

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

長谷川剛, 村田正幸 (大阪大学大学院情報科学研究科)
増尾剛, 中村二郎, 松岡茂彦 (NTT)
松田和浩 (NTTアドバンステクノロジー)

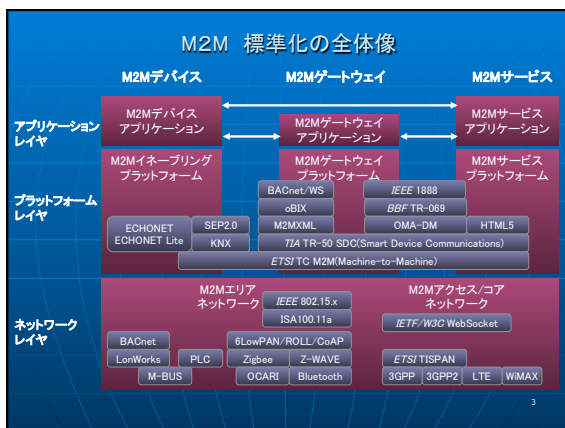
1

M2M, IoT, WoT

- IoT: 地球上のあらゆるモノ=デバイスがインターネットにつながる世界
- WoT: Webインタフェースですべてがつながる世界
- 2020年には、500億個のデバイス
 - Cisco Systems社の予想
 - Sparked: 畜牛センサ, Corventis: 健康監視, Gemalt: 自動車緊急通報
- 爆発的なデータ量

2010 2015 2020

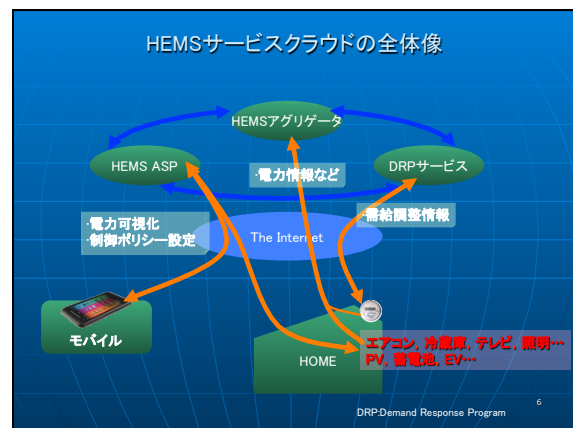
2



EMS (エネルギー管理システム) の動向

- オープン技術によるマルチベンダ化
- システムの広域化
 - BEMS (Building Energy Management System)
 - BACnet/WS, oBIX, IEEE 1888...
 - HEMS (Home Energy Management System)
 - ECHONET, ECHONET Lite (日本), SEP2.0 (米国), KNX (EU)
 - 閉域 (HAN) からWANへ

4



リアルタイム性についての考察

- HEMSに関係するような事例
- PPSにおける需給バランス調整 (30分同時同量制御) → 5分程度の分解能
- 電力ピークカットシステム → 1分程度の分解能

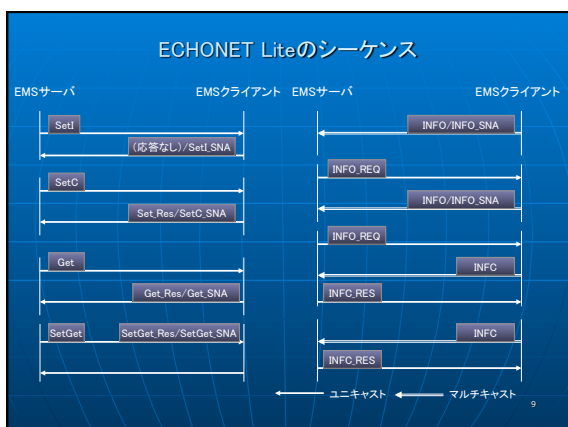
インターネット越し・Web化でもサービスに耐える可能性

Webサービス化の検討

- WSDL/XSDによるサービス定義 -

- ECHONET/ECHONET Liteの成果を活用
 - ・ 網羅的に家電製品のプロパティを定義
 - ・ 下位層プロトコルからの独立: ECHONET Lite
 - UDP/IP, Ethernet, Wi-Fi, Bluetooth, ZigBee, ...
- Webサービス化
 - ・ WSDL, XSDでECHONET Liteを再定義
 - ・ プロトコルシーケンス → WSDLで記述
 - ・ メッセージ(プロパティを含む) → XSD
- 双方向通信
 - ・ 制御信号・制御結果および自律的な状態通知のために双方向通信が必要。

WSDL: Web Services Description Language
XSD: XML Schema Definition



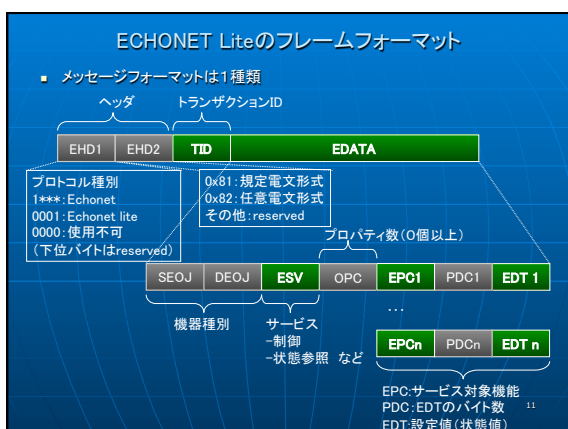
WSDL記述の例

- WSDL2.0の記述要素
 - ・ interface: Webサービスの抽象インターフェース (抽象operationの集合)
 - ・ operation: input message/output message/fault messageからなる処理の1単位

```

<wsdl:interface name="EchonetInterface">
  <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="SetI">
    <wsdl:input element="msg:SetI"/>
    <wsdl:fault element="msg:SetC_SNA"/>
  </wsdl:operation>
  ...
</wsdl:interface>
    
```

インタフェース: EchonetInterface
 オペレーション: SetC (入力メッセージ, 出力メッセージ, フォルト・メッセージ)
 オペレーション: SetI (入力メッセージ, フォルト・メッセージ)



xsd記述 家庭用エアコンの例(1)

```

<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>
    
```

プロパティをセットする型の定義
 トランザクションID
 サービス種別
 動作状態
 動作モード (冷房/暖房...)

メッセージの例

■ エアコン・オンのメッセージ

```

<ns1:SetC 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>
  
```

⇒JSON表記

```

{"TransactionId": 1,
 "ESV": "SetC",
 "operationStatus": "ON"}
  
```

■ エアコン・オンの応答メッセージ

```

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

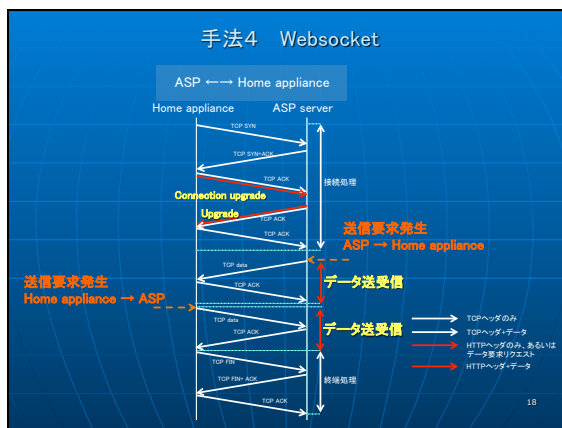
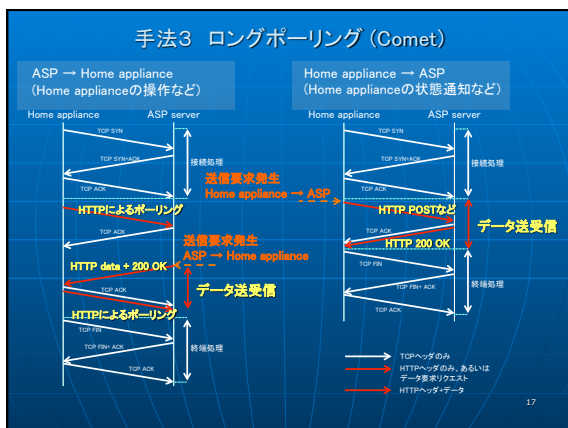
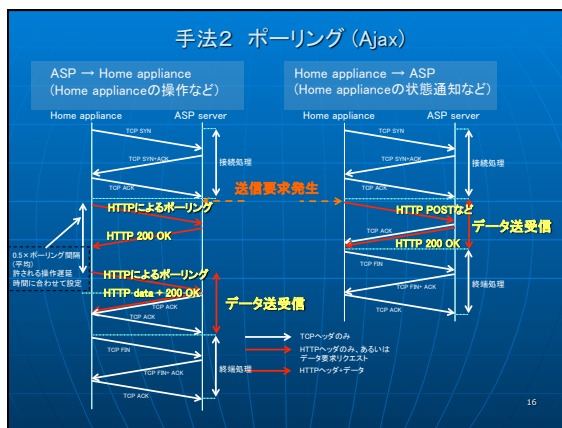
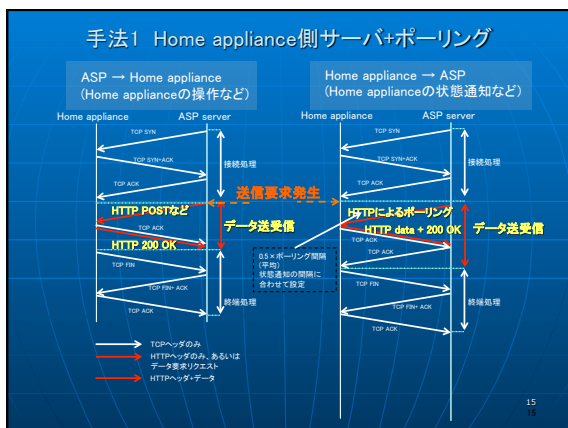
⇒JSON表記

```

{"TransactionId": 1,
 "ESV": "SetC_Res",
 "operationStatus": "ON"}
  
```

Web技術による双方向通信の実現手法の検討

方式	概要	利用シーンから見た得失	
		ホーム側	ASP側
【手法1】 制御サーバ+ポーリング		ホーム側に固定IPアドレスが必要 ホーム側の状態を任意のタイミングで通知不可	ASP側のイベントを任意のタイミングで通知可能
【手法2】 制御サーバ+ポーリング (Ajax)		ホーム側の情報を任意のタイミングで通知可能	ASP側のイベント通知に遅れが生じる コネクション数に制限
【手法3】 制御サーバ+ロングポーリング (Comet)		ホーム側の情報を任意のタイミングで通知可能	ASP側からのイベント任意のタイミングで通知可能 コネクション数の維持コストが大(裏書き保存) 複数ドメインの連携時にセキュリティ上の不安がある
【手法4】 制御サーバ+WebSocket		ホーム側の情報を任意のタイミングで通知可能	ASP側からのイベントを任意のタイミングで通知可能 コネクション数の維持コストが小(裏書き保存)



オーバーヘッド評価

手法	通信時間		パケット量				コネクション 設定	コネクション 開放
	ASPからのイベント通知	Home applianceからの状態通知	ASPからのイベント通知(下り通信)	Home applianceからの状態通知(上り通信)	ASP → Home appliance	Home appliance → ASP		
手法1 Home appliance側サーバへのポーリング	IRTT	1.5RTT	HTTP data	HTTP header	HTTP header (a)	HTTP data	1.5RTT / 3 TOP headers	1.5RTT / 3 TOP headers
手法2 ポーリング(Ajax)	データ送信: IRTT ポーリング: IRTT	IRTT	HTTP data (a)	HTTP header	HTTP header	HTTP data	1.5RTT / 3 TOP headers	1.5RTT / 3 TOP headers
手法3 ロングポーリング(Comet)	IRTT	IRTT	HTTP data	HTTP header	HTTP header	HTTP data	1.5RTT / 3 TOP headers	1.5RTT / 3 TOP headers
手法4 Websocket	IRTT	IRTT	TCP data	TCP header	TCP header	TCP data	2.5RTT / 4 TOP headers + 2 HTTP headers	1.5RTT / 3 TOP headers

* TCP ACKの piggyback はできるだけ行われると仮定
 (a) はポーリングのパケットを表す

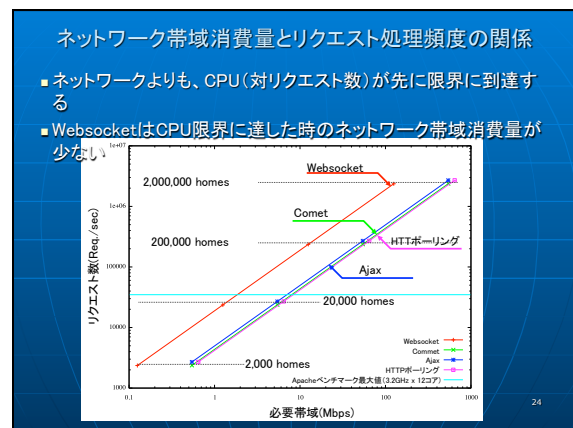
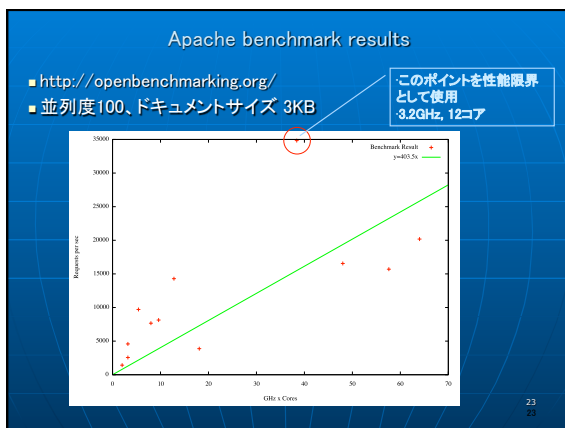
- ### 数値例
- パラメータ
 - ・ 収容戸数: 2,000世帯
 - ・ 1軒あたりの機器と通信頻度
 - 家電: 10台
 - ・ 上り: 1分に1回、運転状態(ON/OFF)や電力の通知
 - ・ 下り: 1時間に1回(家電機器の制御プラン指示、ユーザによる家電操作)
 - スマートメータ: 1台
 - ・ 上り: 5分に1回、電力の通知(30分同量制御を想定)
 - 各方式のオーバーヘッドを評価
 - ・ ポーリングを用いる方式のポーリング間隔は、通信頻度に合わせる
 - ・ ユーザによる家電操作のためのポーリングは1分に1回と想定
 - ・ パケットサイズ
 - ECHONET Liteのデータ: 500バイト
 - HTTPヘッダ: 200バイト、TCPヘッダ: 40バイト

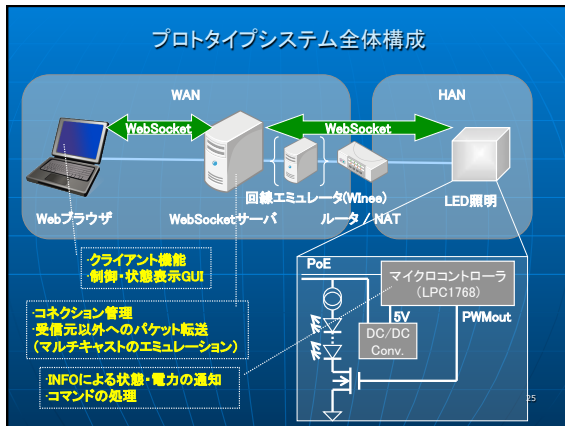
評価結果

- 2,000世帯の場合、どの方式を用いてもネットワークトラフィック量、CPU量いずれも問題のない範囲
- 家電操作を行う側をポーリングによって実現する手法1は、Home→ASPのトラフィックがポーリングによって増大する
- Websocket化により、ネットワークトラフィックを約 1/5程度、同時アクティブセッション数を約1/2程度に削減可能

手法	下り通信トラフィック量 (Base)	上り通信トラフィック量 (Top)	合計トラフィック量 (Mbit)	ASP → Home appliance	Home appliance → ASP	ASP → Home appliance	Home appliance → ASP	同時アクティブセッション数	ASPサーバの処理能力 (Requests/sec)
手法1 Home appliance側サーバへのポーリング	31111.1	8888.96453333.3	23185.6	0.645	0.031	22000	34.3	2687.1	
手法2 ポーリング(Ajax)	31111.1	30111.1	63777.8	0.543	0.532	22000	67.5	2684.4	
手法3 ロングポーリング(Comet)	32888.8	10888.753777.8	23185.6	0.544	0.032	22000	68.1	2372.1	
手法4 Websocket	24000.0	1777.8107855.6	19610.0	0.128	0.020	22000	34.2	2361	

- ### 収容世帯数に対するスケーラビリティ
- 収容する世帯数を増加させた場合に、ネットワーク帯域とCPU性能のどちらにボトルネックが先に発生するかを検証
 - ASPサーバの能力の指標としてApache webサーバのベンチマーク結果を使用
 - ・ サーバスペックとベンチマーク結果 (requests/sec) の関係
 - ・ 収容世帯数に増加にともなうASPサーバの処理頻度とネットワーク帯域の関係





- ### プロトタイプシステム仕様
- LED照明
 - ・ プロセッサ: NXP LPC1768 (コア: ARM-M3, 100MHz, RAM: 32kB, Flash: 512kB)
 - ・ C++ 200行 (Websocket-clientライブラリ)
 - ・ 使用メモリ: Flash: 89kB(17%), RAM: 6.8kB(21%)
 - Websocketサーバ
 - ・ プロセッサ: Intel Atom N280(1.66GHz)
 - ・ OS: Ubuntu10.4
 - ・ Python 72行(Tornado httpserver.websocket)
 - WebClient
 - ・ Firefox 14.0.1, Chrome 23.0.1271.91m
 - ・ Index.html: 300行 (JavaScriptを含む, jQuery・Smoothie Chartsを使用)



- ### 課題
- メモリ消費量の評価
 - 大規模実験による評価
 - スケーラビリティに対する考慮
 - ・ ゲートウェイでのフローアグリゲーション等
 - 他トランスポートプロトコルとの通信

- ### まとめ
- WoTによるHEMSサービスのクラウド化を提案
 - 制御プロトコル
 - ・ ECHONET/ECHONET Liteの成果を活用
 - 双方向通信の実現方法
 - ・ 方式比較, WebSocketの優位性
 - デモ
 - 課題

- ### 参考
- oBIX: open Building Information eXchange
 - OASIS
 - WEBサービス(プロトコル/バインド)はHTTPおよびSOAP
 - BACnet/WS
 - ASHRAE
 - データは文字列(非XML)
 - TR-069
 - BBF フロードバンドフォーラム
 - SOAP 双方向
 - ホームルータと自動設定サーバの通信: ネットワーク管理, 設定自動化
 - OMA-DM: Open Mobile Alliance-Device Management
 - Open Mobile Alliance
 - デバイス管理機能
 - TR-50: Smart Device Communications
 - TTA: 次世代通信工業会
 - アプリケーション, ミドルウェア
 - 共通データフレームワークの標準化
 - ISA100.11a
 - ISA International Society of Automation
 - IEEE 802.15.4準拠
 - M-BUS
 - Meter-Bus