

推薦論文

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実現と評価

石 芳正^{1,2,a)} 川上 朋也^{1,3,b)} 義久 智樹^{1,c)} 寺西 裕一^{1,4,d)}

受付日 2016年5月24日, 採録日 2016年11月1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させる 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, 同数の中継ノードでより多くのセンサデータストリームを配信しうることを確認した.

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 Multiple Sensor Data Streaming System with Consistent Hashing

YOSHIMASA ISHI^{1,2,a)} TOMOYA KAWAKAMI^{1,3,b)} TOMOKI YOSHIHISA^{1,c)}
YUUCHI TERANISHI^{1,4,d)}

Received: May 24, 2016, Accepted: November 1, 2016

Abstract: Our research team proposed some sensor data streaming methods based on peer-to-peer techniques for distributing communication loads when delivering sensor data streams with different data collection cycles. However, those techniques support delivering only a sensor data stream on a delivery network. To support multiple sensor data stream on a delivery network, we have proposed a new delivery system which is composed of 3 layers, sensor data sources, a relaying network, sensor data destinations. To assign a relay node with distributing communication loads in the relaying network, Consistent hashing is used. In this paper, we evaluate the behavior of the proposed system by using the PIAX testbed. From the evaluation result, we confirmed that proposed method provides better load distribution and it could be deliver more sensor data streams to sensor data destinations.

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 株式会社 PIAX
PIAX Inc., Osaka 530-0001, Japan

³ 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

⁴ 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

a) ishi@cmc.osaka-u.ac.jp

b) kawakami@is.naist.jp

c) yoshihisa@cmc.osaka-u.ac.jp

d) teranisi@nict.go.jp

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

本論文の内容は 2016 年 3 月の第 166 回マルチメディア通信と分散処理研究会で報告され, 同研究会主査より情報処理学会論文誌ジャーナルへの掲載が推薦された論文である.

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・高精度にもかか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したりすることにより、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良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へ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。

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ほか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ため、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す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]。

これらの成果をふまえて、複数のセンサデバイスからのセンサデータストリームをそれぞれ異なる配信周期を要求する複数の配信先へと配信する多対多配信を実現する手法を提案している [10]。この手法では、センサデータストリームの配信に際し、1対多のセンサデータストリーム配信を複数収容する中継ネットワークを設け、その中で各センサが出力するセンサデータストリームを必要な配信周期のセンサデータストリームとして再編しつつ配信を行う。この中継ネットワークは複数のノードから構成され、センサデータストリーム配信サービスとしてセンサデータの提供者が配信サービスにセンサデータストリームを提供し、利用者が必要な配信周期のセンサデータストリームを配信サービスより受信することを想定している。中継ネットワークを介したセンサデータストリーム配信に際し、中継ネットワークを構成するノード間で負荷を分散するため、センサデータストリームを中継するノードをコンシステントハッシュ法により自律的に選出する手法が提案の主眼となっているが、その評価はシミュレーションによるものにとどまっており、実機上での挙動は未知数となっている。

そこで本研究では、先行研究において提案した手法を

Peer-to-Peer エージェントフレームワーク PIAX [11]*1 を用いて実機評価システムを実装し、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 が運用している 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、提案手法が想定す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。

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

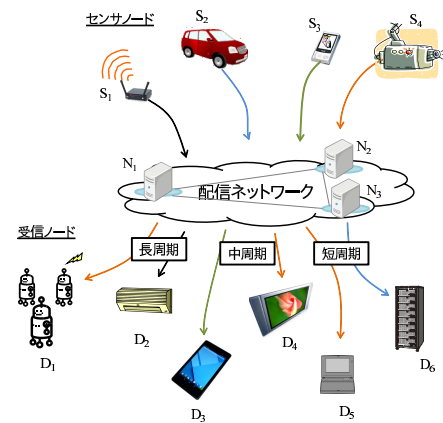


図 1 配信モデル

Fig. 1 Delivery model.

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)	○	○	○	○	○	○	○	...

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、すなわちセンサノード 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。本研究では、このように周期的に送られてくるセンサデータによるセンサデータストリームを対象としており、非周期的なセンサデータを扱うセンサ、たとえば観測値を内部保存しておき、不定期にバルク送信するセンサや、事前設定されたトリガによりイベント通知としてセンサデータを送信するセンサについては取り扱わないものとする。

また、本研究では、配信ネットワーク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た中継ノードにより構成され、センサデータストリーム配信サービスとして、ある種のクラウドサービスの形で提供される用途を想定している。配信ネットワークは、センサノードよりストリームとして任意の時間間隔で周期的に送られてくるセンサデータを配信周期に応じて、逐次適切な受信ノードに再送信する機能を担う。この再送信においてはサーバリソースを有効活用するため、中継ノードの間での負荷の偏りを抑制することが望ましい。このため、次節で配信周期を用いて負荷を分散する配信ネットワークの構成手法について述べる。

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

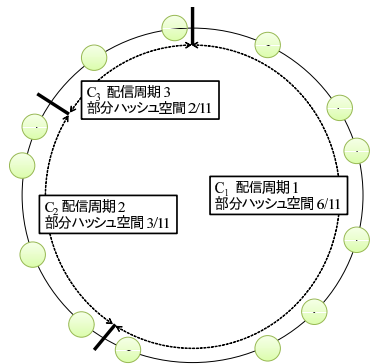


図 2 配信周期によるハッシュ空間の分割

Fig. 2 Dividing hash space by the delivery cycles.

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固有情報からハッシュ値を求め、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 $C_i = i$ ($i = 1, 2, 3$) の場合では、 $1/C_1 : 1/C_2 : 1/C_3 = 1/1 : 1/2 : 1/3 = 6 : 3 : 2$ 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つな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を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。配信パターンは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

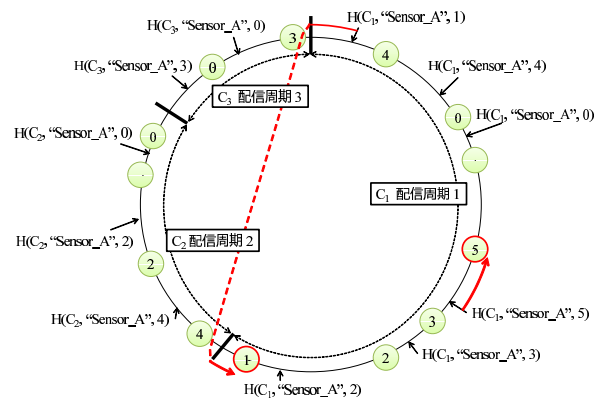


図 3 担当中継ノードの割当て

Fig. 3 Assignment of the relay nodes.

り当てればよいこととなる。本提案では、この割当てをコンシステントハッシュ法により決定する。

まず、ハッシュ関数 H に、部分ハッシュ空間の範囲とセンサ ID、配信インデックスを与えてハッシュ値を求める。ハッシュ関数 H は、与えられた部分ハッシュ空間内で一様に分布するハッシュ値を返すものとする。そして、部分ハッシュ空間内において、得られた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 $H(\text{“Sensor_A”}, 5)$ 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 $H(\text{“Sensor_A”}, 1)$ 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うこととなる。

センサノードは、センサ ID と配信インデックスより求められる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こと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には、分散環境下におけるエンティティ探索機能・ノード間メッセージング機能を有すること、実機での検証・評価環境としてPIAXテストベッドが利用できること、提案手法のデモシステム [13] が存在することから、このデモシステムと同様にPIAXを用いて実装することとした。

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オーバー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

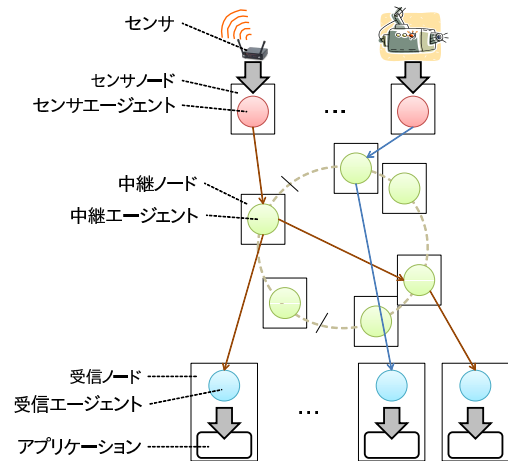


図4 エージェントを用いた配信モデル

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

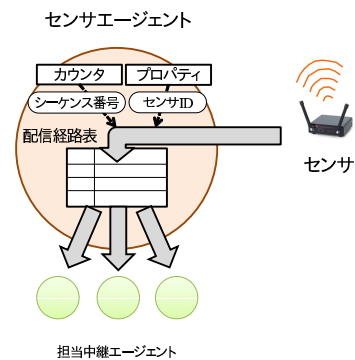


図5 センサエージェント

Fig. 5 Structure of the SensorAgent.

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。

1つのセンサエージェント(図5)は、1つのセンサに対応し、センサより定期的を取得したセンサデータを中継エージェントに送出する機能を持つ。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。

中継エージェント(図6)は、センサエージェントから送られたセンサデータを受信エージェントや他の中継エージェントに転送する中継機能を持つ。また、センサエージェントのセンサデータストリーム配信要求に応じて配信経路表を生成する機能、および受信エージェントからの要求に応じて、対応する配信経路表を提供する機能も持つ。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

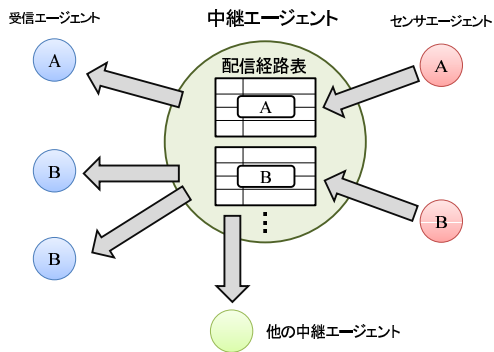


図 6 中継エージェント

Fig. 6 Structure of the RelayAgent.

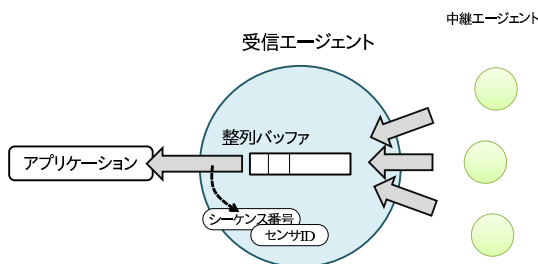


図 7 受信エージェント

Fig. 7 Structure of the ReceiverAgent.

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。

受信エージェント (図 7) は、中継エージェントから送られたセンサデータをセンサデータストリームとして再構成し、同ノード上で動作しているアプリケーションに受け渡す機能を持つ。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ため、受信エージェント内に整列バッファを設けている。

以上 3 種のエージェントに加えて、PIAX 上でハッシュ値によるエージェント探索を実現するため、PIAX が標準でサポートしている SkipGraph [14] の機能を用いて、ハッシュ値による MaxLessThan 探索を実現するオーバーレイネットワークを実装した。これにより、探索キーとして指定されたハッシュ値を超えないハッシュ値をオーバーレイネットワークに登録しているエージェントが探索可能となる。

3.3 センサデータストリーム配信システムの動作

3.3.1 エージェントの初期設定と配信準備

本研究では、センサエージェントと受信エージェントは事前に中継エージェントの情報を持たず、必要に応じて PIAX のオーバーレイ探索機能により中継エージェントを発見することを想定している。まず、中継エージェントは起動時に自身のハッシュ値を 1 つ決定し、オーバーレイネットワークに登録する。これにより、この中継エージェントはセンサエージェントと受信エージェントより探索可能と

なる。

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は、まず任意の中継エージェントを 1 つ発見する。これには、センサエージェントがランダムなハッシュ値を生成し、そのハッシュ値に対応する中継エージェントをオーバーレイネットワーク上で探索することで実現できる。中継エージェントを発見したセンサエージェントは、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中継エージェントに出し、配信経路を構築させる。そして、その応答としてセンサエージェント用の配信経路表を取得する。

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、要求された配信周期に応じた配信経路を構築し、センサエージェント用の配信経路表を応答する。構築された配信経路の情報は、各中継エージェントから参照可能な共有ストレージ、たとえば分散ハッシュテーブルなどにセンサ ID と対応づけて格納する。また、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情報を取得し、要求された配信周期のセンサデータストリームが配信可能か判定する。配信可能な周期であった場合は、配信経路情報より受信エージェント用の配信経路表を作製して受信エージェントに応答する。中継エージェントにおける配信経路の構築方法や各種配信経路表の作製方法は、3.3.2 項で後述する。

受信エージェントがあるセンサデータストリームを受信しようとする際は、まず任意の中継エージェントを 1 つ発見する。これには、センサエージェントが中継エージェントの発見に用いた手法と同手法を用いる。中継エージェントを発見した受信エージェントは、中継エージェントに受信しようとするセンサデータストリームのセンサ ID と配信周期を通知し、受信エージェント用の配信経路表を要求する。配信周期は、対応するセンサエージェントにより提供される周期の中から選択して指定する。要求の応答として得られる受信エージェント用の配信経路表には、配信インデックスとその配信インデックスに対応する中継エージェントの対応が格納されている。受信エージェントは、配信経路表内の各中継エージェントに対し、センサ ID と配信インデックスに対応するセンサデータを自身に配信するよう要求を出す。配信要求を受けた中継エージェントは、受信エージェントの宛先情報を要求されたセンサ ID と配信インデックスに対応づけ、自身が保持する配信経路表に追加する。

3.3.2 配信経路の構築

中継エージェントがセンサエージェントからの配信要求を受けた際、図 3 で示した例の場合では、まず表 2 で示す配信経路表が構築される。この配信経路表は、配信周

表 2 配信経路表の例

Table 2 An example of routing table.

Cycle	Index	Hash Value	Relay Agent
1	0	35...41	1
	1	08...A2	6
	2	83...2C	5
	3	60...9D	4
	4	28...F9	0
2	5	52...87	3
	0	DA...90	10
	2	BF...9B	8
3	4	A3...22	7
	0	F0...74	11
	3	E8...09	12

表 3 中継エージェント間の中継経路情報の例

Table 3 An example of relay table for a relay agent.

Index	Hash Value	Relay Agent
0	DA...90	10
	35...41	1

期 (Cycle) ごとの配信インデックス (Index) とそれに対応するハッシュ値 (Hash Value), 中継エージェント情報 (Relay Agent) を含んでいる. ハッシュ値は 2.2 節で述べた手法に従い, センサ ID と配信周期, 配信インデックスより算出される. また, 表中ではハッシュ値の中間桁を省略して表記している. ハッシュ値が求まると次に, オーバレイネットワークを介してハッシュ値を担当する中継エージェントを発見し, その宛先情報を中継エージェント情報とする. 表中の中継エージェント情報は, 例示のため図 3 中のハッシュ空間の起点を基準として時計回りで採番したものを示しているが, 実際は PIAX のピア ID やエージェント ID など, エージェント間通信に用いる宛先情報が格納されることとなる. ここで生成された配信経路は, センサ ID をキーとして共有ストレージに格納される.

生成された配信経路において, 同一配信インデックスの中継エージェントについては, それら中継エージェント間でのセンサデータ中継処理が生じる. このため, 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. 表 2 の場合, 配信インデックス 0 を担当する中継エージェントは配信周期 1, 2, 3 それぞれに存在するが, 提案手法では最長周期の中継ノードが他の配信周期の中継ノードに中継することとなっている. このため, 中継エージェント 11 に対し, 配信インデックス 0 のセンサデータを中継エージェント 10 と 1 に転送するよう表 3 に示す配信経路情報を与える.

センサエージェント用の配信経路表は, 表 2 より, 配信周期が大きい経路を優先して各配信周期の経路をマージす

表 4 センサエージェントの配信経路表の例

Table 4 An example of routing table for a sensor agent.

Index	Hash Value	Relay Agent
0	FA...74	11
1	07...A2	6
2	BF...9B	8
3	E8...09	12
4	A3...22	7
5	5A...90	3

表 5 受信エージェントの配信経路表の例

Table 5 An example of routing table for a receiver agent.

Index	Hash Value	Relay Agent
0	DA...90	10
2	BF...9B	8
4	A3...22	7

ることで作製する. まず, 配信周期 1 の配信インデックス 0~5 を基盤として, 配信周期 2 の配信インデックス 0, 2, 4 をマージし, 続いて配信周期 3 より配信インデックス 0, 3 をマージする. これにより, センサエージェントに回答する配信経路表, 表 4 が完成する.

受信エージェントより受信エージェント用の配信経路表を要求された中継エージェントでは, 共有ストレージよりセンサ ID をキーとして配信経路表を取得し, 要求された配信周期の経路情報を抜き出して受信エージェントに回答する. 表 2 の例では, 配信周期 2 が要求された場合, 表内の配信周期 2 の部分 (表 5) を回答として返すこととなる. この表を受けた受信エージェントは, 表中の中継エージェントそれぞれに対し, センサ ID と配信インデックスを伝え, 対応するセンサデータを自身に配信するよう要求することとなる.

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. この実装の省略により, 各部分ハッシュ空間の末尾部に位置する中継ノードが担当するハッシュ空間が変わり, 負荷状況が変わることが考えられる. 本来の手法においては, 部分ハッシュ空間の末尾部の中継ノードは, その部分ハッシュ空間の末尾部に加え, 循環処理により同空間の先頭部を担当するが, この先頭部に代わり, 隣接する部分ハッシュ空間の先頭部を担当することとなる. たとえば, 図 3 では部分ハッシュ空間 C_1 で配信インデックス 1 を担当するノードは, 時計回りで隣接する中継ノード (部分ハッシュ空間 C_2 の配信インデックス 4 担当中継ノード) までの区間を担当することとなる. この担当空間の変更は, 各部分ハッシュ空間が空間の先頭部を反時計回りに隣接する各部分ハッシュ空間に委譲することに相当する. 中継ノードが空

間中に一様に分布している状況では、委譲により担当から外れる範囲と加わる範囲がほぼ同等となると期待できることから、部分ハッシュ空間の末尾部に位置する中継ノードに加わる負荷に影響は与えないと考えられる。

3.3.3 センサデータストリーム配信

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、シーケンス番号と巡回周期長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する。メタデータが付与されたセンサデータは、その中継エージェントに送信される。

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。

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。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などのメタデータが削除される。そして、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渡す。

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（NICT）が提供しているPIAXテストベッドを用いて、シミュレーションでは計測困難な処理負荷の検証を行う。

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^{*3}のLinux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、各仮想マシン間は1GbpsのLANおよびJGN-X L2サービスによる拠点間ネットワークを介して接続されている。ユーザには、これらの仮想マシンの中から複数台が割り当てられ、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。また、割り当てられた仮想マシンは、ユーザが占有して利用する形態となっており、別ユーザのプログラムによる負荷の影響を受けずに自身のプログラムを評価することができる。

4.2 評価環境と評価指標

評価環境は、トンネルや橋梁、ダムといったインフラ建造物の状態監視システムを想定し、複数の監視対象より送られるセンサデータストリームを、管理会社や研究機関、防災当局といった複数の機関がそれぞれ必要なセンサデータストリームを必要な配信周期で受信する状況とする。

負荷計測を行う配信環境として、センサノードを10ノード、中継ノードを10ノード、受信ノードを100ノードとする。中継ノードは集約されて設置されることを想定しているため、神奈川拠点の10VMに割り当て、各VM状で1ノードずつ動作させる。センサノードは石川拠点の9VMに割り当て、1VM上で1~2ノードを動作させる。受信ノードは京都拠点の10VMに割り当て、1VM上で10受信ノードを動作させる。

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。

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、数十Hzの周期で多点の状態観測を行う建造物の状態監視システムを想定した20ms間隔（50Hz）で生成される1,024バイトのダミーデータを用いる。ダミーデータの内訳は、タイムスタンプやセンサIDなどのメタデータ部に64バイト、観測点に敷設された加速度センサや歪みセンサによる観測値を含む実データ部に32バイト×30観測点と仮定している。このダミーデータを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。

ハッシュ空間は160bit長の空間とし、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、その境界点のハッシュ値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SHA-1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。

提案手法との比較対象として、文献[10]と同様にハッ

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

^{*3} 2015年5月時点。

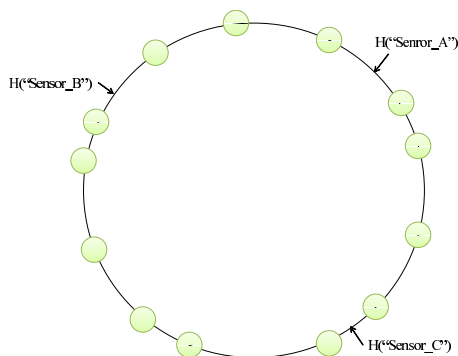


図 8 Source 法による担当中継ノードの割当て

Fig. 8 Assignment of the relay nodes by the Source method.

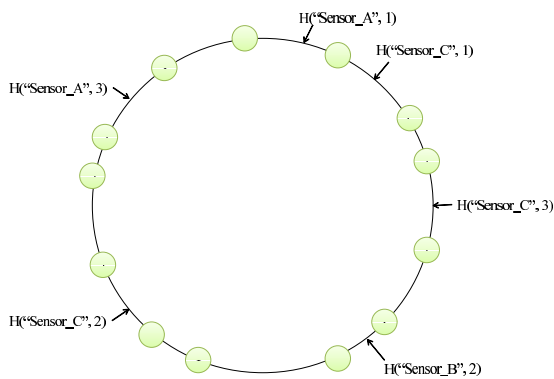


図 9 Cycle 法による担当中継ノードの割当て

Fig. 9 Assignment of the relay nodes by the Cycle method.

シユ空間を分割しない Source 法, Cycle 法, Time 法を用いる. 提案手法は Cycle-Time 法と表記する. Cycle-Time 法では, 担当中継ノードの割当ての際に部分ハッシュ空間の範囲とセンサ ID, 配信インデックスからハッシュ値を求め, 部分ハッシュ空間内の中継ノードを割り当てていた. これに対し, Source 法はセンサ ID のみをハッシュ関数に与えてハッシュ値を得る. このため, センサ 1 つにつき 1 つの中継ノードが割り当てられ, あるセンサのセンサデータはその中継ノードが配信周期に応じて受信ノードに中継することとなる (図 8). Cycle 法は, センサ ID と配信周期よりハッシュ値を求める. これにより Cycle 法では「センサ A の配信周期 1」「センサ A の配信周期 3」「センサ B の配信周期 2」というように, センサ ID と配信周期の組ごと中継ノードが割り当てられ, それぞれの中継ノードが担当するセンサの担当する配信周期のセンサデータを中継することとなる (図 9). 受信ノードは, 自身が受信しようとしているセンサと配信周期を担当する中継ノードよりセンサデータを受け取る. Time 法は, センサ ID と配信インデックスよりハッシュ値を求める. Time 法では「センサ A の配信インデックス 0」「センサ A の配信インデックス 1」「センサ A の配信インデックス 2」「センサ B の配信インデックス 0」「センサ B の配信インデックス 1」というように, センサ ID と配信インデックスの組ごとに中継ノードが割り当てられる (図 10). 受信ノードは, 自身が受信

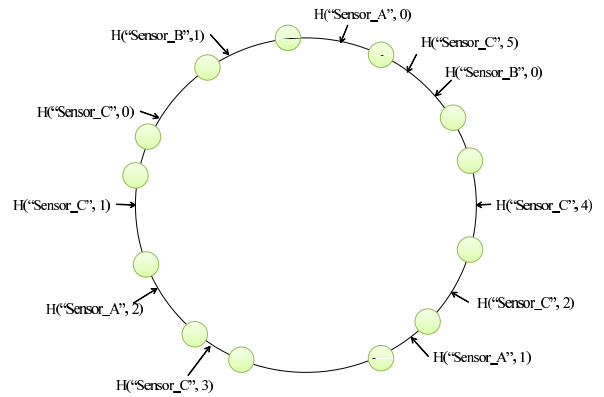


図 10 Time 法による担当中継ノードの割当て

Fig. 10 Assignment of the relay nodes by the Time method.

しようとしているセンサと配信周期に対応するセンサデータを担当している中継よりセンサデータを受け取る.

中継ノードにかかる負荷の分散度合いの指標として, 公平性の評価に一般的に用いられる Jain's fairness index (以下 Fairness Index と表記する) [15] 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いえる.

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

4.3 Fairness Index による評価

Fairness Index を算出する元となる計測値として, 中継ノードが送受信するセンサデータ数と平均 CPU 負荷を用いる. センサデータ数は, 評価機材の状態に影響を受けない値で, シミュレーションにより求められる値と同等となる. 平均 CPU 負荷は, Linux における /proc/stat と /proc/[pid]/stat を参照し, 計測期間中の総 CPU 時間に対して, 配信システムのプロセスがスケジューリングされた CPU 時間の比を求めることで配信システムにより生じる CPU 負荷の平均値を求めた. 評価条件においては, 各中継ノードは最長でも 1.2 秒周期で同じ中継処理を行うことから, 計測期間の 5 分間では十分な平均が得られる. この 1.2 秒の周期は, 配信するセンサデータストリームとして, 配信周期 1~6 のすべてが選択された場合の巡回周期が元となっており, 配信周期の最小公倍数 60 が評価条件下で最長の巡回周期となる. センサデータの送出される間隔が 20 ms 間隔であることから, 最長で 1.2 秒周期となる.

中継ノードの配置が FIX の場合の受信ノード数の変化に対する Fairness Index の変化を図 11 と図 12 に示す. 前者は各中継ノードが送受信したセンサデータ数より求めた Fairness Index, 後者は各中継ノードの平均 CPU 負荷より求めた Fairness Index の変化を表す. 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

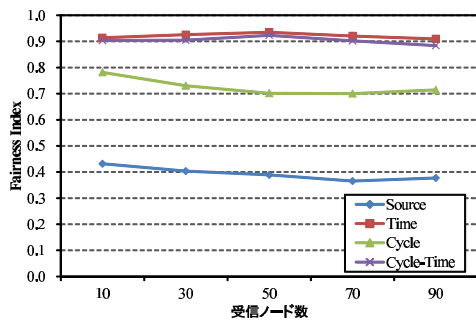


図 11 センサデータ数による Fairness Index の変化 (FIX)
 Fig. 11 Fairness Index by the number of sensor data (FIX).

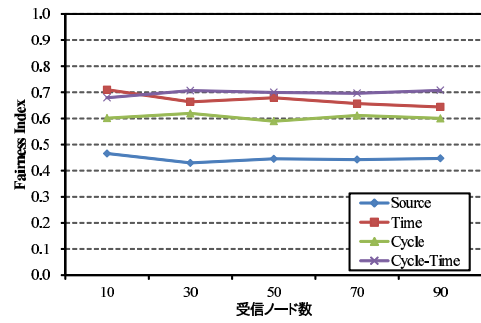


図 13 センサデータ数による Fairness Index の変化 (HASH)
 Fig. 13 Fairness Index by the number of sensor data (HASH).

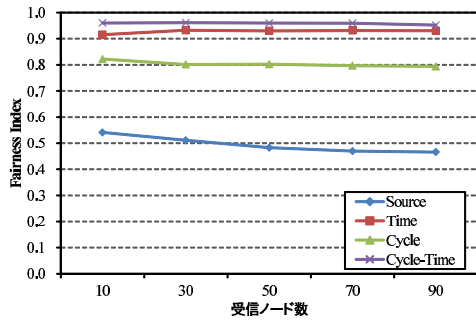


図 12 CPU 負荷による Fairness Index の変化 (FIX)
 Fig. 12 Fairness Index by CPU load (FIX).

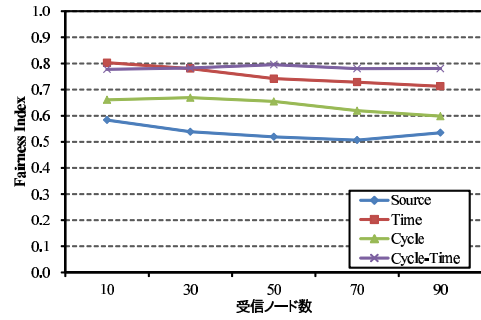


図 14 CPU 負荷による Fairness Index の変化 (HASH)
 Fig. 14 Fairness Index by CPU load (HASH).

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。

センサデータ数より求めた図 11 では、Time 法と Cycle-Time 法がほぼ同値で受信ノード数の大小に関係なくすべて 0.9 以上となっており、他の 2 手法と比較して中継ノード間の負荷は公平な状態となっていることが分かる。公平性が最も低い 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、公平性が低くなっている。

平均 CPU 負荷より求めた図 12 では、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 を超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、公平な状態といえる。特に Time 法と Cycle-Time 法は 0.9 を超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。グラフの傾向はセンサデータ数から求めた場合と相似しており、平均 CPU 負荷より求めた場合では、センサデータ数による Fairness Index より 0.1 程度大きな値となっているが、これはセンサデータ数に依存しない恒常的に計測される CPU 負荷による影響と考えられる。

中継ノードの配置が HASH の場合について図 13 と図 14 に示す。FIX の場合と同様に、前者は各中継ノードが送受信したセンサデータ数より求めた Fairness Index、後者は

各中継ノードの平均 CPU 負荷より求めた Fairness Index の変化を表しており、FIX での結果とほぼ同様に、ほぼ相似であるが平均 CPU 負荷より求めた結果が 0.1 程度大きな値が得られている。

平均 CPU 負荷より求めた図 14 では、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。

以上のとおり、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で比較評価を示す。

4.4 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 節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 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 20 ms 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

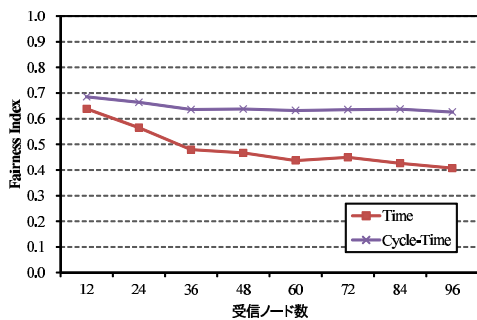


図 15 Time と Cycle-Time Fairness Index の変化

Fig. 15 Fairness Index of the Time method and the Cycle-Time method.

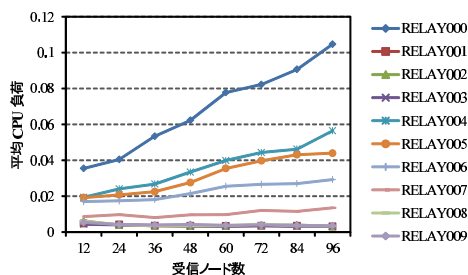


図 16 平均 CPU 負荷の変化 (Time)

Fig. 16 Average of CPU load (Time).

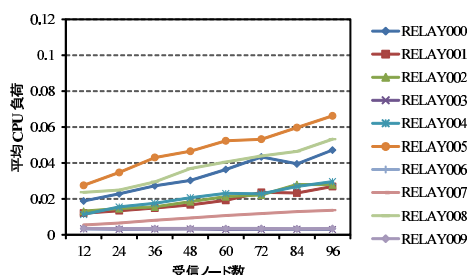


図 17 平均 CPU 負荷の変化 (Cycle-Time)

Fig. 17 Average of CPU load (Cycle-Time).

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 15 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従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 16 と図 17 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いえる。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巡回周期長より多い場合には，Time 法より Cycle-Time 法のほうが広く負荷が分散され公平性が高くなることから，同数の中継ノードでより多くのセンサデータストリームを配信できると考えられる。

その一方で，Cycle-Time 法では異なる配信周期のセンサデータストリームを配信するために，同じセンサデータを部分ハッシュ空間が異なる複数の中継ノードで扱うこととなる。中継ノード間でセンサデータを受け渡すことが必要であることから，Time 法と比較してその中継処理の分，負荷が増加することとなる。1 センサノードからのセンサデータストリームに焦点を当てると，そのセンサデータストリームを受信する受信ノードが存在しない場合や 1 ノード程度の場合では，中継処理による負荷が顕著に表れることとなる。特に短周期で受信する受信ノードは存在するが，長周期で受信する受信ノードが存在しない場合では，Time 法においては中継ノードが受け取ったセンサデータはすべて短周期で受信する受信ノードに受け渡され無駄は生じないが，Cycle-Time 法では，長周期のセンサデータストリームを扱う中継ノードが受け取ったセンサデータは，そのセンサデータを受信する受信ノードが存在しないため中継ノード間で中継されたものの無駄なものになってしまう。しかしながら，こういった無駄な中継処理は，各配信周期において，その周期で受信しようとする受信ノードが現れた時点で中継処理を開始する実装とすることにより削減することができる。また，中継ノード間の中継処理による負荷は，配信システム全体で見ると，複数のセンサノードからのセンサデータストリームを同時に配信していることから，それらの負荷に埋没するものと考えられる。総じて，Cycle-Time 法はセンサノードに対して，受信ノード数が十分に存在する場合において有効に機能する手法といえる。

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，中継ノードが同数である場合，より多くのセンサデータストリーム配信を収容しう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障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」, および文部科学省科学研究費補助金・基盤研究 (B) (15H02702), 挑戦的萌芽研究 (26540045) の研究助成によるものである。ここに記して謝意を表す。

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, *Proc. 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, *Proc. 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, *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).
- [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: 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, *情報処理学会論文誌*, Vol.57, No.2, pp.583-596 (2016).
- [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, *Proc. 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] 吉田 幹, 奥田 剛, 寺西裕一, 春本 要, 下條真司: マルチオーバ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).

- [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 (online), DOI: 10.1145/964723.383071 (2001).
- [13] 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).
- [14] Aspnes, J. and Shah, G.: Skip Graphs, *ACM Trans. Algorithms*, Vol.3, No.4, p.37 (2007).
- [15] 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).

推薦文

センサネットワークにおいて複数の配信先が異なる周期でデータを要求するような状況を想定し, 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の中継ノードを選択することで負荷分散を実現する手法を提案しています。提案手法を実機環境で評価しており, 既存手法に比べて公平にデータを中継できていることを示しています。さらなる場合分けによる検討を行うことなどにより, 研究としてある程度完成に近づくものと思われるため, 推薦論文としてふさわしいと判断いたしました。

(マルチメディア通信と分散処理研究会主査 重野 寛)



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 年より同大学サイバーメディアセンター特任研究員, および株式会社 PIAX 技術顧問。2016 年 3 月大阪大学を退職。同年 10 月より大阪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。



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。