

移動センシング環境における ストリームデータ分割配送手法

村山 知 弥^{†1} 横田 裕 介^{†2} 大久保 英 嗣^{†2}

人や物とともにセンサノードを移動させながらセンシングを行う移動センシング環境では、バイタルデータの高精度な観測が可能になる。しかし、このような環境でバイタルデータのようなストリームデータの観測を行う場合、ノードの移動に伴うデータ損失が問題となる。本稿では、周囲のノードにデータを多重化配送することでデータ損失を防ぐストリームデータ分割配送手法について述べる。本手法により、より多くの観測データを基地局へ届けることが可能となる。

A Data Stream Delivery Mechanism for Mobile Sensing Environment

TOMOYA MURAYAMA,^{†1} YUSUKE YOKOTA^{†2}
and EIJI OKUBO^{†2}

Mobile sensor nodes along with moving humans or objects can retrieve high quality stream sensor data such as vital data. We call it mobile sensing environment. Loss of stream data that occurs during movement of nodes is an issue in mobile sensing environment. In this paper, we propose a data stream delivery mechanism to avoid such data loss based on multiplexed data delivery to adjacent nodes. The proposed mechanism can help to send more stream sensor data to a base station.

^{†1} 立命館大学大学院理工学研究科

Graduate School of Science and Engineering, Ritsumeikan University

^{†2} 立命館大学情報理工学部

College of Information Science and Engineering, Ritsumeikan University

1. はじめに

近年、半導体デバイスの小型化・低価格化に伴い、無線通信機能とセンシング機能を持つ小型のデバイスであるセンサノードの開発が進んでいる。センサノードを多数配置し、センサネットワークを構築することにより、人や物、環境の状態を正確に観測することが可能となる。

我々は、センサノードが人と共に移動するセンサネットワーク環境として、移動センシング環境に関する研究を行っている。センサノードが移動することで、新たなノードを追加することなく、比較的少数のノードで広範囲にわたる観測を行うことが可能となる。また、観測対象と共に移動しながら観測を行うことで、近距離での高精度な観測が可能となる。

これらの特徴から、移動センシング環境のシナリオ例として、被災地におけるバイタルデータの観測が考えられる。バイタルデータのような観測データは、高頻度で取得されるストリームデータであり、継続的に届けられるデータに対する高速な処理が必要とされている。ストリームデータを高速に処理するシステムとしてストリームデータ処理システムが挙げられるが、従来これらのシステムでは、観測データの保持よりも応答時間の短縮が重視されてきた。しかし、移動センシング環境においてこれらの仕組みをそのまま用いた場合、ノードがネットワークから離脱した際、観測データが基地局に届く前に失われるという問題がある。

本研究では、周囲のセンサノードにデータを分配することでデータ損失を防ぐ、ストリームデータ分割配送手法を提案する。本手法では、センサノードの移動によるネットワークからの離脱をセンサノード側で自律的に検知する。また、データ損失の可能性がある場合には、周囲のセンサノードに対してデータの分割配送を行うことで、移動ノードを利用したデータ転送を試みる。本手法により、ネットワーク離脱時および輻輳発生時には観測データをセンサノード上に保持するため、より多くのデータを基地局に届けることが可能となる。

以下、2章で移動センシング環境と既存研究の課題について述べる。次に、3章で課題を解決するためのストリームデータ分割配送手法について述べる。最後に、4章でストリームデータ分割配送手法の評価について述べた後、5章で本論文をまとめる。

2. 移動センシング環境における課題

本章では、本研究の想定環境である、移動センシング環境について述べる。また、移動センシング環境において、ストリームデータを扱う場合の課題について述べる。

2.1 移動センシング環境

現在のセンサネットワークにおいて、センサノードは固定配置が主流となっている。固定配置によるセンシング（固定センシング）は、生態系のモニタリングやオブジェクトトラッキングなど、さまざまな環境下で行われている¹⁾。しかし、固定配置ではなく、人や物と共にセンサノードを移動させたセンシング（移動センシング）を行うことで、観測ノード数を増やすことなく広範囲にわたる観測が可能となる。この移動センシングが行われる観測環境を、移動センシング環境と呼ぶ。

移動センシングの主な利点として、固定配置の場合に問題となっていたセンサノードの配置場所、およびセンサネットワークの拡張性の欠如などに伴う配置規模の検討が不要になることが挙げられる。また、固定センシングと比べると、観測対象との距離が近い状態での観測が可能であるため、より正確なデータの観測が可能となる。

本研究では、移動センシングによって広範囲から集められたデータが、安定した通信が可能な基地局間のネットワークで集約された後、利用者に対して提供される環境を想定している。具体的には、被災地に ZigBee のような無線による低速回線ネットワークが構築され、それらの低速回線ネットワークがインターネットのような高速回線ネットワークで接続されるような環境である（図 1 参照）。このような環境では、高速なネットワークで負荷の大きな処理を行うことによって、システム全体の処理の効率化が可能となる。そのため、サービスの質を向上させるためには、できるだけ多くのデータを低速なネットワークから高速なネットワークに届けることが重要となる。

本研究では、典型的なシナリオとして「体温が一定値を超えた場合、その後の心電図デー

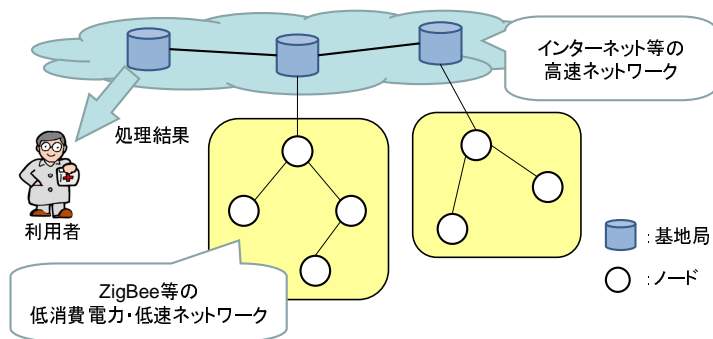


図 1 想定環境

タをリアルタイムに基地局へ送信する」といったクエリが実行されることを想定している。また、ここでの観測対象者は、被災地における避難所などのように周囲にノードが比較的多く存在する場所を移動しながらサービスを受けるものとする。ここでは、センサノードとして健康状態を観測するためにすべての人が持ち歩くことが可能な端末、あるいは、人体に装着してバイタルデータを取得できるセンサデバイスを想定している。一般的に、こういったセンサノードは、観測すべき時間に対して十分な電力資源を持ち合わせているもの、あるいはユーザによる充電が可能であるものと考えられるため、電力資源の枯渇によるデータ損失は考えないものとする。

2.2 ストリームデータ処理システムの課題

本研究で扱うバイタルデータのような観測データは、高頻度で取得されるストリームデータであり、継続的に到着するデータに対する高速な処理が必要とされる。高速なネットワークを前提とした環境においては、ストリームデータを効率的に処理できるシステムとして、DSMS (Data Stream Management System)²⁾ が提案されてきている。ここでは、代表的な DSMS について述べる。

STREAM³⁾ STREAM では、ネットワーク負荷のモニタリングやセンサネットワークによる観測など様々なアプリケーションを想定し、CQL⁴⁾ と呼ばれる問い合わせ言語を用いたストリームデータ処理を提案している。

STREAM では、永続的に入力されるストリームデータを処理するために、Sliding Window を用いたデータの切り取りを行う。STREAM に登録されたクエリは、図 2 に示す Operator に分割される。また、Queue を流れるストリームデータは、Sliding Window によって切り取られ、一時的に Synopses に格納される。それぞれの Operator は、Synopses のデータを共有しながら処理を行うことで、永続的に入力されるストリームデータを効率的に処理することが可能となる。

Aurora⁵⁾ Aurora では、インターネットに加えてセンサネットワークのような資源の限られた環境でのモニタリングを想定し、QoS 情報を用いてシステムの負荷を低減する手法を提案している。

Aurora では、ストリームデータの発生頻度の変化によってシステムの負荷が過剰になった場合、Load Shedding と呼ばれる入力データの削減処理を行うことで負荷を軽減する。この Load Shedding は、アプリケーションごとに指定された QoS 情報を基に行われるため、リアルタイム性が求められるストリームデータ処理を効率的に行うことが可能となる。

このように、既存の DSMS では、高速なネットワークにおけるストリームデータ処理の応答性に重点を置いているため、低速なネットワークでのデータ損失による信頼性の低下を考慮していない。そのため、ノードの移動によりネットワークの状況が動的に変化する移動センシング環境においては、低速なネットワーク内で多くのデータが失われ、高速なネットワークにおいてサービスの質を維持できない可能性が高い。

3. ストリームデータ分割配送手法

本章では、2章で述べた課題を解決するストリームデータ分割配送手法について述べる。以下、3.1節で本手法の概要について述べた後、3.2節で本手法で必要となる離脱検知機能について述べ、3.3節でストリームデータの分割配送機能について述べる。また、3.4節で帯域予約アルゴリズムについて述べる。

3.1 概要

2章で述べたように、移動センシング環境においては、全てのノードが基地局と通信可能であるとは限らない。センサノードが通信不可能な範囲に移動した場合や、通信速度を超える頻度でストリームデータが発生して輻輳が生じた場合、基地局と通信することができない分断されたネットワークが形成される。分断されたネットワークにおいて生成されたストリームデータは、基地局に届くことなく失われてしまう。これらのデータを周囲のノードに転送し、移動ノードを利用して基地局に送信することで、より多くのデータを基地局に到達させることが本研究の目的である。

次に、ノードでのストリームデータ処理の流れについて述べる。各ノードは、与えられたクエリに従って観測処理を行う。ネットワークから離脱した場合には、その後の観測データ

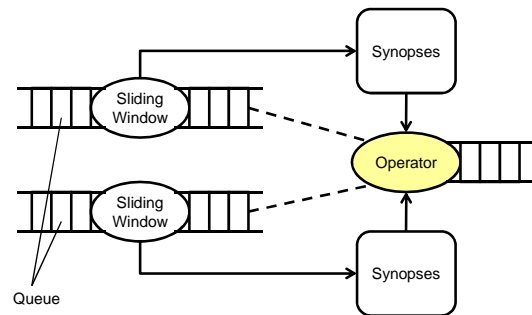


図2 STREAMの構造

は周囲のノードへ分割配送される。以下、それぞれの動作の詳細について述べる(図3参照)。

(1) 観測処理

クエリで指定されたセンサの値を定期的を取得し、ストリームデータを基地局に送信する必要が生じた場合、ノード上のストレージに格納してから送信を試みる。

(2) 離脱検知処理

送信先から受信確認(ACK)が返った場合は、送信したデータを自身のストレージから消去する。ACKが返らない場合は、ネットワークからの離脱と判断し、データを保持する。また、基地局から定期的に送られるパケットが一定時間届かない場合にも、ネットワークからの離脱と判断する。

(3) 分割配送処理

ノード上で保持しているデータが一定量を超えた場合、データ損失の可能性があるので、分割配送要求を送信して周囲のノードのストレージの状態を調べる。周囲のノードのストレージに空き容量があるならば、保持している観測データを分割配送する。

観測処理については、従来のセンサネットワークシステムと同様の処理を行うことで実現できる。しかし、離脱検知処理と分割配送処理については、これらの処理を実現するための機能を新たに追加する必要がある。また、従来のセンサネットワークでは、ネットワークの最大通信速度を超える頻度でのデータ送信については考慮されていないため、多数のノードが同時にストリームデータを送信した場合、基地局から離れたノードの観測データが届かなくなってしまう。このため、基地局から離れたノードのデータ損失を考慮し、すべてのノードがデータ送信の頻度を調整する帯域予約アルゴリズムが必要となる。

3.2 ネットワークからの離脱検知

3.1節でも述べたとおり、各ノードは、ネットワークからの離脱を自律的に検知する必要がある。この離脱検知を実現するために、センサネットワークにおいて基地局から定期的

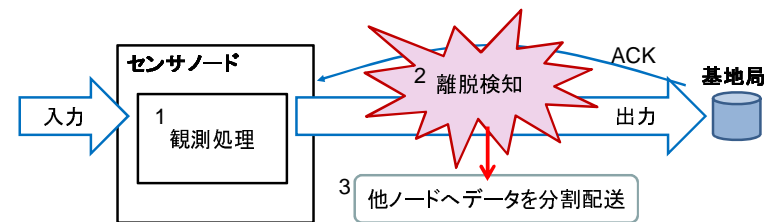


図3 提案手法の概要

にブロードキャストされる BEACON パケットと、観測データを受け取ったノードが送信元ノードに対して返信する ACK パケットを用いる。以下、それぞれのパケットについて述べる。

● BEACON パケット

BEACON パケットは、基地局からネットワーク内のノードに対して、定期的を送られるパケットである。別のネットワークから移動してきたノードが、BEACON パケットを受信した場合、BEACON パケットを送信した基地局をノードの観測データの送信先として設定することで、新たなネットワークに参入することが可能となる。また、ノードがさらに移動することにより所属するネットワークが切替わった場合にも、観測データの送信先を動的に切替えることが可能となる。この BEACON パケットが一定時間届かなかった場合、ネットワークからの離脱と判断される。

● ACK パケット

ACK パケットは、1 ホップ前の送信元ノードに対して送られる確認メッセージであり、基地局あるいは中継ノードが観測データを受信した場合に返信される。送信元のノードは、自らが観測したデータを保持し続け、ACK パケットの受信によりそのデータを消去する。ACK パケットが届かなかった場合、送信元のノードは、ネットワークから離脱したものと判断される。

各ノードは、これらのパケットを用いてネットワークからの離脱を検知する。その後の観測データについては、自らのストレージに空き容量があれば格納が行われ、空き容量がなければ次節で述べる分割配送処理が行われる。

3.3 ストリームデータの分割配送

3.1 節でも述べたとおり、自らのストレージに空き容量がない状態でストリームデータが生成された場合、データ損失を防ぐためには、周囲のノードにデータを分割配送する必要がある。分割配送されたデータが基地局に届くか否かについては、ノードがどのような経路を移動するかによって左右されるため、確実に届くことを保証することはできない。しかし、3.1 節で述べたように、より多くのデータを基地局に届けるという目的のためには、あるノードで持ちきれなくなったデータをネットワーク内の他のノードに渡すという本手法は有効であると考えられる。以下、分割配送処理における、ストリームデータ生成ノードと周囲のノードとの分割配送要求のやり取りについて述べる。

(1) ストリームデータを生成するノードは周囲のノードに対して、送信したいデータのサイズを通知する。

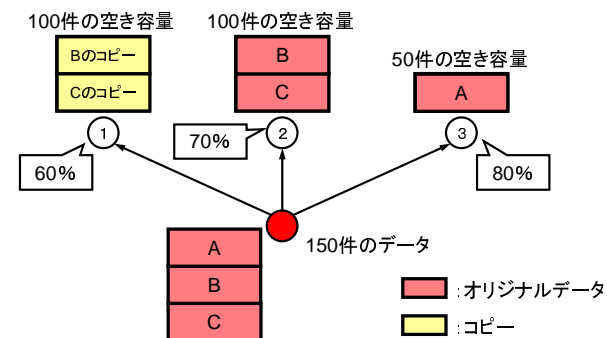


図 4 ストレージの空き容量に応じたデータ配送

- (2) 周囲のノードはストリームデータ生成ノードに対して、基地局との接続確率およびストレージの空き容量を通知する。
- (3) 周囲のノードのうち基地局との接続確率が高いノードから順番にデータの分割配送を行う。空き容量が十分であれば、コピーを生成して多重化配送する。

次に、分割配送について具体的に述べる。この分割配送を行うときには、まず接続確率の高いノードから順にオリジナルデータの配送を行う。その後、オリジナルデータの配送が終了しても空き容量に余裕があるならば、最も接続確率の低いノードに配送されたオリジナルデータから順にコピーを生成し、多重化して配送する。例えば、図 4 のような分断されたネットワークの場合、生成した 150 件のオリジナルデータは、基地局との接続確率が 80% であるノード 3 と 70% であるノード 2 に配送される。その後、接続確率が 60% のノード 1 のストレージには、150 件のデータのうち接続確率の低いノード 2 に配送された、データ B およびデータ C のコピーが配送される。このようにコピーを生成して多重化配送することにより、接続確率の低いノードに分割配送されたデータについても、基地局への到達率を上げることが可能となる。

3.4 帯域予約アルゴリズム

3.1 節で述べたとおり、センサネットワークでストリームデータを扱う場合には、基地局から離れたノードのデータ損失を考慮した帯域予約アルゴリズムが必要となる。本手法では、BEACON を受け取ったノードから順に利用したい帯域を確保し、残りの帯域を子ノードに対して割り当てるという方針に基づいて帯域予約を行う。しかし、この方針のみでは、基地局から離れたノードが帯域不足になる可能性が高い。そのため、BEACON を受け取っ

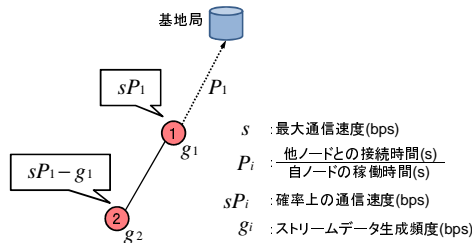


図 5 木構造経路での帯域予約の例

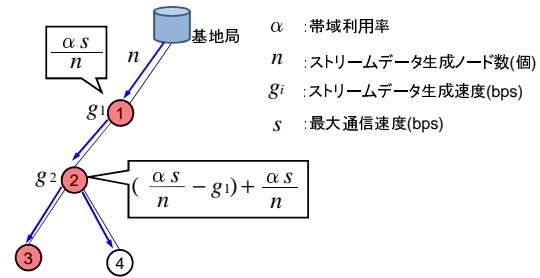


図 6 利用帯域の調整

たノードが帯域を利用するときには、すべてのノードの利用する帯域が、一定の割合を超えないようにする。これにより、基地局から離れたノードも帯域の利用が可能となり、観測データを基地局へ届けることが可能となる。

例えば、図 5 のようなネットワークの場合、ノード 1 に許可された帯域からノード 1 が生成するデータ量を引いた帯域が、ノード 2 に許可される帯域となる。ストリームデータの生成速度を g_i (bps)、ネットワークの平均通信速度を s (bps)、各ノードの接続確率を P_i (自ノードの稼働時間に対する他ノードとの接続時間の割合) と定義した場合に、ノード 1 の確率上の通信速度は sP_1 (bps)、ノード 2 は $sP_1 - g_1$ (bps) となる。ここで、BEACON パケットを受け取ったノードが利用可能な帯域の割合を α ($0 < \alpha < 1$) とすると、基地局に最も近いノードが予約可能な最大の帯域は αs (bps) となり、帯域不足が発生した場合の予備として $(1 - \alpha)s$ (bps) が残される。よって、ノード 1 が送信可能なデータは $(\alpha s)P_1$ (bps)、ノード 2 は $(\alpha s)P_1 - g_1$ (bps) となる。

次に、各ノードでの利用帯域の調整について述べる。図 6 のようなネットワークにおいて、ノード 3 に許可された帯域を上回るストリームデータがノード 3 で生成された場合、ノード 3 は基地局へ向けて帯域不足通知パケットを送信する。帯域不足通知パケットを受け取った基地局は、次に送信する BEACON パケットのネットワーク内のノード数 n を加算する。各ノードは、受け取った BEACON パケットを基に $\frac{\alpha s}{n}$ (bps) を超えない範囲での帯域予約を行う。図 6 において、ノード 1 が $\frac{\alpha s}{n}$ (bps) より小さい g_1 (bps) の帯域を利用した場合、ノード 2 に許可される帯域は $(\frac{\alpha s}{n} - g_1) + \frac{\alpha s}{n}$ (bps) となる。これらの帯域予約により、基地局付近のノードであっても帯域不足になる可能性が生じるが、ネットワーク内の全てのノードに対して帯域利用の公平性が確保される。

4. 評価

本章では、ストリームデータ分割配送手法の評価について述べる。本手法では、多くのデータを基地局に届けることを目的としているため、生成されたパケットの基地局への到達率を評価尺度として用いる。

本手法は、MANET のシミュレータである Network Simulator 2⁶⁾ 上に実装を行った。シミュレーションのパラメータとして、ノードの移動速度を 0~2m/s に、フィールドサイズを 1 辺が 100m の正方形に、シミュレーション時間を 100 秒に設定して、分割配送処理をしない既存手法との比較評価を行った。

以下、4.1 節でネットワーク内のノードの密度を変化させた場合の到達率の変化について述べる。次に、4.2 節でストリームデータ生成ノード数を変化させた場合の到達率の変化について述べる。最後に、4.3 節で本手法が有効な環境について考察する。

4.1 ノードの密度と到達率の関係

ネットワーク内のノードの密度とパケットの到達率の関係について述べる。図 7 に示すグラフの横軸はネットワーク内に存在するノードの数を示しており、数が大きいほどノードの密度が高くなっている。また、縦軸は基地局へのパケットの到達率を示している。この実験では、ストリームデータを生成するノードは、ネットワーク内に 1 つとしている。

図 7 より、ノード数が増えるほど、到達率が向上することがわかる。これは、提案手法と既存手法のどちらにもいえることであるが、ノードがネットワーク内を移動しても基地局までの経路が絶えず存在しているためであると考えられる。

また、ノード数が増えるほど、既存手法との到達率の差が小さくなっていることがわかる。これは、基地局までの経路が常に存在するようになるため、分割配送処理が行われず、

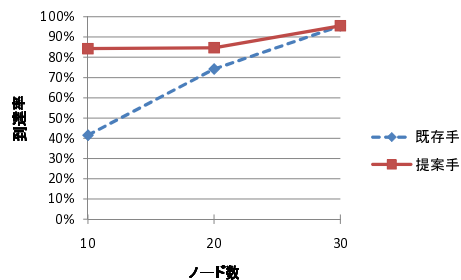


図7 ノードの密度と到達率の関係

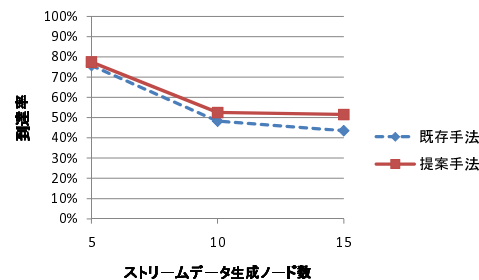


図8 ストリームデータ生成ノード数と到達率の関係

実質的に既存手法と同様の配送処理を行うことになるためである。

4.2 ストリームデータ生成ノード数と到達率の関係

ネットワーク内のストリームデータ生成ノード数とパケットの到達率の関係について述べる。図8に示すグラフの横軸はネットワーク内のストリームデータ生成ノード数を示しており、縦軸は基地局へのパケットの到達率を示している。この実験では、ネットワーク内のノード数として、4.1節の評価においてネットワークの分断が生じた20個を用いている。

図8より、ストリームデータ生成ノード数が増えるほど、到達率は低下することがわかる。これは、提案手法と既存手法のどちらにもいえることであるが、16Kbpsで生成されるストリームデータを送るための十分な帯域が確保できなくなるためであると考えられる。

また、ストリームデータ生成ノード数が増えるほど、既存手法との到達率の差が大きくなっていることがわかる。これは、移動に伴ってネットワークから離脱するノードにおいて、ストリームデータを生成するノードの割合が増加したため、分割配送処理による通信が増加したことが原因であると考えられる。

4.3 考察

4.1節より、ノードの密度が低く分断されたネットワークにおいて本手法は有効である。しかし、ノードの密度が低すぎた場合、分割配送対象となる周囲のノードが存在しなくなってしまうため、到達率は向上しないと考えられる。このため、分断されたネットワークには複数のノードが存在する必要がある。被災地におけるノードの移動を考えた場合、一人ひとりがランダムに移動するのではなく、集団でまとまって避難することが一般的であると考えられる。ゆえに、被災地でのバイタルデータの観測においては、既存手法をそのまま用いるよりも本手法を用いる方が有効であると考えられる。

4.2節より、ストリームデータ生成ノード数が増えるほど本手法が有効となる。ここでは、

移動によって離脱した場合の分割配送のみを対象として評価していたが、ネットワーク内の通信量が多いときのデータを一時的にバッファに保持しておくことも、到達率を上げるために有効である。この理由は、センサネットワークの一般的なストリームデータの観測環境においては、データの通信量にはバースト性があり、時系列で変化する特徴があるためである⁷⁾。本研究で想定するバイタルデータの観測においても、通信量が時系列で変化する特徴があるため、本手法を用いることが有効であると考えられる。

5. おわりに

本稿では、移動センシング環境について述べた後、ストリームデータを扱う場合のデータ損失の課題について述べた。次に、課題を解決するためのストリームデータ分割配送手法について述べた。最後に、本手法をNetwork Simulator 2上に実装し、分断されたネットワークにおいて多数のストリームデータが生成された場合のデータ到達率の向上を確認した。

今後の課題として、帯域予約アルゴリズムに関する実装と追加のシミュレーション評価、および実機における動作確認があげられる。

参考文献

- [1] Samel R. Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong: "TinyDB: An Acquisitional Query Processing System for Sensor Networks," ACM Transaction on Database Systems (TODS), 2005.
- [2] Lukasz Golab, and M. Tamer Ozsu: "Data Stream Management Issues - A Survey," SIGMOD Record, April 2003.
- [3] The STREAM Group: "STREAM: The Stanford Data Stream Management System," IEEE Data Engineering Bulletin, March 2003.
- [4] Arasu, Arvind and Babu, Shivnath and Widom, Jennifer: "The CQL continuous query language: semantic foundations and query execution," The VLDB Journal, vol.15, pp.121-142, 2006.
- [5] Abadi, D., Carney, D., Çetintemel, U., Cherniack, M., Convey, C., Lee, S., Stonebraker, M., Tatbul, N., and Zdonik: "Aurora: a new model and architecture for data stream management," The VLDB Journal, vol.12, pp.120-139, 2003.
- [6] The Network Simulator -ns2. <http://www.isi.edu/nsnam/ns/>.
- [7] Alizai, Muhammad Hamad and Landsiedel, Olaf and Link, J6 Ágila Bitsch and Götz, Stefan and Wehrle, Klaus: "Bursty traffic over bursty links," SenSys '09: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, pp.71-84, 2009.