15 (a)～図 15 (c) は、各ノードを表 2 に従って割り当てた遠隔地のノードを含む場合の配信遅延時間の計測結果を示している。DIRECT では配信先ノードによる再配信がないため、配信元ノードと各配信先ノード間の通信遅延の影響のみを受けており、収集周期割当てパターンの違いによる差異は見られない。

収集周期割当てパターン X2 (図 15 (b)) では、収集周期の倍数関係から配信先ノード 10 以外の配信先ノードは受信するセンサデータの半数を配信元ノードから直接受け取ること、また、配信先ノード 1～3 では要求するセンサデータの多くを同一設置場所にある配信先ノードから再配信を受けることから低遅延となっている。遠隔地に置かれた配信先ノード 6～10 からの再配信によるセンサデータの割合が増える配信先ノード 4 以降は、遠隔地のノードの配信遅延時間の影響を受けて配信遅延時間が長くなる傾向にある。

これに対して、収集周期割当てパターン SEQ (図 15 (a)) では、収集周期割当てパターン X2 と比較して、収集周期の短い配信先ノード 1～5 の配信遅延時間が延びている。これは、収集周期割当てパターン X2 とは異なり、他の配信先ノードから再配信を受ける割合が多くなるためと考えられ、倍数関係に乏しい配信先ノード 7～10 では、DIRECT とほぼ同等の配信遅延となっている。

収集周期割当てパターン GROUP (図 15 (c)) では、LCF においては配信先ノード 1～4 の遅延が他の収集周期割当

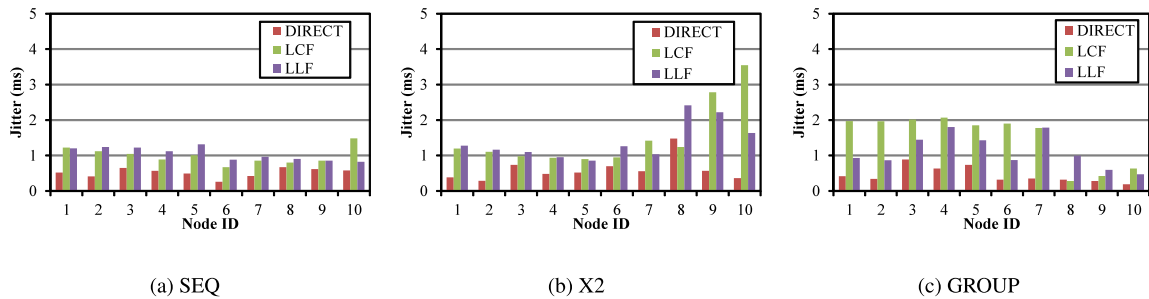


図 16 平均ジッタ (けいはんな)

Fig. 16 Average time of jitter (Keihanna region).

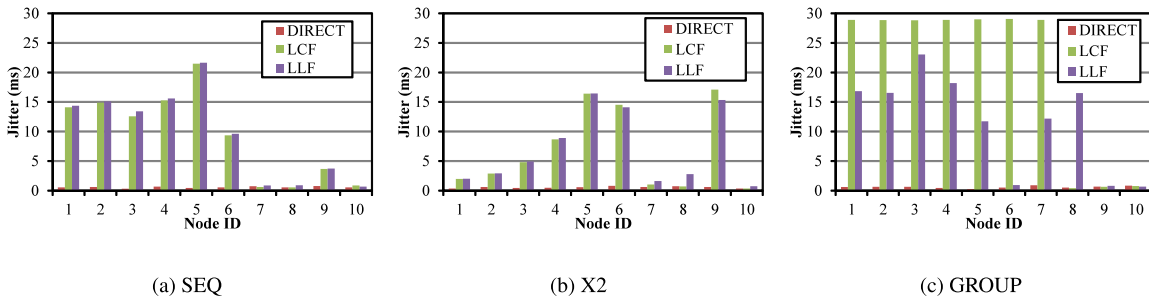


図 17 平均ジッタ (3 拠点)

Fig. 17 Average time of jitter (3 regions).

てパターンと比較して長くなっている。これらのノードが受信するセンサデータの $\frac{2}{3}$ が、ノード 5～10 の大手町・札幌に配置されたノードを介して再配信されており、それらの遅延が付加されることによる。また、LLF ではノード 5 の遅延が目立つが、周期 0 がノード 10, 9, 8 経由、周期 2 がノード 7, 6 経由と札幌・大手町の双方を経由することによる。

4.4.2 基準間隔からのジッタ

ジッタは、配信先ノードで受信されたセンサデータの到着間隔と、各配信先ノードのセンサデータ収集周期の差から求め、アプリケーションが受け取るセンサデータの到着間隔のぶれを表す。たとえば、収集周期 200 ms の受信ノードが他のノードからセンサデータを受け取った際、その時刻と直前のセンサデータ受信時刻の差が 190 ms であった場合、 $|200 - 190| = 10$ ms のジッタがあるとする。

図 16 (a)～図 16 (c) は、けいはんなに設置されたノードのみで配信経路を構築した場合における各配信先ノードでの平均ジッタを示しており、横軸はノード番号、縦軸はジッタの平均値 (ms) としている。各割当てパターンとも DIRECT では、配信元ノードからの直接配信になるため、大きな配信間隔の動揺はなく、1 ms 未満の平均ジッタとなっている。また、他の手法においてもほぼ 2 ms 以下の平均ジッタとなっているが、割当てパターン X2 の LCF (図 16 (b)) では、配信先ノード 9 と 10 でのジッタが 3 ms 前後となっている。これらのノードでは、再配信先のノード数が 8～9 ノードと他のノードと比較して多いため、初回の再配信時に接続を確立する時間を要すること

と再配信処理での処理時間のぶれが生じやすいことから他のノードと比較して大きなジッタとなったと推測される。これは、LLF においても同様の傾向が見られるが、LLF では配信先ノード 10 の再配信先が 6 ノードであるため LCF よりもジッタは小さくなっている。

収集周期割当てパターン GROUP (図 16 (c)) では、LCF での配信先ノード 1～7 でジッタが約 2 ms となっている。まず収集周期 2 の配信先ノード 5～7 では、受け取るセンサデータの $\frac{1}{3}$ が収集周期 3 のノードから受け取ることになる。その他のセンサデータはノード 5 から 7 の一定の経路を通るため、この差がジッタとして現れている。収集周期 1 の配信先ノード 1～4 においても同様に、経路の差異から約 2 ms のジッタとなっている。

図 17 (a)～図 17 (c) は、各ノードを表 2 に従って割り当てた遠隔地のノードを含む場合の配信遅延時間の計測結果を示している。収集周期割当てパターン SEQ (図 17 (a)) では、LCF, LLF とともに配信先ノード 7, 8, 10 のジッタが他の配信先ノードと比較して小さくなっている。これらの配信先ノードでは、収集周期の倍数関係から配信元ノードから直接センサデータを受け取る割合が多いため、センサデータの到着間隔が安定し、ジッタが小さくなっていると考えられる。配信先ノード 5 では、収集周期が 5 であることから、収集周期が 10 である配信先ノード 10 からの再配信が受け取るセンサデータの半数を占めることとなる。配信先ノード 10 は札幌に置かれているため、再配信時には往復分の遅延が入ることとなり、配信元ノードから直接受け取ったセンサデータと比較して、到着間隔の差分が大

きくなることから、約 21 ms のジッタとなっている。配信先ノード 1, 2, 3, 4, 6 においても、同理由により 9 ms から 15 ms のジッタとなっている。

収集周期割当てパターン X2 (図 17(b)) では、収集周期割当てパターン SEQ の場合と比較して、配信先ノード 1~4 のジッタが小さくなっている。配信先ノード 1, 2 においては、配信元ノードが同一の場所に置かれていること、再配信して受け取ったセンサデータのうち、同一の場所に置かれた配信先ノードから受け取る割合が大きいため、センサデータの受信間隔に大きな差は生じずジッタは小さくなる。配信先ノード 3, 4 では、他地域に置かれた配信先ノードから受け取るセンサデータの割合が増えるため、配信先ノード 1, 2 と比較してジッタが大きくなる。配信先ノード 5, 6 や配信先ノード 9 では、他地域に置かれた配信先ノードから受け取るセンサデータの割合が多くなるため、ジッタが大きくなっていると考えられる。

収集周期割当てパターン GROUP (図 17(c)) では、LCF での配信先ノード 1~7 でジッタが約 28 ms となっている。まず収集周期 2 の配信先ノード 5~7 では、受け取るセンサデータの $\frac{1}{3}$ が収集周期 3 のノードから受け取ることになる。このとき、札幌に配置されているノード 8, 9 を介することになる。その他のセンサデータはノード 5 から 7 の一定の経路を通るため、この差が 28 ms のジッタとして現れている。まず収集周期 1 の配信先ノード 1~4 は、受け取るセンサデータの $\frac{2}{3}$ がノード 5~7, 8~10 を介して配信されており、それぞれの経路の差異から 28 ms のジッタとなっている。

4.5 考察

4.3.1 項の評価結果より、DIRECT と比較して LCF 法、LLF 法ともに配信元ノードの負荷を抑制できていることが分かる。特に比較的短周期かつ同一のセンサデータを複数の配信先に送信することとなる GROUP では、配信元ノードの負荷が大きく抑制できている。また、4.3.2 項の評価結果より、各収集周期割当てパターンともに、DIRECT と比較して LCF 法、LLF 法では Fairness Index は大きく改善されており、各ノードの負荷が均等化されたといえる。LCF 法と LLF 法の比較では、SEQ, X2 ではほぼ差が生じていないが、LCF 法が不得手とする GROUP において LLF 法では Fairness Index が改善されていることが分かる。

ここで図 12 と図 13 を比較すると、CPU の負荷から得られた Fairness Index は、送受信センサデータ数から得た Fairness Index より公平度が上昇していることが分かる。差異が目立つ収集周期割当て GROUP での LCF 法において、個々のノードの CPU 負荷と送受信センサデータ数の比較を図 18 に示す。直接の比較はできないため、各最大値を 1 とする正規化を行い併記した。この比較より、送

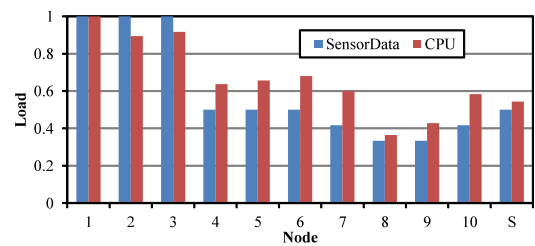


図 18 送受信センサデータ数と CPU 負荷の比較 (LCF, GROUP)

Fig. 18 Comparison of the number of transmitting/receiving sensor data and CPU load (LCF, GROUP).

受信センサデータ数が少ない状態であっても、CPU 負荷はその比率に従って減少せず、相対的に負荷が高くなっていることが分かる。このため、送受信センサデータ数が少ないノードにおいても、相対的に CPU 負荷が高くなるため、CPU 負荷より求められた Fairness Index では送受信センサデータ数より求めた Fairness Index よりも均一であるという結果になったと考えられる。また、この送受信メッセージ数と CPU 負荷の挙動の不一致は、LLF 法での負荷推定にも影響を与えていると考えられる。現在、LLF 法での負荷推定には、2.2 節の負荷モデルを用いており、各ノードが送受信するセンサデータ数と CPU 負荷が比例すると想定しているが、実測評価の結果からはセンサデータ数と CPU 負荷は必ずしも単純な比例関係では表せないことがうかがえる。このため、より正確な負荷推定を行うには負荷推定モデルの改善が必要になると考えられる。

4.4.1 項と 4.4.2 項の評価結果では、LCF 法により構築された経路では、収集周期の分布により一部の配信先ノードに再配信処理が集中し、そのノードの配信遅延時間、ジッタともに大きくなる場合が生じている。また、同じ収集周期を持つ配信先ノード間では、センサデータがチェーン状に配信されるため、配信遅延が累積しやすいといえる。LLF 法では、収集周期の長いノードの中から負荷が低いノードを選択してセンサデータを受け取ることとなるため、配信遅延時間は LCF 法と比べて、均等になる傾向があるといえる。

また、各ノード間の地理的距離の差異により通信遅延が不均一である場合、一部の配信先ノードに再配信処理が集中することによる遅延やジッタの影響は相対的に少なくなるが、通信遅延の大きいノード間で再配信が行われることで、配信遅延やジッタが増大することとなる。これまで提案されていた LCF 法、LLF 法では、これらのノード間の通信遅延については考慮されていないため、負荷の均一化よりも時間面での制約を重視する場合は、経路構築時に通信遅延の小さいノードを優先的に使用したり、最大ホップ数に制約を設けたりするなどの手法が必要になると考えられる。

5. まとめ

本稿では、収集周期の異なるセンサデータストリームのための Peer-to-Peer 型配信システムの PIAX を用いた設計と実装について述べた。また、情報通信研究機構が提供している PIAX テストベッドを用いて、実装したセンサデータストリーム配信システムの配信時の CPU 負荷、および配信遅延時間とジッタについて実測評価を行った。評価の結果、LCF 法、LLF 法の違いによる負荷傾向の差異とノード間の通信遅延の大小により生じる問題点について明らかにした。今後は、複数のセンサデータストリームを同時に扱う配信手法や配信時の遅延時間に焦点をあてた配信手法について検討を進める予定である。

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」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] 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).
- [2] Yu, L., Liao, X., Jin, H. and Jiang, W.: Integrated Buffering Schemes for P2P VoD Services, *Peer-to-Peer Networking and Applications*, Vol.4, No.1, pp.63–74 (2011).
- [3] 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).
- [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] 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).
- [6] Le, T.A. and Nguyen, H.: Application-Aware Cost Function and Its Performance Evaluation over Scalable Video Conferencing Services on Heterogeneous Networks, *Proc. 2012 IEEE Wireless Communications and Networking Conference (WCNC 2012)*, pp.2185–2190 (2012).
- [7] 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).
- [8] 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).
- [9] 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).
- [10] Kaneko, Y., Harumoto, K., Fukumura, S., Shimojo, S. and Nishio, S.: A Location-based Peer-to-Peer Network in a Ubiquitous Environment, *IPSSJ Journal: Database*, Vol.46, No.18, pp.1–15 (2005).
- [11] 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).
- [12] Aspnes, J. and Shah, G.: Skip Graphs, *ACM Trans. Algorithms*, Vol.3, No.4, p.37 (2007).



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となり現在に至る。博士 (情報科学) (2013 年 3 月, 大阪大学)。P2P ネットワークに関する研究に従事。



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。