



Title	ネットワーク境界越が可能なIEEE1888 over Websocketの提案と実装
Author(s)	樽谷, 優弥; 村田, 修一郎; 松田, 和浩 他
Citation	システム制御情報学会研究発表講演会講演論文集. 2016, 60, p. 6-6
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/71022">https://hdl.handle.net/11094/71022</a>
rights	
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

# ネットワーク境界越が可能な IEEE1888 over WebSocket の提案と実装

## IEEE1888 over WebSocket for communicating across a network boundary

大阪大学 ○ 樽谷 優弥, 村田 修一郎, 松田 和浩, 松岡 茂登

Osaka University ○ Yuya Tarutani, Shuichirou Murata, Kazuhiro Matsuda, and Morito Matsuo

**Abstract** There is a network that restrains communication between inside equipment and outside equipment because of ensuring the security. In this case, each equipment cannot communicate with each other by a traditional IEEE1888 protocol. In this paper, we propose *IEEE1888 over WebSocket* for communicating across a network boundary. IEEE1888 over WebSocket provides an intercommunication between equipment across a network boundary without changing the system architecture of IEEE1888. We implemented and tested a proposed system.

### 1 はじめに

インターネット技術の普及に伴い、センサ機器やアクチュエータ機器との通信にインターネット技術が基盤として用いられるようになってきている。この動きは Internet of Things (IoT) と呼ばれる分野に発展し、多くの注目を集めている。特に、ネットワークに接続されているセンサ機器やアクチュエータ機器から得られる情報を用いて、電力使用状況の可視化や、接続機器の制御等を行うことによりエネルギー消費の最適化を実現するエネルギー管理システム (EMS: Energy Management System)[1-4] が注目を集めている。EMS には、家庭向けの HEMS (HomeEMS) [1] や、ビル向けの BEMS (Building EMS) [2]、さらにはより広域を対象とした CEMS (Community EMS) [3] があり、多数のセンサ機器やアクチュエータ機器等が接続され、情報のやり取りがなされる。しかしながら、多種多様な機器が存在する状況では様々なプロトコル [5, 6] が用いられており、その実現は容易ではない。

この EMS 向けの通信プロトコルとして IEEE1888[7] が挙げられる。IEEE1888 はセンサ機器やアクチュエータ機器の情報収集や制御等を SOAP 通信 [8] で交換するプロトコルとして設計されており、その通信には Hyper Text Transfer Protocol (HTTP) が用いられる。これにより、EMS の適用範囲をインターネット環境に拡大すること、および様々な情報システムとの親和性を確保している。このように IEEE1888 は、インターネット基盤が普及した現在では、広域、大規模な EMS に適した EMS 用プロトコルであると言える。しかしながら、セキュリティ等の理由により通信が制限されたネットワーク環境が存在する。例えば、ファイアウォールによって

内部からは通信が行うことができる一方で、外部からはネットワーク内の機器に対して通信できない状況が考えられる。そのような通信が制限されたネットワーク環境下の機器に対しても、ネットワーク外部の機器からセンサ機器の情報の収集やアクチュエータ機器の制御を行うことが必要とされる。従来の IEEE1888 の仕組みでは、このような環境下に存在する IEEE1888 コンポーネントに対して、そのネットワーク外部に存在する IEEE1888 コンポーネントから接続することはできない。

本研究では、このような通信が制限されたネットワーク環境において、IEEE1888 コンポーネント同士による相互接続や通信の実現する方法を提案する。提案方式では、通信制限のあるネットワーク環境下の IEEE1888 サーバの URL を代理する URL をそのネットワークの外部に用意し、かつ外部のネットワークにある IEEE1888 サーバの URL を代理する URL を通信制限のあるネットワーク内に用意することにより、双方のネットワークにある IEEE1888 コンポーネント同士を相互に到達可能にする方法を採用する。また、代理 URL を提供するコンポーネント同士で WebSocket プロトコル [9] を用いたコネクションを確立することで、通信の双方向性をもたせ、双方のネットワークにある機器からの相互アクセスを実現する。提案方式は、IEEE1888 サーバの URL の代理、および通信制限のあるネットワーク環境とその外部ネットワークの間を結ぶ WebSocket コネクションを構築するためのコンポーネントを追加するだけであり、IEEE1888 を使用した既存のネットワークに組み込みが容易である。

本稿の章構成は以下のようになる。2 章では従来の IEEE1888 プロトコルの概要について述べる。3 章では提案方式である IEEE1888 over WebSocket について述

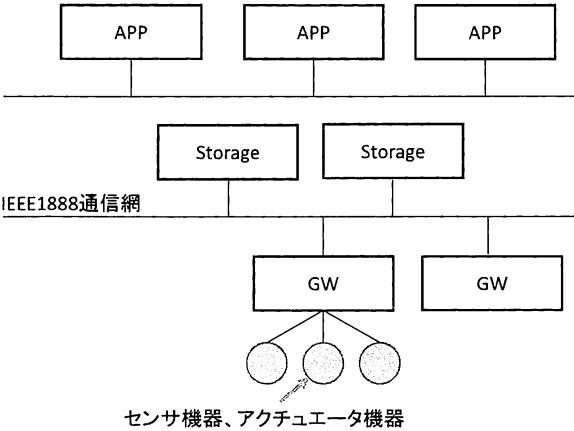


図 1: IEEE1888 のアーキテクチャ例

表 1: IEEE1888 の通信手段

通信手順	内容
FETCH	ストレージ等からデータを取得
WRITE	ストレージ等へデータを書き込む
TRAP	情報の変更等に対し、指定したコンポーネントに通知するように設定

べる。4 章では、提案方式のプロトタイプを実装し、その接続試験について述べる。最後に本稿のまとめと今後の課題について述べる。

2 IEEE1888 プロトコル

IEEE1888 は大規模エネルギーマネジメントシステムを構築、運用するためのプロトコルとして標準化されたプロトコルである。図 1 に IEEE1888 のシステムアーキテクチャを示す。IEEE1888 のシステムアーキテクチャは、各種センサやアクチュエータ機器を扱うゲートウェイ (GW)、データを蓄積するためのストレージ (Storage)、データの加工等の様々な役割を行うアプリケーション (APP) に分類される 3 種類の IEEE1888 コンポーネントとその間を結ぶネットワークから構成される。これらの IEEE1888 コンポーネント間で IEEE1888 プロトコルを用いてデータ交換が行われる。

各 IEEE1888 コンポーネント間では通信手段として表 1 FETCH、WRITE、TRAP の 3 つが定義されている。各 IEEE1888 コンポーネントは識別子として URL を持っており、これらの通信手順は各 URL を指定することで行われる。IEEE1888 プロトコルでは、SOAP over HTTP に基づいて通信を行っている。IEEE1888 では

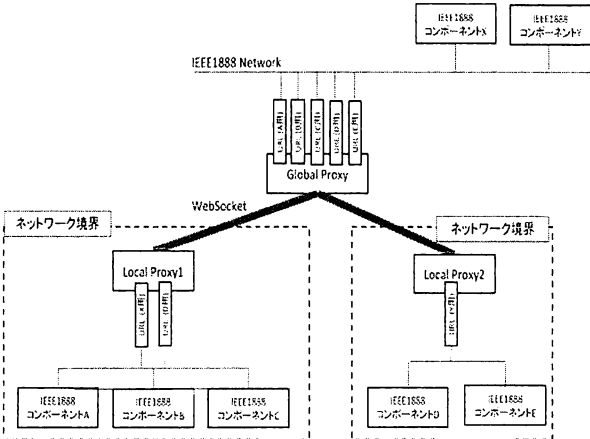


図 2: IEEE1888 over WebSocket のシステムアーキテクチャ例

SOAP を採用することにより、Web 関連技術やデータベースなどの様々な情報システムとの親和性を高めている。また、HTTP による通信が可能な既設の LAN や広域ネットワーク上でも容易に構築可能である。

しかしながらセキュリティ等の理由によって、ファイアウォール等により通信が制限されたネットワーク環境内に存在する IEEE1888 コンポーネントが存在する。このような場合、ネットワーク内部から HTTP 通信が可能であったとしても、その外部のネットワーク下の IEEE1888 コンポーネントからは、HTTP 通信ができない状況がある。そのため、ネットワーク外部の IEEE1888 コンポーネントからは、コンポーネント間の相互通信を提供できない。そこで本稿では、このような状況でも IEEE1888 による通信を行うことができる方式を提案する。

3 IEEE1888 over WebSocket

3.1 システムアーキテクチャ

図 2 に提案する IEEE1888 over WebSocket のシステムアーキテクチャの例を示す。図 2 に示すように、IEEE1888 over WebSocket では、通信が制限されたネットワーク環境下 (以下ローカルネットワーク) の IEEE1888 コンポーネントとその外部ネットワーク側 (以下グローバルネットワーク) のコンポーネントとの相互通信のために、各ネットワーク間を WebSocket プロトコルを用いたコネクション (以下 WebSocket コネクション) で接続する。WebSocket コネクションによって、ローカルネットワーク環境下の IEEE1888 コンポーネントに対して、グローバルネットワーク側の IEEE1888 コンポーネ



図 3: グローバルネットワーク側コンポーネントからみた図 2 の IEEE1888 通信網

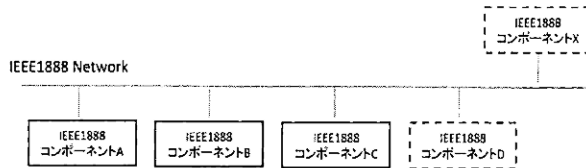


図 4: ローカルネットワーク側コンポーネントからみた図 2 の IEEE1888 通信網

ントから通信を行うことができる。このネットワーク間に WebSocket コネクションを構築する機器をそれぞれ *Global Proxy* および *Local Proxy* と呼ぶ。

### 3.1.1 Global Proxy

Global Proxy はグローバルネットワーク側に設置され、WebSocket 用の URL を持っている。それとは別に、ローカルネットワーク環境下の各 IEEE1888 コンポーネントに対応する複数の Proxy 用の URL を持っている。各 URL 宛に送られた通信は WebSocket コネクションを通じて Local Proxy に転送され、表 2 に示すような変換表に基づいて、対応する IEEE1888 コンポーネントに送信される。このように Global Proxy がローカルネットワーク環境下の IEEE1888 コンポーネントの代理をすることによって、グローバルネットワーク側にある IEEE1888 コンポーネントからローカルネットワーク環境下の各コンポーネントが図 3 のように見える。図中の実線と点線で表されたコンポーネントは、それぞれグローバルネットワーク側の IEEE1888 コンポーネントが Global Proxy を用いずに接続が可能なコンポーネント、および Global Proxy を介して接続するコンポーネントである。

### 3.1.2 Local Proxy

Local Proxy はローカルネットワーク環境下に設置される。Local Proxy は、Global Proxy との間に WebSocket コネクションを構築し、Global Proxy から転送された通信を、表 3 に示すような変換表に基づいて、送られた

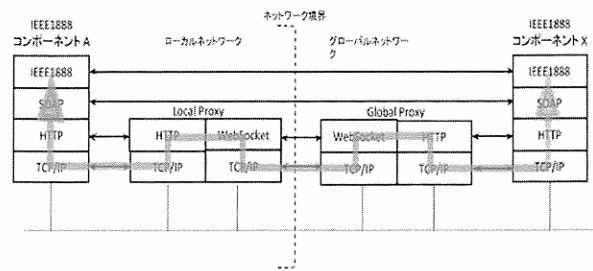


図 5: IEEE1888 over WebSocket のレイヤ・アーキテクチャ

Proxy 用の URL に対応するローカルネットワーク環境下の IEEE1888 コンポーネントに送信する。Local Proxy も Global Proxy と同様に Proxy 用 URL を提供する機能を持つ。例えば、図 2 に示すように、Local Proxy1 が提供している「URL(X 用)」「URL(D 用)」、および Local Proxy2 が提供している「URL(Y 用)」に対して通信を行うと、各 Proxy を経由して、それぞれ対応する IEEE1888 コンポーネントとの通信ができる。そのため、Local Proxy1 が設置されているネットワークの各コンポーネントからは他のコンポーネントがそれぞれ図 4 のように見える。図中の実線と点線で表されたコンポーネントは、それぞれ LocalProxy1 が所属するネットワーク内部の IEEE1888 コンポーネントが直接アクセスできるコンポーネント、および Local Proxy1 を介してアクセスできるコンポーネントである。

### 3.2 レイヤアーキテクチャ

IEEE1888 over WebSocket ではコンポーネント-プロキシ間と Global Proxy-Local Proxy 間 (Proxy 間) で異なる通信方法が用いられる。コンポーネント-プロキシ間では通常の IEEE1888 におけるコンポーネント間の通信を行う。Proxy 間では、TCP/IP のセッションを構築し、HTTP リクエストによってコネクションの UPGRADE 要求を行い、Proxy 間の TCP/IP のセッション上に WebSocket コネクションを構築する。WebSocket コネクションの構築後、各コンポーネントから受信した HTTP メッセージを Proxy 間の接続を通して転送することにより通信を実現する。

図 5 に IEEE1888 over WebSocket のレイヤ・アーキテクチャの例を示す。図 5 に示すように IEEE1888、SOAP のレイヤでは、通信の方向にかかわらず、各コンポーネントから送られてきたメッセージはプロキシを通じて宛先のコンポーネントに送られる。通常、各プロキシでは、これらのレイヤのメッセージは処理を行うことなく、透

表 2: Global Proxy のエンドポイント変換表の例

Global Proxy が提供するエンドポイント URL	実際の接続先 IEEE1888 コンポーネントのエンドポイント URL
http://globalproxy.example.com/A	http://local-ieee1888.exmaple.com/A
http://globalproxy.example.com/B-8888	http://local-ieee1888.example.com:8888/B
http://globalproxy.example.com/192-C	http://192.168.0.10/C

表 3: Local Proxy のエンドポイント変換表の例

Local Proxy が提供するエンドポイント URL	実際の接続先 IEEE1888 コンポーネントのエンドポイント URL
http://localproxy.example.com/A	http://global-ieee1888.example.com/A
http://localproxy.example.com/B-8888	http://global-ieee1888.example.com:8888/B
http://localproxy.example.com/1-C	http://global-ieee1888-1.example.com/C

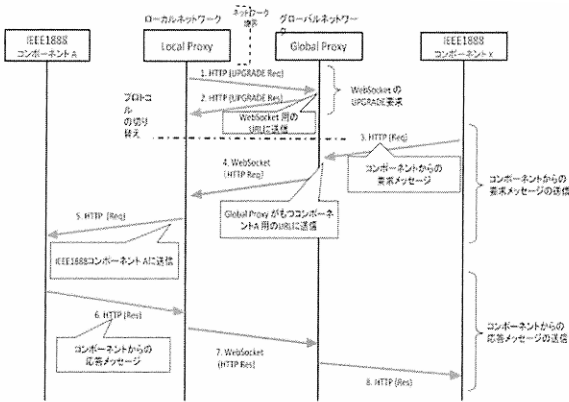


図 6: IEEE1888 over WebSocket の通信手順の例

過的に宛先のコンポーネントへ送信される。プロキシは HTTP ヘッダーを参照し、どのプロキシやコンポーネントにメッセージを転送するかを決定する。図 5 において、コンポーネント X からコンポーネント A への通信の場合、宛先の URL は Global Proxy のもつ URL であり、Global Proxy はそれを参照して、Global Proxy がどの Local Proxy に転送するか、また Local Proxy がどのコンポーネントに送信するかを決定する。Local Proxy も Global Proxy と同様の転送用 URL を提供し、宛先を実際のコンポーネントの URL に変換したのち、Global Proxy に転送する。WebSocket レイヤでは、コンポーネントから受け取った HTTP メッセージを WebSocket の Data フレームに格納して対象のプロキシに転送する。

3.3 通信手順

図 6 に IEEE1888 over WebSokcet を用いた通信手順の例として、グローバルネットワーク側のコンポーネントからローカルネットワーク側のコンポーネントへの通信手順を示す。各手順の具体的な動作内容は以下の通りである。

1. Local Proxy が Global Proxy に対して、WebSocket コネクションの接続要求 (UPGRADE Req) を送信する。
2. Global Proxy は Local Proxy に UPGRADE Res を返す。結果として、Local Proxy、Global Proxy 間に WebSocket コネクションが確立される。
3. IEEE1888 コンポーネント X は Global Proxy が持つコンポーネント A 用の URL 宛に要求メッセージを送信する。
4. Global Proxy は受け取った要求メッセージを WebSocket のコネクションを用いて Local Proxy に転送する。
5. Local Proxy は IEEE1888 コンポーネント A に対して、要求メッセージを送信する。
6. IEEE1888 コンポーネント A は受信した要求メッセージに対しての応答メッセージを、Local Proxy に返す。
7. Local Proxy は WebSocket コネクションを通じて 応答メッセージを Global Proxy に転送する。

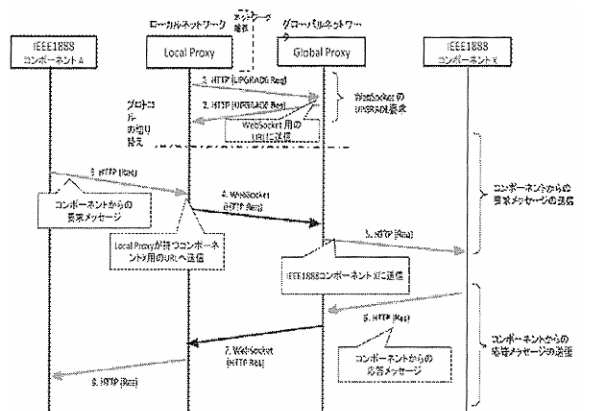


図 7: IEEE1888 over WebSocket の通信手順の例

- Global Proxy は受け取った応答メッセージを IEEE1888 コンポーネント X に対して送信する。

また、図 7 にローカルネットワーク側のコンポーネントからグローバルネットワーク側のコンポーネントへの通信手順を示す。各手順の具体的な動作内容は以下の通りである。

- Local Proxy が Global Proxy に対して、WebSocket コネクションの接続要求 (UPGRADE Req) を送信する。
- Global Proxy は Local Proxy に UPGRADE Res を返す。結果として、Local Proxy、Global Proxy 間に WebSocket コネクションを確立する。
- IEEE1888 コンポーネント A は Local Proxy がもつコンポーネント X 用の URL 宛に要求メッセージを送信する。
- Local Proxy は WebSocket を通じて Global Proxy に要求メッセージを転送する。
- Global Proxy は転送された要求メッセージを IEEE1888 コンポーネント X に送信する。
- 要求メッセージを受け取った IEEE1888 コンポーネント X は Global Proxy に対して応答メッセージを返す。
- Global Proxy は Local Proxy に対して応答メッセージを転送する。
- Local Proxy は転送された応答メッセージを IEEE1888 コンポーネント A に対して返送する。

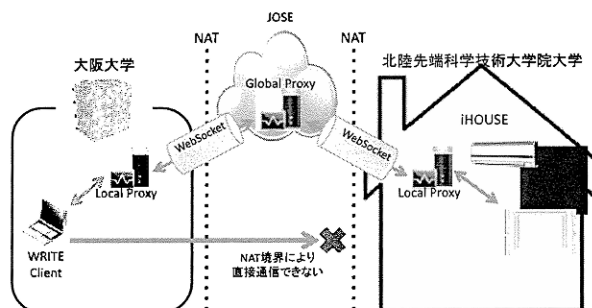


図 8: 動作検証概略

## 4 動作検証

本稿で提案した IEEE1888 over WebSocket のプロトタイプを実装し、その動作検証をおこなった。動作検証のために、3 拠点間における接続テストを行った。図 8 に行った接続テストの概略を示す。本検証では、大規模オープンテストベッドである、JOSE (Japan-wide Orchestrated Smart/Sensor Environment)[10] に Global Proxy を配置し、大阪大学および北陸先端科学技術大学院大学に Local Proxy を設置した。そして、大阪大学内のネットワークの IEEE1888 コンポーネントから、各 Proxy を通じて、北陸先端科学技術大学院大学内にあるテストベッドである iHouse[11] 内の機器に対して、WRITE 要求を送り、機器を制御することで接続試験を行った。本検証における WRITE クライアントおよび GW サーバは Python で実装し、各 Proxy は Java を用いて実装した。WebSocket コネクションはそれぞれの Local Proxy 起動時に Global Proxy に対して構築する。また、大阪大学側の Local Proxy の Proxy 用 URL が北陸先端科学技術大学院大学内の GW サーバの URL に対応するように設定を行った。

結果としては、大阪大学内の WRITE クライアントから Local Proxy の代替 URL を指定して WRITE 命令を送信することで、北陸先端科学技術大学院大学に設置した GW サーバに接続している機器を制御できることを確認した。すなわち、本稿で提案した IEEE1888 over WebSocket が正しく動作し、通常の方法では通信不可能な IEEE1888 コンポーネントに対しても通信が可能となることが確認できた。

## 5 おわりに

本稿では、セキュリティ等の理由によって通信が制限され、通常の方法では相互通信ができない IEEE1888 コンポーネント間において、相互通信の実現のために、

IEEE1888 over WebSocket を提案した。本稿で提案した IEEE1888 over WebSocket は既存の IEEE1888 ネットワークに WebSokcet コネクションを構築するための 2 種類の Proxy を追加することで、通常の方法では相互通信ができない IEEE1888 コンポーネント間においても相互通信を実現することができる。本稿では、提案した IEEE1888 over WebSocket のプロトタイプを実装し、その動作検証を行った。その結果、通常の方法では通信不可能な IEEE1888 コンポーネントに対しても通信が可能となることが確認した。本稿で提案している IEEE1888 over WebSokcet では、各 Proxy を経由することによる遅延への影響や構築する WebSocket コネクションの数による影響を考慮していない。そのため、大規模エネルギーマネジメントシステムへの適用のためにはこれらの検討を行うことが課題として挙げられる。また、本稿で提案している IEEE1888 over WebSocket は、IEEE の標準化を視野にいて活動を行っている。

## 謝辞

本研究の一部は、情報通信研究機構 (NICT) の委託研究「新世代ネットワークの実現に向けた欧州との連携による共同研究開発および実証」の支援による。ここに記して謝意を表す。

## 参考文献

- [1] H. T. Nguyen, D. Nguyen, and L. B. Le, "Home energy management with generic thermal dynamics and user temperature preference," in *Proceedings of IEEE SmartGridComm*, pp. 552-557, Oct. 2013.
- [2] K. Park, Y. Kim, S. Kim, K. Kim, W. Lee, and H. Park, "Building energy management system based on smart grid," in *Proceedings of IEEE INTELEC*, pp. 1-4, Oct. 2011.
- [3] 鈴木浪平, 田熊良行, 井野本正樹, 谷本昌彦, "地域エネルギーマネジメント技術 (CEMS)," 三菱電機技法.
- [4] 中村雅之, 竹内章, 榎本裕幸, "データセンタエネルギー管理システム (DEMS) の ICT 一空調連係制御技術による省電力への取り組み," *NTT 技術ジャーナル*, pp. 15-19, Nov. 2012.
- [5] "ECHONET Web Page." Available at <http://www.echonet.gr.jp/>.
- [6] J. Case, M. Fedor, M. Schoffstall, and C. Davin, "RFC 1157: Simple Network Management Protocol (SNMP)," 1989. Available at <http://www.hjp.at/doc/rfc/rfc1157.html>.
- [7] "IEEE1888 (FIAP) - 東大グリーン ICT プロジェクト." Available at <http://www.gutp.jp/fiap/>.
- [8] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, and D. Winer, "Simple object access protocol (soap) 1.1," 2000.
- [9] "WebSocket Web Page." Available at <http://www.websocket.org/>.
- [10] "大規模オープンテストベッド jos." <http://www.nict.go.jp/nrh/nwgn/jose.html>.
- [11] W. W. Shein, Y. Tan, and A. O. Lim, "PID controller for temperature control with multiple actuators in cyber-physical home system," *International Transaction on Systems Science and Applications*, vol. 8, pp. 149-166, Dec. 2012.