

無線監視制御システムにおけるノードの間欠 周期制御法の検討

森 郁海[†] 西山 博仁[†] 平岡 精一[†]

無線監視制御システムを商用電源の確保が困難な屋外で使用する場合、ノードを電池で長時間稼働させる必要がある。このため、一般的には、ノードは電池を節約するために、間欠動作を行う。一方で、間欠周期と応答性能はトレードオフであるため、刻々と変化するシステムの状態に応じて間欠周期を制御することが必要である。従来の制御方式では、間欠周期を管理する指示ノードが間欠周期とウェークアップ信号長を算出し、各ノードに指示するため、間欠周期変更に伴う制御トラフィックと所要時間が増加し、省電力効果が低下していた。そこで、指示ノードがシステムの状態に応じた最大イベント通知時間を算出し、各ノードに配布することで、各ノードがその時間に基づき自ノードの間欠周期及び、ウェークアップ信号長を計算する方法を提案する。理論評価の結果、トラフィック量を最大 83.3%、間欠周期変更に伴う所要時間と消費電力を各々最大 81.80%、91.32% 抑制できることを確認した。

A Study of How to Control the Intermittent Cycle of the Node in the Wireless Monitoring Control System

Ikumi Mori[†] Hirihito Nishiyama[†] and Seiichi Hiraoka[†]

Wireless Monitoring Control System is used the outdoors that has no commercial power supply. A node to constitute this system is required to drive with battery for a long time. For this reason, a node works intermittently for power-saving. However, the system should control the intermittent cycle of the node depending on the state of a system changing every moment because the intermittent cycle and the response performance are trade-offs. By the traditional method, an instructions node to manage the intermittent cycle orders an intermittent cycle and a wake-up signal length to each node. So, control traffic and the time required with the change of an intermittent cycle increases, and a power-saving effect deteriorates. By the proposal method, the instructions node calculates an end-to-end delay depending on the state of a system changing, and distributes it to each node. When a node receives the end-to-end delay, it calculates its intermittent cycle and a wake-up signal length. Therefore, control traffic is suppressed, and the time required with the change of an intermittent cycle is reduced. So power-saving effect improves. As a result of theoretical evaluation, we confirmed to reduce up to 83.3% of the control traffic, 81.80% of the time required and 91.32% of the energy with the change of an intermittent cycle.

1. はじめに

無線監視制御システム（以下、システムと表記）は、1組1台の基地局と数十台のセンサノード（以下、ノードと表記）で構成され、各ノードが所定時間内に基地局へ観測データやイベント（例えば、災害予兆）を通知する[1]。このシステムでは、数年に一度しかメンテナンスが行われなため、その期間ノードは電池で稼働し続ける必要がある [2][3]。その際、所定時間内にイベントを通知する、というリアルタイム性を損なってはならない。これらの要件から、本稿では、電池による数年間連続稼働とリアルタイム性の保証を両立するための省電力化手法を検討する。

一般的に、ノードは、電池を節約するために間欠動作（2.1参照）を行う。但し、高い省電力効果を得るために間欠周期を延ばすと、応答性能が悪化する。反対に、応答性能を高めるために、間欠周期を縮めると省電力効果が低下する。このように、消費電力と応答性能はトレードオフの関係にある。上記システムでは、気象や監視対象の状態に応じてシステムに求められる応答性能が変化する。従って、限られた電力資源で、数年間連続稼働とリアルタイム性の保証を両立するには、状態に応じて変化する応答性能に対応して間欠周期を制御することが求められる。既存方式として、丸山氏らの方式[8]があるが、ノード数が増加すると、指示に必要な制御トラフィックがシステムの定める制限時間内に処理できないほど膨大になる、という問題があった。

そこで、間欠周期を制御するトラフィックを最小化し、ノード数が増加しても省電力化とリアルタイム性の保証が両立できる方式を検討した。本方式では、基地局がシステムの要求する最大イベント通知時間を算出し、それに基づく制御データを間欠周期変更の対象となるノードに配布する。そして、ノード自身がその制御データを基に間欠周期の変更制御を行う。この手順により、基地局と配下のノードは、制御データの送信を1度に抑えることができる。また、数十台規模のシステムでも、制限時間内に制御トラフィックを処理することができる。

次章以降の構成は、次のようになる。2章で関連研究を説明する。3章で提案方式の説明を行い、4章で理論評価をする。最後に5章でまとめを述べる。

2. 関連研究

この章では、間欠動作を行うための低消費電力MACプロトコルであるLow Power Listening (LPL) と従来の間欠周期の制御方式を説明する。

[†] 三菱電機株式会社 情報技術総合研究所
Information Technology R&D Center, Mitsubishi Electric Corporation

2.1 Low Power Listening (LPL)

LPLには、時刻非同期型と時刻同期型に分けられる。時刻非同期型LPLの代表的な手法として、Berkeley-MAC (BMAC) [4]やX-MAC[5]がある。時刻同期型LPLには、RFWakeUp多重化方式[6]やWireless Sensor MAC (WiseMAC) [7]がある。

例えば、BMACではノードは一定の間欠周期Tごとに、数ミリ秒間自局のアンテナに電波が来ているかをチェック（キャリアセンス）する検波状態とデータの送受信が不可能なスリープ状態をとる（図1）。間欠動作をしているノードを検波状態からデータ送受信待機状態にする手順は次のようになる。まず、送信元ノードがデータの前に間欠周期T以上の長さのウェークアップ信号を付与し、送信先ノードに送信する。送信先ノードは、検波状態の時にウェークアップ信号を受信すると、データ送受信待機状態に移行し、送信元ノードから続いて送信されるデータを受信する。送信元ノードはデータの受信が完了すると、再びスリープ状態に戻る。

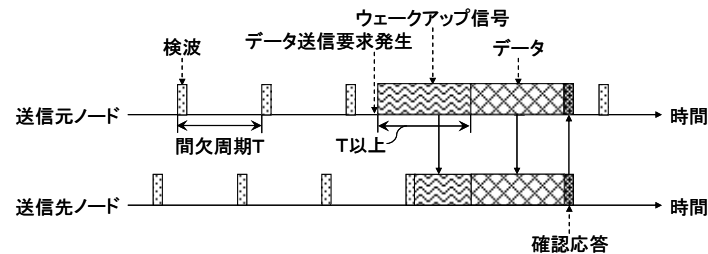


図 1. BMAC の動作

2.2 従来の間欠周期制御手法

ノードが一定周期でしか間欠動作を行わない場合、システムが要求する最も高い応答性能を満たす間欠周期に設定しなければならないため、消費電力が増加する。そこで、丸山氏は、その時どきのシステムの状態に応じて、各ノードの間欠周期とウェークアップ信号長を、指示ノードが制御する方式を提案している[8]。

例えば、図2のようなネットワーク構成であるとする。

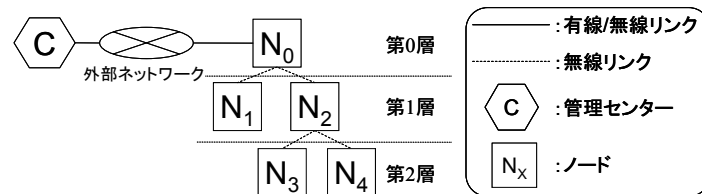


図 2. ネットワーク構成例

N_x ($x=0, \dots, 4$) はノードであり、 N_0 は、 $N_1 \sim N_4$ の監視データを収集して、インターネットや携帯電話回線等の外部ネットワークを通して管理センターCに送信する。ここで、自ノードが接続している N_0 側のノードを親ノード、 N_0 側ではないノードを子ノードと定義する。例えば、 N_2 から見た親ノードは N_0 、子ノードは N_3, N_4 である。

指示ノードが N_0 で、 $N_1 \sim N_4$ の間欠周期を変更する場合の手順を図3に示す。

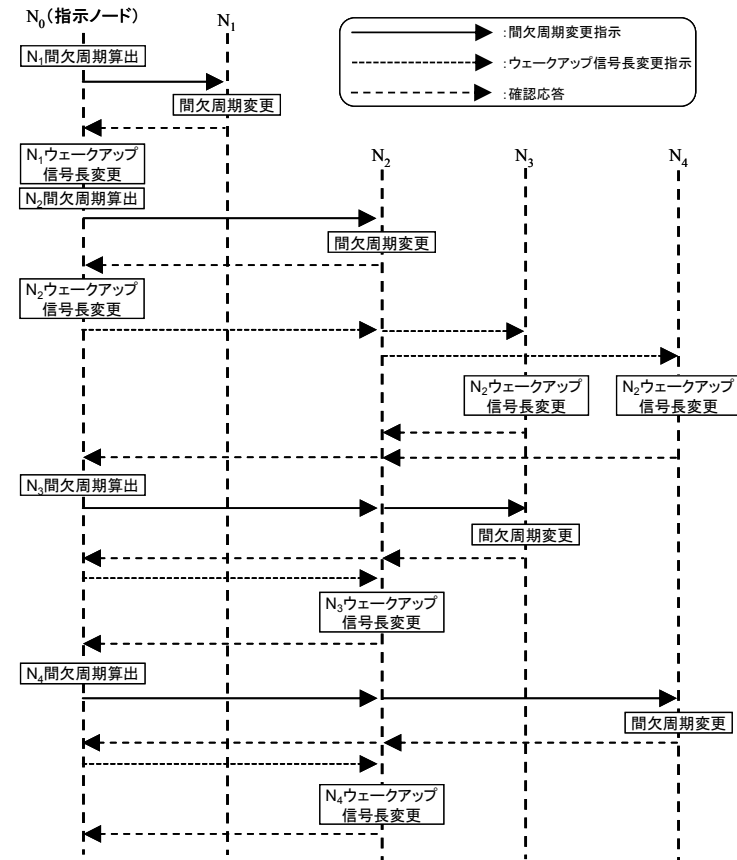


図 3. 間欠周期制御フロー（従来方式）

まず、 N_0 は子ノードである N_1 の間欠周期を計算する。続けて N_0 は、計算した N_1 の間欠周期を、間欠周期変更指示として N_1 に送信する。 N_1 は、間欠周期変更指示を

受信すると、記載された間欠周期に従い自ノードの間欠周期を変更する。N₁は自ノードの間欠周期を変更すると、N₀に確認応答を送信する。N₀はN₁から確認応答を受信すると、N₁に対するウェークアップ信号長を変更する。

次に、N₀はN₂の間欠周期を変更するため、N₂の間欠周期を計算し、N₂へ間欠周期変更指示を送信する。N₂は、間欠周期変更指示に従い間欠周期を変更すると、確認応答をN₀に送信する。N₀はN₂から確認応答を受信すると、N₂に対するウェークアップ信号長を変更する。さらに、N₀はN₂の子ノードであるN₃、N₄に対し、N₂に対するウェークアップ信号長変更指示を送信する。ウェークアップ信号長変更指示はN₂を経由して、N₃、N₄に受信される。N₃、N₄は、ウェークアップ信号長変更指示に従い、N₂に対するウェークアップ信号長を変更し、確認応答をN₀へ送信する。

同様に、N₀はN₃、N₄に間欠周期の変更を指示し、N₃、N₄の親ノードであるN₂にN₃、N₄に対するウェークアップ信号長の変更を指示する。

以上のように、従来方式は、指示ノードが間欠周期とウェークアップ信号長を算出し、各ノードに対して算出結果に基づいた変更を指示する。この方式は、ノードの台数が増加すると、指示に必要な制御トラフィックが急激に増加し、システムの定める制限時間内に処理できないほどになる。

3. 提案方式

従来方式は、ノード数の増加に対して制御トラフィック量が増加する問題があった。そこで、制御するトラフィックを最小化する方式を提案する。具体的には、基地局がシステムの要求する最大イベント通知時間を算出し、それに基づく制御データを間欠周期変更の対象となるノードに1度だけ配布する。そして、ノード自身がその制御データを基にウェークアップ信号長と自身の間欠周期の変更を行う。その際、基地局に近いノードほどデータトラフィックが集中し消費電力が増加するため、消費電力に比例した間欠周期をノードに割り当てるようにした。

以降で、最大イベント通知時間の配布手順と間欠周期の算出手順を説明する。

3.1 最大イベント通知時間の配布手順

提案方式は、間欠周期変更中もデータの通信が行えるように、間欠周期が短くなるように変更する場合と、長くなるように変更する場合で処理手順が異なる。2.2と同様に、図2のネットワーク構成において、指示ノードがN₀で、N₁~N₄の間欠周期が短くなるように変更する場合と、長くなるように変更する場合の手順を説明する。

3.1.1 間欠周期が短くなるように変更する場合

図4は、間欠周期が短くなるように変更する場合の処理手順を示したものである。始めに、N₀はシステムが要求する最大イベント通知時間を算出する。最大イベント通知時間は、N₃あるいはN₄からN₀へデータが届くまでの時間である。次に、N₀は最大イベント通知時間を子ノードであるN₁、N₂に送信する。N₁、N₂は、最大イベント通知時間を受信すると、その時間に基づいて自ノードの間欠周期を算出し、間欠周期を変更する。続けて、N₁、N₂は、確認応答をN₀に送信する。N₂はさらに、N₃、N₄に最大イベント通知時間を転送する。N₀はN₁、N₂から確認応答を受信すると、N₁、N₂に対するウェークアップ信号長を変更する。N₃、N₄は、N₂から最大イベント通知時間を受信すると、最大イベント通知時間に基づいて自ノードの間欠周期を算出し、間欠周期を変更する。さらに、N₃、N₄は、親ノードであるN₂に対するウェークアップ信号長を変更し、確認応答をN₂に送信する。N₂は、N₃、N₄から確認応答を受信すると、N₃、N₄に対するウェークアップ信号長を変更する。

最大イベント通知時間に対する確認応答が返される前に、最大イベント通知時間の送信先ノードへのデータ送信が必要になることがあるが、上記の手順では、確認応答を受信するまでウェークアップ信号長が長いままなので、送信先ノードが間欠周期の変更を完了していなくても確実にウェークアップ信号を受信させることができる。

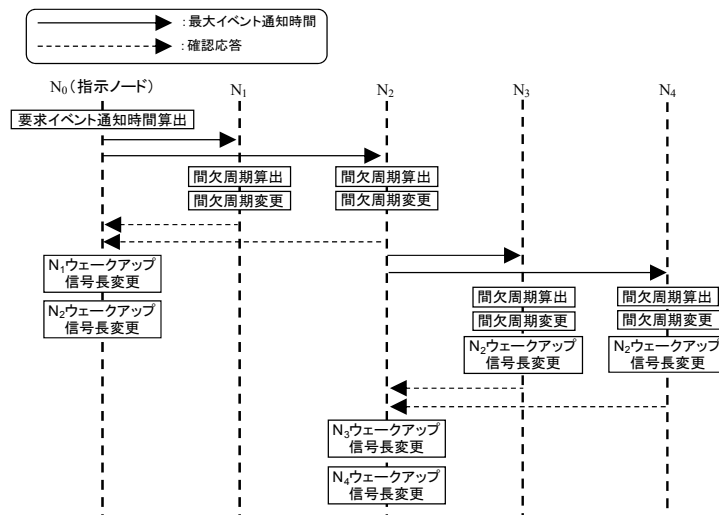


図4. 間欠周期制御フロー（提案方式、間欠周期を短くする場合）

3.1.2 間欠周期が長くなるように変更する場合

図 5は、間欠周期が長くなるように変更する場合の処理手順を示したものである。基本的には、間欠周期が短くなるように変更する場合と同じだが、各ノードは、最大イベント通知時間を子ノードに送信した直後に、子ノードに対するウェークアップ信号長を変更する（図 5灰色網掛）。

こうすることで、最大イベント通知に対する確認応答が返される前に、最大イベント通知時間の送信先ノードへのデータ送信が必要になった場合でも、最大イベント通知時間を送信した直後にウェークアップ信号長を長くするので、確実に送信先ノードにウェークアップ信号を受信させることができる。

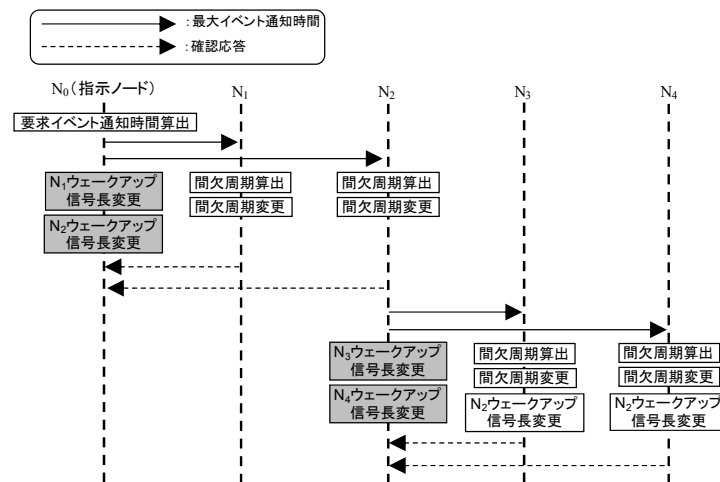


図 5. 間欠周期制御フロー（提案方式、間欠周期を長くする場合）

3.2 間欠周期の算出手順

提案方式は、最大イベント通知時間から自ノード、親ノードおよび子ノードの間欠周期を算出する。図 2において、指示ノードが N_0 で、 $N_0 \sim N_4$ の間欠周期を算出する場合と親ノードと子ノードのウェークアップ信号長を算出する手順を説明する。

3.2.1 自ノードの間欠周期を算出する場合

最大イベント通知時間から自ノードが設定すべき間欠周期を導出する。

今、 $C_{nlevel}^{level}(T)$ を下位ノード[a]数 n_{level} を持つ第 $level$ 層のノードの間欠周期 T での消費電

a) N_x の下位ノードは、 N_x の配下のノードである。図 2において、 N_0 の下位ノードは $N_1 \sim N_4$ である。

力とする（但し、 $C_{nlevel}^{level}(T_x) > C_{nlevel}^{level}(T_{x+1}) \quad s.t. \quad T_x < T_{x+1}, 0 < C_{nlevel}^{level}(T), 0 \leq T \leq T_{max_delay}, T_{max_delay} :=$ 指示ノードが生成した最大イベント通知時間）。 $C_{nlevel}^{level}(T)$ は、下位ノード数に比例する傾向がある。つまり、下位ノードが増加するとデータを中継する頻度が高くなり、消費電力が増加する。

N_0 から順に設定すべき間欠周期を算出する手順を説明する。まず、 N_0 はコストを計算する。コストとは、自ノードから最も深い階層に位置するノードまでの消費電力を足したものである。図 2では、 N_0 から N_3 または N_4 までの消費電力の和である。ノード S からノード D のコストを $Cost(S, D)$ と定義する。

$$Cost(S, D) = \sum_{i=0}^L C_{ni}^i(T_{delay}) \quad (1)$$

ここで、 L は D が位置する階層、 T_{delay} は受信した最大イベント通知時間である。 N_0 において T_{delay} は、 $T_{delay} = T_{max_delay}$ である。ツリールーティングの場合、 N_0 の下位ノードのネットワーク構成（ノードの接続関係）を容易に把握できるので、 N_0 は $Cost(N_0, N_{i \in \{1, 2, 3, 4\}})$ を算出可能である。

N_0 は、コストが最大となる経路 $\{N\}$ を見つける。

$$Cost(N_0, N_{max}) = \max(Cost(N_0, N_1), Cost(N_0, N_2), Cost(N_0, N_3), Cost(N_0, N_4)) \quad (2)$$

図 2では、 $Cost(N_0, N_{max}) = Cost(N_0, N_4)$ で、そのときの経路は $\{N_0, N_2, N_4\}$ である。次に、間欠周期を設定するために、最大と最小の $C_{nlevel}^{level}(T)$ を求める。 $n_0 = 4, n_1 = 2, n_2 = 0$ なので、数式(3)のようになる。

$$\begin{aligned} C_{nmax}^{level_max}(T) &= \max(C_4^0(T_{delay}), C_2^1(T_{delay}), C_0^2(T_{delay})) \\ C_{nmin}^{level_min}(T) &= \min(C_4^0(T_{delay}), C_2^1(T_{delay}), C_0^2(T_{delay})) \end{aligned} \quad (3)$$

$C_{nlevel}^{level}(T)$ が下位ノード数に比例する場合、 $C_{nmax}^{level_max}(T) = C_4^0(T)$ 、 $C_{nmin}^{level_min}(T) = C_0^2(T)$ となる。次に、区間 $[0, T_{threshold}]$ において、 $C_{nmax}^{level_max}(T)$ と $C_{nmin}^{level_min}(T)$ の消費電力が平衡するような間欠周期 $T_{balance_max}$ 、 $T_{balance_min}$ を求める。但し、 $T_{threshold}$ の初期値は T_{delay} である。

$$\begin{aligned} \exists T_{balance_max}, T_{balance_min} \in [0, T_{threshold}] \\ s.t. \quad C_{nmax}^{level_max}(T_{balance_max}) \cong C_{nmin}^{level_min}(T_{balance_min}), T_{balance_min} = T_{threshold} - T_{balance_max} \end{aligned} \quad (4)$$

さらに、 $C_{ni}^i(T_{balance_i}) \cong C_{max}(T_{balance_max}), C_{min}(T_{balance_min})$ を満たす $N_i \in \{N\}$ の間欠周期 $T_{balance_i}$ （但し、 $0 \leq i \leq L, i \neq level_max, level_min$ ）は、区間 $[T_{balance_min}, T_{balance_max}]$ 上に存在するはずである（図 2では、 N_2 の間欠周期を計算していることに相当）。

$$\begin{aligned} \exists T_{balance_i} \in [T_{balance_min}, T_{balance_max}] \\ s.t. \quad C_{ni}^i(T_{balance_i}) \cong C_{nmax}^{level_max}(T_{balance_max}) \cong C_{nmin}^{level_min}(T_{balance_min}) \end{aligned} \quad (5)$$

最後に、求めた間欠周期の和が T_{delay} 内であるかチェックする（図 6）。

$$\sum_{i=0}^L T_{balance_i} \leq T_{delay} \quad (6)$$

数式(6)を満たす場合、このときの間欠周期 $T_{balance_i}$ ($0 \leq i \leq L$) が設定すべき各層の間欠周期である。満たさない場合は、 $T_{threshold} = T_{threshold} - \varepsilon$ ($0 \leq \varepsilon \leq T_{threshold}$) として数式(4)からやり直し数式(6)を満たすまで続ける。

最後に、 N_0 は間欠周期が決定すると、 T_{delay} を $T_{delay} = T_{delay} - T_{balance_N_0}$ と更新し、最大イベント通知時間 T_{delay} および、 $T_{balance_N_0}$ を子ノードである N_1, N_2 に送信する。

N_1, N_2 は、 T_{delay} および、 $T_{balance_N_0}$ を受信すると、 $T_{delay} (= T_{delay} - T_{balance_N_0})$ を基に 数式(1)~(6)を処理し、自ノードの間欠周期を算出する。また、 N_3, N_4 も同様の手順で間欠周期を算出できる。

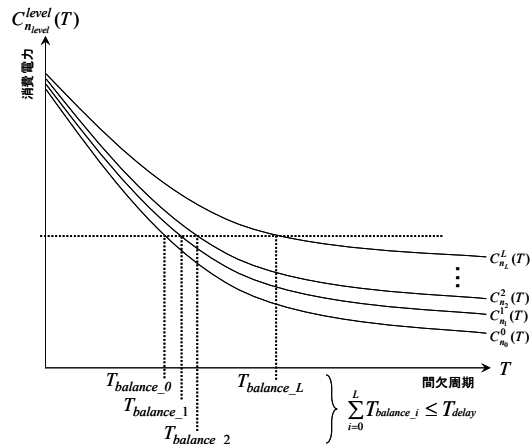


図 6. 間欠周期の決定

3.2.2 ウェークアップ信号長を算出する場合

親ノード及び子ノードに対するウェークアップ信号長の算出に必要な間欠周期を最大イベント通知時間から導出する。

今、 N_0 がウェークアップ信号長を算出する場合を考える。 N_0 の親ノードは存在しないので、子ノードである N_1, N_2 の間欠周期が必要となる。 N_0 は、自ノードの間欠周期を導出する過程 (3.2の数式(2)~(6)) で N_2 の間欠周期を算出しているのので、 N_1 の間欠周期を別途計算する。

N_0 は、 N_1 へ送信する最大イベント通知時間 $T_{delay} (= T_{delay} - T_{balance_N_0})$ から、 N_1 が3.2の数式(2)~(6)で算出する間欠周期を求めることができる。よって、 N_0 は、 N_1, N_2 の間欠周期を算出することができ、ウェークアップ信号長を算出できる。

次に、 N_2 がウェークアップ信号長を算出する手順を説明する。 N_2 は、親ノードである N_0 、子ノードである N_3, N_4 の間欠周期が必要となる。 N_0 の間欠周期は、最大イベント通知時間とともに送信されてくるため、計算不要である。 N_2 は、 N_3, N_4 のいずれかの間欠周期を自ノードの間欠周期を導出する過程で算出している。従って、算出していないノードの間欠周期を別途計算する必要があるが、 N_0 と同様に子ノードへ送信する最大イベント通知時間から間欠周期を求めることができる。

表 1 に各ノード最大イベント通知時間と間欠周期の設定値の関係を示す。

表 1. 各ノードの最大イベント通知時間の構成と自ノードの間欠周期設定値

間欠周期 算出ノード	最大イベント通知時間 (親ノード→自ノード)		自ノード設定値				最大イベント通知時間 (自ノード→子ノード)	
	T_{delay}	$T_{balance}$	親	自	子		T_{delay}	$T_{balance}$
N_0	T_{max_delay}			$T_{balance_N_0}$	$T_{balance_N_1}$	$T_{balance_N_2}$	$T_{max_delay} - T_{balance_N_0}$	$T_{balance_N_0}$
N_1	$T_{max_delay} - T_{balance_N_0}$	$T_{balance_N_0}$	$T_{balance_N_0}$	$T_{balance_N_1}$				
N_2	$T_{max_delay} - T_{balance_N_0}$	$T_{balance_N_0}$	$T_{balance_N_0}$	$T_{balance_N_2}$	$T_{balance_N_3}$	$T_{balance_N_4}$	$T_{max_delay} - \sum_{i=0,2} T_{balance_N_i}$	$T_{balance_N_2}$
N_3	$T_{max_delay} - \sum_{i=0,2} T_{balance_N_i}$	$T_{balance_N_2}$	$T_{balance_N_2}$	$T_{balance_N_3}$				
N_4	$T_{max_delay} - \sum_{i=0,2} T_{balance_N_i}$	$T_{balance_N_2}$	$T_{balance_N_2}$	$T_{balance_N_4}$				

4. 理論評価

提案方式の効果を知るために、提案方式と従来方式の理論評価を行う。具体的には、制御トラフィック、間欠周期変更に伴う所要時間、省電力効果 (消費電力) を従来方式と比較する。

4.1 評価条件

表 2 に評価条件の一覧を示す。ネットワーク構成は、図 7 とする。総ノード数は、制御トラフィック評価時は 20~50 (中継ノード 1 台あたり 1 台の中継ノード ($R_{max}=1$) と 9 台の子ノード ($C_{max}=9$) が接続され、最大階層が 2~5 ($2 \leq L_{max} \leq 5$) で、その他の評価時は、50 ($R_{max}=1, C_{max}=9, L_{max}=5$) である。間欠動作には、RFWakeUp 多重化方式を使用する [6]。間欠周期変更のシナリオは、図 8 とする。P0 (フェーズ 0) ~ P3 の最大イベント通知時間 ($= T_{max_delay}$) や各フェーズに移行 (間欠周期を変更) する時間は、予めシステムで定めるものとし、5, 10, 15 時間経過時点でフェーズ移行が発生する。このとき、指示ノードは N_0 である。但し、 N_0 は商用電源が利用可能で、常時起動するものとする。

表 2. 評価条件

トポロジ	R_{max}	C_{max}	L_{max}		指示ノード				
	1	9	変更所要時間, 省電力効果評価時	5	N_0				
			制御トラフィック評価時	2~5					
シミュレーション シナリオ	変更対象ノード		データ量[byte]	データサンプリング 回数(回/Hour)	最大イベント通知時間[秒]				
	変更所要時間, 省電力効果 評価時	N_0 以外	センシングデータ	60	P0	1/24	P0	60	20.5
			応答確認	20	P1	4	P1	10	1.5
	制御トラフィック 評価時	全てのノード	間欠周期変更指示	60					
			ウェークアップ信号変更指示	60	P2	8	P2	5	1
		最大イベント通知時間	60	P3	16	P3	2.5	0.5	
ノードスペック	伝送速度[kbps]	SLEEP電力[mW]	送信電力[mW]	受信電力[mW]	時刻同期周期	タイムスロット数			
	250	0.05	102.75	108	450[秒/回]	8 (62.5[ms/スロット])			

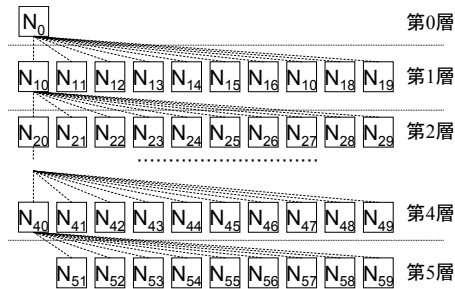


図 7. ネットワーク構成

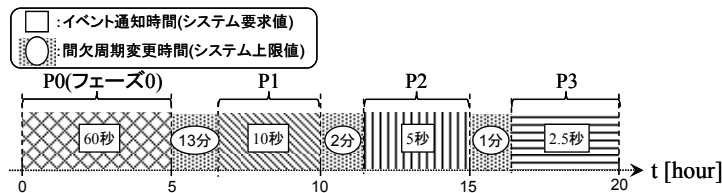


図 8. 間欠周期の変更タイミング

4.2 制御トラフィックの抑制効果

従来方式の制御トラフィック量は、間欠周期変更指示、プリアンブル長変更指示、確認応答の総数であり、数式(7)のようになる。

$$\sum_{i=0}^{L_{max}-1} \{R_{max}^i \cdot Change_{Router}(i)\} + \sum_{i=1}^{L_{max}} \{R_{max}^{i-1} \cdot Change_{Child}(i)\}$$

$$\text{但し, } Change_{Router}(i) = \begin{cases} i = L_{max} - 1, & 4i + 2C_{max} \\ i \neq L_{max} - 1, & 4i + 2(C_{max} + R_{max}) \end{cases}, \quad Change_{Child}(i) = 4i - 2 \quad (7)$$

提案方式の制御トラフィック量は、最大イベント通知時間と確認応答の総数であり、数式(8)のようになる。

$$\sum_{i=0}^{L_{max}-1} \begin{cases} i = L_{max} - 1, & 2R_{max}^i (C_{max}) \\ i \neq L_{max} - 1 & 2R_{max}^i (R_{max} + C_{max}) \end{cases} = \begin{cases} R_{max} = 1 & (1 + C_{max})(L_{max} - 1) + C_{max} \\ R_{max} \neq 1 & 2\{R_{max} + C_{max} - R_{max}^{L_{max}}(1 + C_{max})\} / (1 - R_{max}) \end{cases} \quad (8)$$

図 9は、制御トラフィック量を図示したものである。ノード数が 50 の場合で、従来方式に対して、最大 83.3%抑制する。ノード数に対して抑制効果が比例することが分かる。

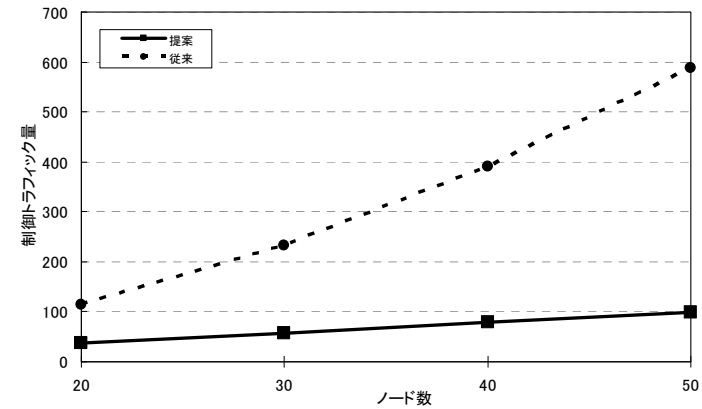


図 9. 制御トラフィック量

4.3 間欠周期変更に伴う所要時間の抑制効果

間欠周期変更に伴う所要時間は、図 8におけるフェーズの移行時間に相当する。始めに、従来方式でのフェーズの移行時間の平均値を求める。間欠周期を短く変更する場合は、浅い階層の中継ノードから順次変更した後、子ノードを変更する(数式(9))。間欠周期を長く変更する場合は、先に子ノードを変更後、深い階層の中継ノードから順次変更していく(数式(10))。

$$Time_{Traditional}^{Narrow} := Time_{Router}(0) + \dots + Time_{Router}(L_{max} - 1) + Time_{Child}(1) + \dots + Time_{Child}(L_{max}) \quad (9)$$

$$Time_{Traditional}^{Widen} := Time_{Child}(1) + \dots + Time_{Child}(L_{max}) + Time_{Router}(L_{max} - 1) + \dots + Time_{Router}(0) \quad (10)$$

このとき、 $Time_{Router}(i)$ は、次のように定義する。 $i=0$ のとき、数式(11),(12)を実行する。但し、 $IC_{Router}(i)$ は、第 i 層の中継ノードの間欠周期であり、 $IC_{Router}^{New}(i)$ は、第 i 層の中継ノードが新しく設定する間欠周期である。

$$IC_{Router}(0) := IC_{Router}^{New}(0) \quad (11)$$

$$Time_{Router}(i) := R_{max} \cdot IC_{Router}(1)/2 + C_{max} \cdot IC_{Child}(1)/2 + R_{max} \cdot IC_{Router}(0)/2 + C_{max} \cdot IC_{Child}(0)/2 \quad (12)$$

$i=L_{max}-1$ のとき、数式(13)~(15)を実行する。 $IC_{Child}(i)$ は、第 i 層の中継ノード以外のノード（リーフノード）の間欠周期であり、 $IC_{Child}^{New}(i)$ は、第 i 層のリーフノードが新しく設定する間欠周期である。

$$Time_{Router}(i) := \sum_{j=1}^i \{IC_{Router}(j)/2\} \quad (13)$$

$$IC_{Router}(i) := IC_{Router}^{New}(i) \quad (14)$$

$$Time_{Router}(i) := + 2 \sum_{j=0}^{i-1} \{IC_{Router}(j)/2\} + C_{max} \cdot IC_{Child}(i)/2 + \sum_{j=1}^{i+1} \begin{cases} j=i+1, & C_{max} \cdot IC_{Child}(j)/2 \\ j \neq i+1, & IC_{Router}(j) \end{cases} \quad (15)$$

$i \neq 0, L_{max}-1$ のとき、数式(16)~(18)を実行する。

$$Time_{Router}(i) := \sum_{j=1}^i \{IC_{Router}(j)/2\} \quad (16)$$

$$IC_{Router}(i) := IC_{Router}^{New}(i) \quad (17)$$

$$Time_{Router}(i) := + 2 \sum_{j=0}^{i-1} \{IC_{Router}(j)/2\} + (C_{max} + R_{max}) \cdot IC_{Child}(i)/2 + \sum_{j=1}^{i+1} \begin{cases} j=i+1, & \{C_{max} \cdot IC_{Child}(j) + R_{max} \cdot IC_{Router}(j)\}/2 \\ j \neq i+1, & IC_{Router}(j)/2 \end{cases} \quad (18)$$

また、 $Time_{Child}(i)$ は、次のように定義する。 $i=1$ のとき、数式(19)~(20)を実行する。

$$IC_{Child}(i) := IC_{Child}^{New}(i) \quad (19)$$

$$Time_{Child}(i) := IC_{Router}(0)/2 \quad (20)$$

$i \neq 1$ のとき、数式(21)~(23)を実行する。

$$Time_{Child}(i) := \sum_{j=1}^{i-1} \{IC_{Router}(j)/2\} + IC_{Child}(i)/2 + \sum_{j=0}^{i-1} \{IC_{Router}(j)/2\} \quad (21)$$

$$IC_{Child}(i) := IC_{Child}^{New}(i) \quad (22)$$

$$Time_{Child}(i) := + \sum_{j=1}^{i-1} \{IC_{Router}(j)/2\} + \sum_{j=0}^{i-2} IC_{Router}(j)/2 \quad (23)$$

次に、提案方式のフェーズの移行時間の平均値を求めると、数式(24)のようになる。

$$Time_{Proposal} := Time(1) + \dots + Time(L_{max}) \quad (24)$$

このとき、 $Time(i)$ は、数式(25)~(27)を実行する。

$$Time(i) := \sum_{i=1}^{L_{max}} \begin{cases} i=L_{max}, & C_{max} \cdot IC_{Child}(i)/2 + C_{max} \cdot IC_{Router}(i-1)/2 \\ i \neq L_{max}, & R_{max} \cdot IC_{Router}(i)/2 + C_{max} \cdot IC_{Child}(i)/2 + (R_{max} + C_{max}) \cdot (IC_{Router}(i-1)/2) \end{cases} \quad (25)$$

$$i \neq L_{max}, \quad IC_{Router}(i) := IC_{Router}^{New}(i) \quad (26)$$

$$IC_{Child}(i) := IC_{Child}^{New}(i) \quad (27)$$

表 3は、4.1条件下における従来方式と提案方式のフェーズ移行時間の平均値である。各ノードの間欠周期は、3.2で述べた算出法を用いた。

従来方式では、システムが要求する変更時間内に変更を完了することができない（表 3灰色網掛上部）。従って、要求変更時間内に変更を終えるには、変更前の間欠周期を短く設定しなければならない（表 3灰色網掛）。

表 3. フェーズ移行時間、フェーズ移行消費電力と抑制効果

	フェーズ移行時間[分]			N ₁₀ のフェーズ移行消費電力[mW]
	P0→P1	P1→P2	P2→P3	
提案方式	60秒→10秒	10秒→5秒	5秒→2.5秒	0.0012
	13.32	1.66	0.78	
従来方式	60秒→10秒	10秒→5秒	5秒→2.5秒	0.0140
	43.75	8.53	4.26	
	20.5秒→1.5秒	1.5秒→1秒	1秒→0.5秒	
	13.26	1.68	0.91	
抑制効果	69.55%	80.51%	81.80%	91.32%

一方、提案方式は要求変更時間内に変更を終えることができるので、間欠周期を短くする必要が無い。間欠周期を短くしない時の従来方式に対する抑制効果は、最大81.80%である。また、システム寿命のボトルネックとなるN₁₀のフェーズ移行時の消費電力も、従来方式に対して91.32%抑制する。

4.4 省電力効果

省電力効果（消費電力の削減率）は、ネットワーク寿命のボトルネックとなるノード（図 7では、N₁₀）の消費電力が、間欠周期を制御しない場合と比較してどれだけ削減できたかを示したものである。

数式(28)は従来方式, 数式(29)は提案方式のノードの総消費電力である. E_{rx} , E_{tx} は観測データや確認応答を受信, 送信するための電力である. E_{mis_rx} は, 誤受信による消費電力である[6]. $E_{time_sync_rx}$, $E_{time_sync_tx}$ は, 時刻同期のための送受信に伴う電力である. E_{new_entry} は, ノードの新規参入処理のための電力である. E_{data} は, データを観測するための電力であるが, 本稿では, この電力は別途確保するものとし, 0とする. $E_{detection}$ は, 検波電力である. E_{sleep} は, スリープ状態の消費電力である. $E_{change}^{Traditional}$ は従来方式, $E_{change}^{Proposal}$ は提案方式の間欠周期変更にかかる電力である.

$$E_{Traditional} := E_{rx} + E_{tx} + E_{mis_rx} + E_{time_sync_rx} + E_{time_sync_tx} + E_{new_entry} + E_{data} + E_{detection} + E_{sleep} + E_{change}^{Traditional} \quad (28)$$

$$E_{Proposal} := E_{rx} + E_{tx} + E_{mis_rx} + E_{time_sync_rx} + E_{time_sync_tx} + E_{new_entry} + E_{data} + E_{detection} + E_{sleep} + E_{change}^{Proposal} \quad (29)$$

図10は, 4.1条件下における従来方式と提案方式および, 間欠周期を制御しない(各ノード500[ms]固定, $500 \times L_{max}(5) = 2.5$ 秒)場合の N_{10} の消費電力を图示したものである. 矢印は, 間欠周期を制御しない場合の消費電力を100%とした時の従来方式と提案方式の増減の割合(%)である.

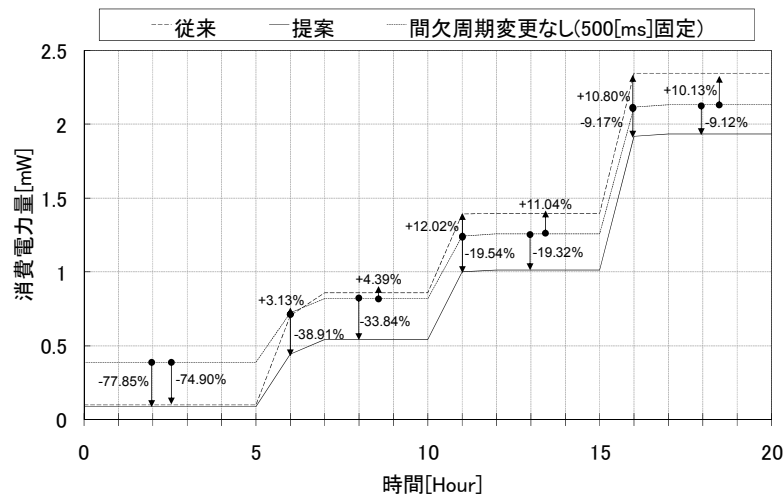


図10. 省電力効果

P0においては, 従来方式, 提案方式とも間欠周期を制御しない場合に対して大幅に削減できている(提案: -77.85%, 従来: -74.90%). しかし, 従来方式は, P1~P3において, +3.13~12.02%間欠周期を制御しない場合より増加している. これは, 4.3で述べたように, 変更前の間欠周期を短く設定しなければならないためである. また, 4.2

より従来方式はトラフィック量が大きいため, フェーズ移行時の消費電力が著しく増加する(表3: フェーズ移行消費電力参照). 一方, 提案方式は, トラフィック量が小さいので間欠周期を制御しない場合の消費電力を超えることがない.

5. おわりに

本稿では, ノード数の増加に対応した, 間欠周期の制御トラフィックを最小化する方式を検討した. 本方式は, 基地局がシステムの要求する最大イベント通知時間を算出し, それに基づく制御データを間欠周期変更の対象となるノードに1度だけ配布する. そして, ノード自身がその制御データを基にウェイクアップ信号長と自身の間欠周期の変更を行う. その際, 基地局に近いノードほどデータトラフィックが集中し消費電力が増加するため, 消費電力に比例した間欠周期をノードに割り当てた.

理論評価の結果, トラフィック量を最大83.3%, 間欠周期変更に伴う所要時間を最大81.80%, 間欠周期変更に伴う消費電力を最大91.32%抑制できることを確認した. これらのことから, 提案方式は省電力化を維持したまま, 応答性能の確保を得ることが出来ると考えられる.

今後は, メッシュネットワークにも対応できるような方式を拡張し, 更なる高信頼化を目指す.

参考文献

- 1) N. Xu : A Survey of Sensor Network Applications, Survey Paper for CS694a, Computer Science Department, University of Southern California (2003).
- 2) L. Selavo, A. Wood, Q. Cao, T. Sookoor, H. Liu, A. Srinivasan, Y. Wu, W. Kang, J. Stankovic, D. Young, and J. Porter : LUSTER: Wireless sensor network for environmental research, In Proc. of the 5th ACM Conf. on Embedded Networked Sensor Systems (SenSys), Sydney, Australia, pp103-116 (2007).
- 3) R. Szwedczyk, J. Polastre, A. Mainwaring, and D. Culler : Lessons from a sensor network expedition, In Proc. of the 1st European Workshop on Wireless Sensor Networks (EWSN2004), Berlin, Germany, 307-322 (2004).
- 4) J. Polastre, J. Hill and D. Culler: Versatile low power media access for wireless sensor networks, Proceedings of the 2nd international conference on Embedded networked sensor systems, Baltimore, MD, USA(2004).
- 5) M. Buettner, G.V. Yee, E. Anderson and R. Han: XMAC: a short preamble MAC protocol for duty-cycled wireless sensor networks, ACM SenSys, pp.307-320(2006).
- 6) 西山博仁, 長川大介, 平岡精一: センサーネットワークにおける低消費電力通信方式, 電子情報通信学会大会講演論文集, Vol.2008, pp.101(2008).
- 7) A. El-Hoiyi, J.-D. Decotignie, and J. Hernandez : Low power MAC protocols for infrastructure wireless sensor networks. In Proceedings of the Fifth European Wireless Conference, (2004).
- 8) 丸山隆夫, 新井正伸: 特開 2005-217548(無線通信方式, 無線通信システム, 及び無線端末), 公開特許公報, 日本国特許庁(2005).