

ネットワーク接続機器の 機器名特定システムの開発

美原義行[†] 山本隆二[†] 佐久間聡[†]
山崎毅文[†] 佐藤 敦[†]

本書では、LAN内に接続された、家電等のIP機器（以下、機器）の機器名を特定するシステムについて記述する。現状、LAN内に接続されている他の機器に対しては、一部のプロトコル搭載機器を除けば、MAC・IPアドレスのペアのみ把握可能であり、機器の区分や製造メーカー名、ブランド名、型番等の機器名を把握することは不可能である。本開発では、各種プロトコルを用いて、機器からの返答を受信し、その機器の返答内容から、機器名を特定するシステムの開発を行った。本システムは、OSGiを用いて開発したことで、ソフトウェアモジュールの追加・削除を容易に行うことができ、新たに提案されるプロトコルへの柔軟な対応も可能となった。

A Device Information Detecting System for IP Devices Connected to a Local Area Network

YOSHIYUKI MIHARA[†] RYUJI YAMAMOTO[†]
SATOSHI SAKUMA[†] TAKEFUMI YAMAZAKI[†]
ATSUSHI SATO[†]

This document shows a system which detects device information of IP devices, for example network appliances, connected to local area network (LAN). Although we can detect some information of the devices which implement specific protocols, we can at least detect only network (MAC/IP) addresses inside LAN. We can not identify device information including device category information, manufacturer name information, model name information, model number information, and so on. We developed a system which receives reply packets and matches it to the database using multiple protocols which is widely used. Consequently, the system detects device information. This system allows both adding and deleting software modules easily using OSGi platform, and easily corresponding a new protocol in the future.

1. はじめに

昨今、ホームネットワーク（以下、ホームNW）に、パーソナルコンピュータ（以下、PC）だけでなく、NWに接続可能な情報家電も接続されてきている。図1は、情報家電等の、ホームNWに接続可能なノンPC機器の予想出荷台数と、それら機器のNW接続率の予測を示したものである。この図1のように、今後さらに多くの機器が、ホームNWに接続され、NWサービスの利用が拡大していくことが予想されている。このような状況下では、NWに接続可能な機器が普及するにつれ、ユーザが利用するNWサービスに発生する不具合も、増加していくことが予想される。

インターネット接続ができなくなった等、NWサービスに不具合が発生した場合、通常は、ホームゲートウェイ（以下、HGW）やインターネット上のサーバに対して、不具合が発生した情報家電等の機器から、ICMP Echo request [2]を送信することにより、HGWやサーバとの接続性を確認できる。これより、IPパケット到達区間を把握でき、原因を切り分けることが可能となる。HGWまでの接続性を確認できなかった場合は、ホームNW内で断線が発生していると判断できる。一方、HGWまでの接続性を確認できたが、インターネット上のサーバに対して接続性を確認できなかった場合は、アクセスNWやインターネットNW内で断線が発生していると判断できる。そして、HGWまでの接続性を確認でき、かつ、インターネット上のサーバとの接続性を確認できた場合は、不具合が発生した機器上のアプリケーションの設定誤りと判断できる。例えば、その機器におけるwebブラウザアプリケーションのプロキシの設定等の誤りが考えられる。しかし、この接続確認処理を行うためには、HGW/サーバのIPアド

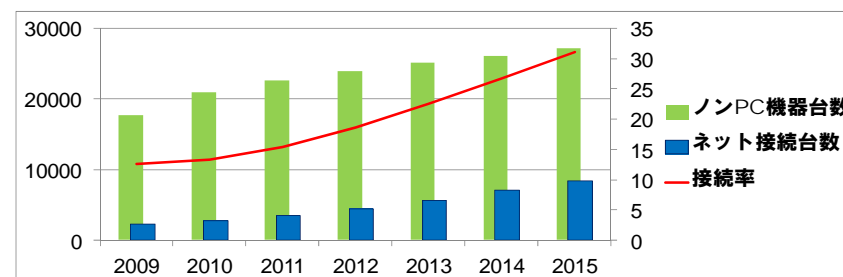


図1 ノンPC機器の予想出荷台数とネット接続率 [1]
Figure 1 The shipment and connecting rate of non-PC devices.

[†]日本電信電話株式会社 サイバソリューション研究所
NIPPON TELEGRAPH AND TELEPHONE CORPORATION
NTT Cyber Solutions Laboratories

レスを把握する必要がある、NW リテラシが低いユーザにとっては、非常に困難な操作となる。結果、NW サービスの不具合に対して、ユーザ自身でトラブルシュートを行うことは困難であると言える。

このような場合、多くのユーザは、NW 事業者のサポートセンタへ問い合わせを行う。問い合わせを受けた、NW 事業者がユーザ宅の HGW まで接続できた場合、HGW 経由で、不具合が発生した情報家電まで接続性を確認できれば、トラブルシュートを行うことが可能となる。このとき、HGW から不具合が発生している機器まで接続性を確認できなかった場合は、遠隔の NW 事業者のサーバから HGW までは、既に接続が確認できているため、ホーム NW 内で断線が発生していることが確認できる。一方、HGW から不具合が発生している機器まで、接続性を確認できた場合は、ホーム NW に問題が発生していないと判断できるため、機器のアプリケーションにおける設定誤りだと判断できる。この接続確認処理を行う場合にも、不具合が発生した機器の IP アドレスを把握する必要がある。

上記のように、HGW から不具合が発生した機器の IP アドレスを把握することや、不具合が発生した機器から HGW の IP アドレスを把握するためには、IP アドレスと機器の対応付けが必要になる。現状、ある端末から、その端末が保持している IP アドレスと同一サブネットマスクの、全ての IP アドレスに対して ARP [3] を実行することで、ホーム NW 内に接続されている全 IP アドレスと MAC アドレスのペアを把握可能である。しかし、現状では、発見できた IP/MAC アドレスを保持する機器までは特定できず、結果、接続性を確認したい機器の IP アドレスと MAC アドレスを把握できない。HGW の IP アドレスを把握できないため、ユーザとして HGW に対して接続性を確認

することができず、同様に、NW 事業者としても、ユーザが不具合を訴えている情報家電に対して、接続性を確認することができない。

以上より、NW サービスに発生した不具合に対応するためには、ホーム NW 内に接続されている全 IP/MAC アドレスを把握した後、ユーザ側では HGW の IP アドレスを特定する必要があり、NW 事業者側では、ユーザの不具合が発生した機器とその IP アドレスを対応付ける必要がある (図 2)。これより、NW に接続されている機器の IP/MAC アドレスだけでなく、その機器名を特定することが必要と言える。

その他、NW に接続されている機器の機器名の特定が可能になれば、以下のような、有効なアプリケーションサービスの実現も考えられる。

操作・設定サポートサービスへの応用

昨今、情報家電は多機能化が進んでおり、操作や設定が難しくなっている傾向になっている。そこで、情報家電の操作や設定ができないユーザに対して、操作・設定方法を教えるサービスが有効であると考えられる。例えば、あるユーザが情報家電の操作や設定で不明な点があった場合、そのユーザは、ある問い合わせセンタへ電話をかける。そして、オペレータは、その情報家電のメーカー名や、シリーズ名、型番等をユーザからヒアリングすることで把握し、その情報を元に、インターネット上でマニュアルを検索し、そのマニュアルを参照することで、ユーザに対して操作方法や設定方法を伝えることが可能となる。しかし、この場合、情報家電の扱いに詳しくないユーザは、操作や設定方法が分からない情報家電のメーカー名や型番を、正確にオペレータへ伝えることができない場合がある。「テレビ」「プリンタ」等の機器種別や、製造メーカー名を、オペレータに電話で伝えることができたとしても、シリーズ名や型番等を正確に調べ、伝えることは、ユーザにとって非常に困難な処理となる。そのため、オペレータが NW 経由で、ユーザ宅 LAN 内に接続されている機器の機器名を特定できれば、ユーザから、操作・設定が不明な情報家電の機器種別とメーカー名程度を聞きだすのみで、それ以外の情報である、シリーズ名や型番等まで把握することが可能となる。以上より、オペレータ側で、接続機器の機器名を特定することにより、マニュアルを検索し、参照可能になるため、ユーザに対しても操作や設定方法を教えることが可能となる。

使用電力の見える化サービスへの応用

操作・設定サポートの応用において記述したように、機器名を特定することで、その機器のマニュアルを引用することが可能になるため、マニュアルに記述されている、その機器の消費電力 (w) を把握することも可能となる。その機器の消費電力を把握できるようになるため、利用時間 (h) を特定できれば、マニュアルから特定した消費電力 (w) と利用時間 (h) を掛け合わせることで、消費電力量 (wh) を把握することが可能となる。

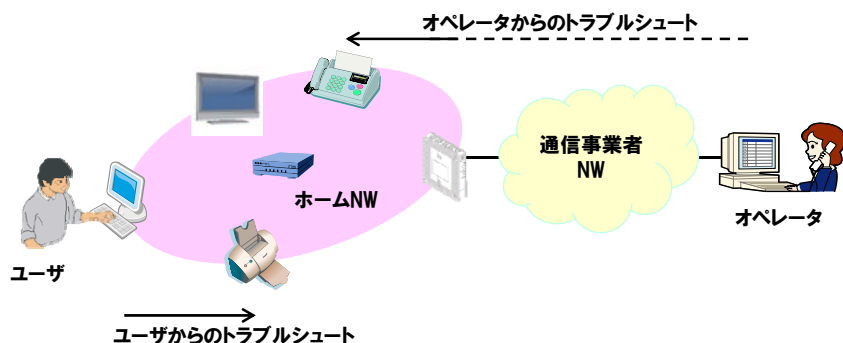


図 2 不具合対応の流れのイメージ。
Figure 2 An idea of troubleshooting.

一方、NWに接続されている機器の機器名を特定することで、IPアドレスと機器名の対応付けが可能になるため、目的のIPアドレスに対してPingを実行し、接続性を確認することが可能となる。接続性の確認ができれば、端末の電源が入っていると判断し、接続性の確認ができなければ、電源が入っていないと判断すると、接続性の確認を定期的に行うことで、電源がON状態になっている時間を推測することも可能となる。この処理によって推測した、機器の連続運転時間から、消費電力量を算出し、電気会社が定める、1kwhあたりの料金（東京電力 [a]では、1kwhあたり 14.82 円）を掛け合わせることで、機器が使用した電気代も推測することが可能となる。

以上より、NWに接続されている機器名を特定することで、メンテナンスサービスに効果があるだけでなく、様々なアプリケーションサービスへの応用も期待できる[4]。これより、本研究では、NW内に接続されている機器の機器名を特定する機能をモジュール化し、様々なアプリケーションサービスで利用できるようにすることを目標とする。

2. 機器名特定モジュール

本章では、機器名特定モジュールの要件をリストアップし、その要件に応じてシステム設計していく。

2.1 機器名の定義

本システムでは、同一 LAN 内に接続されている機器の機器名を特定することを目標としている。本システムは、この機器名特定の目標として、以下の四つの情報を特定することを目標とする。

- (a) 区分
- (b) メーカー名
- (c) 機種名
- (d) 型番

(a)の区分は、「PC」や「テレビ」等のように機器の種別を表す情報である。(b)のメーカー名は、その機器を製造したメーカーの名前情報である。そして、(c)機種名はその機器のブランド名やシリーズ名にあたる情報である。

機器名に必要な情報は多い方が、様々なアプリケーションサービスに対応できると考えられる。例えば、製造年月日やシリアル番号等も、アプリケーションによっては重要である場合も考えられる。しかし、機器名の特定は、上記2章で記述したように、マニュアルを引用することや、ユーザの発言と照合してユーザが指定する機器のIPアドレスを特定することを最初の目的としている。そのため、機器を特定するために必要最低限の情報として、上記4つを選択した。

2.2 機器名特定アルゴリズム

NWに接続される機器は、様々なプロトコルスタックを搭載している。この特徴を利用して、様々なプロトコル信号を、ホームNW内に接続されている機器に対して送信し、その信号に対する応答の特徴をデータベース（以下、DB）と照合することで、機器名を特定する（図3）。この処理の概要を以下に記述する。

0. プロトコル信号の応答であるプロトコル情報の特徴と、機器名の対応表である辞書DBを作成する。
1. あるタイミングで、機器名特定モジュールから、同一LAN内に接続されているIPアドレスとMACアドレスのペアを、ARPにより発見する。まず、機器名特定モジュールが搭載された機器のIPアドレスと、そのIPアドレスのサブネットマスクから、同一LAN内に接続されている機器のIPアドレスの範囲を特定する。例えば、機器名特定モジュールが搭載された機器のIPアドレスが、192.168.1.35/24 だとした場合、同一LAN内に接続されている機器のIPアドレスの範囲は、192.168.1.1~192.168.1.254 である。そして、この範囲のIPアドレス全てに対してARPを送信し、MACアドレスが応答されてきた場合、そのIPアドレスとMACアドレスのペアを保持しておく。
2. 上記1.にて発見した、IPアドレスとMACアドレスの全ペアに対して、様々なプロトコルに即した信号を送信する（表1）。
3. 上記2.で送信したプロトコル信号に対する、各機器からの応答を、辞書DBと照合し、機器名を特定する。

以上のような処理により、LAN内に接続されている機器を発見し、機器の機器名まで特定する。以下で、この処理を行うシステムの設計を行っていく。

2.3 機器名特定モジュールの機能要件

機器名特定モジュールを開発する上で、以下の要件がある。**要件 1. 新規プロトコルへの対応（ソフトウェアの開発・システムアップデート）を容易にすること**

機器名特定に利用するプロトコルとして、現在情報家電にも搭載されているプロトコルを利用していくことを想定するが、今後新たなプロトコルが提案されていくことは容

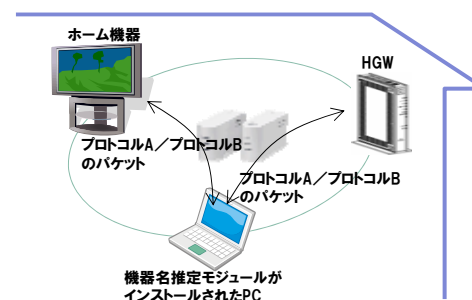


図3 機器名特定イメージ
Figure 3 An idea of process for detecting device information.

a) <http://www.tepco.co.jp/e-rates/individual/data/chargelist/chargelist01-j.html>

表1 機器名特定に利用したプロトコル

Table 1 Protocols which are used in our device information detecting system.

プロトコル名	特定できる機器の 大まかな種別	本プロトコルにおいて利用される文字列部分 の概要
HTTP	カメラ, PC, TV	HTTP GET を送信し, その信号に対する応答メッセージにおける HTTP ボディ部 (HTML) の文字列
NetBIOS [b]	PC	NetBIOS 名と提供サービスタイプ
UPnP Device Architecture [7]	ハードディスクレ コーダ, プリンタ, TV	DDD 内のデバイス基本情報内のエレメントに 格納されている文字列 例. <deviceType>エレメント
SNMP [10]	SOHO 向け複合機	MIB における sysname, sysdescription に記述さ れている文字列
機器固有 プロトコル	カメラ	機器独自のプロトコル

易に想像できる。そのため、新たなプロトコルが提案される毎に、全体システムの改造が発生し、かつモジュールをアップデートするために、OS を含めてファームウェアアップデートをしなければいけなくなると、再起動が必要になり、アプリケーションサービスの停止も発生する。サブシステムとなるべき、機器名特定モジュールのアップデートのために、メインシステムであるアプリケーションサービスを停止することは、避けなければならない。これより、新規プロトコルに対して、開発を容易にし、システムへのアップデートも容易にすることが必要となる。

要件 2. 新規特定アルゴリズムへの対応

我々のアルゴリズムは、辞書 DB との照合により機器名を特定するが、辞書に登録していない機器においても、収集したプロトコル情報から、ある特定アルゴリズムを利用して特定する手法も様々考えられる。新しいプロトコルだけでなく、新しい特定アルゴリズムにも対応し、機器名を特定できることが必要となる。

要件 3. 様々なプラットフォーム上で動作すること

このモジュールは、多くのアプリケーションサービスで利用されることを目的としているため、様々なプラットフォームで動作することが望まれる。Linux や Windows のプラットフォームで動作することが必要となる。

要件 4. 機器名の特定を、即時的に行えること

b) <http://support.microsoft.com/kb/163409/ja>

機器名を特定するためには、上記アルゴリズムのように、同一 LAN 内に対して ARP スキャンを行い、複数のプロトコルを用いて、プロトコル情報を収集し、その結果を辞書 DB と照合して結果を返す。このような処理を行ってからの結果を、アプリケーションから要求を受信し、即時的に機器名特定の結果を返すことは非常に難しい。しかし、利便性向上のため、他アプリケーションから機器名特定の要求を受信した際は、即時的に機器名特定の結果を返すことが必要となる。

要件 5. 一度発見した機器については、ある期間情報を保存し、管理すること

本モジュールの主要なアプリケーションとして、不具合が発生した機器に対して、接続確認を行う利用シーンがある。不具合が発生した機器が、IP 接続できていない場合は、その機器に対してプロトコル信号を送信したとしても、その機器からの応答を受信することは、もちろん不可能である。これより、プロトコル情報の収集と、機器名特定を定期的に行い、一度発見した機器については、ある期間情報を保存することが必要となる。これにより、ユーザから不具合の問い合わせを受け、その機器が IP 接続されていなかったとしても、その機器の IP/MAC アドレスに対して、接続確認を行うことが可能となる。

2.4 機器名特定モジュールのシステム設計

以下で、上記2.3節で示した機能要件に即して、機器名特定モジュールを設計していく。

要件 1. 新規プロトコルへの対応（ソフトウェアの開発・システムアップデート）を容易にすること

情報家電等の機器に新規プロトコルが搭載された場合、そのプロトコルの信号を送信するためのプログラムの開発と、そのプログラムのシステムへの追加を容易にするためには、プログラムの部品化を行う必要がある。そこで、本システムでは、Java VM 上で動作する OSGi [5][6]を採用した。OSGi は、プログラムを部品化する技術であり、機能ごとにプログラムを分離可能な、ソフトウェア実行基盤である。OSGi では、一つのプログラムは、「バンドル」という称号で表記される。

本システムでは、OSGi を用いて、システム全体設計を図のように行い、プロトコル信号を送信・受信し、辞書 DB と受信したプロトコル情報とを照合する。そのため、受信したプロトコル情報を整形する必要がある。これより、プロトコル信号を送信し・返答を受信するバンドルと、受信したプロトコル情報を整形するバンドルを分離した。また、この整形したプロトコル情報を、辞書 DB と照合するバンドルも分離して作成した。このように、機能ごとにバンドルを作成することで、機能追加・削除を容易に行うことが可能となる。以下では、プロトコル信号を送信・返答を受信する機能をもつバンドルを、プロトコルドライババンドルと表記し、受信した情報を、辞書 DB と照合するために整形する機能をもつバンドルを、プロトコル制御バンドルと表記する

ことにする。

本システムで、機器名の特定に用いるプロトコルとしては、多様でかつ多数の情報家電を購入し、搭載しているプロトコルの調査を行った結果を反映させた。搭載されているプロトコルは、表に示されているものであり、

また、この OSGi では、バンドルを追加・削除を行う際には、OS や Java VM の再起動を行う必要がないため、サービスの動作に影響を与えない。

要件 2. 新規特定アルゴリズムへの対応

本モジュールでは、機器名を特定するため、プロトコルドライババンドルと、プロトコル制御バンドルにて、特定のプロトコルに関する情報を収集する。この収集したプロトコル情報を、我々で作成した辞書 DB に照合することで機器名の特定を行うが、我々で作成した辞書 DB 以外の DB との照合や、その他特定アルゴリズムを組み入れることができるよう、プロトコル情報収集部と、機器名特定部分を分離することで、収集したプロトコル情報を、他社アルゴリズムのインプットとすることを可能にした。他アルゴリズムにおいて、異なるプロトコル情報を収集する場合は、他社プロトコルバンドルを配置する。

要件 3. 様々なプラットフォーム上で動作すること

OSGi は、Java VM 上で動作するため、様々なプラットフォームで動作することが可能となる。しかし、一方で OSGi は、Java で提供されている API の範囲でのプログラムのみを記述することが可能である。これより、Java では実現できない低レベルのプロトコルの制御に関しては、OS 毎に作成しなければならない。例えば、ARP のレイヤは、OSI レイヤで表現すると、レイヤ 2 にあたり、Java のみでは実現することが不可能である。そのため、Window 版のドライバと Linux 版のドライバを作成し、JNI 機能を利用してバンドルとの通信を可能にする必要がある。

要件 5. 機器名の特定を、即時的に行えること

機器名特定処理では、同一セグメント内への ARP スキャン、各種プロトコル情報の収集、辞書 DB との照合のように、複数ステップにわたる。このため、他アプリケーションシステムから機器名特定の要望を受け取り、それから上記処理を行い、即時的に機器名を応答することは、現状不可能である。これより、以下の二つの方法を提供するよう設計した。

- ・ 即時的に、プロトコル情報を収集して、機器名を特定する処理では、アプリケーションから、応答を返してほしい時間の指定を受け、その時間内で収集できているプロトコル情報からのみ、特定を行う。
- ・ 定期的に、プロトコル情報を収集して・機器名を特定する処理を行い、アプリケーションから、機器名特定の要望を受けた際、保存しておいた最新の機器名情報を応答することで、要求に対して即座に機器名を応答する。

定期的に LAN 内接続機器からプロトコル情報を収集して、機器名を特定しておき、

表 2 機器名特定モジュールの出力情報

Table 2 The output information of the device information detecting system.

情報	説明
基本情報	IP/MAC アドレス, 最終特定日時, 現在の接続性
機器名 (特定結果)	区分, メーカー名, 機種名, 型番 ただし, 特定できなかった場合は, 空欄となる。
プロトコル情報	各プロトコルドライババンドルで取得した文字列

最新の情報を返すことは、情報の鮮度が課題となる。一方で、情報の鮮度を高くするために、機器名特定処理の間隔を短くすると、NW や機器名特定モジュールだけでなく、機器名特定モジュールからプロトコル情報を収集される機器側においても、負荷が高くなってしまう。これより、機器名特定処理の間隔については、アプリケーション側で設定できるように設計した。

要件 5. 一度発見した機器については、ある期間情報を保存し、管理すること

定期的なプロトコル情報収集と、機器名特定により、不具合発生時で IP 接続できなかったとしても、その機器の IP/MAC アドレスを把握することが可能となる。また、この定期的なプロトコル情報の収集と、機器名特定に加え、接続状態も確認するよう設計した。これにより、即座に機器の接続状態を確認することが可能となる。ある期間連続で接続性を確認できなかった場合は、アプリケーションサービスが管理情報から削除できるようにも設計した。

以上より、機器名を特定する本モジュールでは、OSGi プラットフォームを用いた、図 4 のような構成とした。辞書 DB は、サーバ側に設置し、クライアント側にあたる機器名特定モジュールが適宜、このサーバ側の辞書 DB を利用するよう設計した。新たな情報家電や、その他情報機器が市場に出てくる度に、その機器の情報が新たに加えられ、更新が頻繁になるためである。また、辞書 DB をサーバ側に設置する、つまりは、クライアント側に辞書 DB を設置しないことで、クライアント側のメモリ使用量を抑えることが可能となる。以下、詳細な流れを記述する。

まず他システムからのインタフェース (以下、IF) として、機器名特定ライブラリを用意した。機器名特定ライブラリのインタフェースは、IP アドレスを入力として、その IP アドレスの機器名を返す。IP アドレスを入力しない場合は、NW に接続されている全機器の機器名を返す。その他の入力としては、プロトコル情報取得の有無と、即時的に収集するか/蓄積情報を返すかのフラグ、(即時収集の場合)待ち時間がある。待ち時間は、指定した時間内に収集できたプロトコル情報のみから特定を行うために設けた引数である。そして、出力として、入力で指定された IP アドレスをもつ機器の機器名と、(指定された場合)プロトコル情報を応答する (表 2)

アプリケーションサービスから、機器名の特定依頼を受けると、ライブラリ経由で、機器名提供バンドルに、引数の情報を送信する。そして、機器名提供バンドルは、要求が蓄積情報の取得の場合は、この機器名提供バンドル内で保存していた情報から、指定された IP アドレスの機器の機器名を応答する。一方、即時収集の要求の場合、機器名提供バンドルは、機器名を取得したい IP アドレス、プロトコル情報の有無を、プロトコル情報マージバンドルに送信する。プロトコル情報マージバンドルは、指定された IP アドレスに対して、各種プロトコル制御バンドルに指示を送信して、プロトコル制御バンドルは、プロトコルドライババンドルに対して指示を送信する。ドライババンドルは、収集したプロトコル情報を制御バンドルへ送信し、制御バンドルで整形した情報をプロトコル情報マージバンドルへ応答する。プロトコル情報マージバンドルは、そのプロトコル情報一覧を保存し、かつ、機器名提供バンドルへプロトコル情報一覧を応答する。機器名提供バンドルは、プロトコル情報マージバンドルから受信したプロトコル情報一覧を、辞書 DB に送信し、機器名を取得し、アプリケーションサービスに対して応答を返す。

同様に、定期的な収集においては、設定バンドルから設定された時間間隔毎に、機器名提供バンドルからプロトコル情報マージバンドルから、全ての端末の機器名と、接続状態と取得した時刻を取得する。処理効率化のため、端末の機器名に更新がない

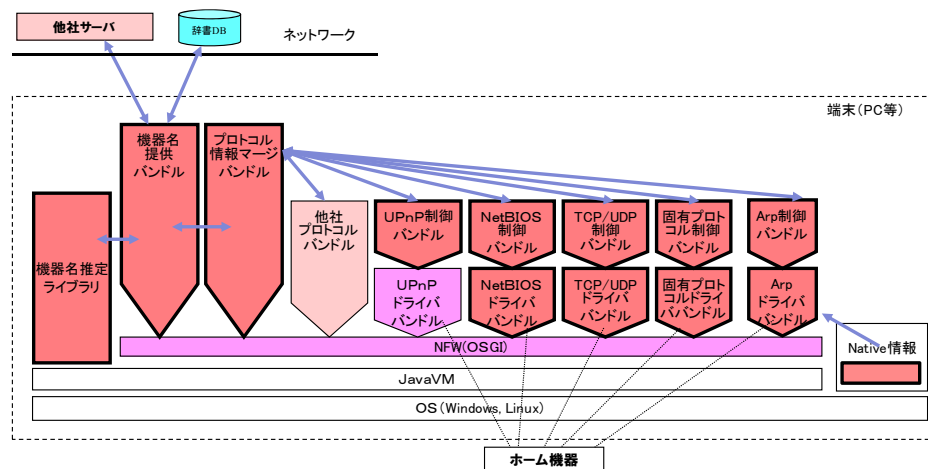


図 4 バンドル構成

Figure 4 The Bundle Construction of the system

場合は、接続状態と取得時刻のみ更新する。これは、Native 情報における ARP テーブルを確認することで、IP アドレスの更新を把握する。以上の設計を元に、機器名を特定するモジュールのシステム開発を行った。

3. 評価

本モジュールには、機器名を即時的に収集して結果を 15 分以内に返すことが求められた。通常、ユーザからの不具合の問合せを受けた際、一度その要件を受け、不具合の対応方法を、ユーザに折り返し連絡する。これは、最初のユーザからの問合せから、その折り返し連絡までの時間として、15 分程度で連絡をすることが、ユーザ満足度低下の観点から、求められるからである [11][12]。これより、本モジュールにおける、プロトコル情報収集時間、辞書 DB との照合時間を含めた、機器名特定時間が 15 分以内であることを、実験により評価した。また、本モジュール動作時の端末負荷 (CPU 使用率とメモリ使用量) と、NW 負荷も同時に求めることで、組み込み機器への搭載可能性と、ユーザ NW への適用可能性も併せて検討する。

他の非機能要件として、安定的動作が求められた。このため、長期にわたり、本モジュールを動作させ、システムの停止率 (=システム停止時間/システム稼働時間) を実験により評価した。このシステム停止率を求めることにより、本モジュールの動作の安定性も評価できる。

本モジュールでは、利用される環境として、以下を想定した。これは、ホーム NW や SOHO で利用されることを想定し、事前にホーム NW や SOHO の利用環境を調査した結果である。

- ・ LAN 内の接続機器は、最大 100 台
 - ・ LAN は、ネットマスク 24 ビットのサブネット
- また、機器名推定モジュールをインストールした機器の性能は以下である。
- ・ OS : Windows XP
 - ・ CPU : 1.00GHz
 - ・ Java VM : CDC [c] PF1.1
 - ・ 物理メモリ : 1GB
 - ・ JDK : jdk1.5.0_22

非機能要件 1. 機器名特定時間が、要求受け付けから 15 分以内に返すこと

100 台の機器として、仮想的に用意し、表 1 の全てのプロトコルに応答するようにした。これら仮想機器に対して、プロトコル情報の収集と機器名特定を行う時間は、10 分 01 秒 333 であり、要求を満たす結果となった。

このとき、OSGi フレームワークと JRE [d] 含めた機器名特定モジュール全体のヒープメモリと非ヒープメモリの使用量は、それぞれ最大 7.9MB、最大 12.6MB であった。

c) Cononnected Device Configuration
 d) Java Runtime Environment

一方、OSGiフレームワークとJREのみを動作させた時のヒープメモリと、非ヒープメモリの使用量は、それぞれ最大 1.9MB、最大 9.0MBであった。これより、上記測定結果から推測される、OSGiフレームワークとJREでの使用量を除いた、機器名特定モジュール)のメモリ使用量は、ヒープメモリが最大 6.0MBと推定でき、非ヒープメモリが、最大 3.6MBと推定できる。また、CPU使用率は平均 12.97%程度であった。

国内の組み込み機器メーカーへヒアリングしたところ、ヒープメモリの使用量が 6.0MB 程度であれば、一般的な組み込み機器で動作可能であるという、ヒアリング結果を得た。これより、本機器名特定モジュールは、スマートフォン含む多くの組み込み機器において、実装可能であることが分かった。

一方、本システムが与えるNWへの負荷においては、このシステムが、ある一定期間内に送受信するパケットのデータ量を推測することで、検討した(表 3)。検討結果は、表 3 である。表 3 では、プロトコル毎に送受信されるパケットサイズを計算し、それが 1 秒間の間に送受信されることを想定した。検討結果は、ARP情報送信時に一番大きな負荷が発生することが分かった。もし、この送信期間の 1 秒間に、想定台数である 100 台の機器から応答があったとしても、304kbps程度(=256kbps+48kbps)である。LAN内の通信では、無線区間でも最低 1Mbpsの帯域は確保できると考えられるため、本システムが与えるNWへの負荷は、非常に低いことが分かった。

メモリの使用量は、時間に依らず一定だったため、一つだけでなく、二つの要望に対応できるようにした場合、メモリを二倍使用することになる。これより、定期的な収集と即時的な収集のタイミングが重なった時は、即時的な収集要望は破棄し、定期的な収集を実行するよう設定した。以下の、非機能要件の評価は、このような設定にした後で行った。

非機能要件 2. 安定的に動作すること

機器名推定モジュールに対し、以下に示す一定の負荷を掛けた状態で連続運転(48時間)を行い、システム停止や致命的なエラーが発生しないことを検査した。

- ・ 機器名提供バンドルからの即時的な収集要求を 30 分間隔で送信(48 時間中 44 回の要求。これは、定期的な収集のタイミングと重なると、定期的な収集を優先させるためである)
- ・ 機器名提供バンドルからの定期的な収集要求を 30 分間隔で送信(48 時間中 43 回の要求)
- ・ プロトコル情報マージバンドルからの常時収集要求を 30 分間隔で送信(48 時間中 96 回)

連続運転にて実施した負荷内容は、上記であり、48 時間の連続運転中、システム停止時間は 0 であった。これより、停止率の目標を達成することができた。

表 3 送信プロトコルとそのデータ量

Table 3 Protocols which this module uses and the Volume of the packets

収集情報の種類	送信	受信(接続機器は 100 台とする)
Arp 情報	ARP リクエスト(42byte)の送信 42(byte)×253(IP)×3(回)× 8(bit)=256Kbps	Arp リプライ(60byte)の受信 60(byte)×100(台)×8(bit)=48Kbps
UPnP 情報	M-SEARCH(143byte)の送信 143(byte)×2(回)×8(bit)=2.3Kbps Device Description(400byte)の取得 (HTTP GET 要求) 400(byte)×8(bit)=3.2Kbps	M-SEARCH のレスポンス 500(byte)×8(bit)=4Kbps Device Description の取得(HTTP GET 応答) 1500(byte)×8(bit)=12Kbps
NetBIOS 情報	Name Query(92byte)の送出 92(byte)×100(台)×8(bit)=73.6Kbps	Name Query のレスポンス 250(byte)×50(スレッド)× 8(bit)=100Kbps
機器固有 プロトコル 情報	機器固有プロトコルのパケット (100byte)送信 100(byte)×2(スレッド)× 8(bit)=1.6Kbps	機器固有プロトコルの返信パケット (512byte)受信 512(byte)×2(台)×8(bit)=8Kbps
その他プロ トコル	TCP/UDP 送信 100(byte)×2(スレッド)× 8(bit)=1.6Kbps	TCP/UDP 受信 5000(byte)×2(スレッド)× 8(bit)=80Kbps

4. 将来課題

以下に、本モジュールでは実現できなかった機能を挙げ、今後の方向性を記述する。
他セグメントに接続された機器の機器名特定

本書で記述した、機器名を特定するモジュールでは、同一 LAN 内に接続された機器の機器名のみが特定可能である。本アルゴリズムでは、情報家電等のホーム NW を想定しているため、IPv4 を想定し、IPv4 のブロードキャストパケットが届く範囲内からのみ、情報を収集するためである。しかし、ホーム NW 内に、HGW 以外にルータが接続される場合があり、そのルータ配下に異なる IP セグメントの NW が存在する環境も存在する(図)。これは、ユーザが無線 NW を利用するために、無線 NW のアクセスポイントとなるルータを設置してしまうためであり、このような接続事例は増えてきているため、無視できない接続形態である。

NW 事業者にとっては、HGW が設置されているセグメントとは異なるセグメントに設置された機器における不具合に関しては、IP アドレスと機器名を対応付けられないため、接続性を確認できず、対応することが不可能である。ホーム NW 内全体で、接続された機器の機器名を特定することは、保守範囲の拡大にもつながるため、今後は異なるセグメントに接続されている機器に関しても、機器名の特定を可能にする方式を検討していく予定である。

NW 機器の特定～HTIP [13][14]の利用～

本書で記述したシステムは、同一 LAN 内に接続された IP 機器の機器名を特定することを目的とした。しかし、スイッチングハブや、無線 AP のように、IP を終端せず、転送する NW 機器は、IP より上位のプロトコル信号に 응답しないため、特定どころか発見することすらできない。NW 機器を特定することが可能になれば、NW マップを特定することが可能になり、不具合が発生した際の切り分けに有効だと考えられる。2012 年 12 月に、IP 機器だけでなく、NW 機器を発見し、その機器の機器名を特定できるプロトコル「HTIP (Home network Topology Identifying Protocol)」が ITU-T にて、G.9973 として勧告化された。これは、民生向けの NW 機器が LLDP [15]というプロトコルを用いて、機器情報と、それに加えて MAC アドレステーブル情報を送信する方式を規定したプロトコルである。この HTIP に対応することで、NW 機器の機器名も特定可能となる。まだ、この HTIP を実装した NW 機器は存在しないが、今後様々な機器に普及していくことが予想されるため、HTIP バンドルを開発し、本モジュールに組み入れていく予定である。

5. まとめ

本書では、ホーム NW 内に接続された機器の機器名を特定する方式を紹介し、その機能を他のアプリケーションシステムからも利用可能なシステムの設計を述べた。本システムでは、Java ベースの OSGi を用いることにより、新規プロトコルへの対応や、アップデートを容易にするだけでなく、様々なプラットフォーム上で動作することが可能となった。本モジュールは、PC 管理サポート商用サービスにおいて、既に運用されている。今後は、切り分けサービスにおいても活用する予定である。今後は、異なるセグメントに接続された IP 機器の機器名の特定機能や、NW 機器の機器名の特定機能を追加していきたい。

参考文献

- 1) 株式会社富士キメラ総研: “2011 次世代ホームネットワーク/エネルギーマネジメント市場の展望”, May 2011.
- 2) J. Postel: RFC 792, “INTERNET CONTROL MESSAGE PROTOCOL”, Internet Engineering Task Force (IETF), September 1981.

- 3) David C. Plummer, RFC 826, “An Ethernet Address Resolution Protocol”, IETF, November 1982.
- 4) 大村弘之: 「やさしいホーム ICT」, 電気通信協会, 2011. ISBN-10: 4885490545, ISBN-13: 978-4885490545.
- 5) 川村, 前大道, 山崎, 森.: “OSGi (Open Service Gateway Initiative) の標準化動向について”, NTT 技術ジャーナル, Vol 19, No. 11, pp.94-98, 2007.
- 6) 山崎毅文, 山崎育夫, 近藤, 美原, “ホーム ICT 関連技術の標準化動向”, NTT 技術ジャーナル, Vol 22, No. 9, pp.49-53, 2010.
- 7) ISO/IEC 29341-1: “Information technology – UPnP Device Architecture – Part 1: UPnP Device Architecture Version 1.0”, Edition 1.0, December 2008.
- 8) J. Postel, and J. Reynolds.: RFC 959, “File Transfer Protocol”, IETF, October 1985.
- 9) J. Postel, and J. Reynolds.: RFC 854, “Telnet Protocol Specificatoin”, IETF, May 1983.
- 10) J. Case, M. Fedor, M. Schoffstall, and J. Davin.: RFC 1157, “Simple Network Management Protocol”, IETF, September 1990.
- 11) 年金事務所・年金相談センター: “お客様満足度アンケート (6 月 21 日公表) の追加的分析”, June 2010.
- 12) 厚生労働省: “受療行動調査の概要 (確定) ”, 2005.
- 13) ITU-T Rec, “G.9973, Protocol for identifying home network topology”, October 2011.
<http://www.itu.int/rec/T-REC-G.9973-201110-P>
- 14) Mihara, Y., Yamazaki, T., and Akama, T.: “Designing HTIP: Home Network Topology Identifying Protocol”, IEEE International Conference on Communications (ICC), pp.1-6, 2011. DOI: 10.1109/icc.2011.5962662.
- 15) IEEE Computer Society: “802.1AB-2009: Local and Metropolitan Area Networks - Station and Media Access Control Connectivity Discovery”, September 2009.