

DLNA／UPnP－ZigBee ゲートウェイ仕様書

Ver 0.81

March 31, 2008

変更履歴

版数	変更日	主な変更箇所	Copyright
0.8	2007/9/6	公開版初版	財団法人情報処理相互運用技術協会 株式会社アルファシステムズ
0.81	2008/3/31	・ZigBee から UPnP の操作仕様 ・一部シーケンスの追加・修正	財団法人情報処理相互運用技術協会 株式会社アルファシステムズ

DLNA/UPnP-ZigBee ゲートウェイ仕様書

目的：

ホームネットワークの異なるプロトコルをシームレスにつなぎ、ユーザーに対し統一されたインターフェイスとユーザービリティを提供していくための一手段として、情報家電の標準規格として普及している DLNA や UPnP とセンサーネットワークの標準規格として注目を集めている ZigBee センサーネットワークをシームレスに接続するゲートウェイデバイスに関する仕様を定義し、実現することを目的としている。

これにより、ユーザーは所持している DLNA や UPnP 端末（例えば携帯やモバイルビューワー、デジタルテレビやパソコン等）から、動画像、音声といった視覚的なコンテンツとして、センサー情報を取得・閲覧可能となる。また、UPnP の機能を利用して UPnP 端末からセンサーデバイスを操作することができるようになる。

活用してもらいたい対象者：

DLNA と ZigBee の相互接続を検討している技術者、事業者、チップベンダー等

利用シーン：

ホーム・オートメーションにおけるサービスとして、照明機器等の操作、室温センサーを含む温度調整器具、ホームセキュリティ等があるが、ゲートウェイでは、これらの ZigBee サービスを UPnP 端末から操作できるサービス機能と、状況を DLNA コンテンツとして把握できるサービス機能を提供することができる。

使用許諾条件

2007年8月

〒113-6591 東京都文京区本駒込二丁目28番8号
財団法人情報処理相互運用技術協会

〒150-0002 東京都渋谷区渋谷二丁目17番5号 シオノギ渋谷ビル12階
株式会社アルファシステムズ

財団法人情報処理相互運用技術協会（以下「当協会」という）と株式会社アルファシステムズ（以下「当社」という）は、以下の条件のもとで本ドキュメント（本使用許諾条件に添付されて提供されるドキュメントをいい、以下同じ）を使用、複製および頒布することを無償で許諾します。本ドキュメントを使用、複製または頒布した場合には、以下の条件に同意したものとします。

1. 本ドキュメントの中に含まれる著作権表示および本使用許諾条件を、本ドキュメントの全部または一部を複製したものに表示してください。
2. 本ドキュメントを使用したサービスの提供を含め営利目的に本ドキュメントを使用することができますが、本ドキュメントのみを単独で販売することはできません。
3. 第4項に定める場合を除き、本ドキュメントを使用したサービスの提供に際して、事前の書面による当協会及び当社の許可なく、それらの宣伝、広告活動に当協会及び当社の名称を使用することはできません。
4. 本ドキュメントを使用して得られた結果を、形態を問わず、出版、発表において公表する場合には、本ドキュメントと当協会及び当社の名称を引用等において明示してください。
5. 本ドキュメントは現状有姿で提供されるものであり、当協会及び当社は、本ドキュメントに関して、商品性および特定目的への適合性、エラー・バグ等の不具合のないこと、第三者の特許権、実用新案権、意匠権、商標権、著作権その他の知的財産権を侵害するものではないことを含め、明示たとと黙示たとを問わず、一切の保証を行わないものとします。また、当協会及び当社は、本ドキュメントの誤りの修正その他いかなる保守についても義務を負うものではありません。
6. 当協会及び当社は、本ドキュメントの使用または使用不能、複製、頒布、その他本ドキュメントまたは本使用許諾条件の規定に関連して生じたいかなる損害（特別損害、間接損害、逸失利益を含みますが、これに限りません）または第三者からのいかなる請求についても、法律上の根拠を問わず一切責任を負いません。当協会及び当社がかかる損害または請求の可能性について知らされていた場合も同様とします。
7. 本ドキュメントは、一般事務用、パーソナル用、家庭用、通常の産業用等の一般的用途を想定して作成されているものであり、原子力施設における核反応制御、航空機自動飛行制御、航空交通管制、大量輸送システムにおける運行制御、生命維持のための医療用機器、兵器システムにおけるミサイル発射制御など、極めて高度な安全性が要求され、仮に当該安全性が確保されない場合、直接生命・身体に対する重大な危険性を伴う用途（以下「ハイセイフティ用途」という）を想定して作成されたものではなく、当該ハイセイフティ用途に要する安全性を確保する措置を施すことなく、本ドキュメントを使用しないものとします。また、ハイセイフティ用途に本ドキュメントを使用したことにより発生する、いかなる請求または損害賠償に対しても当協会及び当社は一切の責任を負わないものとします。

-
-
8. 本ドキュメントへ ZigBee1.0 の仕様に基づいて作成されているので、ZigBee1.0 に関わる内容に関しては、以下の ZigBee Alliance の規定に準じます。

This document/specification contains intellectual property of the ZigBee Alliance. In order to represent a product as ZigBee, ZigBee Compliant, ZigBee Certified or similar or to use any ZigBee Alliance intellectual property (including, without limitation, the ZigBee Specification, ZigBee Profile Specifications, ZigBee Test Plans, ZigBee trademarks or logos) for any commercial purpose, users must be ZigBee Alliance members. Please see www.ZigBee.org for information on becoming a ZigBee Alliance member.

目次

1. はじめに	11
1.1. 背景.....	11
1.2. 目的.....	11
1.3. 概要.....	11
1.4. 本仕様書の位置付け.....	11
1.5. スコープ.....	12
2. プロトコル概要	13
2.1. UPnP 1.0.....	13
2.2. DLNA 1.0.....	15
2.2.1. DLNA スタック構成.....	15
2.2.2. DLNA のユースケース.....	15
2.3. ZigBee 1.0.....	17
2.3.1. ZigBee スタック構成.....	17
3. DLNA/UPnP - ZigBee ゲートウェイ仕様	20
3.1. ユースケースモデル.....	20
3.1.1. DLNA・ZigBee ゲートウェイユースケースモデル.....	20
3.1.2. UPnP・ZigBee ゲートウェイユースケースモデル.....	21
3.2. 全体機能説明.....	24
3.3. UPnP-ZigBee 接続仕様.....	26
3.3.1. UPnP CP から ZigBee デバイスの操作に関わる基本シーケンス.....	26
3.3.2. ZigBee デバイス (コントローラ側) から UPnP デバイスの操作に関わる基本シーケンス (オプション).....	33
3.3.3. 機能説明.....	36
3.3.3.1. データフォーマットの変換.....	36
3.3.3.2. ディスクリプション変換.....	36
3.3.3.3. プロトコル変換.....	36
3.3.4. データフォーマット変換.....	38
3.3.4.1. 各プロトコルのペイロードにおけるデータフォーマット変換.....	38
3.3.4.2. ディスクリプションにおけるデータフォーマット変換.....	41
3.3.5. ディスクリプション変換.....	43
3.3.5.1. UPnP におけるデバイス及びサービス情報.....	43
3.3.5.2. ZigBee におけるデバイス及びサービス情報.....	46
3.3.5.3. 標準 UPnP デバイスと ZigBee 標準プロファイルの対応.....	48

3.3.5.4.	UPnP-ZigBee の識別情報の基本対応.....	48
3.3.5.5.	ZigBee のデバイス及びサービス情報を元にした仮想 UPnP デバイスの生成	49
3.3.5.6.	UPnP のデバイス及びサービス情報を元にした仮想 ZigBee デバイスの生成	62
3.3.6.	プロトコル変換.....	66
3.3.6.1.	アドレッシング	66
3.3.6.2.	Discovery プロトコル変換.....	67
3.3.6.3.	制御プロトコル変換.....	79
3.3.6.4.	イベント変換.....	83
3.3.7.	プロファイルに応じたコンフィギュレーション.....	89
3.3.7.1.	SSDP の応答.....	91
3.3.7.2.	GENA モデュレーション	91
3.3.7.3.	ゲートウェイアプリケーションからの参照.....	91
3.4.	DLNA-ZigBee 接続仕様.....	92
3.4.1.	機能説明	92
3.4.1.1.	Digital Media Server 機能の提供	92
3.4.1.2.	DLNA コンテンツの動的生成と公開	92
3.4.1.3.	ZigBee センサー情報の収集	93
3.4.2.	メディアフォーマット変換.....	94
3.4.2.1.	一つのセンサー情報のメディアフォーマット変換	94
3.4.2.2.	複数のセンサー情報のメディアフォーマット変換	94
3.4.3.	コンテンツ生成エンジン基本仕様	94
3.4.3.1.	コンテンツの生成	94
3.4.3.2.	サービス分類情報の提供	94
3.4.3.3.	生成コンテンツのメディア情報の提供.....	95
3.4.3.4.	対象センサー種別・属性情報の提供.....	95
3.4.4.	基本シーケンス.....	95
3.4.4.1.	DLNA-ZigBee ゲートウェイの初期化	95
3.4.4.2.	コンテンツの生成	98
3.4.4.3.	コンテンツの削除	100
3.4.5.	DLNA コンテンツ生成に関するコンフィギュレーション.....	100
3.4.5.1.	連続したコンテンツ閲覧要求時の生成頻度コントロール	102
3.5.	システムユースケース	103
3.5.1.	概要.....	103
3.5.1.1.	ゲートウェイのシステムユースケースとプロファイルの関係	103
3.5.1.2.	共通のユーザービリティ特性	103

3.5.2.	ホーム・オートメーションへの適用	104
3.5.2.1.	デバイスカテゴリ	104
3.5.2.2.	ゲートウェイが提供するサービス	104
3.5.2.3.	関連する ZigBee Stack Profile	104
3.5.2.4.	推奨されるゲートウェイのコンフィギュレーション設定	105
3.5.3.	ビルディング・オートメーションへの適用	106
3.5.3.1.	デバイスカテゴリ	106
3.5.3.2.	ゲートウェイが提供するサービス	106
3.5.3.3.	関連する ZigBee Stack Profile	106
3.5.3.4.	推奨されるゲートウェイのコンフィギュレーション設定	107
3.5.4.	医療・健康系サービスへの適用例	108
3.5.4.1.	デバイスカテゴリ	108
3.5.4.2.	ゲートウェイが提供するサービス	108
3.5.4.3.	関連する ZigBee Stack Profile	108
3.5.4.4.	推奨されるゲートウェイのコンフィギュレーション設定	108
3.5.5.	アミューズメントへの適用例	109
3.5.5.1.	デバイスカテゴリ	109
3.5.5.2.	ゲートウェイが提供するサービス	109
3.5.5.3.	関連する ZigBee Stack Profile	109
3.5.5.4.	推奨されるゲートウェイのコンフィギュレーション設定	110

用語集

DLNA

DLNA : Digital Living Network Alliance

DMP : Digital Media Player

DMS : Digital Media Server

MT : Media Transport

UPnP

CDS : Content Directory Service

CP : Control Point

GENA : Generic Event Notification Architecture

HTTP : Hypertext Transfer Protocol

HTTPMU : HTTP Multicast over UDP

HTTTPU : HTTP unicast over UDP

ICANN : Internet Corporation for Assigned Names and Numbers

SOAP : Simple Object Access Protocol

SSDP : Simple Service Discovery Protocol

UDN : Unique Device Name

UPC : Universal Product Code

UPnP : Universal Plug and Play

URI : Uniform Resource Identifier

URL : Uniform Resource Locator

URN : Uniform Resource Name

UUID : Universally Unique Identifier

XML : Extensible Markup Language

ZigBee

AF : Application Framework

AIB : APS Information Base

APDU : APS Protocol Data Unit

APL : Application Layer

APS : Application Support sub-layer

APSDE : APS Data Entity

APSDE-SAP : APS Data Entity – Service Access Point

APSME : APS Management Entity

APSME-SAP : APS Management Entity – Service Access Point

CSMA/CA : Carrier Sense Multiple Access / Collision Avoidance

FFD : Full Function Device

KVP : Key Value Pair

MPCS-SAP : MAC Common Part Sub-layer – Service Access Point

MLME-SAP : MAC Layer Management Entity – Service Access Point

MSG : Message service type

NIB : Network layer Information Base

NLDE : Network Layer Data Entity

NLDE-SAP : Network Layer Data Entity – Service Access Point

NLME : Network Layer Management Entity

NLME-SAP : Network Layer Management Entity – Service Access Point

NPDU : Network layer Protocol Data Unit

NSDU : Network Service Data Unit

PAN : Personal Area Network

RFD : Reduce Function Device

ZDO : ZigBee Device Object

その他

ECHONET : Energy Conservation and Homecare Network

UOPF : Ubiquitous Open Platform Forum

1. はじめに

1.1. 背景

現在 DLNA, UPnP, ECHONET, UOPF, ZigBee などのフォーラムに代表されるように、様々な通信規格の標準化がホームネットワーク環境を中心に盛んに進められている。これらのフォーラムで進められている規格に共通する特徴として、物理層からアプリケーションプロファイルや基本サービスといった上位アプリケーション層までを定め、それによって厳密に、異なるベンダー製品間の相互接続性と運用性を確保しようとしている点がある。各々の規格はそれぞれが対象とするサービス分野とネットワーク環境に最適化されており、消費者の生活空間の至る所にネットワークが存在し情報サービスを受けることができる、ユビキタスネットワーク・サービス環境が実現しつつある。

このような状況の中、今後の議論の中心は、それら異なるネットワークサービスをどうシームレスに接続して、ユーザーに対し統一されたインターフェイスとユーザービリティを提供していくかということに移りつつある。その具体的な例の一つとして、ZigBee に代表されるセンサーネットワークの観点から、DLNA/UPnP 規格のテレビにセンサー情報を表示して、閲覧したいという要求が挙げられる。ユーザーの視点から考えると、ユーザーが普段所持し、利用している端末、例えば携帯などのモバイル端末、デジタルテレビやパソコン、カーナビ等から、動画像、音声といった視覚的なコンテンツとしてセンサー情報を取得・閲覧可能であることが望ましい。つまり、マルチメディアコンテンツの閲覧と共有を規格する DLNA との接続性をゲートウェイ機能により確保することで、ユーザーは普段家庭にてコンテンツ閲覧に利用している、DLNA 端末をそのまま利用して、センサー情報も閲覧することができるようになることが求められている。同様に UPnP との相互接続性においても、ゲートウェイで実現することで、例えば UPnP 側からセンサーデバイスに命令を送ることが可能になる。このような機能を提供できるゲートウェイによって、ホームネットワークを利用する消費者、一般ユーザーは専用端末の新規購入の必要や、煩雑な使い分けの必要がなくなるというメリットが生じる。

1.2. 目的

この仕様書では、1.1 節に記述されているようなニーズに応えうる DLNA/UPnP と ZigBee センサーネットワークサービス間をシームレスに接続するゲートウェイデバイスに関する仕様を定義し、実現することを目的としている。(図 1)これにより、1.3 節のようなメリットを、消費者に対して提供できる。

1.3. 概要

このゲートウェイ仕様に基づいた、ゲートウェイデバイスによって、ユーザーは所持している DLNA/UPnP 端末 (例えば携帯やモバイルビューワー、デジタルテレビやパソコン等) から、動画像、音声といった視覚的なコンテンツとして、センサー情報を取得・閲覧可能となる。また、UPnP の機能を利用して UPnP 端末からセンサーデバイスを操作することができるようになる。

ネットワークインフラとしてゲートウェイサービスを実現することで、端末に複数のネットワークインターフェイスを実装する場合と比べ、シンクライアントを実現することが可能となり、メーカーやアプリケーション開発におけるコスト低下をもたらすことができる。

1.4. 本仕様書の位置付け

本ゲートウェイ仕様の技術的な位置づけは図 1 で示される。

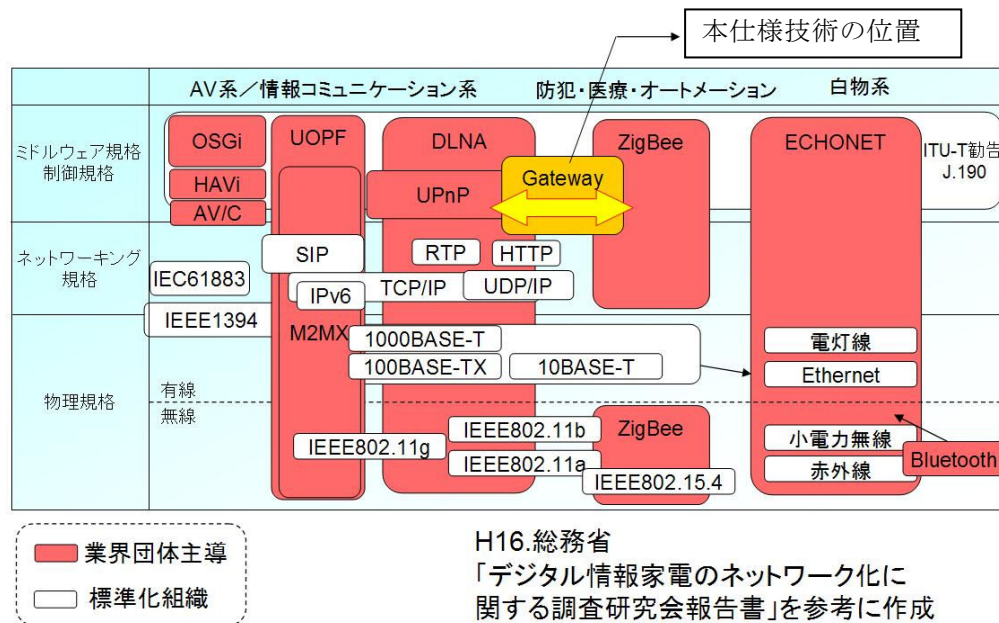


図 1. ゲートウェイの位置づけ

1.5. スコープ

本仕様の対象となる、DLNA、UPnP、ZigBee のバージョンは以下のとおりである。

DLNA

DLNA 1.0

UPnP

UPnP Device Architecture 1.0

UPnP AV Architecture 0.83

ZigBee

ZigBee1.0

2. プロトコル概要

2.1. UPnP 1.0

UPnP (Universal Plug and Play) は、ホームネットワーク上において機器を自動認識し、機器制御機能をプラグアンドプレイで提供することを目的としており、UPnP Forum によって規格化されている。図 2 に UPnP のスタック構成を示す。

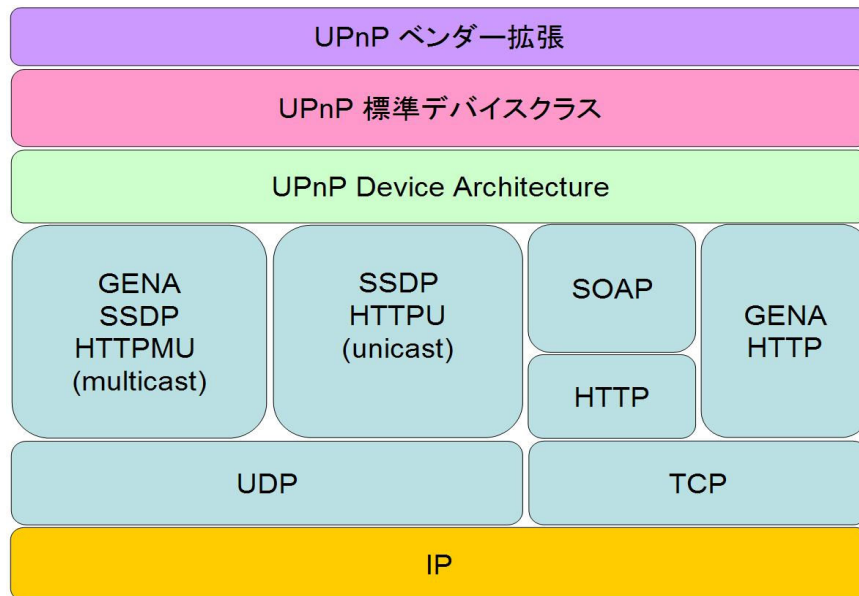


図 2. UPnP スタック構成

UPnP Device Architecture では、このようなアーキテクチャ上に、コントロールポイントとデバイスという 2 つの基本クラスを規定している。コントロールポイントは、他のデバイスの検出、及び、制御が可能なコントローラである。デバイスは、1 つ、もしくは、複数のサービスのコンテナであり、入れ子にすることもできる。サービスとは UPnP ネットワークにおける最小ユニットであり、状態を定義する状態変数 (State Variable) と、サービスを制御するアクションを持つ。

ープロトコル概要

ーSSDP (Simple Service Discovery Protocol)

ローカルエリア内におけるデバイスやサービスを発見するディスカバリー機能を実現するためのプロトコルである。コントロールポイントは、SSDP 検索要求 (discover) をマルチキャストすることによって関連するデバイス、又は、サービスを発見する。また、デバイスが新たにネットワークに追加された時、SSDP によるアナウンス (alive) をマルチキャストする。

ーGENA (Generic Event Notification Architecture)

状態変化を通知するためのイベントのプロトコルである。関連するコントロールポイントは、このイベント通知を受信するためにサブスクライブする。

ーSOAP (Simple Object Access Protocol)

デバイス制御を行うコントロールのためのプロトコルである。UPnP では SOAP を使用して制御メッセージを送信し、その結果をコントロールポイントに返す。

ーUPnP ネットワークのステップ

1) アドレッシング

UPnP では機器の起動後にアドレッシングを開始する。アドレッシングでは IP アドレスの設定が行われ、DHCP サーバーが利用可能な場合は割り当てられた IP アドレスが用いられ、そうでなければ AutoIP を使用する。

2) ディスカバリー

IP アドレスが設定されると、ディスカバリーが行われる。ディスカバリーは SSDP によって行われ、コントロールポイントはデバイスに関する情報を知ることができる。

3) ディスクリプション

ディスカバリーの後、デバイスとの対話を可能にするためにディスクリプションが行われる。デバイスディスクリプション (XML ファイル) を HTTP によって取得することで、デバイスが持つ詳細な情報 (所持するサービスの情報も含まれる) を知ることができる。

4) コントロール

そのデバイスから HTTP によってサービスディスクリプション (XML ファイル) を取得することでデバイスの制御が可能になる。デバイスを制御するために、コントロールポイントはアクション要求をデバイスのサービスに送信する。この制御メッセージは SOAP を使用して送信される。

5) イベント

デバイスのサービスは、状態変数が変化するとイベント通知を行う。また、コントロールポイントはこの情報を受信するためにサブスクライブを行う。イベントメッセージでは、GENA が使用される。

6) プレゼンテーション

プレゼンテーションのステップでは、デバイスにプレゼンテーション用の URL がある場合、コントロールポイントはこの URL からのページの取得や、ページのブラウザへの読み込み等ができる。

2.2. DLNA 1.0

2.2.1. DLNA スタック構成

DLNA は家庭内ネットワークにおけるマルチメディアコンテンツの共有と閲覧における、ベンダー製品間での相互運用性の確保を目的とした標準化を進めている。2005 年末には ver1.5、2006 年には ver2.0 が発表される予定で、ユースケースとデバイスクラスの追加、DTCP-IP 等のコンテンツ保護技術やモバイル対応、IPv6 への対応等の機能拡張を行っていく予定となっている。DLNA は以下の図のように既に標準化され、利用実績の高いプロトコル群から構成されている。(図 3)

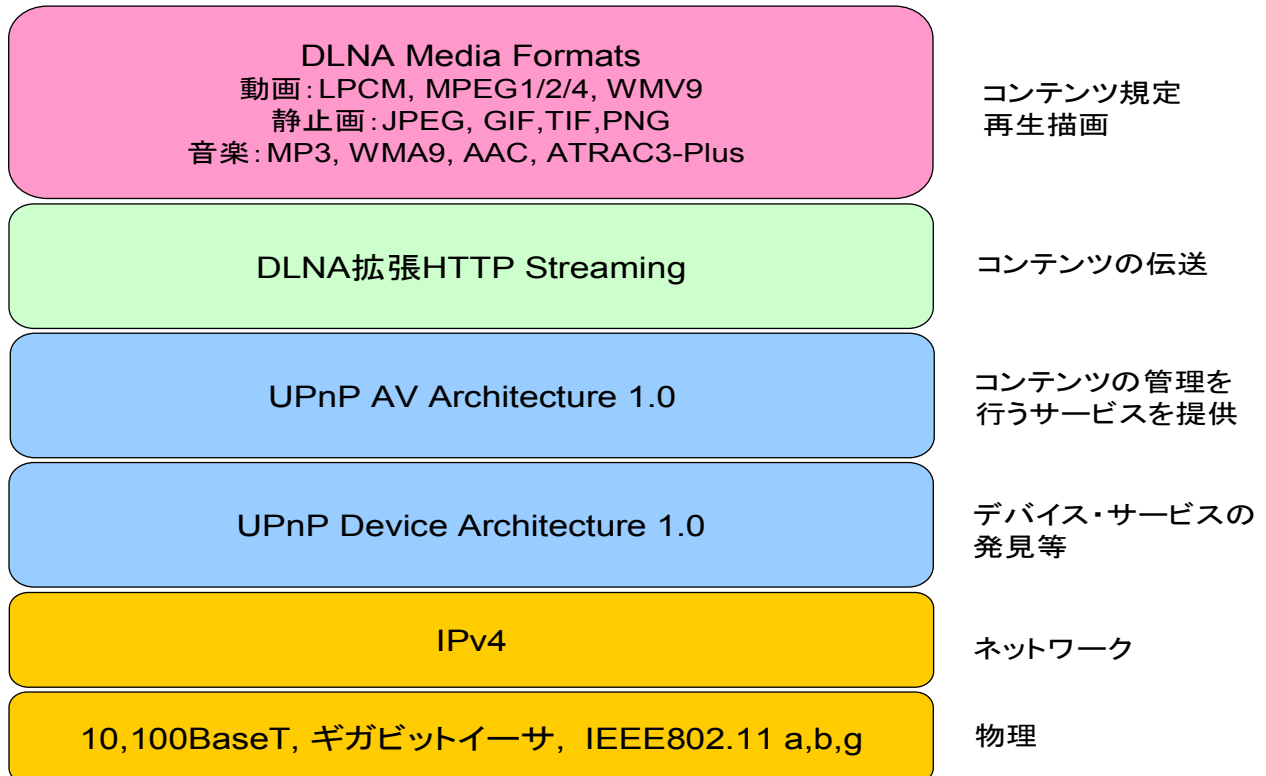


図 3. DLNA スタック構成

2.2.2. DLNA のユースケース

DLNA ではいくつかのユースケースシナリオが想定されており、それを実現するための Digital Media Device が規定されている。そのうち特に代表的なものとして 2Box Pull Usage がある。このユースケースシナリオは、Digital Media Server (DMS) と Digital Media Player (DMP) の二つのデジタルメディアデバイスから構成される。(図 4) ユーザーが所持する DMP は、DMS を UPnP Device Architecture の機能を用いて探索し、UPnP AV Architecture のサービスの一つである Content Directory Service(CDS)を用いて、コンテンツのインデックスライブラリを参照することができる。DMS は DMP からのコンテンツ要求に対して、HTTP ストリーミングによって DMP にコンテンツを配信する。

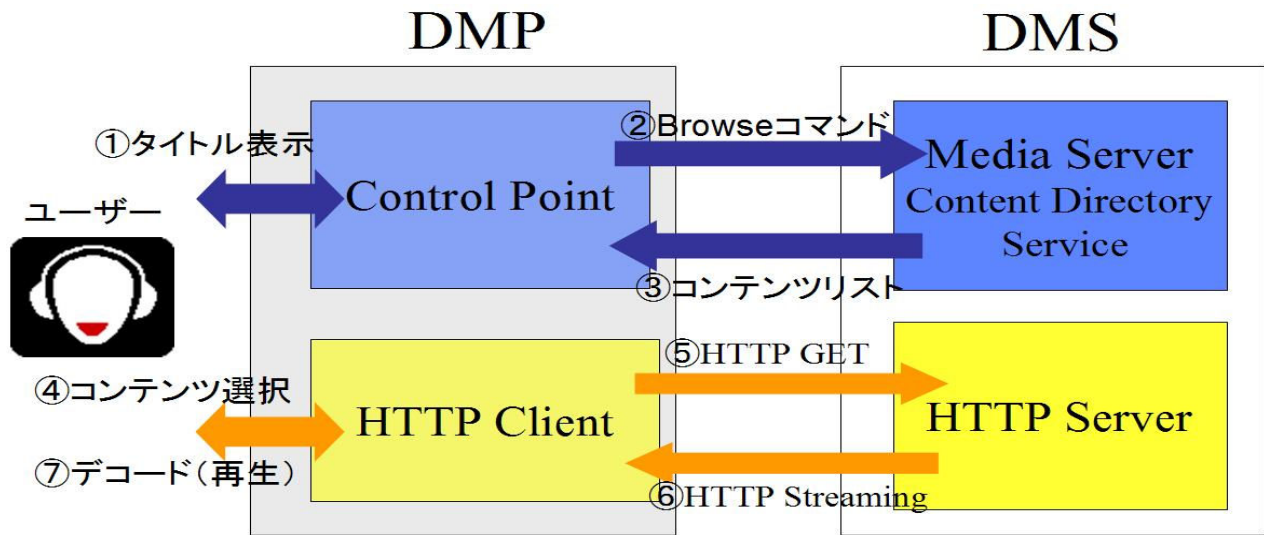


図 4. DLNA1.0 Digital Media Player と Digital Media Server

2.3. ZigBee 1.0

ZigBee は、アドホック、メッシュネットワーク、マルチホップを特徴とする無線 PAN の通信規格であり、ZigBee Alliance によって標準化が進められている。2004 年 12 月に ver1.0 が発表され、2005 年 6 月にその仕様が公開された。その後、2006 年 11 月に ver1.1 仕様が公開されているが、本 DLNA/UPnP-ZigBee ゲートウェイの仕様においては ver1.0 版を対象とする。ZigBee はセンサーネットワーク向けの仕様となっており、低消費電力、低コストの特徴を持つ。

ZigBee では、デバイスは 3 種類の ZigBee 論理デバイスタイプ (ZigBee コーディネータ、ZigBee ルータ、ZigBee エンドデバイス) に分類される。ZigBee コーディネータは、ネットワークの初期化を行う機能を持つデバイスであり、ZigBee ネットワーク上に一つだけ存在する。また、ZigBee ルータとしての機能も果たす。ZigBee ルータは、メッセージのルーティングとデバイスをネットワークへ接続する機能を提供する。ZigBee エンドデバイスは、ルーティングを行わず、アプリケーションが実装されるのみのデバイスである。

2.3.1. ZigBee スタック構成

ZigBee のスタック構成を図 5 に示す。

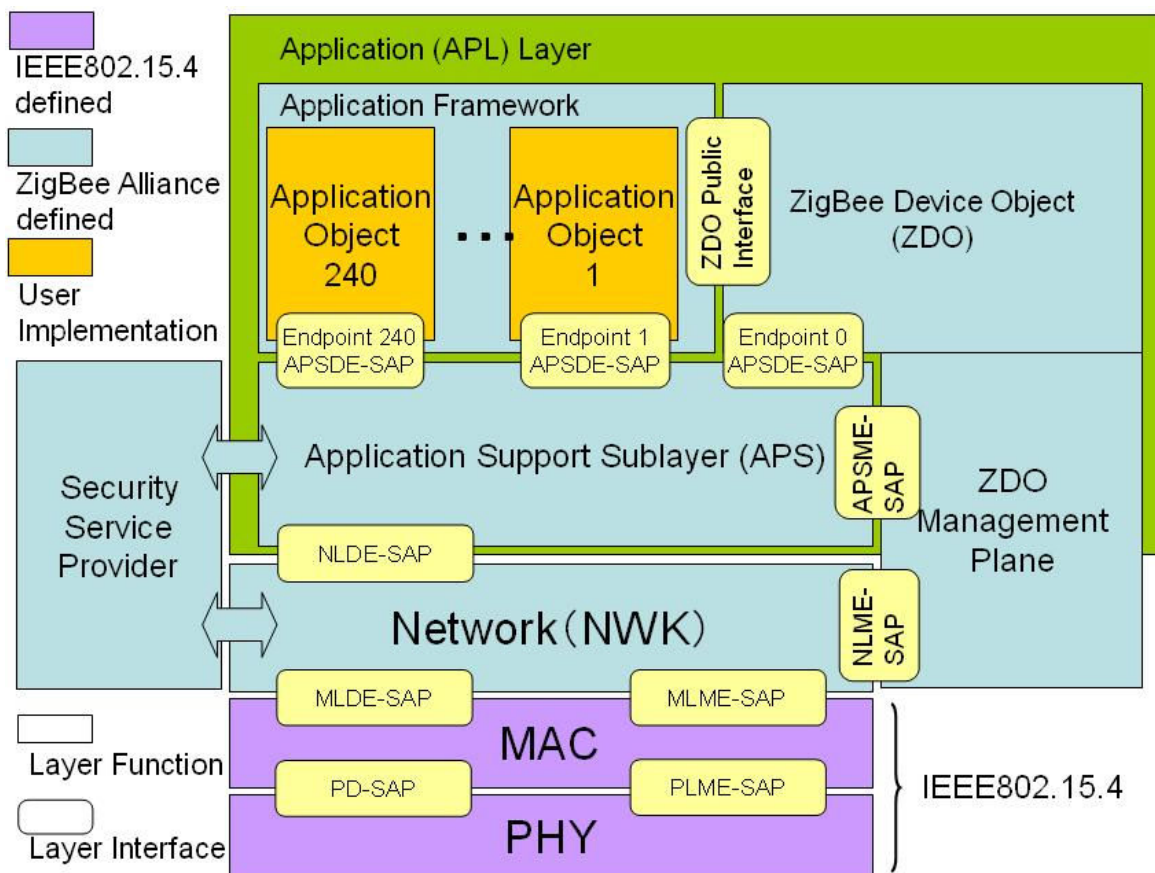


図 5. ZigBee スタック構成図

— 物理層(PHY)、MAC 層(MAC)

物理層と MAC 層には、IEEE802.15.4 が採用されている。周波数帯は、868MHz、915MHz、2.4GHz 帯を利用することができる。また、チャンネルアクセスには CSMA/CA が利用される。

ー ネットワーク層 (NWK)

ネットワーク層では、ネットワークの構築、及び、ルーティング制御が行われる。ZigBee では図 6 に示すように、3 種類のネットワークトポロジを構成できる。

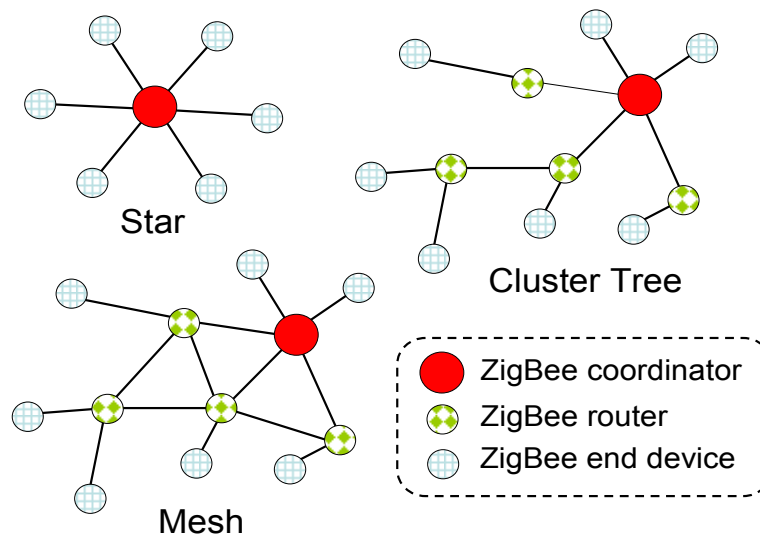


図 6. ZigBee におけるネットワークトポロジ

ZigBee ネットワークの構築は、まず ZigBee コーディネータによって開始される。ZigBee コーディネータは PAN ID を決定し、ネットワークの初期化シーケンスに入り、ネットワークを起動する。その後、ZigBee ルータや ZigBee エンドデバイスは、ZigBee コーディネータとネゴシエーションし、16bit short アドレスがネットワークアドレスとして割り当てられることによって、ネットワークに参加する。

ルーティング制御については、クラスタツリー型トポロジにおいてはクラスタツリールーティングが用いられ、ツリー構造に沿ったルーティングが行われる。メッシュ型トポロジにおいては IETF MANET WG によって標準化された AODV (Ad hoc On-demand Distance Vector) がルーティングに用いられる。

ー Application Support Sub-layer (APS)

ネットワーク層とアプリケーション層とのインターフェースの役割を果たし、通信する Application Object を識別するためのエンドポイント (1 から 240 番までが利用可能) を管理する。また、デバイスのバインディング機能を提供する。

ー Application Object

Application Object は、アプリケーションプロファイルに基づいて、ベンダーによってアプリケーションが実装されるオブジェクトである。アプリケーションプロファイルでは cluster が定義され、個々の cluster は 8 ビットの cluster ID により識別される。cluster は、状態変数となる attribute を最大 65,536 個まで持つことができる。Application Object は、図 7 のように input/output cluster 間で他のデバイスと attribute の値を交換することにより、アプリケーション機能の実行を行う。

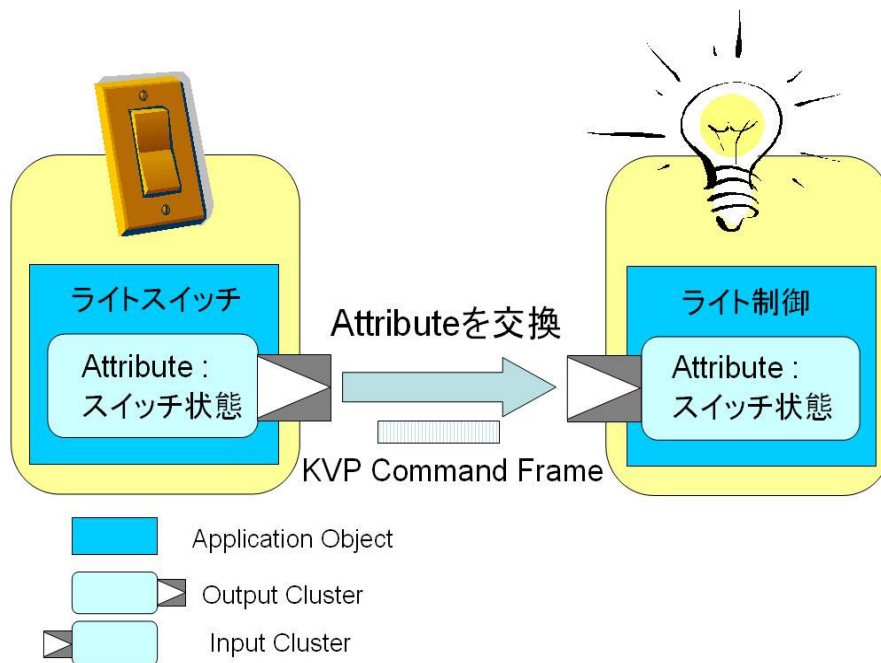


図 7. Application Object の attribute 交換

－ Application Framework

デバイスやアプリケーションが持つ能力を記述したディスクリプタと、アプリケーションオブジェクト間で通信を行うための仕組みである Key Value Pair(KVP)コマンドフレームや MSG フレームフォーマットをアプリケーションオブジェクトに対し提供する。

－ ZigBee Device Object (ZDO)

ZigBee Device Object は、必要となる共通機能を提供する。代表的な機能として、Service Discovery、Device Discovery、バインディング等がある。これらの機能を用いて、実装された Application Object 間でサービスを提供するための連携動作を行う。

3. DLNA/UPnP - ZigBee ゲートウェイ仕様

3 節では DLNA と UPnP、そして ZigBee ネットワークを接続するためのゲートウェイ仕様を規定する。3.1 節ではゲートウェイのシステムユースモデルをあげる。本仕様に基づいたゲートウェイが提供するユースケースは、採用される DLNA1.0 と UPnP Device Architecture1.0、及び ZigBee1.0 のユースケースモデルに基づいており、ユースケースの拡張可能性は DLNA, UPnP, ZigBee の今後のバージョンにおける拡張に依存する。**エラー! 参照元が見つかりません。**節ではゲートウェイに求められる接続仕様の概要について述べる。ここでは大きくデバイス間接続とサービス間接続に関する概略を述べる。3.3 節ではゲートウェイ仕様をさらに、UPnP と ZigBee 間の接続仕様と DLNA と ZigBee 間の接続仕様に分け、接続仕様をそれぞれ3.3 節と**エラー! 参照元が見つかりません。**節にて記述する。本仕様はネットワーク層を含まずに、UPnP と ZigBee のアプリケーション層における相互接続仕様と、ZigBee センサーネットワーク情報の DLNA コンテンツへの変換を行う際に必要となるアプリケーション用件のみを規定するものであり、特定のハードウェア環境や、ネットワーク環境には依存しない仕様であることに注意する必要がある。**エラー! 参照元が見つかりません。**節では、ゲートウェイが様々なサービスシーンにおいて用いられる可能性があること、及び各々のユースケースにおける様々なプロファイル設定を考慮する必要があることを示す。このことは本ゲートウェイ仕様 ver0.3 では確定していないことに注意すること。

3.1. ユースケースモデル

DLNA、UPnP と ZigBee ネットワークを接続し、相互運用を行うゲートウェイを構築するに当たり、まず、ゲートウェイのユースケースモデルを決定する。ユースケースモデルでは、主に以下の観点からゲートウェイの使い方やサービスを提示する必要がある。

- ・ UPnP ネットワーク上の UPnP デバイスに対して、ゲートウェイが提供するサービスと利用方法
- ・ UPnP ネットワーク上の DLNA デバイスに対して、ゲートウェイが提供するサービスと利用方法
- ・ ZigBee ネットワーク上の ZigBee デバイスに対してゲートウェイが提供するサービスと利用方法

以下に上述の3つに対する説明を行い、本仕様のゲートウェイが前提とするユースケースとする。

3.1.1. DLNA - ZigBee ゲートウェイユースケースモデル

本仕様において対象としている DLNA のバージョンは 1.0 であり、DLNA1.0 では Digital Media Server (DMS) と Digital Media Player (DMP)による 2-box pull system usage のみがサポートされている。そのため、DLNA との関係においてもこれに準ずる必要がある。本ゲートウェイでは DLNA デバイスが構築するネットワーク内において DMS として動作し、DLNA ネットワーク上の DMP に対し、センサーネットワークの現在の情報を画像情報として出力する。(図 8)

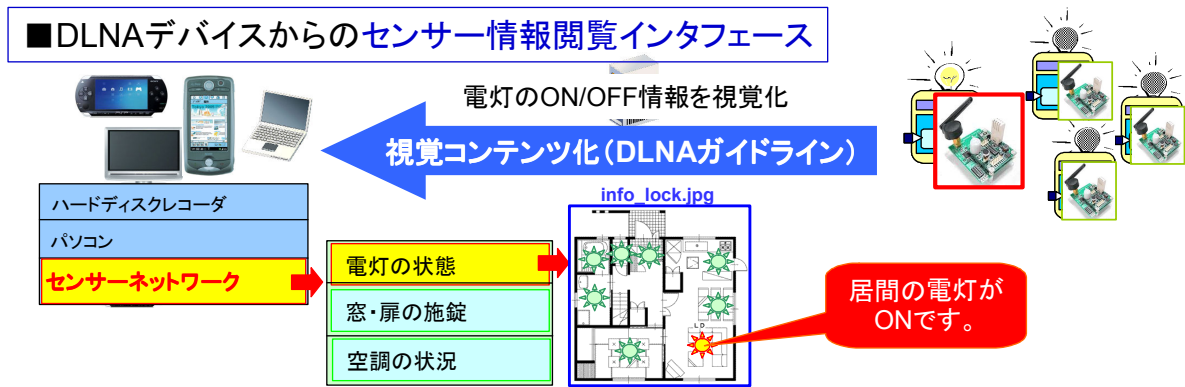


図 8. DLNA Digital Media Player からの ZigBee ネットワーク参照

この図の例では、DLNA ネットワーク上にある DMP から見て、ゲートウェイは DMS として見えていることを示している。ここではゲートウェイは「センサーネットワーク」という名前のフレンドリーネームを持つ DLNA DMS デバイスとして見えている。DMP はゲートウェイを発見し、CDS を利用することができる。DMP はゲートウェイの CDS に対し、Browse の SOAP リクエストを送信することで、ゲートウェイが提供するセンサー情報の分類である、コンテンツのインデックス情報を受け取ることができる。図 8 の例では、「電灯の状態」や「窓・扉の施錠」、「空調の状況」がそれに当たる。コンテンツを選択すると、ある一定のアプリケーションルールに基づいて DLNA コンテンツに加工されたセンサー情報が DMP に渡される。

UPnP-ZigBee 間の変換機能から見て、DLNA-ZigBee 変換機能は上位アプリケーション仕様となる。ここで留意すべき点としては、後述の UPnP と ZigBee の相互変換はレイヤとして同じレイヤ間での変換となるのに対し、DLNA と ZigBee では変換対象となる階層に釣り合いがとれず、ZigBee デバイスのデータを DLNA コンテンツへ変換する際の約束事（新たなアプリケーション仕様）を追加する必要が生じるということである。特に以下の点を新たな仕様として決定する必要が生じる。

- ・ DLNA コンテンツを生成する際に必要となる、ZigBee デバイスの属性情報の取得方法やタイミングに関する考慮
- ・ 取得した ZigBee 属性情報をどのようにして、まだどのような DLNA コンテンツへ整形を行うべきかのルールと、整形のためのルール（DLNA のコンテンツ閲覧におけるリクエストレスポンスの 30 秒ルールなどの固定パラメータを考慮する必要がある）、センサー情報からコンテンツを動的に生成し、Content Directory Service のコンテナに動的にデプロイするための仕組みと、コンテンツ化に利用するテンプレートやスタイルシートのものが必要となる問題に対する考慮
- ・ 生成したコンテンツのキャッシュや再利用性に関する考慮。例えば ZigBee ネットワーク上の ZigBee デバイスの電源供給が、電池駆動であったりする場合は、コンテンツの再利用率が高いと、電源が長持ちし、またレスポンスが大幅に上昇するが、トレードオフの関係でセンサー情報の信頼度は低下する。ZigBee デバイスの供給電源が、ACアダプタのような常時供給状態の場合は、キャッシュしたコンテンツを参照することなく、リアルタイムにその都度センサーデバイスに、属性情報取得クエリを送信することも考えられる。この場合データの信頼性は高いが、応答性において遅延が発生するかもしれない。

3.1.2. UPnP - ZigBee ゲートウェイユースケースモデル

本仕様において提供される UPnP と ZigBee の変換機能は、デバイス情報と、デバイスが提供するサービス（デバイスとサービスのディスカバリー、イベントメッセージ通信、制御メッセージ通信）が、UPnP ネットワークと ZigBee ネットワーク相互にやり取りを行うための機能を提供する。本ゲートウェイでは次のようなユースケースを実現するための機能を提供しなければならない。

■UPnPデバイスからのセンサー制御インターフェース



図 9. UPnP デバイスからの ZigBee デバイス操作

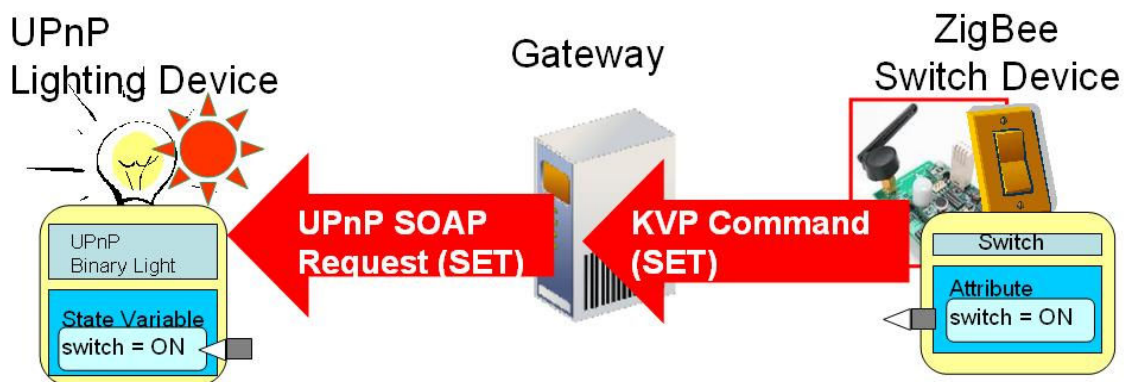


図 10. ZigBee デバイスからの UPnP デバイス操作

図 9 の例では、UPnP デバイスがゲートウェイ越しに ZigBee デバイスを発見し、制御を行うユースケースが描かれている。つまり、ゲートウェイでは UPnP からのメッセージを適切に変換し、UPnP が提供する発見・制御・イベント通信のサービス機能を ZigBee センサーネットワークに対し働かせる必要がある。このユースケースは本ゲートウェイにおける必須要求項目となる。

また、同様に図 10 で書かれているように ZigBee デバイスが UPnP をゲートウェイ越しに発見し、制御を行うユースケースにも対応する。つまりゲートウェイでは、UPnP ネットワークと ZigBee ネットワーク、両サイドからのメッセージを適切に変換し、望まれるレスポンスと機能を提供する必要がある。ただし、UPnP 側からの ZigBee センサーネットワーク側に対する制御に比べて、ZigBee センサーネットワークからの UPnP デバイスの制御は利用シーンがごく限られているため、このユースケースに関してはオプション項目となる。

3.2. 全体機能説明

ゲートウェイは、UPnP ネットワーク、または、ZigBee ネットワークから到達するパケットの宛先を適切に判断し、ルーティングを行う基本機能を提供する。これは図 11 の①End-End ルーティング変換に対応する。また、各々のメッセージパケット内部のデータを適切に変換し、メッセージパケットの種類を特定した上でプロトコルを適切に変換する。これは図 11 の②プロトコル変換に対応する。この①と②の機能はゲートウェイの UPnP-ZigBee 変換機能が提供する機能であり、ゲートウェイの基盤機能となる。

ゲートウェイは ZigBee ネットワーク上のデバイスが持つ属性情報を収集し、DLNA コンテンツを生成する機能、及び DLNA DMS として動作するアプリケーション機能を提供する。この機能は図 11③コンテンツ変換が相当する。DLNA-ZigBee 変換機能が提供するこの③の機能は、ゲートウェイの UPnP-ZigBee 変換機能から見て、上位の拡張アプリケーションとして位置づけられる。

注意しなければならない点として、本ゲートウェイで用いられる UPnP機能は3.1.1に記述された DLNA - ZigBee 間の相互接続仕様を中心の目的としたものであり、上位アプリとして位置づけられる DLNA コンテンツ生成をサポートする場合は、DLNA スタックの内部で利用される DLNA 拡張ヘッダ等を UPnP 変換機能において、別途サポートする必要がある。

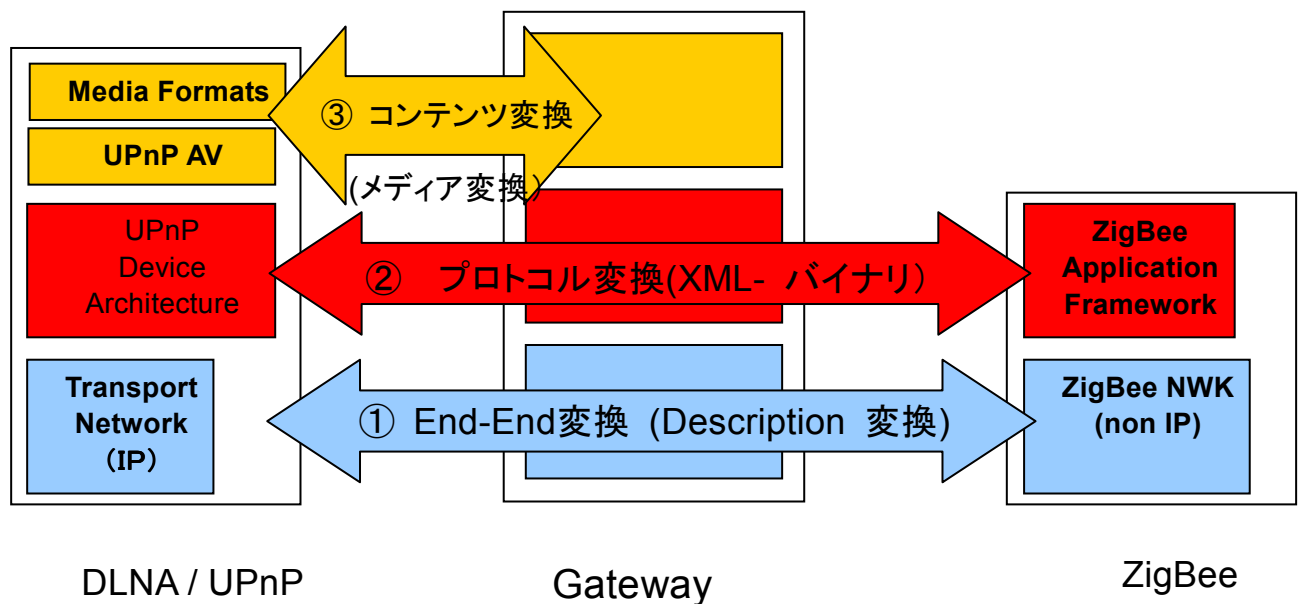


図 11. ゲートウェイが提供する変換機能

参考的なゲートウェイの内部構造を挙げると（あくまでも参考であり、以下に記述されている内部構造と、その実装構造は仕様レベルでは規定されない）、DLNA/UPnP-ZigBee ゲートウェイは図 12 のような内部構造となる。

DLNA/UPnP ネットワークに対しては、インターフェイスとして UPnP Device 及び ControlPoint が内部実装されている。また、DLNA コンテンツ配信のために、Digital Media Server が内部実装されている。ZigBee ネットワークに対しては、ZigBee coordinator がインターフェイスとして内部実装され、ZigBee ネットワークの情報を取得することができるようになっている。デバイスマッピングテーブルは、DLNA/UPnP デバイスと ZigBee デバイス間で双方向にメッセージングを行うための宛先アドレス情報を含んだデバイス情報のマッピングテーブルである。プロトコルメッセージ変換部では UPnP のメ

メッセージプロトコルと ZigBee のメッセージプロトコルを相互変換するための機能を提供する。ゲートウェイデータベースは変換情報データベースと状態情報データベースに内部で分割される。変換情報データベースは ZigBee デバイスが持つ機能と UPnP サービス情報を相互変換するための変換情報をストアするデータベースであり、状態情報データベースは、UPnP サービスの StateVariable や、ZigBee のクラスター内の attribute といった各デバイス上の状態情報をストアする目的を持ったデータベースである。Digital Media Server はゲートウェイ上に内蔵されたサーバー機能であり、それが持つ仮想 Content Directory Service 機能を用いて、DLNA 端末に対しセンサーの状況が判るコンテンツの index を渡す役目を持つ。センサー情報コンテンツ生成エンジンは、状態情報データベースにストアされた情報を元に、DLNA のメディアフォーマットに従ったコンテンツ、例えば決められたサイズの JPEG 画像などを動的に生成し、Media Server が発行しているコンテンツのインデックスで示される位置に配置する。

実際に実装をする上で重要な点として、インターフェイス上に動的に生成される、UPnP デバイス、CP オブジェクトや、ZigBee デバイスオブジェクト以外に、UPnP ネットワーク及び、ZigBee ネットワークをスキャンし、初期化をするためのデバイスオブジェクトが、最初に1つ必要となることに注意すること。

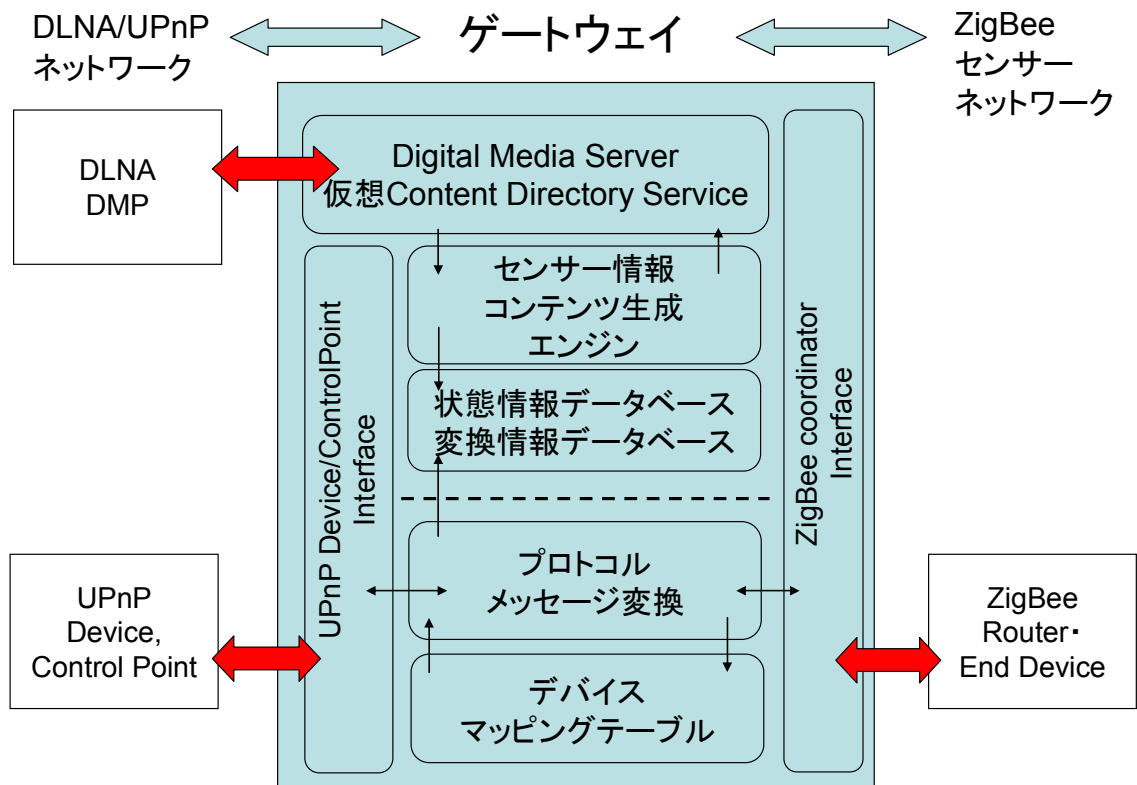


図 12. ゲートウェイの内部構造

3.3. UPnP-ZigBee 接続仕様

本節では、UPnP-ZigBee 間の接続仕様に関する要求を規定する。
UPnP-ZigBee の接続仕様の要求項目は表 1 のようになっている。

表 1. ゲートウェイ接続要求

操作側	操作される側	対応要求レベル
UPnP CP	ZigBee Device (インプット クラスタ)	MUST
ZigBee Device (アウト プットクラスタ)	UPnP Device	OPTION

UPnP CP から ZigBee Device を操作するための機能は必須で提供しなければならない。一方 ZigBee Device からの UPnP Device の操作に関してはオプションな扱いとなる。
これらの操作のシーケンスの詳細を以下に詳述する。

3.3.1. UPnP CP から ZigBee デバイスの操作に関わる基本シーケンス

UPnP CP から ZigBee デバイスの操作を行う場合のシーケンスは以下の通りとなる。

(0) ゲートウェイの初期化

まず、ゲートウェイ起動前の前提として、ZigBee コーディネータ機能が動作し ZigBee ネットワークアドレスの初期化が為されている必要がある。また、ゲートウェイ端末自体の IP アドレッシングも終了している必要がある。

ゲートウェイが起動する時には、ゲートウェイは初期化される。

初期化の内容として、ゲートウェイはまず、ZigBee ネットワークに対し、ZigBee 端末の発見クエリを送信し、ZigBee デバイス情報 (ノードディスクリプタ、シンプルディスクリプタ) の取得を行わなければならない。取得したディスクリプタ情報を用いて、ZigBee デバイスを制御するための仮想的な ZigBee Application Object(仮想 ZigBee Controller)をプロファイル ID ごとに1つ、立ち上げる。また、取得したディスクリプタ情報を元に、仮想的な UPnP Device のインスタンスを生成する。生成された UPnP Device のインスタンスは UPnP ネットワークに対し、SSDP-alive のメッセージを投げ、新規の仮想 UPnP デバイスがゲートウェイ上に作成されたことを通知する。通知を受けた UPnP ネットワーク上の UPnP コントロールポイントはデバイスからデバイスディスクリプションの取得を試みて、ダウンロードを行い、サービスディスクリプションの取得を行い、制御メッセージやイベント購読メッセージを送ることができるようになる。

ゲートウェイは以下の動作モードの分岐に従って、初期化動作を実行する。

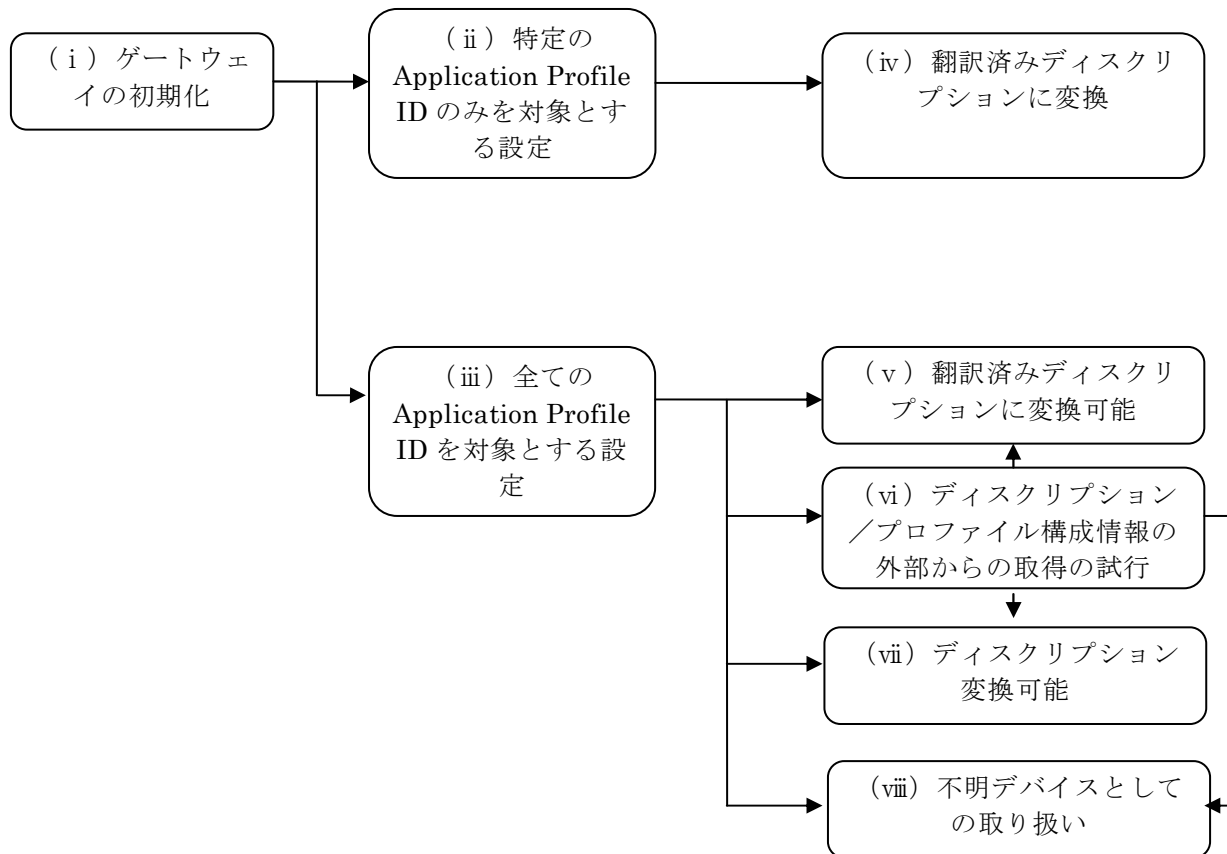


図 13. ゲートウェイの初期化動作時の ZigBee ネットワークの探索方法

(i) ゲートウェイの初期化

ゲートウェイの初期化時、ゲートウェイは変換対象とする ZigBee センサーネットワーク上のデバイスを決定する ii もしくは、iii へと遷移する。ii と iii の選択方法については、任意に選択・設定が可能であっても良いし、片方のみの実装も許可される。

(ii) 特定の Application Profile ID のみを対象とする設定

ゲートウェイが起動後、ZigBee ネットワーク上のデバイスを探索する際に、あらかじめ指定された Application Profile ID を指定し探索することで、特定の Application Profile ID を持った ZigBee デバイスのみの仮想 UPnP デバイスを生成するモードであり、探索の結果 iv に遷移する。なお、このモードにおいて指定可能な Application Profile ID は、ZigBee アライアンスによって規定され、ベンダーに依存しない標準プロファイルを想定しており、翻訳済みディスクリプション / プロファイル構成情報をゲートウェイ内部に予め設定して保持している（もしくは外部取得が可能である）ものに限定される必要がある。

(iii) 全ての Application Profile ID を対象とする設定

ゲートウェイが起動後、ZigBee ネットワーク上のデバイスを探索する際に、特定の Application Profile ID のみを対象とするのではなく、ZigBee ネットワーク上に存在する全てのデバイスを探索し、仮想デバイスを生成するモードである。探索結果に応じて、v, vi, vii, viii のいずれかに遷移する。

(iv) 翻訳済みディスクリプションに変換

ii の動作によって、指定された特定の ZigBee デバイスの探索を行った結果として、ZigBee の Descriptor が入手される。入手された Descriptor を元に仮想 UPnP デバイスを生成する際に、あらかじめ用意された翻訳済みの UPnP Description を参照し、仮想 UPnP デバイスの生成を行う。

(v) 翻訳済みディスクリプションに変換可能

iii もしくは vi の動作結果として、発見できた ZigBee Device が持つ ZigBee Application Profile ID のうち、翻訳済みの UPnP ディスクリプションをゲートウェイが所持している場合は、その翻訳済みディスクリプションの情報を元に、仮想 UPnP デバイスを生成する。

(vi) ディスクリプション／プロファイル構成情報の外部からの取得の試行

iii の動作結果として、発見できた ZigBee Device が持つ ZigBee Application Profile ID に対して、翻訳済みの UPnP ディスクリプション、プロファイル構成情報をゲートウェイが所持していない場合は、外部の参照先（http などによるダウンロードや、Web Service による実現などが考えられる）からの取得を試み、取得できた場合は、v の動作に遷移する。翻訳済み UPnP ディスクリプションが取得できず、プロファイル構成情報が取得できた場合は vii の動作に遷移し、プロファイル構成情報が取得できなかった場合は viii の動作に遷移する。

(vii) ディスクリプション変換可能

iii もしくは vi の動作結果として、発見できた ZigBee Device が持つ ZigBee Application Profile ID に対して、プロファイル構成情報をゲートウェイが所持している場合は、ディスクリプション変換対象デバイスとして取り扱う。この場合、UPnP Device Description の変換に関しては、取得できた ZigBee Descriptor 情報と、DeviceDescription 変換ルールに基づき生成する。UPnP ServiceDescription の変換に関しては、取得できた ZigBee Descriptor 情報の中に含まれる Cluster 情報とプロファイル構成情報、及び ServiceDescription 変換ルールに基づき、ServiceDescription を生成する。これらの UPnP Description 情報に基づき、仮想 UPnP デバイスの生成を行う。

(viii) 不明デバイスとしての取り扱い

iii もしくは vi の動作結果として、発見できた ZigBee Device が持つ ZigBee Application Profile ID に対して、翻訳済みの UPnP ディスクリプション、及び、プロファイル構成情報をゲートウェイが所持していない場合は、サービス不明デバイスとして取り扱う。この場合、UPnP Device Description の変換に関しては取得できた ZigBee Descriptor 情報と、DeviceDescription 変換ルールに基づき生成する。ServiceDescription の変換に関しては、取得できた ZigBee Descriptor 情報の内容に関わらず、固定的な UPnP ServiceDescription を生成する。これらの UPnP Description 情報に基づき、仮想 UPnP デバイスの生成を行う。

(i) から (viii) における遷移は、ZigBee Descriptor 情報と UPnP Device Description、ZigBee Cluster 情報と UPnP Service Description、それぞれの変換において、独立に遷移することが考えられることに注意すること。

(1) デバイス発見

UPnP ネットワーク上で UPnP コントロールポイントが立ち上がると、デバイスのディスカバリー動作である SSDP-discover メッセージの送信を行う。このメッセージは HTTP-MU によるマルチキャストメッセージである。ゲートウェイがこのディスカバリーメッセージを受け取ると、ゲートウェイではまず、初期化の際に既に生成している仮想 UPnP デバイスがそのまま UPnP デバイスとしてのレスポンスを返す。具体的には SSDP-alive メッセージである。

次にゲートウェイは、ZigBee ネットワークに対して内部に所持する ZigBee コーディネータの機能を用いて、ZigBee デバイスのディスカバリー要求を行う。ただし、このフォワーディングにおけるルールは後述のキャッシング、ノンキャッシングのモードに応じた振る舞いを行う必要がある。ZigBee デバイスのディスカバリー要求は具体的には、ZigBee ネットワークに参加しているデバイスが持つ、ノー

ドディスクリプタとシンプルディスクリプタの取得要求である。この際、発見したデバイスのプロファイル ID に対応する仮想的な ZigBeeController が存在しない場合、新規に ZigBeeController を生成する。取得できた ZigBee デバイスのディスクリプタ情報は、翻訳済みディスクリプションが存在しない場合、後述のディスクリプション変換ルールに従い UPnP のデバイスディスクリプションフォーマットに適切に変換される。翻訳済みディスクリプション、もしくは変換されたデバイスディスクリプションを元に、新規に仮想的な UPnP デバイスのインスタンスを生成する。(この詳細に関しては (2) にて記述される) 生成された仮想 UPnP デバイスは SSDP-alive を UPnP ネットワークに対し通知する。こうすることで、UPnP ネットワーク上のコントロールポイントは、ZigBee デバイスをゲートウェイ上の仮想的な UPnP デバイスとして認識する。

(2) 仮想 UPnP デバイスの生成

仮想 UPnP デバイスを生成するに当たり、デバイスディスクリプションとサービスディスクリプションを動的に生成する必要がある。この際、対象とするアプリケーションプロファイル ID が前述(1)の、(iv)、(v)の翻訳済みディスクリプション変換シーケンスに入る場合、デバイスディスクリプションには、翻訳済みディスクリプションを使用する。(vii)のディスクリプション変換可能シーケンス、(viii)の不明デバイスとしての取り扱いシーケンスに入る場合は、シンプルディスクリプタ情報を元に、ディスクリプタ-デバイスディスクリプション変換ルールに基づいて変換が行われる。

サービスディスクリプションの変換については、対象とするアプリケーションプロファイル ID が前述(1)の、(iv)、(v)の翻訳済みディスクリプション変換シーケンスに入る場合、サービスディスクリプションは、翻訳済みディスクリプションを使用する。(vii)のディスクリプション変換可能シーケンスに入る場合には、クラスター-サービスディスクリプション変換ルールに基づいて変換が行われる。(viii)の不明デバイスとしての取り扱いシーケンスに入る場合は、プロファイル中に含まれるクラスターの詳細情報が不明となるため、アクションの存在しないデフォルト(lookup)サービスディスクリプションに変換される。

こうして変換されたデバイスディスクリプションと、サービスディスクリプションを元に、仮想デバイスを初期化し立ち上げる。仮想デバイスは ZigBee アプリケーションオブジェクト一つに対し、ルートデバイスとして一つが割り当てられる関係となる。複数のルートデバイスに対するポートの割り当て方法は実装に依存する。生成された仮想 UPnP デバイスは、変換されたディスクリプションによって初期化され、起動する時には UPnP ネットワークに対し、アドバタイズ SSDP-alive を行う必要がある。

(3) サービスの発見

仮想的な UPnP デバイスから SSDP-alive を受け取った UPnP コントロールポイントは、その仮想 UPnP デバイスに対し、デバイスディスクリプションを要求する (http リクエスト)。UPnP コントロールポイントでは取得できた UPnP デバイスディスクリプションの情報を元に、デバイスが提供するサービスディスクリプション情報の取得 (http リクエスト) を試みる。ゲートウェイは (2) に書いてあるように、ZigBee ネットワーク上の ZigBee デバイスを、ディスクリプション変換ルールと、クラスター変換ルールに基づいて、UPnP デバイスディスクリプション及びサービスディスクリプションに変換して、UPnP ネットワークに対し公開する。UPnP ネットワーク上のコントロールポイントは、ゲートウェイ上に展開された仮想的な UPnP ルートデバイスを発見すると、次にゲートウェイの複数のポートに展開されたルートデバイスからデバイスディスクリプションを取得し、次に取得したデバイスディスクリプションからサービス情報を取得し、指定された URL から、(2) で記述されている、新しく生成したサービスディスクリプションの取得 (http リクエスト) を行う。

(4) サービス制御

(3) で取得したサービスディスクリプションの記述内容によって、実現可能なアクションと、操作対象となる状態変数と、操作の範囲指定などの情報に従った振る舞いを、UPnP コントロールポイント側では行う。ゲートウェイ上の仮想 UPnP デバイスに対するサービス制御は具体的には、SOAP リクエス

トとして実現されるが、ゲートウェイではそれを仮想 ZigBee Controller から KVP コマンドフレームと
いった ZigBee のデータフレームに変換し送りだす必要がある。その際に必要な変換として以下を考慮
する必要がある。

- SOAP リクエストメッセージの KVP コマンドフレームへの変換
- 宛先の変換として、UPnP サービス URL から、ZigBee デバイスのネットワークアドレス+エン
ドポイント番号+プロファイル ID への変換
- SOAP アクションの KVP コマンドへの変換
- 操作対象の変換として、UPnP StateVariable から ZigBee Attribute への XML バイナリ名前変換
- データの変換として、StateVariable の設定値から、Attribute の設定値への XML バイナリ値変
換

一般に、ZigBee で与えられる標準的な操作として GET と SET のコマンドが提供されているため、SOAP
コマンドにおいても、GET と SET が標準対応として提供される。また、操作対象の変換と、そのデー
タの変換はデータフォーマットの変換を行わなければならないことに注意する必要がある。

(5) イベント制御

イベント制御のためのメッセージ通信は、ZigBee デバイスから UPnP コントロールポイントに向けた
方向となる。そのために必要な前処理として、UPnP コントロールポイントと、ゲートウェイ上の仮想
UPnP デバイス間での、UPnP プロトコル上通常のイベントメッセージ通知のためのシーケンス、つま
り、GENA-Subscribe (UPnP コントロールポイント) > HTTP 200 OK (仮想 UPnP デバイスから
の確認応答) > GENA-Notify (仮想 UPnP デバイスからのイベント通知) > HTTP 200 OK (UPnP コ
ントロールポイントからの確認応答) という流れの中の最初の GENA-Subscribe 要求メッセージを行っ
ておかなければならない。ZigBee デバイスがネットワークに対してイベントを送信した場合、検知し
たゲートウェイ上の仮想 ZigBeeController は KVP-Event を GENA に変換し、送信元 ZigBee デバイス
に対応する仮想 UPnP デバイスへと Subscribe を行った全てのコントロールポイントに対して
GENA-Notify を送信する。この際、KVP コマンドフレームの GENA プロトコルへの変換として、以下
を考慮する必要がある。

- KVP コマンドフレームの GENA-Notify メッセージへの変換
- 送信元デバイス情報の変換として、ZigBee デバイスのネットワークアドレス+エンドポイント番
号から、対応する仮想 UPnP デバイスの識別子(デバイス UUID)
- イベント送信対象の変換として、ZigBee Cluster から、対応する仮想 UPnP デバイスサービスの
識別子 (サービス ID)
- イベント対象の変換として、ZigBee Attribute から UPnP State Variable への XML バイナリ名
前変換
- データの変換として、Attribute の設定値から State Variable の設定値への XML バイナリ値変換

イベント対象の変換と、そのデータの変換はデータフォーマットの変換を行わなければならない。また、
注意すべき点として、UPnP ネットワーク上のコントロールポイントは、もともと、ZigBee デバイス
と通信する目的を持って UPnP 通信機能が組み込まれている必要がある。具体的には ZigBee デバイス
を通信する目的を持って、あらかじめゲートウェイによって変換されることを前提としたデバイスディ
スクリプション情報を持っている必要がある。その機能ははじめから組み込まれていても良いし、動的
にダウンロードして組み込まれても良く、その方法についてこの仕様書では規定しない。

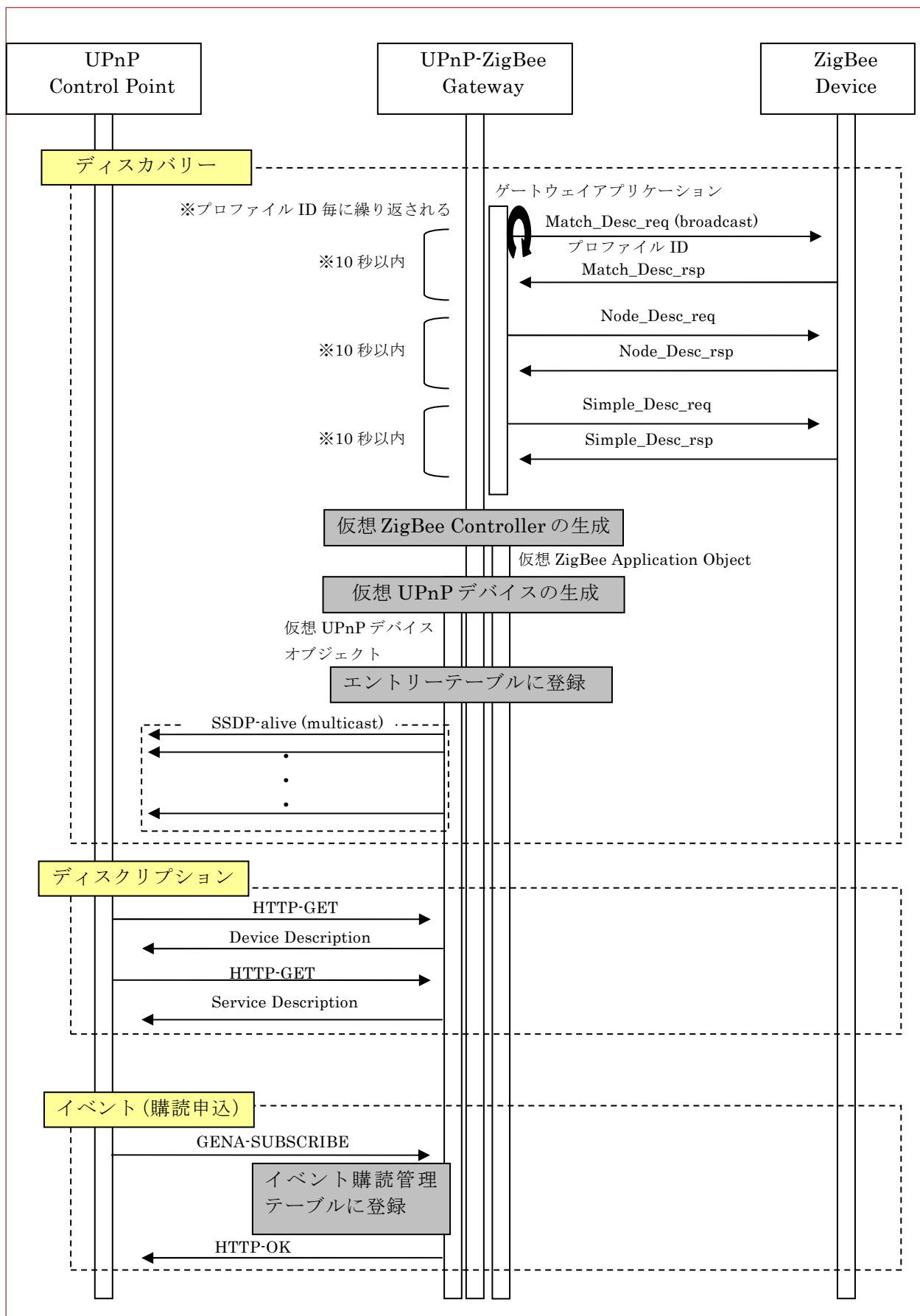


図 14. UPnP CP からの ZigBee デバイス操作に関する初期化シーケンス
(対象とするアプリケーションプロファイルを指定する場合)

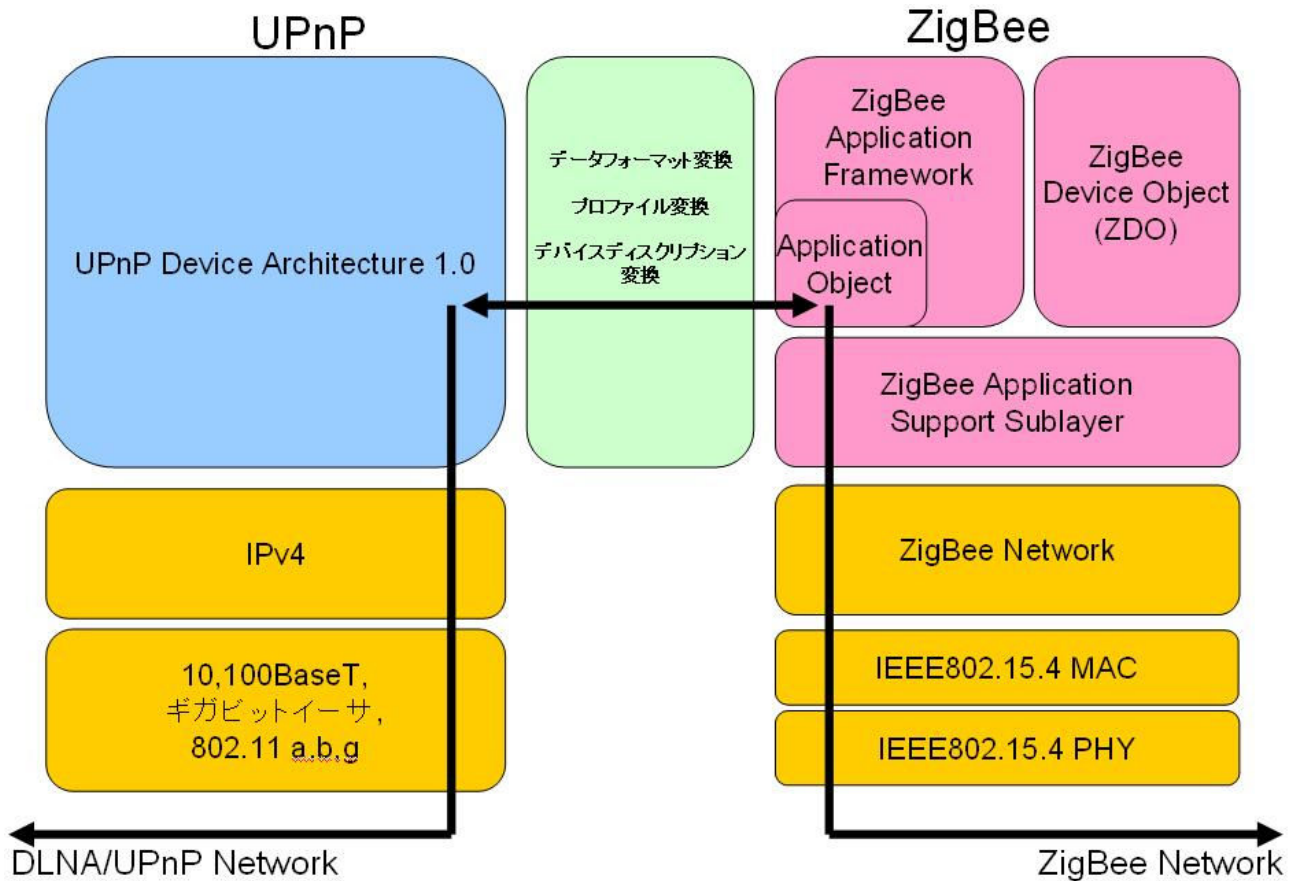


図 15. ゲートウェイのスタック間接続図

3.3.2. ZigBee デバイス (コントローラ側) から UPnP デバイスの操作に関わる基本シーケンス (オプション)

ゲートウェイは、コントローラとなる役目を持つ ZigBee デバイスから、UPnP デバイス进行操作するための機能も提供することができる。3.3.1 節と組み合わせることにより、UPnP – ZigBee 間双方向での操作が可能となるが、将来的な拡張の意味における仕様である位置づけが高く、2006 年時点において、ZigBee デバイスから UPnP デバイス进行操作する要求が低いことと、ゲートウェイの負荷と、必要となる動作環境を下げる意味合いから、この節における機能はオプションとして位置づけられ、実装は強要されない。

(0) ゲートウェイの初期化

ゲートウェイが起動する時には、ゲートウェイはネットワークを初期化し、ネットワーク上のリソース情報を初期化する必要がある。

初期化の内容として、ゲートウェイはまず、UPnP ネットワークに対し、UPnP デバイスの発見メッセージ (SSDP-discover) を送信し、UPnP デバイスが持つデバイス情報 (デバイスディスクリプション) の取得を行わなければならない。取得したデバイスディスクリプション情報を元に、仮想的な ZigBee Application Object (仮想 ZigBee アプリケーションオブジェクト) のインスタンスを生成し、ZigBee coordinator を立ち上げる必要がある。生成された ZigBee Application Object を所持する ZigBee coordinator は ZigBee ネットワークを立ち上げ、初期化を行う。その後、ZigBee ネットワークに参加した ZigBee デバイス (コントローラ) は、ZigBee ネットワーク上に、接続したい ZigBee デバイスの発見要求を行うと、ZigBee coordinator 上に生成された、仮想的な ZigBee オブジェクトを発見することができる。

(1) デバイスの発見

ZigBee ネットワーク上で ZigBee デバイス (コントローラ側) が立ち上がり、ZigBee coordinator が初期化し、提供を開始している ZigBee ネットワークに参加すると、ZigBee ネットワーク上で、操作対象となる ZigBee デバイスの発見要求のためのシーケンスを実行する。このとき発見要求を受け取った ZigBee coordinator 上に生成された仮想的な ZigBee Application Object は、目的のものとして応答を返す。この際の応答としては、ネットワークアドレスや、マックアドレス応答 (この場合はゲートウェイに組み込まれている ZigBee coordinator のアドレスである) や、ディスクリプタ要求がある。

また、既にゲートウェイの初期化が完了している状態で、UPnP デバイスが新たに UPnP ネットワーク上で起動したとき、ゲートウェイに向けてアドバタイズメッセージ (SSDP-alive) が送信される。アドバタイズメッセージを受信したゲートウェイは、アドバタイズの送信元デバイスのデバイスディスクリプション情報を取得する。そして、デバイスディスクリプションからサービスディスクリプションを取得する。取得したデバイス及びサービスディスクリプションを適切なシンプルディスクリプタ及び、クラスター情報に変換し、ZigBee coordinator の空きエンドポイント上に、仮想的な ZigBee アプリケーションオブジェクトとして配置する。その情報はネットワーク中にアナウンスする必要がある。

(2) 仮想 ZigBee アプリケーションオブジェクトの生成

仮想的な ZigBee アプリケーションオブジェクトを生成するに当たり、シンプルディスクリプタ情報とクラスターを動的に生成する必要がある。シンプルディスクリプタは、変換元になる UPnP デバイスのディスクリプション情報から生成する。また、クラスターはデバイスディスクリプション情報から取得できるサービスディスクリプションから生成する。クラスターを構成する属性情報 (Attribute) を、サービスディスクリプション中に記載されている変数情報 (StateVariable) から生成し、クラスター ID に

は、サービスディスクリプションに記載されるサービス ID から一意なものを、ベンダー拡張範囲で動的に設定する必要があることに注意する。

生成されるべきアプリケーションオブジェクトを構成する要素は、属性を含んだクラスターと、その操作方法であり、操作は GET と SET に限定され、KVP コマンドフレームを通してクラスターはアクセスされる。

(3) サービスの発見

ZigBee デバイス (コントローラ) は、ゲートウェイが提供する ZigBee coordinator 上に展開された仮想的な ZigBee Application Object から取得したシンプルディスクリプタ情報に含まれるインプットクラスター情報を元に、接続すべきクラスター ID 情報のリストを取得することができる。

(4) サービス制御

(3) で取得したクラスター ID に対して、あらかじめ決められている値の、書き込み (SET)、読み込み (GET) に限定される操作を、KVP コマンドフレームを用いて行う。送られた KVP コマンドフレームはゲートウェイによって、UPnP SOAP メッセージに変換され、UPnP ネットワーク上の宛先となる UPnP デバイスに送り届けられる。

一般に、ZigBee で与えられる標準的な操作として GET と SET のコマンドが提供されていることを考慮し、変換先の UPnP SOAP メッセージにおいても、GET と SET アクションが標準対応として提供されることに注意する必要がある。また、操作対象の変換とそのデータの変換はデータフォーマットの変換を通す必要があることに注意する。

(5) イベント制御

ここにおける、イベント制御のためのメッセージ通信は、UPnP デバイスから ZigBee デバイス (コントローラ) に向けたイベントの方向となる。UPnP デバイスから見て、イベントを送信すべき ZigBee ネットワーク上の ZigBee アプリケーションオブジェクトは、ゲートウェイの持つ UPnP コントロールポイントとして見えている必要がある。そのために必要な前処理として、ゲートウェイは発見した全ての UPnP デバイスに対し Subscribe を行い、HTTP 200 OK の了承通知メッセージを受け取る。このイベント通知設定がなされた状況で、UPnP デバイスが GENA Notify を UPnP コントロールポイントに通知すると、GENA を適切に解釈し、UPnP デバイスにマッピングされている仮想 ZigBee アプリケーションオブジェクトから、EVENT で設定された KVP コマンドフレームをブロードキャストで送信する。ZigBee デバイス(コントローラ)は KVP コマンドフレームが監視対象のイベントであった場合、当該メッセージを受信する。

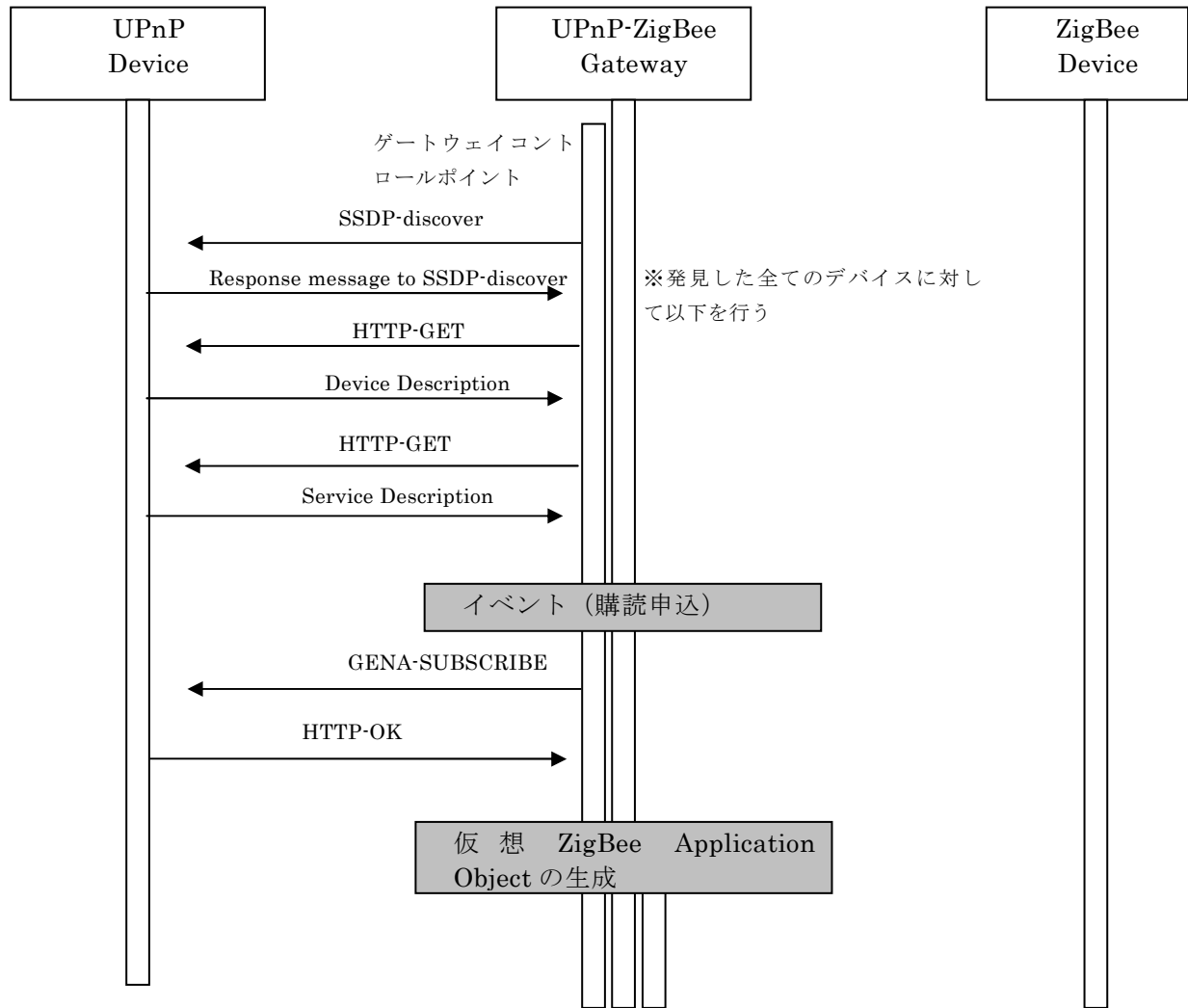


図 16. ZigBee デバイス(コントローラ)からの UPnP デバイス操作に関する初期化シーケンス

3.3.3. 機能説明

ゲートウェイのユースケースにおいて利用される UPnP と ZigBee 間におけるデバイス相互接続および、サービス間接続に関する対応は以下のようになっている。

3.3.3.1. データフォーマットの変換

ZigBee 上でやりとりされる情報は基本的にバイナリであるのに対し、UPnP ネットワーク上でやり取りされる情報は、UTF-8 文字エンコードされた XML ないしは、テキスト表現の情報である。ZigBee が扱うデータフォーマットの型は、仕様で規定されているため、その情報からやりとりするデータを適切な型として読み取り、UTF-8 エンコード文字列に変換した上で、制御メッセージのプロトコルにラップし、UPnP ネットワークの宛先デバイスに向けて送信する必要がある。また、UPnP ネットワークでやり取りするデータは、サービスディスクリプション中に記述される UPnP の型に従い、UTF-8 エンコード文字列から適切な ZigBee データ型に変換した上で、バイナリデータとして制御メッセージのプロトコルにラップし、ZigBee ネットワークの宛先デバイスに向けて送信する必要がある。

3.3.3.2. ディスクリプション変換

- UPnP デバイスと ZigBee アプリケーションプロファイル
UPnP デバイスが持つ XML 表記される UDN (デバイスネーム) 情報は ZigBee アプリケーションプロファイル ID とマップされる。
- UPnP ルートデバイスと ZigBee アプリケーションオブジェクト (エンドポイント番号が 1 から 240)
UPnP デバイスディスクリプションは、ZigBee デバイスが持つノードディスクリプタ情報と ZigBee アプリケーションオブジェクトが持つシンプルディスクリプタ情報から変換・生成される。
- UPnP サービスと ZigBee クラスタ
UPnP サービスディスクリプションの XML 表記されるサービスタイプには、ZigBee クラスタ ID を記述する必要がある。ZigBee の異なるエンドポイント上にある同一 ZigBee クラスタ ID を持つクラスタを、UPnP サービスディスクリプションに変換する際には、異なるサービス ID を生成し割り振る必要がある。

3.3.3.3. プロトコル変換

- UPnP SOAP メッセージと ZigBee KVP コマンドフレーム
- UPnP SOAP メッセージと ZigBee MSG コマンドフレーム
制御メッセージとして、UPnP SOAP メッセージと ZigBee KVP コマンドフレームが変換される。また、MSG コマンドフレームは、同様に UPnP SOAP メッセージに変換されるが、フォーマットフリーであるため、相互接続の保障は提供しないベンダー拡張依存とする。
- UPnP SOAP Action と ZigBee KVP コマンド
UPnP SOAP Action は ZigBee KVP コマンドに変換される。基本コマンドとして、SET with ACK, GET with ACK コマンドが用いられる。
- UPnP StateVariable と ZigBee Attribute
UPnP サービスディスクリプション中に記載される状態変数セットを構成する StateVariable の名前は、ZigBee Attribute ID が記述される必要がある。同時に状態変数に利用される変数の型や、範囲のパラメータもマップされる必要がある。

-
- **UPnP GENA イベントメッセージと ZigBee KVP コマンドフレーム**
UPnP のイベントメッセージである GENA-notify メッセージは、ZigBee KVP コマンドフレームの EVENT with ACK コマンドと変換される。
 - **UPnP SSDP メッセージと ZigBee Device Profile**
UPnP SSDP メッセージが、透過的にゲートウェイを超える場合は、ゲートウェイの ZigBee ネットワークに対するディスクリプタ要求を行う ZigBee Device Profile メッセージに変換される。

3.3.4. データフォーマット変換

UPnP ネットワークと ZigBee ネットワークを接続するときには、プロトコルやディスクリプションの方式の変換が必要となるが、同時に内容であるデータのフォーマットを変換する必要が生じる。UPnP では UTF-8 による文字エンコード表現が用いられるが、ZigBee においてはバイナリ表現と一部テキスト表現が用いられる。データフォーマットの変換が必要となるのは、UPnP においては、デバイスディスクリプション、サービスディスクリプション、SOAP リクエストメッセージボディ、SOAP レスポンスメッセージボディ、GENA(Notify, Response)メッセージボディである。ZigBee では、ディスクリプタ（ノードディスクリプタ、シンプルディスクリプタ、コンプレックス等のオプションディスクリプタ）、KVP コマンドフレーム内の Attribute の情報がデータフォーマット変換の対象となり、デバイス・サービス情報の取得と、プロトコル変換時にデータフォーマットも同時に翻訳される必要がある。

3.3.4.1. 各プロトコルのペイロードにおけるデータフォーマット変換

この節では、ZigBee アプリケーションオブジェクトからの KVP コマンドフレームのデータ送信時において、データペイロードにある属性情報に関するデータフォーマット変換についての取り決めを行う。

以下に KVP コマンドフレームのフレームフォーマットをあげる。

表 2. ZigBee KVP コマンドフレームフォーマット

Bits : 4	4	16	0/8	Variable
Command type identifier	Attribute data type	Attribute identifier	Error code	Attribute data

表 2 で示されるフレームフォーマットの 2 つ目にある Attribute data type によって表 3 のようにデータ型を判別することができる。型判別されたデータは表 4 で示される UPnP データ型に適切に変換され、UTF-8 エンコードされた文字表現として、UPnP プロトコル内のデータとして利用される必要がある。

表 3. ZigBee の KVP コマンドフレームにおけるデータ型指定

Data type identifier b3b2b1b0	Data type	Length of data (octets)
0000	No data	0
0001	Unsigned 8-bit integer	1
0010	Signed 8-bit integer	1
0011	Unsigned 16bit integer	2
0100	Signed 16-bit integer	2
0101 – 1010	Reserved	-
1011	Semi-precision	2
1100	Absolute time	4
1101	Relative time	4
1110	Character string	Defined in first octet
1111	Octet string	Defined in first octet

表 4. UPnP メッセージにおいて定義されるデータ型指定

Data type identifier	Data type definition
ui1	Unsigned 1 Byte integer
ui2	Unsigned 2 Byte integer
ui4	Unsigned 4 Byte integer
i1	1 Byte integer
i2	2 Byte integer
i4	4 Byte integer
int	Fixed point, integer number.
r4	4 Byte float
r8	8 Byte float
number	8 Byte float
fixed.14.4	Same as r8 but no more than 14 digits to the left of the decimal point and no more than 4 to the right.
float	Floating point number.
char	Unicode string. One character long.
string	Unicode string. No limit on length.
date	Date in a subset of ISO 8601 format without time data.
dateTime	Date in ISO 8601 format with optional time but no time zone.
dateTime.tz	Date in ISO 8601 format with optional time and optional time zone.
time	Time in a subset of ISO 8601 format with no date and no time zone.
time.tz	Time in a subset of ISO 8601 format with optional time zone but no date.
boolean	0,false,or no for false;1,true,or yes for true
bin.base64	MIME-style Base64 encoded binary
bin.hex	Hexadecimal digits representing octets.
uri	Universal Resource Identifier.
uuid	Universally Unique ID. Hexadecimal digits representing octets.

以下に ZigBee データ型を基準として UPnP データ型に変換する場合、及び UPnP データ型を基準として ZigBee データ型に変換する場合それぞれについて、データフォーマットの変換仕様を示す。

3.3.4.1.1. ZigBee データフォーマットから UPnP データフォーマットへの変換

ZigBee データフォーマットを基準として UPnP データフォーマットへの変換を行う作業は、ZigBee デバイス情報を元にして、仮想 UPnP デバイスを生成する場合に発生する。従って、本データフォーマット変換が必要なユースケースは以下の通りとなる。

- ・ UPnP コントロールポイントから、ZigBee デバイス操作を行う場合(必須)
 - UPnP SOAP リクエストメッセージから、KVP コマンドフレーム(SET_WITH_ACK, GET_WITH_ACK)への変換時
 - KVP コマンドフレーム(SET_RESPONSE,GET_RESPONSE)から、UPnP SOAP レスポンスメッセージへの変換時
 - KVP コマンドフレーム(イベント)から、UPnP GENA メッセージへの変換時
 - UPnP SSDP メッセージから、ZigBee Device Profile メッセージへの変換時

それぞれの操作において、ゲートウェイは、表 5 に定めるデータフォーマット変換仕様に基づき、適切に UTF-8 文字列として格納されているデータを読み取り、対応するデータ型のバイナリ値へと変換する(または、バイナリ値として格納されているデータを読み取り、対応するデータ型の UTF-8 文字列へと変換する)必要がある。表中、必須の指定のあるデータ型は、ゲートウェイにおいて対応が必須であるデータ型であり、オプションの指定のあるデータ型は、必ずしも実装を行う必要はない。また、複数のデータ型への変換が可能となっている場合、データ型のマッピングポリシーは実装に依存する。

表 5. ZigBee を基準とするデータフォーマットの変換ルール

ZigBee Data type identifier	ZigBee Data type	Convert UPnP Data Type identifier	対応
0000	No data	-	必須
0001	Unsigned 8-bit integer	ui1	必須
0010	Signed 8-bit integer	i1	必須
0011	Unsigned 16bit integer	ui2	必須
0100	Signed 16-bit integer	i2	必須
1011	Semi-precision	float	オプション
1100	Absolute time	dateTime.tz	オプション
1101	Relative time	time	オプション
1110	Character string	string	必須
1111	Octet string	bin.hex	必須

3.3.4.1.2. UPnP データフォーマットから ZigBee データフォーマットへの変換

UPnP データフォーマットを基準として ZigBee データフォーマットへ変換を行う作業は、UPnP デバイス情報を元にして、仮想 ZigBee Application Object を生成する場合に発生する。従って、本データフォーマット変換が必要なユースケースは以下の通りとなる。

- ZigBee デバイス(コントローラ)から、UPnP デバイスの操作を行う場合(オプション)
 - KVP コマンドフレーム(SET,SET_WITH_ACK, GET_WITH_ACK)から、UPnP SOAP リクエストメッセージへの変換時
 - UPnP SOAP レスポンスメッセージから、KVP コマンドフレーム(SET_RESPONSE, GET_RESPONSE)への変換時
 - UPnP GENA メッセージから、KVP コマンドフレーム(イベント)への変換時
 - ZigBee Device Profile メッセージから、UPnP SSDP メッセージへの変換時

それぞれの操作において、ゲートウェイは、表 6 に定めるデータフォーマット変換仕様に基づき、適切にバイナリ値として格納されているデータを読み取り、対応するデータ型の UTF-8 文字列へと変換する(または、UTF-8 文字列として格納されているデータを読み取り、対応するデータ型のバイナリ値へと変換する)必要がある。ここで、注意すべきこととしては、UPnP のデータ型と ZigBee のデータ型を比較した場合、一般に UPnP のデータ型の方が ZigBee のデータ型よりも対象とするデータの範囲が広いことがある。そのため、ZigBee との通信に対応した UPnP デバイスには、使用可能な UPnP のデータ型に一定の制限が発生することに留意する必要がある。

表中、必須の指定のあるデータ型は、ゲートウェイが ZigBee デバイス(コントローラ)から、UPnP デバイスの操作を行うユースケースに対応した際、実装が必須となるデータ型であり、オプションの指定のあるデータ型は、必ずしも実装を行う必要はない。また、複数のデータ型への変換が可能となっている場合、データ型のマッピングポリシーは実装に依存する。

表 6. UPnP を基準とするデータフォーマットの変換ルール

UPnP Data type identifier	Convert ZigBee Data type identifier	Convert ZigBee Data type	UPnP Data の制限	対応
ui1	0001	Unsigned 8-bit integer	なし	必須
ui2	0011	Unsigned 16bit integer	なし	必須
ui4	0011	Unsigned 16bit integer	maximum:65535	オプション
i1	0010	Signed 8-bit integer	なし	必須
i2	0100	Signed 16-bit integer	なし	必須
i4	0100	Signed 16-bit integer	minimum:-32768 maximum:32767	オプション
int	0100	Signed 16-bit integer	minimum:-32768 maximum:32767	オプション
r4	1011	Semi-precision	minimum:-65504 maximum:65504	オプション
r8	1011	Semi-precision	minimum:-65504 maximum:65504	オプション
number	1011	Semi-precision	minimum:-65504 maximum:65504	オプション
fixed.14.4	1011	Semi-precision	minimum:-65504 maximum:65504	オプション
float	1011	Semi-precision	minimum:-65504 maximum:65504	オプション
char	1110	Character string	なし	必須
string	1110	Character string	データ最大長 255byte	必須
date	1100	Absolute time	2000/1/1 00:00:00 以降	オプション
dateTime	1100	Absolute time	2000/1/1 00:00:00 以降	オプション
dateTime.tz	1100	Absolute time	2000/1/1 00:00:00 以降	オプション
time	1101	Relative time	なし	オプション
time.tz	1101	Relative time	なし	オプション
boolean	0001	Unsigned 8-bit integer	なし	オプション
bin.base64	1111	Octet string	データ最大長 255byte	オプション
bin.hex	1111	Octet string	データ最大長 255byte	必須
uri	1110	Character string	データ最大長 255byte	オプション
uuid	1111	Octet string	データ最大長 255byte	オプション

3.3.4.2. ディスクリプションにおけるデータフォーマット変換

この節では、ZigBee デバイス情報を表すディスクリプタと、UPnP デバイスディスクリプションとを相互変換する際、そして ZigBee Cluster 情報と、UPnP Service Description 間を相互変換する際に行われる記述内容のデータフォーマット変換についての取り決めを行う。

このデータフォーマット変換の作業がゲートウェイ上で発生するのは以下である。

デバイス情報の変換

- デバイス発見時に取得した、ZigBee ディスクリプタを UPnP デバイスディスクリプションに変換する際
- デバイス発見時に取得した、UPnP デバイスディスクリプションを ZigBee ディスクリプタに変換する際

ZigBee ディスクリプタには、ノードディスクリプタ、シンプルディスクリプタ、ノードパワーディスクリプタ、コンプレックスディスクリプタ、ユーザーディスクリプタがある。ノードディスクリプタ、シンプルディスクリプタの変換は必ず行う必要があり、その他オプションとして提供される、ノードパワーディスクリプタ、コンプレックスディスクリプタ、ユーザーディスクリプタの変換はオプションとなる。

コンプレックスディスクリプタにおける言語変換は、2 オクテット ISO 639-1 language code と 1 オクテットで表現される Character set identifier フィールドを読み取り、適切に UTF-8 Char set の XML に変換する必要がある。0.1 版現在では Character set identifier は、0x00 に、ISO646 ASCII character セット(8bit 文字)が指定されており、0x01 – 0xff が予約されている。

サービス情報の変換

- ZigBee ディスクリプタから取得したクラスター情報を、UPnP サービスディスクリプションに変換する際
- UPnP デバイスディスクリプションから取得した UPnP サービスディスクリプションを ZigBee クラスターに変換する際

ZigBee クラスターID を、UPnP サービスディスクリプションに含まれるサービス ID から生成し、StateVariable 及び、アクション (SET, GET に限定) を配置する必要がある。

3.3.5. ディスクリプション変換

UPnP 機器と ZigBee 機器との通信において、End-to-End 通信を行うためには異なるネットワーク上の機器を認識する必要がある。本ゲートウェイでは、ゲートウェイ上に仮想デバイスを生成することにより、相互に機器の存在を認識する。UPnP コントロールポイントが ZigBee デバイスを認識するために ZigBee デバイス情報を元に仮想 UPnP デバイスを生成し、ZigBee デバイスが UPnP デバイスを認識するために UPnP デバイス情報を元に仮想 ZigBee デバイスを生成する。本節では、仮想デバイスを生成するためのディスクリプション変換について記述する。

3.3.5.1. UPnP におけるデバイス及びサービス情報

3.3.5.1.1. UPnP ディスクリプション

UPnP デバイスは、デバイス情報を記述するデバイスディスクリプションと、サービス情報を記述するサービスディスクリプションを持つ。UPnP コントロールポイントはディスカバリー動作の結果として、デバイスディスクリプションを取得し、デバイスディスクリプションから関連する、サービスディスクリプションデバイスの情報を取得することでデバイスが提供するサービス情報（サービスディスクリプション）を知ることができる。UPnP のコントローラである UPnP コントロールポイントには、デバイスディスクリプション、及びサービスディスクリプションは存在しないので注意すること。デバイスディスクリプションのテンプレートを図 17 に、サービスディスクリプションのテンプレートを図 18 エラー! 参照元が見つかりません。に示す。

UPnP Device Description Template

```

<?xml version="1.0"?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <URLBase>base URL for all relative URLs</URLBase>
  <device>
    <deviceType>urn:schemas-upnp-org:device:deviceType:v</deviceType>
    <friendlyName>short user-friendly title</friendlyName>
    <manufacturer>manufacturer name</manufacturer>
    <manufacturerURL>URL to manufacturer site</manufacturerURL>
    <modelDescription>long user-friendly title</modelDescription>
    <modelName>model name</modelName>
    <modelNumber>model number</modelNumber>
    <modelURL>URL to model site</modelURL>
    <serialNumber>manufacturer's serial number</serialNumber>
    <UDN>uuid:UUID</UDN>
    <UPC>Universal Product Code</UPC>
    <iconList>
      <icon>
        <mimetype>image/format</mimetype>
        <width>horizontal pixels</width>
        <height>vertical pixels</height>
        <depth>color depth</depth>
        <url>URL to icon</url>
      </icon>
      XML to declare other icons, if any, go here
    </iconList>
    <serviceList>
      <service>
        <serviceType>urn:schemas-upnp-org:service:serviceType:v</serviceType>
        <serviceId>urn:upnp-org:serviceId:serviceID</serviceId>
        <SCPURL>URL to service description</SCPURL>
        <controlURL>URL for control</controlURL>
        <eventSubURL>URL for eventing</eventSubURL>
      </service>
      Declarations for other services defined by a UPnP Forum working
      committee (if any) go here
      Declarations for other services added by UPnP vendor (if any) go here
    </serviceList>
    <deviceList>
      Description of embedded devices defined by a UPnP Forum working
      committee (if any) go here
      Description of embedded devices added by UPnP vendor (if any) go here
    </deviceList>
    <presentationURL>URL for presentation</presentationURL>
  </device>
</root>

```

図 17. UPnP Device Description Template

UPnP Service Description Template

```

<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <actionList>
    <action> z
    <name>actionName</name>
    <argumentList>
      <argument>
        <name>formalParameterName</name>
        <direction>in xor out</direction>
        <retval />
        <relatedStateVariable>stateVariableName</relatedStateVariable>
      </argument>
      Declarations for other arguments defined by UPnP Forum working
      committee (if any) go here
    </argumentList>
    </action>
    Declarations for other actions defined by UPnP Forum working
    committee (if any) go here
    Declarations for other actions added by UPnP vendor (if any) go here
  </actionList>
  <serviceStateTable>
    <stateVariable sendEvents="yes">
      <name>variableName</name>
      <dataType>variable data type</dataType>
      <defaultValue>default value</defaultValue>
      <allowedValueList>
        <allowedValue>enumerated value</allowedValue>
        Other allowed values defined by UPnP Forum working committee (if any) go here
      </allowedValueList>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>variableName</name>
      <dataType>variable data type</dataType>
      <defaultValue>default value</defaultValue>
      <allowedValueRange>
        <minimum>minimum value</minimum>
        <maximum>maximum value</maximum>
        <step>increment value</step>
      </allowedValueRange>
    </stateVariable>
    Declarations for other state variables defined by UPnP Forum working
    committee (if any) go here
    Declarations for other state variables added by UPnP vendor (if any) go here
  </serviceStateTable>
</scpd>

```

図 18. UPnP Service Description Template

3.3.5.2. ZigBee におけるデバイス及びサービス情報

3.3.5.2.1. ZigBee ディスクリプタ

ZigBee デバイス情報を示すディスクリプタには、複数の種類がある。ZigBee では、ZigBee Device Profile (ZDP) で規定されるメッセージを使用して、各々の ZigBee ディスクリプタを取得することができる。以下にデバイス情報の変換に関する ZigBee ディスクリプタを挙げる。

- ZigBee ノードディスクリプタ

ZigBee ノードディスクリプタは ZigBee デバイスのタイプと、提供する能力に関する情報が記載される。ZigBee ノードディスクリプタのテンプレートは以下の通りである。

表 7. ZigBee ノードディスクリプタ

Field name	Length (bits)
Logical type	3
Reserved	5
APS flags	3
Frequency band	5
MAC capability flags	8
Manufacture code	16
Maximum buffer size	8
Maximum transfer size	16

- ZigBee ノードパワーディスクリプタ

ZigBee ノードパワーディスクリプタはデバイスの電源状況といった情報が記載される。ZigBee ノードパワーディスクリプタのテンプレートは以下の通りである。

表 8. ZigBee ノードパワーディスクリプタ

Field name	Length (bits)
Current power mode	4
Available power sources	4
Current power source	4
Current power source level	4

- ZigBee シンプルディスクリプタ

ZigBee シンプルディスクリプタは実装される個々の ZigBee アプリケーションオブジェクトに関する情報が記載される。ZigBee シンプルディスクリプタの構成は以下である。

表 9. ZigBee シンプルディスクリプタ

Field name	Length (bits)
Endpoint	8
Application profile identifier	16
Application device identifier	16
Application device version	4
Application flags	4
Application input cluster count	8
Application input cluster list	8*i (where i is the value of the application input cluster count)
Application output cluster count	8

Application output cluster list	8*o (where o is the value of the application input cluster count)
---------------------------------	---

- ZigBee コンプレックスディスクリプタ

ZigBee コンプレックスディスクリプタはノードに含まれるデバイスに関する拡張情報が含まれる。
ZigBee コンプレックスディスクリプタは以下のようなになる。

表 10. ZigBee コンプレックスディスクリプタ

Field name	XML tag	Compressed XML tag value b3b2b1b0	Data type
Reserved	-	0000	-
Language and character set	<languageChar>	0001	0x00 - ISO646: ASCII character 0x01 - 0xff: Reserved
Manufacture name	<manufactureName>	0010	Character string
Model name	<modelName>	0011	Character string
Serial number	<serialNumber>	0100	character string
Device URL	<deviceURL>	0101	character string
Icon	<icon>	0110	Undefined
Icon URL	<iconURL>	0111	Character string
Reserved	-	1000 - 1111	-

- ZigBee ユーザディスクリプタ

ZigBee ユーザディスクリプタはノードに付与された任意の名称の情報が含まれる。ZigBee ユーザディスクリプタは以下のようなになる。

表 11. ZigBee ユーザディスクリプタ

Field name	Length
User description	16

3.3.5.2.2. ZigBee Cluster

ZigBee Cluster はプロファイル（あるいはアプリケーションオブジェクト）空間中に 8bit で識別される ClusterID を持ち、さらに 16bit の attribute_ID で識別され、4bit の dataType によって定義される型を持つ Attribute(属性値)を有する。

ZigBee Cluster Template

```
Cluster ID {
    attribute1(attribute ID,dataType...) # Cluster ID は 8bit 値で表現される
    attribute2(attribute ID,dataType...) # attribute ID は 16bit 値で表現される
    . . .
}
```

ZigBee ではクラスターとその中の属性値に対する操作によって、機能実現を達成する。操作は、KVP コマンドフレームが提供する GET と SET コマンドに限定されており、アプリケーションオブジェクトには、操作コマンドの送受信によって生じる動作の実装がなされる。

3.3.5.3. 標準 UPnP デバイスと ZigBee 標準プロファイルの対応

UPnP フォーラムにて標準化された標準 UPnP デバイス、及び ZigBee アライアンスにて標準化された標準プロファイルに対しては、ベンダーに依存することのない変換を行うために、統一した変換ルールを採用する。2008 年 3 月現在において公開済みであり、変換可能な標準 UPnP デバイスと ZigBee 標準プロファイルの対応をエラー! 参照元が見つかりません。に示す。

表 12. 標準 UPnP デバイスと ZigBee 標準プロファイルの対応

標準 UPnP デバイス	ZigBee 標準プロファイル
Binary Light	Home Automation Profile On/Off Light Device
Dimmable Light	Home Automation Profile Dimmable Light Device
HVAC System	Home Automation Profile HVAC Device
HVAC_ZoneThermostat	Home Automation Profile HVAC Device (Thermostat)

3.3.5.4. UPnP-ZigBee の識別情報の基本対応

仮想デバイス生成のために、UPnP プロトコル、ZigBee プロトコルそれぞれのデバイス、サービスを識別するための情報を相互に対応付ける必要がある。

ゲートウェイでは、起動時のネットワークの初期化に伴って、UPnP ネットワークに対しては UPnP デバイスの探索を、ZigBee ネットワークに対しては ZigBee デバイスの探索を行う。この探索の結果として、UPnP ネットワークからは UPnP デバイスディスクリプション、サービスディスクリプションを取得し、ZigBee ネットワークからは ZigBee のアドレス情報、ZigBee ディスクリプタを取得する。

仮想デバイスを生成する際の、UPnP 機器、及び ZigBee 機器が持つ情報と各々に対する仮想デバイスが持つ情報の基本対応は表 13 のようになる。

表 13. UPnP-ZigBee の識別情報の基本対応

UPnP Device	ZigBee Device
UDN (デバイスディスクリプション)	IEEE 拡張アドレス
	エンドポイント (シンプルディスクリプタ)
base URL (デバイスディスクリプション)	ネットワークアドレス
デバイスタイプ (デバイスディスクリプション)	プロファイル ID (シンプルディスクリプタ)
	デバイス ID (シンプルディスクリプタ)
サービスタイプ (デバイスディスクリプション)	クラスターID (シンプルディスクリプタ)
argument、又は state variable (サービスディスクリプション)	アトリビュート ID (プロファイルによって定義)

3.3.5.5. ZigBee のデバイス及びサービス情報を元にした仮想 UPnP デバイスの生成

ZigBee のデバイス及びサービス情報を元にした仮想 UPnP デバイスの生成方法について記述する。ゲートウェイは、取得した ZigBee の宛先情報、及びディスクリプタを元に UPnP デバイスディスクリプションへの変換を行う必要がある（デバイスディスクリプション変換）。同時にサービス情報の変換も行う必要がある。サービス情報の変換では、ZigBee プロファイル毎に規定されるプロファイル構成情報（クラスター/Attribute 情報）を元に UPnP サービスディスクリプションのアクション情報（Action/Attribute/StateVariable 情報）への変換が行われる（サービスディスクリプション変換）。

3.3.5.5.1. 仮想 UPnP デバイス生成のための識別情報の基本対応

表 13 の識別情報の基本対応に基づく変換ルールを以下に示す。

- UDN

デバイスディスクリプションに記述される。ZigBee デバイスを一意に識別するための情報として、エンドポイント毎に 1 つの UDN が割り当てられる。各 UDN は uuid のルールに従って生成される。詳細はデバイスディスクリプション変換(3.3.5.5.2 節)を参照。

- base URL

デバイスディスクリプションに記述される。ZigBee のネットワークアドレスに対して 1 つの base URL が割り当てられる。ゲートウェイ上に新規生成した仮想 UPnP ルートデバイスのアドレス（このアドレスはゲートウェイの IP アドレスとなる）+ポート（ポートは仮想デバイス毎に変わる）として表現される。

ex : 192.168.0.2:30002

192.168.0.2 はゲートウェイのアドレス

30002 はゲートウェイの UPnP ネットワーク上に生成される仮想 UPnP ルートデバイスに割り当てられたポート。割り当ては実装依存。

- デバイスタイプ

デバイスディスクリプションに記述される。ZigBee のプロファイル ID とデバイス ID の組に対して 1 つのデバイスタイプを決定する。詳細はデバイスディスクリプション変換(3.3.5.5.2 節)を参照。

- サーマスタイプ

デバイスディスクリプションに記述される。ZigBee のクラスターに対して 1 つのサービスタイプを決定する。詳細はデバイスディスクリプション変換(3.3.5.5.2 節)を参照。

- **argument**、又は **state variable**

サービスディスクリプションに記述される。ZigBee のアトリビュート 1 つに対して 1 つの **state variable** を決定する。また、GET 操作、SET 操作を可能にするため、アクション内の **argument** の記述も必要となる。アトリビュート ID は ZigBee ディスクリプタから取得できないため、プロファイル構成情報をゲートウェイで持つか、又は外部からプロファイル構成情報を取得する必要がある。詳細はサービスディスクリプション変換(3.3.5.5.3 節)を参照。

また、基本対応以外の識別情報として、以下の対応が必要となる。

- **friendly name**

デバイスディスクリプションに記述される。エンドポイント毎に 1 つの **friendly name** が生成される。その生成ルールは、デフォルトでは、「ZigBeeDevice-“IEEE Extended Address”-“EndPoint”」を採用し、コンプレックスディスクリプタが取得可能であれば、「“Model Name”-“IEEE-Extended Address”-“EndPoint”」を採用する。詳細はデバイスディスクリプション変換(3.3.5.5.2 節)を参照。

3.3.5.3 節の通り、ZigBee デバイスから取得したプロファイル ID が ZigBee 標準プロファイルに対応していれば、そのプロファイル構成情報を元に UPnP ディスクリプションへ変換することができる。詳細はプロファイル変換(3.3.5.5.4 節)を参照。

3.3.5.5.2. デバイスディスクリプション変換

ZigBee のデバイス情報である ZigBee ディスクリプタを元にした、UPnP のデバイス情報である UPnP デバイスディスクリプションへの変換ルールについて記述する。

3.3.5.5.2.1. ZigBee ディスクリプタの UPnP デバイスディスクリプションへの変換

UPnP コントロールポイントから ZigBee デバイスを操作するような場合、ZigBee デバイスは UPnP デバイスとして表現される必要がある。必須で変換される必要があるディスクリプタは、ノードディスクリプタと、シンプルディスクリプタであり、ノードパワーディスクリプタと、コンプレックスディスクリプタ、そしてユーザディスクリプタは、オプションとして入手することが可能な場合に適切な XML タグの値に変換される。この時の変換される UPnP デバイスディスクリプションは表 14 のテンプレートのようなになる。

デバイスディスクリプションの変換対応は以下に従う必要がある。

UPnP デバイスディスクリプション中の、必須情報は表中で色によって種別してあり、Required な ZigBee ディスクリプタである、Node Descriptor と Simple Descriptor は、変換に必ず必要となる。

表 14. デバイスディスクリプション変換テンプレート

赤字 : Required 緑字 : Optional 紫字 : Recommended

UPnP Device Description	参照元となる ZigBee Descriptor 情報
<root>	
<specVersion> <major>1</major> <minor>0</minor> </specVersion>	参照元はなし。必ず左のように設定する必要がある。

<p><URLBase> ゲートウェイ上に新規生成した仮想 UPnP ルートデバイスのアドレス（このアドレスはゲートウェイの IP アドレスとなる）+ポート（ポートは仮想デバイス毎に変わる）として表現される。 ex : 192.168.0.2:30002 192.168.0.2 はゲートウェイのアドレス 30002 はゲートウェイの UPnP ネットワーク上に生成される仮想 UPnP ルートデバイスに割り当てられたポート。割り当ては実装依存。 </URLBase></p>	参照元はなし。仮想 UPnP デバイスを生成する場合に、新規に生成し、パラメータの値として設定する。
<p><Device> <deviceType> urn:intap-or-jp:device:ZigBeeDevice+ "Application Profile identifier"- "Application Device identifier":1 </deviceType></p>	UPnP Forum 運営委員会によって定義される標準デバイスではないため、非標準デバイスとして deviceType を定義する。この場合の urn の形式は「urn:ベンダーICANN ドメインネーム:device:deviceType:v」である。ここではベンダーICANN ドメインネームとして intap-or-jp を用い、deviceType サフィックスには、データフォーマットの変換を行った Simple Descriptor の“Application Profile identifier”と“Application device identifier”とを組み合わせ設定する。
<p><friendlyName> ZigBeeDevice-“IEEE Extended Address”- “EndPoint” or “Model Name”-“IEEE Extended Address”-“EndPoint” </friendlyName></p>	デフォルトでは仮想 UPnP デバイスを生成する度に、ZigBeeDevice-“IEEE Extended Address”-“EndPoint”を設定する。 もし、Complex Descriptor が取得可能であれば、Complex Descriptor の “model Name”-“IEEE Extended Address”-“EndPoint”を設定する。（データフォーマット変換による文字コードセット変換後）
<p><manufacture> Manufacture code or “manufacture name” </manufacture></p>	Node Descriptor の “Manufacture code” をここに記述する。（データフォーマット変換後） もし、Complex Descriptor が取得可能であれば、Complex Descriptor の “manufacture Name” を設定する。（データフォーマット変換による文字コードセット変換後）
<p><manufacturerURL> </manufacturerURL></p>	
<p><modelDescription> <ZigBeeNodeDescInfo> <LogicalType> value of logical type </LogicalType> <Reserved> value of reserved </Reserved> <APSFlags> value of aps flags </APSFlags> <FrequencyBand></p>	ZigBee Node Descriptor 情報と ZigBee Node Power Descriptor 情報をここに組み込んで表現する。（データフォーマット変換後） ※<modelDescription>タグに記述可能な文字数制限があるため、実装においては Reserved など意味のない情報はなるべく省略することが望ましい。

<pre> value of frequency band </FrequencyBand> <MacCapabilityFlags> value of mac capability flags </MacCapabilityFlags> <ManufactureCode> value of maximum buffer size </ManufactureCode> <Maximum buffer size> value of maximum buffer size </Maximum buffer size> <MaximumTransferSize> value of maximum transfer size </MaximumTransferSize> </ZigBeeNodeDescInfo> <ZigBeeNodePowerDescInfo> <CurrentPowerMode> value of current power mode </CurrentPowerMode> <AvailablePowerSouces> value of available power souces </AvailablePowerSouces> <CurrentPowerSouce> value of current power souce </CurrentPowerSouce> <CurrentpowerSouceLevel> value of current power souce level </CurrentPowerSouceLevel> </ZigBeeNodePowerDescInfo> </modelDescription> </pre>	
<pre> <modelName> "Application Device identifier" </modelName> </pre>	Simple Descriptor に記述される Application Device identifier を設定する。(データフォーマットの変換後)
<pre> <modelNumber> "Application Device version" </modelNumber> </pre>	Simple Descriptor に記述される Application Device version を設定する。(データフォーマットの変換後)
<pre> <modelURL> "DeviceURL" </modelURL> </pre>	もし、Complex Descriptor が取得可能であれば、Complex Descriptor の Device URL を設定する。(データフォーマット変換による文字コードセット変換後)
<pre> <serialNumber> "Serial number" </serialNumber> </pre>	もし、Complex Descriptor が取得可能であれば、Complex Descriptor の Serial number を設定する。(データフォーマット変換による文字コードセット変換後)
<pre> <UDN> uuid-*****-*****-*****-*****-***** </UDN> </pre>	仮想デバイス生成の際に uuid を生成し、設定する。
<pre> <UPC> </UPC> </pre>	
<pre> <iconList> #icon 情報を含む場合 required </pre>	Complex Descriptor を取得可能で、かつ、取得

	した ComplexDescriptor に<icon>が含まれている場合のみタグが生成される。
<icon>	Complex Descriptor を取得可能かつ、<icon>フィールドが含まれている場合のみタグが<iconList>に続いて生成される。
<mimetype> value of mimetype </mimetype>	Complex Descriptor の Icon フィールドから取得できるアイコンデータの形式に応じて、設定する必要がある。(＊アイコンデータのフォーマットは ZigBee1.0 仕様書内では記載されていないことに注意)
<width> value of width </width>	Complex Descriptor の Icon フィールドから取得できるアイコンデータから、アイコンの幅情報を取得して設定する必要がある。(アイコンデータのフォーマットは ZigBee1.0 仕様書内では記載されていないことに注意)
<height> value of height </height>	Complex Descriptor の Icon フィールドから取得できるアイコンデータから、アイコンの高さ情報を取得して設定する必要がある。(アイコンデータのフォーマットは ZigBee1.0 仕様書内では記載されていないことに注意)
<depth> value of depth </depth>	Complex Descriptor の Icon フィールドから取得できるアイコンデータから、アイコンの色深度に関する情報を取得して、設定する必要がある。(アイコンデータのフォーマットは ZigBee1.0 仕様書内では記載されていないことに注意)
<url> value of Icon URL </url>	Complex Descriptor の Icon URL フィールドの値をここに記入する。(データフォーマット変換による文字コードセット変換後)
</icon>	
</iconList>	
<serviceList>	本ディスクリプションで記述されるプロファイルをサポートする場合、Simple Descriptor の Application input cluster count と Application output cluster list の総数分の <service></service>を生成する必要がある。 プロファイルがサポート外である場合、cluster の総数に係わらず一つの <service></service>を生成する。
<service>	本ディスクリプションで記述されるプロファイルをサポートする場合、一つの input cluster もしくは output cluster に対し一つ、生成される。 仮想 UPnP デバイスに変換される ZigBee アプリケーションオブジェクトは複数の Cluster を持ち、一つ一つが <service></service>に投影される必要があることに注意する。 また、プロファイルがサポート外である場合、cluster の総数に係わらず一つのみ、生成される。
<serviceType> サポートプロファイルの場合	UPnP Forum 運営委員会によって定義される標準サービスではないため、非標準サービスと

<p>urn:intap-or-jp:service: ZigBeeService-“InputCluster”:v (or OutputCluster)</p> <p>サポート外プロファイルの場合 urn:intap-or-jp:service: ZigBeeLookUpService:1 </serviceType></p>	<p>して serviceType を定義する。この場合の urn の形式は「urn:ベンダーICANN ドメインネーム:service:serviceType:v」である。ここではベンダー ICANN ドメインネームとして intap-or-jp を用いる。</p> <p>serviceType サフィックスは、本ディスクリプションで記述されるプロファイルをサポートする場合、input cluster であるなら、 ZigBeeService-InputCluster output cluster であるなら、 ZigBeeService-OutputCluster を設定する。</p> <p>また、プロファイルをサポートしていない場合は ZigBeeLookUpService を設定する。</p>
<p><serviceId> サポートプロファイルの場合 urn:intap-or-jp:serviceId: “Application Profile ID”-“Cluster ID”</p> <p>サポート外プロファイルの場合 urn:intap-or-jp:serviceId:LookUpService </serviceId></p>	<p>UPnP Forum 運営委員会によって定義される標準サービスではないため、非標準サービスとして serviceID を定義する。この場合の urn の形式は「urn:ベンダーICANN ドメインネーム:serviceId:serviceID」である。ここではベンダーICANN ドメインネームとして intap-or-jp を用いる。</p> <p>serviceID サフィックスは本ディスクリプションで記述されるプロファイルをサポートする場合、Simple Descriptor から取得できる Application Profile ID と、Application input cluster list の個々の Cluster から、取得できる Cluster ID を設定する。(データフォーマット変換後)</p> <p>また、プロファイルをサポートしていない場合は LookUpService を設定する。</p>
<p><SCPDURL> サポートプロファイルの場合 <u>/scpd/in(or out)-“Cluster_ID”/service.xml</u></p> <p>サポート外プロファイルの場合 <u>/scpd/lookup/service.xml</u> </SCPDURL></p>	<p>本ディスクリプションで記述されるプロファイルをサポートする場合、 <u>/scpd/“Direction”-“Cluster_ID”/service.xml</u> を設定する。Direction は対象 Cluster が input クラスターの場合は”in”、output クラスターの場合は”out”を用いる。”Cluster_ID”は対象となる ZigBee アプリケーションオブジェクトが持つ一つの Cluster の ID (データフォーマット変換後の ID) であり、<SCPDURL>、<controlURL>、<eventSubURL>で同じ ID である必要がある。</p> <p>また、プロファイルをサポートしていない場合は、/scpd/lookup/を設定する。 “/scpd/”及び”service.xml”は実装に依存し、任意に変えることができる。</p>
<p><controlURL> サポートプロファイルの場合 <u>/control/in(or out)-“Cluster_ID”/</u></p>	<p>本ディスクリプションで記述されるプロファイルをサポートする場合、 <u>/control/“Direction”-“Cluster_ID”</u>を設定する。</p>

<p>サポート外プロファイルの場合</p> <pre> /control/lookup/ </controlURL> </pre>	<p>Direction は対象 Cluster が input クラスターの場合は”in”、output クラスターの場合は”out”を用いる。”Cluster_ID”は対象となる ZigBee アプリケーションオブジェクトが持つ一つの Cluster の ID (データフォーマット変換後の ID) であり、<SCPDURL>、<controlURL>、<eventSubURL>で同じ ID である必要がある。また、プロファイルをサポートしていない場合は、/control/lookup/を設定する。</p> <p>“/control/”は実装に依存し、任意に変えることができる。</p>
<pre> <eventSubURL> サポートプロファイルの場合 event/in(or out)-"Cluster_ID"/ サポート外プロファイルの場合 "" </eventSubURL> </pre>	<p>本ディスクリプションで記述されるプロファイルをサポートする場合、/event/”Direction”-”Cluster_ID”/を設定する。Direction は対象 Cluster が input クラスターの場合は”in”、output クラスターの場合は”out”を用いる。”Cluster_ID”は対象となる ZigBee アプリケーションオブジェクトが持つ一つの Cluster の ID (データフォーマット変換後の ID) であり、<SCPDURL>、<controlURL>、<eventSubURL>で同じ ID である必要がある。また、プロファイルをサポートしていない場合は、空文字を設定する。</p> <p>“/event/”は実装に依存し、任意に変えることができる。</p>
<pre> </service> </pre>	
<pre> </serviceList> </pre>	
<pre> <deviceList> </deviceList> </pre>	<p>本仕様では、ZigBee Application Object に対し、一つのルートデバイスを割り当てるため、この値は利用されない。</p>
<pre> <presentationURL> if Information url http://IP_Address_of_Gateway/UDN"/ </presentationURL> </pre>	<p>もし、何らかの形で、ゲートウェイが、この ZigBee 端末に関する情報(html)を取得している場合は、http://ゲートウェイのアドレス/UDN”で示される url で公開する。</p>
<pre> </device> </pre>	
<pre> </root> </pre>	

3.3.5.5.3. サービスディスクリプション変換

最初に変換対象となる UPnP サービスディスクリプションと、ZigBee クラスターのフォーマットを説明し、次に変換ルールについて記述する。

UPnP サービスディスクリプションと ZigBee クラスターの変換は ZigBee デバイスの持つプロファイルごとに定義される。ゲートウェイが対象となるプロファイルのサービス変換をサポートするかについてはオプションであり、サポート時の構成（即ち、プロファイル構成情報を内部実装するか、3.3.5.5.4 節に規定される、プロファイル構成情報の外部取得を行うか、あるいはその双方）については実装に依存する。サポート外のプロファイルである場合、lookup サービスを記述したサービスディスクリプションへの変換が行われる。ゲートウェイが対象となるプロファイルをサポートする場合、取得したシンプルディスクリプタから得られる、クラスター ID 情報をサービスネームとしてサービスディスクリプションの生成を行う必要がある。

3.3.5.5.3.1. lookup サービスのサービスディスクリプションへの変換

ゲートウェイが対象となるプロファイルのサービス変換をサポートしていない場合(即ち、プロファイル構成情報を保持していない場合)、制御オペレーションの存在しない、lookup サービスのサービスディスクリプションを生成する必要がある。このとき生成されるサービスディスクリプションはクラスターに関わらず固定であり、図 19 に表されるものとなる。

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <serviceStateTable>
    <stateVariable sendEvents="no">
      <name>defaultVariable</name>
      <dataType>uri</dataType>
    </stateVariable>
  </serviceStateTable>
</scpd>
```

図 19. lookup サービスのために記述されるサービスディスクリプション

3.3.5.5.3.2. ZigBee クラスターの UPnP サービスディスクリプションへの変換

UPnP コントロールポイントから ZigBee デバイスを操作するような場合、ZigBee が提供するクラスターとその中に含まれている属性値、及び、属性に対する操作は UPnP デバイスが提供するサービスとして認識される必要がある。ZigBee が提供する属性に対する操作は、SET 及び GET であり、UPnP コントロールポイントが、ゲートウェイ上の仮想的な UPnP デバイスに送った操作メッセージは、ゲートウェイ内部でプロトコル変換され、KVP コマンドフレームとなって、ターゲットにまで運ばれる。このとき変換に使用するデータは ZigBee プロファイル構成情報及び ZigBee SimpleDescriptor から取得した Cluster 情報であり、変換は表 15 のテーブル対応に基づき行われなければならない。

注意すべき点として、ZigBee Device Object が提供する基本サービスと関わりのある Cluster 操作は Service Description 内に変換されないことである。これら基本サービスは UPnP では Device が提供する基本サービスとして、隠蔽される。

UPnP サービスディスクリプション中の、必須情報は表中で色により種別してあり、Required な ZigBee ディスクリプタである、Node Descriptor と Simple Descriptor は、ディスクリプション変換には必ず必要となる。

表 15. サービスディスクリプション変換テンプレート

赤字 : Required 緑字 : Optional 紫字 : Recommended

変換後の UPnP Service Description	変換元となる ZigBee Cluster 情報
<pre><scpd xmlns="urn:schemas-upnp-org:service-1-0"> <specVersion> <major>1</major> <minor>0</minor> </specVersion></pre>	

<actionList>	
<action>	
<name> GET </name>	ここでは GET アクションに関する記載がなされる。
<argumentList>	GET, もしくは SET の操作対象となるクラスターに含まれる個々の属性は argument として表現され、argumentList としてまとめられる。
<argument>	個々の属性は argument として表現される。
<name> ChangedAttributeID </name>	GET, SET の操作の際の引数は、変化する AttributeID とする。
<direction> in xor out </direction>	もし、この attribute が属しているクラスターがインプットクラスターであれば、in に設定する。もし、アウトプットクラスターであれば、out に設定される必要がある。
<retval />	使用しない。
<relatedStateVariable> Attribute_ID </relatedStateVariable>	操作対象 Cluster の中の ChangedAttributeID と関連付けられる Attribute_ID を記入する。ここで記入される Attribute_ID は stateVariable の name に記述される ID と同じでなければならない。
</argument>	
</argumentList>	
</action>	
<action>	
<name> SET </name>	ここでは SET に関する情報が記載される
<argumentList>	GET, SET の操作対象となるクラスターに含まれる個々の属性は argument として表現され、argumentList としてまとめられる。
<argument>	個々の属性は argument として表現される。
<name> ChangedAttributeID </name>	GET, SET の操作の際の引数は、変化する AttributeID とする。
<direction> in xor out </direction>	もし、この attribute が属しているクラスターがインプットクラスターであれば、in に設定する。もし、アウトプットクラスターであれば、out に設定される必要がある。
<retval />	
<relatedStateVariable> Attribute_ID </relatedStateVariable>	操作対象 Cluster の中の ChangedAttributeID と関連付けられる Attribute_ID を記入する。ここで記入される Attribute_ID は stateVariable の name に記述される ID と同じでなければならない。
</argument>	
</argumentList>	
</action>	
</actionList>	
<serviceStateTable>	操作可能な attribute の数だけ

	<StateVariable>の設定を行う。
<stateVariable sendEvents="yes">	対象の attribute がイベントを生成する場合、sendEvents は"yes"を設定する。そうでなければ"no"を設定する。
<name> Attribute_ID </name>	操作対象 Cluster 中の Attribute_ID を記入する。(データフォーマット変換後)
<dataType> ZigBee data Type </dataType>	操作対象 Attribute のデータ型を記入する。(データフォーマット変換後)
<defaultValue> defaultValue </defaultValue>	もし、操作対象 Attribute が規定値を持つのであれば、その値を記入する。(データフォーマット変換後)
<allowedValueList>	もし、操作対象 Attribute のデータタイプが"Character string"であり、legal state value が存在する場合には、<allowedValueList></allowedValueList>を生成する。legal state value が複数存在する場合には、<allowedValue></allowedValue>をその数分用意する。
<allowedValue>value</allowedValue>	legal state value を記述する。
</allowedValueList>	
<allowedValueRange>	もし、操作対象 Attribute のデータタイプが"Unsigned 8-bit integer" "Signed 8-bit integer" "Unsigned 16-bit integer" "Unsigned 16-bit integer" "Semi-precision"であり、legal state value の境界値が存在する場合には、<allowedValueRange></allowedValueRange>を生成する。
<minimum>value</minimum>	下位の境界を記述する。
<maximum>value</maximum>	上位の境界を記述する。
<step>increment value</step>	増加オペレーションのサイズを記入する。
</allowedValueRange>	
</stateVariable>	
</serviceStateTable>	
</scpd>	

3.3.5.5.4. 仮想 UPnP デバイス生成におけるプロフィール変換

3.3.5.5.2 節で提供されるディスクリプション変換を行う場合、一般に変換後の XML 記述された UPnP デバイスディスクリプションは、UPnP コントロールポイントにとって操作不可能な記述として、存在を知ることができるのみとなり、不明デバイスとして扱われる。UPnP コントロールポイントから、変換された仮想 UPnP デバイスを経由して、ZigBee デバイスを操作可能とするためには、(1)内部で予め変換対象となるプロフィール構成情報を有し ZigBee クラスターの UPnP サービスディスクリプションへの変換を行う、(2)翻訳済みディスクリプションを参照するプロフィール変換機能を持つ、のどちらかの手段が必要となる。(2)が必要である理由はデバイスディスクリプションやサービスディスクリプションを変換する際に、純粋な変換機能を用いただけでは足りない情報を補足できることがあるためであり、それを解決するためには、UPnP フォーラムおよび、ZigBee アライアンス内にて標準化された標準プロフィールに対して、ベンダーに依存することのない、ベンダー間共通の、翻訳されたディスクリプションの提供が行われる機能が必要となる。(標準プロフィール変換)そして、ベンダーが独自に提供するアプリケーションに関しては、個々のベンダーが翻訳したディスクリプションの提供が行われる。(ベンダープロフィール変換)特定の ZigBee プロファイル ID が提供するディスクリプタを完全に翻訳した UPnP デscriプションを取得するプロフィール変換の機能は実装に依存する。

3.3.5.5.4.1. 標準プロフィール変換

標準プロフィール変換は、ZigBee アライアンスと UPnP フォーラムが各々標準化済みのデバイス情報やサービス情報に関して、事前にデータベースを構築し、そこにアクセスすることによって、特定のプロフィールに対応する翻訳済みの完全なディスクリプションを取得するための変換機能である。

ZigBee 標準プロフィールが規定するディスクリプタ類、及び、クラスター、属性をあらかじめ翻訳した仮想 UPnP デバイスを示すデバイスディスクリプション及び、仮想的なサービスを示すサービスディスクリプションに関しては、あらかじめ完全な状態で用意することができ、ゲートウェイでは特定のプロフィールであることを読み取ると、翻訳作業をせずに、あらかじめ準備されたディスクリプションの使用を試みることができる。

例として、発見した ZigBee デバイスのプロフィールが”Home Automation Profile”であり、デバイスが”On/Off Light”である場合を挙げる。この場合、標準 UPnP デバイス”Binary Light”への変換を行う。変換が必要な情報は表 16 に示す通りである。(2008 年 3 月現在で公開されている ZigBee 標準プロフィールは ZigBee2006 以上に対応することに注意すること)

表 16. ZigBee”On/Off Light”デバイスにおけるディスクリプション変換

UPnP ディスクリプションの項目	変換方法
Base URL	(ゲートウェイの IP アドレス) : (ポート番号)
UDN	uuid のルールに従って生成される
デバイスタイプ	urn:schemas-upnp-org:device:BinaryLight:1.0 (”Binary Light”のデバイスタイプ)
friendly name	ZigBeeDevice- (IEEE 拡張アドレス) - (エンドポイント)
サービスタイプ	urn:schemas-upnp-org:service:SwitchPower:1 (”Switch Power”のサービスタイプ)
サービス ID	urn:upnp-org:serviceId:SwitchPower:1 (”Switch Power”のサービス ID)

以下に、”On/Off Light”デバイスから”Binary Light”のデバイスディスクリプションへの変換のテンプレートを示す。

```
<?xml version="1.0"?>
```

```

<root xmlns="urn:schemas-upnp-org:device-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <URLBase> (ゲートウェイの IP アドレス) : (ポート番号) </URLBase> ←変換
  <device>
    <deviceType>urn:schemas-upnp-org:device:BinaryLight:1.0</deviceType> ←変換
    <friendlyName> ZigBeeDevice-(IEEE Extended Address)-(End Point) </friendlyName> ←変換
    <manufacturer>manufacturer name</manufacturer>
    <manufacturerURL>URL to manufacturer site</manufacturerURL>
    <modelDescription>long user-friendly title</modelDescription>
    <modelName>model name</modelName>
    <modelNumber>model number</modelNumber>
    <modelURL>URL to model site</modelURL>
    <serialNumber>manufacturer's serial number</serialNumber>
    <UDN>uuid:UUID</UDN>
    <UPC>Universal Product Code</UPC>
    <iconList>
      <icon>
        <mimetype>image/format</mimetype>
        <width>horizontal pixels</width>
        <height>vertical pixels</height>
        <depth>color depth</depth>
        <url>URL to icon</url>
      </icon>
      XML to declare other icons, if any, go here
    </iconList>
    <serviceList>
      <service>
        <serviceType>urn:schemas-upnp-org:service:SwitchPower:1</serviceType> ←変換
        <serviceId>urn:upnp-org:serviceId:SwitchPower:1</serviceId> ←変換
        <SCPDURL>URL to service description</SCPDURL>
        <controlURL>URL for control</controlURL>
        <eventSubURL>URL for eventing</eventSubURL>
      </service>
      Declarations for other services added by UPnP vendor (if any) go here
    </serviceList>
    <deviceList>
      Description of embedded devices added by UPnP vendor (if any) go here
    </deviceList>
    <presentationURL>URL for presentation</presentationURL>
  </device>
</root>

```

また、以下に、ZigBee”On/Off Light”デバイスからの”Binary Light”のサービスディスクリプションへの変換を示す。”Binary Light”では、サービスとして”Switch Power”を用いられるが、ZigBee Cluster Library (ZCL) の”On/Off”クラスターがそのサービスに対応する。

```

<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">

```

```

<specVersion>
  <major>1</major>
  <minor>0</minor>
</specVersion>
<actionList>
  <action>
    <name>SetTarget</name>
    <argumentList>
      <argument>
        <name>newTargetValue</name>
        <relatedStateVariable>Target</relatedStateVariable>
        <direction>in</direction>
      </argument>
    </argumentList>
  </action>
  <action>
    <name>GetTarget</name>
    <argumentList>
      <argument>
        <name>RetTargetValue</name>
        <relatedStateVariable>Target</relatedStateVariable>
        <direction>out</direction>
      </argument>
    </argumentList>
  </action>
  <action>
    <name>GetStatus</name>
    <argumentList>
      <argument>
        <name>ResultStatus</name>
        <relatedStateVariable>Status</relatedStateVariable>
        <direction>out</direction>
      </argument>
    </argumentList>
  </action>
</actionList>
<serviceStateTable>
  <stateVariable sendEvents="no">
    <name>Target</name>
    <dataType>boolean</dataType>
    <defaultValue>0</defaultValue>
  </stateVariable>
  <stateVariable sendEvents="yes">
    <name>Status</name>
    <dataType>boolean</dataType>
    <defaultValue>0</defaultValue>
  </stateVariable>
</serviceStateTable>
</scpd>

```

このサービスディスクリプション変換において、各アクションは表 17 に示すように実行する。

表 17. ZigBee”On/Off Light”デバイスに対するアクションの実行

アクション	実行対象 ZigBee クラスター	実行対象 ZigBee アトリビュート	操作
SetTarget	“On/Off”クラスター (0x0006)	“OnOff”アトリビュート (0x0000)	“OnOff”アトリビュートに Set コマンドを送信する。
GetTarget	“On/Off”クラスター (0x0006)	“OnOff”アトリビュート (0x0000)	“OnOff”アトリビュートに Get コマンドを送信する。
GetStatus	“On/Off”クラスター (0x0006)	“OnOff”アトリビュート (0x0000)	“OnOff”アトリビュートに Get コマンドを送信する。

ZigBee”On/Off”クラスターの”OnOff”アトリビュートからのイベント通知は、state variable ”status”からのイベント通知として扱う。また、”Binary Light”デバイスの仮想 UPnP デバイスを生成する際には、state variable の Status、Target は同じ変数として実装すること。

3.3.5.6. UPnP のデバイス及びサービス情報を元にした仮想 ZigBee デバイスの生成

ZigBee デバイスから UPnP デバイスを操作する場合、UPnP のデバイス及びサービス情報を元にして仮想 ZigBee デバイスを生成する必要がある。

ゲートウェイは、取得した UPnP デバイスのデバイスディスクリプション、サービスディスクリプションを元に、ZigBee ディスクリプタへの変換（ディスクリプタ変換）、プロファイル構成情報（クラスター/Attribute 情報）への変換（クラスター変換）を行い、仮想的な ZigBee デバイスとして ZigBee Application Object を生成する。

以下では、仮想 ZigBee デバイスの生成について詳述する。

3.3.5.6.1. 仮想 ZigBee デバイス生成のための識別情報の基本変換ルール

表 13 の識別情報の基本対応に基づく変換ルールを以下に示す。

- ・ IEEE 拡張アドレス
どの仮想 ZigBee デバイスに対しても、ゲートウェイとして動作している ZigBee コーディネータの IEEE 拡張アドレスを使用する。
- ・ ネットワークアドレス
どの仮想 ZigBee デバイスに対しても、ゲートウェイとして動作している ZigBee コーディネータのネットワークアドレス (0x0000) を使用する。
- ・ エンドポイント
UDN に対して 1 つのエンドポイントを割り当てる。割り当て方法は、以下のルールに従う。
 - ・ 1 から 240 の範囲の値を使用すること。
 - ・ 他の UPnP デバイスに対応するエンドポイントと重複しないこと。
 - ・ UPnP デバイスが離脱した際、その UPnP デバイスに対応するエンドポイントを記録しておき、再度同じ UPnP デバイスが接続した際に、記録していたエンドポイントを割り当てること。
 - ・ 1 から 240 までの値全てを割り当てた場合、さらに接続した UPnP デバイスに対してはエンドポイントの割り当ては行わない。
- ・ プロファイル ID、デバイス ID

シンプルディスクリプタに記述される。UPnP デバイスタイプに対して 1 つのプロファイル ID と 1 つのデバイス ID を決定する。詳細はディスクリプタ変換(3.3.5.6.2 節)を参照。

- ・ クラスターID

シンプルディスクリプタに記述される。サービスタイプとサービス ID の組に対して 1 つのクラスターID を決定する。詳細はクラスター変換(3.3.5.6.3 節)を参照。

- ・ アトリビュート ID

UPnP のアクションの argument、又は state variable に 1 つのアトリビュート ID を割り当てる。アトリビュートのデータタイプについては state variable の dataType タグ情報を元に、3.3.3.1 節に従い ZigBee のデータタイプに適切に変換される(データフォーマット変換)。アトリビュート ID の変換についての詳細は、クラスター変換(3.3.5.6.3 節)を参照。

3.3.5.6.2. ディスクリプタ変換

各仮想 ZigBee デバイスは、ZigBee ディスクリプタを保持しなければならないため、UPnP ディスクリプションを元に ZigBee ディスクリプタに変換する必要がある。

3.3.5.6.2.1. ディスクリプタ変換ルール

必須の ZigBee ディスクリプタとして、ノードディスクリプタ、ノードパワーディスクリプタ、シンプルディスクリプタがあり、オプションの ZigBee ディスクリプタとして、コンプレックスディスクリプタ、ユーザディスクリプタがある。ただし、UPnP ディスクリプションから変換しなければならない ZigBee ディスクリプタはシンプルディスクリプタのみであり、その他の ZigBee ディスクリプタはゲートウェイの ZigBee コーディネータが持つ ZigBee ディスクリプタを使用する。シンプルディスクリプタの変換について表 18 に示す。

表 18 シンプルディスクリプタへの変換

シンプルディスクリプタの項目	変換方法 (参照元となる UPnP ディスクリプション情報)
エンドポイント	前述の変換情報の基本対応 (3.3.5.6.1 参照) によって割り当てられた値。
アプリケーションプロファイル ID	デバイスディスクリプションのデバイスタイプを参照。 デバイスタイプが標準 UPnP デバイスである場合、対応する ZigBee 標準プロファイルがあれば、そのプロファイル ID に変換する。
アプリケーションデバイス ID	デバイスディスクリプションのデバイスタイプを参照。 デバイスタイプが標準 UPnP デバイスである場合、対応する ZigBee 標準プロファイルがあれば、そのデバイス ID に変換する。
アプリケーションデバイスバージョン	0x0000
アプリケーションフラグ	0x0000
インプットクラスターカウント	クラスター変換 (3.3.5.6.3 節参照) によって変換されたインプットクラスターの数。
インプットクラスターリスト	クラスター変換 (3.3.5.6.3 節参照) によって変換されたインプットクラスターのクラスターID のリスト。
アウトプットクラスターカウント	0
アウトプットクラスターリスト	NULL

3.3.5.6.3. クラスタ変換

ZigBee デバイスから UPnP デバイス进行操作する場合、UPnP のサービス情報を元に ZigBee のサービス情報を生成する必要がある。UPnP デバイスが標準 UPnP デバイスであり、対応する ZigBee 標準プロファイルが存在する場合は、標準プロファイル変換を適用するため、その変換ルールに従ってクラスタ ID、アトリビュート ID を決定する。標準プロファイル変換を適用できない場合は、UPnP ディスクリプションを元に ZigBee クラスタ、アトリビュートに変換する。その変換方法について記述する。

3.3.5.6.3.1. クラスタ ID の決定

UPnP ディスクリプションを元に ZigBee クラスタ ID を決定する方法について記述する。仮想 ZigBee デバイスにおける ZigBee クラスタ ID は、UPnP デバイスディスクリプションに記述される <serviceType> と <serviceId> の組に対して 1 つ決定される。UPnP デバイスディスクリプションに記述される <serviceList> に含まれる <service> 毎に、0 から 255 の値の範囲内で、仮想 ZigBee デバイス内で重複しないように決定する。ここで決定されたクラスタは、全てインプットクラスタとなる。

3.3.5.6.3.2. アトリビュート ID の決定

UPnP ディスクリプションを元に ZigBee アトリビュート ID を決定する方法について記述する。仮想 ZigBee デバイスにおける ZigBee アトリビュート ID は、以下のルールによって割り当てられる。

まず、UPnP サービスディスクリプションに記述されるアクションの情報を元にした割り当てルールについて記述する。このルールでは、アクション内の各 <argument> に対して 1 つのアトリビュート ID を割り当てる。ただし、アクションの条件により、アトリビュートへの操作は以下のように制限される。

- 1 つの <action> 内に、“in” の <direction> を持つ <argument> が存在しない場合、その <action> 内の <argument> に対応するアトリビュートへの制御については、KVP コマンドフレームの Get with Acknowledgement のみ許可する。
- 1 つの <action> 内に、“in” の <direction> を持つ <argument> が 1 つ以上存在する場合、その <action> 内の <argument> に対応するアトリビュートへの制御については、“in” の <direction> を持つ <argument> に対応するアトリビュートに対しては KVP コマンドフレームの Set、又は Set with Acknowledgement を許可し、その数と同数の KVP コマンドフレームをゲートウェイが受け取らなければ、アクションは実行されない。“out” の <direction> を持つ <argument> に対応するアトリビュートに対しては、他の ZigBee デバイスからの KVP コマンドフレームを許可しない。
- “out” の <direction> を持つ <argument> に対応するアトリビュートについては、そのアトリビュートからの KVP コマンドフレームの Event によるイベント通知が可能でなければならない。

次に、UPnP サービスディスクリプションに記述される state variable の情報を元にした割り当てルールについて記述する。このルールでは、<serviceStateTable> の <stateVariable> において <sendEvent=”yes”> と記述されている state variable に対して 1 つの ZigBee アトリビュート ID を割り当てる。

3.3.5.6.4. 仮想 ZigBee デバイス生成における標準プロファイル変換

デバイスタイプが3.3.5.3節に記載された ZigBee 標準プロファイルに対応している標準 UPnP デバイスである場合は、統一ルールに従った変換を行う。

以下では、標準 UPnP デバイス”Binary Light”から、ZigBee 標準プロファイル”Home Automation Profile”のデバイス”On/Off Light”への変換を例に、標準プロファイル変換について記述する。

(2008年3月現在で公開されている ZigBee 標準プロファイルは ZigBee2006 以上に対応することに注意すること)

表 19 にディスクリプタ変換を示す。

表 19 標準 UPnP デバイス”Binary Light”におけるディスクリプタ変換

シンプルディスクリプタの項目	変換方法 (参照元となる UPnP ディスクリプション情報)
エンドポイント	前述の識別情報の基本変換ルール (3.3.5.6.1 参照) によって割り当てられた値。
アプリケーションプロファイル ID	0x0104 (Home Automation Profile の Profile ID)
アプリケーションデバイス ID	0x0100 (On/Off Light の Device ID)
アプリケーションデバイスバージョン	0x0000
アプリケーションフラグ	0x0000
インプットクラスターカウント	1
インプットクラスターリスト	{0x0006}
アウトプットクラスターカウント	0
アウトプットクラスターリスト	NULL

クラスター変換について以下に記述する。

表 20 に示すように ZigBee Cluster Library (ZCL) に基づいてクラスターID が決定される。

表 20 標準 UPnP デバイス”Binary Light”に対するクラスターID の決定

ZigBee クラスターID	備考
0x0006 (ZCL の”On/Off”クラスターのクラスターID)	標準 UPnP デバイス”Binary Light”で使用されるサービス”Switch Power”に対応する ZigBee クラスターは ZCL の”On/Off”クラスター。

また、決定したクラスターにおけるアトリビュート ID を表 21 に示す。

表 21 標準 UPnP デバイス”Binary Light”に対するアトリビュート ID の決定

ZigBee アトリビュート ID	備考
0x0000 (”On/Off”クラスターのアトリビュート”OnOff”のアトリビュート ID)	<ul style="list-style-type: none"> このアトリビュートに対して Get 操作が行われた場合、サービス”Switch Power”の”GetStatus”アクションを実行する このアトリビュートに対して Set 操作が行われた場合、サービス”Switch Power”の”SetTarget”アクションを実行する。 state variable ”Status”からのイベント通知を、このアトリビュートの Event として送信する。

3.3.6. プロトコル変換

3.3.5 節で規定されるディスクリプション変換によって、仮想的な UPnP デバイスやコントロールポイントと、仮想的な ZigBee アプリケーションオブジェクトが生成され、通信可能な状態になることが望まれる。本節では、通信可能な状態となったデバイス間で発生するメッセージが、ゲートウェイでどう変換されるのかを規定している。規定される必要のあるプロトコルは、ディスカバリーに関するプロトコルと、制御に関するプロトコル、及びイベント送信に関するプロトコルであり、各々以下のような観点からそれぞれ変換される必要が生じる。

まず、UPnP コントロールポイントからのディスカバリーメッセージをゲートウェイが受け取ると（つまりこの時点でディスカバリーメッセージを受け取るためのデバイス機能がゲートウェイには必要となることを意味する）、ゲートウェイは、特定のアプリケーションプロファイルが実装されている ZigBee アプリケーションオブジェクトのディスカバリーメッセージに変換し、ZigBee ネットワーク上の適切な宛先に向けて転送する。レスポンスとして取得できた ZigBee ディスクリプタ情報を適切な UPnP デバイスディスクリプションのフォーマットに変換する。この際にもしアプリケーションプロファイル ID が標準のもので既に翻訳済みのディスクリプションがある場合にはそれが参照される。また、もしもアプリケーションプロファイル ID がベンダーオリエンテッドなものであり、かつ、翻訳済みのディスクリプションがベンダーから提供されている場合はそれが参照され、仮想的なデバイスとサービスの初期化が行われ、生成された仮想デバイスは自身をネットワークに対してアドバタイズを行う。

UPnP のコントロールメッセージを受け取ると、ゲートウェイは ZigBee の制御コマンドフレームに変換し、仮想 ZigBee Controller から ZigBee ネットワーク上の適切な宛先に向けて転送する必要がある。制御コマンドのレスポンスは適切な UPnP コントロールレスポンスメッセージに変換し、仮想 UPnP デバイスから発信元の UPnP コントロールポイントに返す必要がある。

UPnP コントロールポイントからのイベント購読メッセージを仮想 UPnP デバイスが受け取ると、ゲートウェイは仮想 UPnP デバイスごとの購読管理テーブルにその情報を記録する。ZigBee デバイスからイベントコマンドフレームが発生し仮想 ZigBee Controller において受信した場合、ゲートウェイはそのイベントコマンドフレームを UPnP のイベントメッセージに変換し、送信元 ZigBee デバイスに対応する仮想 UPnP デバイスへと転送する。仮想 UPnP デバイスは購読管理テーブルを参照し、UPnP ネットワーク上の適切な宛先となるコントロールポイントに向けてイベントメッセージを転送する必要がある。

3.3.6.1. アドレッシング

プロトコル変換の際には、同時に宛先アドレスの変換作業が必須となる。宛先アドレスの変換は、3.3.5.4 節で決められた情報を保持する仮想デバイスの識別情報マッピングテーブルのアドレス対応テーブルに基づき変換され、利用される。アドレッシング変換ルールは以下の通りである。

- ・仮想 UPnP デバイスに対するメッセージの場合
宛先アドレスを、仮想 UPnP デバイスに対応する実 ZigBee デバイスのネットワークアドレス及びエンドポイントに変換する。送信元アドレスを、送信元の UPnP CP に割り付けた ZigBee デバイスを表すアドレスへと変換する。
- ・仮想 ZigBee デバイスに対するメッセージの場合
宛先アドレスを、仮想 ZigBee デバイスに対応する実 UPnP デバイスのアドレスに変換する。送信元アドレスを、送信元の ZigBee デバイスに割り付けた UPnP CP を表すアドレスへと変換する。
- ・仮想 UPnP デバイスに対応する実 ZigBee デバイスからのメッセージの場合

宛先アドレスを、対応する UPnP CP を表すアドレス（メッセージによってはマルチキャストアドレス）に変換する。送信元アドレスを、送信元 ZigBee デバイスに対応する仮想 UPnP デバイスを表すアドレスへと変換する。

- ・仮想 ZigBee デバイスに対応する実 UPnP デバイスからのメッセージの場合
宛先アドレスを、対応する ZigBee デバイスを表すアドレス（メッセージによってはブロードキャストアドレス）に変換する。送信元アドレスを、送信元 UPnP デバイスに対応する仮想 ZigBee デバイスを表すアドレスへと変換する。

3.3.6.2. Discovery プロトコル変換

UPnP と ZigBee 双方で、デバイス情報を取得し、サービス情報を取得するまでの一連のプロトコル機能を、ゲートウェイ上で提供する必要がある。また、デバイスがネットワークから離脱する際、ネットワークに対し行う告知のプロトコル機能もゲートウェイ上で提供する必要がある。

3.3.6.2.1. UPnP における発見の動作

UPnP におけるデバイスとサービスの発見（ディスカバリー）動作には SSDP が利用されている。SSDP による UPnP サービスディスクリプションの取得を行うまでの手順は以下となる。

1. SSDP M-SEARCH 要求がコントロールポイントから、デバイスに向けて発信される。
2. SSDP レスポンスメッセージによる告知情報がコントロールポイントに向けて、返信される。
3. コントロールポイントは SSDP レスポンスメッセージ中の LOCATION 情報を見て、デバイスディスクリプションの取得を試みる。
4. コントロールポイントは、取得したデバイスディスクリプション内に含まれている SCPDURL 情報から、サービスディスクリプションの取得を試みる。

3.3.6.2.2. ZigBee における発見の動作

ZigBee における発見の動作は、ZigBee デバイスオブジェクトが提供するクライアントサービスとサーバーサービス間のディスクリプタ要求によって実現される。すべての要求とレスポンスは ZigBee Device Profile メッセージフレームを用いて実現される。

1. ZigBee デバイスオブジェクトクライアントサービスから、サーバーサービスに向けて、ディスクリプタ照合要求が発信される。（ブロードキャスト）
このとき、特定のアプリケーションプロファイル ID を指定することができる。
2. ネットワークアドレスとエンドポイントを含んだ応答が、サーバーサービスからクライアントサービスに対して返される。プロファイル ID を指定していた場合は、指定したプロファイル ID を持つデバイスからのみ、応答が返される。
3. ZigBee デバイスオブジェクトクライアントサービスは、発見したネットワークアドレスとエンドポイントを持つサーバーサービスに対し、ディスクリプタ(シンプルディスクリプタ、ノードディスクリプタ)要求を発信する。
4. ディスクリプタを含んだ応答が、サーバーサービスからクライアントサービスに対して返される。

5. 要求元の ZigBee デバイスは、取得したディスクリプタに含まれるリスト構造から、対照のアプリケーションオブジェクトが所持するクラスターID 情報を取得する。

3.3.6.2.3. UPnP における離脱の動作

UPnP におけるデバイスとサービスの離脱動作には SSDP が利用されている。離脱デバイスはネットワークに対し、自身が利用できなくなることを示す Notify メッセージ (ssdp:byebye) をマルチキャストで送信する。従って、仮想 UPnP デバイスに対応する実 ZigBee 端末がネットワークから取り除かれた (ネットワーク離脱コマンドを受信するか、定期的な生存確認において発見されなかった場合) 際、仮想 UPnP デバイスも同様のメッセージを送信した後、削除される必要がある。

3.3.6.2.4. ZigBee における離脱の動作

ZigBee におけるデバイスとサービスの離脱動作の通知は、ZDO のネットワークマネージャ、バインディングマネージャを用いてコーディネータに対しネットワークの離脱とバインディングの解消を要求する他は、任意の端末に対する通知は特に規定されていない。従って、仮想 ZigBee デバイスに対応する実 UPnP デバイスから ssdp:byebye を受信した際、もしくは新たに更新されることなく ssdp の有効期限が過ぎた場合、仮想 ZigBee 端末は該当端末のバインディングを解消した後、削除される必要がある。

3.3.6.2.5. ゲートウェイによる発見プロトコルの変換シーケンス

ゲートウェイは UPnP が実現する発見プロトコルと、ZigBee における発見プロトコルが透過的に相互運用可能とするための機能を提供する。ゲートウェイにおける変換は次の図の通りである。

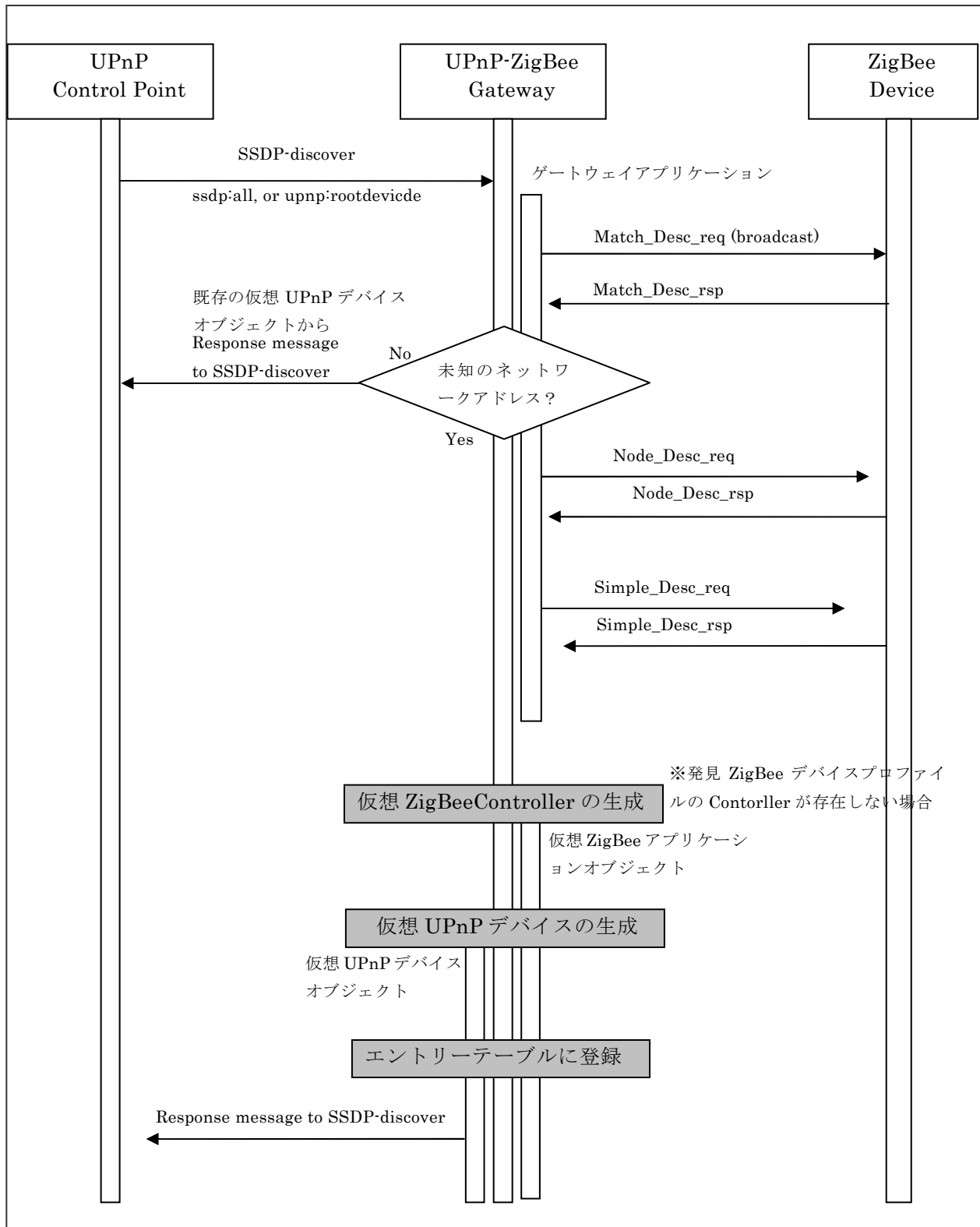


図 20. UPnP CP がネットワークに参加したときのディスカバリーシーケンス
(SSDP : M-SEARCH の検索対象が全てのデバイス、又は、ルートデバイスである場合)

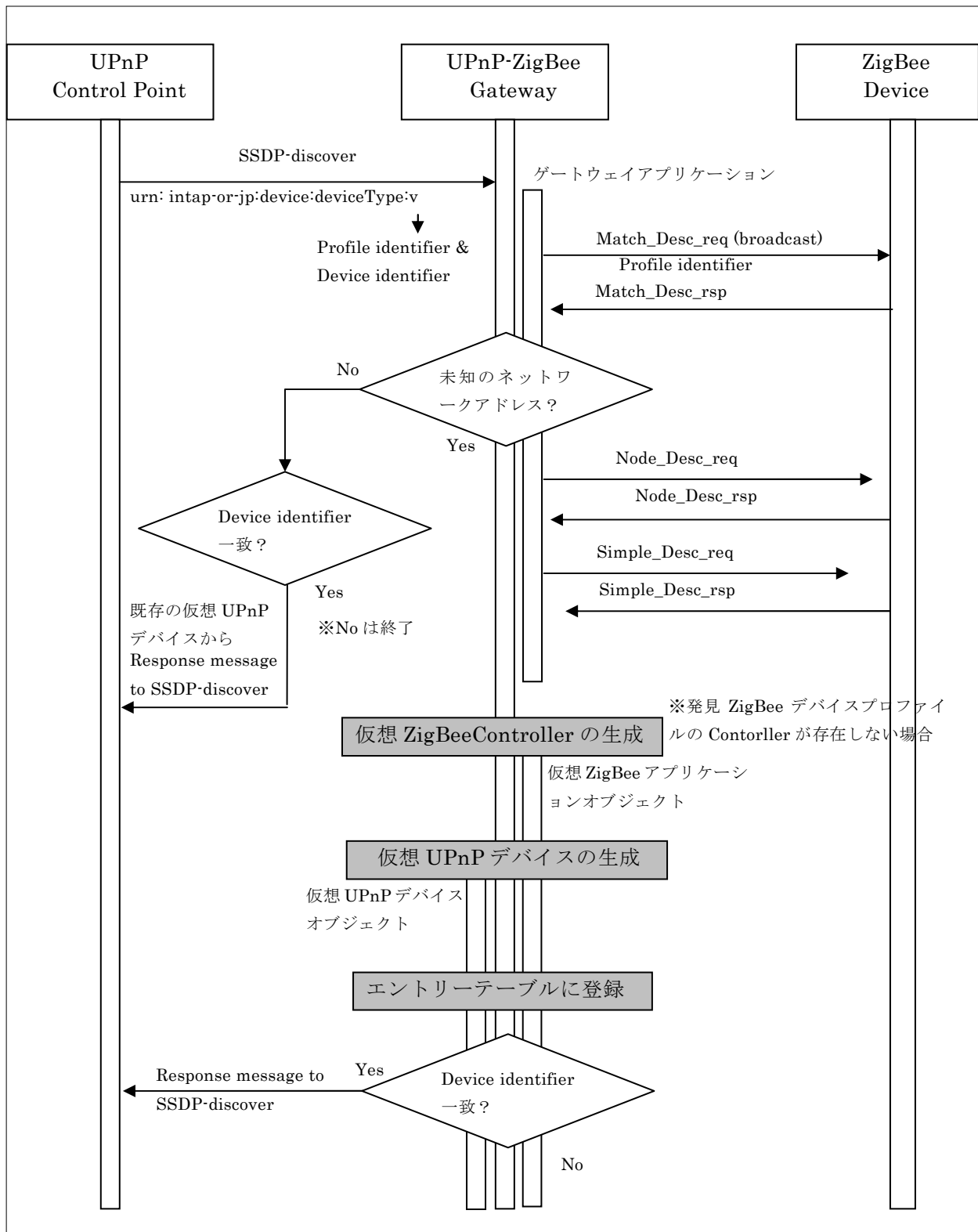


図 21. UPnP CP がネットワークに参加したときのディスカバリーシーケンス
(SSDP : M-SEARCH の検索対象がデバイスタイプである場合)

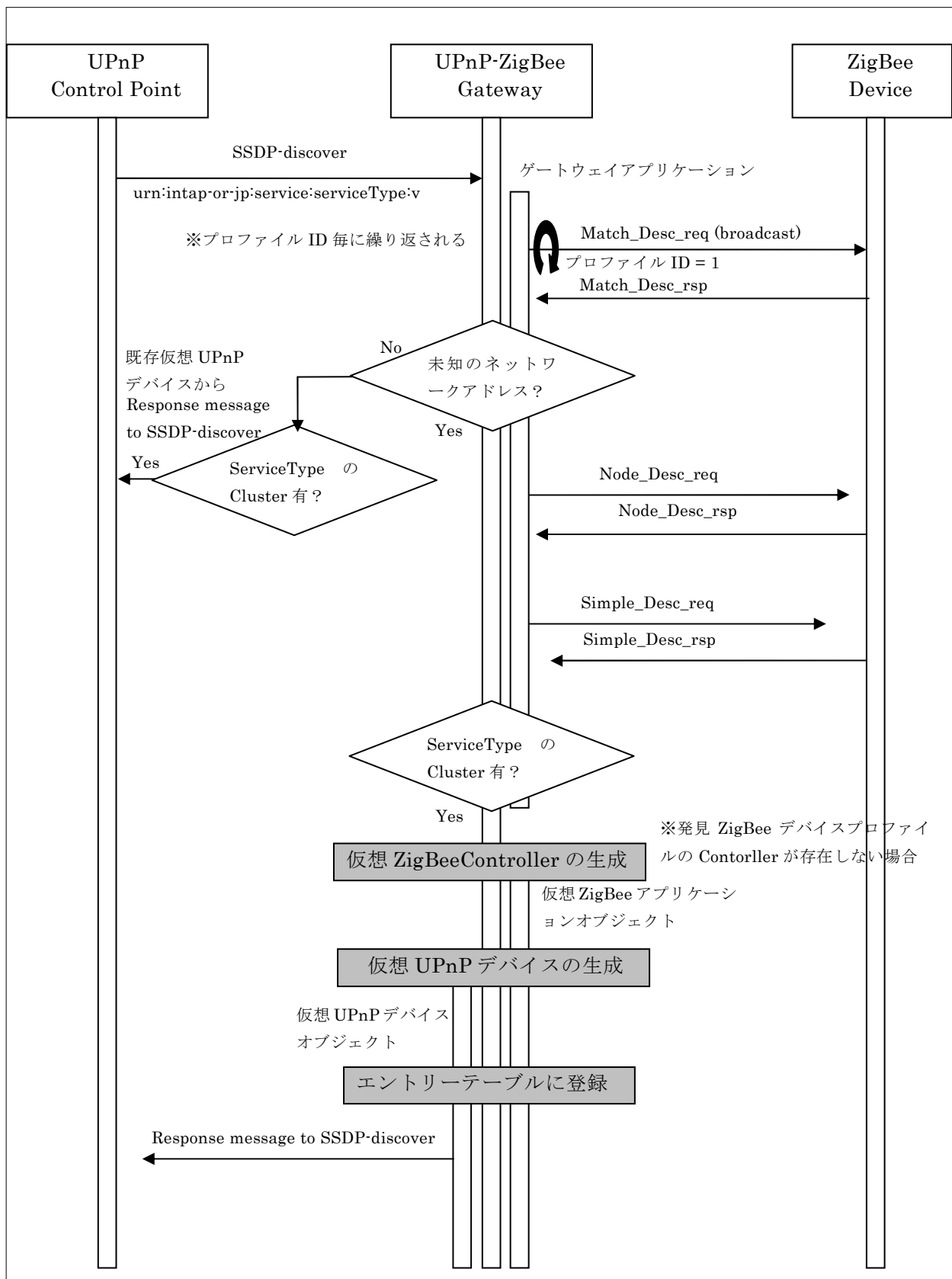


図 22. UPnP CP がネットワークに参加し時のディスカバリーシーケンス (SSDP : M-SEARCH の検索対象がサービスタイプである場合)

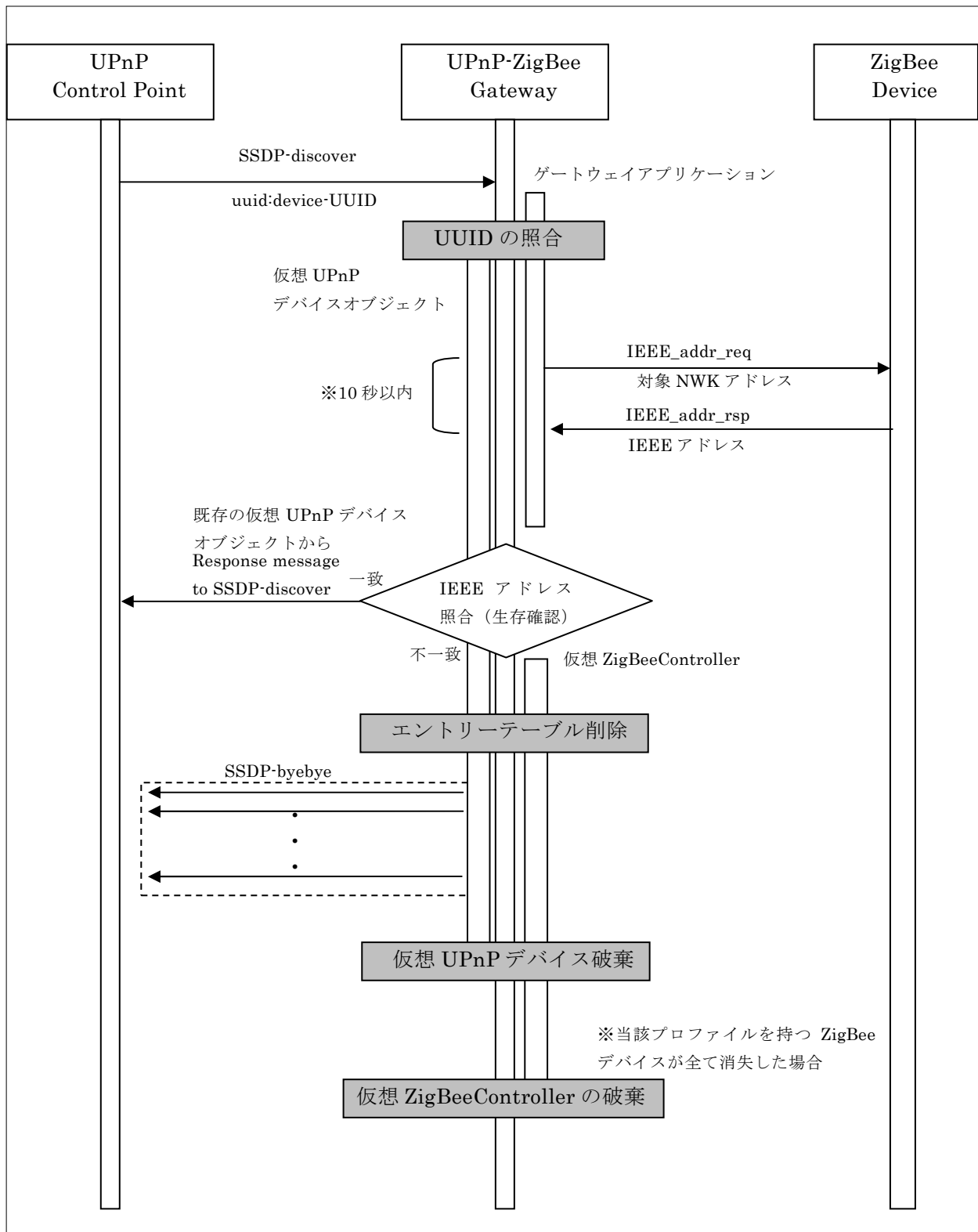


図 23. UPnP CP がネットワークに参加した時のディスカバリーシーケンス
(SSDP : M-SEARCH の検索対象が特定のデバイスである場合)

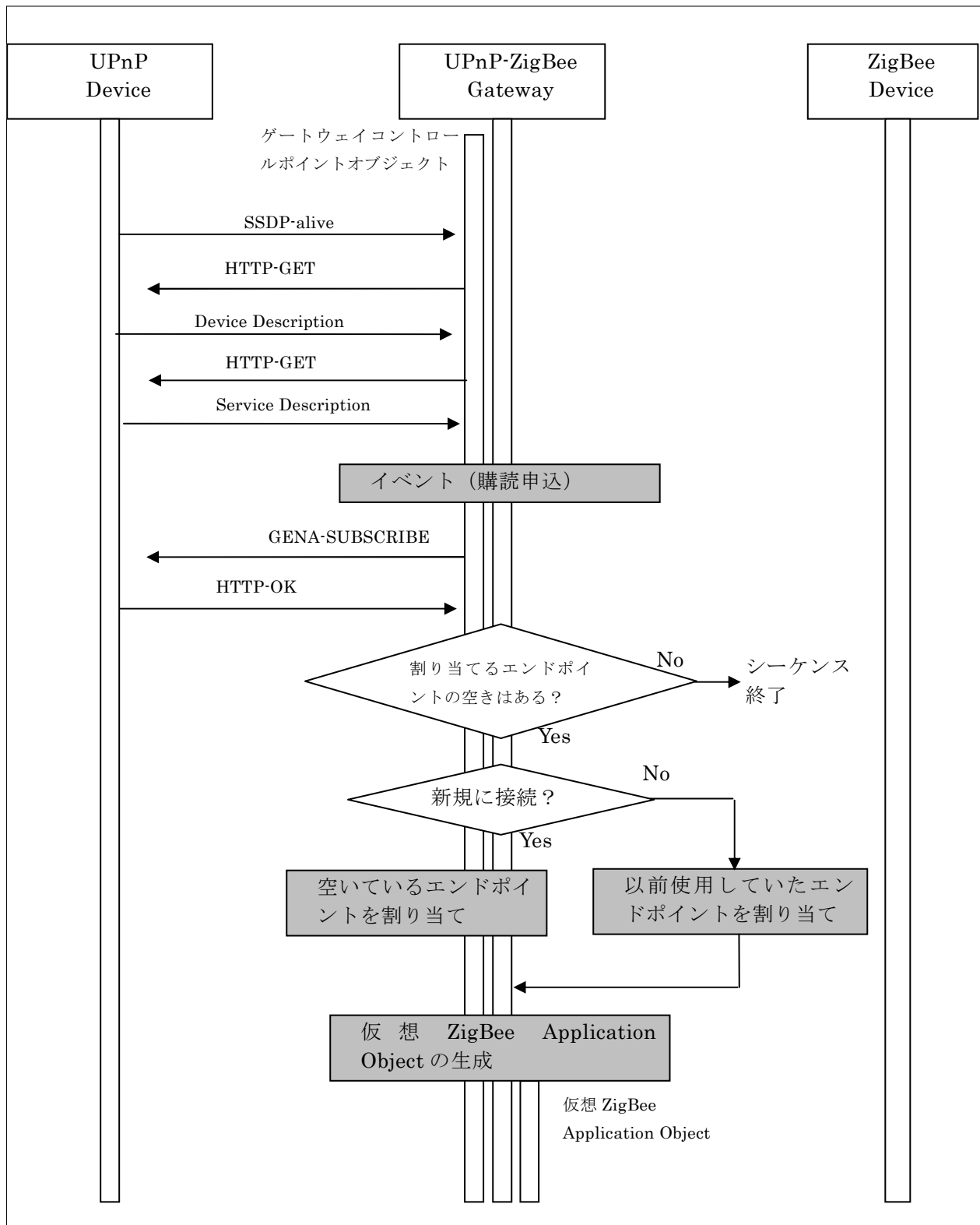


図 24. UPnP デバイスがネットワークに参加した時のディスカバリーシーケンス

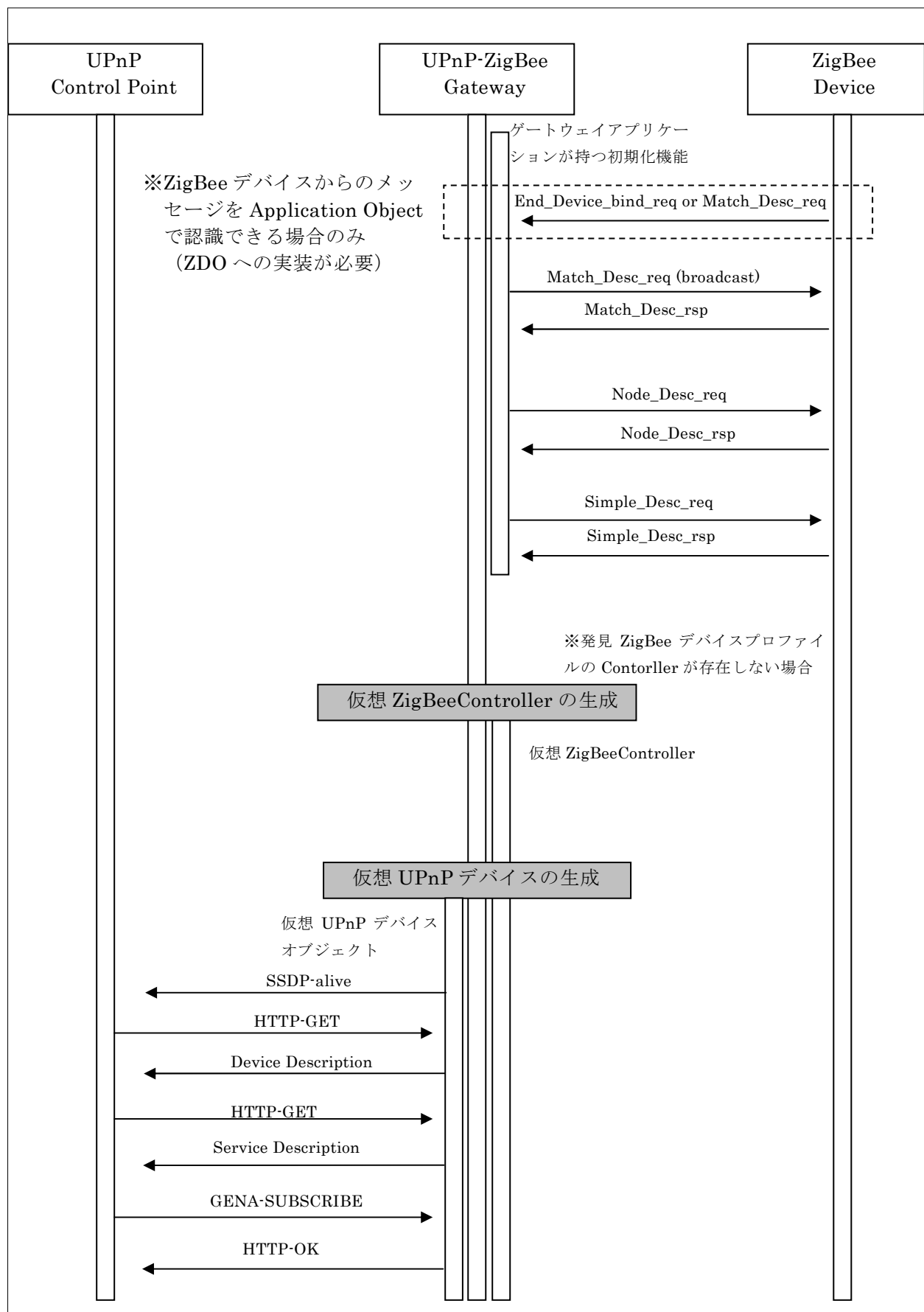


図 25. ZigBee デバイスがネットワークに参加した時のディスカバリーシーケンス (ZigBee デバイスが仮想的な UPnP Device となる場合)

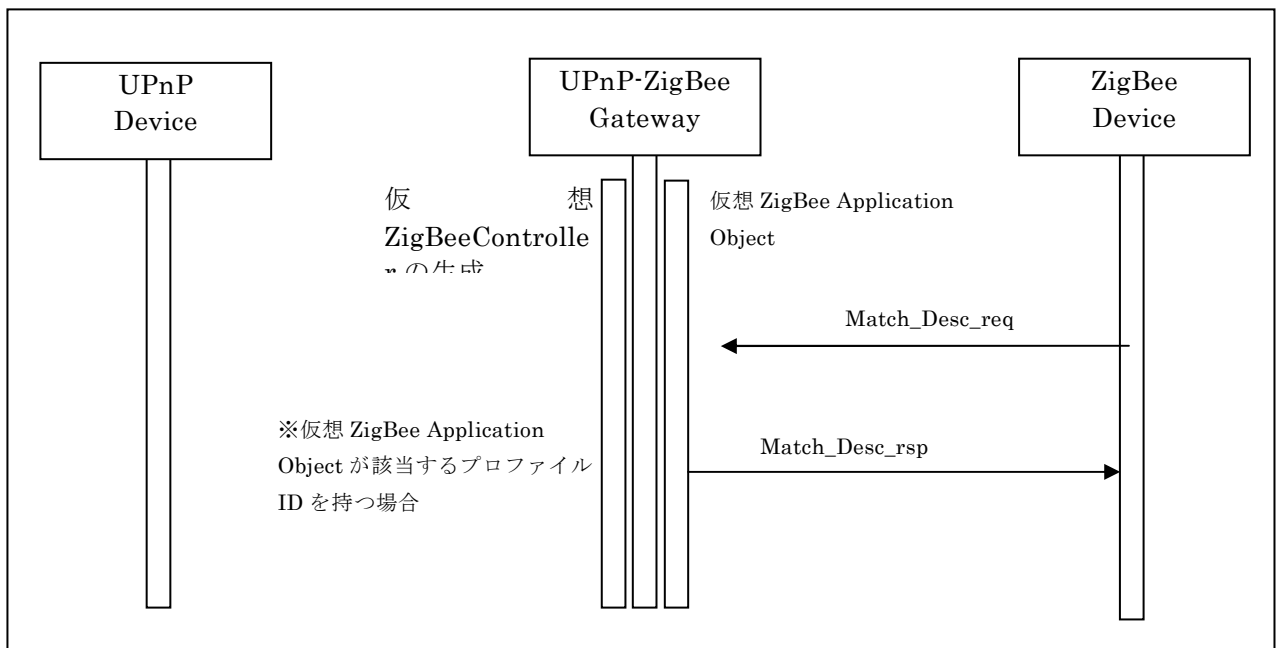


図 26. ZigBee デバイスがネットワークに参加した時のディスカバリーシーケンス
(ZigBee デバイスが UPnP デバイスの発見を要求した場合)

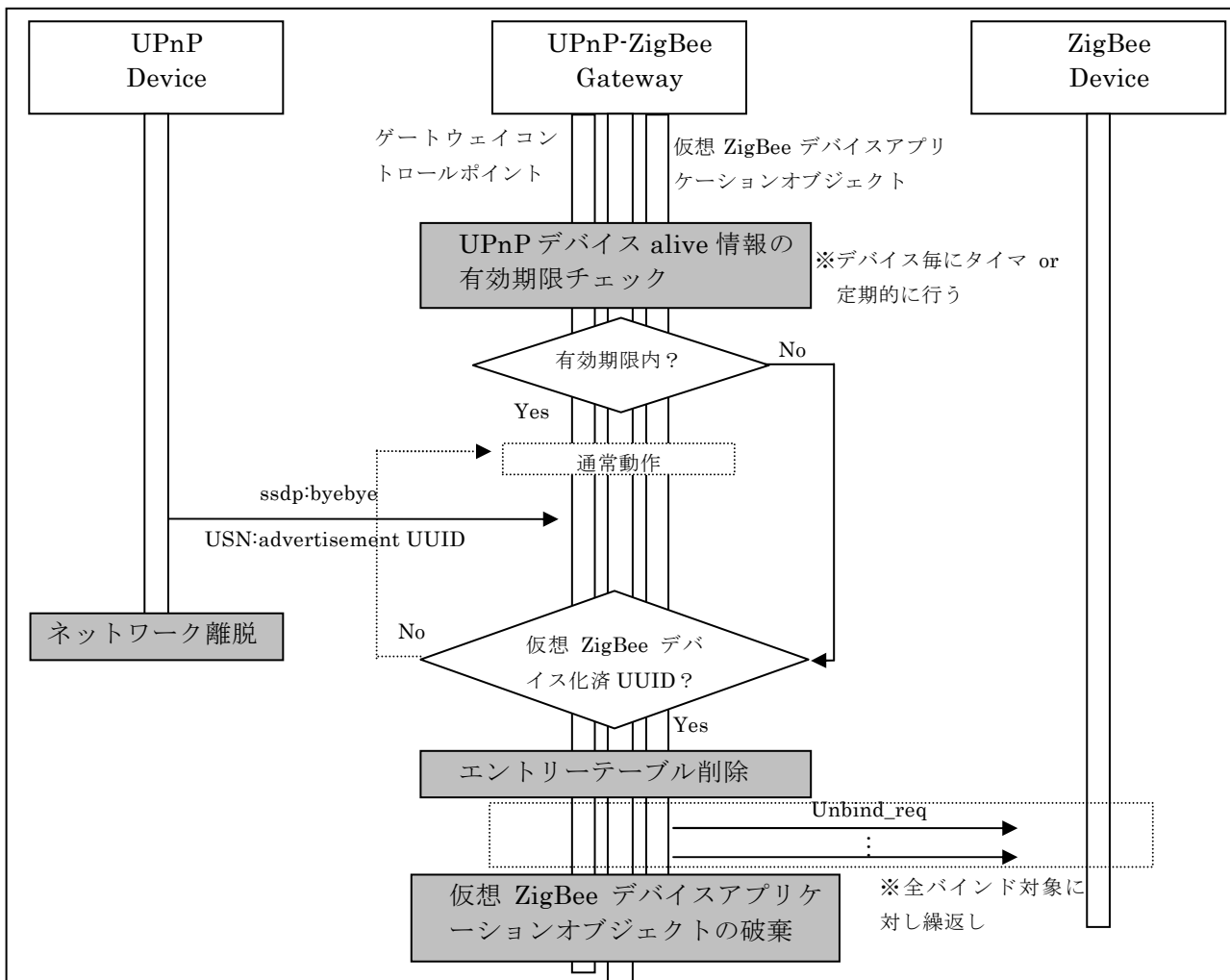


図 27 UPnP デバイスがネットワークから離脱した際のシーケンス
(ssdp による通知があった or ssdp の有効期限が切れた場合)

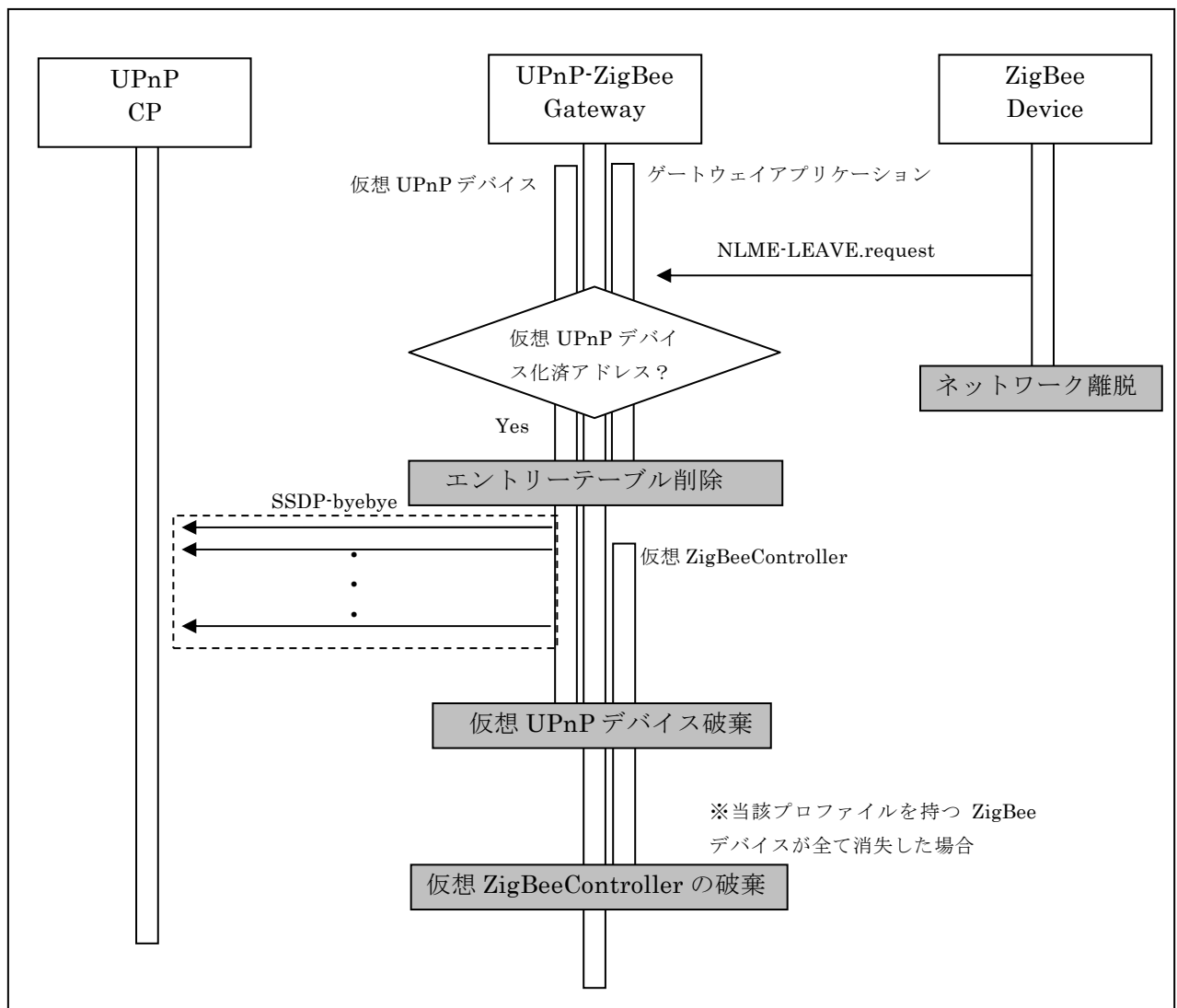


図 28 ZigBee デバイスがネットワークから離脱した際のシーケンス
(ネットワーク離脱の要求があった場合)

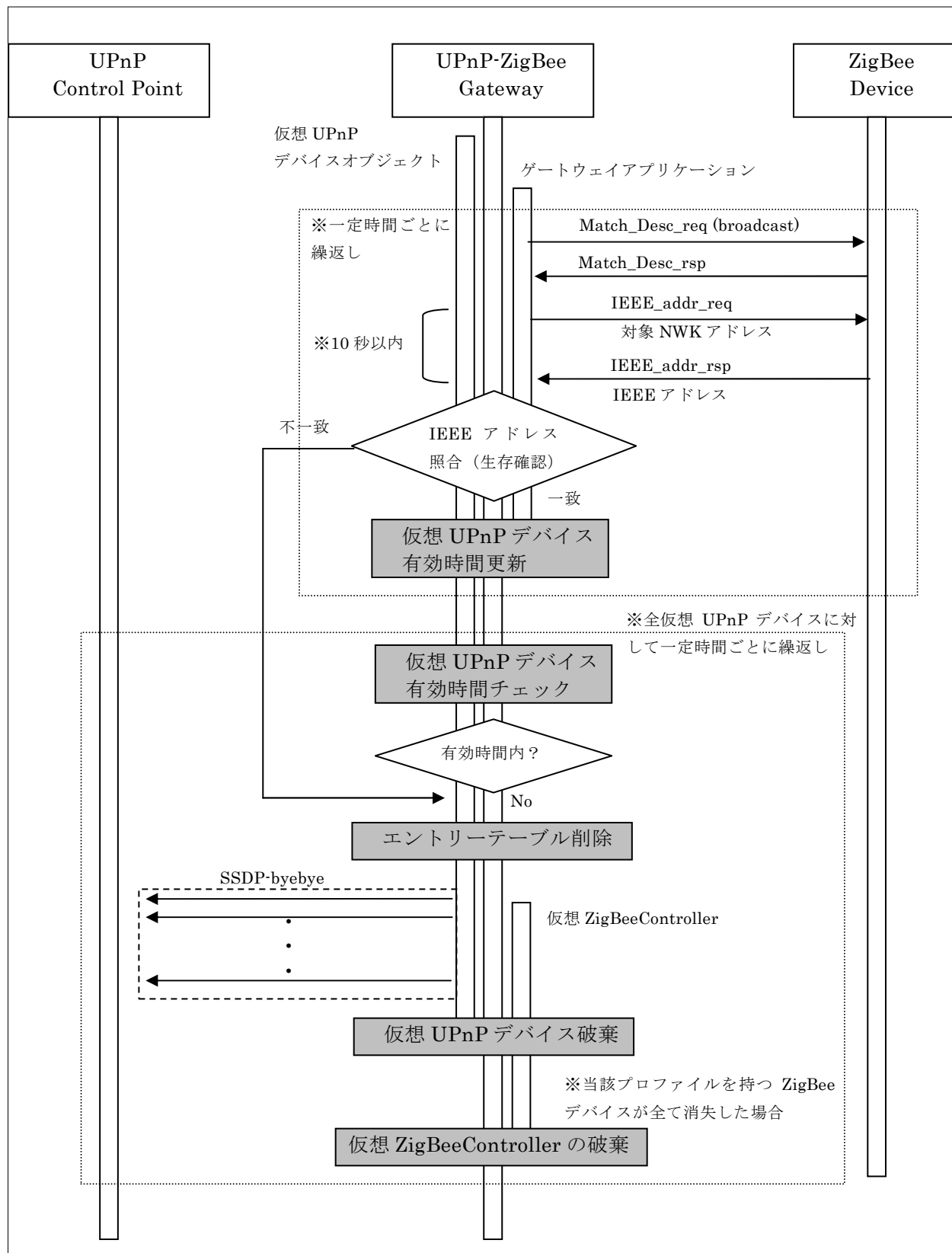


図 29 ZigBee デバイス生存確認シーケンス

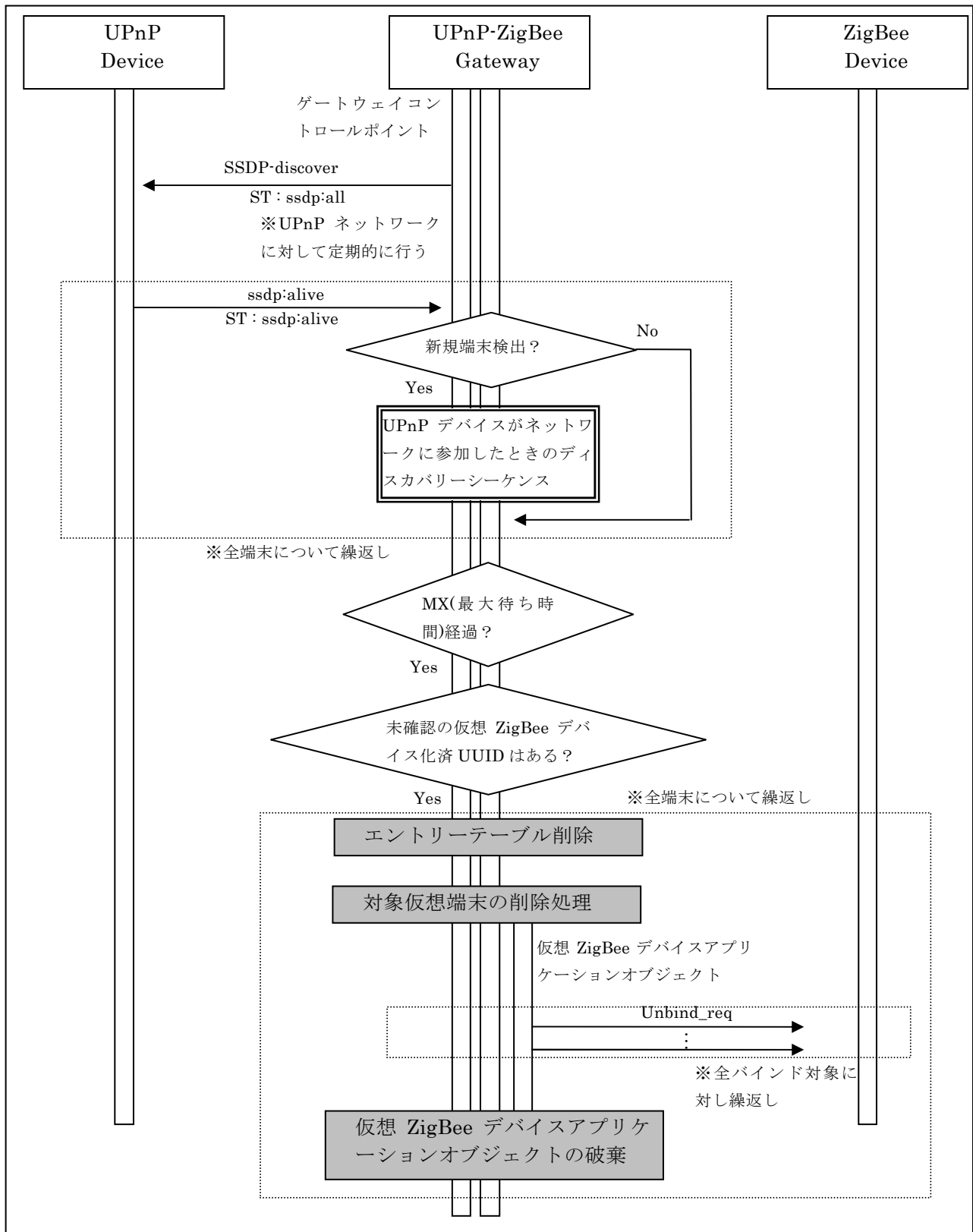


図 30 UPnP デバイスの生存確認シーケンス

3.3.6.3. 制御プロトコル変換

UPnP と ZigBee 双方で、デバイスが提供するサービス制御を行うための機能を、ゲートウェイ上で提供する。変換の対象となる各々の制御プロトコルの動作手順は以下である。

3.3.6.3.1. UPnP における制御プロトコル

UPnP におけるサービスの制御（コントロール）動作には SOAP が利用されている。SOAP による UPnP サービスの制御を行う手順は以下のようになっている。

1. SOAP リクエストメッセージがコントロールポイントから、デバイスが持つサービスに対して送信される。
2. SOAP リクエストメッセージによる制御要求が実行された後、実行結果を含んだレスポンスメッセージがコントロールポイントに向けて返信される。(リクエスト送信からレスポンス受信までの伝送時間は 30 秒以内でなくてはならない)

3.3.6.3.2. ZigBee における制御プロトコル

ZigBee におけるサービス制御はクラスターを構成する属性への操作という形で実現される。属性への操作は KVP コマンドフレームによって転送される。

1. KVP コマンドフレームが、コントロール元となる ZigBee デバイスから、対象となる ZigBee デバイスに送信される。
2. KVP コマンドフレームを受け取った ZigBee アプリケーションオブジェクトはフレームに含まれる Command Type identifier を読み取り、Attribute identifier に指定される属性に対する操作を行い、レスポンスとなる Command Type identifier がセットされた KVP コマンドフレームを返す。

3.3.6.3.3. ゲートウェイにおける制御プロトコルの変換シーケンス

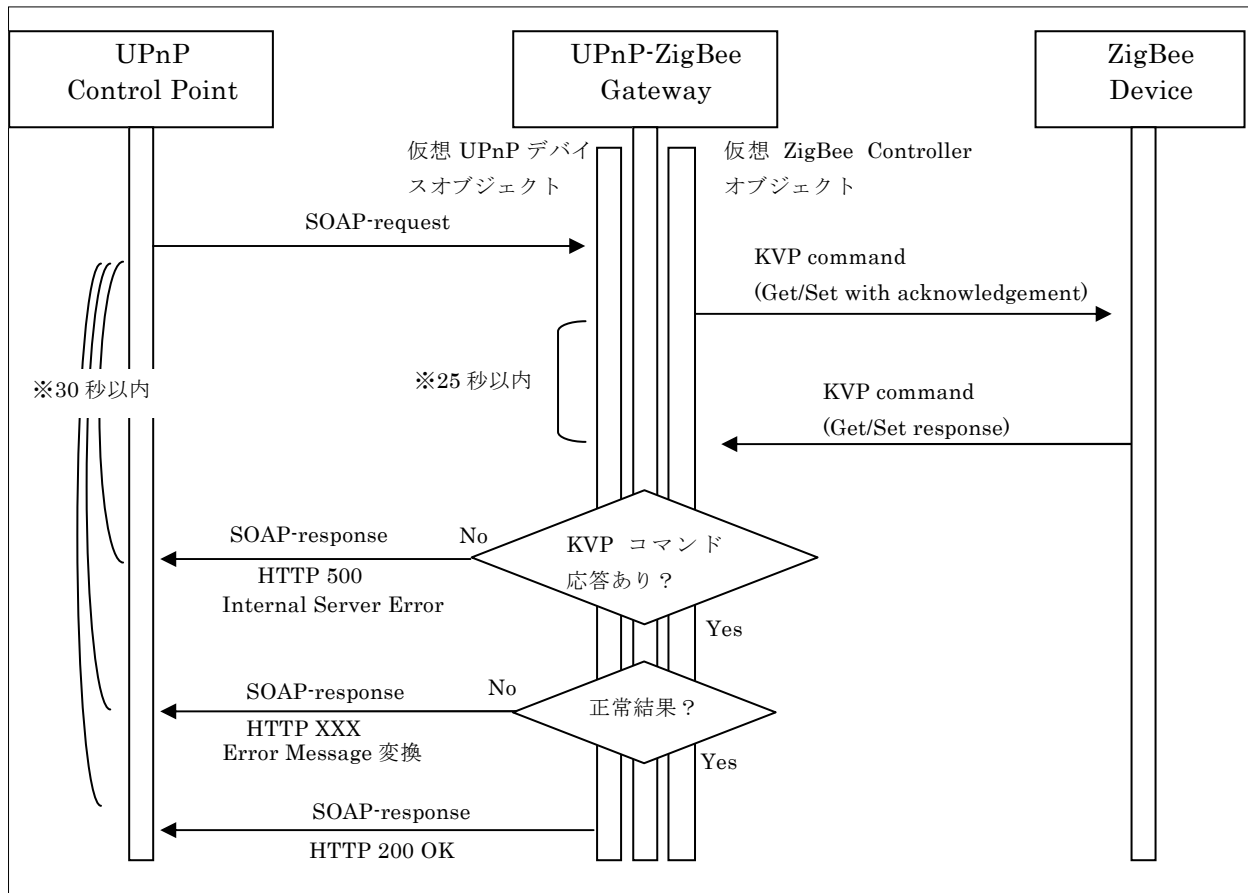


図 31. UPnP CP から ZigBee デバイスへのコントロールシーケンス

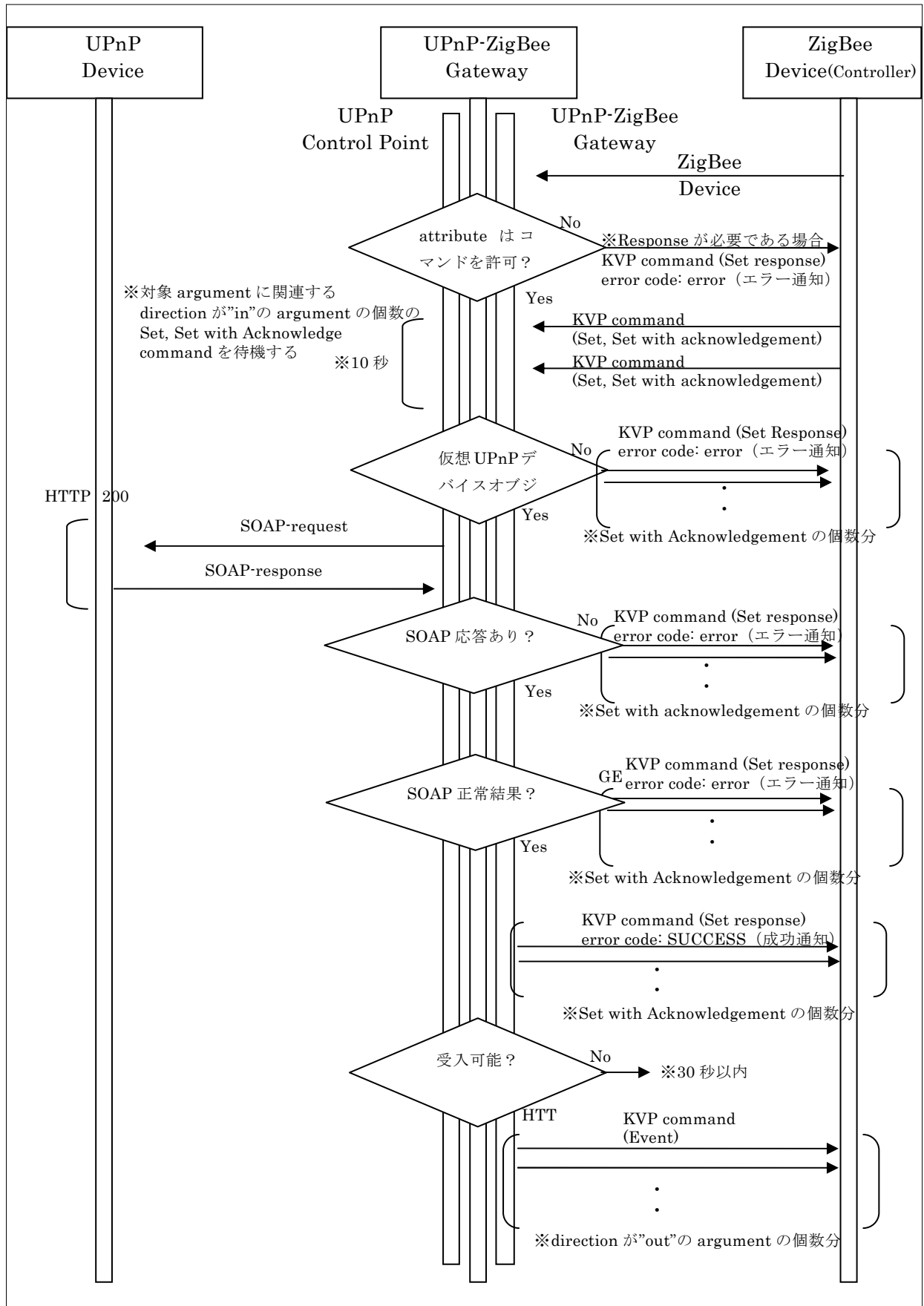


図 32. ZigBee デバイスから UPnP デバイスへのコントロールシーケンス (Set、又は Set with acknowledgement の場合)

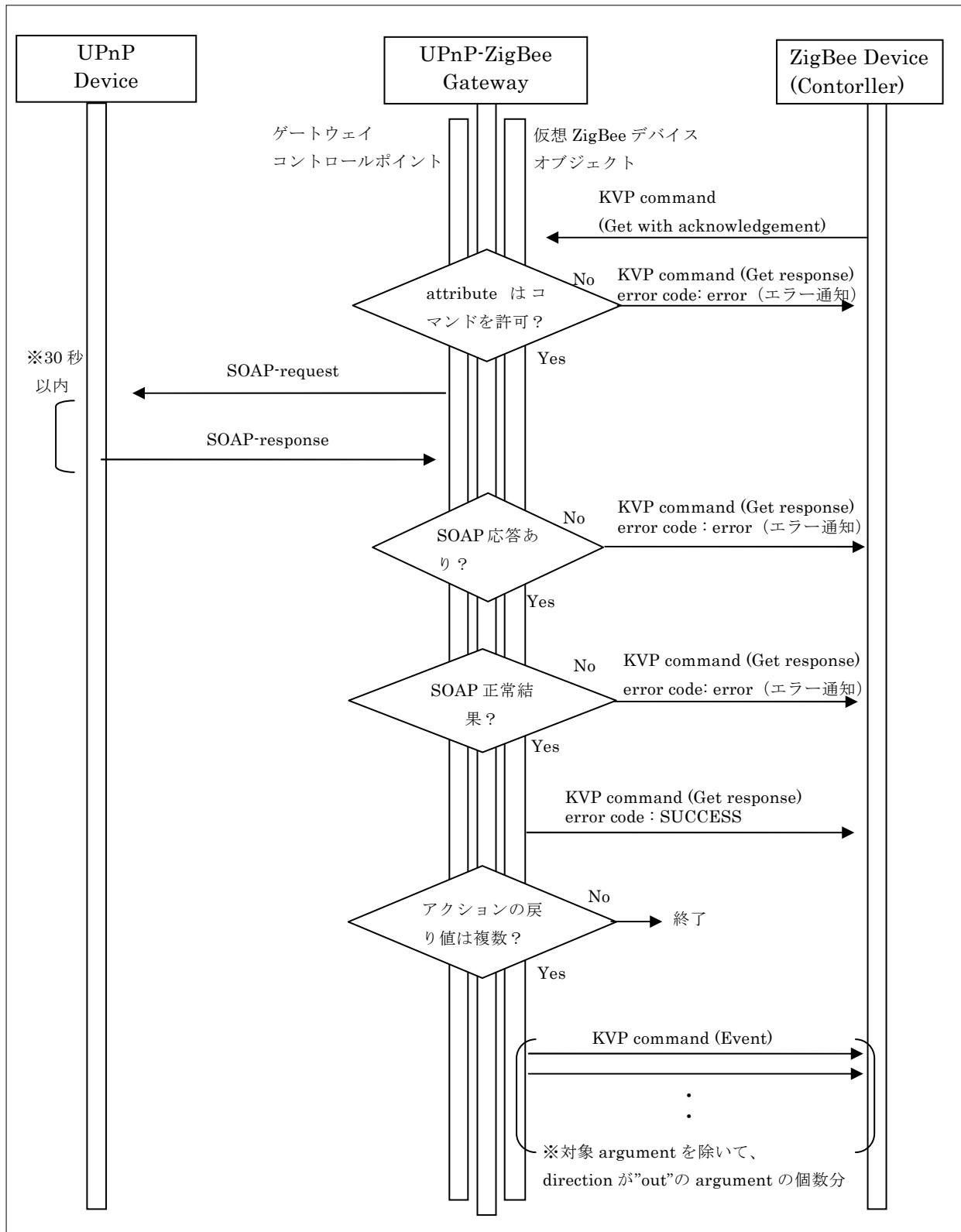


図 33. ZigBee デバイスから UPnP デバイスへのコントロールシーケンス
(Get with acknowledgement の場合)

3.3.6.4. イベントینگ変換

3.3.6.4.1. UPnP におけるイベントینگプロトコル

UPnP におけるイベント制御（イベントینگ）動作には GENA が利用されている。
GENA による UPnP が提供するサービス間のイベント処理を行う手順は以下のようになっている。

1. コントロールポイントが、デバイスに向けて、イベント購読要求（Subscribe）を送信する。
2. デバイスは、コントロールポイントに対して、レスポンス（HTTP 200 OK）を返信する。
3. イベント発生時にデバイスは、コントロールポイントに対して、イベント情報（GENA NOTIFY）を送信する。
4. イベント情報を受け取ったコントロールポイントは、イベント情報に対する返信（HTTP 200 OK）を返信する。

3.3.6.4.2. ZigBee におけるイベントینگプロトコル

ZigBee におけるイベント制御動作にはコマンドタイプに”EVENT”が指定された KVP コマンドフレームが用いられる。EVENT コマンド指定された KVP コマンドフレームによる ZigBee アプリケーションオブジェクト間のイベント処理の手順は以下のようになっている。ZigBee では UPnP と異なり前もってイベント購読の手続きを行う前処理がないことに注意する必要がある。

1. イベント送信を行うアプリケーションオブジェクトは宛先となるアプリケーションオブジェクトに向けて、”EVENT”もしくは”EVENT with ACKNOWLEDGEMENT”のコマンドタイプで指定された KVP コマンドフレームを送信する。
2. ”EVENT with ACKNOWLEDGEMENT”のコマンドタイプ指定の KVP コマンドフレームを受け取ったアプリケーションオブジェクトは、”EVENT RESPONSE”のコマンドタイプが指定された KVP コマンドフレームを送信元に対して送り返す。

3.3.6.4.3. ゲートウェイにおけるイベントینگプロトコルの変換

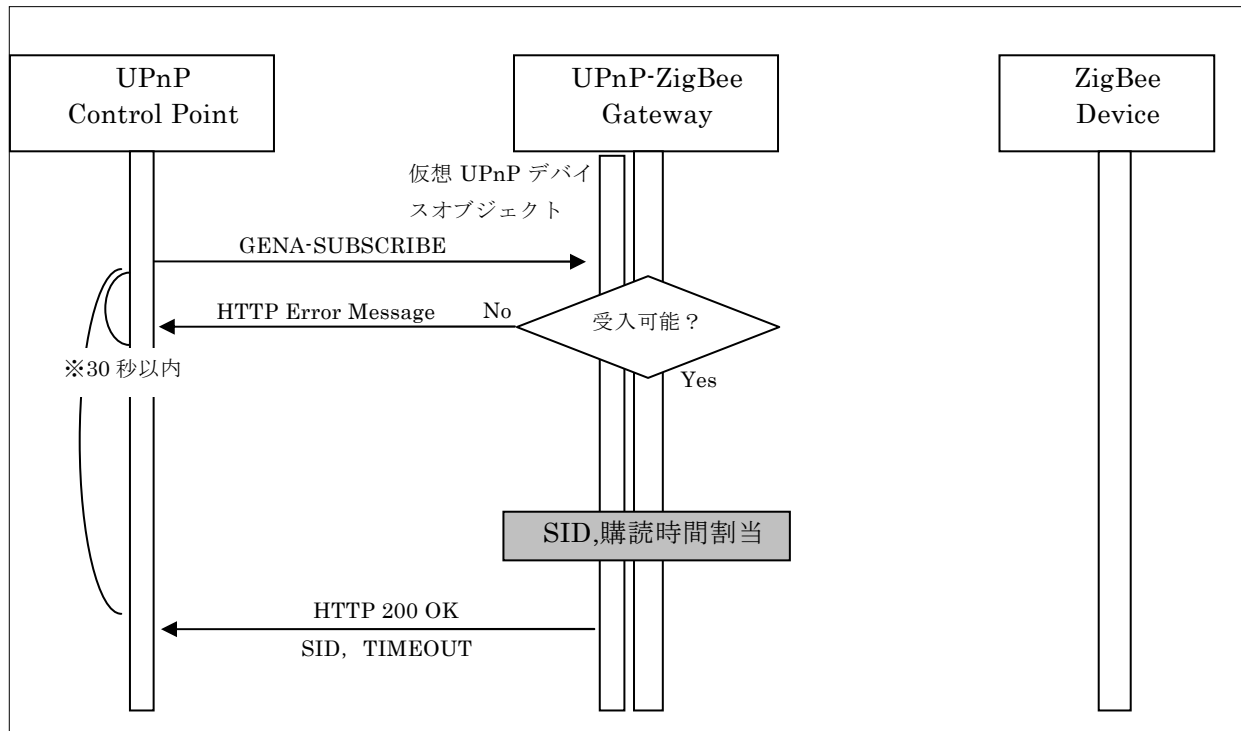


図 34. UPnP Control Point からのイベント購読申込シーケンス

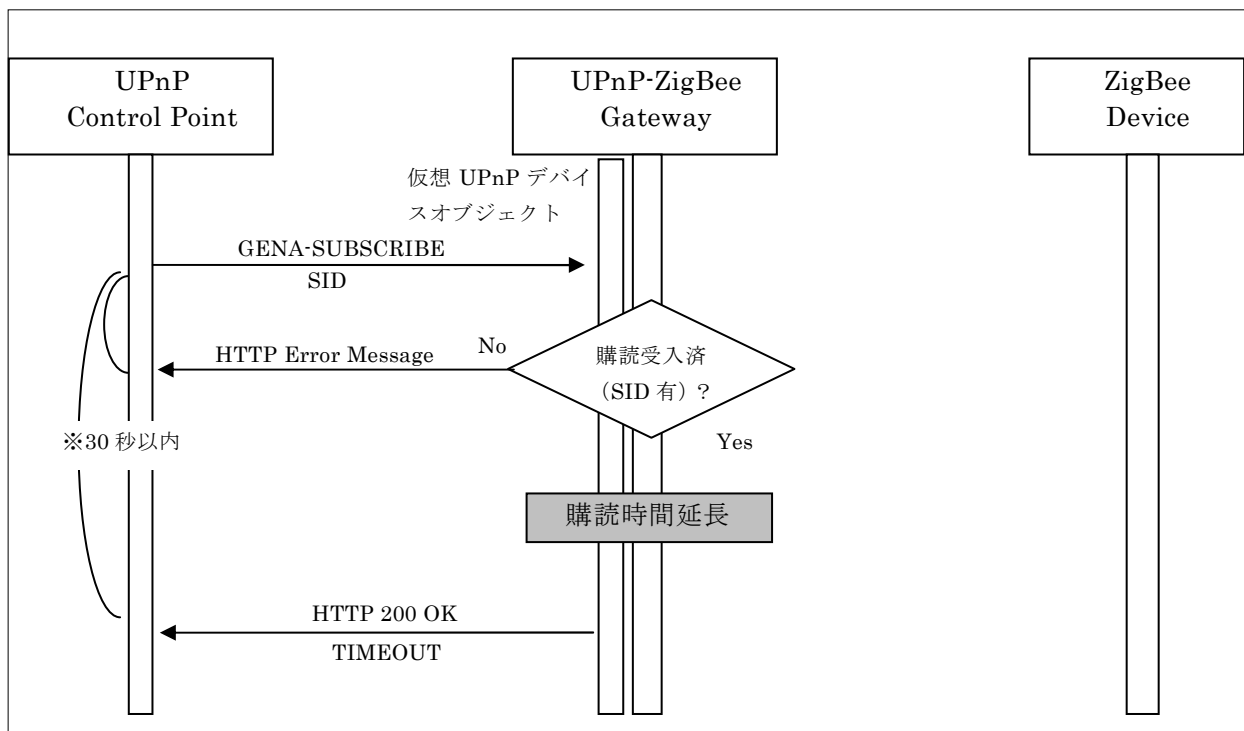


図 35. UPnP Control Point からのイベント購読更新シーケンス

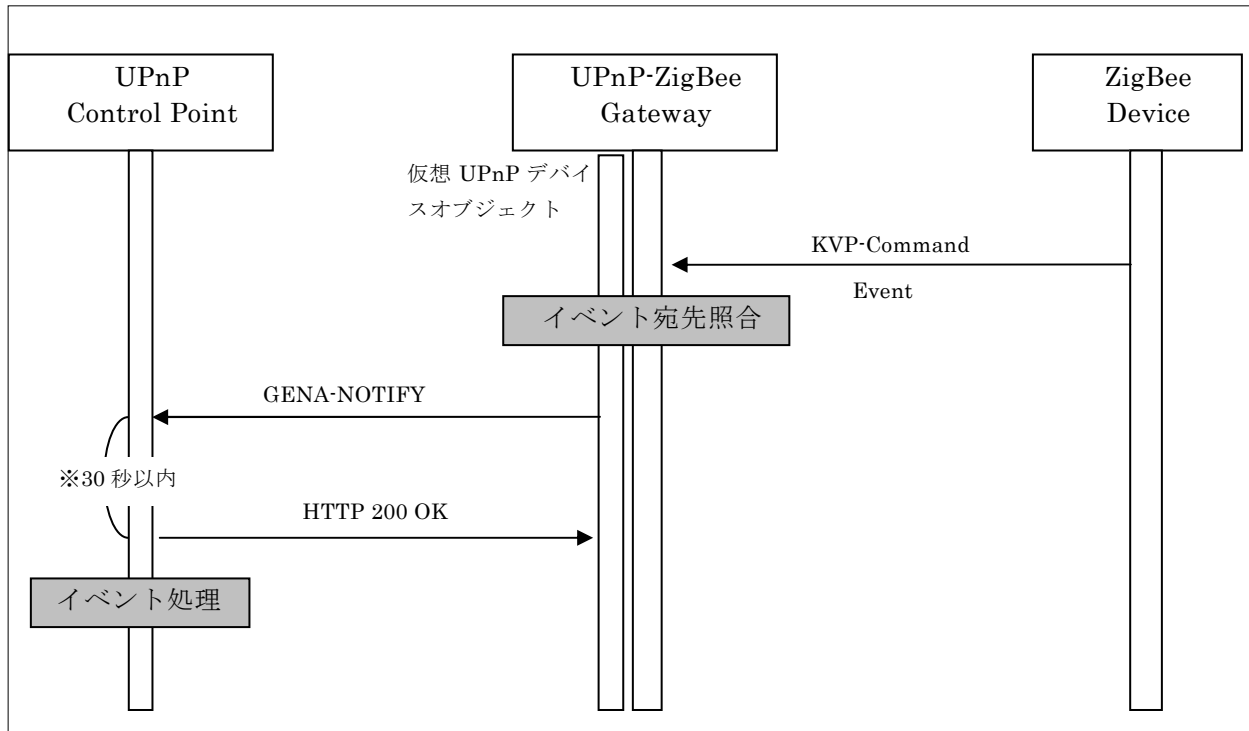


図 36. ZigBee デバイスから UPnP Control Point へのイベント処理シーケンス
(KVP コマンドフレームが Event の場合)

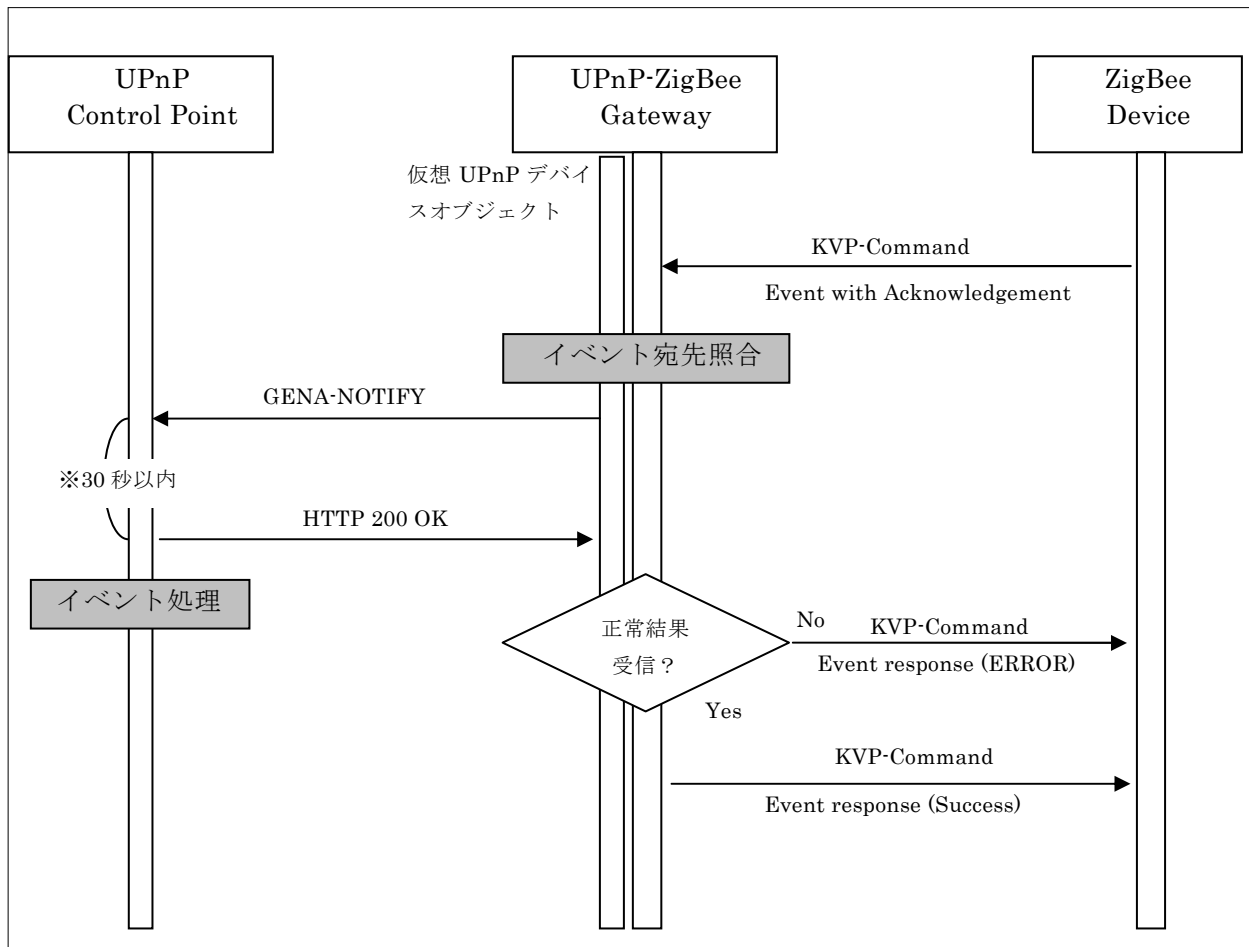


図 37. ZigBee デバイスから UPnP Control Point へのイベント処理シーケンス
(KVP コマンドフレームが Event with Acknowledgement の場合)

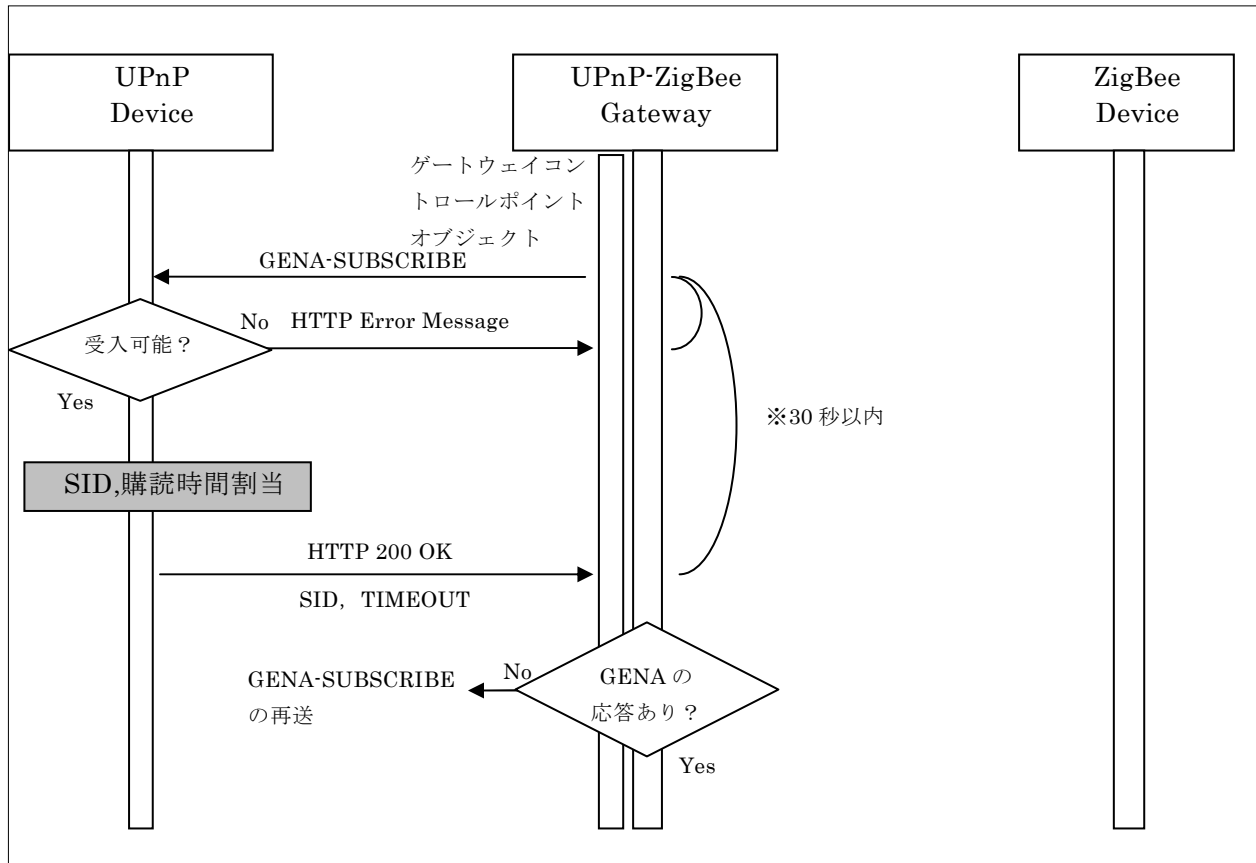


図 38. UPnP デバイスへのイベント購読申込シーケンス

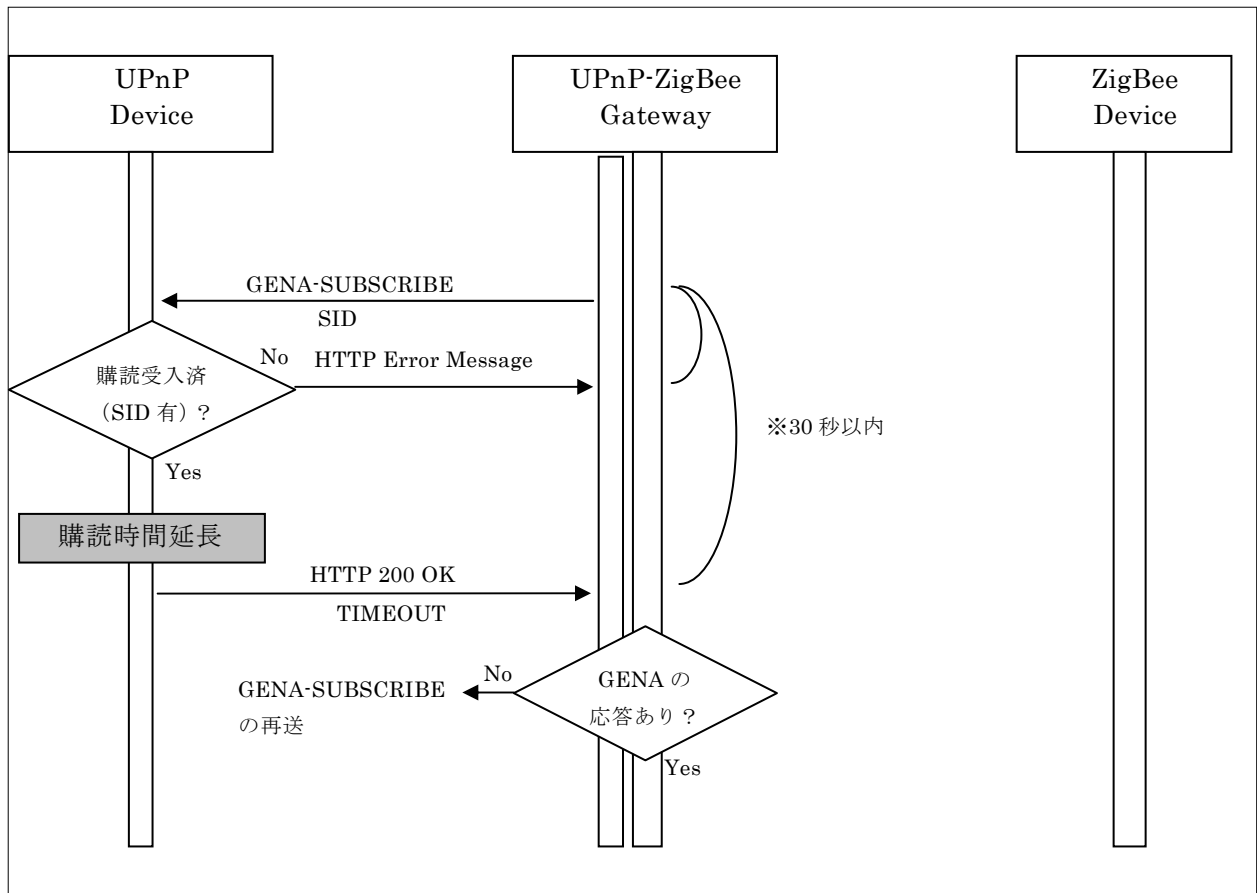


図 39. UPnP デバイスへのイベント購読更新シーケンス

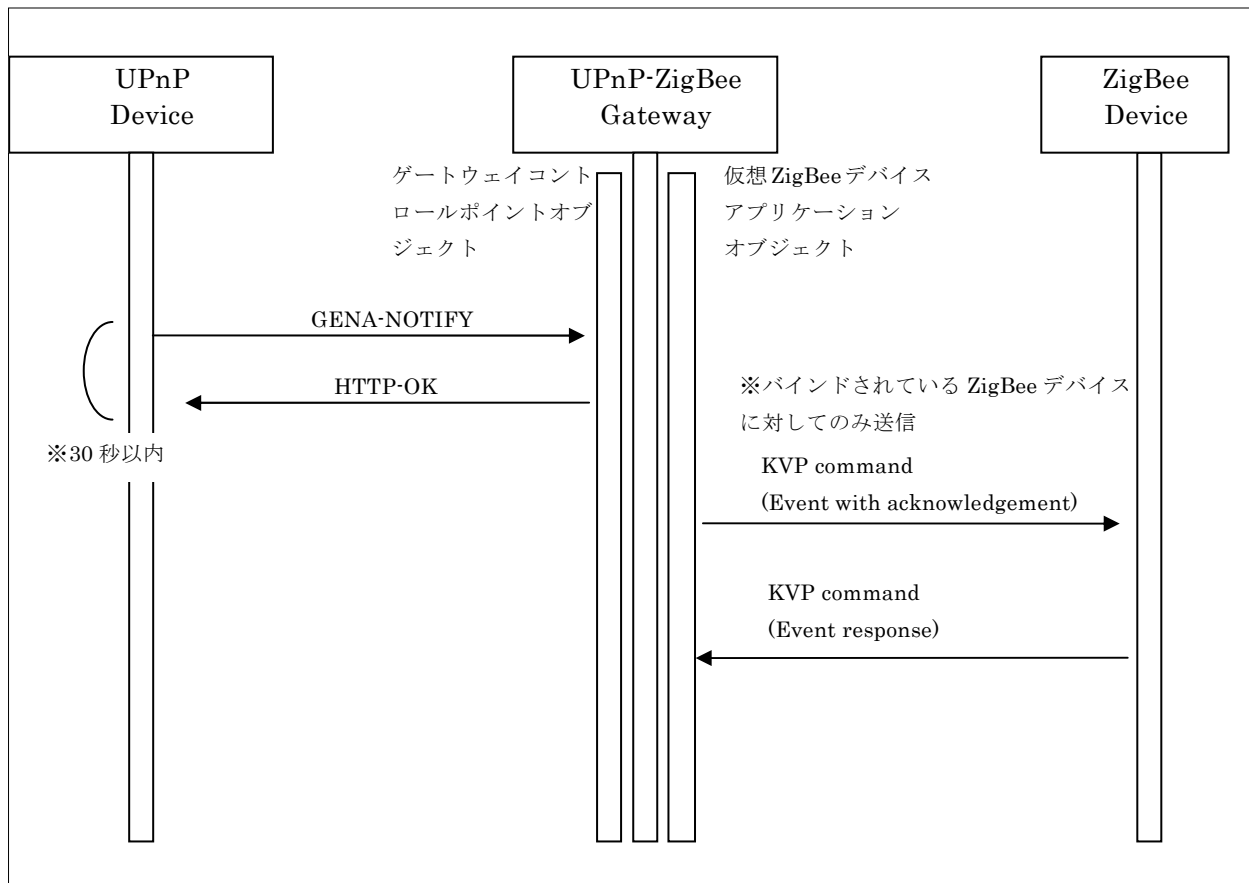


図 40. UPnP デバイスから ZigBee デバイスへのイベント処理シーケンス

3.3.7. プロファイルに応じたコンフィギュレーション

ゲートウェイが提供する機能として、ディスクリプションの翻訳とマッピングテーブルの作成、仮想デバイスの作成機能、データフォーマット変換機能、プロトコル変換機能が必須となっている。これらの機能によって UPnP ネットワークと ZigBee ネットワーク間における透過的なデバイス探索とサービス探索、サービス間の制御メッセージの送受信、サービス間のイベントメッセージの送受信が可能となる。

これら基本的なゲートウェイの変換機能以外に考える必要があることとして、ZigBee ネットワーク上のプロファイル特性に応じたコンフィギュレーションを施す必要がある。具体的には、ZigBee センサーネットワークにおいては、個々の ZigBee デバイスは電池駆動である場合が多く、データフレームの送受信に比例した消費電力はセンサーアプリケーションが機能するかどうかの死活問題に直結する場合があります。仮に UPnP のプロトコルを全て透過した場合に、センサーネットワークが機能しなくなる可能性がある。

また、ゲートウェイを設けることによって、本来 ZigBee ネットワーク上で Peer to Peer だった通信トラフィックが、ゲートウェイという特異点を必ず通ることによるトラフィック上の影響も生じてくる可能性があり、パケットロスや遅延といった影響も生じる。

このような問題は、本質的には ZigBee ネットワーク上にどのような種類（アプリケーションプロファイル）のサービスが展開されるか、ということに依存する問題であり、アプリケーションプロファイルに応じたコンフィギュレーションが必要となる。

本節ではこのような問題点を踏まえ、ゲートウェイのオプション的な位置づけとしてのコンフィギュレーション機能を規定することを目的とする。コンフィギュレーションによって、ゲートウェイでは、UPnP ネットワークと ZigBee ネットワークをエンド-エンド間でまたいだ、リアルタイム通信トラフィックを吸収し、上記のような問題を解決できなければならない。

本仕様では、そのようなコンフィギュレーションとして、以下の節のノンキャッシングモードとキャッシングモードの二つを規定している。

表 22. コンフィギュレーション設定のパラメータ

Parameter	説明
Discovery Protocol Transparency	Discovery Protocol の透過率設定
Eventing Protocol Transparency	Eventing Protocol の透過率設定
Control Protocol Transparency	Control Protocol の透過率設定
Property Search	DLNA コンテンツを生成する際の、ZigBee ネットワークのサーチ手法。ユニキャストもしくはブロードキャストで行い、ユニキャストの場合は、ユニキャスト送信間隔の設定が含まれる。

ゲートウェイがプロトコル変換等の機能によって実現するメッセージ転送機能は、以下のようなフローに従って分岐動作する。

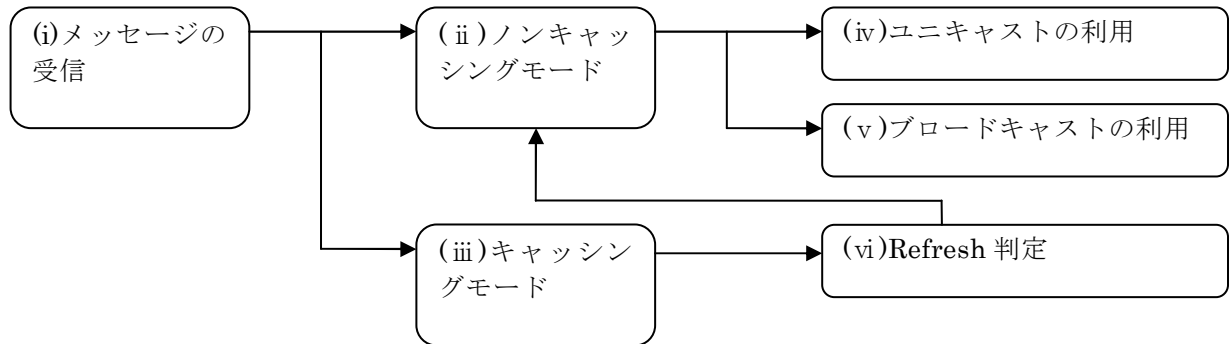


図 41. ゲートウェイの ZigBee ネットワークのサーチ手法

(i)メッセージの受信

ゲートウェイは、SSDP や、SOAP、GENA といったメッセージを受け取ると、ZigBee ネットワークに転送する際に、いくつかの転送モードに従った動作を行う。

(ii) ノンキャッシングモード

UPnP が提供するプロトコルによって生じるメッセージをゲートウェイでそのまま全て透過するモードであり、ZigBee ネットワーク上の ZigBee デバイスとゲートウェイに生じる負荷は高いが、リアルタイムでの情報のやりとりを行いたい場合には有用となる。このとき具体的な送信手段としては、ユニキャスト送信とブロードキャスト送信があり、iv、もしくは、v に動作は遷移する。

(iii)キャッシングモード

UPnP プロトコルは頻繁なパケットのやりとりを行うことによってリアルタイムな情報取得を行うことができる。しかし、ZigBee ネットワークに対して頻繁なメッセージ通信をそのまま透過させることは、ZigBee デバイスのバッテリー残量を極端に減らしたり、応答性を悪化させたりすることもある。キャッシングモードは、全てのメッセージをそのまま透過するノンキャッシングモードと異なり、ゲートウェイに情報を蓄えておき、その情報をゲートウェイが参照返答することで、反対側のネットワークには余計なメッセージを流さないというモードである。特に UPnP ネットワーク側からのメッセージに対してゲートウェイがキャッシュ参照することは効果があり、SSDP や SOAP、GENA といった UPnP プロトコルに対する応答に関して、ゲートウェイではキャッシュ情報を参照し、ZigBee ネットワークに対してはプロトコル変換を行ったコマンドフレームを送信しない。

(iv)ユニキャストの利用

パケット転送の際にユニキャストを用いて、メッセージの転送や、DLNA コンテンツの生成時に必要となるネットワークのサーチ（状態情報の一斉取得）を行う。

(v)ブロードキャストの利用

パケット転送の際にブロードキャストを用いて、メッセージの転送や、DLNA コンテンツの生成に必要なネットワークのサーチ（状態情報の一斉取得）を行う。

(vi)リフレッシュの判定

キャッシングモード時の、キャッシュ情報が TTL で表される閾値以下の場合には、ノンキャッシングモードの振る舞いに遷移する。閾値以上の場合には、キャッシュ情報をゲートウェイはレスポンスとして返す。

3.3.7.1. SSDP の応答

キャッシングモードのゲートウェイは、UPnP ネットワークからの SSDP 検索要求に対して、生成済みの仮想 UPnP デバイスの応答を行い、ZigBee ネットワーク上のデバイスディスクリプタ要求は行わない。

3.3.7.2. GENA モデューレーション

UPnP では連続した GENA イベントメッセージは、最大1 イベントメッセージにモデューレーションがかけられる。あまりにも頻繁な GENA イベントメッセージの頻度に対しては、UPnP 仕様以上のモデューレーションをゲートウェイがかけることが許可される。

3.3.7.3. ゲートウェイアプリケーションからの参照

次節の DLNA-ZigBee ゲートウェイのような、UPnP-ZigBee ゲートウェイ機能を用いるゲートウェイアプリケーションが情報を参照する頻度をコントロールすることが許可される。

3.4. DLNA-ZigBee 接続仕様

DLNA-ZigBee ゲートウェイ機能は、3.3 節で決められる UPnP-ZigBee ゲートウェイ機能が提供する機能上にあるアプリケーションとして位置づけられ、センサーネットワークの状況をユーザーが所持する DLNA 対応のレンダリングデバイスに表示することで、センサーネットワークに対するユーザーのアクセサビリティ向上を図る目的を持つ。DLNA-ZigBee ゲートウェイ機能は、UPnP-ZigBee ゲートウェイが提供するオプション機能である。

3.4.1. 機能説明

DLNA-ZigBee ゲートウェイは、UPnP-ZigBee ゲートウェイインフラストラクチャデバイス上に成り立つゲートウェイアプリケーションであり、以下のような機能を持つ。

3.4.1.1. Digital Media Server 機能の提供

ゲートウェイは、UPnP ネットワーク側に対し、Digital Media Server としての機能を提供する。このゲートウェイ上に実装される Digital Media Server 機能は通常の DLNA DMS とは異なり、コンテンツの取得リクエストを受信したタイミングでコンテンツを生成する要求を出し、生成された通知を受けるまで待機し、生成された段階でコンテンツを配信する機能を持つ。また、提供するコンテンツは ZigBee センサーネットワークの状況をユーザーに伝えることのできるメディアコンテンツである必要がある。DLNA-ZigBee ゲートウェイにおいて Digital Media Server 機能の提供は必須である。公開される Media Server としてのデバイスのフレンドリーネームは、ゲートウェイデバイスであることが明記される必要がある。

3.4.1.2. DLNA コンテンツの動的生成と公開

DLNA-ZigBee ゲートウェイでは、UPnP-ZigBee ゲートウェイの機能を用いて、ZigBee センサー情報（属性値）を動的に収集する。その結果取得できた ZigBee センサー情報を、DLNA JPEG 画像フォーマットに変換したものを公開する。DLNA では他に動画や音声のメディアを取り扱うことができるが、どのメディアとフォーマットを用いるかは、DLNA ガイドラインに従った範囲で選択は自由であることが許される。

3.4.1.2.1. コンテンツ生成エンジンの配備と制御

ZigBee ネットワークに配備されるセンサーや同一のセンサーから生成可能なメディアコンテンツの多様性に対応するために、DLNA-ZigBee ゲートウェイはメディアコンテンツを生成するためのアプリケーション機能(コンテンツ生成エンジン)を外部モジュールとして読み込み、制御する機能が求められる。DLNA-ZigBee ゲートウェイは、コンテンツ生成エンジンの配備が完了すると、コンテンツ生成エンジンが生成可能なコンテンツのメディア情報(メタデータ)を取得し、コンテンツを生成するために必要な ZigBee デバイスが存在する場合、Digital Media Server が持つ仮想的な Content Directory Service に登録する。また、各コンテンツ生成エンジンによって指定されるアプリケーションプロファイル ID に属する ZigBee アプリケーションオブジェクトが持つクラスターの、特定の属性値を取得し、コンテンツ生成エンジンに入力してコンテンツの生成を要求する。コンテンツ生成エンジンはアプリケーションごとに定められる仕様に従って動的に DLNA ガイドラインに従ったコンテンツを生成する。コンテンツを生成するタイミングはあらかじめ作成しておく(キャッシュ)か、もしくは、リクエスト毎に動的に生成(ノンキャッシュ)するかを選択することができる。

3.4.1.2.2. Digital Media Server への配置

コンテンツ生成エンジンが生成する DLNA コンテンツは、DLNA DMP からのリクエストに応じて動的に生成されるか、あらかじめ生成しているが、それらのコンテンツを Digital Media Server に配置する仕組みが求められる。Digital Media Server は仮想 Content Directory Service コンテナに、コンテンツのインデックスを配置し、実態となるコンテンツは動的にリンクすることで、動的なコンテンツの配信を可能にする必要がある。

3.4.1.2.3. 仮想 Content Directory Service コンテナへの配置

動的に配置されるコンテンツはコンテンツディレクトリサービスが扱うコンテナ上に配置される。この配置は通常の DLNA コンテンツと同じ位置に配置される。

3.1.1 節のユースケースで見たように、DLNA DMP からゲートウェイは DMS として発見される。(図 8) そして ZigBee デバイスの状態情報は AV コンテンツとして視覚化され DMP から閲覧される。このゲートウェイ上の Media Server 内に実装される Content Directory Service(CDS)は、一般のものとは異なり、デバイスマッピングテーブルにエントリーされた ZigBee デバイスのサービス状況 (アプリケーションプロファイル, デバイスディスクリプション, Cluster, attribute) の情報を参照して、実体のないコンテンツのインデックス情報を公開する。インデックスを取得した DMP がこの仮想コンテンツにアクセスしようとする時、CDS はセンサー情報コンテンツ生成エンジンに問い合わせる。コンテンツ生成エンジンは状態情報データベースに問い合わせ、対象となる ZigBee デバイス群の状態情報を知り、その情報を元に DLNA 対応のコンテンツフォーマット、静止画像(JPEG, GIF, PNG など, サイズ指定あり)、動画 (MPEG1,2,4, WVM など) , 音声(LPCM, MP3, ATRAC など)を生成する。

3.4.1.2.4. 配置したコンテンツの削除

仮想 Content Directory Service コンテナに配置されたコンテンツは、対応する ZigBee センサーが消失した場合には登録を解除する必要がある。DLNA-ZigBee ゲートウェイが各コンテンツ生成エンジンによって指定されるアプリケーションプロファイルIDに属する ZigBee デバイスが全て消失したことを定期的な確認、もしくは UPnP-ZigBee ゲートウェイからの通知によって検出した場合、対象となるコンテンツ生成エンジンが提供する全てのコンテンツの登録を仮想 Content Directory Server から解除する。また、コンテンツの実体が既に生成されていた場合、当該コンテンツの削除も行う。

3.4.1.3. ZigBee センサー情報の収集

DLNA-ZigBee ゲートウェイは、センサーネットワークの状況を示すコンテンツを生成するための情報を提供するものとして、UPnP-ZigBee ゲートウェイの機能を用いた ZigBee センサー情報の収集機能を提供する必要がある。ZigBee センサー情報の収集には、DLNA-ZigBee ゲートウェイ機能と UPnP-ZigBee ゲートウェイ機能の間において、デバイスマッピングテーブルにエントリーされた ZigBee デバイスのサービス状況の確認と、ZigBee デバイスもしくはセンサー情報をキャッシュしておく状態情報データベースを参照し、指定された属性値の情報を取得するためのインターフェースが必要になる。DLNA-ZigBee ゲートウェイは (1)特定のデバイス、(2)特定の種別のデバイス(特定の Profile)、(3)ネットワーク中の全てのデバイスに対する属性値情報の取得の機能を提供し、コンテンツ生成エンジンの指定に従い、属性値情報の取得を実施する。

3.4.2. メディアフォーマット変換

ZigBee センサーのデータ（属性値）をコンテンツへ加工する方法としては以下の二通りが選択できる。どちらの方法を取るかは自由に選択できるが、出力されるコンテンツは一つとなる。

3.4.2.1. 一つのセンサー情報のメディアフォーマット変換

一つのセンサー情報が持つ属性値の変化を連続的に監視し、一つの DLNA コンテンツとして変換する。

3.4.2.2. 複数のセンサー情報のメディアフォーマット変換

複数のセンサーがある瞬間に持つ属性情報をまとめて、一つの DLNA コンテンツに変換し、出力する。この時、対象となる属性情報は、同一の種類センサーの同一種類の属性、同一の種類センサーの別の種類の属性、別の種類のセンサーの持つ属性のいずれであっても良く、コンテンツ生成エンジンアプリケーションの実装に依存する。

3.4.3. コンテンツ生成エンジン基本仕様

DLNA-ZigBee ゲートウェイの外部モジュールとしてコンテンツ生成エンジンアプリケーションを作成する際には、コンテンツ生成エンジンは、メディアフォーマットの変換を用いセンサーの属性値情報からメディアコンテンツを生成して提供する基本機能と、センサー情報の収集やコンテンツ情報の登録のための各種データを保持する機能を持つ必要がある。また、ゲートウェイとの接続のために、基本となるインターフェイスを実装しなければならない。以下にコンテンツ生成エンジンアプリケーションを生成する際に満たすべき仕様を挙げる。なお、生成されるコンテンツの内容、メディアフォーマット変換方法はアプリケーションの実装に依存し、本仕様においては規定しない。

3.4.3.1. コンテンツの生成

コンテンツ生成エンジンは DLNA-ZigBee ゲートウェイからの指示に従い、与えられたセンサー属性値情報をベースとしてメディアフォーマットの変換を行い、生成したコンテンツの識別子を返答する機能を持つ必要がある。ゲートウェイから与えられるセンサー属性値情報は、属性値の識別情報、取得センサーノードの識別情報、取得した属性の値、取得時間から構成される。ZigBee センサー情報の取り扱い(保存するかその都度取得を行うか、時間判定を行うか否か、データが足りない場合はどうするかなど)、生成されるコンテンツの内容、生成方法は各アプリケーションの仕様に依存する。生成したコンテンツの実体は、ゲートウェイによって指定されたコンテンツ格納領域に配置し、DMS に登録するメディア情報にリンクする。

3.4.3.2. サービス分類情報の提供

コンテンツ生成エンジンごとの識別のために、コンテンツ生成エンジンアプリケーションが提供するサービスを表した名前を持つ必要がある。サービス名は DLNA-ZigBee ゲートウェイによる DMS へのコンテンツインデックス登録時に参照され、ゲートウェイが提供するセンサー情報の分類名としてユーザーの持つ DMP により取得される。コンテンツ生成エンジンが提供するコンテンツは全て、このサービス(センサー分類)配下のコンテンツとして、ユーザーから閲覧される。

3.4.3.3. 生成コンテンツのメディア情報の提供

コンテンツ生成エンジンは生成するコンテンツを DMS において公開するため、コンテンツのメディア情報(メタデータ)を持たなくてはならない。具体的には、生成するメディアコンテンツの種類(MPEG2,MPEG4,JPEG,MP3 などのメディアフォーマットの拡張子)、生成したコンテンツの実体に付けられるファイル名、生成したコンテンツのタイトル、コンテンツ保護の有無などである。その他コンテンツの種別ごとに規定されるメタ情報(例えば静止画像であれば画像サイズなど)の提供はアプリケーションの実装に依存する。ここで、DLNA-ZigBee ゲートウェイにおいて許容されるメディアコンテンツのタイプは、DLNA 対応のコンテンツフォーマットのみであるので注意すること。

3.4.3.4. 対象センサー種別・属性情報の提供

コンテンツ生成エンジンはコンテンツ変換時に対象とするセンサーの種類とセンシング対象の属性値を特定する情報を、最低 1 組持たなければならない。具体的には Profile、Cluster、Attribute の ID、Attribute のデータタイプであり、それらを DLNA-ZigBee ゲートウェイに提供するためのインターフェイスを有する必要がある。また、コンテンツ生成エンジンは必要であれば ZigBee センサーの IEEE64bit 拡張アドレスを指定し、属性値を取得する対象を限定することもできる。この場合、コンテンツ生成エンジンは取得したい全てのセンサーの IEEE64bit アドレスを持つ。

3.4.4. 基本シーケンス

DLNA-ZigBee ゲートウェイは、UPnP ネットワーク上に構築される DLNA ネットワーク上に存在する Digital Media Player に対してサービスを提供する。DLNA ネットワークから見たゲートウェイと DLNA Digital Media Player との間のプロトコルシーケンスは以下ようになる。

3.4.4.1. DLNA-ZigBee ゲートウェイの初期化

DLNA-ZigBee ゲートウェイの初期化シーケンスは二段階の動作により構成される。

(1) Digital Media Server の初期化

ゲートウェイに実装されている Digital Media Server は、ゲートウェイの初期化時に同時に初期化される。DLNA-ZigBee ゲートウェイは UPnP-ZigBee ゲートウェイの上位に位置するアプリケーション機能として位置づけられるため、UPnP-ZigBee ゲートウェイの初期化が終了した状態で初期化が開始される必要がある。従って、DLNA-ZigBee ゲートウェイの初期化が始まる段階では、ゲートウェイの IP アドレッシング、UPnP ネットワークの初期化、ZigBee ネットワークの初期化が全て終了していることになる。

初期化の内容としては、UPnP-ZigBee ゲートウェイから指定されたフレンドリーネームを設定し、UPnP の SSDP を用いて Digital Media Server は自分自身を DLNA ネットワークに対してアドバタイズする。これにより DLNA Digital Media Player から Digital Media Server がアクセス可能となるが、この時点ではまだコンテンツは登録されていない。

(2) コンテンツ情報の登録

次に DLNA-ZigBee ゲートウェイは外部モジュールとして与えられる各コンテンツ生成エンジンを読み込み、画像化などの DLNA コンテンツ生成対象となるアプリケーションプロファイル ID とターゲット属性の情報を取得する。更に UPnP-ZigBee ゲートウェイに対し指定し、対象とするアプリケーションプロファイル ID を持つ ZigBee アプリケーションオブジェクトの属性が取得可能な状態とする。

UPnP-ZigBee ゲートウェイは上位アプリケーションでもある DLNA-ZigBee ゲートウェイから、対象とする ZigBee アプリケーションプロファイル ID の指定を受けると、エントリーテーブルを参照し、対

象プロファイルを持つデバイスが存在する場合はその情報を返答する。エントリーテーブル内に対象となるプロファイルを持つデバイスが存在しない場合、**ZigBee** のネットワークに対し、そのプロファイル ID を持つ **ZigBee** アプリケーションオブジェクトの探索を実行し、その結果を返答する。**ZigBee** アプリケーションオブジェクトが発見できた場合、**UPnP-ZigBee** ゲートウェイは仮想 **UPnP** デバイスを生成する。

DLNA-ZigBee ゲートウェイは対象プロファイルのデバイスが発見できた場合、コンテンツ生成エンジンからサービス情報、コンテンツメディア情報を取得し、**Digital Media Server** の持つ仮想 **CDS** に対し、サービス情報、コンテンツ情報を登録する。これにより、コンテンツのメタ情報が **DLNA** ネットワークに対して公開され、**DLNA Digital Media Player** から **Digital Media Server** にアクセスすると、**DLNA-ZigBee** ゲートウェイが提供するサービス/コンテンツの情報を取得することができる。この時点ではコンテンツの実体は存在しないが、ユーザーが閲覧したいコンテンツを選択すると、コンテンツの生成が行われる。対象プロファイルのデバイスが発見できなかった場合、当該コンテンツ生成エンジンのサービスは登録されず、対象デバイスが **ZigBee** ネットワークに加入してくるまでコンテンツ生成エンジンの動作は中断する。

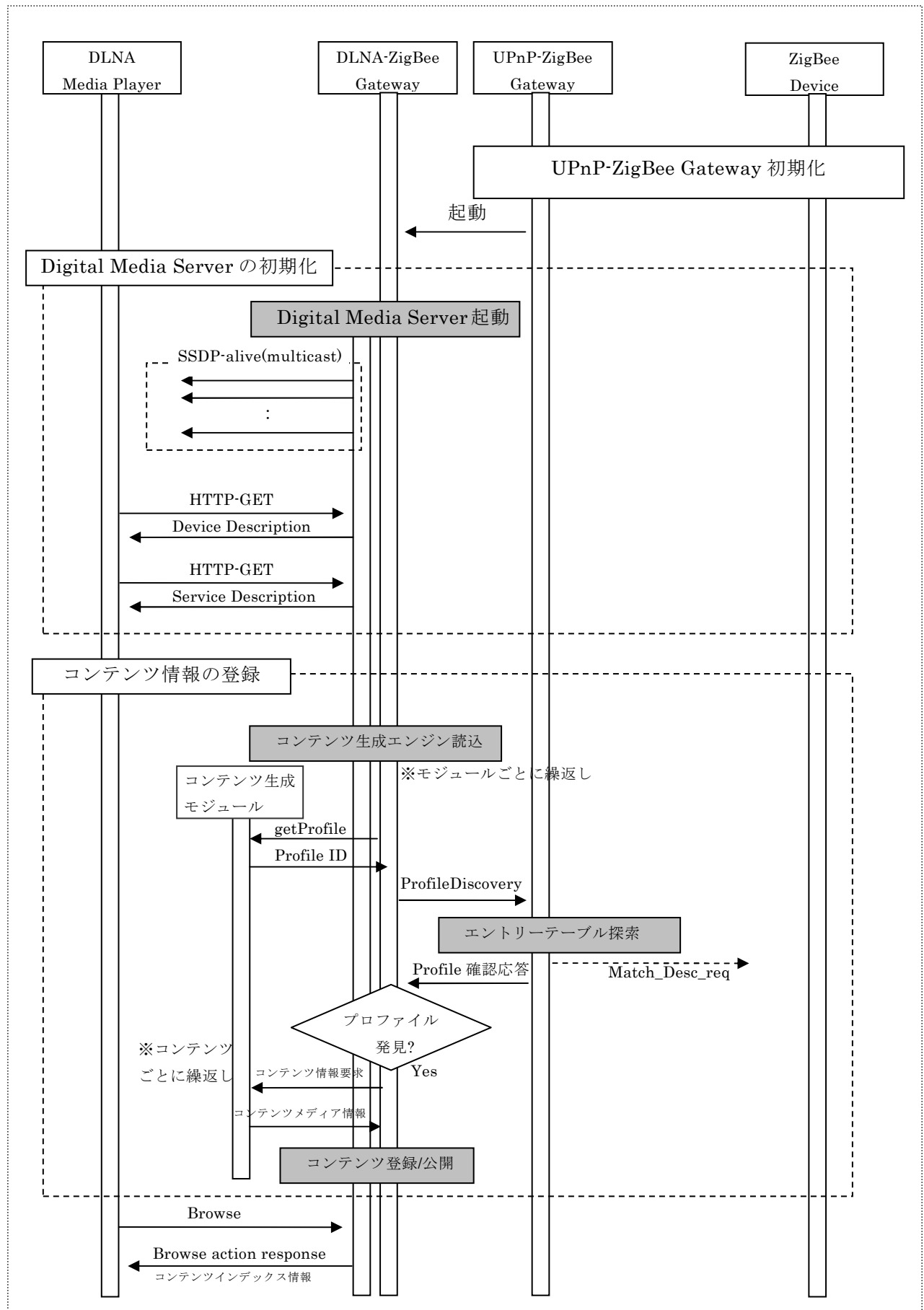


図 42. DLNA-ZigBee ゲートウェイ初期化基本シーケンス

3.4.4.2. コンテンツの生成

DLNA-ZigBee ゲートウェイで公開されたコンテンツを DLNA Digital Media Player から取得するリクエストを Digital Media Server が受信すると、Digital Media Server は DLNA-ZigBee ゲートウェイに対しコンテンツの生成を要求する。ゲートウェイはリクエスト対象のコンテンツにマッピングされているコンテンツ生成エンジンを参照し、センサー種別及び属性を指定して UPnP-ZigBee ゲートウェイ機能に対し属性値の取得を要求する。UPnP-ZigBee ゲートウェイはコンフィギュレーションによって指定された動作を行い、ZigBee デバイスもしくは内部のキャッシュから対象の属性値を取得する。DLNA-ZigBee ゲートウェイは取得した属性値を元に、コンテンツ生成エンジンに対してコンテンツの生成を依頼する。コンテンツ生成エンジンによるコンテンツの生成に成功すると、実コンテンツへのリンクを行い、Digital Media Server から DLNA Digital Media Player に対して配信を行う。コンテンツの生成の要求を行ってから 27 秒以上経過するか、いずれかの段階でコンテンツの生成に失敗した場合、Digital Media Server は適切なエラーメッセージを DLNA Digital Media Player に対して配信する。

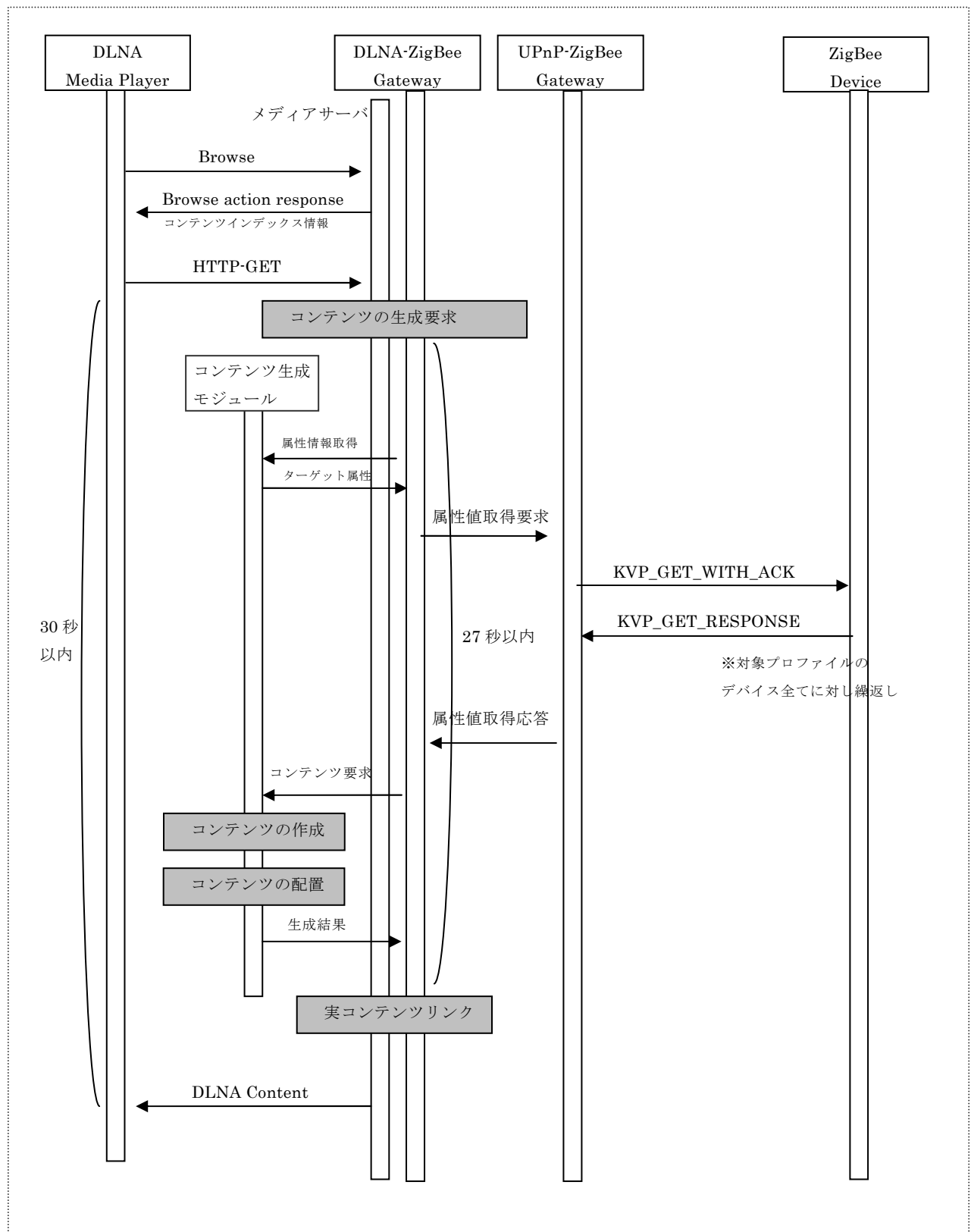


図 43. コンテンツ生成シーケンス

3.4.4.3. コンテンツの削除

DLNA-ZigBee ゲートウェイからの定期的な確認、もしくは UPnP-ZigBee ゲートウェイからの通知により、コンテンツ生成エンジンが対象とするデバイスがネットワーク内から消失したことを検出した場合、DLNA-ZigBee ゲートウェイは関係する全てのコンテンツ生成エンジンの情報を参照し、Digital Media Server に登録しているコンテンツの識別子を指定して登録情報削除を要求する。削除要求を受けた Digital Media Server は仮想 CDS から対象となるコンテンツ情報を削除し、コンテンツの公開を停止する。また、ゲートウェイは対象となるコンテンツの実体が配置されているかを調査し、コンテンツの実体が生成されている場合、コンテンツの実体を削除する。

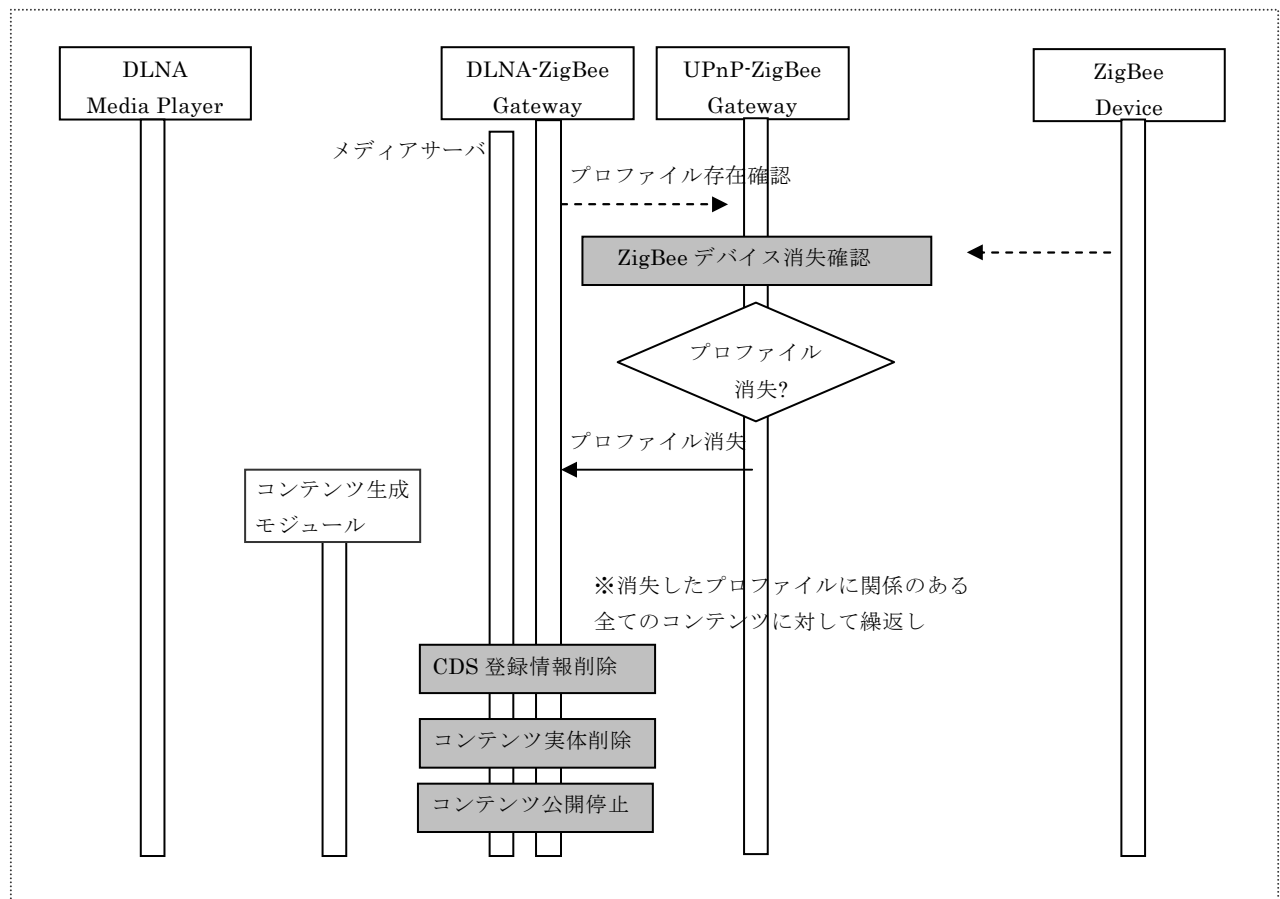


図 44. コンテンツ削除シーケンス

3.4.5. DLNA コンテンツ生成に関するコンフィギュレーション

DLNA-ZigBee ゲートウェイは ZigBee センサーの情報を視覚コンテンツなどの DLNA コンテンツに変換し、ユーザーに提供する機能であるが、コンテンツを生成する際に各センサーの情報を収集する必要が生じるため、3.3.7 節に規定された ZigBee ネットワーク上のプロフィール特性に応じたコンフィギュレーションが有効である。生成するコンテンツの総量が多い場合や ZigBee エンドデバイスの電源供給が例えば電池駆動であったりする場合は、UPnP-ZigBee ゲートウェイ上でキャッシュを行い、ZigBee ネットワークに対する情報の参照頻度を下げることが望ましい。逆に、コンテンツの総量が少ない場合や ZigBee エンドデバイスの電源が確保されている場合には、キャッシュを行わずその都度情報を取りに行くことにすると、コンテンツの信頼性を高めることができる。

ここで注意しなければならないのは、UPnP-ZigBee ゲートウェイにおけるコンフィギュレーションはゲートウェイアプリケーション全体からのアクセス頻度をコントロールするため、例えば DLNA-ZigBee ゲートウェイ上で、信頼性が重要で頻繁な更新が必要なコンテンツと、応答性が重要で

コンテンツの更新頻度は少なくてもよいコンテンツが共存している場合、細かな対応ができないことである。

また、コンテンツの要求頻度によっては、新たにコンテンツを生成するのではなく、既に生成してあるコンテンツを用いて配信を行う方が応答性や ZigBee ネットワークへのデータフレームを減らす点から望ましい場合もある。これはコンテンツを生成するのに必要なデバイス数が多い場合も同様であり、その場合は更に、リクエストを先取りして予めコンテンツを生成して用意しておく機能が有効となる。本節ではそのような問題点を踏まえ、DLNA-ZigBee ゲートウェイのオプション的な位置づけとしてのコンフィギュレーション機能を規定する。

本仕様では DLNA-ZigBee ゲートウェイのコンフィギュレーションとして、キャッシングモードとノンキャッシングモードを規定する。また、キャッシングモードにおいてはコンテンツごとのセンサー情報の取得頻度の違いを吸収するために、コンテンツ生成エンジンごとの更新間隔の設定を規定し、通常のキャッシング動作を行うモードと、コンテンツを予めバックグラウンドで更新しておくプレキャッシングモードを選択可能とする。これにより、DLNA メディアプレイヤーからのコンテンツの取得要求ごとの DLNA コンテンツ生成は図 45 のフローに従って分岐動作する。

表 23. コンフィギュレーション設定のパラメータ

Parameter	説明
DLNA Content Create Property	DLNA コンテンツを生成・配信する際の、動作モード・頻度の設定。

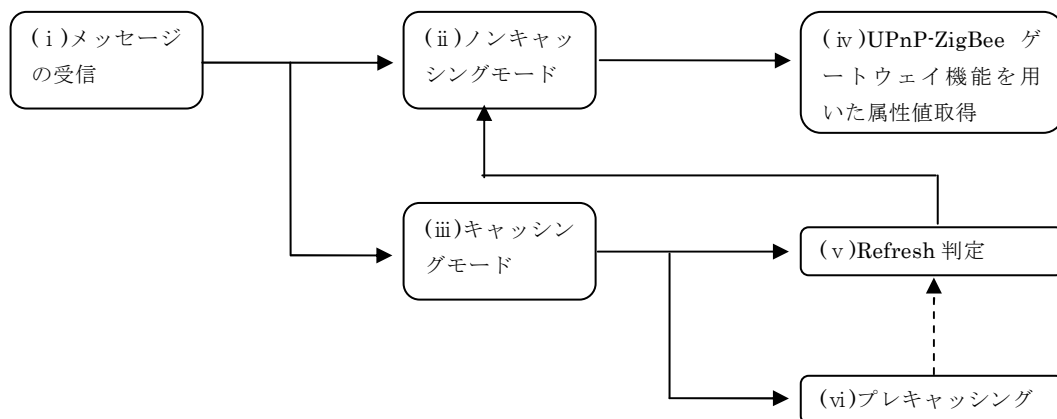


図 45. DLNA コンテンツ生成における動作モード

(i)メッセージの受信

ゲートウェイは DLNA コンテンツの取得の要求を受信すると、コンテンツの配信を行う前に、幾つかの生成動作モードに従った動作を行う。

(ii)ノンキャッシングモード

全てのコンテンツ要求に対し、対象となるコンテンツの生成を実行するモードであり、リクエストに対するコンテンツのレスポンス性は悪いが、コンテンツに反映される ZigBee センサーデータの信頼性は高くなる。ZigBee センサーの状況をリアルタイムに閲覧したい場合に有用であるが、センサーデータの最終的な信頼性は UPnP-ZigBee ゲートウェイのコンフィギュレーション設定に依存することを考慮しなければならない。DLNA-ZigBee ゲートウェイの動作がこのモードにある場合、動作は iv に遷移する。

(iii)キャッシングモード

コンテンツ要求に対し、以前に生成し、内部に保存してあるキャッシュコンテンツを参照するモードであり、リクエストに対するコンテンツのレスポンス性は良いが、コンテンツに反映される ZigBee センサーデータの信頼性は低くなる。コンテンツのキャッシュ動作は、その都度リフレッシュ判定を行う通常のキャッシュモードと、リクエストとは関係なく事前に決められた間隔ごとにコンテンツを更新するプレキャッシングモードに分かれ、コンテンツ生成エンジンごとにモード/リフレッシュレートの設定が許容される。ZigBee センサーの動作にあまり変化が見られない場合や複数のコンテンツを提供するにあたりコンテンツ内容に合わせた情報の更新を行いたい場合に有用であり、リクエスト対象となるコンテンツに指定された動作モードに従い、v または vi に遷移する。

(iv)UPnP-ZigBee ゲートウェイ機能を用いた属性値取得

コンテンツ生成にあたり、コンテンツ生成エンジンから指定されたセンサー属性の情報を、UPnP-ZigBee ゲートウェイに対し取得の要求を行う。この結果、UPnP-ZigBee ゲートウェイにおいて対象となるセンサー属性値情報を ZigBee センサーまで取得に行くか、内部に保持するキャッシュ情報から取得するかについては UPnP-ZigBee ゲートウェイのコンフィギュレーション設定に従う。取得に成功した場合、その値を元に新規コンテンツの生成を行い、生成したコンテンツをレスポンスとして配信する。

(v)リフレッシュの判定

キャッシュ情報がそのコンテンツに設定された TTL で表される閾値以下の場合、ノンキャッシングモードの振る舞いに遷移する。閾値以上の場合、ゲートウェイはキャッシュコンテンツをレスポンスとして配信する。

(vi)プレキャッシングモード

対象となるコンテンツがプレキャッシングモードにある場合、通常のキャッシングモードとは異なり、リクエスト時のリフレッシュ判定は行わない。ゲートウェイはコンテンツのリクエストに対し、内部で保持されているキャッシュコンテンツを、直ちにレスポンスとして配信する。コンテンツがこのモードに設定されている時、リクエストとは関係なしにゲートウェイはバックグラウンドでリフレッシュの判定を行い、コンテンツ情報を更新する動作を行う。このモードはノンキャッシングとキャッシングの中間に位置し、ノンキャッシングモードよりは ZigBee センサーデータの信頼性は落ちるが、通常のキャッシングモードよりも信頼性をあげることができる。常にリフレッシュ判定なしにレスポンスを行うことができるため、生成に時間のかかるコンテンツのレスポンス性を向上させたい場合に有用である。

3.4.5.1. 連続したコンテンツ閲覧要求時の生成頻度コントロール

DLNA-ZigBee ゲートウェイの動作モードに関わらず、ゲートウェイでは同一コンテンツに向けて 1 秒以下の間隔で連続して行われたコンテンツの閲覧要求に対しては、より少ない数(最低 1 つ)のコンテンツ生成要求が行われたものとしてモデュレーションをかけることが許可される。これはあまりにも短い時間でリクエストが行われた場合、コンテンツの生成元となる ZigBee センサー情報のデータが連続するリクエスト間で変化する期待値は非常に低く、生成されるコンテンツが同一になるとみなせるためであり、コンテンツ生成にかかるコスト(ZigBee 伝送路にかかる負荷、ZigBee の電力消費、コンテンツ生成時間、複数処理による遅延 etc)の軽減を図ることでパフォーマンスを確保することを目的とする。この場合、モデュレートされた各コンテンツの閲覧要求は対応するコンテンツの生成が完了するまでフックされ、その後キャッシュコンテンツを用いて応答が返される。

3.5. システムユースケース

3.5.1. 概要

本節では、ゲートウェイを用いたシステムユースケースに関する概要を記載する。

DLNA/UPnP-ZigBee ゲートウェイによって提供されるアプリケーションの基本動作は、ZigBee デバイスの操作と、ZigBee ネットワークの状況のモニタリングの二つである。

またシステムユースケースは、サービス分野に応じたものであり、対象とするサービスアプリケーションによって求められる機能やサービス品質が異なることには注意すべきである。

3.5.1.1. ゲートウェイのシステムユースケースとプロファイルの関係

ゲートウェイ自体は汎用性のある変換機能を提供するが、コンフィギュレーション仕様に関する節で述べたように、ZigBee デバイスやネットワークにプロファイルの概念があること、つまり、それによってゲートウェイに各 ZigBee プロファイル体系に応じた適切なパラメータ設定が必要となることは留意すべきことである。このプロファイルに対するゲートウェイのコンフィギュレーションの取り扱いは、現状では ZigBee 仕様でまだ明確になっていないため、本仕様書でも規定していないが、大雑把なプロファイルの概念として、ある程度のサービス分野を区分し、システムユースケースの提案と共に行う必要がある。

システムユースケースをプロファイルといった観点から考える場合に関係してくる情報として、アプリケーション特性を示すアプリケーションプロファイルと、ネットワーク特性を示すスタックプロファイルがある。ZigBee センサーネットワークはこれらのプロファイル上に成り立つネットワーク・アプリケーション環境であり、ゲートウェイはこれを壊すような振る舞いをしてはならない。

ZigBee1.0 において標準化されている ZigBee スタックプロファイルは現状では以下のように規定されている。

表 24. 標準 ZigBee スタックプロファイル

ZigBee Stack Profile Name	Stack Profile Identifier
Network Specific	0x0
Home controls	0x1
Building Automation	0x2
Plant Control	0x3
Reserved	0x4 – 0xf

3.5.1.2. 共通のユーザービリティ特性

ユーザーに対してゲートウェイは以下のようなユーザービリティを提供することが求められる。

まずユーザーは DMP を所持していれば、AV コンテンツを閲覧するのと同感覚で、家のドアや窓のロックに代表される ZigBee センサーネットワークの各センサーの属性の値、状況を知ることができる。また、ドアや窓のロック制御を行うための UPnP のサービスとアクション機能をユーザーの持つ機器に追加すれば、ゲートウェイ越しにリモート制御を行うことが可能となる。情報の収集対象となるセンサーアプリケーションのプロファイル ID と、どのようなコンテンツフォーマットに加工するかを指定することで、医療・健康系、ビル・ファクトリーオートメーション、リモート家電制御等、多分野に渡ったセンサー情報のコンテンツサービスを提供することができる。

3.5.2. ホーム・オートメーションへの適用

3.5.2.1. デバイスカテゴリ

3.5.2.1.1. 想定される DLNA, UPnP デバイスカテゴリ

ユーザーが所持する DLNA Digital Media Player として、テレビ、パソコン、QVGA サイズの解像度を持った携帯電話が考えられる。UPnP 端末としては Windows デスクトップパソコンや大型のテレビが考えられる。

3.5.2.1.2. 想定される ZigBee デバイスカテゴリ

ホームネットワークを構成する ZigBee デバイスとしては、照明スイッチや照明ライトといったものが一般に考えられる。それ以外には扉の開閉スイッチや窓の戸締り状況を示す防犯のためのデバイスが考えられる。

3.5.2.2. ゲートウェイが提供するサービス

ホーム・オートメーションにおけるサービスとしては、照明機器等の家電操作、室温センサーを含む温度調整器具、ホームセキュリティがある。ゲートウェイでは、これらの ZigBee サービスを UPnP 端末から操作できるサービス機能と、状況を DLNA コンテンツとして把握できるサービス機能を提供することができる。

3.5.2.3. 関連する ZigBee Stack Profile

関連する ZigBee スタックプロファイルとして Home Controls スタックプロファイルが挙げられる。Home Controls スタックプロファイルは以下のようなパラメータ設定となっている。

表 25. "Home Controls Stack Profile" Network Setting

Parameter Name	Setting
Beacon Order	0x0f (no beacon)
Superframe Order	0x0f (ignored)
NwkMaxDepth	5
nwkMaxChildren	20
NwkMaxRouters	6
Size of the routing table (minimum)	8
Size of the neighbor table (minimum)	ZigBee coordinator : 24 ZigBee router : 25 ZigBee end device : 1
Size of the route discovery table (minimum)	4
Number of reserved routing table entries (minimum)	8
Number of packets buffered pending route discovery (minimum)	0
Number of packet buffered on behalf of enddevices (minimum)	1
Routing cost calculation	True
NwkSymLink	False

表 26. "Home Controls Stack Profile" Application Setting

Parameter Name	Setting
Logical device type	ZigBee Coordinator -1 ZigBee Router – no more than 6 per coordinator/router to join at the higher level of the tree ZigBee End Device – no more than 20 per coordinator/router
Stack profile and beacon payload parameters	Stack profile -0x1 NwkProtocolVersion – 0x0 NwkSecurityLevel – 0x5
Number of active endpoints per device (minimum)	3
Discovery information cache size (minimum, coordinator/routers only, per sleeping child devices)	1036 bytes
Binding table size (minimum, coordinator only)	100 entries (1900 bytes)
End to end response messaging (per cluster/attribute per profile)	Agreed to for each cluster in each profile within Home Controls)
Acknowledged service in APS	Agreed to for each cluster in each profile within Home Controls.

3.5.2.4. 推奨されるゲートウェイのコンフィギュレーション設定

Home Controls Stack Profile に基づいたホーム・オートメーションサービスでは、他のスタックプロファイルと比較して、小規模なネットワーク構成となり、スター型のトポロジが形成されやすい環境にある。またデバイスの特性として、アプライアンス内に組み込まれることが多く、メンテナンス性とあいまって、ホーム・オートメーションにおいては、各センサーデバイスのバッテリー消費が問題となるため、ZigBee センサーネットワーク上のデバイスが長時間動作可能であることが要求される。反対にリアルタイム性はそれほど必要ではなく、頻繁なパケットのやりとりによって実現されるような情報の信頼性や、パケットの応答性、遅延の無さ、頻繁なネットワークからの離脱や参加から生じるディスカバリー動作への対応は要求されない。

表 27. "Home Automation" Gateway Configuration Setting

Parameter	Setting
Discovery Protocol Transparency	ZigBee 側へ検索要求は透過しない。デバイスの発見動作はゲートウェイ初期化時のみとする。
Eventing Protocol Transparency	頻繁な GENA notify に関してはモデュレーションを利用可能とする。
Control Protocol Transparency	SOAP リクエストはそのまま透過する。
Property Search	DLNA コンテンツを生成するためのプロパティ参照はオンデマンド(ノンキャッシング)で行う。
DLNA Content Create Property	DLNA コンテンツの生成はオンデマンドで行う。ただし、頻繁なコンテンツの閲覧要求に関してはコンテンツ生成のモデュレーションを利用可能とする。

3.5.3. ビルディング・オートメーションへの適用

3.5.3.1. デバイスカテゴリ

3.5.3.1.1. 想定される DLNA, UPnP デバイスカテゴリ

ビルの屋内ネットワークを構成する DLNA, UPnP デバイスとしては、管理サーバーがあげられる。

3.5.3.1.2. 想定される ZigBee デバイスカテゴリ

ビルの屋内ネットワークを構成する ZigBee デバイスとしては、照明スイッチ、照明ライトや室温センサーを含む温度調整器具といったものが一般に考えられる。それ以外には扉の開閉スイッチや窓の戸締り状況を示す防犯のためのデバイスや、入室管理センサー等も考えられる。

3.5.3.2. ゲートウェイが提供するサービス

ビルディング・オートメーションにおけるサービスとしては、照明機器等の家電操作、室温センサーを含む温度調整器具、ビルセキュリティがある。ゲートウェイでは、これらの ZigBee サービスを UPnP 端末から操作できるサービス機能と、状況を DLNA コンテンツとして把握できるサービス機能を提供することができる。また、入室管理センサー機器との連携も考えられ、ホーム・オートメーションと比較して、DLNA/UPnP ネットワーク、ZigBee ネットワーク双方で、相対的に距離的に広範囲なネットワークを形成することが多い。

3.5.3.3. 関連する ZigBee Stack Profile

関連する ZigBee スタックプロファイルとして Building Automation Stack Profile が挙げられる。Building Automation Stack Profile は以下のようなパラメータ設定となっている。

表 28. "Building Automation Stack Profile" Network Setting

Parameter Name	Setting
Beacon Order	0x0f (no beacon)
Superframe Order	0x0f (ignored)
NwkMaxDepth	9 (2 nd choice:7)
nwkMaxChildren	6 (2 nd choice:12)
NwkMaxRouters	3 (2 nd choice :4)
Size of the routing table (minimum)	16
Size of the neighbor table (minimum)	ZigBee coordinator : 15 ZigBee router : 16 ZigBee end device : 1
Size of the route discovery table (minimum)	8
Number of reserved routing table entries (minimum)	8
Number of packets buffered pending route discovery (minimum)	0
Number of packets buffered on behalf of end devices (minimum)	1
Routing cost calculation	True
NwkSymLink	False

表 29. "Home Contorols Stack Profile" Application Setting

Parameter Name	Setting
Logical device type	ZigBee Coordinator -1 ZigBee Router – no more than 3 per coordinator/router to join at the higher level of the tree ZigBee End Device – no more than 9 per coordinator/router
Stack profile and beacon payload parameters	Stack profile -0x2 NwkProtocolVersion – 0x0 NwkSecurityLevel – 0x6
Number of active endpoints per device (minimum)	7
Discovery information cache size (minimum, coordinator/routers only, per sleeping child devices)	588 bytes
Binding table size (minimum, coordinator only)	50 entries (950 bytes)
End to end response messaging (per cluster/attribute per profile)	Agreed to for each cluster in each profile within Home Controls)
Acknowledged service in APS	Agreed to for each cluster in each profile within Home controls.

3.5.3.4. 推奨されるゲートウェイのコンフィギュレーション設定

Building Automation Stack Profile に基づいたビルディング・オートメーションサービスでは、ホーム・オートメーションと比較して、ホップ数は長い、接続子デバイス数は少ないという広範囲に広がるネットワーク構成となり、ツリーやメッシュ型のトポロジが形成されやすい環境にある。またデバイスの特性として、ビル屋内のインフラとして静的に設置されていることが多く、各センサーデバイスのバッテリー消費は電池駆動よりも安定した電源供給を受けていることが多い。リアルタイム性はそれほど必要ではないが、デバイス数からネットワーク上のトラフィックは多い。一方、パケットのやりとりによって実現されるような情報の信頼性や、パケットの応答性、遅延の無さ、頻繁なネットワークからの離脱や参加から生じるディスカバリー動作への対応は要求されない。

表 30. "Building Automation" Gateway Configuration Setting

Parameter	Setting
Discovery Protocol Transparency	ゲートウェイの初期化時のみに行い、原則透過しない。
Eventing Protocol Transparency	頻繁な GENA notify に関してはモデュレーションを利用可能とする。
Control Protocol Transparency	SOAP リクエストはそのまま透過する。
Property Search	DLNA コンテンツを生成するためのプロパティ参照はオンデマンドで行う。
DLNA Content Create Property	一般に参照しなければならないデバイス数が多くなるため、DLNA コンテンツの生成はプレキャッシングで行う。

3.5.4. 医療・健康系サービスへの適用例

3.5.4.1. デバイスカテゴリ

3.5.4.1.1. 想定される DLNA, UPnP デバイスカテゴリ

医療・健康系サービスネットワークを構成する DLNA, UPnP デバイスとしては、テレビやパソコン、ケータイ電話等、管理用のサーバーが上げられる。

3.5.4.1.2. 想定される ZigBee デバイスカテゴリ

医療・健康系サービスネットワークを構成する ZigBee デバイスとしては、体温、血圧、心拍数といった測定器具等が考えられ、これらの器具は連続的に数値を計測する特徴がある。

3.5.4.2. ゲートウェイが提供するサービス

医療・健康系サービスネットワークにおけるサービスとしては、体温、血圧、心拍数といった測定器具から送られるデータを連続的に、PC等に送信しモニタリングするサービスが考えられる。PCに送られた情報を元に、インターネットを経由した遠隔カウンセリング等がある。

3.5.4.3. 関連する ZigBee Stack Profile

医療・健康分野に直結する ZigBee Stack Profile はないが、ネットワーク規模に応じて使い分けることができる。具体的には、宅内における医療・健康器具への適用を行うのであれば、ネットワーク規模はそれほど大きくある必要はなく、ZigBee Home Controls Profile が利用できる。また、病院といった施設である場合、比較的広範囲なネットワークを敷設する必要が生じるため、Building Automation Stack Profile に基づいた構成となる。

3.5.4.4. 推奨されるゲートウェイのコンフィギュレーション設定

医療健康系のサービスでは、医療情報の連続したイベントをサポートする必要がある。そのため、そのようなデバイスが発するイベント情報の透過性のレベルを、これまでのコンフィギュレーション設定と比べて上げる必要が生じる。また、イベントのモデュレーションレベルを下げることによって、ZigBee ネットワーク全体の電力消費との間にトレードオフが発生することには留意する必要がある。

表 31. "Home Health" Gateway Configuration Setting

Parameter	Setting
Discovery Protocol Transparency	定期的なデバイス探索とデバイスからのアダプタイズ要求を受け取ったタイミングで仮想デバイスを生成する。
Eventing Protocol Transparency	連続したデータのイベント発信を許可する。モデュレーションのレベルは無いが、低く設定する。
Control Protocol Transparency	SOAP リクエストはそのまま透過する。
Property Search	DLNA コンテンツを生成するためのプロパティ参照はオンデマンドで行う。
DLNA Content Create Property	DLNA コンテンツの生成はオンデマンドで行う。モデュレーションのレベルは無いが、低く設定する。

3.5.5. アミューズメントへの適用例

3.5.5.1. デバイスカテゴリ

3.5.5.1.1. 想定される DLNA, UPnP デバイスカテゴリ

ユーザーが所持するモバイル端末、ケータイ、管理サーバー、アミューズメントパークに置かれるキオスク端末などが考えられる。

3.5.5.1.2. 想定される ZigBee デバイスカテゴリ

アミューズメント施設におけるセンサーを構成するデバイスとしては、アミューズメント器具、人感センサー等が考えられる。

3.5.5.2. ゲートウェイが提供するサービス

アミューズメント施設におけるサービスとしては、施設のいたるところにセンサーが置かれているようなアミューズメントパークと、ユーザーが所持する DLNA 端末とが連携したサービスを提供することができる。

3.5.5.3. 関連する ZigBee Stack Profile

大規模パークである場合、ZigBee Stack Profile としては、Plant Control Stack が適用される。

表 32. "Plant Control Stack Profile" Network Setting

Parameter Name	Setting
Beacon Order	0x0f (no beacon)
Superframe Order	0x0f (ignored)
NwkMaxDepth	5
nwkMaxChildren	22
NwkMaxRouters	7
Size of the routing table (minimum)	16
Size of the neighbor table (minimum)	ZigBee coordinator : 30 ZigBee router : 31 ZigBee end device : 1
Size of the route discovery table (minimum)	8
Number of reserved routing table entries (minimum)	8
Number of packets buffered pending route discovery (minimum)	0
Number of packets buffered on behalf of end devices (minimum)	3
Routing cost calculation	True
NwkSymLink	False

表 33. "Plant Control Stack Profile" Application Setting

Parameter Name	Setting
Logical device type	ZigBee Coordinator -1 ZigBee Router – no more than 7 per coordinator/router to join at the higher level of the tree ZigBee End Device – no more than 22 per coordinator/router
Stack profile and beacon payload parameters	Stack profile -0x3 NwkProtocolVersion – 0x0 NwkSecurityLevel – 0x7
Number of active endpoints per device (minimum)	7
Discovery information cache size (minimum, coordinator/routers only, per sleeping child devices)	1036 bytes
Binding table size (minimum, coordinator only)	100 entries (1900 bytes)
End to end response messaging (per cluster/attribute per profile)	Agreed to for each cluster in each profile within Home Controls)

3.5.5.4. 推奨されるゲートウェイのコンフィギュレーション設定

Plant Control Stack Profile に基づいたアミューズメントサービスでは、他のスタックプロファイルと比較して、大規模なネットワーク構成となり、メッシュ型のトポロジが形成されやすい環境にある。またデバイスの特性として、キオスク端末といった施設機器に組み込まれることが想定され、デバイスの電源は AC 電源といったもので安定供給されているケースが多いと考えられる。ネットワーク規模が大きく、DLNA/UPnP 端末は来客者が個々に所持していることを想定した場合、サービス要求トラフィックが大きく跳ね上がり、ゲートウェイへの一極集中による問題が生じる可能性が高いことは留意すべきである。

表 34. "Amusement Park" Gateway Configuration Setting

Parameter	Setting
Discovery Protocol Transparency	基本的に透過しない。
Eventing Protocol Transparency	高いレベルでのモデュレーションを設定する。
Control Protocol Transparency	SOAP リクエストはそのまま透過する。
Property Search	あらかじめキャッシュされた情報を参照する。キャッシュ情報は定期的に更新する。
DLNA Content Create Property	コンテンツの更新数に対して閲覧者数が非常に多くなるため、DLNA コンテンツの生成はキャッシングで行う。また、モデュレーションのレベルは高く設定する。