

# コンシステントハッシュ法を用いた複数センサデータ ストリーム配信システムの PIAX テストベッドを用いた評価

石 芳正<sup>1</sup> 川上 朋也<sup>1,2</sup> 義久 智樹<sup>1</sup> 寺西 裕一<sup>1,3</sup>

**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。現在、複数の配信元と複数の配信先からなる多対多での配信を実現するため、配信元-中継網-配信先の 3 階層からなる配信モデルを検討しており、コンシステントハッシュ法を用いてセンサデータストリームを中継するノードを選択することで中継網内での負荷分散を行う手法を提案している。本研究では、提案手法とその実装システムを PIAX テストベッドを用いた実機環境において評価し、その結果、提案手法では中継ノード間の負荷を公平化し、1 つのセンサノードが発するセンサデータストリームをより多くの受信ノードへ配信しうることを確認した。

YOSHIMASA ISHI<sup>1</sup> TOMOYA KAWAKAMI<sup>1,2</sup> TOMOKI YOSHIHISA<sup>1</sup> YUICHI TERANISHI<sup>1,3</sup>

## 1 はじめに

IoT (Internet of Things) や M2M (Machine to Machine) という言葉が一般誌や報道においても頻繁に用いられるようになり、様々なデバイスをネットワークに接続し相互に情報をやりとりすることで、これまで個別に動作していた機器では実現困難なサービスの実現を目指す技術開発に注目が集まっている。その中でも、環境情報の収集源となるセンサデバイスとの連携に関わる技術はセンサデバイスから得られる情報やイベントトリガを用いてリアルタイム・準リアルタイムで動作するサービスや、センサデバイスからの情報を適切なアーカイブ先に転送する仕組みといったサービス実現の基盤として重要な役割を果たす。

センサデバイスにおいても、近年の技術開発やスマートフォンの普及にともなう量産効果により、小型・高精度にも関わらず安価に利用できるセンサ素子が登場しており、それらのセンサ素子を搭載したセンサデバイスを安価に製

造できるようになっている。これにより、センサデバイスを様々な場所に設置することが容易となり、インフラの状態監視用途や省エネといった各分野において導入が進むと予想されている。そして、このように設置された多数のセンサデバイスからは多種多様な観測データ、センサデータが得られることとなる。

これらのセンサデバイスは、多くの場合、観測対象を周期的に観測し、その観測結果を周期的に出力する。たとえば、気象センサの場合では数秒から数分に 1 回、カメラデバイスの場合では、1 秒間に 10 枚から 60 枚程度の画像が出力されている。このような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をセンサデータストリームと呼び、センサデバイスより逐次送出される観測値のストリームデータを直接監視したり解析処理をすることにより、監視業務や環境モニタリングといったリアルタイムサービスに利用できる。さらにセンサデータストリームをネットワークを介して様々な利用先へと配信することにより、1 つのセンサを複数のサービスや利用者で共用することが可能となる。これにより、センサデータストリームの利用者からはそれぞれ個別に観測網を構築する必要がなくなり、センサ機器の重複設置を避けた効率のよい観測網を構築することができる。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

<sup>1</sup> 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

<sup>2</sup> 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

<sup>3</sup> 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

トリーム配信と呼ばれている。

データストリームの配信技術については、これまでにビデオストリームの配信を中心とした様々な手法が提案されている [1,2,3,4]。これらの既存研究は、あるデータストリームを複数の配信先に配信する場合に、配信元のサーバにかかる負荷を軽減するためデータストリーム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となっている。これらの手法は、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おいて有効に機能するが、配信するデータストリームがセンサデータストリームである場合には、配信先の要望を効率良く実現することが難しい。

センサデータストリーム配信では、従来一般的なデータストリーム配信とは異なり、配信先により異なる周期のセンサデータが要求される場合、すなわち時間方向で異なる解像度のセンサデータを要求される場合が想定される。たとえば、同じ交通カメラが出力する画像ストリームであっても、通過車両の識別を目的とするサービスと、交通流量を監視・記録するサービスでは必要とされるフレームレートが異なる場合が考えられる。前者では、撮影範囲においてそれぞれの車両を抽出・識別できる映像フレームを探索するため、可能な限り多くのフレーム、一般にはカメラが出力する全フレームを要求される。その一方で、後者の場合ではフレームの間隙において通過してしまう車両が出ない程度であればよいから、前者ほどのフレームレートは要求されない。一般的な映像ストリーム配信技術では、両者に同じ映像ストリームを配信するため、後者のサービスにおいては過剰品質な映像ストリームが配信されることとなり、ネットワーク帯域や映像ストリームの配信先端末に無駄な負荷を生じさせることとなる。

我々の研究グループは、配信先が異なる周期のセンサデータを要求し得るというセンサデータストリーム配信において、不必要なセンサデータの配信を抑制し、配信元・配信先の送受信負荷を分散させた配信経路を構築する手法を提案・実装してきた。1つのセンサデバイスから得られる1センサデータストリームを複数の配信先に配信する手法として、LCF法 (Longest Cycle First; 最長周期優先) や LLF法 (Lowest Load First; 最小負荷優先)、LLF-H法 (Lowest Load First considering Hops) といった手法を提案している [5,6,7]。また、これらの実装システムを用いた PIAX テストベッド上での実機評価において、実際に負荷が公平化することを確認している [8,9]。さらに、複数のセンサデバイスが生成するセンサデータストリームを複数の配信先へと配信する多対多の配信を実現するため、コンシステントハッシュ法を用いて、センサデータストリームを中継するノードを選出する手法 [10] を提案しているが、その評価はシミュレーションによるものに留まっている。

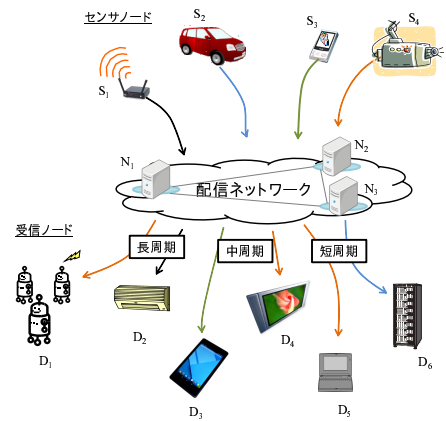


図1 配信モデル  
Fig. 1 Delivery model.

そこで本研究では、Peer-to-Peer エージェントフレームワーク PIAX による実装システム [11] をもとに、提案手法が実環境において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 配信モデル

図1に、本システムで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。図中上部の  $S_1 \sim S_4$  は、それぞれセンサデバイスを表しており、これらのセンサデバイスはそれぞれ温度や騒音といった特定の観測対象を周期的にセンシングし、その観測値と任意のメタデータからセンサデータを生成する。そして、一連のセンサデータよりセンサデータストリームを生成する。これらのセンサデバイスは、単純なセンサ素子そのものではなく TCP/IP などの通信機能を有し、ある程度のインテリジェント性を持つデバイスを想定しており、インターネットを介して図中中部の配信ネットワーク内の任意のノードに接続する機能を持つ。センサデータストリームを生成し、それらを配信ネットワークに送信する機能を持つセンサデバイスをセンサノードと呼ぶ。図中中部の  $N_1, N_2, N_3$  は、センサデータストリームの中継・処理を行う配信ネットワークを構成している中継ノードを示している。配信ネットワークでは、センサノードから受けたセンサデータストリームをもとに、事前に指定・設定された任意の配信周期のセンサデータストリームを生成する。本研究では、これらの中継ノードはデータセンタなどの安定した環境下に集約して設置され、クラウドサービスとしてセンサデータストリーム配信サービスが提供される用途を想定している。図中下部の  $D_1 \sim D_6$  は、センサデータストリームを受信し、それぞれに利用する受信端末を示しており、これら受信端末を受信ノードと呼ぶ。受信ノードでは、配信ネットワークが配信して

表 1 受信ノードが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	...
$S_4$ (Cycle=1)	○	○	○	○	○	○	○	...
$D_1$ (Cycle=3)	○			○			○	...
$D_4$ (Cycle=2)	○		○		○		○	...
$D_5$ (Cycle=1)	○	○	○	○	○	○	○	...

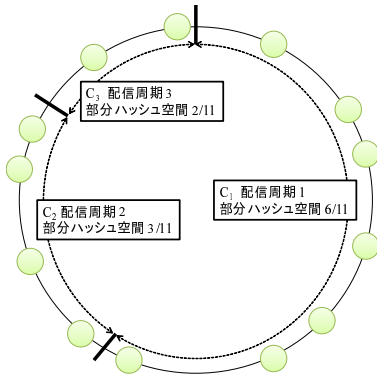


図 2 配信周期によるハッシュ空間の分割  
Fig. 2 Dividing hash space by delivery cycles.

いるセンサデータストリームとその配信周期を選択し、自身の希望に添った配信周期のセンサデータストリームを受信することとなる。センサノードから配信ネットワークへの矢印と、配信ネットワークから受信ノードへの矢印はセンサデータストリームの流れを示しており、色分けにより元のセンサデータストリームを生成したセンサノードの違いを表している。

表 1 は、センサノード  $S_4$  が送信しているセンサデータストリームと、そのセンサノードのセンサデータを受信する受信ノード  $D_1, D_4, D_5$  がそれぞれ必要とするセンサデータの配信周期 (Cycle) と、その配信周期において受け取るセンサデータを ○ で示している。

センサノード  $S_4$  は、自身の観測周期に合わせてセンサデータを生成し、その周期によるセンサデータストリームを送信する。これに対して受信ノード  $D_1$  は、長周期でのセンサデータストリームを要求しており、センサノードが生成したセンサデータストリームから、時刻 3 ごとに受け取る配信周期が 3 のセンサデータストリームを受信する。同様に  $D_4$  は、中周期のセンサデータストリームとして、センサデータを時刻 2 ごとに受け取る配信周期が 2 のセンサデータストリームを受信する。 $D_5$  は、短周期のセンサデータストリームとして、配信周期が 1、即ちセンサノード  $S_4$  が生成したセンサデータのすべてを受け取ることとなる。

## 2.2 配信ネットワークの構成

提案手法 [10] では、センサノードから送信されるセン

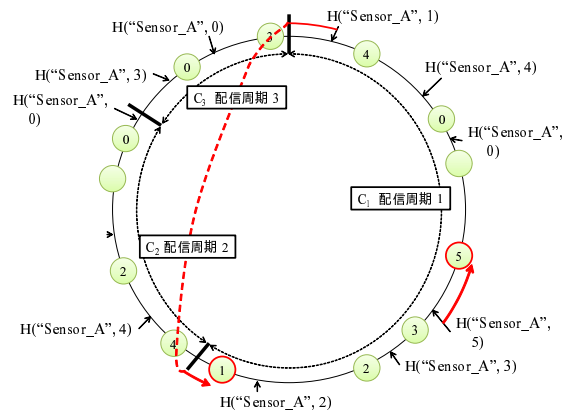


図 3 担当中継ノードの割当て  
Fig. 3 Assignment of relay nodes.

サデータストリームを中継するノードを決定するために、コンシステントハッシュ法の考えを利用している (図 2)。この手法では、まず中継ノードを分散ハッシュテーブルの一手法である Chord [12] と同様に環状のハッシュ空間上に分散配置する。これには、中継ノードのノード名やノードの IP アドレスなど、中継ノードごとに異なる情報のハッシュ値を求め、その値をハッシュ空間上での中継ノードの位置とすることで行う。次に、環状のハッシュ空間を、配信ネットワークにおいて生成する配信周期ごとの部分ハッシュ空間に分割する。この際、部分ハッシュ空間の大きさを各周期の逆比で割り当てることにより、単位時間あたりにより多くのセンサデータを扱う短い周期のセンサデータストリームほど多くの中継ノードを割り当てられることとなる。たとえば、選択可能な周期  $C_i = i$  ( $i = 1, 2, 3$ ) の場合には、 $1/C_1 : 1/C_2 : 1/C_3 = 1/1 : 1/2 : 1/3 = 6 : 3 : 2$  の比で全ハッシュ空間を分割し、部分ハッシュ空間とする。それぞれの部分ハッシュ空間は、部分ハッシュ空間ごとに始点と終点を繋いだ環状として扱い、その部分ハッシュ空間上のノードによりその配信周期の配信グループを構成する。周期の部分ハッシュ空間上にノードが存在しない場合には、その前の部分ハッシュ空間上の最近傍ノードがその周期グループも担当する。

部分ハッシュ空間への分割後は、各部分ハッシュ空間において個々のセンサデータの中継する中継ノードを割り当てる。異なる周期のセンサデータストリーム配信を行う場合、それぞれの周期の最小公倍数周期で同じ配信パターンが繰り返される。たとえば、表 1 の場合、各配信周期の最小公倍数の 6 周期ごとに同パターンでの配信が繰り返されることとなる。この繰り返される配信パターンの 1 周期を 1 巡回周期とし、配信時刻と巡回周期長の剰余により得られる巡回周期内の各周期を配信インデックスと呼ぶこととする。配信パターンが周期的に繰り返されることから、中継ノードの割当ては各配信グループにおいて、配信イン

デックスに対応する中継ノードを担当中継ノードとして割り当てればよいこととなる。この割り当ては、コンシステントハッシュ法により決定する。まず、センサ ID と配信インデックスをハッシュ関数  $H$  に与えてハッシュ値を求め、そのハッシュ値を超えない最大のハッシュ値を持つ中継ノードがその配信インデックスのセンサデータを中継する担当中継ノードとなる (図 3)。たとえば、配信周期 1 の部分ハッシュ空間において、配信インデックス 5 のセンサデータはセンサ ID と配信インデックスのハッシュ値により、ハッシュ円上 4 時の位置に  $H("Sensor\_A", 5)$  で示す点となり、そのセンサデータの担当中継ノードは反時計回りで隣の中継ノードになる。また、ハッシュ円上 0 時半の位置となる  $H("Sensor\_A", 1)$  では、反時計回りで隣のノードは配信周期 3 の領域のノードとなるが、部分ハッシュ空間は環状として扱うため、図中赤破線に従いハッシュ円上 7 時の中継ノードが担当することとなる。

センサノードは、選択可能とする配信周期の最小公倍数と配信インデックスをもとにハッシュ値を求め、ハッシュ値に対応する担当中継ノードにセンサデータを送信する。送信しようとする配信インデックスにおいて、送出先が複数となる場合、たとえば配信インデックスが 0 の場合では、すべての部分ハッシュ空間に担当中継ノードが存在するが、この場合は単位時間あたりに中継するメッセージ数が少なく、相対的に負荷が低い配信周期が最も長いグループの中継ノードに送信する。

センサデータを受け取った中継ノードは、そのセンサデータを担当している配信周期を要求している受信ノードに送信する。また、そのセンサデータが他の配信周期と共通であった場合、センサノードから直接センサデータを受け取った最長周期のグループの中継ノードはその他のグループの担当中継ノードに送信することで配信グループ間のセンサデータの受け渡しを行う。

以上の手順に従い、センサノードが生成するセンサデータストリームごとに中継ノードのグループ分け、担当ノードの割り当てを行い、各中継ノードが複数のセンサデータストリームを扱う事で配信ネットワーク内の負荷の公平化を図る。

### 3 センサデータストリーム配信システムの実装

本研究でのセンサデータストリーム配信システムの実装では、先行研究と同様に PIAX [13]\*1 を用いて実装した。

#### 3.1 PIAX の概要

PIAX は、NICT や大阪大学を中心に研究開発されている 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プラットフォームミドルウェアである。

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

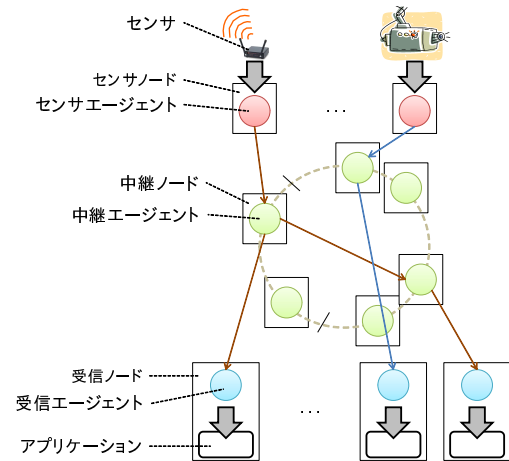


図 4 エージェントを用いた配信モデル

Fig. 4 Structure of the sensor data stream delivery system.

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。ピア間は、PIAX がサポートするオーバレイネットワークにより相互に接続され、オーバ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の相互メッセージング、任意オブジェクトの送受信、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。本実装では、センサノード、受信ノードともに初期状態ではセンサデータを中継する中継ノードの情報を持たない前提とし、センサノードがそのセンサデータストリームの配信を始める際、受信ノードがセンサデータストリームを要求する際に必要な中継ノードへの接続を行う。その際に、PIAX の探索機能やメッセージング機能を用いて相互発見やその後のセンサデータの送受信を行う。

#### 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。センサデータストリーム配信システムは、センサが接続された 1 台以上のセンサノードと、配信ネットワークを構成する 1 台以上の中継ノード、センサデータストリームの利用者となる 1 台以上の受信ノードにより構成される。各ノード上では、それぞれ PIAX のピアを動作させ、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する。センサには、センサを識別するためのユニークなセンサ ID が割り振られているものとする。受信ノード上では、最終的にセンサデータストリームを利用するアプリケーションが動作している。センサノード上にはセンサに対応したセンサエージェント、中継ノード上には中継エージェント、受信ノード上には受信エージェントをそれぞれ配置し、これらのエージェント間の連携により配信システムを実現する。

センサエージェントの構造を図 5 に示す。1 つのセンサ



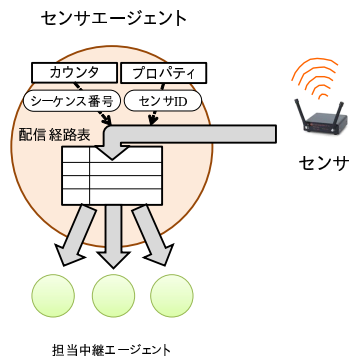


図 5 センサエージェント  
Fig. 5 Structure of the SensorAgent.

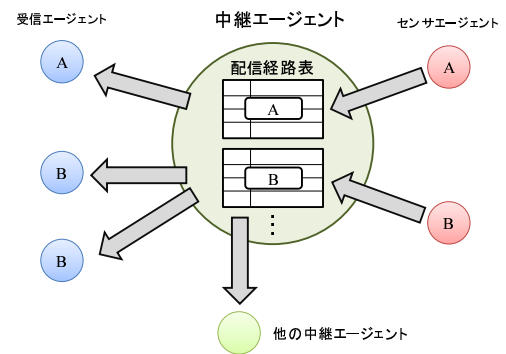


図 6 中継エージェント  
Fig. 6 Structure of the RelayAgent.

エージェントは、1つのセンサに対応し、センサより定期的にセンサデータを取得する機能を持つ。センサエージェントは、プロパティとカウンタ、配信経路表を持つ。プロパティは、センサIDなどの属性情報を保持する。カウンタは、1つのセンサデータを取得するごとに1ずつ加算され、連続したシーケンス番号を生成する。配信経路表は配信インデックスと担当中継エージェントの対応表になる。センサエージェントは、センサから得られたセンサデータにセンサに対応するセンサIDとカウンタが発行するシーケンス番号をメタデータとして付与する。次に、シーケンス番号と選択可能な配信周期の最小公倍数との剰余から配信インデックスを算出し、その配信インデックスに該当する中継エージェントを配信経路表より取得し、その中継エージェントにセンサデータを送信する。

センサエージェントがセンサデータストリームの配信を始める際には、後述する中継エージェントに自身のセンサIDと配信するセンサデータストリームの選択可能な配信周期、たとえば(1, 2, 3)をパラメータとしてセンサデータストリーム配信要求を出し、配信経路を構築させる。このとき、要求をだす中継エージェントは任意の中継エージェントでよい。そして、その応答として自身の配信経路表に必要な情報として、配信インデックスと中継ノードの対応表を取得する。

図6は、中継エージェントの構造を示している。中継エージェントは、センサごとに対応する配信経路表を持ち、これらはセンサIDにより識別される。図の例では、センサA用とセンサB用の配信経路を持っている。中継エージェントの配信経路表は、配信インデックスとその周期のセンサデータを受信する受信ノード、あるいは他の中継ノードとの対応表となる。センサエージェントから配信されたセンサデータは、そこに付与されていたセンサIDから対応する配信経路表を特定し、センサデータのシーケンス番号から配信先のエージェントを決定して、対応する受信エージェント、中継エージェントにセンサデータを送信する。

センサデータに付与されていたセンサIDが、未知のセンサIDであった場合は不正データと見なし破棄する。

中継エージェントは、以上のセンサデータの中継機能に加えて、配信経路表の生成・共有機能も持つ。中継エージェントは、センサエージェントからのセンサデータストリーム配信要求に対して、その要求の配信周期に応じた配信経路を生成し、配信インデックスとそれに対応する担当中継エージェントの情報をセンサエージェントに応答する。生成された配信経路において、中継エージェント間でのセンサデータの授受がある場合は、中継元となる中継エージェントに中継すべきセンサデータのセンサID、配信インデックス、送信先となる中継エージェントの情報を与え、センサIDに対応する配信経路表として保持させる。ここで生成された配信経路は、中継エージェントで構成する共有ストレージに格納する。受信エージェントよりセンサデータストリームの受信要求が与えられた場合、中継エージェントは共有ストレージより配信経路を取得し、要求された配信周期のセンサデータストリームを配信可能か判定し、配信可能な周期であった場合は、そのセンサデータストリームを中継している中継エージェントの情報を受信エージェントに応答する。

図7は、受信エージェント内での処理を示している。受信エージェントは、中継エージェントからセンサデータを受け取ると、その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えられたセンサデータは順番に取り出され、センサエージェントにより付与されていたシーケンス番号やセンサIDなどのメタデータを削除し、もとのセンサデータのみをセンサデータストリームとして再構築し、アプリケーションに逐次受け

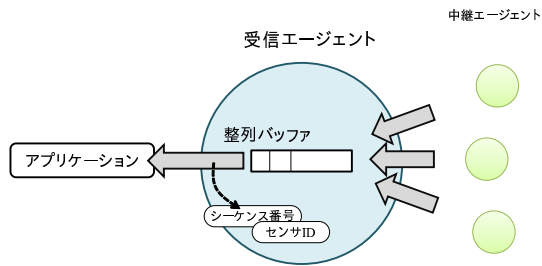


図 7 受信エージェント

Fig. 7 Structure of the ReceiverAgent.

渡す。

受信エージェントは、センサデータストリームを受信しようとする際、そのセンサデータストリームのセンサ ID と受信しようとする配信周期を任意の中継エージェントに要求する。その応答として、配信インデックスと中継エージェントの対応表が得られるため、対応表内の各中継エージェントに対し配信インデックスに対応するセンサデータを送信するよう要求を出す。

なお、現実装においては部分ハッシュ空間の循環処理は未実装となっており、ハッシュ値に対応する中継ノードを探索する際には、部分ハッシュ空間の境界を考慮せずに反時計回りで隣接している中継ノードを取得している。

## 4 PIAX テストベッドを用いた評価

3章で述べた実装をもとに負荷計測機能を追加し、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評価を行った。

### 4.1 PIAX テストベッドの概要

PIAX テストベッド<sup>\*2</sup>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、国内 5 拠点（北海道、東京、神奈川、石川、京都）に設置された計 1,322 台<sup>\*3</sup>の仮想マシン上に PIAX エージェントを用いたプログラムの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。ユーザは、ブラウザを用いて自身の PIAX エージェントプログラムをアップロードし、その評価を行うことができる。各仮想マシンは JGN-X L2 サービスを介して接続され、実験用ネットワークを構成している。

<sup>\*2</sup> PIAX テストベッド <https://piax.jgn-x.jp/>

<sup>\*3</sup> 2015 年 5 月時点

## 4.2 多対多配信の評価

### 4.2.1 評価環境と評価指標

負荷計測を行う配信環境として、石川拠点で 9 VM、神奈川拠点と京都拠点でそれぞれ 10 VM を割り当てる。石川拠点では 1 VM 上で 1~2 センサノード、神奈川拠点では 1 VM 上で中継ノードを 1 ノード、京都拠点では、1 VM 上で 10 受信ノードを動作させ、センサノード 10 ノード、中継ノード 10 ノード、受信ノード 100 ノードとした。

ハッシュ空間上での中継ノードの配置は、理想的な中継ノード配置の FIX と、実運用時の状態を反映した HASH の 2 種とする。FIX はハッシュ空間を中継ノードの数で等分割し、その境界点を各中継ノードに割り当てる。HASH では、中継ノードのノード名のハッシュ値によりその中継ノードをハッシュ空間上に割り当てる。

センサノードは 1~6 の 6 種の配信周期から 1 つ以上の配信周期を選択し、自身が送信するセンサデータストリームから選択可能な配信周期とする。受信ノードは、任意のセンサノードとそのセンサノードが提供する配信周期の 1 つをランダムに選択してセンサデータストリームを受信する。

提案手法との比較手法として、文献 [10] と同様にセンサ ID のみから求めたハッシュ値により担当中継ノードの決める Source 法、センサ ID と配信インデックスから求めたハッシュ値を用いる Time 法、センサ ID と配信インデックスから求めたハッシュ値を用いる Cycle 法を用いる。提案手法は Cycle-Time 法と表記する。

センサデータストリームとして配信す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 20ms 間隔で 5 分間センサノードから送出し、中継ノードを介して受信ノードへ配信することで計測を行う。

中継ノードにかかる負荷の公平性の指標として、公平性の評価に一般的に用いられる Jain's fairness index を用いる。中継ノードの負荷を  $L_i$  ( $i = 1 \sim N$ ) とすると、Fairness Index; FI は式 1 のとおりとなる。この評価では、中継ノードが 10 ノードなので  $N = 10$  となる。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と言える。

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

### 4.2.2 Fairness Index

各中継ノード計測期間中の平均 CPU 負荷より Fairness Index を求めた。平均 CPU 負荷は Linux における /proc/stat を参照し、計測時間と計測中に配信システムのプロセスに CPU が割り当てられた時間の比により求めている。

中継ノードの配置が FIX の場合の受信ノード数の変化に

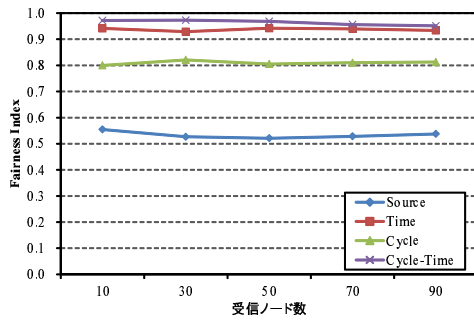


図 8 Fairness Index の変化 (FIX)

Fig. 8 Fairness Index by CPU load (FIX).

対する Fairness Index の変化を図 8 に示す。FIX での中継ノードの配置では、各中継ノードが受け持つハッシュ空間の大きさが均等であることから、中継ノードの配置による偏りの影響を排除した評価となる。受信ノード数の違いには関係なく Source, Cycle, Time, Cycle-Time の順で公平な状態となっている。Cycle 法, Time 法, Cycle-Time 法では、それぞれほぼ 0.8 を越えた値となっていることから、中継ノード間での負荷の偏りは小さく、公平な状態と言える。特に Time 法と Cycle-Time 法は 0.9 を越えていることから十分に公平な状態となっていると考えられる。Source 法においては、センサノード数が 10 ノードであることから、担当中継ノードとなる中継ノード数は最大でも 10 ノードであり、ハッシュ値の算出結果により実際には担当中継ノードとならず、中継処理にまったく携わらないノードがあったため、公平性が低く表れている。

中継ノードの配置が HASH の場合について図 9 に示す。この場合においても、受信ノード数が 10 ノードの場合を除いて各種法の順序は FIX の場合と同様となるが、Cycle 法, Time 法, Cycle-Time 法については FIX の場合より公平性が悪化しており、Cycle-Time 法のみが 0.78 ~ 0.79 と 0.8 に近い値となっている。これは中継ノードの配置が HASH では不均等な配置となることから、各中継ノードの担当領域の大小により生じたものと推測される。

中継ノードの配置として FIX と HASH の 2 種で評価を行ったが、それぞれ Time 法と Cycle-Time 法では同程度の公平性となったため、この 2 手法を抜き出し次節にて比較評価を示す。

### 4.3 Time 法と Cycle-Time 法の比較評価

負荷計測を行う配信環境には、4.2.1 項と同じ環境を利用し、ハッシュ空間上での中継ノードの配置には、FIX により割り当てる。計測に用いるセンサデータストリームも 4.2.1 節と同様に、0 パディングされた 1,024 バイトのダミーデータを 20ms 間隔で 5 分間配信するものとする。

センサノードは 1 ノードのみとし、1 ~ 3 の 3 種を配

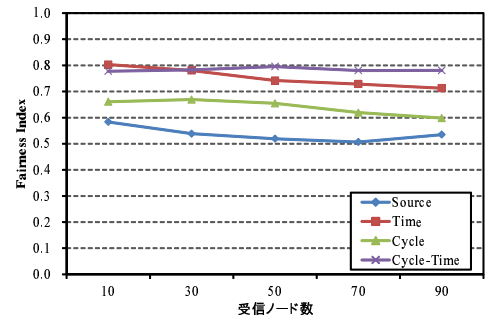


図 9 Fairness Index の変化 (HASH)

Fig. 9 Fairness Index by CPU load (HASH).

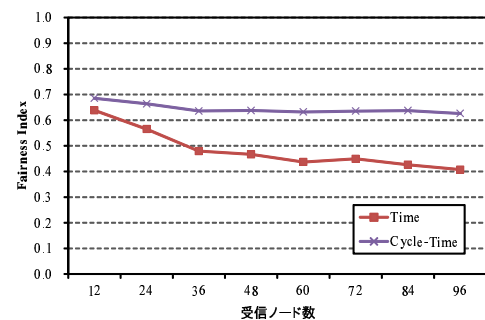


図 10 Time と Cycle-Time Fairness Index の変化

Fig. 10 Fairness Index of Time and Cycle-Time.

信可能な周期とする。受信ノードは、配信周期 1 のセンサデータストリームを受信する受信ノード、配信周期 2 を受信する受信ノード、配信周期 3 を受信する受信ノードを 1 組として、これらを 4 組、12 ノードずつ増やしつつ中継ノードの CPU 負荷を計測する。

この計測により得られた平均 CPU 負荷から求めた Fairness Index が図 10 となる。図から読み取れるように、Time 法は受信ノード数が増えるにしたがい公平性が悪化しているが、Cycle-Time 法では Time 法ほど悪化していない。Time 法, Cycle-Time 法での各中継ノード RELAY000 ~ 009 の平均 CPU 負荷の変化は、それぞれ図 11 と図 12 となる。それぞれ、最も負荷が高いノード、Time 法では RELAY000、Cycle-Time 法では RELAY005 の負荷を比較すると、受信ノード数が 96 の点で前者が 0.104 に対して後者では 0.066 となっている。これは、Time 法では負荷がかかっているノードが 5 ノードであるのに対し、Cycle-Time 法では 7 ノードとなっていることから、その分負荷が分散されていると言える。担当中継ノードの割当てに用いるハッシュ値が Time 法では巡回周期長の 6 個、Cycle-Time 法では各配信周期の逆比の和である 11 個となることから、Time 法の場合は中継ノード 10 ノードのうち、最大でも 6 ノード

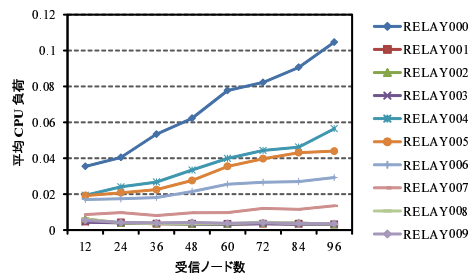


図 11 平均 CPU 負荷の変化 (Time)

Fig. 11 Average of CPU load (Time).

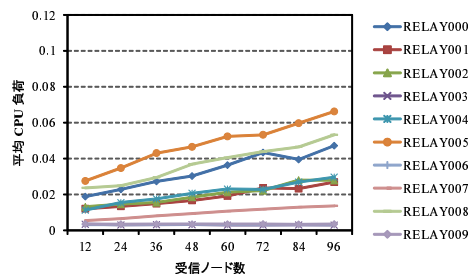


図 12 平均 CPU 負荷の変化 (Cycle-Time)

Fig. 12 Average of CPU load (Cycle-Time).

しか利用されない。中継ノードが十分に存在し、提供される配信周期の最小公倍数より多い場合では、Time 法より Cycle-Time 法のほうがより公平性が高く、広く負荷が分散されることから、1つのセンサノードが配信するセンサデータストリームに対してより多くの受信ノードを収容できると考えられる。

## 5 まとめ

本研究では、コンシステントハッシュ法を用いて複数の異なる周期のセンサデータストリーム配信を行う配信システムの PIAX による実装について述べ、その実装システムを PIAX テストベッドを用いた実機環境での挙動について評価を行った。その結果、提案手法では他の比較手法より公平に中継ノードを利用することができ、1つのセンサノードが発するセンサデータストリームをより多くの受信ノードで受信できることが確認できた。

今後の方向性として、現実装では未実装となっている、中継ノードの離脱による配信傷害への耐久性の確保とその評価や、中継ノードの増設によりスケラブルに配信能力を拡張できる配信経路構築維持手法の開発が考えられる。

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」による成果である。

## 参考文献

- [1] 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).
- [2] 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).
- [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] Le, T. A. and Nguyen, H.: Application-Aware Cost Function and Its Performance Evaluation over Scalable Video Conferencing Services on Heterogeneous Networks, *Proceedings of the 2012 IEEE Wireless Communications and Networking Conference (WCNC 2012)*, pp. 2185–2190 (2012).
- [5] Kawakami, T., Ishi, Y., Yoshihisa, T. and Teranishi, Y.: A Delivery Method considering Communication Loads for Sensor Data Stream with Different Collection Cycles, *Proceedings of The 28th ACM Symposium on Applied Computing (SAC 2013)*, pp. 611–618 (2013).
- [6] Kawakami, T., Ishi, Y., Yoshihisa, T. and Teranishi, Y.: A P2P Delivery Method for Sensor Data Stream Based on Load Estimation from Collection Cycles, *Proceedings of The 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).
- [7] 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).
- [8] 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).
- [9] 石 芳正, 川上朋也, 義久智樹, 寺西裕一: ホップ数を考慮した P2P 型センサデータストリーム配信システムの PIAX テストベッドを用いた評価, *情報処理学会マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO 2014) 論文集*, pp. 1421–1427 (2014).
- [10] Kawakami, T., Ishi, Y., Yoshihisa, T. and Teranishi, Y.: A Load Distribution Method Based on Distributed Hashing for P2P Sensor Data Stream Delivery System, *Proceedings of the 3rd IEEE International Workshop on Modeling and Verifying of Distributed Applications (MVDA 2014) in Conjunction with the 38th Annual International Computer Software and Applications Conference (COMPSAC 2014)*, pp. 716–721 (2014).
- [11] 石 芳正, 川上朋也, 義久智樹, 寺西裕一: コンシステントハッシュ法を用いた複数センサデータストリーム配信システムの一実装, *情報処理学会マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO 2015) 論文集*, pp. 1852–1856 (2015).
- [12] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F. and Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications, *SIGCOMM Comput. Commun. Rev.*, Vol. 31, No. 4, pp. 149–160 (2001).
- [13] 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).