

# オブジェクトの監視・追跡を行う 無線マルチメディアセンサネットワークの 稼働時間延長および QoS 確保のための ルーティング手法

藤本 恭平<sup>†1</sup> 安本 慶一<sup>†1</sup> 孫 為華<sup>†1</sup>  
山内 由紀子<sup>†2</sup> 伊藤 実<sup>†1</sup>

本稿では、移動オブジェクトを追跡し、ユーザ端末に動画をリアルタイム配送する無線マルチメディアセンサネットワーク (WMSN) において、配送される動画の QoS を保証し、WMSN 全体の稼働時間を最大化することを目指したデータ配送経路制御手法を提案する。対象 WMSN では、常に最短経路での動画配送を行うと、ノード間でのバッテリー消費に偏りが生じ、WMSN の稼働時間を縮めてしまうという問題がある。さらに、追跡すべきオブジェクトが多い場合には、動画配信によるトラフィックの増加のため、ネットワークの輻輳が起りやすい。本稿では、移動オブジェクトとユーザの地理的距離が大きいほど動画の許容配送遅延時間を長く設定可能な WMSN のアプリケーションを想定する。提案手法では、最短経路を含む複数の経路候補について、許容配送遅延時間と動画配信に必要な帯域の制約を満たし、動画配送終了後におけるバッテリー残量最小のノードのバッテリー残量を最大化するような経路を選択することで、WMSN の稼働時間を最大化する。本手法の有効性を示すため、典型的な例を用いて、提案手法と最短経路手法による WMSN の稼働時間を比較する。

## QoS-Aware Routing Method for Maximizing Lifetime of Wireless Multimedia Sensor Networks for Moving Object Tracking

KYOHEI FUJIMOTO,<sup>†1</sup> KEIICHI YASUMOTO,<sup>†1</sup>  
WEIHUA SUN,<sup>†1</sup> YUKIKO YAMAUCHI<sup>†2</sup> and MINORU ITO<sup>†1</sup>

In this paper, targeting a wireless multimedia sensor network (WMSN) that tracks moving objects and streams their video to user terminals in real time, we

propose a QoS-aware routing method which maximizes lifetime of the WMSN. In such a WMSN, always delivering video along the shortest path may deplete batteries of specific nodes, resulting in shorter WMSN lifetime. In addition, a large amount of traffic produced by video streams is likely to cause congestion due to radio interference among streams passing through a common radio range. In this paper, we target WMSN applications that allow a certain video playback delay depending on geographical distance between a user and the target object. We propose a method where among multiple possible video forwarding paths within allowable delay, each user node selects a path that maximizes the minimum residual battery node after the video delivery. We demonstrate effectiveness of the proposed method through a case study with a typical setting.

### 1. はじめに

近年、無線通信技術や半導体技術の発展により、無線通信デバイスやセンサデバイスの小型化、高機能化が進んでいる。また、小型で安価なカメラ、マイクロフォンなども利用可能になり、マルチメディア情報を扱うセンサデバイスを用いて構築するワイヤレスマルチメディアセンサネットワーク (WMSN) が実現可能になった<sup>1)</sup>。

本稿では、WMSN を用いたオブジェクト追跡システムに焦点を当てる。センサノードには人感センサとビデオセンサが搭載されており、移動する何らかの恒温動物 (オブジェクト) に対して人感センサが反応し、その場の様子をビデオセンサが撮影する。得られた動画はセンサノードによってエンコードされ、無線マルチホップ通信によって、ユーザが持つ携帯端末へと配送される。オブジェクトやユーザの移動に従い、撮影ノードや中継経路を再設定し、追跡を行いながらリアルタイムビデオストリーミングを行う。また、移動するオブジェクトに対して、ユーザが離れた位置から何らかの対応を取ることを可能にするため、オブジェクトの位置、すなわちイベントの発生地点からユーザまでの地理的距離が小さいほど緊急性が高いとみなし、それに応じて配送遅延許容時間を決定する。

提案システムの活用例として、猛獣対策が挙げられる。山間部や農村地域では、害獣・猛獣による人間への襲撃や、農作物への食害などの問題が以前から存在している。それらに対

<sup>†1</sup> 奈良先端科学技術大学院大学

Nara Institute of Science and Technology

<sup>†2</sup> 九州大学

Kyushu University

して様々な対策が実際に講じられているが、このシステムを用いることにより、ユーザは、送られてくる映像によって離れた場所の様子を確認しながら、猛獣に対して何らかの対応をとることができる。

WMSN ではマルチメディア情報を扱うため、データ量が大きく、多くのリソースを費やす。そのためネットワークの輻輳などが起こりやすくなり、動画の QoS を保証することが困難になる<sup>2)</sup>。また、データの配送を常に最短経路で行なった場合、特定のノードがデータ中継に長く使用され、ノード間でバッテリー消費量が偏ってしまう可能性がある。それにより、バッテリーが枯渇するノードが出現し、監視エリアの被覆性が損なわれたり、中継経路が切断されるなどして WMSN が満足に機能しなくなることが考えられる。

上記の問題に対して、データを分割してマルチパスで伝送することでネットワークの輻輳を軽減する手法<sup>3)</sup>や、中継途中のノードでパケットをバッファリングし、エラー時の高速な再送によりエンド間遅延を削減する手法<sup>4),5)</sup>が提案されている。しかし、これら既存手法は、ノードの電力消費について考慮しておらず、WMSN の稼働時間を延長する目的には使用できない。

本稿では、動画の QoS 保証と WMSN の稼働時間の最大化を目指した経路制御手法を提案する。提案手法では、人感センサおよびビデオセンサが搭載されたセンサノードが、ある広さの監視対象領域（フィールド）を全被覆するよう設置されていると想定し、フィールド内において、ユーザから一定距離以内で検出された移動オブジェクトの動画を、距離に基づいて設定される遅延許容時間内にユーザの携帯端末にストリーミング配信する。その際に、WMSN 全体の稼働時間を最長化するよう動画配送経路を決定することが目的である。

提案手法では、まず、あるノードの人感センサがオブジェクトを感知（イベントが発生）すると、そのノードを含む周辺ノードが協調して、オブジェクトを撮影するノード（ソースノード）を決定する。次に、ソースノードはイベントの発生とその位置を知らせるためのパケット（イベント発生通知パケット）をフラッディングする。その際、パケットには中継ノードの情報（ID、バッテリー残量など）を付加していく。ユーザ端末は異なる経路で送信される複数のイベント発生通知パケットを一定期間受信し、それぞれの経路候補について、動画配送にかかる配送遅延、および各ノードの使用帯域幅を計算する。そして、それらのうち、動画配送の制約（遅延時間および帯域）を満たす経路について、各ノードの現在のバッテリー残量から、動画配送終了後におけるバッテリー残量を予測し、残量が最小のノードのバッテリー残量が最大となるような経路を選択する。これにより、WMSN 全体の稼働時間を最大化するという目的を達成する。

以下、2 章では、関連研究について述べ、提案手法の位置づけを明確化する。3 章では、想定する環境、制約条件、配送経路決定問題を定式化する。4 章では、制約条件を満たし、かつ、WMSN の稼働時間を最大化するようなデータ配送経路を決定するアルゴリズムを述べる。5 章では、提案手法の適用例について述べ、最後に、6 章で、本稿におけるまとめを述べる。

## 2. 関連研究

WMSN の設計では、バッテリー消費の最小化、効率化に注力することが重要である。バッテリー容量や通信性能などの制約がある小型ノードにより、データ量が大きく、処理の複雑なマルチメディア情報を取り扱うため、ノードにかかる負担が大きくなる。そのため、ノードの寿命が早まり、システムの稼働時間が短縮されてしまうことが問題として挙げられる。さらに、WMSN では QoS についての課題が多く存在する<sup>1)</sup>。例として、バッテリー、メモリ、処理能力、最大データレートなどが制限されているため、これらの効率の良い利用が必須である。また、電波は干渉し、送信電力や経路制御、データレートなどによって干渉の程度は変化するため、リンクの容量、遅延は地理的關係に依存する。そのような環境での QoS の保証は困難である。

本章では、これらの課題のうち、リンクの容量や遅延に関する課題に着目し、取り組んでいるいくつかの研究について述べる。

Mao らは、データ配送に複数の経路を利用した、Multi-flow real-time transport protocol (MRTP)<sup>3)</sup> を提案した。これは、配送すべきデータをいくつかのバーストに分割し、それぞれのバーストを、異なる経路を通じて配送する手法である。送られたデータは受信機側でバッファリングされ、正しい順序に再構成される。これにより、個々のノードに課される通信量が減少するため、通信におけるノードの負担を軽減することができる。さらに、通信が狭い領域に集中することも回避でき、ネットワークの輻輳を防ぐ効果が期待できる。また、通信が分散することで、ノード間の電力消費の偏りを軽減する効果も期待できる。

Li らは、既存の手法である、複数の経路を用いてデータを配送する方法に加えて、途中のノードにデータのバックアップを残すことで、そのデータに対するリクエストがあった場合、本来のソースノードからではなく、バックアップを持っているノードからデータを配送することで、複数のソースノードの存在を仮想的に実現する手法を提案した<sup>6)</sup>。これにより、同じリクエストに対して応答するオーバーヘッドを最小限にし、遅延を軽減する。

Stann らは、Reliable multi-segment transport (RMST)<sup>4)</sup> を提案した。これは、データ

を中継途中のノードに一時保存し、データの損失などが起こったときに、データが保存されているノードから再送を行う手法である。通常、データの再送は最終的な目的地であるシンクが要求するが、この手法では中継経路上のノードもその機能を持ち、正しいデータが送られているか監視している。それにより、基地局に到着してから、ソースノードに再送を要求し、end-to-endで再送を行うよりも迅速な再送が実現できる。また、中継路上のノードでもデータの検査を行うことで、信頼性も向上する。

Wanらは、Pump slowly fetch quickly (PSFQ)<sup>5)</sup>を提案した。この手法は、先に示したRMSTと同様に、ノードのバッファにデータを一時保存する方法であるが、次々と送られてくるデータストリームに対し、受け取ったデータを先のノードに転送する動作よりも、バッファリングしたデータを再送する動作を優先するという手法である。これにより、シーケンス番号の小さいパケットが遅れて目的地に到達することを防ぎ、遅延の軽減が実現できる。また、RMSTと同様に、データの信頼性も向上する。

上記で述べた既存研究は、通信負荷の分散や、データの信頼性の向上、データ配送遅延の短縮などを目的としているが、これらの手法は、動画配信に対するQoSの保証やWMSN全体の稼働時間を延長するための資源割当機構や適応的制御機構を持たないため、移動オブジェクトの動画配信を実現するためにそのまま使用することはできない。

提案手法では、ノードのバッテリー残量を考慮し、残量の少ないノードを避けることや、使用可能な合計帯域幅の範囲内では配送できない経路を選択肢から外すような経路制御を行うという方針をとる。既存手法と異なる点は、配送する動画のQoSの保証と、ノードの電力消費の双方を考慮している点である。

### 3. WMSNの稼働時間を最大化するためのデータ配送経路決定問題

本章では、対象とするWMSNのモデル、および想定する環境について示し、WMSNの稼働時間を最大化するようなデータ配送経路を決定する問題を形式的に定義する。

#### 3.1 対象とするWMSNモデル、仮定、および諸定義

オブジェクト追跡WMSNでは、オブジェクトを撮影するためのビデオセンサと、オブジェクトの存在を検知するための人感センサを搭載したバッテリー駆動のセンサノードがフィールド上に多数配置される。人感センサによるオブジェクトの検知(イベントと呼ぶ)が発生すると、その地点の撮影に最適な視野を持つノードが選ばれる。ここで、最適視野ノードは、周囲のノードとの、人感センサのセンシング範囲の重なりから一意に計算できるものとする。また、それにかかる計算時間や電力消費などは考慮しない。選択されたノードは、ビデ

オセンサを起動し、撮影を開始する。得られた動画はセンサノード自身によってエンコードされ、無線マルチホップ通信で基地局に対して送信される。ここでの基地局とは、ユーザが所持する携帯端末であり、ユーザは携帯端末を通じて、撮影された動画をリアルタイムで閲覧することができる。また、携帯端末の画面にはオブジェクトの現在位置が表示され、複数のオブジェクトを検知している場合は、ユーザから最も地理的距離の近いオブジェクトの動画が配送されるものとする。オブジェクトがユーザから十分離れている場合は、動画の配送や位置情報の提供は行われない。その閾値を $r_{th}$ で表す。

対象とするフィールドを $Field$ 、フィールド内に存在するセンサノードの集合を $S = \{s_1, \dots, s_l\}$ 、センサノード $s \in S$ の位置を $s.pos$ で表す。

##### 3.1.1 ハードウェアについての仮定

フィールドに配置されたセンサノードは、バッテリー、人感センサ、ビデオセンサ、無線通信機能、動画エンコード機能を持っている。人感センサのセンシング範囲は、そのノードを中心とした半径 $r_s$ の円内である。この円内にオブジェクトが存在すれば、100%の確率で検知できる。ビデオセンサのセンシング範囲は、そのノードを中心とした、半径 $r_v$ 、視野角 $\theta$ の扇形であり、向きは固定されている。無線通信の電波到達範囲は、そのノードを中心とした半径 $r_c$ の円内である。通信量が通信可能帯域幅(後述)の範囲内なら、この円内に含まれるノードは100%の確率で通信内容を受信でき、この円内に含まれないノードは全く受信できない。また、1ホップあたりのデータの中継にかかる遅延を $PHD$ で表し、本稿では、どのような通信においても一定の値をとるものとする。エンコードされた動画のビットレートは $bw$ であり、すべてのストリームにおいて一定であるものとする。

携帯端末はバッテリー、無線通信機能、動画ストリーミング再生機能を持っている。バッテリー容量は十分に大きく、対象とする時間内に枯渇することはないものとする。

##### 3.1.2 フィールドについての仮定

実空間上でのシステム構築の際には、障害物や高低差などの条件により、ノードの配置に様々な制約が与えられると考えられるが、本稿では、簡単のため、障害物のない平面領域のみを想定し、すべてのノードは図2に示すようなハンカム構造で配置されているものとする。ノード間の距離を $d$ と表す。ここで、 $d$ は、センサノードの通信電波到達範囲 $r_c$ が自身の隣接ノードのみを含み、2ホップ先のノードは含まないように設定する。

また、本稿で想定する環境において、1つの動画ストリームを配送する場合、図1に示すように、あるノードの無線範囲内で、合計 $3bw$ の通信が行われると考えられる。

同一無線範囲内に存在する全ノードの合計使用可能帯域幅を $BW$ とし、 $BW > 3bw$ で

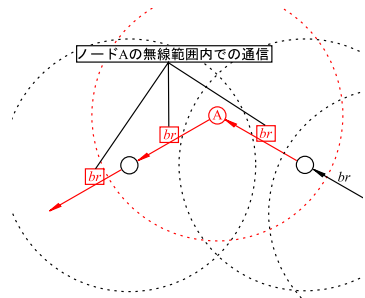


図 1 1つの動画ストリームを配送するために必要な帯域幅

あるとする。ビデオセンサおよび人感センサのセンシング範囲はそれぞれフィールド上のすべての範囲を、少なくとも1つ以上のノードがセンシング可能（フィールド全被覆状態）であるように配置されているとする。フィールド全被覆状態の例を図2に示す。

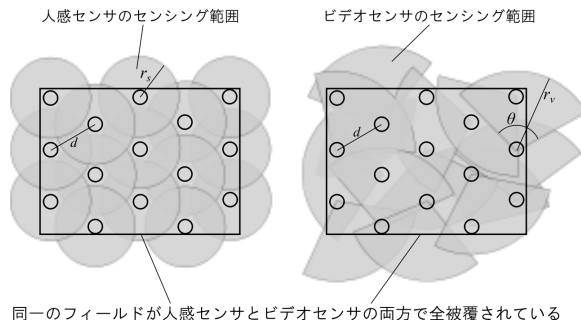


図 2 フィールド全被覆状態の例

フィールド上には、複数のユーザと複数の移動オブジェクトが存在する。また、オブジェクトの集合を  $V = \{v_1, \dots, v_l\}$  で表し、オブジェクト  $v$  の位置を  $v.pos$  で表す。

ユーザは一定速度で移動し、対象とする時間内に数が増減することはない。また、携帯端末はフィールドに配置されたすべてのセンサノードの位置情報を持っているとする。ユーザが所持する携帯端末の集合を  $U = \{u_1, \dots, u_l\}$ 、ユーザ端末  $u$  の位置を  $u.pos$  で表す。

イベントが発生すると、オブジェクトかユーザのどちらかがフィールドの外に出るか、もしくはオブジェクトとユーザの距離が  $r_{th}$  以上になるまで動画の配送が続く。したがって、現

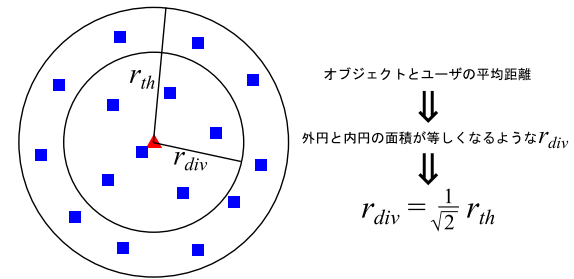


図 3 オブジェクトとユーザの平均距離の算出例

実は、どのイベントをどの程度の長さ配信することになるかはわからない。本稿では、動画の配送が始まってから一定時間  $T_{stream}$  でイベントが終了するものとする。一般に、ユーザは動画の配信が始まると、動画を確認しながら、オブジェクトから遠ざかる方向に移動すると考えられる。フィールド上に、イベントが等確率で発生するとした場合、各イベントの発生地点を重ね合わせると、ユーザは図3の四角で示すような位置に存在していると考えられる。したがって、オブジェクトとユーザの平均距離は、半径  $r_{th}$  の円の面積を2等分するような小円の半径であり、 $r_{th}/\sqrt{2}$  となる。ゆえに、ユーザの平均移動速度を  $V_u$  とすると、 $T_{stream} = r_{th}/(\sqrt{2}V_u)$  とすることで、平均的な動画の配信時間として用いることができると考えられる。 $T_{stream}$  の間に配送されるデータ量  $B_{stream}$  は、 $T_{stream} \times bw$  で計算される。

### 3.1.3 電力消費モデル

センサノードは、バッテリー駆動を想定し、バッテリーの交換はできないものとする。センサノード  $s \in S$  のバッテリー残量を  $s.energy$  で表すと、 $s.energy$  は無線通信によるデータの送受信時、ビデオセンサによるオブジェクト撮影時、動画のエンコード時に消費される。

$x[\text{bit}]$  を  $d[\text{m}]$  通信するときの消費電力  $Trans(x, d)$ 、および  $x[\text{bit}]$  を受信するときの消費電力  $Recep(x)$  は式 (1), (2) に従う<sup>7)</sup>。

$$Trans(x, d) = E_{elec} \times x + \epsilon_{amp} \times x \times d^\alpha \quad (1)$$

$$Recep(x) = E_{elec} \times x \quad (2)$$

ここで、 $E_{elec}$  はハードウェアの消費電力係数、 $\epsilon_{amp}$  は信号増幅器の消費電力係数、 $\alpha (\geq 0)$  は電波の減衰係数である。本稿では、屋外を想定し、 $\alpha = 2$  とする。

$y[\text{s}]$  時間オブジェクトを撮影するための電力  $Sens(y)$  は式 (3) に従う<sup>7)</sup>。

$$Sens(y) = E_{sens} \times y \quad (3)$$

$x[\text{bit}]$  の動画をエンコードするための電力  $Encode(x)$  は式 (4) に従う<sup>8)</sup>.

$$Encode(x) = E_{encode} \times x \quad (4)$$

ここで,  $E_{sens}$ ,  $E_{encode}$  はそれぞれオブジェクト撮影時の単位時間あたりの電力消費係数, エンコード時の単位ビットあたりの電力消費係数である.

### 3.1.4 制約条件

ノード間の接続関係の集合を  $E = (s_i, s_j) | s_i, s_j \in S \wedge |(s_i, s_j)| \leq r_c$  とする. ここで,  $|(s_i, s_j)|$  はノード  $s_i$  と  $s_j$  のユークリッド距離である. あるノード  $s \in S$  からあるユーザ端末  $u \in U$  までの経路を  $\langle s, s_{i_1}, \dots, s_{i_k}, u \rangle$  で表す. 既に動画を配信中の経路の集合を  $Path$  とする. 新たに発生したイベントに対して割当てる動画配信経路の集合を  $Path'$  とする. 各経路  $path \in Path \cup Path'$  は連結でなければならない. 従って, 式 (5) を満たす必要がある.

$$\forall \langle s, s_{i_1}, \dots, s_{i_k}, u \rangle \in Path \cup Path', \\ (s, s_{i_1}) \in E \wedge (s_{i_1}, s_{i_2}) \in E \wedge \dots \wedge (s_{i_k}, u) \in E \quad (5)$$

動画の配送には, 中継ノードの無線通信資源を必要とする. また, 中継ノードに含まれていない場合でも, 隣接ノードが通信を行なっている場合は, 傍受により自身の帯域が使用される. そのため, ノード  $s$  が送信するデータ量と受信する (傍受も含む) データ量の総和が合計使用可能帯域幅  $BW$  を超えてはならない. したがって, 各ノードの無線範囲内で行われる通信の, 合計使用帯域幅は式 (6) を満たす必要がある.

$$\forall s \in S, BWinUse(s) < BW \quad (6)$$

ここで,  $BWinUse(s)$  はノード  $s$  の無線範囲内で, 現在使用中の帯域幅を表し, 式 (7) に従う.

$$BWinUse(s) = \sum_{s \in \{s\} \cup Neigh(s)} \left[ 3bw \cdot incl(s', Path \cup Path') \right. \\ \left. - bw \cdot is\_source(s', Path \cup Path') \right] \quad (7)$$

ここで,  $bw$  は各動画配信が使用する帯域,  $Neigh(s)$  はノード  $s$  の隣接ノードの集合,  $incl(s, Path)$  は,  $s$  が集合  $Path$  に所属するいずれかの経路に含まれる時 1, そうでない時 0 を返す関数,  $is\_source(s, Path)$  は,  $s$  が,  $Path$  のいずれかの経路のソースノードである時 1 を返し, そうでない時 0 を返す関数である.

イベントが発生すると, ビデオセンサによるオブジェクトの撮影が開始され, データの配送経路が決定した後, 中継ノードを通じてユーザの携帯端末へと動画が配送される. このとき, ユーザとオブジェクトとの距離に基づいて決定される一定の動画配送遅延が許容される

ものとする. ユーザとオブジェクトの距離は, ソースノードから携帯端末までの直線距離とし, 距離  $dist$  の時の遅延許容時間は式 (8) に従う<sup>\*1)</sup>.

$$Delay(dist) = D \times dist \quad (8)$$

ここで,  $D$  はアプリケーション毎に予め決定される係数である<sup>\*2)</sup>. 各イベントに対する動画は, この遅延許容時間内に配送されなければならない. したがって, 求めるべき各経路  $path \in Path'$  において, 撮影してから携帯端末に届くまでの時間  $delay(path)$  は式 (9) を満たす必要がある.

$$\forall path \in Path \cup Path', delay(path) < Delay(|src(path).pos, dst(path).pos|) \quad (9)$$

ここで,  $src(path)$ ,  $dst(path)$  は, それぞれ, 経路  $path$  におけるソースノード, 配送先ノードを表す. また, 式 (9) はすべての動画配送が, それぞれの遅延許容時間内に配送されることを示している.  $delay(path)$  は経路  $path$  の end-to-end 遅延であり, 式 (10) に従う.

$$delay(path) = PHD \times |\{s | incl(s, path) = 1\}| \quad (10)$$

ここで,  $|path|$  は経路  $path$  のホップ数である.

新規にイベントが発生したノードの集合を  $S'$  とする. 各イベント発生ノード  $s \in S'$  について, 距離  $r_{th}$  以内にユーザが存在する場合には, 動画が配送されなければならない. この制約を式 (11) に示す.

$$\forall s \in S', \exists path \in Path', s = src(path) \vee \forall u \in U, |s.pos, u.pos| > r_{th} \quad (11)$$

各ユーザ  $u \in U$  の距離  $r_{th}$  以内にイベントが発生した場合, 最も近いイベントの動画が  $u$  に配送されなければならない. この制約を式 (12) に示す.

$$\forall u \in \{u | u \in U \wedge s \in S' \wedge |u.pos, s.pos| \leq r_{th}\} \\ \exists path \in Path', u = dst(path) \wedge src(path) = \min_{s \in S'} |s.pos, u.pos| \quad (12)$$

## 3.2 問題の定式化

本問題の目的は, 個々のイベントに対して, 無線通信資源の制約を満たし, 遅延許容時間内に動画を配送するような, 連結な中継経路の中から, 動画配送終了後における, バッテリー残量最小のノードのバッテリー残量を最大化する経路を選択することである.

入力として,  $Field$ ,  $S'$ , ノード  $s \in S$  の  $s.pos$ ,  $s.energy$ , オブジェクト  $v \in V$  の  $v.pos$ , ユーザ  $u \in U$  の  $u.pos$ , 定数  $E_{elec}$ ,  $\epsilon_{amp}$ ,  $E_{sens}$ ,  $E_{encode}$ ,  $D$ ,  $\alpha$ ,  $bw$ ,  $BW$ ,  $PHD$ ,

\*1 一般に, 猛獣などの移動オブジェクトとの距離が近いほど危険であり短い遅延での動画配信が要求される.

\*2 例として,  $D = 1/V_v$  とすることで, ユーザがオブジェクトと出会う前の動画受信を保証できる.

$r_{th}$ ,  $T_{stream}$ ,  $d$  を与える.

問題の出力は, 新規に発生したイベント群に対する, 各ソースノード  $s \in S$  から携帯端末  $u$  までのパス  $\langle s, s_{i_1}, \dots, s_{i_k}, u \rangle$  の集合  $Path'$  である.

イベント発生時点での, ノード  $s \in S$  のバッテリー残量を  $s.energy$ , 経路集合  $Path'$  の全ての動画配送終了後のバッテリー残量を  $s.energy'$  とすると,  $s.energy'$  は式 (13) に従う.

$$s.energy'(Path') = \begin{cases} s.energy - \Delta_s & \text{if } (is\_source(s, Path \cup Path') = 1) \\ s.energy - \Delta_m & \text{if } (incl(s, Path') = 1) \\ s.energy & \text{otherwise} \end{cases} \quad (13)$$

ここで,  $\Delta_s$  はデータのセンシング, エンコード, 送信にかかる電力量,  $\Delta_m$  は受信と送信にかかる電力量を表し, 式 (14), (15) に従う.

$$\Delta_s = Sens(T_{stream}) + Encode(bw \cdot T_{stream}) + Trans(bw \cdot T_{stream}, d) \quad (14)$$

$$\Delta_m = Trans(bw \cdot T_{stream}, d) + Recep(bw \cdot T_{stream}) \quad (15)$$

本問題の目的関数を式 (16) のように設定する.

$$\text{maximize } \min_{s \in S} (s.energy'(Path')) \quad \text{subject to } (5), (6), (9), (11), (12) \quad (16)$$

## 4. WMSN の稼働時間を最大化するためのデータ配送経路制御手法

本章では, 前章で定義した問題に対して解を導くアルゴリズムを述べる.

### 4.1 基本方針

前章で述べたとおり, 提案手法では, 各イベントに対して, オブジェクトとユーザの距離に応じた動画配送の遅延を許容し, その遅延許容時間を活用して, 最短経路以外のデータ中継経路を選択肢に入れることを基本方針とする. これにより, 最短経路上にバッテリーの消耗したノードが存在する場合や, 通信が混雑し, 帯域不足の領域が存在する場合に, それらのノードや領域を中継経路に含まない迂回経路を選択できるようにする.

#### 4.1.1 イベント発生メッセージのフラッディング

イベントが発生すると, ソースノードはイベント発生通知メッセージ  $rreq$  を周囲にフラッディングする. これは, ユーザが持つ携帯端末にイベント発生を知らせるだけでなく, 経由したノードの情報を  $rreq$  メッセージに付加していくことで, 中継ノードのバッテリー残量などの情報を得るためである. また, 経由したノードの ID を順番に付加していくことで,  $rreq$  メッセージ内に配送経路の候補を構築する. 本手法では, いくつかの  $rreq$  メッセージ

を受信した携帯端末が, その情報を用いて動画の配送経路決定のための計算を行う. フラッディングする情報は, ネットワーク内を無限にループすることを防ぐために, ホップカウンタ  $cnt$  が付加される. ここで,  $cnt$  の初期値を  $C$  とする.  $rreq$  メッセージがユーザ端末に到達し, ユーザ端末が返信する動画配信要求メッセージ ( $rrep$  メッセージ) がソースノードに到達, そしてソースノードから配信する動画がユーザ端末に到達するまで, 合計 3 回の end-to-end 配送が行われる. また, 本稿では通信 1 ホップあたりの遅延を一定の値  $PHD$  と仮定している. したがって,  $C = \text{floor}(\text{Delay}(r_{th})/3PHD)$  とすることで, 最も大きく迂回する場合でも遅延許容時間内に動画を配送できるようにする.

#### 4.1.2 携帯端末のメッセージ待機時間

あるイベントに対してフラッディングされた  $rreq$  メッセージを, ユーザが所持する携帯端末が受け取ったとき, ネットワーク内には他の経路を辿っている  $rreq$  メッセージが多数存在していると考えられる. そのため, 最初の  $rreq$  メッセージを受信してから時間  $T_{wait}$  の間, 待機を行う. ソースノードから自身までの距離を  $dist$ , 最初に受信した  $rreq$  メッセージに含まれる  $path$  のホップ数を  $n$  とすると,  $T_{wait} = (\text{Delay}(dist)/3) - n \cdot PHD$  とすることで, 受信したすべての  $rreq$  メッセージに含まれる経路に対して, 遅延許容時間内に動画を配送することができる.

#### 4.1.3 使用帯域幅の予測

$rreq$  メッセージには, 経由したノードの, その時点での使用帯域幅が付加されている. その値に, 新たに動画を配送する場合に確保すべき帯域幅を加えたものが, 使用帯域幅の予測値である. 本手法では, その予測値が使用可能無線帯域幅を越えている場合, そのノードを含む経路を配送経路候補から除外する. 3.1 節で述べたように, あるノード  $s$  を経由して帯域幅  $bw$  の動画を配信するには,  $s$  の無線範囲内に  $3bw$  以上の空き帯域が必要である.

#### 4.1.4 バッテリー残量の予測

$rreq$  メッセージに付加された, そのノードの, その時点でのバッテリー残量に, 動画配送におけるデータの送信と受信のコストを差し引いたものが, そのノードの動画配送終了後におけるバッテリー残量の予測値である. バッテリー残量最小のノードのバッテリー残量を最大化するという目標を達成するために, 本手法ではこの予測値を各経路の評価値として用いる.

### 4.2 アルゴリズム

提案手法では, 動画の QoS 保証を実現し, かつ, WMSN 全体の稼働時間を最大化するために, ある時刻でのイベント  $v \in V$  の発生に対して, 動画の中継に使用する配送経路を決定する. この目的のために, センサノードの動作についてのアルゴリズムと, ユーザが所

持する携帯端末の動作についてのアルゴリズムを提案する。

#### 4.2.1 センサノードにおける動作アルゴリズム

センサノード上で動作するアルゴリズムの擬似コードをアルゴリズム 1 に示す。

---

#### アルゴリズム 1 センサノード $s \in S$ における動作アルゴリズム

---

```
1: while true do
2:   if イベントが検出され  $s$  がソースノードに決定 then
3:      $cnt = C$  とし,  $s.id$  を付加した rreq メッセージをブロードキャスト
4:   end if
5:   if rreq メッセージ受信 then
6:     if  $cnt > 0$  and メッセージに  $s.id$  が含まれていない and  $|src(path).pos, s.pos| \leq r_{th}$  then
7:        $cnt = cnt - 1$  とする
8:       rreq メッセージに  $s.id, s.energy, BWinUse(s)$  を付加してブロードキャスト
9:     else
10:      rreq メッセージを破棄
11:    end if
12:  end if
13:  if rrep メッセージ受信 then
14:    if  $s$  がソースノード then
15:      rrep メッセージに含まれる次ホップノードに動画を配信
16:    else if  $s$  が中継ノード then
17:      rrep メッセージに含まれる親ノードに rrep メッセージを送信
18:      経路制御表に経路 (ソース, 宛先, 次ホップノード) を記録
19:    else
20:      rrep メッセージを破棄
21:    end if
22:    if 動画データを受信 then
23:      経路制御表の次ホップノードに転送
24:    end if
25:  end if
26: end while
```

---

センサノード  $s \in S$  について、以下を繰り返す (1 行目)。

イベントが検出され、ノード  $s \in S$  がソースノードとなった場合、自身のノード ID  $s.id$  と  $cnt = C$  としたホップカウンタを付加した rreq メッセージをブロードキャストする (2-4 行目)。

rreq メッセージを受信したノード  $s \in S$  は、 $cnt = cnt - 1$  とし、rreq メッセージに自身のノード ID  $s.id$ 、その時点でのバッテリー残量  $s.energy$ 、使用帯域幅  $BWinUse(s)$  を付加し、

ブロードキャストする。 $cnt = 0$  になるまでこれが繰り返され、rreq メッセージがフラグディンクされる (5-8 行目)。このとき、受け取った rreq メッセージに、自身のノード ID がすでに含まれていた場合は、rreq メッセージがループしているので破棄する。また、ホップカウンタが 0 になった場合、および、ソースノードから自身までの距離  $|src(path).pos, s.pos|$  が  $r_{th}$  より大きい場合も rreq メッセージを破棄する (9-10 行目)。

携帯端末からの rrep メッセージを受信した場合、ノード  $s$  が rrep に指定されている経路のソースノードなら、rrep メッセージに含まれる子ノードに対して動画を送信する (13-15 行目)。ノード  $s$  が rrep に指定されている経路上の中継ノードなら、rrep メッセージを親ノードに対して送信し、経路制御表に経路 (ソース, 宛先, 次ホップノード) を記録する (16-18 行目)。ノード  $s$  が rrep の経路に含まれなければ、rrep メッセージを破棄する (19-20 行目)。

ノード  $s$  が動画を受信した場合、親ノードから受信した動画を次ホップノードに配信する (22-23 行目)。

本アルゴリズムでは、各ノードの資源割り当てに関する排他制御を行っていない。そのため、連続で起こった 2 つ以上のイベントで、同じノードを経由する配送経路が計算される可能性がある。そのノードの利用可能帯域が  $3bw$  以上  $6bw$  未満であった場合、利用可能帯域の制約を満たしているとアルゴリズムが判断したにもかかわらず、動画配送時に帯域不足が発生する。しかし、本稿ではそのような状況は発生しないと仮定し、各イベントの発生には十分な間隔が空いているものとする。

#### 4.2.2 携帯端末における動作アルゴリズム

携帯端末上で動作するアルゴリズムの擬似コードをアルゴリズム 2 に示す。

---

**アルゴリズム 2** 携帯端末  $u \in U$  における動作アルゴリズム

---

```
1: while true do
2:   if rreq メッセージを受信 then
3:     遅延許容時間  $Delay(|src(path).pos, u.pos|)$  を計算する
4:      $T_{wait}$  時間 rreq メッセージを受信し続け集合  $M$  に格納する
5:      $max\_eval = 0$  とする
6:     for each  $m \in M$  do
7:        $path\_eval = \infty$  とする
8:       if  $delay(m.path) > Delay(|src(m.path).pos, u.pos|)/3$  then
9:         メッセージ  $m$  を破棄
10:      end if
11:      for each  $s \in \{s' | incl(s', m.path) = 1\}$  do
12:        if  $BWinUse(s) + 3bw > BW$  then
13:          メッセージ  $m$  を破棄
14:        else
15:           $eval = s.energy - (Trans(bw \cdot T_{stream}, d) + Recep(bw \cdot T_{stream}))$  とする
16:          if  $eval < path\_eval$  then
17:             $path\_eval = eval$ 
18:          end if
19:        end if
20:      end for
21:      if  $max\_eval < path\_eval$  or  $max\_eval = path\_eval$  and  $|m.path| < |path|$  then
22:         $path = m.path, max\_eval = path\_eval$  とする
23:      end if
24:    end for
25:     $path$  から動画配信要求メッセージ rreq を構成しソースノードへ送信
26:  end if
27: end while
```

---

携帯端末  $u \in U$  について、以下を繰り返す (1 行目) .

携帯端末が rreq メッセージを受信すると、ソースノードの位置から、動画配信の遅延許容時間  $Delay(|src(path).pos, u.pos|)$  を計算する。また、その他の経路を経由してくる rreq メッセージの到着を待つため、 $T_{wait}$  時間受信し続ける (2-4 行目) . 経路の最小バッテリー残量のこれまでの最大値を表す変数  $max\_eval$  を 0 で初期化する (5 行目) . ホップカウンタが 0 になる前にユーザが所持する携帯端末に届いたメッセージの数は、それぞれのメッセージが経由してきたノードを中継経路とする動画配信経路の候補の数となる。受け取った rreq メッセージの集合を  $M = \{m_1, \dots, m_l\}$ , rreq メッセージ  $m \in M$  が経由した経路を  $m.path$ , 経路  $m.path$  含まれるノードの集合を  $s \in \{s' | incl(s', m.path) = 1\}$  で表す。

まず、すべての rreq メッセージ  $m \in M$  について、以下を実行する (6 行目) . 現在対象にしている経路のノードの中で最小のバッテリー残量を表す変数  $path\_eval$  の初期値を  $\infty$  とする (7 行目) . 次に、その経路の合計遅延時間が動画配信の遅延許容時間より大きい場合、メッセージ  $m$  を破棄する (8-9 行目) . 各ノード  $s \in \{s' | incl(s', m.path) = 1\}$  の現在の使用帯域幅  $BWinUse(s)$  に、それぞれ動画配信に使用する帯域  $3bw$  を足し合わせ、動画を配信した場合の合計予測使用帯域幅を計算する。合計予測使用帯域幅が合計使用可能帯域幅  $BW$  を超えていれば、そのメッセージを破棄する (11-13 行目) .  $BW$  を超えていなければ、そのノードの現在のバッテリー残量  $s.energy$  から、データの送信にかかるコスト  $Trans(bw \cdot T_{stream}, d)$ , 受信にかかるコスト  $Recep(bw \cdot T_{stream})$  をそれぞれ引き、バッテリー残量の予測値  $eval$  を算出する (14-15 行目) .  $eval < path\_eval$  なら、 $path\_eval = eval$  とする (16-17 行目) . すべての  $s \in \{s' | incl(s', m.path) = 1\}$  について以上の動作が終了すると、もし  $max\_eval < path\_eval$ , または  $max\_eval = path\_eval$  かつ  $|m.path| < |path|$  なら、 $path = m.path, max\_eval = path\_eval$  とする (21-22 行目) . ここまでを、すべての rreq メッセージ  $m \in M$  について繰り返す。

最後に、配送経路情報  $path$  と配送遅延許容時間  $Delay(|src(path).pos, u.pos|)$  を付加した rreq メッセージをソースノードに宛てて送信する (25 行目) .

### 4.3 提案するアルゴリズムの動作例

提案アルゴリズムの携帯端末における動作例について述べる。図 4 は、あるイベントに対していくつかのメッセージを受信した携帯端末が、動画配信経路決定のための計算を行う例を示している。図 4(a) は、携帯端末が受け取った各メッセージがそれぞれ経由してきた経路を、動画の配信経路候補とするときの動作を示している。現実には、さらに多くの候補が存在すると考えられるが、ここでは簡単のため、A~E の 5 つの候補のみが遅延時間の制約を満たしたものとする。図 4(b) は、動画の配信を行った場合の使用帯域幅を考慮し、最大使用可能帯域幅を超えてしまう経路を除外するときの動作を示している。ここでは、候補 B と候補 D に含まれるノードにおいて、帯域不足が予想されるため、これら 2 つの候補を除外する。図 4(c) は、動画の配信を行った場合の消費電力を考慮し、動画配信終了後のバッテリー残量の予測値が、各経路上で最小となるノードのバッテリー残量を算出するときの動作を示している。図 4(d) は、図 4(c) で求めた各経路候補の予測値を比較し、最も大きい値を持つ経路を動画配信経路として選択するときの動作を示している。このとき、予測値が最大となる経路が複数存在した場合は、よりホップ数の少ない経路が選択される。以上の動作により、ある時刻でのイベントに対して、ソースノードから携帯端末までの動画配信経路を



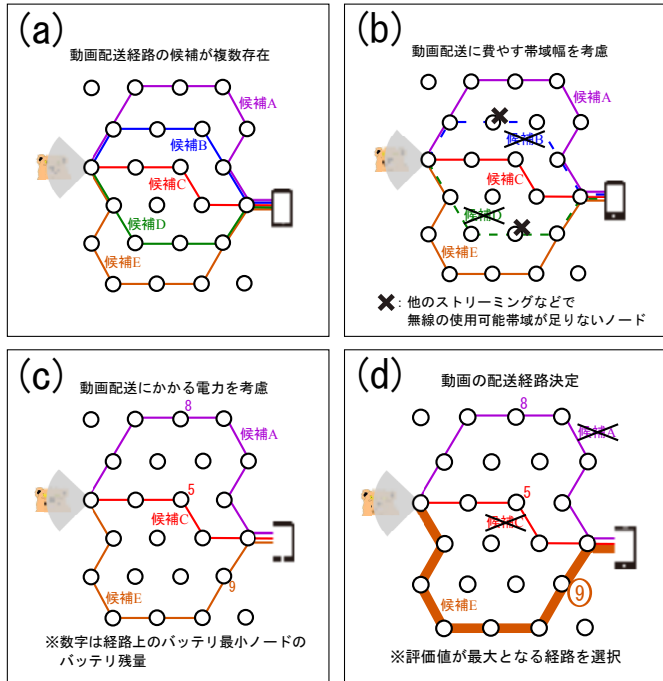


図 4 アルゴリズム 2 の動作例

決定する。

## 5. 提案手法の評価

本章では、提案手法の有効性を示すために、典型的な例を用いて評価を行う。常に最短のホップ数で経路を形成する最短経路法と提案手法で、いくつかのイベントが終了した時点での、バッテリー残量最小のノードのバッテリー残量を比較する。

### 5.1 適用環境

対象 WMSN として、図 5 に示すように、フィールド上に 25 個のノードをハニカム構造で配置したものを扱う。表 1 は、イベントの発生箇所とユーザの位置の組み合わせ、およびそれぞれの組み合わせにおける動画の配信経路を示している。イベントの発生箇所とユーザの位置は、乱数を用いて決定した。これらは、記号で示すノードのセンシング範囲、もしくは

は通信範囲内に位置することを示している。例えば、イベント発生箇所が D ならば、ノード D がソースノードとなる位置でイベントが発生することを示し、ユーザの位置が W ならば、ノード W から動画が送られる位置に存在することを示している。動画の配信経路は、各手法を適用して決定したものである。バッテリー残量は初期値を 100 % とし、1 回のイベントに伴う動画配信で、ソースノードは 20 %、中継ノードは 10 % バッテリーを消費するものとする。

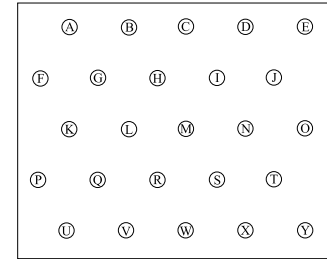


図 5 センサノードの配置図

表 1 イベントの発生箇所とユーザの位置、および各手法を適用したときの動画配信経路

イベント番号	イベントの発生箇所	ユーザの位置	最短経路法での経路	提案手法での経路
1	O	B	⟨ O,N,I,H,B ⟩	⟨ O,N,I,H,B ⟩
2	M	P	⟨ M,L,Q,P ⟩	⟨ M,L,Q,P ⟩
3	K	D	⟨ K,G,H,I,D ⟩	⟨ K,G,H,C,D ⟩
4	D	Q	⟨ D,I,M,L,D ⟩	⟨ D,J,N,S,R,Q ⟩
5	J	L	⟨ J,I,M,L ⟩	⟨ J,I,H,L ⟩
6	G	S	⟨ G,L,M,S ⟩	⟨ G,L,R,S ⟩
7	V	H	⟨ V,R,M,H ⟩	⟨ V,W,X,T,N,I,H ⟩
8	E	X	⟨ E,J,N,S,X ⟩	⟨ E,J,O,T,X ⟩
9	A	T	⟨ A,G,H,M,N,T ⟩	⟨ A,F,K,P,Q,R,S,T ⟩
10	T	F	⟨ T,N,M,L,G,F ⟩	⟨ T,X,W,V,U,P,K,F ⟩

### 5.2 シミュレーション結果と考察

表 1 が示すような経路を用いて動画を配信したときの、各ノードのバッテリー残量を計算し、各イベント終了後のバッテリー残量最小のノードのバッテリー残量を、各手法の評価値とする。

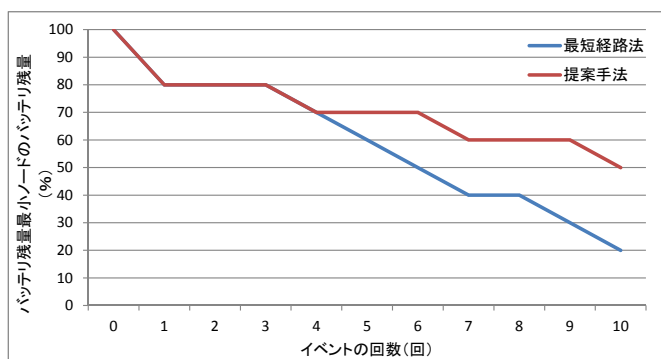


図6 各手法を適用したときのイベント発生回数と最小バッテリー残量の関係

イベントの回数と、バッテリー残量最小のノードのバッテリー残量の関係を図6に示す。グラフは、横軸がイベントの通算回数を表し、縦軸がバッテリー残量最小のノードのバッテリー残量を表す。縦軸の値が大きいほど、良い結果を示していると言える。図6より、イベントの回数が増えるに従い、最短経路法と提案手法の、バッテリー残量最小のノードのバッテリー残量の差が開いていくことがわかる。表1より、序盤では、各ノードのバッテリー残量が十分にあり、2つの手法がほぼ同じ経路を形成していることがわかる。しかし、4回目からは経路の形成に異なった傾向が見られ、それに従い、各手法の評価値にも差が出てきている。10回のイベントを終了した時点で、提案手法は最短経路法に比べて2倍以上のバッテリーを残すことができた。以上より、提案手法が、常に最短経路で動画を配送する手法に比べて、WMSNの稼働時間を大きく延長することが可能であることがわかった。また、より広いフィールドに対して、より多くのノードを配置して実験を行えば、配送経路の選択肢が増えるため、2つの手法の差がより顕著に表れると考えられる。

### 5.3 問題拡張についての議論

本稿では、送られてくる動画ストリームに対し、中継ノードは転送速度を変化させることなく、次のノードにデータを転送するという仮定に基づき、問題設定を行った。しかし、中継ノードに十分な容量のバッファが搭載されており、データを一時保存できると仮定すれば、通信混雑時に、一時的に使用帯域を小さくすることが可能となる。図7に示すように、ビットレート  $bw_1$  で、 $[t_0, t_1]$  の時間、動画を配送するとき、使用帯域を小さくすることにより送信できなくなるデータを、バッファに一時保存することで、 $[t_0, t_2]$  の時間に、ビット

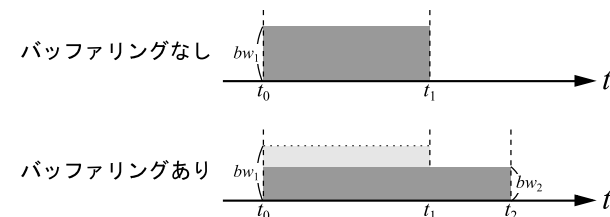


図7 中継ノードでのバッファリングを考慮したときの使用帯域の変化

レート  $bw_2 (< bw_1)$  で、データを破棄することなく配送することができる（ただし、遅延許容時間を満たす範囲でのみ配送を遅らせることができる）。

今後の課題として、上記のような仮定に基づいた問題設定を行い、それを解くためのアルゴリズムの設計・評価を行うことなどが挙げられる。

## 6. おわりに

本稿では、移動オブジェクトの監視・追跡を目的とした WMSN において、要求される遅延許容時間内に動画を配送するという制約を満たしながら、WMSN の稼働時間を最大化するようなデータ配送経路を求めるためのアルゴリズムを提案した。提案手法では、各ノードにおける使用可能帯域幅の制限を超えないという条件の下で、動画配送終了後における各ノードのバッテリー残量を予測し、バッテリー残量最小のノードのバッテリー残量を最大化するような配送経路の選択を行う。提案手法の有効性を示すために、常に最短経路でデータを配送する手法と提案手法で、典型的な例を用いて評価を行った。その結果、10回のイベントが終了した時点で、バッテリー残量最小のノードのバッテリー残量について、提案手法が最短経路法よりも2倍以上大きい値を示した。以上より、提案手法が WMSN の稼働時間最大化に有効であることがわかった。

今後、より詳細な設定を用いたシミュレーションにより、評価実験を行う予定である。また、中継ノードのバッファにデータを一時保存できる場合の問題を設定し、それを解くためのアルゴリズムを設計することが課題として挙げられる。

## 参考文献

- 1) I.F. Akyildiz, T. Melodia, K.R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, Vol. 51, pp. 921-960, 2007.

- 2) S. Misra, M. Reisslein, and G. Xue, "A survey of multimedia streaming in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 10, pp. 18–39, 2008.
- 3) S. Mao, D. Bushmitch, S. Narayanan, and S. S. Panwar, "MRTP: A Multi-Flow Realtime Transport Protocol for Ad Hoc Networks," In *IEEE Transactions on Multimedia*, vol. 8, No. 2, pp. 356–369, 2006.
- 4) F. Stann and J. Heidemann. "RMST: Reliable data transport in sensor networks," In *Proceedings of 2003 IEEE Sensor Network Protocols and Applications (SNPA)*, pp. 102–112, 2003.
- 5) C. Y. Wan, A. T. Campbell, L. Krishnamurthy, "PSFQ: a reliable transport protocol for wireless sensor networks," in: *Proceedings of the 1st ACM International workshop on Wireless Sensor Networks and Applications*, 2002, pp. 1–11.
- 6) D. Li, Q. Zhang, C.-N. Chuah and S. J. Ben Yoo, "Multi-source multi-path video streaming over wireless mesh networks," *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, p.698, 2006.
- 7) 勝間 亮, 村田佳洋, 柴田直樹, 安本慶一, 伊藤 実, "過剰にノードを用いることによるセンサネットワークの稼働時間延長方式," マルチメディア, 分散, 協調とモバイル (DICOMO2009) シンポジウム, pp. 322 - 332, 2009.
- 8) 孫タオ 玉井森彦 安本慶一 柴田直樹 伊藤実, "場面の重要度に基づいて再生品質制御を行う省電力ビデオストリーミングシステム," 情報処理学会論文誌 46(2), 546–555, 2005.