

## センサノードの実行履歴を用いた負荷予測に基づく DVS と DMS による協調型省電力手法

濱千代 貴大<sup>†</sup> 金丸 達雄<sup>††</sup> 横田 裕介<sup>†</sup> 大久保 英嗣<sup>†</sup>

<sup>†</sup>立命館大学情報理工学部 <sup>††</sup>立命館大学大学院理工学研究科

### 1 はじめに

センサノードの省電力化手法として、DVS (Dynamic Voltage Scaling) および DMS (Dynamic Modulation Scaling) の利用が考えられる。

DVS は、プロセッサの動作電圧を下げることで、省電力化を実現する手法である。これは、プロセッサが動作可能な最大周波数は動作電圧と比例し、消費電力は動作電圧の 2 乗に比例する性質 [1] を利用している。ただし、動作電圧の減少に伴い動作周波数も減少するため、処理速度が減少し、処理時間が増加する。このため、DVS は、アイドル時間を利用することで省電力化を実現する手法であるといえる。

DMS は、シンボル毎のビット数を下げることで、省電力化を行う手法である。これは、シンボル毎のビット数を下げることで、ビットあたりの消費電力が減少する性質 [2] を利用している。ただし、シンボル毎のビット数を下げるため、通信速度は低下し、通信時間が増加する。このため、DMS も DVS と同様、アイドル時間を利用することで省電力化を実現する手法であるといえる。

プロセッサの省電力化を行う DVS と無線装置の省電力化を行う DMS を共に用いることで、これらを単独で用いるより電力効率のよい処理を実現できる。しかし、DVS と DMS が共にアイドル時間を利用して省電力化を実現する手法であるため、両者を協調させて制御する必要がある。

本稿では、ノードの動作状況の詳細なログに基づいて DVS と DMS の効果的な協調動作を実現する手法を提案する。

### 2 DVS と DMS による協調型省電力手法

本研究では、スロットと呼ばれる時間枠を用いたスロットベーススケジューリング方式 [3] を利用することを想定する。スロットベーススケジューリングは、通信が発生するスロット以外では無線を休止して省電力化を実現する手法である。省電力化を目的とする本手法とスロットベーススケジューリングを同時に利用することで、更なる省電力化が実現できる。

提案手法は、ロギング、負荷予測、DVS と DMS の実施の 3 つの処理から構成される。ロギングは、負荷予測で用いるスレッドおよび通信のログを取る。負荷予測は、DVS と DMS で利用可能なアイドル時間を算出する。DVS と DMS の実施は、省電力化効果のよい動作周波数およびシンボル毎のビット数を求め、DVS と DMS を実施する。

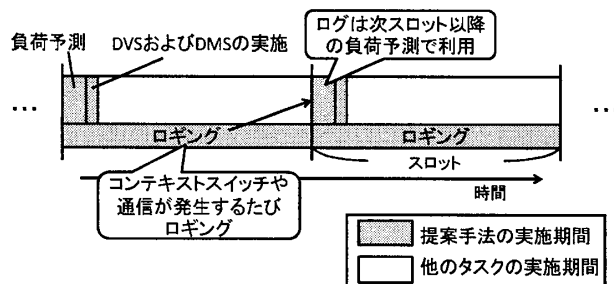


図 1 スロット内での提案手法の処理期間

スロット内での提案手法の処理期間を図 1 に示す。まず、スロットの開始時にプロセッサおよび無線装置がどの程度利用されるかを予測する。次に、DVS と DMS で利用可能なアイドル時間を算出し、DVS と DMS を実施する。また、コンテキストスイッチや通信が発生するごとに、ロギング処理を行う。ロギングで取得するログは、各スレッドと無線処理の起動時刻および実行時間である。以下、負荷予測と DVS と DMS の実施の詳細について述べる。

**負荷予測** 負荷予測では、現在のスロットでのプロセッサと無線装置の負荷を予測する。プロセッサの負荷として、合計実行時間  $T_p$  を求める。また、無線装置の負荷として、合計通信時間  $T_c$  を求める。負荷予測手順を次に示す。

まず、現在のスロットで起動するスレッドを予測する。ここでは、各スレッドの周期と前回の起動時刻を基に予測を行う。スレッドの周期は、 $(N$  回目の起動時刻)  $- (N-1$  回目の起動時刻) の過去数回分の平均によって求める。前回の起動時刻に周期を足した時刻が現在のスロットの時間枠に含まれるとき、そのスレッドは現在のスロットで起動すると判断する。

次に、起動が予測されるスレッドの実行時間の和により、 $T_p$  を求める。ここで、 $T_p$  を最大周波数  $f_{MAX}$  での実行時間として (1) 式で求める。これは、スレッド  $i$  の実行時間は、 $i$  が動作していた時の動作周波数  $f_i$  に依存するため、同一の動作周波数に換算して実行時間を求める必要があることによる。また、 $T_c$  も同様にシンボル毎のビット数の最大値  $b_{MAX}$  での通信時間として (2) 式で求める。

$$T_p = \sum_{i=1}^{\text{スレッド数}} \left( \text{実行時間}_i \cdot \frac{f_i}{f_{MAX}} \right) \quad (1)$$

$$T_c = \sum_{i=1}^{\text{通信数}} \left( \text{通信時間}_i \cdot \frac{b_i}{b_{MAX}} \right) \quad (2)$$

A Cooperative Power-Saving Technique using DVS and DMS based on Overload Prediction in Sensor Networks

Takahiro Hamachiyo<sup>†</sup>, Tatsuo Kanamaru<sup>††</sup>, Yusuke Yokota<sup>†</sup> and Eiji Okubo<sup>†</sup>

<sup>†</sup>College of Information Science and Engineering, Ritsumeikan Univ.

<sup>††</sup>Graduate School of Science and Engineering, Ritsumeikan Univ.

**DVS と DMS の実施** DVS と DMS の実施では、現在のスロットでの動作周波数およびシンボル毎のビット数を決定し、DVS と DMS を実施する。動作周波数およびシンボル毎のビット数の決定手順を次に示す。

1. 任意の動作周波数  $f$  とシンボル毎のビット数  $b$  の組み合わせを決定する。
2. 手順 1 で決定した  $f, b$  で DVS と DMS を適応したときに、デッドラインミスが発生するかをスロット時間  $S$  (スロットの開始から終了までの時間) を用いて下記 (3) 式で判断する。(3) 式が成立しない(デッドラインミスが発生する)場合、手順 1 に戻り、別の  $f, b$  の組み合わせを決定する。

$$T_p \frac{f_{MAX}}{f} + T_c \frac{b_{MAX}}{b} \leq S \quad (3)$$

3.  $f, b$  における単位時間あたりのプロセッサの消費電力  $E_p$ , 無線装置の消費電力  $E_c$  を求める。これは、プロセッサや無線装置の消費電力モデルに従い決定する。
4. 下記 (4) 式より現在のスロットで予測されるノードの消費電力  $E_{tot}$  を求める。

$$E_{tot} = E_p T_p \frac{f_{MAX}}{f} + E_c T_c \frac{b_{MAX}}{b} \quad (4)$$

5. 1 から 4 の手順を利用可能なすべてのシンボル毎のビット数に実施し、最小の  $E_{tot}$  を算出するシンボル毎のビット数と動作周波数の組み合わせを求める。

### 3 評価

本手法と DVS, DMS のシミュレーションを MICAz 上で行い、消費電力を比較した。実機を用いてシミュレーションを行ったため、ハードウェア割り込みや OS のオーバヘッドに関して、実際にセンサノードを運用するときと同様の環境で評価を行うことができた。また、MICAz では DMS を実施することができないため、シンボル毎のビット数に応じて処理時間が変わる仮想的な通信ライブラリを用いた。シミュレーションの結果、本手法は省電力化手法を用いない場合と比較し、約 30% の省電力化を実現できた。

本評価では、プロセッサと無線装置の電力消費比(以下、電力消費比)を変化させてシミュレーションを行う。これは、電力消費比によっては、DVS と DMS のどちらかのみを実施することで十分な省電力化を実現できることによる。例えば、プロセッサの消費電力が無線装置の消費電力より極めて高い場合、DVS のみを実施すればよい。

**シミュレーション環境** 本評価では、周期が 1,000msec のスロットに表 1 に示すスレッドを動作させた。プロセッサの消費電力モデルは、文献 [4] を参考にした。また、この式に用いるパラメータは、CRUSOR プロセッサの実測値 [5] を元に算出した。無線装置の消費電力モデルとパラメータは、文献 [2] を参考にした。

表 1 動作スレッド

種類	実行時間	周期
演算	300msec	1 スロット
通信	300msec	2 スロット

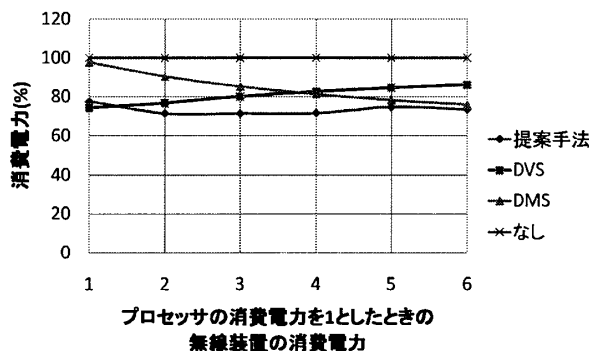


図 2 電力消費比を変化させたときの消費電力

**シミュレーション結果** シミュレーション結果を図 2 に示す。図 2 から提案手法を用いた場合、省電力化手法を用いない場合と比べて、消費電力は約 70% であった。また、DVS は無線装置の消費電力が高い場合に省電力の効果が低く、DMS はプロセッサの消費電力が高い場合に省電力化の効果が低い。しかし、提案手法では電力消費比に依存せず省電力の効果があつた。このため、DVS と DMS を協調させて実施することにより、ハードウェア構成に依存しない省電力化を実現できる。

### 4 おわりに

本稿では、センサノードの実行履歴を用いた負荷予測に基づく DVS と DMS による協調型省電力手法について述べた。今後は、実際に DVS と DMS を実現可能なハードウェアを利用した評価を行う予定である。

### 参考文献

- [1] Rex Min, Travis Furrer, Anantha Chandrakasan, Dynamic Voltage Scaling Techniques for Distributed Microsensor Networks, IEEE VLSI Workshop, pp.43-46, April 2000
- [2] Curt Schurgers, Olivier Aberthorne, Mani B. Srivastava, Modulation Scaling for Energy Aware Communication Systems, ISLPED'01, pp.96-99, August 2001
- [3] 松尾 英治, 鈴木 和久, 横田 裕介, 大久保 英嗣, センサノード向け OS における消費電力の低減のための協調型タスクスケジューリング, 情報処理学会研究報告 2008-ARC-176/2008-EMB-7, pp.65-70
- [4] T. Burd and R. W. Brodersen, Energy efficient CMOS microprocessor design, Proc. 28th Hawaii Int '1 Conf. on System Sciences, Vol.1, pp. 288-297, Jan 1995.
- [5] 後藤 弘茂, 後藤 弘茂の Weekly 海外ニュース, <http://pc.watch.impress.co.jp/docs/article/20000126/kai gai01.htm>