Vol.2013-GN-86 No.39
Vol.2013-CDS-6 No.39
2013/1/17

サービスゲートウェイ向け ECHONET Lite バンドルの開発

寺岡秀敏[†,a)]　今井光洋[†]　小坂忠義[†]　奈良祐樹[†]　小田輝[†]

概要: 国内で公知な標準インタフェースとして推奨された ECHONET™ Lite に対応したサービスゲートウェイ（SGW）向けのミドルウェアを OSGi™ のバンドルとして開発した．設計に当たり，マルチサービスを実現する SGW 上にミドルウェアを実装するにあたっての課題及び ECHONET Lite 規格の実装上の課題を検討した．検討した課題に基づいて，OSGi フレームワークの機能を最大限活用して，アプリケーション開発効率の向上，省リソースとアプリケーション停止時間最小化の両立，任意の ECHONET Lite 機器特定の容易化，下位通信層への柔軟な対応を実現する ECHONET Lite バンドルの構成を提案する．さらに，提案する構成のバンドルを実装して評価システムを構築し，前記要件の観点から提案する ECHONET Lite バンドルの有効性を評価した．

キーワード: Service Gateway ， HEMS， ECHONET Lite， OSGi

Development of ECHONET Lite Bundle for Service Gateway

HIDETOSHI TERAOKA[†,a)]　MITSUHIRO IMAI[†]
TADAYOSHI KOSAKA[†]　MASAKI NARA[†]　KAGAYAKI ODA[†]

Abstract: In this paper ECHONET Lite Bundle based on OSGi is proposed. ECHONET Lite is recommended by The Ministry of Economy, Trade and Industry as the standard interface for connecting electric appliances and a home energy management system (HEMS) in the home. . In the first part of this paper the problems to implement ECHONET Lite Bundle as a middleware of service gateway are clarified. In the second part, the software structure to solve the problems is proposed. Using functions of OSGi effectively, the proposed method achieves (a)reduction of application development cost, (b)resource-saving and minimizing stop time of applications, (c)tracing ECHONET Lite devices easily, and (d)attaching various low layer communication interfaces easily. Finally, we implement the proposed method and evaluated the effectiveness of the ECHONET Lite Bundle.

Keywords: Service Gateway, HEMS, ECHONET Lite, OSGi

1. はじめに

家庭内の様々な機器を接続してサービスを提供するホームネットワークシステムについて，さまざまな取り組みがなされてきた．例えば，ユーザの利便性を向上するホームオートメーション（HA）機能や，防犯などのセキュリティサービスなどが専用の端末を利用して提供されている．このような状況で，近年，ホームネットワークを活用して家庭のエネルギーマネジメントを行う，ホームエネルギーマネジメントシステム（HEMS）が注目されている．HEMS において，ネットワークで接続される家電機器や住宅設備は家庭ごとに異なっており，機器接続のプロトコル，伝送メディアも多様である．このため，共通の手段で情報取得や制御が行えるような標準技術の重要性が改めて認識されている．そのような標準技術として，国内では ECHONET Lite[1]が公知な標準インタフェースとして推奨され，平成 23 年度「エネルギー管理システム導入促進事業費補助金（HEMS 導入事業）」においても当該規格の搭載が補助対象機器の要件となっている[2].

一方，これまで専用の端末上に実装されていた HA やセキュリティ，HEMS といった複数の機能を一つの端末（SGW）上に搭載するため，さまざまなサービスやデバイスに対応できるホーム ICT（Information and Communications Technology）実行基盤として OSGi™ が注目されている[3]. OSGi では，Java™ ベースのソフトウェアモジュール（バンドル）をネットワーク経由で動的に更新することができる．また，更新の際，複数のバンドルが稼働しているときに，特定のバンドルだけを停止・再起動したり，更新したりすることが可能であり[4]，家庭ごとに環境や提供するサービスが異なっていても，必要なバンドルの組み合わせで適切なシステムを構築することが可能となる． 図 1 に OSGi を利用したシステムの構成例を示す.

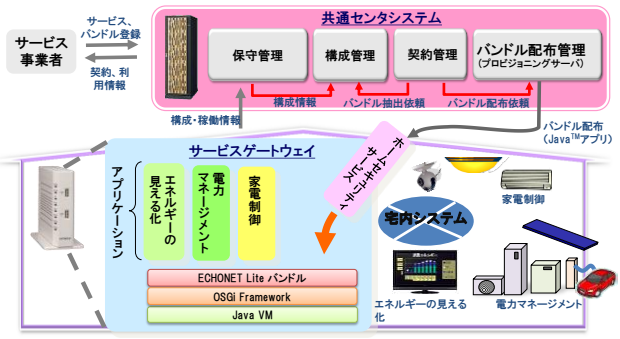

図 1　OSGi を利用したホームネットワークシステム概要
Figure 1　Overview of Home Network System based on OSGi

そこで本研究では，国内の HEMS 標準プロトコルである ECHONET Lite に準拠したコントローラ用のミドルウェア

† (株)日立製作所　Hitachi Ltd. Yokohama, Kanagawa 244-0817,Japan
a) hidetoshi.teraoka.rf@hitachi.com

を OSGi バンドルとして構成する方法について検討し，PC 及び SGW 上に実装して評価を行った．

2. 先行研究

ECHONET Lite 規格に先立ち標準化された ECHONET 規格では，ソフトウェア実装のための標準 API として，アプリケーション（AP）がミドルウェアを利用するための「基本 API 仕様」及びミドルウェアと下位通信層の間の「共通下位通信インタフェース仕様」が定義されており[5]，本仕様を参照した OSGi 上で ECHONET 通信ミドルウェアバンドルが開発されている[6].

また, OSGi 上で ECHONET バンドルを構成し PUCC（P2P Universal Computing Consortium）プロトコルと接続して ECHONET 機器の制御を行うシステムも提案されている[7].

さらに，ECHONET Lite 規格を Java 言語で実装したミドルウェアソフトとして，商用ベースの製品[8]やオープンソースで公開されている実装[9]がある．

本研究では，これらの先行研究を踏まえ，OSGi 上の ECHONET Lite に準拠したコントローラ用のミドルウェアとして，ECHONET Lite ミドルウェアバンドルの検討及び評価を行った．

3. SGW の ECHONET Lite バンドル実現における課題の考察

3.1 SGW 上の OSGi バンドル実装における課題

(1) AP 開発効率の向上

本提案の ECHONET Lite を搭載するコントローラ機器の基本的なユースケースとしては以下のようなステップが考えられる．ここで, 操作とは1制御または状態取得のことを言う．

(a)ネットワーク上に接続された機器を探索・発見する
(b)発見した機器から操作対象を選択する
(c)選択機器に対して，操作要求を発行する．
(d)要求に応じて ECHONET Lite 電文を構成し，
　ネットワークに送出する．
(e)機器からの応答や通知を受信し，電文を解析する．
(f)前記解析結果に基づいて要求の成否を判定し，
　結果に応じた処理を行う．
(g)他機器からの ECHONET Lite 電文による要求を受信し，
　応答する

AP は，上記(a)から(f)のステップが実行できれば,「使用電力をスマートメータから取得し，表示する」ことや「取得した室温や使用電力に基づいてエアコンの設定温度を変更する」などのサービス実現が可能になる．

上記(a)から(f)のステップのうち，(b)(c)(f)のステップ，すなわち，ネットワークに接続された機器のうち，どれを選択してどのような要求を行い，その結果に応じてどのような処理を行うかについては，AP ごとに異なる．一方で，(a)(d)(e)は複数の AP が共通的に必要とする機能である．(a)に関し，ECHONET Lite 規格は，プラグアンドプレイにより機器の発見ができるという特徴を有している．しかし，どのような手順で機器の発見を行うか, などについては AP 開発者が個別に設計及び実装を行う必要が生じる．

また，本提案のバンドルを利用する AP の主要機能はコントローラとして他機器を制御することであるため，他の機器から情報を取得されることや制御されること（前記(g)に相当）は想定する必要がない場合が多い．一方，ECHONET コンソーシアムによる規格適合性認証[10]では, 被制御機器としての項目が中心となっている．すなわち，本バンドルを搭載するコントローラが ECHONET Lite 対応機器として認証を取得するためには，主要なユースケースである他の機器を制御する機能(a)～(f)ではなく, (g)の機能を実現する必要がある．

以上から，SGW において，複数の AP に共通的に利用される(a)(d)(e)の機能,及びユースケースとして利用される可能性が低いが認証取得のためには重要度の高い(g)の機能を実現するにあたって，AP の開発工数をできるだけ低減することが課題となる．

(2) 省リソースと AP 停止時間の最小化の両立

SGW は，エアコンのような被制御機器ほどではないが，組込み機器として構成され，省リソースが要求される場合が多い．さらに，マルチサービスを実現する SGW においては AP の要求により, SGW 機器そのものや ECHONET Lite ミドルウェアの無停止を求められるケースもある．一方，ECHONET Lite 規格では機器を表現するための情報として，多数の機器オブジェクト及びプロパティの値が定義されている．これは，他のホームネットワークプロトコルと比較した場合の ECHONET 規格の強みである一方で，これらを最初からフルに実装した場合，定義値の数だけプログラムサイズが大きくなってしまい，大きな ROM 容量が必要となる．これを解決し，省リソースを実現する 1 つの方法として，必要最小限の機器オブジェクト定義のみ実装し，必要に応じてソフトウェアの更新を行う方法がある．しかしこの場合, SGW またはソフトウェアの再起動が必要となり，無停止が要件である AP には都合が悪い．以上から，ECHONET Lite バンドルでは，省リソースかつ AP の停止時間をできるだけ短くできることが課題となる．

3.2 ECHONET Lite 規格の実装上の課題

(1) ECHONET Lite 機器の識別

ECHONET Lite を用いて接続された機器をユーザが利用するにあたって，プロトコルや下位通信層で定められた識別情報のままではどの機器がどれに当たるか分かりにくい．そのため，例えば「リビングのエアコン」といった機器名称などの人間に分かりやすい識別情報と紐づけ，ネットワークの構成や機器のアドレス情報などが変更されてもそのまま対応付けができる必要がある. 例えば, UPnP（Universal

Plug and Play）では UUID（Universally Unique Identifier）と機器名称を紐づけておけば，IP アドレスが変わっても接続された機器が一意に識別できる．一方， ECHONET Lite 規格では機器を永続的に一意に識別（トレース）するためのいくつかの方法は提供されているが，どの情報を利用可能にするかはメーカの実装依存となっており，すべての ECHONET Lite 機器で統一的に利用できない． 表 1 に ECHONET Lite 規格において機器識別に利用できる情報と当該情報を利用するにあたっての課題を示す．本課題について検討し，個体識別番号あるいはメーカコードと製造番号の組み合わせのどちらかを使うとトレース可能という考察がなされている [12].

表 1　ECHONET Lite における機器識別のための情報
Table 1　Information for tracing ECHONET Lite devices.

識別情報	内容	課題
下位通信層における識別情報	IP アドレスまたは MAC アドレス	下位通信層のプロトコル・接続方法に依存
識別番号	オブジェクトをドメイン内で一意に識別するための情報（8～16byte）	v1.01 では，0x00(未設定)も設定可能
個体識別情報	ドメイン内で各ノードを一意に識別可能とし，かつ機器の移動（サブネットの変更など）後も常に同一ノードは不変なものとして取扱い可能とするための情報（2byte）．初期値は任意（乱数など）．コントローラから変更可能	コントローラが複数存在する場合の競合回避方法などのガイドライン無し
メーカコード+製造番号	機器の製造者が一意であることを保証する番号	製造番号はオプション

上記のように，すべての機器で統一的に利用できる方法がないことから，コントローラ用のミドルウェアは，AP が ECHONET Lite 機器を容易にトレースする仕組みを提供する必要がある．

(2) **下位通信層の多様性**

ECHONET Lite 規格は，OSI 参照通信レイヤ 5～7 層を規定した規格であり，通信アドレスは IP アドレスまたは伝送メディアの MAC アドレスを利用することが前提とされている．これは，プロトコルの軽量化と 4 層以下の下位通信層はグローバルな標準仕様を利用したいという要望に基づいて策定されたものである．しかし一方で下位通信層に規定がないため，伝送メディアや下位通信層の実装の相違によって相互接続性に問題が生じる可能性がある．ECHONET Lite の下位通信層の候補としては UDP/IP/Ethernet 以外にも複数の通信層（920MHz 帯無線を利用する ZigBee™ IP など）が想定されており [11]，ECHONET Lite を使用するシステムでどの下位通信層を採用するかはユーザまたは機器ベンダの選択によるため家毎，機器毎に多様な構成となってしまう．また，図に示すように SGW は複数の下位通信層にまたがる機器と接続するというユースケースも考えられる．そのため，さまざまな機器と接続する可能性のあるコントローラ用のミドルウェアは複数の下位通信層に柔軟に対応できる構成である必要がある．下位通信処理部をバンドル内に実装する構成とした場合，家毎の環境に合わせて複数のバージョンのバンドルが必要となり，開発効率や保守性に問題が生じる．また，新しい下位通信層に追加対応する場合にはバンドルの更新が必要になり，その間 AP を停止させてしまう可能性がある．

図 2　2 つのサブネットにまたがる HEMS の例
Figure 2　Example of HEMS that has 2 subnets

以上の課題を考慮し，ECHONET Lite バンドルの要件を以下のように定める．

R1:AP 側の実装を最小限にして ECHONET Lite 規格に準拠し，開発効率向上に寄与すること．

R2:省リソースと AP の停止時間最小化を両立できること．

R3:AP が容易に ECHONET Lite 機器を一意に識別する仕組みを提供すること．

R4:様々な下位通信層に柔軟に対応できること．

本稿では，これらを満足するソフトウェアを開発することを目標とする．

4. ECHONET Lite バンドルの設計

4.1 AP 開発効率向上への対応

AP 開発効率向上の観点から，ECHONET Lite バンドルが提供すべき機能について検討した．

ECHONET Lite プロトコルを実装したミドルウェアの基本機能として，**通信処理機能**を提供する．通信処理機能は，電文生成や解釈といったプロトコル処理に加え，すなわち不正な電文を発行しないためのチェックや，不正な電文を受信した場合の異常処理などを行う．

ネットワークに接続された ECHONET Lite 機器を検出する機能は，複数の AP に共通的に必要となる．そこで，機器検出を容易にするために，OSGi フレームワークを活用した**他機器管理機能**を提供することとする．他機器管理機能は， ECHONET Lite 機器の発見，切断監視を行い，発見した機器を AP から利用可能な OSGi サービス（EchonetLiteDevice サービス）として OSGi フレームワークに登録する．また，応答状態を監視し，応答がなくなった機器は OSGi サービスから削除する． AP は，EchonetLiteDevice サービスが提供する**サービス API** を利用することで，発見した機器を操作したり，イベントを受信することができる．ECHONET Lite バンドルは，サービス API による AP からの要求（API コール）に従ってネットワークに電文を送出し，対応する応答を API の戻り値やイベ

ントとして AP に通知する．

このように，SGW のプラットフォームとして，プラグアンドプレイによる機器発見及び機器管理機能や，発見した機器に対する操作インタフェースを AP に提供することにより，AP は所望の機器を制御するためには OSGi サービスの検索・取得のみ実装すればよく，差別化機能の実装に注力することができると考える．

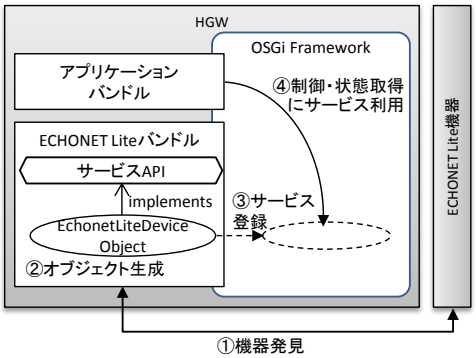

図 3　OSGi を利用した機器登録・利用方法

Figure 3　Device search and use method using OSGi.

ECHONET Lite の規格適合性認証は，大きく「送信電文は適切か」「受信電文を正しく解釈し，適切に自機器・自ノードオブジェクトの処理ができているか」という点の確認が行われる．前者は主に前述の通信処理機能でサポートされるが，後者を実現するためには，自機器・自ノードオブジェクトの管理が必要になる．このような自機器・自ノードオブジェクトの管理はミドルウェアの範囲外として，AP 側での対応が必要とされる場合が多い[5].

前述の通り，AP は接続された機器の状態取得及び制御が主眼であるため，自機器・自ノードオブジェクトの管理を最小限にできることが望ましい．そこで，ECHONET Lite バンドルでは，**自機器管理機能**を提供する．自機器管理機能は，コントローラとして搭載が必要な機器オブジェクトスーパークラスの及びノードプロファイルクラスの必須プロパティを管理し，読み出し要求に対する応答，書き込み要求に対する管理値の更新や状態変更通知を行い，被制御機器としての機能を提供する．

本機能により，AP は自機器について特別な管理をしなくても，ECHONET Lite 規格でいうコントローラとしての最小限の要求を満足し，サブネット内の他機器からは，一つの ECHONET Lite「コントローラ」機器として認識される．

これにより，AP 開発者は認証取得のための実装に工数を低減でき，ECHONET Lite バンドルを搭載した機器について，ECHONET Lite コントローラ機器としての認証を容易に取得することが可能となると考える．

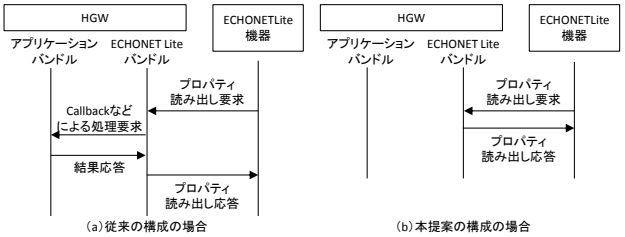

図 4　電文受信時のシーケンス比較

Figure 4　Comparison of sequence at receiving ECHONET Lite frame.

4.2 省リソースと AP 停止時間の最小化への対応

次に，サービス API の定義方法を，省リソース及び AP 停止時間の観点から検討した．サービス API は ECHONET Lite 規格において具体的に何をどう制御するかにあたる ECHONET プロパティの定義値をどこで管理するかによって 2 通りの定義方法が考えられる．すなわち，ミドルウェアのレイヤで定義値を管理して操作毎の API（setTemperature()など）を定義し AP は ECHONET 規格の定義値を意識しないようにする方法（a）と，AP が ECHONET 規格を意識し定義値を管理して EPC 及び EDT を渡す全操作に共通の API（setProperty()など）を定義する方法（b）である．

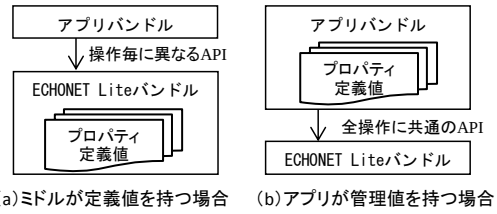

図 5　サービス API の提供方法

Figure 5　Implementation methods of service API

また，プロパティの定義値については，最初から Full 実装する場合と，必要に応じ，バンドルを更新して順次追加していく方法の 2 通りが考えられる．表 2 に，省リソースと AP 停止時間の観点から各方法を比較した結果を示す．

表 2　サービス API 定義方法の比較

Table 2　Comparison of implementation methods of service API.

実装方法		省リソース	停止時間
(a)	Full	×	○
	順次追加	○	×
(b)	Full	×	○
	順次追加	○	△*1

*1 更新対象以外の AP には影響しない．

比較の結果，省リソースと AP 停止時間の最小化を両立するためには，プロパティの定義値をバンドル外で管理し，必要に応じて順次追加していく方法がよいと考えられる．ただし，このままでは，更新対象の AP は更新時に停止する必要があることと，AP 開発者は ECHONET Lite プロトコルの詳細定義を意識する必要が生じてしまうという課題

がある．

そこで，本バンドルを利用する AP 周辺の構成として，図 6 の(b)’のような構成を提案する．具体的には，AP 向けインタフェースとして別途操作内容を抽象化したインタフェース（機器操作 I/F 抽象化バンドル）を定義し，これを実装した機器操作 I/F ドライババンドルを用意する．EPC,EDT 定義値をこのドライババンドルで保持し，EchonetLiteDevice サービスを呼び出すことにより，開発者は ECHONET Lite 規格の詳細を知らなくても AP を開発できるようになる．さらに EchonetLiteDevice サービスを Device Access[13]に準拠させておく．これによって，新しい種別の機器を ECHONET Lite バンドルが発見したときに自動的に対応するドライバを検索，及びダウンロードできる仕組みが容易に構成できる．Device Access サービスは，機器を表現する Device サービスと，機器に接続するための実装を担う Driver サービス等から構成されこれらを結合する機能を提供する．また，適切な Driver サービスがフレームワーク内に存在しない場合，サーバからこれを取得する機能も提供する．ここでは，EchonetLiteDevice サービスを Device サービス，機器操作 I/F ドライバを Driver サービスとすることで，新規機器に対応するドライバの自動取得を実現できる．

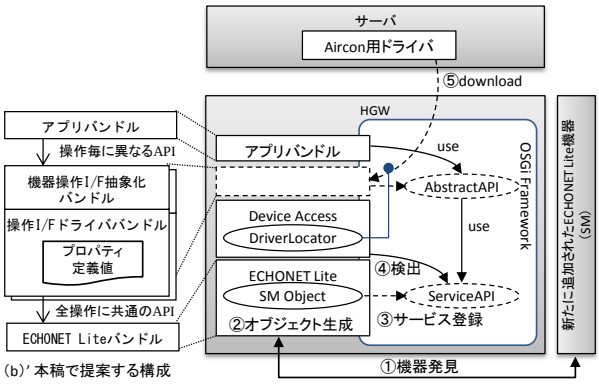

図 6　サービス API の実装方法とそれを利用した抽象化インタフェースの構成

Figure 6　Proposed implementation method of service API and the structure of abstract interface using that.

以上の構成とすることにより，各家庭の機器状況に応じた最小限の構成とし，省メモリを実現しつつ，機器追加時の AP 停止時間を最小限にすることができると考える．

4.3 ECHONET Lite 機器の識別

前述のとおり，ECHONET Lite 規格では，機器を一意にトレースする統一的な方法が規定されていない. このため，AP は，機器情報のトレースに機器毎に異なる情報を用いる必要が生じる可能性がある．例えば，機器 A は識別番号でのトレースを想定して，識別番号に有意な値が設定されるが，機器 B は MAC アドレスでの識別を想定して識別番号は 0 であるといったケースが考えられる．この場合，ユーザ（または設置者）は機器外装に印刷された識別番号や MAC アドレスを使って機器登録を行い，ユーザフレンドリーな情報（機器名称など）との紐付けを行うというユースケースが考えられる．このようなユースケースを実現するためには AP は，ユーザが登録した情報に応じて機器から必要な情報を取得し，かつそれをなんらかのデータベース（DB）にて管理する必要が生じる．この際，機器をトレースするための情報が一意ではないため，情報登録や検索の方法が複雑になってしまう可能性がある．

一方，OSGi フレームワークでは，登録するサービスにサービスプロパティと呼ばれる付加情報を設定しておくことで，登録されたサービスの検索及び追跡に強力なフィルタ機能を利用することができる．このフィルタは LDAP フィルタ[14]と呼ばれるフィルタであり，複数の条件を含む複雑な検索式を構成することができる．そこで，本バンドルを利用する AP が，EchonetLiteDevice サービスを取得する際に，このフィルタ機能を利用することができるように機器のトレースに利用される可能性のある情報を予め取得し，サービスプロパティとして登録しておくこととする（図 7）．

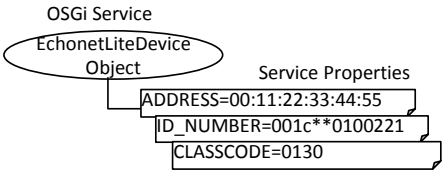

図 7　EchonetLiteDevice サービスのプロパティ構成

Figure 7　The structure of properties of EchonetLiteDevice service.

このように ECHONET Lite 機器の情報を OSGi のサービスとして管理することで，OSGi フレームワークが提供するサービスレジストリの仕組みを機器管理 DB として利用できる．さらに，機器管理 DB のインデックスとして ECHONET プロパティ情報を利用できるようにすることで，AP が機器を容易に識別できるようになると考えられる．

4.4 様々な下位通信層への柔軟な対応

ECHONET Lite バンドルでは下位通信層との結合を疎なものにするため，下位通信層とのインタフェースとして下位通信層の構成に依存しない抽象的なネットワークインタフェース（**下位通信層抽象化インタフェース**）のみ規定する．すなわち，下位通信層を実現するソフトウェアは ECHONET Lite バンドルの外部で前記インタフェースを実装したバンドルとする構成をとる．これにより，通信用リソース（ソケットや COM ポート）の確保，各種プロトコルやデバイス実装に依存したパケット化などは，前記インタフェースを implement したクラス実装にて対応するものとして，下位通信層と ECONET Lite ミドルウェア バンドルを切り離し，異なる物理層のネットワークインタフェースが複数存在する場合も対応可能にする. 本構成によって，下位通信層を変更・追加する場合でも ECHONET Lite バン

ドルの修正を行うことなく，新しく定義された下位通信層に対応することができる．

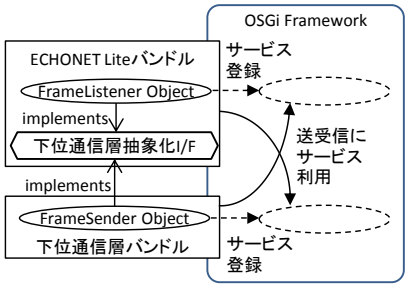

図 8　下位通信層バンドル構成

Figure 8　The structure of Low Layer Communication Interface Bundle

さらに，OSGi のバンドル追跡機能を利用して，下位通信バンドルの新規登録を追跡するようにすることで，ECHONET Lite バンドルを停止することなく，新しく追加された下位通信バンドルを利用して新しい下位通信層に接続された機器を発見・制御することが可能となると考えられる．

また，新規追加されたネットワークに対応した下位通信層バンドルをサーバからダウンロードして追加することも可能となる．この場合のシーケンス例を図 9 に示す．

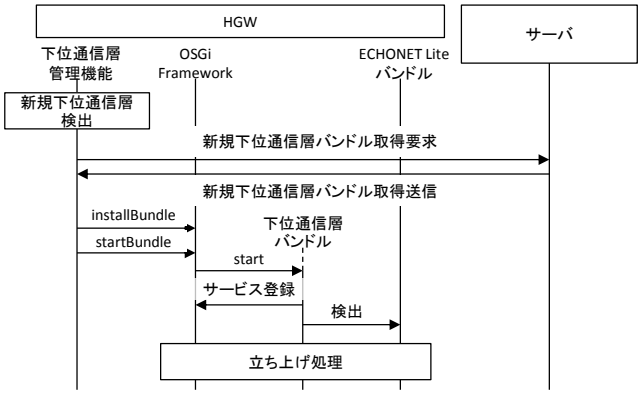

図 9　新規インタフェース追加時のシーケンス

Figure 9　Sequence of attaching new Low Layer Interface

例えば, SGW の USB ポートに ZigBee IP スタックを搭載したドングルを挿入した際，USB デバイスの挿入及びそれに対応した下位通信バンドルをダウンロードして実行する仕組みを作り込んでおくことで，以降，当該インタフェース（USB-ZigBee IP）を経由した機器の操作が可能になる．

このように，OSGi を活用してできるだけ柔軟に下位通信層を構成し，下位通信層の動的な変更も可能なバンドルを実現することにより，異なる下位通信層を柔軟に選択・追加できるとともに, SGW のように連続運転が要求される機器においても，機能を停止することなく新しい下位通信層を導入することが可能になると考えられる．

以上の検討結果から，ECHONET Lite バンドル及び関連するバンドルについて，以下の構成を提案する．

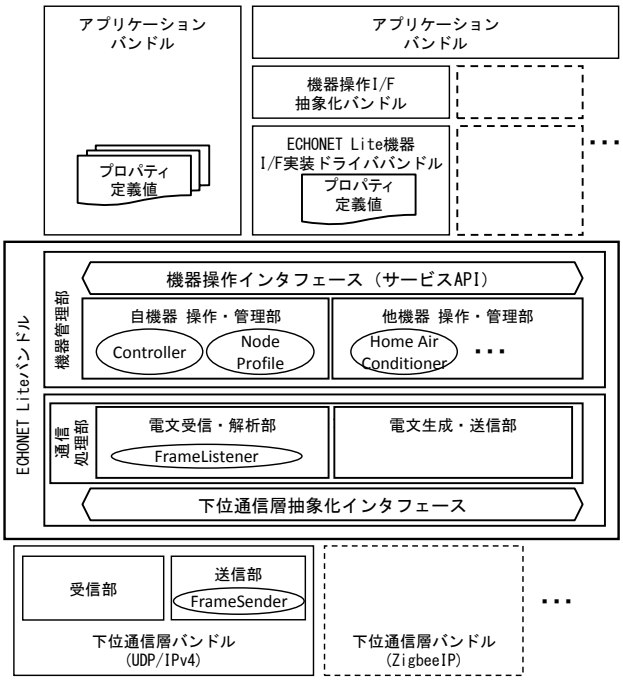

図 10　ECHONET Lite バンドルとその周辺構成

Figure 10　Structure of ECHONET Lite bundle and peripheral bundles

5. 実装と評価

4 章で提案した ECHONET Lite バンドルを PC 及び SGW 実機上で試作し，評価を行った．

表 3　試作機器仕様

Table 3　Specifications of Prototype Devices.

	SGW	PC
CPU	650MHz	2.7GHz
Memory	RAM:512MB, ROM:128MB	RAM:8GB HDD:280GB
LAN interface	4port(LAN), 1port(WAN)	1port
USB interface	2port	4port
OS	Linux	Windows7 64bit
Application Platform	JavaME (SuperJ Engine™)/ OSGi Framework (SuperJ Engine Framework™)	J2SE6 OSGi Framework (SuperJ Engine Framework™)

表 3 に評価に用いた機器の主要諸元を示す．また，図 11 に評価用システムの概要を示す. なお, PC と SGW 上では，下位通信層バンドルの一部を除いて実装の変更なくバンドルを動作させることができた．

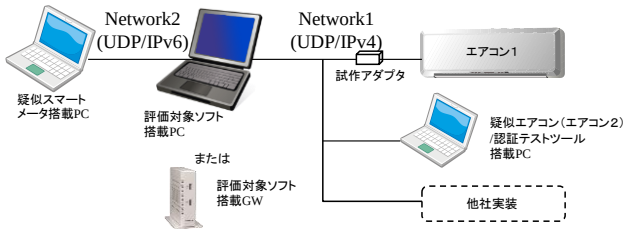

図 11　評価システム概要

Figure 11　Overview of Evaluation System

5.1 ECHONET Lite 規格実装上の課題解決に対する評価

ECHONET コンソーシアムの主催するプラグフェストに参加し，実装した試作ソフトウェアを他社実装と接続して ECHONET Lite バンドル単体で機器発見し，OSGi フレームワークに接続された機器のサービスが登録されること及び取得した ECHONET プロパティが前記サービスのサービスプロパティとして登録されていることを確認した．これにより，実装したバンドルのプロトコル実装が相互接続性を確保できていることが確認できた．

次に，評価システムを利用し，以下のシナリオで評価を実施した．本シナリオの目的は，3 章で目標とした要件 **R3** と **R4** について評価することである．

<前提条件>
・ユーザ宅の既存ネットワーク 1 にはエアコン 1，エアコン 2 が接続
・エアコン 1，2 は DHCP でアドレスを取得
・SGW とエアコン 1，2 は UDP/IPv4 で通信
<シナリオ>
①ユーザは 2 つのエアコンを区別しやすいように名称を登録
　エアコン 1：リビングエアコン
　エアコン 2：寝室エアコン
②登録にあたって，ユーザが知ることができる情報
　エアコン 1：MAC アドレス
　エアコン 2：識別番号
③SGW 上の AP は，エアコンから室温を取得し，前記名称とともにリアルタイム(1 秒更新)でユーザに提示.
④ユーザ宅に新規にスマートメータ(SM)が導入され，新規ネットワーク（UDP/IPv6）に接続
⑤SGW は SM からの電力量をリアルタイムでユーザに提示.

なお，評価用 GW は試作機で IPv6 未対応のため，評価は PC 上で実施した．

上記シナリオにおいて，機器を特定する情報として異なる識別情報を登録した 2 つの機器を正しく識別し，情報を取得できた．表 4 に本シナリオにおける AP の機器管理テーブル例を示す．これにより，利用する識別情報が異なる場合でも，OSGi フィルタを利用することで容易に機器のトレースができることが確認できた．そのため，要件 R3 は満足されていると言える．

表 4　OSGi のフィルタを利用した機器管理テーブル
Table 4 Device Management Table using OSGi Filter.

機器名称	検索フィルタ
リビングエアコン	(ADDRESS=00:11:22:33:44:55)
寝室エアコン	(ID_NUMBER=FE00000100****03)

また，シナリオの④では，手動で IPv6 に対応した下位通信層バンドルを新規追加し，すでに接続されたエアコンからの情報取得が途切ないことを確認した．これにより，ECHONET Lite バンドルを変更することなく複数の下位通信層に対応し，複数の異なる下位通信層が存在する環境でも容易にネットワークを追加し，かつ既存 AP に影響を与えないことが確認できた．そのため，要件 R4 は満足されていると言える．

5.2 AP 開発効率向上に関する評価

表 5 に, AP の開発効率向上を目的として ECHONET Lite バンドルに実装した他機器管理機能，自機器管理機能，及び機器の識別のために活用した OSGi フレームワークのフィルタ機能の概算ステップ数とステップ数から概算した工数を示す．ここに示すとおり，前述の構成とすることで，AP は約 5.4 人月の工数を低減できる．

表 5　共通機能の工数概算
Table 5　Production cost estimation of common functions .

	Step 数	工数(人月)
他機器管理機能	480	1.1
自機器管理機能	570	1.3
フィルタ実装部	1240	3.0

他機器管理機能を利用する場合，AP が機器を操作するために対象の機器インスタンスを特定するには，OSGi フレームワークからサービスを取得する手順に従って，約 10 ステップの実装のみを行えばよい．このことから，他機器管理機能の提供によって AP の開発を効率化できると言える．

また，認証項目を確認するツールを作成し，自機器管理機能を初めとした本バンドル実装の適合性の確認を行った．表 6 に ECHONET Lite における規格適合性認証の自己認証の項目数を示す．コントローラとして適合性認証を取得するためには, 少なくとも 88 の必須項目の確認を行う必要がある．このうち，ほぼ 2/3 の項目が ECHONET Lite フレーム受信時の要求仕様に関するものである．前述のように本バンドルでは自機器管理機能を備えているため，AP は改めて何らかの実装を行うことなく，大半の項目に適合させることができる．電文送信に関しても，不正な電文を送信しない機能を提供することから, AP に依存して（AP 追加・変更時に,）再テストが必要な項目は他機器への SET 要求送信に関する 4 つとなる．

表 6　ECHONET Lite における自己認証試験項目数
Table 6　Item number of Self Certification.

	全体	必須	受信	送信	
				自	他
フレーム処理	31	30	15	15	
オブジェクト処理	98	58	52	2	4
合計	149	88	68	20	

これは，省リソース化を実現するため，他機器オブジェクトのプロパティ（EPC 及び EDT）の構築は，バンドル外で行うこととしたため，上記認証項目のうち他機器オブジェクトに送信する EPC と EDT の整合性を確認する項目及び EDT の不正チェックを行う項目については，本提案のバンドルでは担保しないためである．しかし，88 項目中 2 項目を除いて，バンドルの提供する機能で対応できるため，本バンドルを利用した場合，認証取得は容易化でき，AP の開発を効率化できる．

5.3 省リソース化と AP 停止時間の評価

OSGi では，1 つのフレームワーク上で複数の AP バンドルを動作させることができ，AP ごとに JVM を立ち上げる

必要がないため，RAM 容量を節約できる．次に，ROM 容量の低減について評価するため，実行モジュールのサイズ比較を実施した．表 7 に結果を示す．

表 7　モジュールの ROM サイズ(KB)

Table 7　ROM size of software module (KB).

	本提案	A 社実装 (Java)	B 社実装 (C)
ECHONETLite バンドル	97	約 1,200[*1] 約 95[*2]	156[*1] 39[*2]
下位通信層バンドル	15		
家庭用エアコン操作 I/F ドライババンドル	50		
合計	162	約 1,200[*1] 約 95[*2]	156[*1] 39[*2]

*1 全機器オブジェクトを含む場合
*2 家庭用エアコンオブジェクトのみの場合

A 社実装では，すべての機器オブジェクト及びプロパティに対応しているため，モジュールサイズが大きなものとなっている．一方，本提案では，家庭用エアコンクラスのみを含む場合の構成で, 約 162KB でモジュールを構成できる．A 社実装でも対応するオブジェクトを家庭用エアコンのみにしてモジュールを構成した場合，約 95KB と省リソースの構成にできるが，今度は新しい機器追加時のソフトウェア更新による機能停止という課題が残る．

そこで，次に，機能（機器オブジェクト定義や下位通信層）追加時の AP 停止時間について検討した．機能追加方法毎の AP の停止時間を見積もるため，システムの再起動に要する時間，バンドルの更新（ここでは，ECHONET Lite バンドルが停止直前の状変通知を発行してから，再起動してインスタンス通知を発行するまでの）時間を計測した．表 8 にその結果を示す．

表 8　機能追加時の AP 停止時間の見積（秒）

Table 8　Estimation of stop time of AP to add new functions (sec).

機能追加方法	停止時間
ファームウェア更新（システム再起動）	86
バンドル更新（バンドルのみ再起動）	1.7
バンドル追加（本提案）	0

SGW のファームウェアを丸ごと更新する場合は, システムを再起動する必要のあるため, 約 86 秒と最も長く時間がかかる．バンドルを更新する場合は，バンドルのみ再起動すればよく，再起動と比較して短時間の AP 停止でよいと考えられる．しかし，再度機器を発見する時間や，バンドル更新時は更新対象のバンドルに依存するバンドルはすべて再起動が必要となるため，その時間を勘案すると，前記バンドル単体の再起動よりも長い時間が必要になる可能性が高い．一方で，バンドルを追加するだけの場合，稼働中の AP には影響を与えることなく，新しい機能を利用することが可能になる．

本提案では, プロパティ定義と Device Access サービスを活用したドライバダウンロード機構，及び下位通信層の別バンドル化によって，各家庭の状況に応じた最小限のソフトウェア構成にでき，かつ AP の停止時間を最小にできると言える．

6. おわりに

本研究では，国内の HEMS 標準プロトコルである ECHONET Lite に準拠したコントローラ用の OSGi バンドルについて検討し，PC 及び SGW 上に実装して評価を行った．設計にあたっては，マルチ AP を実現する SGW 上のミドルウェアとして実装するための課題及び規格実装上の課題に着目して 4 つの要件を抽出し，これを基に ECHONET Lite バンドル及びその周辺バンドルの構成を提案した．その際，OSGi フレームワークの持つ機能を最大限利用できるよう考慮した．その後，SGW 上に設計したバンドルを実装，評価し，提案する構成の有効性を確認できた．

参考文献

1) ECHONET コンソーシアム ECHONET Lite 規格書 Ver1.01 http://www.echonet.gr.jp/spec/spec_v101_lite.htm
2) 「平成 23 年度 エネルギー管理システム導入促進事業（HEMS 導入事業）－対象機器の公募－ 公募要領 」 http://sii.or.jp/hems/file/koubo.pdf
3) 丹康雄「ホームネットワーク(OSGi, ECHONET)モデルに基づく家庭内エネルギーマネジメント」, 情報処理学会 Vol.51 No.8, P959-965, 2010 年 8 月
4) OSGi Alliance「OSGi Service Platform Core Specification」 http://www.osgi.org/
5)ECHONET コンソーシアム「ECHONET 規格書 Version 3.21」 http://www.echonet.gr.jp
6) 大和ハウス工業株式会社「平成 21 年度スマートハウス実証プロジェクト報告書 第 2 章 テーマ 2-1 : マッシュアップを促進するホームサーバ向け統合 API の開発実証およびテーマ 3-1 : マルチベンダによる家電・設備機器統合コントロールシステムの開発」, 2010 年 3 月 http://www.jipdec.or.jp/dupc/forum/eships/results/doc/h21project_report1-2.pdf
7) Fukushima, K.; Tanaka, Y.; Kato, H.; Ishikawa, N.; , "Home Network System for Gas Appliances Using PUCC Technologies," Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE , vol., no., pp.1-5, 9-12 Jan. 2010
8) 日新システムズ　EW-ENET Lite http://www.co-nss.co.jp/p-org/enetlite.html
9) SonyCSL「OpenECHO」 https://github.com/SonyCSL/OpenECHO
10) ECHONET コンソーシアム「ECHONET 機器認証試験仕様書」 ECHONET Lite 規格 Ver.1.0*用 http://www.echonet.gr.jp
11) TTC TR-1043:ホームネットワーク通信インタフェース実装ガイドライン
12）宮本善則他「ECHONET Lite による蓄電池管理システムの開発」日本学術振興会産学協力研究委員会 第 31 回インターネット技術第 163 委員会研究会
13)OSGi Alliance「OSGi Service Platform Release4 Device Access Specification Version1.1」http://www.osgi.org/
14)RFC1960　A String Representation of LDAP Search Filters http://www.ietf.org/rfc/rfc1960.txt