無線センサネットワークにおける天井照明の光度制御を用いたアプリケーションの動的変更

堂面 拓也
Takuya DOMEN

1　はじめに

センサノード上のアプリケーションは，センサネットワークで実現するシステムそのものの変更やシステムの要求によるアプリケーション内部のパラメタ変更など，センサネットワークの運用目的や運用状況によって変更する必要がある．センサノード上のアプリケーションを動的に変更するには，センサノード上にミドルウェアを導入し，無線機能を用いてアプリケーションデータを送信することで実現する．しかし，無線機能によるデータ送信はパケットロスの発生確率を増加させる．

本研究では，コンピュータ制御による調光可能な天井照明を用いてデータ送信を行い，室内に設置したセンサノードのアプリケーションを動的に変更するミドルウェア「dLitup」を提案する．実機を用いて dLitup を実装し，基本的性能を評価する．

2　提案ミドルウェア dLitup

2.1　概要

アプリケーションデータを無線機能によって送信するとき，パケットロス増加の原因となるため，本研究は，コンピュータ制御による調光可能な天井照明の光度制御を用いたデータ送信手法 [1] を用いる．天井照明からのデータ送信を用いてセンサノード上のアプリケーションを動的に変更するために，本研究は，センサノード上で動作するミドルウェア「dLitup」を提案する．既存のミドルウェアを用いた場合，送信するアプリケーションデータの大きさは，動的変更内容の粒度の大きさに依存する．dLitup は，動的変更内容が大きいアプリケーション全体の動的変更のみを対象とした仮想マシンとして実装することで，通信コストの削減を図る．

dLitup は，天井照明から送信される数十ビットのデータを用いて，アプリケーションを動的に変更する．アプリケーションの動的変更に関する命令は，アプリケーションの追加または削除の 2 種類とする．dLitup 上で動作するアプリケーションは，センサノードにあらかじめ搭載したテンプレートプログラムを組み合わせることで構成される．たとえば，センサからセンシングデータを取得するテンプレートプログラムは，センサデバイスとセンシング周期を指定することでセンシングプログラムが機能できるように設計する．アプリケーションの動的変更は，使用するテンプレートプログラムの種類，およびテンプレートプログラムで用いるパラメタを天井照明からデータ送信することで実現する．

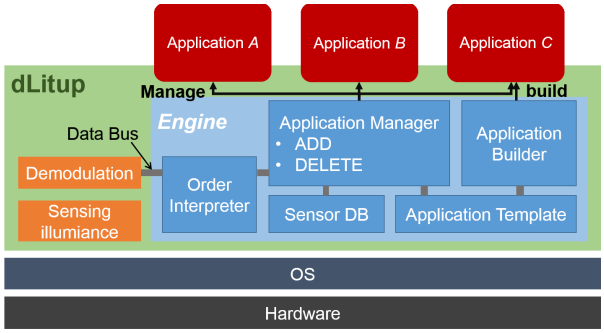

Fig. 1　dLitup のアーキテクチャ

2.2　dLitup のアーキテクチャ

dLitup は照度取得，照度データから受信データの復調，命令解釈，アプリケーションの実行など，複数のコンポーネントから構成される．dLitup のアーキテクチャを Fig. 1 に示す．

Fig. 1 中において，**Sensing illuminance** および **Demodulation** は，天井照明の光度制御を用いたデータ送信手法 [1] で必要な照度データの取得および照度データから受信データを復調するコンポーネントである．また，Fig. 1 中の **Order Interpreter** は復調データからアプリケーションを動的に変更する際のパラメタを構成する．**Application Manager** は，**Order Interpreter** が構成したパラメタをもとに，アプリケーションの動的変更命令を実行する．動的変更命令がアプリケーション追加だった場合，追加アプリケーションの内容に応じて **Sensor DB** を照会したあと，**Application Template** コンポーネントを呼び出す．動的変更命令がアプリケーション削除であった場合，天井照明から送信されるデータ内容に応じて，アプリケーションを削除する．**Application Template** は，アプリケーションを構成するプログラムをテンプレートとして保存，管理している．また，動的変更命令で指定された内容をもとにテンプレートプログラムを選択し，アプリケーショ

ンの動作順序を記述したアプリケーションプロシージャを構築する．**Application Builder** は，**Application Template** で構築したプロシージャを用いて，アプリケーションを動的に構成する．アプリケーションの構成は，天井照明から送信されるパラメタを，プロシージャ内に記述されたテンプレートプログラムに渡すことで実現する．**Application Builder** 内にあるインタプリタ上でアプリケーションを実行する．

3 dLitup の性能評価

3.1 評価概要

天井照明の光度制御を用いたアプリケーションの動的変更を実現するミドルウェア「dLitup」の性能評価を行う．本評価は，天井照明の光度制御を用いたデータ送信手法によるデータ受信が完了したことを前提とする．評価項目は次の 2 点である．

(1) データ受信完了後から動的変更命令が完了するまでのオーバヘッド

(2) アプリケーション実行時におけるネイティブコードとの実行性能比較

dLitup の評価にあたり，本研究は Zigduino を用いて dLitup を実装した．OS は Contiki を用いた．本評価における実装では，テンプレートプログラムとしてセンシングプログラム，データ送信プログラム，カウントアッププログラムの 3 つを採用した．本評価における実装では，dLitup は 49872 Byte のプログラムメモリを消費した．

3.2 動的変更命令完了までのオーバヘッド

データ受信が完了したあとから動的変更命令が完了するまでのオーバヘッドを測定した．測定した動的変更命令はアプリケーション追加命令とアプリケーション削除命令である．アプリケーション追加命令とアプリケーション削除命令の命令長はそれぞれ 25 ビットとした．データ受信完了後から動的変更命令の完了まで測定するとき，動作するコンポーネントは複数ある．

2 種類の動的変更命令に関して，データ受信完了後から動的変更命令完了までのオーバヘッド測定結果を Table 1 に示す．Table 1 からわかるとおり，動的変更命令が完了するまでに動作するコンポーネント数が少ないアプリケーション削除命令の方がオーバヘッドが小さい．また，両動的変更命令もオーバヘッドがマイクロ秒オーダで完了するため，データ受信完了後すぐにアプリケーションの動的変更が完了する．

3.3 ネイティブコードとの実行性能比較

dLitup 上で動作するアプリケーションの実行性能を検証するため，ネイティブコードで同様のアプリケーションを実行したときとの実行性能比較を行った．評価対象とするアプリケーションとして，照度を取得するごとにカウントアップを行うアプリケーション（SC），照度を取得したのちにランダム値を生成するアプリケーション（SR），カウントアップを行ったのちに，ランダム値を生成し加算するアプリケーション（CR）の 3 つのアプリケーションを用いた．

Table 1　動的変更命令実行時のオーバヘッド

Order	Components	Overhead [µs]
ADD	4	39.0
DELETE	2	19.9

Table 2　アプリケーション実行性能

Application	dLitup [µs]	Contiki [µs]
SC	58.3	56.6
SR	71.9	69.7
CR	15.1	13.4

測定したアプリケーションの実行速度を Table 2 に示す．dLitup の実行速度はネイティブコードと比較して，SC が 3.0 %，SR が 3.1 %，CR が 11.3 %のパフォーマンス低下となった．SC，SR はそれぞれセンサを使用したアプリケーションであるため，外部 I/O 待ちが大きく，dLitup のアプリケーション実行インタプリタのオーバヘッドを相対的に削減できる．一方，CR は計算処理アプリケーションであるため，dLitup のアプリケーション実行性能はネイティブコードと比較して 10 %以上低下した．

4 まとめ

本研究は，天井照明の光度制御を用いてセンサノード上のアプリケーションを動的に変更するミドルウェア dLitup を提案した．実機上で dLitup を実装し性能評価を行った結果，オーバヘッドは dLitup 内で動作するコンポーネント数に比例することがわかった．また，アプリケーションの実行性能は，外部 I/O 処理があるアプリケーションにおいてネイティブコードと変わらない性能を示した．

参考文献

1) 間博人, 堂面拓也, 本田雄亮, 三木光範, “オフィスにおける天井照明の光度制御を用いたセンサノードへのデータ送信手法,” 電子情報通信学会論文誌 B, Vol.J100-B, No.12, pp.1023–1032, Dec. 2017.