

リアルタイム Web 技術による HEMS サービスクラウド化の検討

増尾 剛[†] 中村二朗[†] 松岡 茂登[†] 長谷川 剛^{††} 村田 正幸^{†††}

松田 和浩^{††††}

[†] 日本電信電話株式会社 〒180-8585 東京都武蔵野市緑町 3-9-11

^{††} 大阪大学 サイバーメディアセンター 〒560-0043 大阪府豊中市待兼山町 1-32

^{†††} 大阪大学 大学院情報科学研究科 〒565-0871 大阪府吹田市山田丘 1-5

^{††††} NTT アドバンステクノロジー 〒180-0013 東京都三鷹市下連雀 3-35-1

E-mail: [†]{masuo.tsuyoshi,nakamura.jiro,matsuoka.morito}@lab.ntt.co.jp, ^{††}hasegawa@cmc.osaka-u.ac.jp,

^{†††}murata@ist.osaka-u.ac.jp, ^{††††}kazuhiro.matsuda@ntt-at.co.jp

あらまし 家電機器の省電力化の分野において HEMS (Home Energy Management System) が省エネ化の一つのキー技術として注目されており、HEMS に用いられるプロトコルとして、日本では ECHONET、ECHONET Lite の検討及び普及促進が図られている。現在 HEMS は HAN (Home Area Network) 内に閉じたシステムとして実装されているが、コストの低減を図るための方法として、家電製品を WoT (Web of Things) 化し、HEMS をクラウド型 ASP サービスとして提供することが考えられる。そこで本稿では、クラウド型 ASP サービスとして HEMS を実現するために必要となる、家電製品の WoT 化について検討を行った結果を報告する。具体的には、全体のアーキテクチャ、HEMS の実現に必要なとなる双方向 Web サービスの実装方法の比較検討結果を示し、最後にプロトタイプ実装を示す。

キーワード Home Energy Management System (HEMS)、リアルタイム Web、WebSocket、ECHONET

Study on HEMS over Cloud System utilizing Realtime Web Technologies

Tsuyoshi MASUO[†], Jiro NAKAMURA[†], Morito MATSUOKA[†], Go HASEGAWA^{††}, Masayuki

MURATA^{†††}, and Kazuhiro MATSUDA^{††††}

[†] Nippon Telegraph and Telephone 3-9-11 Midori-cho, Musashino-shi, Tokyo, 180-8585 Japan

^{††} Cybermedia Center, Osaka University 1-32, Machikaneyama-cho, Toyonaka, Osaka, 560-0043 Japan

^{†††} Graduate School of Information Science and Technology, Osaka University 1-5, Yamadaoka, Suita, Osaka, 565-0871 Japan

^{††††} NTT Advance Technology Corporation 3-35-1 Shimo-renjaku, Mitaka-shi, Tokyo, 180-0013 Japan

E-mail: [†]{masuo.tsuyoshi,nakamura.jiro,matsuoka.morito}@lab.ntt.co.jp, ^{††}hasegawa@cmc.osaka-u.ac.jp,

^{†††}murata@ist.osaka-u.ac.jp, ^{††††}kazuhiro.matsuda@ntt-at.co.jp

Abstract Home Energy Management System (HEMS) is paid much attraction as one of power-saving technologies for home appliances. In Japan, ECHONET and ECHONET Lite are possible candidates as HEMS protocol. Although the current HEMS is implemented as a closed system within Home Area Network (HAN), increased cost becomes a serious problem in large-scale deployment of HEMS. Therefore, in this report, we propose a method to apply ASP service for implementing HEMS at a low cost. We show the overall architecture, performance evaluation of protocol implementations for interactive Web service, and prototype implementation of the proposed system.

Key words Home Energy Management System (HEMS), Realtime Web, WebSocket, ECHONET

1. はじめに

インターネットの応用分野は近年大きく広がり、センサやア

クチュエータなどを全てインターネットで繋ぐ Internet of Things (IoT)、さらに、機器間の通信を Web 技術に基づいて行う Web of Things (WoT) [1] へとその概念が拡張されてきている。Energy

Management System (EMS) の分野においても、オープン技術によるマルチベンダ化やシステムの広域化を容易に行うために、Web 技術を適用する動きがある。例えば、ビルオートメーション (Building Energy Management System (BEMS)) にも Web サービス化された BACnet/WS [2] や IEEE 1888 が適用され始めている。一方、家電機器の省電力化に関しては、家庭用エネルギー管理システム (Home Energy Management System (HEMS)) が省エネ化の一つのキー技術として注目されている。HEMS に用いられるプロトコルとして、日本発の ECHONET 及び ECHONET Lite [3]、米国の SEP [4]、欧州の KNX [5] がある。現在 HEMS は Home Area Network (HAN) 内に閉じたシステムとして実装されているが、ユーザ利便性の向上やサービスコストの低減を図るためには、HEMS をクラウド型 Application Service Provider (ASP) サービスとして提供することが有効と考えられる。

そこで本稿では、クラウド型 ASP サービスとして HEMS を実現するために必要となる、家電製品の WoT 化について検討を行った結果を報告する。具体的には、システム全体のアーキテクチャを説明した後、HEMS サーバと家電機器間の通信のために双方向 Web サービスを実装する複数の手法を挙げ、それらのオーバーヘッドを比較し、WebSocket 技術を用いた手法が有効であることを示す。さらに、WebSocket を用いて、LED 照明を遠隔コントロールするプロトタイプ実装を示し、提案システムが十分実用的であることを示す。

本稿の構成は以下の通りである。2. 章では、提案するクラウド型 ASP サービスを用いた HEMS の基本的なアーキテクチャを示す。3. 章では、提案システムを実現するために重要となる、双方向 Web サービスの実現方法を挙げ、それらのオーバーヘッドを定量的に評価する。4. 章では、WebSocket 技術を用いた LED 照明制御システムのプロトタイプの紹介を行う。最後に 5. 章で本稿のまとめと今後の課題を示す。

2. 提案アーキテクチャ

HEMS サービスにおけるリアルタイム性への要求条件の明確な定義はないが、例えば PPS (Power Producer and Supplier: 電力事業者) における需給バランス調整では 30 分同時同量制御を行う必要があるため、5 分程度の分解能が必要となると考えられる。また、最近の蓄電池を用いた電力のピークカットシステムの例では、1 分毎の電力計測を行うものが存在する。本検討では、1 分以内程度の分解能での計測および制御を目標としたソフトリアルタイムシステムの実現を目標とする。

2.1 ユーザクライアント

家庭ユーザが用いるクライアントとしては、PC、スマートフォン、タブレットなど様々な汎用端末が使用できることが望ましい。また、サービス開発の効率化の観点から、それぞれの端末に応じてアプリケーションを開発するのではなく、ブラウザから操作できる Web サービスであることが望ましい。

2.2 HEMS サーバ

HEMS サーバをクラウド型 ASP サービスとして実現することによって、多くのユーザを集約することができるため、HEMS サービスの低コスト化が可能になると考えられる。また、近年、省エネルギーや電力不足対策のキー技術として注目されているデマンド・レスポンス・プログラム (DRP) との連携においても、

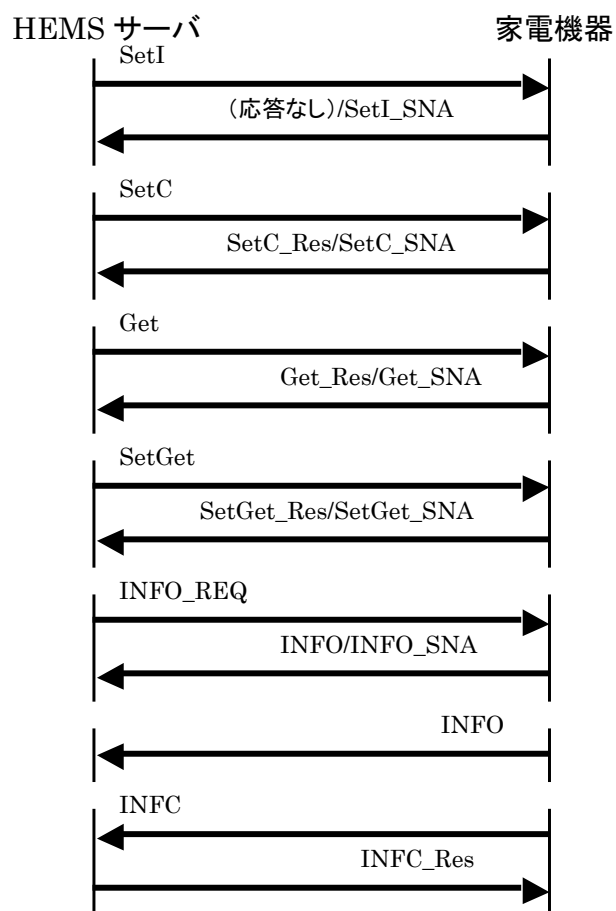


図1 ECHONET Lite のプロトコルシーケンス

ASP サービスとして HEMS サーバが集約されることによって、ユーザ情報を容易に収集することができるため、HEMS サーバが各家庭に設置されるよりも有利であると考えられる。

2.3 家電機器

家電機器を WoT することによって、家電機器を URI で識別することができるため、トランスポート層以下の下位層については規定が不要になる。このため、下位レイヤの実装は各メーカーが得意とする技術を用いることが可能であり、新たな技術が適用される場合にもサービスに影響を与えることがない。

2.4 Web サービス定義

家電機器を制御するための Web サービス定義は、新規に規定することも考えられるが、本稿では、日本での家電制御プロトコル標準である ECHONET Lite 仕様をベースに Web 化することを試みた。図1と図2に、ECHONET Lite のプロトコルシーケンス、及び対応する Web Service Description Language (WSDL) の記述例をそれぞれ示す。

図3に ECHONET Lite のフレームフォーマットを示す。Web サービス化にあたり、トランザクション ID (TID)、送信元オブジェクト (SEOJ)、相手先オブジェクト (DEOJ)、サービス種別 (ESV)、サービス対象プロパティ (EPC) とその値 (EDT) が必要となる。一方、XML もしくは JSON などの構造化データの使用を前提とすれば、処理プロパティ数 (OPC)、プロパティデー

```

<wsdl:interface name="EchonetInterface">
  <wsdl:operation name="SetI">
    <wsdl:input element="msg:SetI" />
    <wsdl:fault element="msg:SetI_SNA" />
  </wsdl:operation>
  <wsdl:operation name="SetC" >
    <wsdl:input element="msg:SetC" />
    <wsdl:output element="msg:SetC_Res" />
    <wsdl:fault element="msg:SetC_SNA" />
  </wsdl:operation>
  <wsdl:operation name="Get" >
    <wsdl:input element="msg:Get" />
    <wsdl:output element="msg:Get_Res" />
    <wsdl:fault element="msg:Get_SNA" />
  </wsdl:operation>
  . . .

```

図2 ECHONET Lite の WSDL 記述例

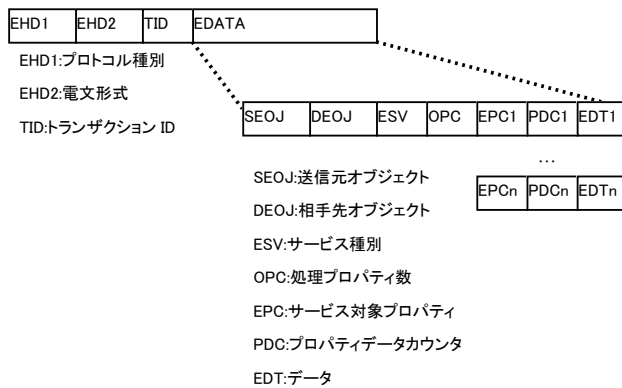


図3 ECHONET Lite のフレームフォーマット

カウンタ (PDC) は不要となる。図4に家庭用エアコンのプロパティのXML Schema Document (XSD) 記述の例を示す。

ECHONET Lite はシンプルなバイナリプロトコルであるためフレーム長が短い。これに対し、Web サービス化した場合にはフレーム長が増加する。図4に示した家庭用エアコンのプロパティにおいて、エアコンの電源を入れる場合について比較すると、ECHONET Lite では14バイトであるのに対し、Web サービス化したものをREST バインディングした場合には、図5のようにになり、170バイトが必要になる。しかしながら、近年、組み込み機器用途のプロセッサの性能向上、低価格化が進んでいるため、この程度のフレームサイズの増加は許容できる可能

```

<complexType name="setPropertyType">
  <sequence>
    <element name="TransactionId" type="integer" />
    <element name="ESV" type="tns:ESVType" />
    <element name="operationStatus" type="tns:operationStatusType" />
    <element name="operationModeStatus" type="tns:operationModeStatusType" />
  </sequence>
  <complexType>
    <simpleType name="operationStatusType">
      <restriction base="string">
        <enumeration value="ON" />
        <enumeration value="OFF" />
      </restriction>
    </simpleType>
  </complexType>

```

図4 家庭用エアコンのプロパティの XSD 表現例

```

<ns1:SetC_Res
xmlns:ns1="http://www.echonet.gr.jp/echonet_lite/xsd">
  <ns1:TransactionId>1</ns1:TransactionId>
  <ns1:ESV>SetC</ns1:ESV>
  <ns1:operationStatus>ON</ns1:operationStatus>
</ns1:SetC_Res>

```

図5 エアコンをオンにする場合の XML メッセージ例

```

{"TransactionId":1,
"RSV:"SetC",
"operationStatus":"ON"}

```

図6 エアコンをオンにする場合の JSON メッセージ例

性がある。また、XML の代わりに JSON を用いて縮約した場合は、図6に示すように42バイトで同様の表現が可能となる。

また、データを他のシステムに引き継ぎ連携するようなシステムを構築する場合には、データの可読性が重要となる。この点においても、バイナリプロトコルではなく、XML や JSON による記述が優位であると考えられる。

3. Web サービスとしての実装

2.4 節に示した Web サービス記述では、多くのサービスを、HEMS サーバを起点として家電機器へサービス要求するよう記述した。これをそのまま Web サーバ、Web クライアントの関係

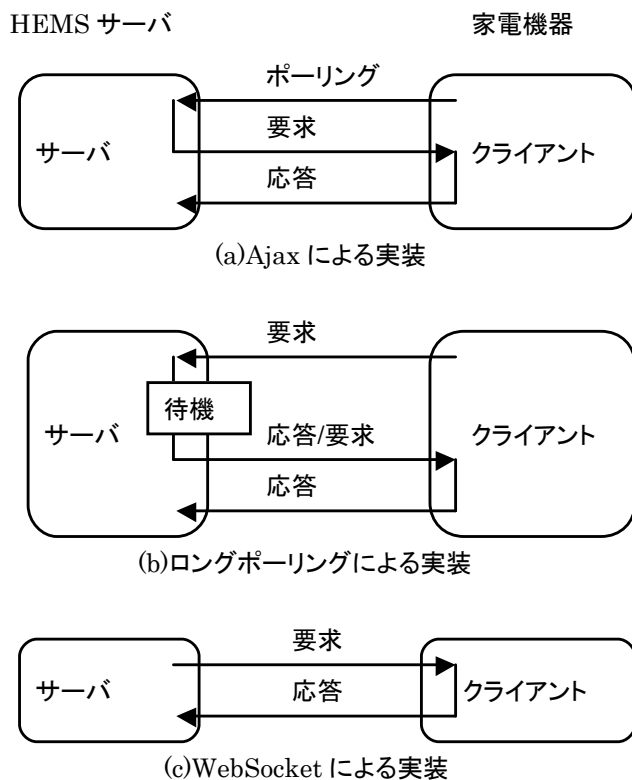


図7 双方向 Web サービスの実現方法

に置き換えると、家電機器を Web サーバとし、HEMS サーバを Web クライアントとする構成が考えられる。しかしながら、家電機器は一般的に HAN 内のローカルアドレスを持ち、Network Address Translation (NAT) を介してインターネットに接続されている。このようなネットワーク形態との整合性を考えると、家電機器を Web クライアントとし、HEMS サーバを Web サーバとするのが望ましい。この場合には、サービスへの接続の起点は Web クライアントである必要があるのに対し、サービスの起点は HEMS サーバとなることから、単純な Web サービスとしては実装できず、双方向の Web サービスを実装する手段が必要となる。本章では、双方向 Web サービスを実現する手段を複数挙げ、それらのオーバーヘッド比較を行うことにより、提案システムに適した実現手段を明らかにする。

3.1 双方向 Web サービスの実現方法

図7に、現在用いることができる双方向の Web サービスを実装する手段を示す。図7(a)は Ajax (Asynchronous Javascript+XML) と呼ばれる手法であり、クライアント側からの定期的なポーリングに、サーバからの要求を載せて応答することで、双方向のサービスを実現する。この実装方法では、HEMS サーバからのサービス要求がクライアントに伝わるまでの遅延時間が、クライアントからのポーリング間隔に依存する。遅延時間を小さくするためにはポーリング周期を短く設定する必要があるため、無駄なトラフィックが発生する問題がある。

図7(b)はロングポーリングと呼ばれる手法であり、クライアントからの要求をサーバ側で保持し、サーバからのサービス要求があった時点でサービス要求を載せて応答することで、双方

向通信を実現する。ロングポーリングは Ajax に比べて HEMS サーバが制御したいタイミングでサービス要求を送信できるため、遅延を小さくすることができる。しかしながら、接続の維持コストが大きいという問題がある。

また、Ajax、ロングポーリングともに、複数ドメインでの連携は JSONP (JSON with padding) を適用することで可能となるが、セキュリティに対する配慮が必要である。

図7(c)は HTML 5 の一つの基盤技術として最近標準化された WebSocket [6] による実装である。WebSocket 技術を用いることによって、HTTP セッションを双方向通信が可能なセッションとして用いることができるようになるため、HEMS サーバからのサービス要求が家電機器に伝える際に遅延が生じない。また、Ajax 及びロングポーリングは HTTP 通信によって HEMS サーバと家庭機器間の通信が実現されるが、WebSocket は TCP 通信を用いるため HTTP のヘッダを必要とせず、接続の維持コストが小さい。複数ドメインの連携についてもプロトコル上規定されているため、セキュリティ上の問題も少ないと考えられる。

3.2 オーバヘッド評価

本節では、3.1 節で示した 3 種類の実現方法のオーバーヘッドの比較を定量的に行った結果を示す。

3.2.1 評価環境

評価においては、下記のような環境を想定する。各家庭には 10 台の家電機器と、1 台のスマートメータが設置されており、HEMS サーバは ASP サービスとして外部ネットワークに設置される。各機器と HEMS サーバ間の通信内容と通信頻度を表 1 に示す。このような通信が発生する際に、図7に示した 3 つの実現方法に従って制御パケットが発生した場合の、発生するネットワークトラフィック量、HEMS サーバにおける同時アクティブセッション数、HEMS サーバが処理するリクエスト頻度を評価する。なお、Ajax (図7(a)) およびロングポーリング (図7(b)) において、家庭機器からの通知をポーリングによって実現する場合には、ポーリング間隔は表 1 に示した通信頻度に応じて行うものとする。一方、ユーザによる家電機器の操作をポーリングによって実現する場合には、ユーザの操作に対する遅延時間を考慮し、ポーリング間隔を 1 分に 1 回と想定する。

ネットワークトラフィック量の評価の際には、ECHONET の通信内容は 1 回あたり 500 バイトとし、それを HTTP 通信によって運ぶ場合には 200 バイトのヘッダ、TCP 通信によって運ぶ場合には 40 バイトのヘッダが付与されるものとする。また、Ajax およびロングポーリングの手法におけるポーリングは HTTP ヘッダのみの通信で実現されるものとする。

3.2.2 評価結果と考察

表 2 に、2000 世帯を収容する場合における、各実現手法におけるネットワークトラフィック量、HEMS サーバにおける同時アクティブセッション数、及び HEMS サーバにおける処理頻度を示す。結果より、HEMS サーバの処理頻度に比べて、ネットワークトラフィック量が小さいことがわかる。これは、ECHONET の 1 回の通信が 500 バイトと小さいためである。一方、HEMS サーバにおける処理頻度は比較的大きく、収容世帯数を増加させた場合には、サーバにおける CPU 能力がボトルネックになることが考えられる。また、Ajax によって実現した場合に、他方式に比べて下りのトラフィック量が多いことがわかる。これ

表1 HEMS サーバと家庭機器間の通信パラメータ

通信機器と方向	通信内容	通信頻度
家電機器→ HEMS サーバ	運転状態 (ON/OFF) や電力の通知	1 分に 1 回
HEMS サーバ→家電機器	家電機器の制御プラン指示、ユーザによる家電機器の操作	1 時間に 1 回
スマートメータ→ HEMS サーバ	消費電力の通知	5 分に 1 回

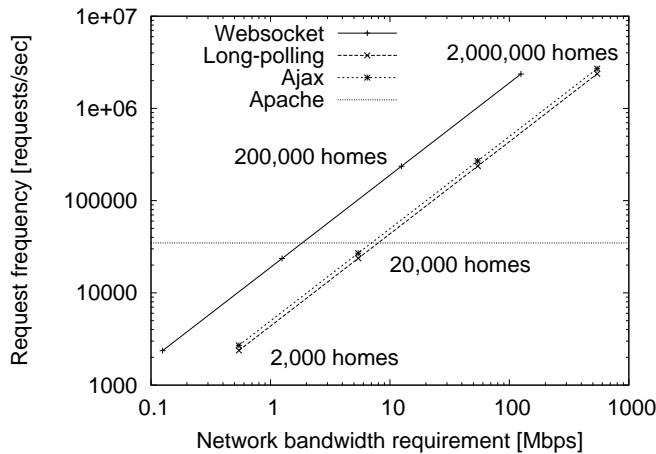


図8 収容世帯数の増加が与える影響

は、ユーザの家電操作のための通信をポーリングによって実現するため、ポーリングのためのトラフィック量が増大していることが原因である。この結果から、ユーザの家電操作のための通信をポーリングによって実現することは避けるべきであると考えられる。

そこで、HEMS サーバのリクエスト処理能力を評価するために、[7] から得られる Web サーバに対するベンチマーク結果を用いる。このベンチマークは、100 台のクライアントから、300KB のドキュメントのダウンロードを継続的に行い、スコアとして、サーバが処理したリクエストの頻度を requests/sec の単位で出力するものである。ECHONET のデータサイズに比べてドキュメントサイズが大きいいため、直接的な比較はできないが、本稿では目安として用いる。

図8に、各実現方式において、収容する世帯数を2,000世帯から2,000,000世帯まで増加させた場合における、ネットワークトラフィック量とHEMSサーバのリクエスト処理頻度の変化を示している。図中には、[7]から得られるベンチマーク結果のうち、3.2GHz動作、12コアのCPUを持つサーバに対するベンチマークスコアを合わせてプロットしている。図から、ネットワークトラフィック量の増加よりも、リクエスト処理頻度の増加によって、システムのボトルネックが発生すると推定される。また、WebSocketを用いて実現することによって、他方式に比べてネットワークトラフィック量を1/5程度に削減できることがわかる。

これらの検討結果から、本研究においては、WebSocketを用いた実装について検討を行うこととした。

本章における評価結果は、各機器からの通信が平均的に発生することを想定したものである。しかし、実際の環境においては、定期的なイベントの発生や、電力制限などのイベントの発生にともない、多数の機器が同時に通信を行う状況が想定される。そのような状況を想定したオーバヘッド評価や、過負荷時

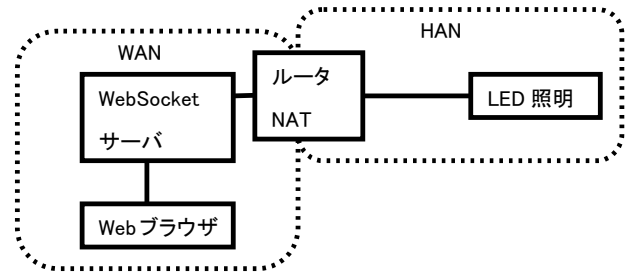


図9 デモシステムのブロック図

のHEMSサーバにおける対策などについては今後の課題としたい。

4. プロトタイプ実装

本研究においては、WebSocket技術に基づく提案システムの動作検証のために、図9に示すようなデモシステムを構築している。WebSocketサーバにはHEMSサービスは実装せず、複数のWebSocketコネクションを終端し、マルチキャスト転送する機能のみを実装した。また、家電制御はJavascriptによりPC等のブラウザ側で実行する構成とした。

4.1 WebSocketサーバ

ECHONET LiteではINFOメッセージはマルチキャストされることにより、連携可能な家電機器の発見などを可能としている。しかし、HTTPやWebSocketはユニキャスト通信に基づくプロトコルである。そのため本稿では、サーバが複数存在する場合や家電機器間で直接情報交換を行うような実装への拡張性を考え、WebSocketメッセージを複数の家電機器間で交換する機能を実装した。具体的には、WebSocketサーバはあるWebSocketクライアントから受信したメッセージを、送信元以外のWebSocketクライアントに送出する。

実装にあたっては、Tornado httpサーバおよびWebSocketライブラリを使用した。

4.2 Webブラウザ

本稿では、単純なLED照明の制御を対象とし、電源のオン/オフ制御、照度制御、及び電力情報の取得を行うシステムを実装した。ユーザがLED照明を制御するためのWebページはHTMLおよびJavascriptにより実装した。Javascriptを実行したWebブラウザとサーバ間もWebSocketで通信を行うことにより、制御信号はブラウザに表示されたWebページ中のボタンを押下したタイミングで送出される。また、LED照明から送られた状態情報(電源のオン/オフおよび消費電力)をリアルタイムに表示する。図10に画面イメージを示す。

4.3 家電機器

図11に今回試作したLED照明の回路図を示す。マイクロコ

表2 オーバヘッドの評価結果

実現方式	上りトラフィック量 (Mbps)	下りトラフィック量 (Mbps)	同時アクティブセッション数	HEMS サーバの処理頻度 (requests/sec)
Ajax	0.543	0.532	67.5	2694.4
ロングポーリング	0.544	0.032	68.1	2372.1
WebSocket	0.125	0.020	34.2	2361



図10 LED 照明の制御画面

ントローラには、NXP セミコンダクター社の LPC1768 の評価ボードを用いた。プログラム環境は Web 上のオンライン IDE を使用した。mbed には活発なコミュニティが存在し、WebSocket も既にライブラリ化されている。LPC1768 はイーサネット、CAN、I2C、SPI などのデジタルインタフェースの他にアナログ入出力、PWM 出力を持ち、今回のようなプロトタイピングが容易に行える。

LED 照明とルータ間は PoE (Power over Ethernet) で接続し、通信およびルータから LED 照明への電源供給を行っている。LED の点灯は制御用の FET をマイクロコントローラからの PWM 信号で行い、照度制御は PWM のデューティ比を変更することで行った。

図10に示した Web ブラウザの画面においては、照度制御を行うと、LED 照明の状態情報 (オン/オフ)、消費電力がリアルタイムで変化する。LED 照明からの照度・電力情報を定期的に INFO として送信するように実装した。

現在の簡単な実験環境では、ネットワークの遅延が 10ms 以下、背景トラフィックも無いため、良好な動作を確認している。今後、ネットワーク輻輳に伴うパケット廃棄や遅延時間の変動が発生した際の挙動の検証などを行う予定である。

5. まとめと今後の課題

本稿では HEMS サービスをクラウド型 ASP サービスによって提供するための技術検討を行った結果について報告した。全

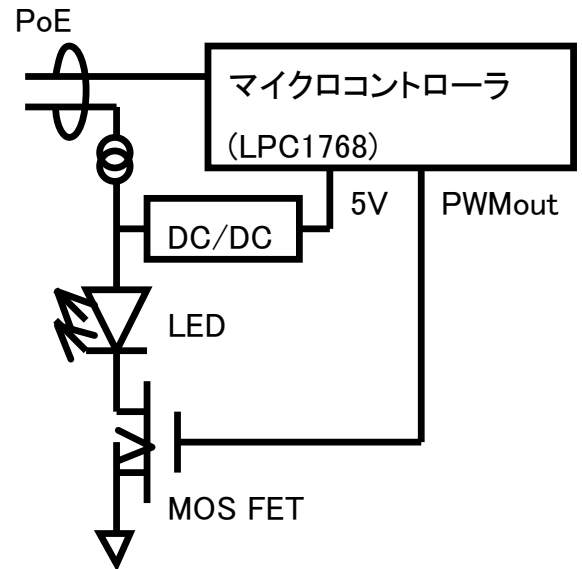


図11 LED 照明機器の回路図

体構成および ECHONET Lite をベースとしたサービス仕様、下位プロトコルへのマッピング方法について述べた。また、サービス実現のために必要となる、双方向 Web サービスの実装方法について比較評価を行い、WebSocket に基づく手法が優位であることを示した。さらに、プロトタイプシステムの概要と評価について述べた。

今後の課題としては、遅延時間や背景トラフィックを付加した状態でのプロトタイプの評価や、クライアントを増やした場合の評価が挙げられる。また、イベントが短時間に集中して発生するような状況における、HEMS サーバにおけるセッション制御方法などを検討する必要があると考えられる。

文献

- [1] S. Duquennoy, G. Grimaud, and J.-J. Vandewalle, "The Web of Things: Interconnecting devices with high usability and performance," Proceedings of ICCESS 2009, pp.323-330, June 2009.
- [2] ASHRAE, "ANSI/ASHRAE addendum c to ANSI/ASHRAE standard 135-2004," Available from <http://www.bacnet.org/Addenda/Add-2004-135c.pdf>, 2004.
- [3] EchoNET Web Page. Available from <http://www.echonet.gr.jp/>.
- [4] State Energy Program Web Page. Available from <http://www1.eere.energy.gov/wip/sep.html>.
- [5] KNX Association Web Page. Available from <http://www.knx.org/>.
- [6] Websocket Web Page. Available from <http://www.websocket.org/>.
- [7] OpenBenchmarking Web Page. Available from <http://openbenchmarking.org/>.