

## 受信端末始動型マルチホップ無線ネットワークにおける 制御パケットの衝突を考慮した性能改善

小南 大智<sup>†</sup> 菅野 正嗣<sup>††</sup> 村田 正幸<sup>†</sup> 畠内 孝明<sup>†††</sup> 町田 潤一<sup>†††</sup>

<sup>†</sup> 大阪大学大学院情報科学研究科 〒 565-0871 吹田市山田丘 1-5

<sup>††</sup> 大阪府立大学総合リハビリテーション学部 〒 583-8555 羽曳野市はびきの 3-7-30

<sup>†††</sup> 富士電機システムズ 〒 191-8502 日野市富士町 1

E-mail: <sup>†</sup>{d-kominami,murata}@ist.osaka-u.ac.jp, <sup>††</sup>sugano@rehab.osakafu-u.ac.jp,

<sup>†††</sup>{hatauchi-takaaki,machida-junichi}@fesys.co.jp

**あらまし** 無線センサネットワークの主な課題として消費電力の抑制がある。掲記方式では、受信側始動型の非同期式間欠動作を行い、各無線機は間欠的に自身の ID を送信することにより受信可能状態であることを通知する。我々はこれまでの研究において、間欠送信の周期が掲記方式に大きな影響を与えることを明らかにしてきた。本稿では、掲記方式における制御パケットの衝突を防ぐために、衝突確率を最小化する間欠周期の解析的な導出及びデータアグリゲーションの導入について議論する。シミュレーションの結果から、間欠周期を固定的に割り当てる場合と比較して、パケット収集率を向上させた上で、最大 90 % 以上の消費電力削減が可能となることを示す。

**キーワード** センサネットワーク, 間欠動作, シミュレーション, 消費電力

### Performance Improvement by Collision Avoidance Mechanism in Receiver-driven Multi-Hop Wireless Networks

Daichi KOMINAMI<sup>†</sup>, Masashi SUGANO<sup>††</sup>, Masayuki MURATA<sup>†</sup>, Takaaki HATAUCHI<sup>†††</sup>, and

Junichi MACHIDA<sup>†††</sup>

<sup>†</sup> Graduate School of Information Science and Technology, Osaka University

1-5 Yamadaoka, Suita-shi, Osaka, 565-0871, Japan

<sup>††</sup> School of Comprehensive Rehabilitation, Osaka Prefecture University

3-7-30 Habikino, Habikino-shi, Osaka, 583-8555, Japan

<sup>†††</sup> Fuji Electric Systems

1, Fujicho, Hino-shi, Tokyo, 191-8502, Japan

E-mail: <sup>†</sup>{d-kominami,murata}@ist.osaka-u.ac.jp, <sup>††</sup>sugano@rehab.osakafu-u.ac.jp,

<sup>†††</sup>{hatauchi-takaaki,machida-junichi}@fesys.co.jp

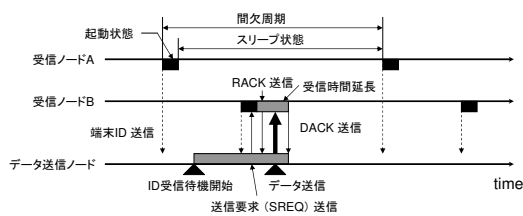
**Abstract** A major challenge in wireless sensor networks is energy saving. In the intermittent receiver-driven data transmission (IRDT) protocol, which aims to save energy, communication between nodes commences when multiple receiver nodes transmit their own IDs intermittently and a sender node receives them. Our previous research focused on the performance characteristics of IRDT when this intermittent transmission interval changes. In this paper, we analyze the probability of control packet collisions as a function of the intermittent interval and introduce a procedure for determining the proper interval that minimizes this probability. We also present data aggregation in IRDT. Through simulations, we show that IRDT with a proper interval and data aggregation can attain a higher packet collection ratio and a large reduction in energy consumption. Particularly, our proposed method achieves power consumption that is 90 % lower than that of the original IRDT.

**Key words** sensor network, intermittent operation, simulation, energy consumption

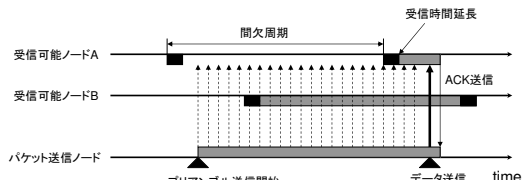
#### 1. ま え が き

センサネットワークにおいて解決すべき技術的課題として、センサ端末の小型化や低コスト化、低消費電力化等があり、中

でも有限容量のバッテリーによって動作するセンサ端末においては消費電力の低減化は重要な課題である。消費電力の低減化には様々なレイヤからのアプローチがあり、例えば、半導体技術の高度化や回路の微細化、充電機器の搭載、ノードにスリープ



(a) Intermittent Receiver-driven Data Transmission (IRDT) 方式



(b) Low Power Listening (LPL) 方式

図 1 アドホックネットワークにおける非同期間欠通信方式

状態と起動状態を設けることで間欠的にデータを送受信するようなアクセス制御、送信電力が通信距離の2~4乗に比例することから通信に中継を用いたマルチホップルーティングアルゴリズム等が研究されている[1-4].

特に、通信を行わない時には無線ノードをスリープ状態とし、間欠的に起動状態へ遷移して通信を行う間欠動作によって、消費電力の大きな削減が可能である。これはスリープ時の消費電力が起動時の数百から数千分の一であることに基づいている[5]. 間欠動作に基づく通信を行うための起動タイミング制御方式は、同期式[4, 6]と非同期式[7-10]に分類できるが、同期のためのオーバーヘッドを要しない非同期式が、電力消費の抑制では優れていると考えられる。そこで我々は、受信側始動型の非同期式間欠動作を行なう IRDT (Intermittent Receiver-driven Data Transmission) 方式を提案し[9], 送信側始動型の LPL (Low Power Listening)[8]と比較することで、その基本性能を明らかにした[11]. 図1に2つの方式の基本動作を示す。

IRDT方式では、各無線機が間欠的に自身のIDを周囲に送信し、自身が受信可能状態であることを通知する(図1(a)). データを所持しているノードはIDの受信を待機し、受信したIDが通信先候補からのものであれば送信要求(SREQ)パケットを送信する。SREQ受信確認を通知するRACKパケットの到着後にデータパケットを送信して、データ受信確認のDACKパケットを待つ。このように、IRDT方式では通信先候補として複数のノードを待ち受けることが可能である。

一方IRDT方式の問題点として、1度のIDパケット送信に対して複数のノードからSREQパケットが同時に返信されることによるSREQパケット同士の衝突がある。例えばシンクノードのID送信に対して、データを所持する複数のノードがIDを受信すると、シンクノードにデータを送るために同時にSREQパケットを送信してSREQ同士の衝突が発生する。この問題を解決するアプローチとしては、間欠周期をパケット発生率に応じて適切な値に設定する方法とデータ送信回数を減らす方法が考えられる。

前者に関しては[11]で既に取り上げており、負荷の状況に適した間欠周期で動作させるために、SREQパケットの衝突に応じて間欠周期を動的に制御する手法を提案した.[11]での提案

手法では、SREQ待機中にビットエラーを検出したノードは、負荷の集中を予測して間欠周期を $T_{min}$ にまで下げ、通常はID送信に要する電力を抑制するために間欠周期を $T_{max}$ まで徐々に増加させる。しかしこの方式では、衝突が発生してから対処を行うために、SREQ衝突の発生頻度は抑制されるが再発生する確率が高い点が問題であるといえる。各ノードにおいてトポロジが既知の場合には、各ノードは自身の負荷の推定が可能となるため、あらかじめ自身の間欠周期として負荷に対して適切な値を設定することでSREQ同士の衝突回避が期待できる。後者についてはデータアグリゲーションによる実現が考えられる。データアグリゲーションを行うことで、ノードに送られてきた複数のデータを1つのデータに集約してデータ送信回数を減らすことが可能となる[12].

本稿では、以前の研究で提案した受信側始動型間欠動作による低消費電力通信方式であるIRDT方式において、間欠周期の設定方法の改善、データアグリゲーション機能の導入が性能に及ぼす影響を明らかにする。また、それらを共に用いる時の低消費電力化の効果を、送信側始動型の間欠動作を行うLPL方式と比較することで検証する。

## 2. 間欠周期による制御パケットの衝突回避

この章では、IRDT方式における制御パケットの衝突確率を間欠周期の関数として定式化し、制御パケットの衝突確率を最小化する間欠周期の導出を行う。

間欠周期がIRDT方式の性能へ与える影響については[11]で述べたことから以下の3点にまとめられる。

### (1) SREQパケット同士の衝突

あるノードが送信したIDに対して複数のノードが同時にSREQパケットを返信すると互いに衝突してしまう。その結果、データ送信側ノードの起動状態が続き大きく電力を消費してしまう。

### (2) IDパケットと他のパケットとの衝突

IRDT方式ではデータを受信可能な状態のノードが間欠的にIDパケットを送信するため、IDパケットが通信中の他のパケットと衝突を起こしてしまう。この問題は短い間欠周期で顕著に発生し、再送の必要が生じるため消費電力は増加する。

### (3) IDパケット待機時間

IRDT方式において電力の多くは、データ送信時ではなくIDパケット待機のための無線起動時に消費され、そのため間欠周期が短いほど消費電力は抑制できる。逆に非常に短い間欠周期は端末の起動時間を増加させ、消費電力を増やすことになる。

SREQパケット同士の衝突率・IDパケットの衝突率・IDパケット待機時間は、間欠周期による制御が可能といえる。本節では制御パケットの衝突を最小化させる間欠周期(以降は $T^*$ と表記)の解析的な導出方法を提案する。

### 2.1 制御パケットの衝突確率の導出

まずは以降で用いる用語と仮定について述べる。ネットワーク中の各ノードはネットワークトポロジを把握しており、隣接する1ホップ先(後)のノードを、前向きノード(後向きノード)、同ホップ数のノードを横向きノードとして登録している。各ノードは間欠周期 $T$ で動作し、毎秒平均 $\lambda$ 個のデータパケットを生成してシンクノードに送る。この際、前向きノードに対してのみデータを送ると仮定し、複数の前向きノードがある際

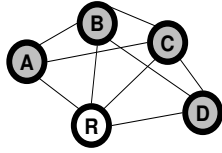


図2 ネットワークモデル

にはそれぞれ等確率でデータを送るものとする。いずれの packets 送信時にも CSMA/CA による衝突回避を行うが、隠れ端末問題により packet 受信中に別の packet を受信した場合には、いずれの packet も正常に受信できないものとする。

上記仮定より任意のノード  $r$  の 1 秒あたりの平均データ受信個数  $G(r)$  は以下で与えられる。ただし、 $N_b(r)$  はノード  $r$  の後向きノードの集合であり、その要素を  $n$  とする時、 $|N_f(n)|$  は  $n$  の前向きノード数である。

$$G(r) = \sum_{n \in N_b(r)} \frac{1}{|N_f(n)|} \lambda(G(n) + 1) \quad (1)$$

SREQ が衝突するのは 2 つ以上のノードが SREQ を送信する場合である。ここでノード  $R$  が各後向きノードから受け取るデータ個数は等しいと仮定すると、各後向きノードが ID を受信した際に SREQ を送信する確率はそれぞれ  $1 - e^{-G_b(R)T}$  となる。ただし  $G_b(R) = \frac{G(R)}{|N_b(R)|}$  である。以上から、SREQ が衝突する確率  $P_{SREQ}$  は以下の式で表される。

$$P_{SREQ} = 1 - \frac{1 - |N_b(R)| + |N_b(R)|e^{G_b(R)T}}{e^{G(R)T}} \quad (2)$$

ここで、図 2 に示すネットワークモデルにおいてノード  $R$  が ID を送信する場合の SREQ 同士の衝突を考えると、この時互いに隠れ端末となるのはノード  $A$  とノード  $D$  のみである。パケットの衝突が発生するのはノード  $A \cdot D$  間のみであると考え、式 (2) における  $|N_b(R)|$  は  $4 (= |\{A, B, C, D\}|)$  から  $2 (= |\{A, D\}|)$  に修正する必要がある。このように CSMA/CA を考慮した  $P_{SREQ}$  は以下の式で与えられる。

$$P_{SREQ} = \sum_{k=2}^{|N_b(R)|} C_n(k) \left( 1 - \frac{1 - k + ke^{G_b(R)T}}{e^{G(R)T}} \right) \quad (3)$$

ここで  $C_n(k)$  は、 $N_b(R)$  から異なる  $k$  個のノードを選ぶ時に、 $k$  個のノード全てが互いに隠れ端末となる組み合わせ数である。ただし、互いに異なる  $k$  個のノードの集合を  $S(k)$  とし、 $m > k$  なる  $m$  が存在する時、任意の  $S(k)$  に対して  $S(m) \supset S(k) \cup S(k)$  が成り立つものとする。

次に ID パケットの衝突を考える。ノード  $R$  がデータを受信する際に受信状態を取るのには SREQ 受信とデータ受信の際であり、この時 ID パケットが送信されると衝突が発生する。ここではノード  $R$  がデータを受信する場合を考える。まずはノード  $R$  についての平均隠れ端末数  $H(R)$  を次のように定義する。

$$H(R) = \frac{1}{|N_a(R)|} \sum_{n \in N_a(R)} h(R, n) \quad (4)$$

$H(R)$  は、ノード  $R$  に隣接するノードがノード  $R$  にデータを送る際に、隠れ端末となっているノードの平均個数を表している。この時、 $N_a(R)$  はノード  $R$  の隣接ノードの集合、 $|N_a(R)|$

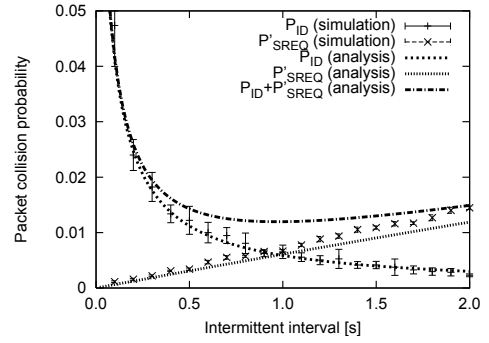


図3 制御パケットによる衝突確率

はその要素数である。また、 $T_r$  は SREQ パケットとデータパケットの受信時間であり、通信速度を  $C$ 、SREQ とデータの packet サイズの和を  $M$  とする時、 $\frac{M}{C}$  である。 $h(R, n)$  は、ノード  $R$  の隣接ノード  $n$  と互いに隠れ端末となるノードのうち、 $N_a(R)$  に含まれるノードの個数である。

$H(R)$  を用いると、ノード  $R$  がデータを受信する際の平均 ID 受信間隔は  $\frac{T}{H(R)}$  で表される。以上から、ID 送信による衝突が発生する確率  $P_{ID}$  は以下の式で表される。

$$P_{ID} = \frac{T_r H(R)}{T} \quad (5)$$

図 3 は、図 2 のトポロジでの、制御パケットの衝突に関する解析及びシミュレーションの結果である。ここで  $\lambda = 0.024$  であり、エラーバーは 95 % の信頼区間を表している。 $P_{ID}$  については、解析とシミュレーションの結果がほぼ一致していることが確認できる。 $P_{SREQ}$  に関しては、間欠周期が長くなるほど解析結果よりもシミュレーション結果のほうが大きな値を示しており、これは CSMA/CA によって完全に衝突回避できることを仮定した点が原因である。CSMA/CA を用いた場合もバックオフ時間が一致した場合にはパケットの衝突が発生するため、より多くの隣接ノードがデータパケットを所持している場合ほどこの影響は大きくなる。また、図 2 はシングルホップネットワークであるが、マルチホップネットワークを考える際には、IRDT 方式が横向き・後向きの転送も行うために隣接ノードの 1 秒あたりの packet 受信率にはばらつきが生まれる。さらに MAC 層レベルでの再送も行うために、1 秒あたりの packet 受信率が増加し、 $P_{SREQ}$  は大きくなる。一方で  $P_{ID}$  に関しては、1 秒あたりの packet 受信率が大きくなるために ID 送信頻度が僅かに下がり、また、ID 送信自体がキャリアセンスの結果によっては行われず、結果的に減少することになる。

## 2.2 間欠周期の設定

制御パケットの衝突確率を最小化する間欠周期を導出するために、 $P_{SREQ}$  に修正を加える。式 (3) では  $P_{SREQ}$  はノード  $R$  が ID を 1 つ送信する際の SREQ 衝突確率を表しており、式 (5) では  $P_{PROP}$  はノード  $R$  がデータを受信する際に ID による障害が発生する確率を表している。前者をデータ 1 つの受信あたりの衝突確率とした  $P'_{SREQ}$  は  $P_{SREQ}$  と  $(G(R)T)^{-1}$  の積から得られ、この  $P'_{SREQ}$  と  $P_{ID}$  の和を最小化させる間欠周期として  $T^*$  が得られる。 $T^*$  を定式化することは困難であるため、区間 (0.0 s, 2.0 s) において 10 ms 刻みで  $P'_{SREQ} + P_{ID}$  の値を計算することで、近似的に最小値を与える  $T^*$  を得てい

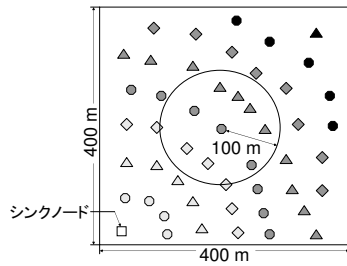


図4 ネットワークモデル

る。ただし、最小値が得られない場合に  $T^*$  は、自身と後向きノードの消費電力和を最小化する周期  $T_m$  に設定している。  $E_s$  ,  $E_w$  をそれぞれ 1 秒あたりの送信電力, 受信待機電力とする時, ノード  $R$  の後向きノードが ID の待ち時間によって 1 秒あたりに消費する電力の総和は  $\sum_{n \in N_b(R)} \frac{E_w T}{[N_f(n)]+1} G_n$  である。またノード  $R$  が間欠動作によって 1 秒あたりに要する電力は, ID 送信に要する時間  $T_p$  とその後の SREQ 待機時間  $T_s$  を用いて,  $\frac{E_s T_p + E_w T_s}{T}$  となる。パケット待機時間を最小化させる間欠周期  $T_m$  はそれぞれの和を最小化させる値であれば良く, 以下の式で与えられる。

$$T_m = \sqrt{\frac{(E_s T_p + E_w T_s)}{E_w \sum_{n \in N_b(R)} \frac{G(n)}{[N_f(n)+1]}} \quad (6)$$

### 3. IRDT 方式におけるデータアグリゲーション

データアグリゲーションの導入によりデータパケットの送信回数の減少が期待される。本稿で対象とするデータアグリゲーションでは複数のデータを集約する際にデータサイズは増加するものとし,  $m$  個のデータを集約する際にはデータサイズは  $m$  倍になると考える。IRDT 方式においてデータアグリゲーションの機能を追加する方法には以下の 2 通りが考えられる。

#### (1) データ所持時の間欠動作継続

IRDT 方式において, データパケット受信時あるいは発生時に, ノードは直ちに ID 待機状態へと遷移する。この時, ID 待機状態へすぐには遷移せずに一定時間だけ間欠動作を行い, データの受信を行うことで, データアグリゲーションが実現できる。一定時間の経過, あるいはデータの集約個数が一定数を超過した際に ID 待機状態へと遷移する。

#### (2) ID 待機状態でのデータパケット受信

これまでの IRDT 方式では, データ所持時には ID を待機して自身は ID を送信していなかった。ID 受信待機時にも送信先からの ID が届くまで間欠的に ID 送信を行うことで, データの受信を行うことが可能となる。データの集約個数が一定数を超過した際には, それ以上のアグリゲーションは行わない。

(1) の方法ではデータアグリゲーションを積極的に行うことでデータ送信回数が減少可能であり, (2) の方法では遅延を増加させずにデータアグリゲーションが行える。本稿では, より高い省電力効果が期待される前者を扱うこととする。

ここでデータアグリゲーションを行う際に, [11] で行った前向き転送を優先する送信先の選択方法を変更する。SREQ パケットの衝突が起こるのは, 同一ホップ数の隣接ノード同士がデータパケットを持っている場合である。この点は, データアグリゲーションが可能な場合には, 前向き転送に加えて横向き転送

表1 パラメータ設定値

パラメータ名	値
無線速度	100 kbps
送信時電流	20 mA
受信時電流	25 mA
スリープ時電流	0 mA
ID パケット	40 byte
SREQ, RACK, DACK パケット	26 byte
データパケット	128 byte

も優先的に行うことで解決できる。ID を送信してきたノードがデータを所持しているかどうかは, ID 内にその情報を含めることで知ることができる。

また, アグリゲーションを行う際のデータ集約個数に定めた制限は, 式 (5) における  $T_r$  の増加による  $P_{ID}$  の増加, チャネル占有時間の増加を防ぐためである。このため, ID パケットを受信したノードが SREQ を返信する条件として, あらかじめ定めた集約可能なデータ個数を超えない点を追加する。これは, 先ほどの横向き転送のための「データを所持しているかどうかの情報」を, 「現在所持しているデータがいくつのデータを集約したものか」という情報で表せば実現できる。

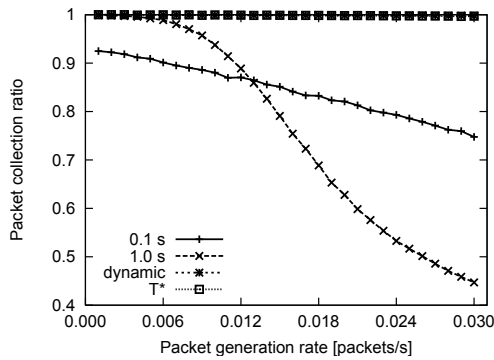
## 4. シミュレーション結果

本章では IRDT 方式に対して (1) 制御パケットの衝突を最小化させる間欠周期の設定 (2) データアグリゲーション機能の追加, 上記 2 点をそれぞれ導入する際の性能特性を明らかにする。さらに, (1), (2) を共に導入する際の IRDT 方式の性能特性を LPL 方式と比較する。

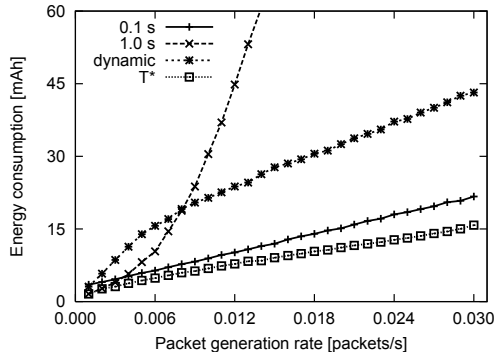
ネットワークモデルには, 一辺 400 m の正方形領域上に 1 個のシンクノードと 49 個のセンサノードを配置したものを利用する (図 4)。ノードの通信可能距離は 100 m であり, シンクノードからのホップ数ごとにノードを異なる形と色で図に示している。通信速度, 電流, パケットサイズは表 1 に示した値を用いており, その他のパラメータに関しては [11] と同様である。ネットワーク上のシンクノード以外のノードはポアソン過程に従いデータを発生させ, マルチホップでシンクノードにデータを送信する。この際に 1 秒あたりに発生される平均のデータ個数であるパケット発生率を変化させる時の, パケット発生率・消費電力 (負荷が最大のノードにおける消費電力及び全ノード間の平均消費電力; 以降では最大消費電力及び平均消費電力と記す) それぞれの変化について以降の節では調べていく。

### 4.1 間欠周期 $T^*$ 導入時の性能評価

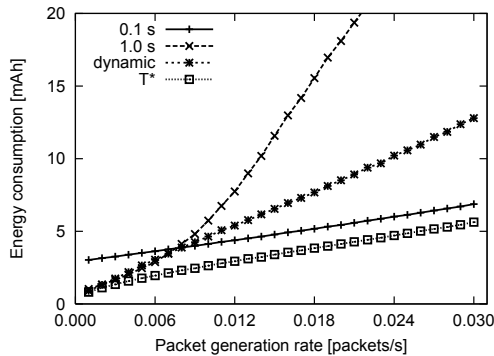
図 5 は, 2.2 節で得られた  $T^*$  を各ノードが設定する際の収集率及び消費電力特性である。また, 間欠周期を 0.1 s, 1.0 s に設定した場合及び [11] で提案した動的な設定を行う際の結果も示している。図 5(a) をみると, 間欠周期が 0.1 s の時は, パケット発生率が低い場合でも収集率は 100 % 付近にならない。しかしパケット発生率が増加しても収集率の低下はほとんど起こらず, 常に 75 % 以上の収集率が得られている。一方間欠周期が 1.0 s の時はパケット発生率が低い際には 100 % に近い収集率が得られているが, パケット発生率の増加に従い急激に低下している。これらは式 (3), (5) から説明ができる。式 (3) あるいは図 3 から, 間欠周期が長いほど SREQ パケットの衝



(a) パケット収集率



(b) 最大消費電力

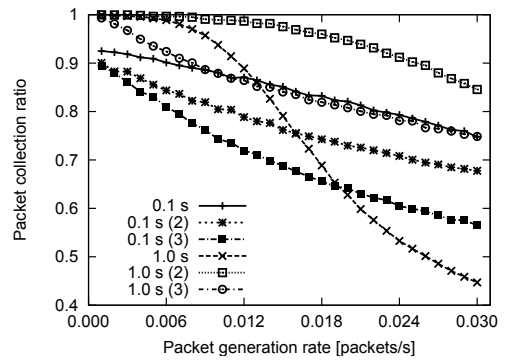


(c) 平均消費電力

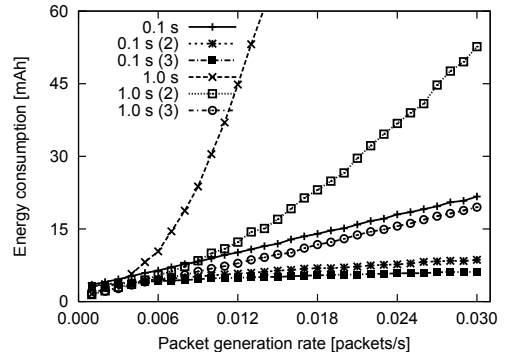
図5  $T^*$  導入時の性能特性

突確率は増加する。さらに2.2節で述べたように、パケット発生率の増加と共に SREQ パケットの衝突の繰り返しの影響が大きくなる。その結果、間欠周期が 1.0 s の時に急激に収集率が低下する。一方、式 (5) より、間欠周期が一定の場合には、一定の確率で ID パケットによる衝突が起こりパケット収集率が低下することが分かる。0.1 s の間欠周期ではこの影響が強くみられている。動的な間欠周期の設定を行う場合、 $T^*$  を設定する場合のいずれにおいてもこれらは改善され、図のパケット発生率の範囲内において収集率が 100 % に近い値を常に示していることが分かる。

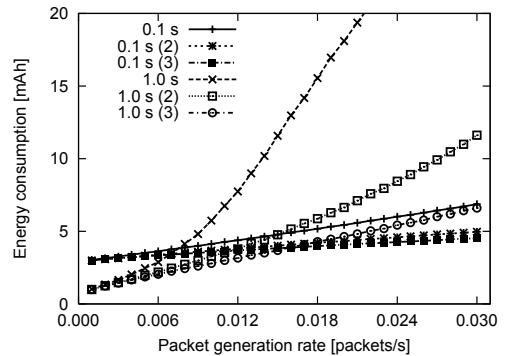
消費電力に関しても、制御パケットの衝突を抑えることで消費電力の増加を抑えることができている (図 5(b), 5(c))。最大消費電力に関して動的な設定方法と比較すると、パケット発生率が 0.001 の時は 51 %、パケット発生率が 0.030 の時では 36 % にまで削減できている。同様の比較を平均消費電力についても行うと、パケット発生率 0.001 の時には 90 %、パケット発生率 0.030 の時には 56 % にまで削減できている。



(a) パケット収集率



(b) 最大消費電力



(c) 平均消費電力

図6 データアグリゲーション導入時の性能特性

#### 4.2 データアグリゲーション導入時の性能評価

図 6 には、データアグリゲーション機能を導入する際の IRDT の性能特性を示しており、集約可能なデータの個数をラベルの括弧内に示した数字で表している。各ノードはアグリゲーションのためにデータ受信直後に転送を行わずに 5 s だけ受信状態を取っている。間欠周期が 1.0 s の時をみると、2 個までのアグリゲーションを行う場合にはパケット収集率は増加しているが、3 個とすると逆に減少している。他方で 0.1 s の場合にはアグリゲーションを行うことで常に減少している。これらはデータパケットサイズの増加によって ID との衝突確率が増加することが原因である。また、大きなデータ集約個数は一度のパケットロスが大きく収集率を低下させてしまう。以上より、パケット収集率に関しては、間欠周期が長い際は 2 個までのアグリゲーションが SREQ 衝突回避の点で効果があるが、間欠周期が短くなるほどに ID の衝突が発生し逆効果となることがわかる。

消費電力に関しては最大、平均消費電力共にアグリゲーション個数を増やすほど減少しているが、パケット発生率が低い場

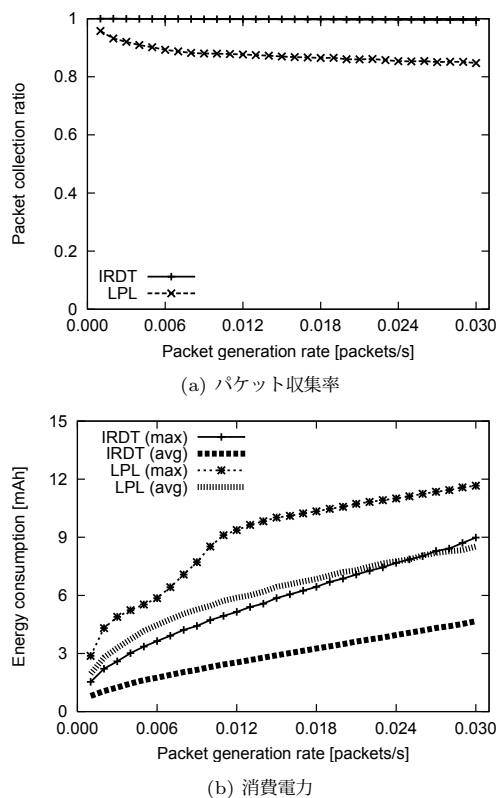


図7  $T^*$  及びデータアグリゲーション導入時の性能特性

合には 5 s の待機ではアグリゲーションがほとんど行われな  
 ために効果はみられない。また間欠周期 0.1 s の場合では、最  
 大 2 個行う時と最大 3 個行う時に差がほとんど無く、ID の通  
 信阻害による再送が消費電力を増加させていると考えられる。  
 パケット発生率が 0.030 の時、間欠周期が 1.0 s の時には最大  
 消費電力で 83 % の、平均消費電力で 77 % の削減ができており、  
 間欠周期が 0.1 s の時にも同様に最大消費電力では 70 %、平均  
 消費電力で 47 % 削減されている。以上の消費電力抑制は横  
 向き転送の選択が特に効果的であり、シンク隣接ノードにお  
 ける SREQ 同士の衝突を抑えることができていると考えられる。

#### 4.3 LPL 方式との性能比較

図 7 には IRDT 方式に対して制御パケット衝突回避のため  
 の間欠周期の設定とデータアグリゲーションを導入する時の  
 LPL 方式との比較結果を示している。ただし、データアグリ  
 ゲーションを行う際に集約可能なデータ個数は 2 個として  
 いる。さらに個数を増やすことで消費電力を抑制が可能であるが、  
 パケット収集率の低下を防ぐためにここでは 2 個としている。  
 LPL 方式においても最大 2 個のデータアグリゲーションを行  
 っており、間欠周期は式 (6) と同様に消費電力を最小化させる  
 値に設定しているが、BMAC で設定可能な間欠周期は限定さ  
 れているため [3]、最も近い値を利用している。

収集率に関しては IRDT 方式が LPL 方式に対して高い値を  
 示していることが分かる。さらに消費電力に関しても、示した  
 パケット発生率において常に IRDT 方式が消費電力を抑制でき  
 ていることが確認でき、最大消費電力では 33 % から 52 % 削減  
 できており、平均消費電力では 38 % から 54 % 削減できている。  
 またこの時、本来の IRDT (間欠周期 1.0 s) と比較すると、平  
 均消費電力では最大 84 %、最大消費電力では最大 92 % 削減さ

れている。特に、負荷が集中するノードの消費電力を抑制する  
 ことはネットワークの長寿命化の観点から重要であり、制御パ  
 ケット衝突回避の有効性が確認できる。

## 5. まとめ

本稿では、受信端末始動型の IRDT 方式における間欠周期と  
 制御パケットの衝突確率に注目し、制御パケットの衝突確率を  
 最小化する間欠周期を IRDT 方式に対して用いた。また、制御  
 パケットの衝突を抑制する方法としてデータアグリゲーション  
 を IRDT 方式に導入した。以上の改善について計算機上のシ  
 ミュレーションによって調べた結果、提案した間欠周期の設定  
 及びデータアグリゲーションを組み合わせることで、間欠周期  
 を固定した場合と比較して、平均消費電力を最大 84 %、最大  
 消費電力を最大 92 % 削減可能であることを示した。今後の課  
 題としては、特定ノードへの負荷の集中の軽減、電界強度の変  
 動やノードの故障・電力枯渇を考慮した、実環境により近い状  
 況での性能評価を考えている。

謝辞 本研究の一部は、文部科学省科学研究費基盤研究 (A)  
 21240004 によるものである。ここに記して謝意を表す。

## 文 献

- [1] X. Du, Y. Xiao and F. Dai: "Increasing Network Lifetime by Balancing Node Energy Consumption in Heterogeneous Sensor Networks", *Wireless Communications and Mobile Computing*, **8**, 1, pp. 125–136 (2008).
- [2] S. J. Baek and G. de Veciana: "Spatial Energy Balancing through Proactive Multipath Routing in Wireless Multihop Networks", *IEEE/ACM Trans. Networking*, **15**, 1, pp. 93–104 (2007).
- [3] J. Polastre, J. Hill and D. Culler: "Versatile Low Power Media Access for Wireless Sensor Networks", *Proc. ACM Sensys '04*, pp. 95–107 (2004).
- [4] K. Pister and L. Doherty: "TSMP: Time Synchronized Mesh Protocol", *Proc. 20th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'08)* (2008).
- [5] "MICA2", [http://www.xbow.com/products/Product\\_pdf\\_files/Wireless\\_pdf/MICA2\\_Datasheet.pdf](http://www.xbow.com/products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf).
- [6] Y. Wei, H. John and D. Estrin: "An Energy-Efficient MAC Protocol for Wireless Sensor Networks", *Proc. IEEE INFOCOM '02*, pp. 1567–1576 (2002).
- [7] M. Buettner, G. Yee, E. Anderson and R. Han: "X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks", *Proc. ACM Sensys '06*, pp. 307–320 (2006).
- [8] R. Jurdak, P. Baldi and C. V. Lopes: "Adaptive Low Power Listening for Wireless Sensor Networks", *IEEE Trans. Mobile Computing*, **6**, 8, pp. 988–1004 (2007).
- [9] 畠内孝明, 福山良和, 石井美里, 四蔵達之: "メッシュネットワークのためのポーリングによる低消費電力型アクセス方式の提案", *電気学会論文誌. C, 電子・情報・システム部門誌*, **128**, 12, pp. 1761–1766 (2008).
- [10] E. A. Lin, J. M. Rabaey and A. Wolisz: "Power-Efficient Rendez-vous Schemes for Dense Wireless Sensor Networks", *Proc. IEEE ICC '04*, pp. 3769–3776 (2004).
- [11] 小南大智, 菅野正嗣, 村田正幸, 畠内孝明, 福山良和, 四蔵達之: "受信端末始動型間欠動作データ転送方式の性能評価", *信学技法*, IN2008-155, pp. 139–144 (2009).
- [12] I. Chalermek, E. Deborah, G. Ramesh and H. John: "The Impact of Data Aggregation in Wireless Sensor Networks", *Proc. 22nd International Conference on Distributed Computing Systems (ICDCS '02)*, pp. 457–458 (2002).