TTC標準
Standard

JJ-300.10

ECHONET Lite 向け
ホームネットワーク通信インタフェース
(IEEE802.15.4/4e/4g　920MHz 帯無線)

Home network Communication Interface for ECHONET Lite
(IEEE802.15.4/4e/4g　920MHz-band Wireless)

第 2.2 版

2015 年 3 月 11 日制定

一般社団法人
情報通信技術委員会

THE TELECOMMUNICATION TECHNOLOGY COMMITTEE

本書は、一般社団法人情報通信技術委員会が著作権を保有しています。
内容の一部又は全部を一般社団法人情報通信技術委員会の許諾を得ることなく複製、転載、改変、転用及びネットワーク上での送信、配布を行うことを禁止します。

目　次

<参考>

１．　国際勧告等との関係

本標準に関連する国際標準等については、本文中に記載している。

２．　上記国際勧告等に対する追加項目等

本標準に関連する国際標準等に対するオプション選択項目、国内仕様として追加した項目、原標準に対する変更項目等については本文中に記載している。

３．　改版の履歴

版　数	改　訂　日	改　版　内　容
1	2013 年 2 月 21 日	制定
2	2014 年 2 月 20 日	方式 A に関する仕様内容の追加　(5.6　セキュリティ処理、5.7 フレームフォーマット、5.9 シングルホップスマートメーター・HEMS 間推奨通信仕様、を追加、他)
2.1	2014 年 5 月 22 日	方式 B に関し、ZigBee IP の改定に合わせてパラメータ値を修正。 (6.6.1, 6.6.2, 6.6.3, 6.7, 6.7.3, 表 6-29（旧版の表 6-31）の記述変更、および旧版の表 6-34 を削除)
2.2	2015 年 3 月 11 日	誤記訂正。(5.9.3.2.1 (3), 5.9.3.2.4 (4), 6.2.10.1, 6.3.5.1 11, 6.3.8.4)

４．　工業所有権

本標準に係る「工業所有権等の実施に係る確認書」の提出状況は TTC のホームページでご覧になれます。

５．　その他

(1) 参照する主な勧告、標準

本文中に記載する。

６．　標準作成部門

第 1 版：次世代ホームネットワークシステム専門委員会

第 2 版：次世代ホームネットワークシステム専門委員会

第 2.1 版：次世代ホームネットワークシステム専門委員会

第 2.2 版：次世代ホームネットワークシステム専門委員会

1. 標準の概要

本標準は、ECHONET Lite プロトコル[EL], [ELOBJ]を使用した家電機器の遠隔制御やモニタリング等を実現するホームネットワークを構築するためのプロトコルのうち、920MHz 特定小電力無線における仕様を規定した文書である。

2. 本標準で規定する内容

2.1. 規定の対象

ECHONET Lite を 920MHz 帯無線(IEEE802.15.4/4e/4g)の無線で利用するときには、以下の様な選択肢がある。

a. ネットワーク層プロトコルとして IPv6 ならびに 6LoWPAN を用いる

b. ECHONET Lite 電文を直接 IEEE802.15.4 フレームに載せる

表2-1: 920MHz 帯無線

<table>
<tr><th>プロトコルスタック</th><th colspan="3">プロトコル・規定</th></tr>
<tr><td>セッション～アプリケーション層</td><td colspan="2">ECHONET Lite</td><td rowspan="3">b. Layer2 のフレーム上に ECHONET Lite</td></tr>
<tr><td>トランスポート層プロトコル</td><td>UDP</td><td>TCP</td></tr>
<tr><td>ネットワーク層プロトコル</td><td colspan="2">a. IPv6 / 6LoWPAN</td></tr>
<tr><td>データリンク層プロトコル</td><td colspan="3">IEEE802.15.4, IEEE802.15.4e/g</td></tr>
<tr><td>物理層プロトコル</td><td colspan="3">IEEE802.15.4, IEEE802.15.4g</td></tr>
<tr><td>媒体</td><td colspan="3">電波(920MHz 帯)</td></tr>
</table>

本標準のスコープは、a および b であり、a には、方式 A、方式 B の 2 方式が、b には、方式 C の 1 方式がある。

2.2. 各方式の概要

本標準では、以下の 3 つの方式を規定する。

表2-2:本標準で規定する 3 方式

<table>
<tr><th>方式</th><th>表 1 における選択肢</th><th colspan="2">関連する団体</th></tr>
<tr><td>方式 A</td><td>a</td><td rowspan="3">エコーネットコンソーシアム</td><td>Wi-SUN Alliance</td></tr>
<tr><td>方式 B</td><td>a</td><td>ZigBee Alliance</td></tr>
<tr><td>方式 C</td><td>b</td><td>Wi-SUN Alliance</td></tr>
</table>

方式 A、方式 B は、物理層、データリンク層(IEEE802.15.4/4e/4g)の上に、IPv6/6LoWPAN、UDP 層（およびオプションとして TCP 層）を設けて ECHONET Lite の電文を載せる。ここで方式 A はシングルホップを提供し、方式 B はシングルホップに加えマルチホップ機能を提供する。

方式 C は、物理層、データリンク層(IEEE802.15.4/4e/4g)の上に、直接 ECHONET の電文を載せるものであり、シングルホップを提供し、マルチホップ機能は提供しない。

3. 参照規格・参考文献

本標準が規定する仕様の一部を構成する内容を含む規格および関連する規格を以下に示す。

参照規格・参考文献について改訂があった場合は、本標準に基づく実装は改訂後の最新版を適用することを推奨する。他の参照規格については、その限りではない。

[6LOWPAN]	Transmission of IPv6 Packets over IEEE 802.15.4 Networks (6LoWPAN), IETF RFC 4944
[6LPHC]	Compression Format for IPv6 Datagrams in 6LoWPAN Networks, IETF RFC 6282
[6LPND]	Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs), IETF RFC 6775
[802.15.4]	IEEE Std. 802.15.4 - 2011™ , IEEE Standard for Information Technology - Telecommunications and Information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), September 2011
[802.15.4e]	IEEE Std. 802.15.4e-2012™ , Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) - Amendment 1: MAC sub-layer, April 2012.
[802.15.4g]	IEEE Std. 802.15.4g-2012™ , Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) - Amendment 3: Physical Layer (PHY) Specifications for Low-Data-Rate, Wireless, Smart Metering Utility Networks, April 2012.
[T108]	ARIB STD-T108 920MHz 帯テレメータ用、テレコントロール用及びデータ伝送用無線設備
[AES-CCM]	NIST SP800-38C
[AES-GCM]	NIST SP800-38D
[AH]	IP Authentication Header, IETF RFC 4302
[CMAC]	NIST SP800-38B
[EL]	The ECHONET Lite Specification Version 1.01
[ELOBJ]	ECHONET Specification APPENDIX: ECHONET 機器オブジェクト詳細規定 Release B
[EAP]	Extensible Authentication Protocol (EAP), IETF RFC 3748
[EAP-PSK]	The EAP-PSK Protocol: A Pre-Shared Key Extensible Authentication Protocol (EAP) Method, IETF RFC 4764
[EAP-TLS]	The EAP-TLS Authentication Protocol, IETF RFC 5216
[ESP]	IP Encapsulating Security Payload (ESP), IETF RFC 4303
[HMAC-SHA256]	Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec, IETF RFC 4868
[IPv6]	Internet Protocol, Version 6 (IPv6) Specification, IETF RFC 2460

[IPv6-DHCP]	"IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6, IETF RFC 3633
[IPv6-MIB]	Management Information Base for IP Version 6: ICMPv6 Group, IETF RFC 2466
[IPv6-RH]	Deprecation of Type 0 Routing Headers in IPv6, IETF RFC 5095
[IPv6-SAA]	IPv6 Stateless Address Autoconfiguration, IETF RFC 2462
[ICMP6]	Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, IETF RFC 4443
[IP6ADDR]	IP Version 6 Addressing Architecture, IETF RFC 4291
[MLE]	Mesh Link Establishment, IETF draft-kelsey-intarea-mesh-link-establishment-03
[NAI]	The Network Access Identifier, IETF RFC 4282
[ND]	Neighbor Discovery for IP version 6 (IPv6), IETF RFC 4861
[PANA]	Protocol for Carrying Authentication for Network Access (PANA), IETF RFC 5191
[PANA-RELAY]	Protocol for Carrying Authentication for Network Access (PANA) Relay Element, IETF RFC 6345
[PANA-ENC]	Encrypting PANA AVPs, IETF RFC6786
[RPL]	RPL: IPv6 Routing Protocol for Low power and Lossy Networks, IETF RFC 6550
[RPL-HDR]	An IPv6 Routing Header for Source Routes with RPL, IETF RFC 6554
[RPL-OPT]	RPL Option for Carrying RPL Information in Data-Plane Datagrams, IETF RFC 6553
[RPL-MRHOF]	The Minimum Rank with Hystersis Objective Function, IETF RFC6719
[SE-TRD]	ZigBee document 095449, ZigBee SmartEnergy Profile 2.0 Technical Requirements
[SLAAC]	IPv6 Stateless Address Autoconfiguration, IETF RFC 4862
[SMHEMSIF]	ECHONET CONSORTIUM, スマート電力量メータ・HEMS コントローラ間アプリケーション通信インタフェース仕様書 Version 1.00
[TCP]	Transmission Control Protocol (TCP), IETF RFC 793 [TLS]The Transport Layer Security (TLS) Protocol Version 1.2, IETF RFC 5246
[TLS-PSK]	Pre-Shared Key Ciphersuites for Transport Layer Security (TLS), IETF RFC 4279
[TLS-ECC]	Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS), IETF RFC 4492
[TLS-AEAD]	An Interface and Algorithms for Authenticated Encryption, IETF RFC 5116
[TLS-GCM]	AES Galois Counter Mode (GCM) Cipher Suites for TLS, IETF RFC 5288
[TLS-PSK-GCM]	Pre-Shared Key Cipher Suites for TLS with SHA-256/384 and AES Galois Counter Mode, IETF RFC 5487
[TLS-ECC-GCM]	TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM), IETF RFC 5289

[TLS-CCM] AES-CCM Cipher Suites for TLS, IETF draft-mcgrew-tls-aes-ccm-04

[TLS-ECC-CCM] AES-CCM ECC Cipher Suites for TLS, IETF draft-mcgrew-tls-aes-ccm-ecc-02

[TTC TR-1043] ホームネットワーク通信インタフェース実装ガイドライン

[TRKL-MCAST] Multicast Forwarding Using Trickle, IETF draft-ietf-roll-trickle-mcast-00

[UDP] User Datagram Protocol (UDP), IETF RFC 768

[ULA] Unique Local IPv6 Unicast Addresses, IETF RFC 4193

[Wi-SUN-PHY] Wi-SUN PHY specification document for ECHONET Lite, 20120212-PHYWG-Echonet-Profile-0v01

[Wi-SUN-MAC] WI-SUN MAC specification document for ECHONET Lite, 20120212-MACWG-Echonet-Profile-0v01

[Wi-SUN-IF] WI-SUN Interface specification document for ECHONET Lite, 20131023-Wi-SUN-Echonet-Profile-2v01

[Wi-SUN-CTEST] Wi-SUN conformance test specification for ECHONET Lite

[Wi-SUN-ITEST] Wi-SUN interoperability test specification for ECHONET Lite

[ZIP] ZigBee Internet Protocol Specification 1.0, ZigBee Alliance Document

4. 定義・略語

4.1. 定義

6LBR

[6LPND]で定義される。

6LR

[6LPND]で定義される。

Authentication Server(認証サーバ)

ネットワークアクセスサービスを要求する PaC の証明書を認証することに責任を持つサーバ実装をされたサーバ。AS は PaC の代わりに PAA からの要求を受け取り、認証の結果を応答する。このサーバが EAP と EAP メソッドを完結する。AS は PAA と同じノードに搭載されても良い。またはアクセス・ネットワーク上の専用ノード、あるいはインターネット上の中央サーバにあっても良い。

Border router (ボーダー・ルータ)

それ自身に送られたのではなく異なったルーティング・ドメインにパケットを転送するルータノード。

Coordinator (コーディネータ)

本標準に規定されるノードにより構成されるネットワークを開始し保守するノード。このノードは[802.15.4]が規定する PAN コーディネータである。IP レベルでのルータとしての機能は持たなくてもよい。「親機」と記載することもある。[802.15.4]が規定するコーディネータと異なりデータリンク層のみではなくシステム全体としてのコントローラ機能を持つノードを意味する。

Enforcement Point (エンフォースメント・ポイント)

アクセス制御実装。他によってアクセスが妨げられている認証されたクライアントのアクセス(データ・トラフィック)の許可を管理する。

Global address (グローバルアドレス)

[SLAAC]で定義される。

Link local address (リンクローカルアドレス)

[SLAAC]で定義される。

Host (ホスト)

コーディネータ、もしくはルータではないノード。「子機」と記載することもある。

Node (ノード)

本標準に規定されるプロトコルを実装したノード。

PAN

パーソナル・エリア・ネットワーク。[802.15.4]参照。

Router (ルータ)

それ自身に送られたのではないネットワーク層パケットを転送するノード。

RPL

IETF RFC 6550 で規定された IPv6 のルーティング・プロトコル。

RPL Instance(RPL インスタンス)

[RPL]で定義される。

RPL Root (RPL ルート)

[RPL]で定義される。

ZIP

ZigBee IP の略称。

ZIP Coordinator (ZIP コーディネータ)

ZigBee IP ネットワークを開始し保守する ZigBee IP ノード。このノードは、MAC PAN コーディネータ、

6LoWPAN LBR ルート、PANA 認証エージェント、EAP サーバ機能を実装する。

ZIP Router (ZIP ルータ)

それ自身に送られたのではないネットワーク層パケットを転送する ZigBee IP ノード。

ZIP Host (ZIP ホスト)

ZIP ルータではない ZigBee IP ノード

ZIP Node (ZIP ノード)

本標準にて規定されるプロトコル・スートを実装するデバイス。

シングルホップ

中継機によるパケットのフォワーディングが存在せず、送信機器と受信機器の間で直接通信を行う通信形態。

マルチホップ

送信機器と受信機器の間にルータが存在することがあり、ルータによりパケットのフォワーディングが実施される可能性がある通信形態。

4.2. 略語

AES	Advanced Encryption Standard
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
DAD	Duplicate address detection. An algorithm used to ensure the uniqueness of an address in an IP network. See [6LPND]
DAG	Directed Acyclic Graph. See [RPL]
DODAG	Destination Oriented DAG. See [RPL]
EAP	Extensible Authentication Protocol. See [EAP]
EUI	Extended Unique Identifier. See [802.15.4]
FFD	Full Function Device. See [802.15.4]
ETX	Expected Transmission Count. See RFC 6551
IETF	Internet Engineering Task Force
IEEE	Institute of Electrical and Electronic Engineers
MAC	Medium Access Control
OCP	Objective Code Point. See [RPL]
OF	Objective Function. See [RPL]
ND	Neighbor Discovery, 近隣探索
PAA	PANA Authentication Agent. See [PANA]
PaC	PANA Client. See [PANA]
PRE	PANA Relay Element. See [PANA-RELAY]
RFD	Reduced Function Device [802.15.4]
ULA	Unique Local Address. See RFC 4193
UDP	User Datagram Protocol [UDP]

4.3. 表現の定義

「でなければならない」(MUST, SHALL)、「してはならない」(MUST NOT, SHALL NOT)、「要求される」(REQUIRED)、「すべきである」(SHOULD)、「すべきではない」(SHOULD NOT)、「してもよい」(MAY)などの各キーワードは、RFC2119 における定義のとおりに解釈される。

5. 方式 A

5.1. 概要

本章では、コーディネータとホスト間で IP と IEEE802.15.4/4e/4g を利用した ECHONET Lite 通信に必要となる、物理層部、データリンク層部、インタフェース部について定義する(5.3,5.4,5.5)とともに、EHONET Lite を用いてシングルホップネットワークを構成する場合の推奨仕様を規定する(5.8)。

物理層部、データリンク層部は、IEEE802.15.4/4e/4g 規格の中で選択された機能で構成されている。一方、インタフェース部は、主に、アダプテーション層、ネットワーク層、トランスポート層からなり、ECHONET Lite アプリケーション部からの送信データをデータリンク層、物理層を使用して相手デバイスに送信し、相手装置から受信データを ECHONET Lite アプリケーション部に通知する。**図 5-1**に各部の位置づけを示す。なお、本章において、”M”は標準化ドキュメント[802.15.4], [802.15.4e], [802.15.4g]として必須機能(マンダトリ)を意味し、”O”はオプション機能、”Y”は ECHONET Lite を動作させる上で必要性がある機能、“N”は必要性がない機能を示している。適合試験仕様、手順および相互接続試験仕様、手順は[Wi-SUN-PHY], [Wi-SUN-MAC], [Wi-SUN-IF], [Wi-SUN-CTEST], [Wi-SUN-ITEST]によって提供される。

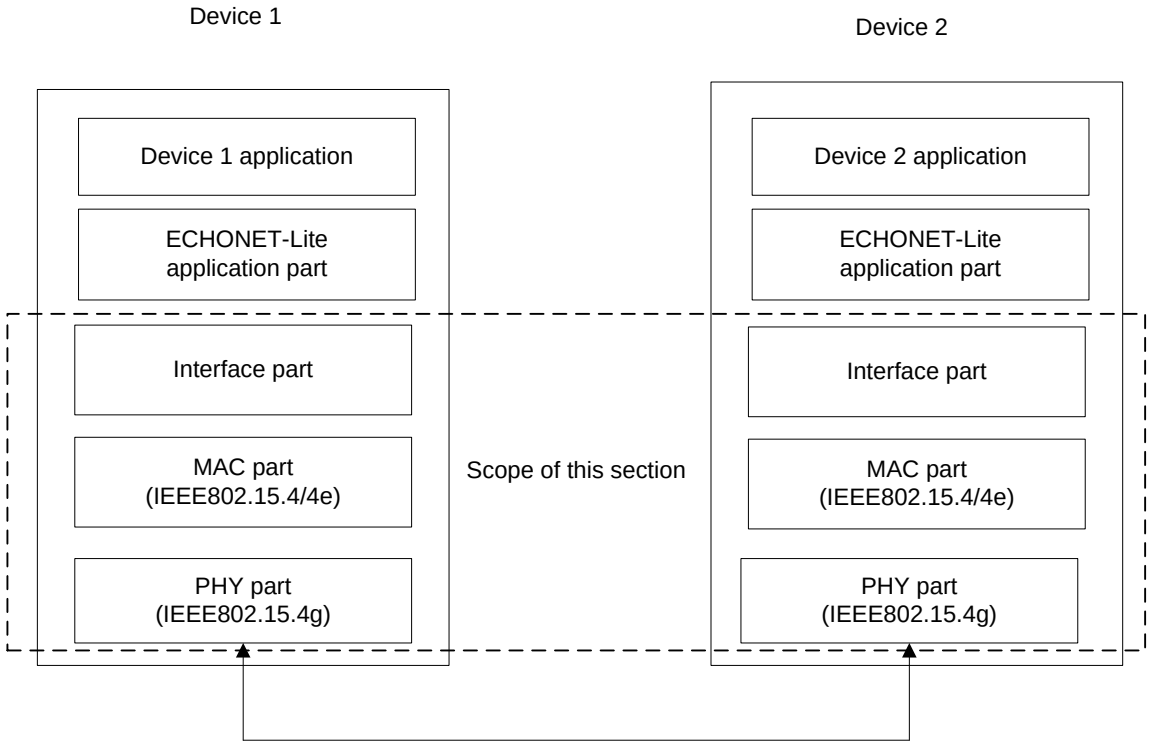


図5-1：本章で対象とする範囲

5.2. プロトコルスタック

本方式が規定するノードが搭載するプロトコルスタックは**図 5-2**のようになる。

物理層は、本方式で使用する範囲内では次のサービスを提供する。

- 最大 2047 オクテットの PSDU の転送(ただし、システムとしては後述するように 255 オクテット以下を推奨)

データリンク層は、本方式で使用する範囲内では次のサービスを提供する。

- 無線到達距離内における、IEEE802.15.4 PAN の発見
- スリープ状態と起床状態を繰り返す省電力ホストのサポート
- 暗号化・改ざん検出・リプレイ攻撃対策の機能を含むセキュリティ機能(鍵管理はこのレイヤで実行されていないことに注意すること)

6LoWPAN アダプテーション層は、本方式で使用する範囲内では次のサービスを提供する。

- IPv6 および UDP ヘッダのヘッダ圧縮と解凍
- データリンク層フレームの中で可能な最大ペイロードを超える IPv6 パケットのフラグメンテーションおよび再構築
- 近隣探索（ネットワーク層で行う場合は、必要ではない）

ネットワーク層は、本方式で使用する範囲内では次のサービスを提供する。

- IPv6 アドレス管理とパケット・フレーミング
- 近隣探索（アダプテーション層で行う場合は、必要ではない）
- IPv6 ステートレスアドレス自動設定、及び重複アドレス検出(DAD)
- IPv6 のパケット転送
- ICMPv6 メッセージ
- IPv6 パケットのマルチキャスト送受信

トランスポート層は、本方式で使用する範囲内では次のサービスを提供する。

- UDP による保証されていないパケットの配信サービス

アプリケーション層は、次のサービスを提供する。

- ネットワーク内の他ノードが保有する機能単位(ECHONET オブジェクト)の検出
- 他ノードが有する各種パラメータ・状態(ECHONET プロパティ)の取得
- 他ノードの各種パラメータ・状態の設定
- 自ノードが有する各種パラメータ・状態の通知

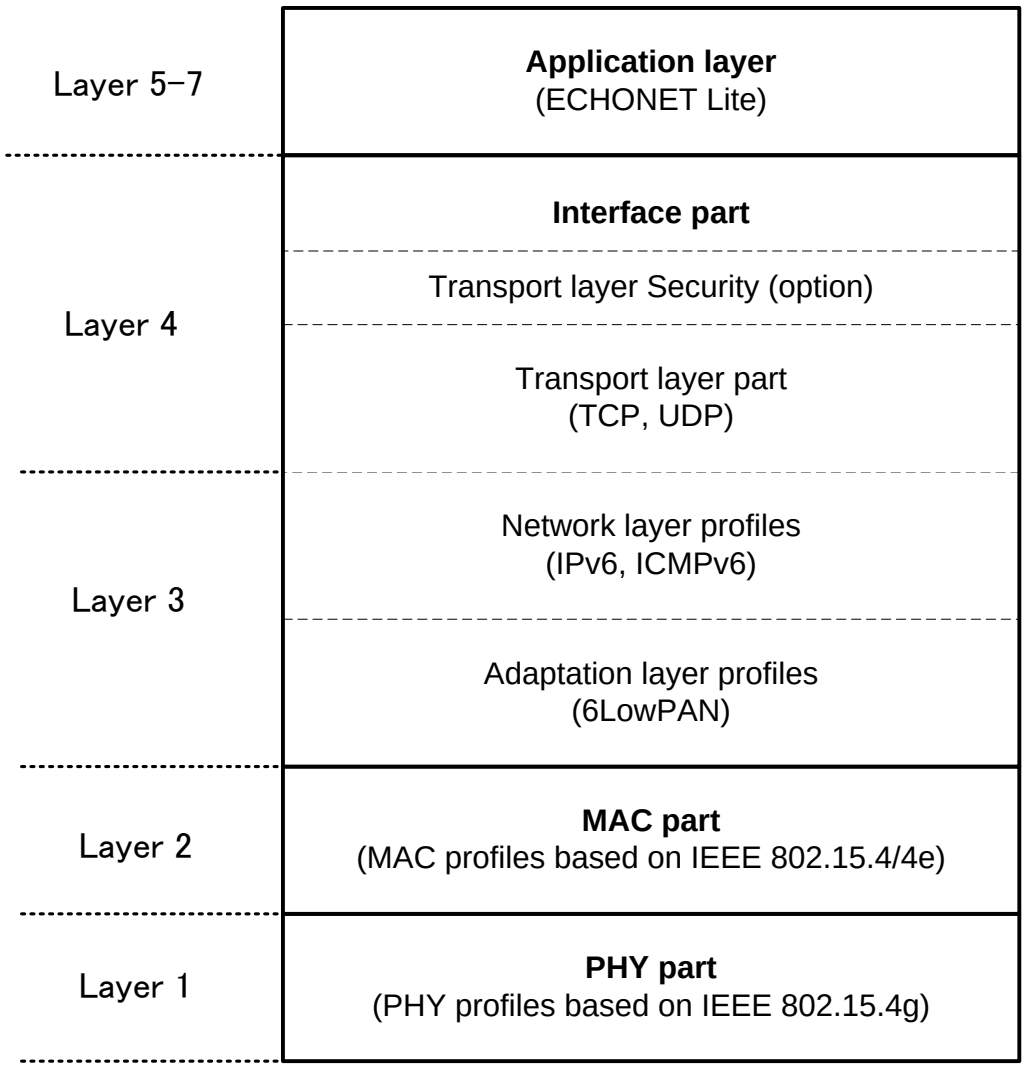


図5-2：本章で定義するレイヤ構成

5.3. 物理層部

5.3.1. 概要

本章では、ECHONET Lite をサポートするために実装上必要となる物理層部を構成するプロファイル（PHY Profile）を定義する。このプロファイルは、標準化ドキュメント[802.15.4], [802.15.4g]の中で定義された特性と機能をベースにしている。各プロファイルにおいて、標準化ドキュメント[802.15.4], [802.15.4g]の中の対応する章が記載されている。

5.3.2. 物理層プロファイル

5.3.2.1. PLF/PLP 機能

PHY Layer Function (PLF) 及び PHY Layer Packet (PLP) の必須項目を**表 5-1**に記述する。

表5-1:PLF /PLP 機能

Item number	Item description	Reference section in standard	Status in standard (M:Mandatory, O:Option)	Support (Y:Yes, N:No, O:Option)
PLF1	Energy detection (ED)	[802.15.4]8.2.5	FD1:M	FD1:Y
PLF2	Link quality indication (LQI)	[802.15.4]8.2.6	M	Y
PLF3	Channel selection	[802.15.4]8.1.2	M	Y
PLF4	Clear channel assessment (CCA)	[802.15.4]8.2.7	M	Y
PLF4.1	Mode 1	[802.15.4]8.2.7	O.2	Y
PLF4.2	Mode 2	[802.15.4]8.2.7	O.2	N
PLF4.3	Mode 3	[802.15.4]8.2.7	O.2	N
PLP1	PSDU size up to 2047 octets	[802.15.4g]9.2	FD8:M	Y

5.3.2.2. RF 機能

RF 機能に関する必須項目を**表 5-2**に記述する。

表5-2 : RF 機能

Item number	Item description	Reference section in standard	Status in standard (M:Mandatory, O:Option)	Support (Y:Yes, N:No, O:Option)
RF12	SUN PHYs			
RF12.1	MR-FSK	[802.15.4g] 18.1	FD8:M	Y(*1)
RF12.2	MR-OFDM	[802.15.4g] 18.2	FD8:O	N
RF12.3	MR-O-QPSK	[802.15.4g] 18.3	FD8:O	N
RF12.4	MR-FSK-Generic PHY	[802.15.4g] 8.1.2,10.2	RF12.1:O	N
RF12.5	Transmit and receive using CSM	[802.15.4g] 8.1a	M	Y
RF12.6	At least one of the bands given in 表 66 [802.15.4g]	[802.15.4g] 8.1	FD8:M	Y (920 MHz*2)
RF13	SUN PHY operating modes			
RF13.4	Operating mode #1 and #2 in 920 MHz or 950 MHz band	[802.15.4g] 18.1	FD8:M	Y
RF 13.5	Operating mode #3 and #4 in 920 MHz band	[802.15.4g] 18.1	FD8:O	N
RF14	MR-FSK Options			
RF14.1	MR-FSK FEC	[802.15.4g] 18.1.2.4	O	N
RF14.2	MR-FSK interleaving	[802.15.4g] 18.1.2.5	O	N
RF14.3	MR-FSK data whitening	[802.15.4g] 18.1.3	O	Y
RF14.4	MR-FSK mode switching	[802.15.4g]18.1.4	O	N

*1: The frequency tolerance requirements in [802.15.4g] 18.1.5.3 do not apply. The frequency tolerance shall be +-20ppm.

*2: All channels shown in [802.15.4g] Table 68d within the supported operating mode(s) for the respective band shall be supported.

5.4. データリンク層 （MAC 層）部

5.4.1. 概要

本方式規定に基づくコーディネータ機能を有するノードは、[802.15.4]で定義されている FFD として機能する。本節では、15.4 および 15.4e をベースにした MAC 部を構成する MAC プロファイルを定義する。それらの機能は標準化ドキュメント[802.15.4], [802.15.4e]の中から出されたもので、それらが表にまとめられている。

本方式に基づくノードは、[802.15.4]が規定する MAC レベルアドレッシングモードのうち、64bit アドレッシングモードを使用する。64 ビットの EUI-64 アドレスが製造時に各デバイスに固定的に設定されていなければならない。このアドレスはグローバルにユニークであり、デバイスに対して生涯固定であることが期待されている。

5.4.2節はビーコンモードの場合に必要となる項目が定義されており、5.4.3節はノンビーコンモードの場合に必要となる項目が定義されている。データリンク層プロファイルとしてこのいずれか一つのモードを実装しなければならない。

5.4.2. Beacon mode profile

この節は、ビーコンモードが使用された時の Wi-SUN 15.4/4e を用いた ECHONET Lite に対する MAC profile を定義している。

5.4.2.1. Functional device (FD) types

表 5-3に functional device type に関する必要項目を記述している。

表5-3:Functional device type

Item number	Item description	Reference section in standard	Status in standard (M:Mandatory, O:Option)	Support (Y:Yes, N:No, O:Option)
FD1	FFD	[802.15.4] 5.1	O.1	O.1
FD2	RFD	[802.15.4] 5.1	O.1	O.1
FD3	Support of 64 bit IEEE address	[802.15.4] 5.2.1.1.6	M	Y
FD4	Assignment of short network address (16 bit)	[802.15.4] 5.1.3.1	FD1:M	FD1:Y
FD5	Support of short network address (16 bit)	[802.15.4] 5.2.1.1.6	M	Y
FD8	SUN PHY device	[802.15.4g] 8.1	O.2	Y (#1)

O.1: Optional but at least one of the features described in FD1 and FD2 is required to be implemented

O.2: At least one of these features is supported

#1 MR-FSK is employed.

5.4.2.2. MAC sub-layer に対する主な機能

MAC sub-layer に対する主な機能を本節で記述している。

5.4.2.3. MAC sub-layer functions

MAC sub-layer function に対する必要項目を**表 5-4**にまとめる.

表5-4:MAC sub-layer function

Item number	Item description	Reference section in standard	Status in standard (M:Mandatory, O:Option)	Support (Y:Yes, N:No, O:Option)
MLF1	Transmission of data	[802.15.4] 6.3	M	Y
MLF1.1	Purge data	[802.15.4]6.3.4,6.3.5	FD1:M FD2:O	FD1:Y FD2: N
MLF2	Reception of data	[802.15.4] 6.3	M	Y
MLF2.1	Promiscuous mode	[802.15.4] 5.1.6.5	FD1:M FD2:O	FD1:Y FD2: N
MLF2.2	Control of PHY receiver	[802.15.4] 6.2.9	O	N
MLF2.3	Timestamp of incoming data	[802.15.4] 6.3.2	O	N
MLF3	Beacon management	[802.15.4] 5	M	Y
MLF3.1	Transmit beacons	[802.15.4] 5, 5.1.2.4	FD1:M FD2:O	FD1:Y FD2: N
MLF3.2	Receive beacons	[802.15.4] 5, 6.2.4	M	Y
MLF4	Channel access mechanism	[802.15.4] 5, 5.1.1	M	Y
MLF5	Guaranteed time slot (GTS) management	[802.15.4] 5, 6.2.6, 5.3.9, 5.1.7	O	N
MLF5.1	GTS management (allocation)	[802.15.4] 5, 6.2.6, 5.3.9, 5.1.7	O	N
MLF5.2	GTS management (request)	[802.15.4] 5, 6.2.6, 5.3.9, 5.1.7	O	N
MLF6	Frame validation	[802.15.4] 6.3.3, 5.2, 5.1.6.2	M	Y
MLF7	Acknowledged frame delivery	[802.15.4] 5, 6.3.3, 5.2.1.1.4, 5.1.6.4	M	Y
MLF8	Association and disassociation	[802.15.4] 5, 6.2.2, 6.2.3, 5.1.3	M	Y
MLF9	Security	[802.15.4] 7	M	Y
MLF9.1	Unsecured mode	[802.15.4] 7	M	Y
MLF9.2	Secured mode	[802.15.4] 7	O	Y
MLF9.2.1	Data encryption	[802.15.4] 7	O.4	Y
MLF 9.2.2	Frame integrity	[802.15.4] 7	O.4	Y
MLF10.1	ED	[802.15.4] 5.1.2.1, 5.1.2.1.1	FD1:M FD2:O	FD1:Y FD2: N
MLF10.2	Active scanning	[802.15.4] 5.1.2.1.2	FD1:M FD2:O	FD1:Y FD2:Y
MLF10.3	Passive scanning	[802.15.4] 5.1.2.1.2	M	Y
MLF10.4	Orphan scanning	[802.15.4] 5.1.2.1, 5.1.2.1.3	M	Y

MLF11	Control/define/determine/declare superframe structure	[802.15.4] 5.1.1.1	FD1:O	FD1:O
MLF12	Follow/use superframe structure	[802.15.4] 5.1.1.1	O	Y
MLF13	Store one transaction	[802.15.4] 5.1.5	FD1:M	FD1:Y
MLF14	Ranging	[802.15.4] 5.1.8	RF4:O	N
MLF14.1	DPS	[802.15.4] 5.1.8.3,6.2.15	O	N
MLF15(4g)	MPM for all coordinators when operating at more than 1% duty cycle	[802.15.4g] 5.1.13	M	FD8:Y
MLF15	TSCH Capability	[802.15.4e]Table 8a	O	N
MLF16	LL Capability	[802.15.4e]Table 8b	O	N
MLF17	DSME Capability	[802.15.4e] 6.2, Table 8c	O	N
MLF18	EBR capability	[802.15.4e] 5.3.12	O	Y
MLF18.1	EBR commands	[802.15.4e] 5.3.7	MLF18:O	Y
MLF18.1.1	EBR Enhanced Beacon request command	[802.15.4e] 5.3.7.2	FD1:M FD2:O	FD1:Y FD2:Y
MLF19	LE capability	[802.15.4e] 5.1.1.7, 5.1.11	O	O (#1)
MLF19.1	LE specific MAC sub-layer service specification	[802.15.4e] 6.4.3.7	MLF19:M	MLF19:Y
MLF19.2	Coordinated Sampled Listening (CSL) capability	[802.15.4e]5.1.11.1	MLF19:O.1	N
MLF19.3	Receiver Initiated Transmission (RIT) capability	[802.15.4e]5.1.11.2	MLF19:O.1	N
MLF19.4	LE superframe	[802.15.4e] 5.1.1.7.1, 5.1.1.7.2, 5.1.1.7.3	MLF19:O.1	MLF19:Y
MLF19.5	LE-multipurpose Wake-up frame	[802.15.4e]5.2.2.8	MLF19.2:M	N
MLF19.6	LE, CSL Information Element	[802.15.4e]5.2.4.7	MLF19.2:M	N
MLF19.7	LE RIT Information Element	[802.15.4e]5.2.4.8	MLF19.3:O	N
MLF19.8	LE-commands	[802.15.4e]5.3.12	MLF19.3:M	N
MLF20	MAC Metrics PIB Attributes	[802.15.4e]6.4.3.9	O	N
MLF21	FastA commands	[802.15.4e]5.1.3.3	O	N
MLF23	Channel Hopping	[802.15.4e] Table 52f	O	N
MLF23.1	Hopping IEs	[802.15.4e]5.2.4.16, 5.2.4.17	MLF18:M	N

O.1: Optional but at least one of the features described in FD1 and FD2 is required to be implemented

O.4: At least one of these features shall be supported.

#1 : Implementation is optional.

5.4.2.3.1. MAC frames

MAC frame に対する必須項目を **表 5-5**にまとめる。

表5-5:MAC frames

Item number	Item description	Reference section in standard	Status in standard (M:Mandatory, O:Option)		Support (Y:Yes, N:No, O:Option)
			Transmitter	Receiver	
MF1	Beacon	[802.15.4] 5.2.2.1	FD1:M	M	Y
MF2	Data	[802.15.4] 5.2.2.2	M	M	Y
MF3	Acknowledgment	[802.15.4] 5.2.2.3	M	M	Y
MF4	Command	[802.15.4] 5.2.2.4	M	M	Y
MF4.1	Association request	[802.15.4] 5.2.2.4, 5.3.1	M	FD1:M	Y
MF4.2	Association response	[802.15.4] 5.2.2.4, 5.3.2	FD1:M	M	Y
MF4.3	Disassociation notification	[802.15.4] 5.2.2.4, 5.3.3	M	M	Y
MF4.4	Data request	[802.15.4] 5.2.2.4, 5.3.4	M	FD1:M	Y
MF4.5	PAN identifier conflict notification	[802.15.4] 5.2.2.4, 5.3.5	M	FD1:M	Y
MF4.6	Orphaned device notification	[802.15.4] 5.2.2.4, 5.3.6	M	FD1:M	Y
MF4.7	Beacon request	[802.15.4] 5.2.2.4, 5.3.7	FD1:M	FD1:M	Y
MF4.8	Coordinator realignment	[802.15.4] 5.2.2.4, 5.3.8	FD1:M	M	Y
MF4.9	GTS request	[802.15.4] 5.2.2.4, 5.3.9	MLF5:O	MLF5:O	N
MF5	4-octet FCS	[802.15.4g] 5.2.1.9	FD8:M	FD8:M	FD8:Y

5.4.3. Non-beacon mode profile

この節は、ノンビーコンモードが使用された時の Wi-SUN 15.4/4e を用いた ECHONET Lite に対する MAC profile を定義している。

5.4.3.1. Functional device (FD) types

表 5-6に functional device type に関する必要項目を記述している。

表5-6:Functional device types

Item number	Item description	Reference section in standard	Status in standard (M:Mandatory, O:Option)	Support (Y:Yes, N:No, O:Option)
FD1	FFD	[802.15.4] 5.1	O.1	O.1
FD2	RFD	[802.15.4] 5.1	O.1	O.1
FD3	Support of 64 bit IEEE address	[802.15.4] 5.2.1.1.6	M	Y
FD4	Assignment of short network address (16 bit)	[802.15.4] 5.1.3.1	FD1:M	FD1:Y
FD5	Support of short network address (16 bit)	[802.15.4] 5.2.1.1.6	M	Y
FD8	SUN PHY device	[802.15.4g] 8.1	O.2	Y (#1)

O.1: Optional but at least one of the features described in FD1 and FD2 is required to be implemented

O.2: At least one of these features is supported

#1 : MR-FSK is employed.

5.4.3.2. MAC sub-layer に対する主な機能

MAC sub-layer に対する主な機能を本節で記述している。

5.4.3.2.1. MAC sub-layer functions

MAC sub-layer function に対する必要項目を**表 5-7**にまとめる。

表5-7:MAC sub-layer function

Item number	Item description	Reference section in standard	Status in standard (M:Mandatory, O:Option)	Support (Y:Yes, N:No, O:Option)
MLF1	Transmission of data	[802.15.4] 6.3	M	Y
MLF1.1	Purge data	[802.15.4] 6.3.4, 6.3.5	FD1:M FD2:O	FD1:Y FD2: N
MLF2	Reception of data	[802.15.4] 6.3	M	Y
MLF2.1	Promiscuous mode	[802.15.4] 5.1.6.5	FD1:M FD2:O	FD1:Y FD2: N
MLF2.2	Control of PHY receiver	[802.15.4] 6.2.9	O	O
MLF2.3	Timestamp of incoming data	[802.15.4] 6.3.2	O	N
MLF3	Beacon management	[802.15.4] 5	M	Y
MLF3.1	Transmit beacons	[802.15.4] 5, 5.1.2.4	FD1:M FD2:O	FD1:Y FD2: N
MLF3.2	Receive beacons	[802.15.4] 5, 6.2.4	M	Y
MLF4	Channel access mechanism	[802.15.4] 5, 5.1.1	M	Y
MLF5	Guaranteed time slot (GTS) management	[802.15.4] 5, 6.2.6, 5.3.9, 5.1.7	O	N
MLF5.1	GTS management (allocation)	[802.15.4] 5, 6.2.6, 5.3.9, 5.1.7	O	N
MLF5.2	GTS management (request)	[802.15.4] 5, 6.2.6, 5.3.9, 5.1.7	O	N
MLF6	Frame validation	[802.15.4] 6.3.3, 5.2, 5.1.6.2	M	Y
MLF7	Acknowledged frame delivery	[802.15.4] 5, 6.3.3, 5.2.1.1.4, 5.1.6.4	M	Y
MLF8	Association and disassociation	[802.15.4] 5, 6.2.2, 6.2.3, 5.1.3	M	Y
MLF9	Security	[802.15.4] 7	M	Y
MLF9.1	Unsecured mode	[802.15.4] 7	M	Y
MLF9.2	Secured mode	[802.15.4] 7	O	Y
MLF9.2.1	Data encryption	[802.15.4] 7	O.4	Y
MLF 9.2.2	Frame integrity	[802.15.4] 7	O.4	Y
MLF10.1	ED	[802.15.4] 5.1.2.1, 5.1.2.1.1	FD1:M FD2:O	FD1:Y FD2: N
MLF10.2	Active scanning	[802.15.4] 5.1.2.1.2	FD1:M FD2:O	FD1:Y FD2:Y
MLF10.3	Passive scanning	[802.15.4] 5.1.2.1.2	M	Y
MLF10.4	Orphan scanning	[802.15.4] 5.1.2.1, 5.1.2.1.3	M	Y

MLF11	Control/define/determine/declare superframe structure	[802.15.4] 5.1.1.1	FD1:O	N
MLF12	Follow/use superframe structure	[802.15.4] 5.1.1.1	O	N
MLF13	Store one transaction	[802.15.4] 5.1.5	FD1:M	FD1:Y
MLF14	Ranging	[802.15.4] 5.1.8	RF4:O	N
MLF14.1	DPS	[802.15.4] 5.1.8.3,6.2.15	O	N
MLF15(4g)	MPM for all coordinators when operating at more than 1% duty cycle	[802.15.4g] 5.1.13	M	Y
MLF15	TSCH Capability	[802.15.4e] Table 8a	O	N
MLF16	LL Capability	[802.15.4e] Table 8b	O	N
MLF17	DSME Capability	[802.15.4e] 6.2, Table 8c	O	N
MLF18	EBR capability	[802.15.4e] 5.3.12	O	Y
MLF18.1	EBR commands	[802.15.4e] 5.3.7	MLF18:O	Y
MLF18.1.1	EBR Enhanced Beacon request command	[802.15.4e] 5.3.7.2	FD1:M FD2:O	FD1:Y FD2:Y
MLF19	LE capability	[802.15.4e] 5.1.1.7, 5.1.11	O	O (#1)
MLF19.1	LE specific MAC sub-layer service specification	[802.15.4e] 6.4.3.7	MLF19:M	MLF19:Y
MLF19.2	Coordinated Sampled Listening (CSL) capability	[802.15.4e] 5.1.11.1	MLF19:O.1	MLF19:O.1
MLF19.3	Receiver Initiated Transmission (RIT) capability	[802.15.4e] 5.1.11.2	MLF19:O.1	MLF19:O.1
MLF19.4	LE superframe	[802.15.4e] 5.1.1.7.1, 5.1.1.7.2, 5.1.1.7.3	MLF19:O.1	N
MLF19.5	LE-multipurpose Wake-up frame	[802.15.4e] 5.2.2.8	MLF19.2:M	MLF19.2:Y
MLF19.6	LE, CSL Information Element	[802.15.4e] 5.2.4.7	MLF19.2:M	MLF19.2:Y
MLF19.7	LE RIT Information Element	[802.15.4e] 5.2.4.8	MLF19.3:O	MLF19.3:O
MLF19.8	LE-commands	[802.15.4e] 5.3.12	MLF19.3:M	MLF19.3:Y
MLF20	MAC Metrics PIB Attributes	[802.15.4e] 6.4.3.9	O	N
MLF21	FastA commands	[802.15.4e] 5.1.3.3	O	N
MLF23	Channel Hopping	[802.15.4e] Table 52f	O	N
MLF23.1	Hopping IEs	[802.15.4e] 5.2.4.16, 5.2.4.17	MLF18:M	N

O.1: Optional but at least one of the features described in FD1 and FD2 is required to be implemented

O.4: At least one of these features shall be supported.

#1 : Implementation is optional.

MAC frame に対する必須項目を **表 5-8**にまとめる。

表5-8:MAC frame

Item number	Item description	Reference section in standard	Status in standard (M:Mandatory, O:Option)		Support (Y:Yes, N:No, O:Option)
			Transmitter	Receiver	
MF1	Beacon	[802.15.4] 5.2.2.1	FD1:M	M	Y
MF2	Data	[802.15.4] 5.2.2.2	M	M	Y
MF3	Acknowledgment	[802.15.4] 5.2.2.3	M	M	Y
MF4	Command	[802.15.4] 5.2.2.4	M	M	Y
MF4.1	Association request	[802.15.4] 5.2.2.4, 5.3.1	M	FD1:M	Y
MF4.2	Association response	[802.15.4] 5.2.2.4, 5.3.2	FD1:M	M	Y
MF4.3	Disassociation notification	[802.15.4] 5.2.2.4, 5.3.3	M	M	Y
MF4.4	Data request	[802.15.4] 5.2.2.4, 5.3.4	M	FD1:M	Y
MF4.5	PAN identifier conflict notification	[802.15.4] 5.2.2.4, 5.3.5	M	FD1:M	Y
MF4.6	Orphaned device notification	[802.15.4] 5.2.2.4, 5.3.6	M	FD1:M	Y
MF4.7	Beacon request	[802.15.4] 5.2.2.4, 5.3.7	FD1:M	FD1:M	Y
MF4.8	Coordinator realignment	[802.15.4] 5.2.2.4, 5.3.8	FD1:M	M	Y
MF4.9	GTS request	[802.15.4] 5.2.2.4, 5.3.9	MLF5:O	MLF5:O	N
MF5	4-octet FCS	[802.15.4g] 5.2.1.9	FD8:M	FD8:M	O(#1)

#1: Implementation is optional.

5.5. インタフェース部

5.5.1. 概要

インタフェース部はトランスポート層、ネットワーク層、アダプテーション層から構成されなければならない。トランスポート層/ネットワーク層からの情報はアダプテーション層を経由して物理層/データリンク層のデータに変換される。一方で、物理層/データリンク層からのデータはアダプテーション層を経由してトランスポート層/ネットワーク層のデータに変換される。トランスポート層プロトコルとして UDP, TCP がサポートされてもよい。

5.5.2. 所要条件

(1) インタフェース部は Network Interfaceを供給しなければならない。Network Interface内のMACアドレスはIEEE802.15.4 MAC部で抽出されたEUI-64アドレスでなければならない。

(2) インタフェース部は、MAC部で利用しているアドレス形態を事前に知らなければならない。

(3) インタフェース部は、MAC部で利用しているアドレス形態に合わせてIPv6ヘッダを解析しなければならない。そして、IPv6ヘッダ宛先アドレスをMAC部が送信するアドレスに変換する必要がある。

(4) インタフェース部は、IPv6ヘッダを解析し、送信先アドレスがマルチキャストアドレスの場合、MAC部に対してブロードキャスト送信を指示しなければならない。

(5) インタフェース部は、IPv6もしくは6LowPANをベースにした近隣探索を用いる。この近隣探索はノードごとではなく、システムごとに選択を行う。

5.5.3. アダプテーション層

インタフェース部におけるアダプテーション層は、6LoWPAN [6LOWPAN]および6LoWPAN における IPHC [6LPHC]をサポートし、IPv6 ヘッダの圧縮および、必要に応じてフラグメント処理を実施しなければならない。6LoWPAN を用いたアダプテーション層の必須項目を**表 5-9**に示す。

表5-9:6LoWPAN のアダプテーションレイヤ

Item number	Item description	Reference section in standard	Support (Y:Yes, N:No, O:Option)
6LP1.1	Addressing Mode (EUI-64)	[6LOWPAN] 3	Y
6LP1.2	Addressing Mode (short address)	[6LOWPAN] 3	N
6LP2	Frame Format	[6LOWPAN] 5	O (#1)
6LP3	Stateless Address Autoconfiguration	[6LOWPAN] 6	Y
6LP4	IPv6 Link Local Address	[6LOWPAN] 7	Y
6LP5	Unicast Address Mapping	[6LOWPAN] 8	Y (#2)
6LP6	Multicast Address Mapping	[6LOWPAN] 9	N
6LP7	Encoding of IPv6 Header Fields	[6LOWPAN] 10.1	N (#3)
6LP8	Encoding of UDP Header Fields	[6LOWPAN] 10.2	N (#3)
6LP9	Non-Compressed Fields	[6LOWPAN] 10.3	Y
6LP10	Frame Delivery in a Link-Layer Mesh	[6LOWPAN] 11	N

(#1) Header Type = LOWPAN_HC1 は使用しない、また Header Type = LOWPAN_BC0 および[6LOWPAN]5.2 はオプション

(#2) 16bit アドレス(short address)は使用しない

(#3) ヘッダ圧縮には、[6LOWPAN]記載の HC1, HC2 ではなく IPHC[6LPHC]を使用する。

5.5.3.1. Fragmentation

[6LOWPAN]に規定されるフラグメンテーションをサポートしなければならない。実装しなければならない 6LoWPAN の Fragmentation の必須項目を**表 5-10**に示す。全てのノードは、[6LOWPAN]に規定されるフラグメンテーションをサポートしなければならない。

表5-10:6LoWPAN の Fragmentation

Item number	Item description	Reference section in standard	Support (Y:Yes, N:No, O:Option)
6LPF1	Fragmentation type and Header	[6LOWPAN] 5.3	Y

5.5.3.2. Header compression

実装しなければならない 6LoWPAN の Header compression の必須項目を**表 5-11**に示す。基本的にすべてのノードは[6LPHC]に規定されるヘッダ圧縮をサポートしなければならない。ただし、コンテキスト ID を用いたヘッダ圧縮は(ステートフルなマルチキャストアドレスの圧縮を含めて)サポートしない。また、LOWPAN_NHC による IPv6 拡張ヘッダや UDP ヘッダの圧縮はサポートしない。IPv6 パケットを受信するノードは、ヘッダ圧縮を施していない IPv6 パケット、および[6LPHC]で規定されたヘッダ圧縮のうち前記非サポート機能を用いない方法でエンコードされた IPv6 パケットを受信できなければならない。これは、[6LPHC]で規定されたヘッダ圧縮の一部のみを適用してエンコードされた IPv6 パケットも含む。

表5-11:6LoWPAN の Header Compression

Item number	Item description	Reference section in standard	Support (Y:Yes, N:No, O:Option)
6HC1.1	LOWPAN_IPHC (Base Format)	[6LPHC] 3.1.1	Y
6HC1.2	Context Identifier Extension	[6LPHC] 3.1.2	N
6HC2.1	Stateless Multicast Address Compression	[6LPHC] 3.2.3	Y
6HC2.2	Stateful Multicast Address Compression	[6LPHC] 3.2.4	N
6HC4	LOWPAN_NHC (IPv6 Extension Header Compression)	[6LPHC] 4.2	N
6HC5	LOWPAN_NHC (UDP Header Compression)	[6LPHC] 4.3	N

コンテキスト ID をサポートしないこと、並びに後述するように IPv6 アドレスとして EUI-64 アドレスに基づくリンクローカルアドレスを用いることにより、本方式に基づくノードが送信するユニキャストの IPv6 パケットの LOWPAN_IPHC encoding ヘッダ**[**6LPHC**]**は**図 5-3**のようになる。

(bit)

0	1	2	3 4	5	6 7	8	9	10	11	12	13	14	15
0	1	1	TF *1	NH *2	HLIM *3	0	0	1	1	0	0	1	1

図5-3：LOWPAN_IPHC encoding ヘッダ(ユニキャストの場合)

*1: TF = 0b11(Traffic Class and Flow Label are elided)

*2: NH = 0b0(Full 8 bits for Next Header are carried in-line)

*3: HLIM = 0b11(The Hop Limit field is compressed and the hop limit is 255)

5.5.3.3. 近隣探索

近隣探索は、基本的に IPv6 向けに定義された RFC 4861 [ND]を使用するが、6LoWPAN 向けに最適化された RFC6775 を使用してもよい。RFC6775 を使用する場合に実装しなければならない 6LoWPAN の Neighbor discovery の必須項目を**表 5-12** に示す。なお、マルチホップ機能を実現するために使用するルーティングの規定については、本仕様の対象外とする。

表5-12:6LoWPANによる近隣探索

Item number	Item description	Reference section in standard	Support (Y:Yes, N:No, O:Option)
6ND1	DHCPv6 Address Assignment for 6LBR, 6LR and Host	[6LPND] 3.2	O
6ND2	DHCPv6 Prefix Delegation for 6LBR	[6LPND] 3.2, 7.1	O
6ND3	DHCPv6 Prefix Delegation for 6LR and Host	[6LPND] 3.2, 7.1	O
6ND4	Static IPv6 address configuration on 6LBR	[6LPND] 5.4.1	O
6ND5	Static IPv6 address configuration on 6LR and Host	[6LPND]5.4.1	O
6ND6	EUI-64 based IPv6 Address Generation	[6LPND] 5.4.1	Y
6ND7	802.15.4 16-bit short address	[6LPND] 1.3	N
6ND8	802.15.4 64-bit extended address	[6LPND] 1.3	Y
6ND9	Duplicate Address Detect	[6LPND] 4.4	O
6ND10	Duplicate Address messages (DAR and DAC)	[6LPND] 4.4	O
6ND11	Support Source Link-Layer Address Option (SLLAO)	[6LPND] 4.1, 5.3	Y
6ND12	Support Address Registration Option (ARO)	[6LPND]5.5	Y
6ND13	Support Authoritative Border Router Option (ABRO)	[6LPND] 3.3, 3.4, 4.3, 6.3	O
6ND14	Support Prefix Information Option (PIO)	[6LPND]3.3, 5.4	O
6ND15	Support 6LoWPAN Context Option (6CO)	[6LPND] 4.2	O
6ND16	Multihop Prefix and Context Distribution	[6LPND] 8.1	O
6ND17	Multihop DAD	[6LPND] 8.2	O
6ND18	Support Router Discovery	[6LPND]	Y
6ND19	Support RA based Address Configuration on 6LR and Host	[6LPND] 5.4.1	O
6ND20	Support Neighbor Cache Management	[6LPND] 3.5	Y
6ND21	Support Address Registration	[6LPND] 3.2	Y
6ND22	Support Address unregistration	[6LPND] 3.2	Y
6ND23	Support Neighbor Unreachable Detection	[6LPND]5.5	Y
6ND24	Send Multicast NS	[6LPND] 6.5.5	O
6ND25	Send Unicast NS	[6LPND] 5.5	Y

5.5.4. ネットワーク層

インタフェース部におけるネットワーク層は、[IPv6]で定義するIPv6プロトコルをベースに**表5-13**に示す項目を実装しなければならない。Hop-by-Hop Options拡張ヘッダ、Routing拡張ヘッダ、Fragment拡張ヘッダ、

Destination Options 拡張ヘッダ、および IPSec に関連する AH 拡張ヘッダと ESP 拡張ヘッダはサポートしなくてもよい。なお、各拡張ヘッダは、[IPv6] 記載の推奨順序に従って送信しなければならない。

また、**表 5-14**に示す ICMPv6 [ICMPv6]をサポートしなければならない。メッセージ種別としては、エコー要求(タイプ 128)およびエコー応答(タイプ 129)に加え、宛先未到達(タイプ 1)、時間超過(タイプ 3)およびパラメータ問題(タイプ 4)の各エラーメッセージもサポートしなければならない。パケットサイズ超過(タイプ 2)メッセージに関しては、送信機能を持たなくてもよいが受信した際は適切に処理されなければならない。

表5-13:Network Layer: IPv6

Item number	Item description	Reference section in standard	Support (Y:Yes, N:No, O:Option)
IP1	Header Format	[IPv6] 3	Y
IP1.1	Extension Headers	-	Y
IP1.2	Extension Header Order	[IPv6]4.1	Y
IP1.3	Options	[IPv6] 4.2	Y
IP1.4	Hop-by-Hop Options Header	[IPv6] 4.3	O
IP1.5	Routing Header	[IPv6]4.4	O
IP1.6	Fragment Header	[IPv6] 4.5	O
IP1.7	Destination Options Header	[IPv6] 4.6	O
IP1.8	No Next Header	[IPv6]4.7	Y
IP1.9	AH Header	[IPv6-SAA]	O
IP1.10	ESP Header	[IPv6-MIB]	O
IP2	Deprecation of Type 0 Routing Headers	[IPv6-RH]	Y
IP3	Path MTU Discovery	[IPv6] 5	Y
IP4	Flow Labels	[IPv6] 6	Y
IP5	Traffic Classes	[IPv6] 7	Y

表5-14:Network Layer: ICMPv6

Item number	Item description	Reference section in standard	Support (Y:Yes, N:No, O:Option)
ICMP1	Message Format	[ICMP6] 2.1	Y
ICMP2	Message Source Address Determination	[ICMP6] 2.2	Y
ICMP3	Message Checksum Calculation	[ICMP6] 2.3	Y
ICMP4	Message Processing Rules	[ICMP6] 2.4	Y
ICMP5	Destination Unreachable Message	[ICMP6] 3.1	Y
ICMP6	Packet Too Big Message	[ICMP6] 3.2	Y
ICMP7	Time Exceeded Message	[ICMP6] 3.3	Y
ICMP8	Parameter Problem Message	[ICMP6] 3.4	Y
ICMP9	Echo Request Message	[ICMP6] 4.1	Y
ICMP10	Echo Reply Message	[ICMP6] 4.2	Y

5.5.4.1. IP addressing

文献[IP6ADDR]で規定される IPv6 addressing 及び、文献[SLAAC]で規定される IPv6 Stateless Address Autoconfiguration をベースに**表 5-15**に示す項目を実装しなければならない。本方式で定義するネットワークでは、常に EUI-64 アドレスをベースとしたリンクローカルアドレスを使用する。その際、[6LOWPAN]と[SLAAC]の記載に従い、プレフィックスとして well known link-local prefix FE80::0/64 を使用した上で、ノードの EUI-64 アドレスからインタフェース識別子を生成する。[802.15.4]が規定するショートアドレスをベースとした IPv6 リンクローカルアドレス、およびグローバルアドレスとユニークローカルアドレスは、本標準内では使用しない。

表5-15:Network Layer: IP Addressing

Item number	Item description	Reference section in standard	Support (Y:Yes, N:No, O:Option)
IPAD1	IPv6 Addressing	[IP6ADDR]	Y (#1)
IPAD1.1	Global Unicast Address	[IP6ADDR] 2.5.4	N
IPAD1.2	Link Local Unicast Address	[IP6ADDR] 2.5.6	Y (#2)
IPAD1.3	Unique Local Unicast Address	[ULA]	N
IPAD1.4	Anycast Address	[IP6ADDR] 2.6	N
IPAD1.5	Multicast Address	[IP6ADDR] 2.7	Y (#3)
IPAD1.6	Prefix Length		/64
IPAD2	Stateless Address Autoconfiguration	[SLAAC]	Y
IPAD2.1	Creation of Link Local Address	[SLAAC] 5.3	Y
IPAD2.2	Creation of Global Addresses	[SLAAC] 5.5	N

(#1) 一部機能は使用しない

(#2) MAC の EUI-64 アドレスベースの Link Local Address を利用する

(#3) 送信は ff02::1 を使用

5.5.4.2. 近隣探索

近隣探索は、IPv6 向けに定義された RFC 4861 [ND]を使用する。[ND]を使用する場合、実装しなければならない IPv6 の Neighbor discovery の必須項目を**表 5-16**に示す 。[ND]が定義する機能のうち、本方式規定に従うノードがサポートしなければならない機能は、アドレス解決、重複アドレス検出の 2 機能である。また、[ND]に定義されている ICMPv6 メッセージのうち、本方式規定に従うノードがサポートしなければならないメッセージは、近隣要請メッセージ(Neighbor Solicitation message: Type = 135)と近隣応答メッセージ(Neighbor Advertisement message: Type = 136)の 2 つである。

表5-16:Network Layer: IPv6 による近隣探索

Item number	Item description	Reference section in standard	Support (Y:Yes, N:No, O:Option)
ND1	Router and Prefix Discovery	[ND]6	N
ND2	Address Resolution	[ND] 7.2	Y
ND3	Neighbor Unreachability Detection	[ND] 7.3	N
ND4	Duplicate Address Detection	[SLAAC] 5.4	O
ND5	Redirect Function	[ND] 8	N
ND6	Router Solicitation Message	[ND]4.1	N
ND7	Router Advertisement Message	[ND] 4.2	N
ND8	Neighbor Solicitation Message	[ND] 4.3	Y(*1)
ND9	Neighbor Advertisement Message	[ND] 4.4	Y(*2)
ND10	Redirect Message	[ND] 4.5	N
ND11	Source/Target Link-layer Address Option	[ND] 4.6.1	Y
ND12	Prefix Information Option	[ND] 4.6.2	N
ND13	Redirected Header Option	[ND] 4.6.3	N
ND14	MTU Option	[ND] 4.6.4	N

*1: Source Link-Layer Address オプションには EUI-64 フォーマットのアドレスを含める

*2: Target Link-Layer Address オプションには EUI-64 フォーマットのアドレスを含める

5.5.4.3. マルチキャスト

ECHONET Lite 電文のマルチキャスト送信時は、ECHONET Lite 仕様[EL]の規定に従い ff02::1 を宛先として設定する。

5.5.5. トランスポート層

UDP[UDP] を使用するが、TCP[TCP]も使用してよい。ただし、TCP を使用する場合も UDP は常に使用可能でなければならない。UDP フレーム／TCP フレームの宛先ポート番号や TCP 使用時の規定は [EL]記載の内容に従わなければならない。

5.5.6. アプリケーション層

アプリケーション層としては、ECHONET Lite [EL]を使用する。本方式記載の仕様に基づくノードは、[EL]に規定される必須機能をすべてサポートしなければならない。

5.6. セキュリティ処理

5.6.1. 概要

本仕様では、通信セキュリティとしてPANAによるネットワーク接続認証及びMAC層による通信の保護（暗号化）を実施する。また、PANAが使用するEAPメソッドとしてEAP-PSKを採用し、MAC層の暗号化アルゴリズムは[802.15.4]に記載されるAES-128-CCM*を使用する。

5.6.2. 認証

本仕様ではコーディネータがPAAとなり、ホストがPaCとなる。

5.6.2.1. PANA

- インターネットプロトコルはバージョン6(IPv6)及びUDPを使用する
- PAAのIPアドレスはPaCによるPANAセッション開始時には既知であるとする
- PAA/PaCが用いる宛先ポート番号は716(PANAデフォルト値)とする
- PaCによるPANAセッション起動のみをサポートする(PAAによるPANAセッション起動はサポートしない)
- 鍵導出アルゴリズム(PRF-Algorithm)にはPRF_HMAC_SHA2_256(AVP Value=5)を使用する
- メッセージ認証アルゴリズム(Integrity-Algorithm)にはAUTH_HMAC_SHA2_256_128(AVP Value=12)を使用する
- EAP-Responseメッセージは必ずPANA-Auth-Answerメッセージに含める(Piggyback)
- Nonce値のサイズは16オクテットとする
- ライフタイム値は符号無し4オクテットの幅を持つ秒数で指定可能であるが、60秒よりも少ない値を指定してはならない

5.6.2.2. EAP

- EAP認証メソッドとして、共通鍵ベースのEAP-PSKを使用する
- EAP-PSKの認証鍵のサイズは16オクテットとする
- EAPレイヤからPANAプロトコルレイヤに渡すMSK(Master Session Key)、EMSK(Extended Master Session Key)の鍵サイズは64オクテットとする
- サーバ側認証子であるEAP ID_Sは[NAI]で規定されるNAIとする

本仕様においては、NAIの長さは63オクテットを超えないこととする[1]。

- クライアント側認証子であるEAP ID_Pは[NAI]で規定されるNAIとする

本仕様においては、NAIの長さは63オクテットを超えないこととする。

- EAPレイヤでのメッセージ再送は無効とする

5.6.3. 鍵更新

PANA自身を保護するために使用される鍵(PANA_AUTH_KEY)、及びPANAによる接続認証の成功の結果コーディネータとホスト間で共有されるMAC層で使用する鍵の有効期限はPANAセッションのライフタイムと同一とする。PANAセッションが更新（Re-Authenticationフェーズによる PANAセッションの更新、もしくはAuthentication and Authorizationフェーズによる PANAセッションの新規確立）された場合、新規に導出される鍵を使用する。また、PANAセッションがライフタイム中途で終了した場合においては、その時点で導出された鍵を無効とすること。

---

[1] RFC4282 2.2によるとRADIUSの制限内に収める必要があるため。

5.6.3.1. PANA 鍵導出関数

PANA メッセージの完全性を担保する AUTH AVP を生成するために必要となる PANA_AUTH_KEY は[PANA]に従うが、prf()関数は PRF-HMAC-SHA-256 を使用する。

生成された PANA_AUTH_KEY を用いて AUTH AVP の値を導出する際に使用する PANA_AUTH_HASH()関数は Integrity-Algorithm AVP によってネゴシエートされたハッシュ関数であるが、本仕様では AUTH_HMAC_SHA_256_128 を使用する。

5.6.3.2. EAP-PSK 鍵導出関数

EAP-PSK のネゴシエーションによって生成する TEK(16 オクテット)、MSK(64 オクテット長)、EMSK(64 オクテット長)の導出は[EAP-PSK]に従う。

5.6.3.3. MAC 層鍵導出関数

MAC 層で使用するセキュリティ鍵は EAP-PSK のネゴシエーションの結果導出される EMSK を用いて導出する。まず MAC 層用鍵を生成するためのマスター鍵 SMMK を USRK 導出関数[USRK]より生成し、SMMK を使用して機器間の MAC 層鍵 SMK-HH を導出する。

```
SMMK = KDF(EMSK, “Wi-SUN JP SH-HAN” | “¥0” | optional data | length)
•    optional data = NULL(0x00)
•    length = 64

SMK-HH = KDF(SMMK, “Wi-SUN JP SH-HAN” | “¥0” | optional data | length)
•       optional data = EAP ID_P | EAP ID_S | IEEE802.15.4 Key Index
•       length = 16
```

KDF として PANA の鍵導出関数と同じもの、つまり PRF_HMAC_SHA2_256 を用いた prf+()を使う。SMMK と SMK-HH 生成に必要な optional data 内の length は符号無し 8 ビット整数とする。IEEE802.15.4 の Key Index は SMMK の KEY ID(つまり当該 PANA セッション中で PAA より示された Key-Id AVP 中の 32 ビット MSK Identifier)の下位 8 ビットとする。このため PAA は同一 PaC に対して下位 8 ビットが連続して同じ値となるような MSK Identifier を割り当ててはならない。

尚、この MAC 層用鍵（SMK-HH）は、PANA による認証成功の結果として、機器間のみで共有されるマスター鍵（EMSK）から導出される。このため、機器間は 1:1 接続の構成となる。

5.6.4. 暗号化と改ざん検知

PANA セッション確立によって得られる MAC 層鍵(SMK-HH 鍵)を使用して[802.15.4]に基づく MAC Data フレームの暗号化を実施する。

PANA セッションの新規確立及び更新によって新規の MAC 層鍵が作成された場合、最新の MAC 層鍵を使用して送信 MAC フレームを暗号化すること。

MAC フレームの Frame Counter の値は新規 MAC 層鍵を使用する毎にリセットし、ホストは、既存 PANA セッションのライフタイムが有効期限内であっても送受信 MAC フレームの Frame Counter の値があふれる前に再度 PANA セッションの更新を行うこと。

暗号化にあたっては、秘匿(confidentiality)と認証(authenticity)の両方を実施するため、ENC-MIC-32(Security level 5)を使用すること。受信した MAC フレームの MIC 検証に失敗した場合は、フレームを廃棄する。

Key identifier モードとして 0x01 を使用し、Key Identifier フィールドには Key Source は使用せず、1 オクテットの Key Index のみ使用する

暗号化適用の例外

PANA メッセージ(UDP 宛先ポート番号 716)及び IPv6 Neighbor Solicitation (NS)(ICMPv6 Type 135 Code 0)/Neighbor Advertisement (NA)(ICMPv6 Type 136 code 0)メッセージは暗号化適用を除外し、MAC Auxiliary Security ヘッダを付与しない。

5.6.5. リプレイアタック対策

MAC フレームの暗号化対象となるメッセージについては、[802.15.4]における MAC Auxiliary Security ヘッダの Frame Counter 処理によってリプレイアタック対策を実施する。つまり、新たに受信した MAC フレームの Frame Counter 値が受信済みの MAC フレームの Frame Counter 値よりも小さい場合は当該 MAC フレームを廃棄する。

5.7. フレームフォーマット

UDP 通信を行うときの各レイヤでのフレームフォーマットの手順を**図 5-4**、**図 5-5**、**図 5-6**、**図 5-7**に示す。

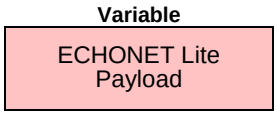


図5-4：ECHONET Lite ペイロード

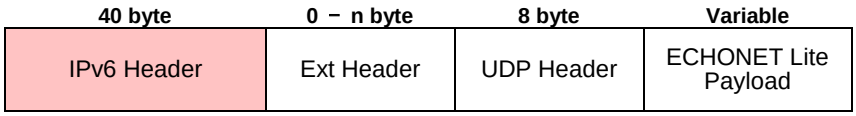


図5-5： インタフェース部で構築される IPv6 フレームフォーマット

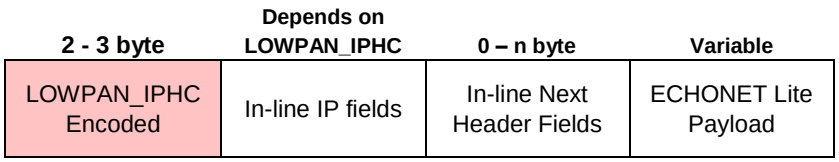


図5-6：インタフェース部で構築される 6LoWPAN フレームフォーマット

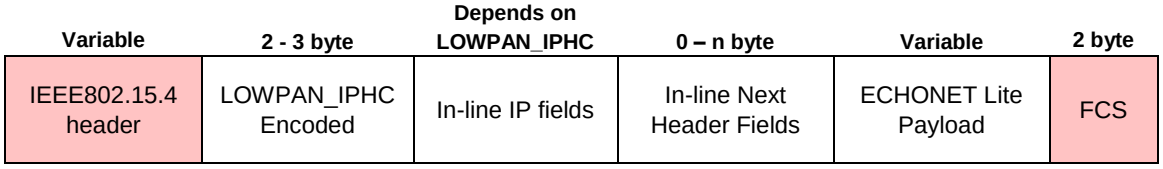


図5-7：MAC 部で構築される IEEE802.15.4 フレームフォーマット

5.8. シングルホップネットワークを構成する場合の推奨仕様

5.8.1. 概要

本節では、方式 A を用い、IPv6 上で ECHONET Lite を利用するシングルホップネットワークを構築する場合の推奨仕様を示す。ただし、方式 A の範囲内においてこれ以外の仕様を排除するものではない。

本節の仕様に基づくノードは、コーディネータを中心としたシングルホップネットワークを構築する。また、外部ネットワークとの接続方法としてアプリケーションレベルのゲートウェイ接続を想定することで、本方式内に閉じた IP ネットワークを想定している。これらの前提事項により、ECHONET Lite を用いた宅内

ネットワークの構築を可能としながらも、実装の容易性を実現している。

5.8.2. 新しいネットワークの形成

コーディネータが起動すると、本方式仕様に基づく新しいネットワークを形成する。ネットワークの形成は、(1) データリンク層の設定、(2) ネットワーク層の設定、(3) セキュリティの設定の順に実施される。ネットワーク形成手順の概要を、**図 5-8**に示す。

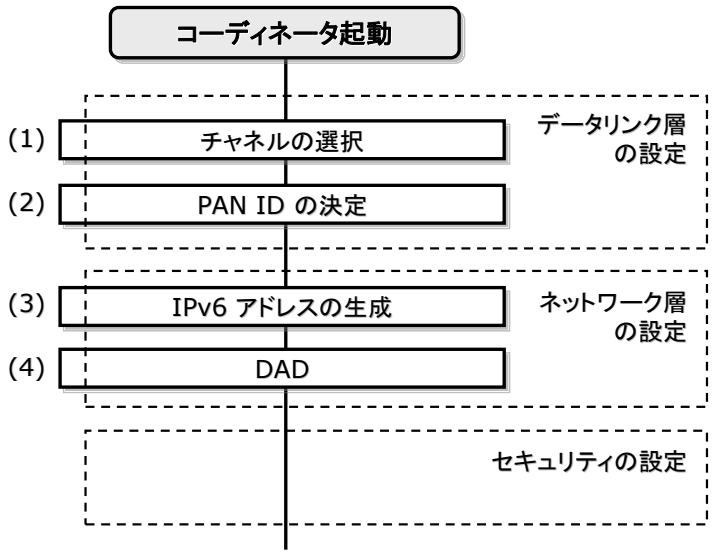


図5-8：ネットワーク形成手順概要

5.8.2.1. データリンク層の設定

コーディネータが起動すると、IEEE802.15.4 PAN を形成する。PAN 形成に関する詳細な手順は以下のとおりである。

コーディネータはまず使用するチャネルの選択を行う。チャネル選択は、ED スキャンやアクティブスキャンを利用して実施する。その際、他システムとの干渉が小さいと想定されるチャネルを選択することが好ましい。(ステップ 1)

次に、コーディネータはステップ 1 で選択したチャネルにおいて検出されたいずれの PAN も使用していない PAN ID を選択し、自身の管理するネットワークにて使用する PAN ID とする。ステップ 1 で選択したチャネルにおいて検出されたいずれの PAN も使用していない PAN ID の中からどのような値を自ネットワークの PAN ID として選択するかについては、本方式では規定しない。(ステップ 2)

以上を実施した後、コーディネータは決定した無線チャネルと PAN ID により PAN の形成を完了する。

5.8.2.2. ネットワーク層の設定

データリンク層の設定が完了すると、コーディネータはネットワーク層(IPv6)の初期設定を行う。

まず、コーディネータは自身の IPv6 アドレスを生成する。その際のプレフィックスは FE80::0/64、インタフェース識別子は[6LOWPAN]と[SLAAC]の記載に従い自身の MAC アドレス(EUI-64)に基づいて生成する。(ステップ 3)

コーディネータは、ステップ 3 で生成した IP アドレスを設定する IEEE802.15.4/4e/4g インタフェースにグローバルアドレスやユニークローカルアドレスを設定しても構わないが、それらについては本方式仕様の範囲外である。また、コーディネータは本ネットワークで用いる IEEE802.15.4/4e/4g 以外のインタフェースを有する可能性があるが、それらについても本方式既定の範囲外である。

通常、IPv6 アドレスを構成する場合はこの時点で重複アドレス検出(Duplicate Address Detection: DAD)を実施して、ネットワーク内の他のノードと IP アドレスが重複していないことを確認するが、本方式規定のノードは常に EUI-64 アドレスから IPv6 アドレスを生成するため、本方式ネットワーク内では基本的に IP アドレスが重複しない。このため、DAD は実施しなくてもよい。(ステップ 4)

5.8.2.3. セキュリティの設定

コーディネータは、データリンク層およびネットワーク層の設定に続き、セキュリティの設定を行う。ここで形成されるネットワークで利用するセキュリティ技術は、アプリケーション要件に応じて適切に選択する。本方式ではコーディネータによるセキュリティ設定の具体的な手順は記載しない。

なお、セキュリティの設定は、(データリンク層の設定と)ネットワーク層の設定が実施される中で実施する場合もあることに注意すること。

5.8.3. ネットワークへの参加

新規ホストが起動すると、本方式が規定する既存のネットワークへの参加を試みる。ホストのネットワークへの参加も、コーディネータによるネットワーク形成と同様、(1) データリンク層の設定、(2) ネットワーク層の設定、(3) セキュリティの設定の手順に大別される。新規ホストが既存のネットワークに参加するための手順の概要を、**図 5-9** に示す。

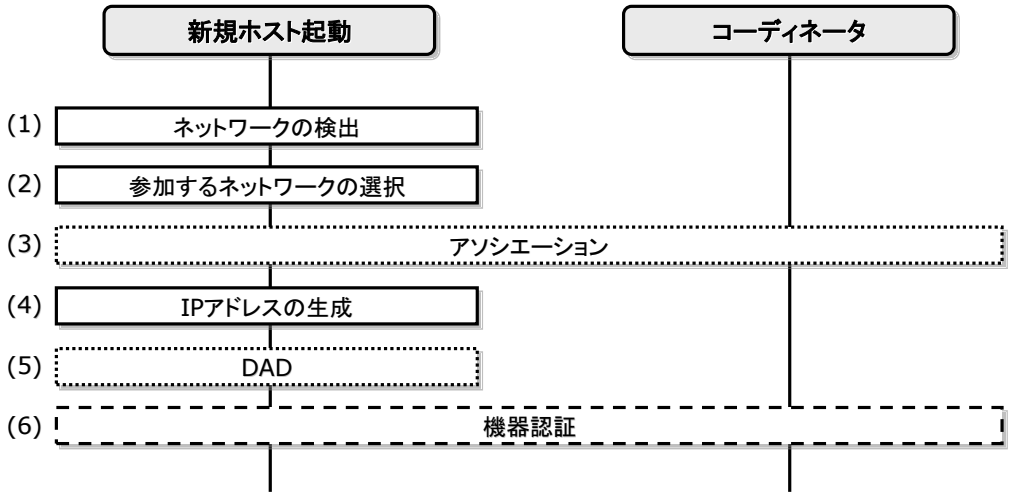


図5-9：ネットワーク参加手順概要

5.8.3.1. データリンク層の設定

新規ホストが起動すると、まず周囲に存在する IEEE802.15.4 PAN の検出を行う。PAN の検出は、新規ホストが[802.15.4]および[T108]で規定される無線チャネルのうち使用可能なすべてのチャネルにおいて[802.15.4]で規定されるビーコン要求コマンドメッセージを送信し、それを受信したコーディネータが応答としてビーコンフレームを送信、新規ホストがこのビーコンを受信することで実現される。また、新規ホストはこれらの手順の結果として、コーディネータが使用する無線チャネルと PAN ID を識別することができる。(ステップ 1)

ステップ 1 において PAN が 1 つのみ検出された場合、その PAN に対して次のステップに進む。複数の PAN が検出された場合はいずれか 1 つの PAN を選択して次のステップに進むが、どの PAN を選択するかは実装依存とする。(ステップ 2)

ここで選択した PAN について以降のネットワーク参加手順を実施した結果としてネットワークへの参加に失敗した場合、新規ホストはステップ 1 もしくはステップ 2 に戻って参加手順を再実施することが推奨される。また、その場合はステップ 2 においては既に参加に失敗したネットワーク以外のネットワークを選択すべきである。

この時点で、新規ホストは[802.15.4]で規定されるアソシエーションを実施してもよいが、上位層レベルでコーディネータを認識するため、データリンク層レベルでのアソシエーションの実施は必須ではない。(ステップ 3)

5.8.3.2. ネットワーク層の設定

IEEE802.15.4 PAN への参加が完了すると、新規ホストは自身の IPv6 アドレスを生成する。その際のプレフィックスは FE80::0/64、インタフェース識別子は[6LOWPAN]と[SLAAC]の記載に従い自身の MAC アドレス(EUI-64)に基づいて生成する。(ステップ 4)

通常、IPv6 アドレスを構成する場合はこの時点で重複アドレス検出(Duplicate Address Detection: DAD)を実施して、ネットワーク内の他のノードと IP アドレスが重複していないことを確認するが、本方式規定のノードは常に EUI-64 アドレスから IPv6 アドレスを生成するため、本方式ネットワーク内では基本的に IP アドレスが重複しない。このため、DAD は実施しなくてもよい。(ステップ 5)

この時点で、新規ホストはコーディネータからの機器認証を受ける。機器認証の仕組みについては本方式の規定範囲外であるが、新規ホストは認証元のノードをコーディネータと認識し、コーディネータのアドレス情報を保管する。(ステップ 6)

5.8.3.3. セキュリティの設定

データリンク層およびネットワーク層の設定が完了すると、新規ホストは、コーディネータとのセキュリティの設定を行う。ここで形成されるネットワークで利用するセキュリティ技術は、アプリケーション要件に応じて適切に選択する。本方式ではコーディネータによるセキュリティ設定の具体的な手順は記載しない。

5.8.4. 推奨仕様を実現するためのデバイス/物理層/MAC 層の仕様

本節の仕様を動作させる場合の最低必要な IEEE802.15.4/4e/4g に関する仕様を**表 5-17**、**表 5-18**、**表 5-19** に示す。この表の動作仕様において‘Y’の機能は本例にて使用する機能であり、‘N’の機能は使用しない機能である。また、‘O’の機能は、注釈の条件によって使用する場合と使用しない場合がある機能である。本節の仕様に基づく場合、MAC 層は non-beacon mode を利用する。

表5-17:動作例を実現するためのデバイス/物理層の仕様

番号 ※1	動作仕様： Support	番号 ※2	動作仕様： Support	番号 ※3	動作仕様： Support	番号 ※3	動作仕様： Support
FD1	O.1	PLF1	Y	RF12	―	RF13.4	100kbit/s または 50kbit/s の少なくとも一方を使用
FD2	O.1	PLF2	Y	RF12.1	Y	RF13.5	N
FD3	Y	PLF3	Y	RF12.2	N	RF14	―
FD4	N	PLF4	Y	RF12.3	N	RF14.1	N
FD5	N	PLF4.1	Y	RF12.4	N	RF14.2	N
FD5	N	PLF4.1	Y	RF12.4	N	RF14.2	N
FD8	Y	PLF4.2	N	RF12.5	N	RF14.3	Y
		PLF4.3	N	RF12.6	Y	RF14.4	N
		PLP1	255 オクテットまで使用	RF13	―		

※1：**表 5-6** function device type の item number に対応

※2：**表 5-1** PLF/PLP 機能 の item number に対応

※3：**表 5-2** RF 機能 の item number に対応

表5-18:動作例を実現するための MAC 層の仕様

番号 ※1	動作仕様： Support	番号 ※1	動作仕様： Support	番号 ※1	動作仕様： Support	番号 ※2	動作仕様： Support
MLF1	Y	MLF7	Y	MLF15	N	MF1	Y
MLF1.1	O※3※5	MLF8	O※6	MLF16	N	MF2	Y
MLF2	Y	MLF9	Y	MLF17	N	MF3	Y
MLF2.1	N	MLF9.1	Y	MLF18	Y	MF4	Y
MLF2.2	O※4	MLF9.2	Y	MLF18.1	Y	MF4.1	O※6
MLF2.3	N	MLF9.2.1	Y	MLF18.1.1	Y	MF4.2	O※6
MLF3	Y	MLF9.2.2	Y	MLF19	N※8	MF4.3	O※6
MLF3.1	Y※5	MLF10.1	Y※5	MLF19.1	N※8	MF4.4	O※3
MLF3.2	Y	MLF10.2	Y	MLF19.2	N※8	MF4.5	N
MLF4	Y	MLF10.3	N	MLF19.3	N	MF4.6	O※3
MLF5	N	MLF10.4	O※3	MLF19.4	N	MF4.7	Y※9
MLF5.1	N	MLF11	N	MLF19.5	N※8	MF4.8	O※3
MLF5.2	N	MLF12	N	MLF19.6	N※8	MF4.9	N
MLF6	Y	MLF13	O※3	MLF19.7	N	MF5	Y※10
		MLF15(4g)	O※7	MLF19.8	N		
				MLF20	N		
				MLF21	N		
				MLF23	N		
				MLF23.1	N		

※1　**表 5-7** MAC sublayer function の item number に対応

※2　**表 5-8** MAC frame の item number に対応

※3：常時電源機器のみで構成されるネットワークの場合は使用しなくてもよい

※4：必要に応じて使用可能

※5：子機は使用しない

※6：上位層で代替できる場合は、本機能を使用しなくてもよい

※7：50kbit/s と 100kbit/s を共存させる場合は使用する

※8：シングルホップ通信を想定するため、使用しない

※9：子機も使用可能(参照規格にない FD2 規定を明確化)

※10：PSDU サイズが 255 オクテット以下では、16bit 誤り検出を使用

表5-19:動作例を実現するための物理層の仕様

項目	実装に関する規定	備考
変調方式	GFSK	
伝送速度	100kbit/s、または 50kbit/s	
送信出力	20mW	
周波数チャネル	ARIB 規定 33～60 チャネルの(奇数＋偶数)チャネル、または ARIB 規定 33～61 チャネル	33 ～ 38 チャネルは送信出力 250mW のシステムも利用する
周波数チャネル幅	400kHz(2ch 束ね)、または 200kHz	
送信プリアンブル長	15byte 以上	

5.9. シングルホップスマートメーター･HEMS 間推奨通信仕様

5.9.1. 概要

本節では、方式 A を用い、IPv6 上で ECHONET Lite を利用するスマートメーター・HEMS 間のシングルホップネットワークを構築する場合の推奨仕様を示す。

本節の仕様に基づくノードは、コーディネータとなるスマートメーターが HEMS との間で 1：1 接続のシングルホップネットワークを構築する。

5.9.2. 物理層

本節の仕様を動作させる場合の最低必要な IEEE802.15.4/4e/4g に関する仕様を表 5-20 に示す。この表の動作仕様において‘Y’の機能は本例にて使用する機能であり、‘N’の機能は使用しない機能である。本節の仕様に基づく場合、MAC 層は non-beacon mode を利用する。

表 5-20: 動作例を実現するための物理層の仕様

番号 ※1	動作仕様：Support	番号 ※2	動作仕様：Support	番号 ※3	動作仕様：Support	番号 ※3	動作仕様：Support
FD1	O.1	PLF1	Y	RF12	―	RF13.4	
FD2	O.1	PLF2	Y	RF12.1	Y	RF13.5	
FD3	Y	PLF3	Y	RF12.2	N	RF14	
FD4	N	PLF4	Y	RF12.3	N	RF14.1	
FD5	N	PLF4.1	Y	RF12.4	N	RF14.2	
FD8	Y	PLF4.2	N	RF12.5	N	RF14.3	
		PLF4.3	N	RF12.6	Y	RF14.4	
		PLP1	255 オクテットまで使用	RF13	―		

※1：表 5-3 function device type の item number に対応

※2：表 5-1 PLF/PLP 機能 の item number に対応

※3：表 5-2 RF 機能 の item number に対応

無線インタフェース仕様を表 5-21 に示す。

表 5-21:　無線インタフェース仕様

項目	実装に関する規定	備考
変調方式	GFSK	
伝送速度	100kbit/s	
送信出力	20mW	
周波数チャネル	ARIB 規定 33～60 チャネルの(奇数＋偶数)チャネル、または ARIB 規定 33～61 チャネル	33 ～ 38 チャネルは送信出力 250mW のシステムも利用する
周波数チャネル幅	400kHz(2ch 束ね)	
受信感度	-88dBm 以下@PER<10%,250 octets（受信感度の規定点はアンテナコネクタ端）	
送信プリアンブル長	15byte 以上	1200us～4000us
受信プリアンブル長	15byte	1200us
空中線利得	3dBi 以下	
アンテナダイバーシティ	2 アンテナの選択ダイバーシティを推奨	

5.9.3.　データリンク(MAC)層

5.9.3.1.　IEEE802.15.4/4e/4g に関する仕様

本節の仕様を動作させる場合の最低必要な IEEE802.15.4/4e/4g に関する仕様を表 5-22に示す。この表の動作仕様において‘Y’の機能は本例にて使用する機能であり、‘N’の機能は使用しない機能である。本節の仕様に基づく場合、MAC 層は non-beacon mode を利用する。

表 5-22:　動作例を実現するための MAC 層の仕様

番号 ※1	動作仕様： Support	番号 ※1	動作仕様： Support	番号 ※1	動作仕様： Support	番号 ※2	動作仕様： Support
MLF1	Y	MLF7	Y	MLF15	N	MF1	Y
MLF1.1	N	MLF8	N	MLF16	N	MF2	Y
MLF2	Y	MLF9	Y	MLF17	N	MF3	Y
MLF2.1	N	MLF9.1	Y	MLF18	Y	MF4	Y
MLF2.2	N	MLF9.2	Y	MLF18.1	Y	MF4.1	N
MLF2.3	N	MLF9.2.1	Y	MLF18.1.1	Y	MF4.2	N
MLF3	Y	MLF9.2.2	Y	MLF19	N	MF4.3	N
MLF3.1	Y※5	MLF10.1	Y※5	MLF19.1	N	MF4.4	N
MLF3.2	Y	MLF10.2	Y	MLF19.2	N	MF4.5	N
MLF4	Y	MLF10.3	N	MLF19.3	N	MF4.6	N
MLF5	N	MLF10.4	N	MLF19.4	N	MF4.7	Y※9
MLF5.1	N	MLF11	N	MLF19.5	N	MF4.8	N
MLF5.2	N	MLF12	N	MLF19.6	N	MF4.9	N
MLF6	Y	MLF13	N	MLF19.7	N	MF5	Y※10
		MLF15(4g)	N	MLF19.8	N		
				MLF20	N		
				MLF21	N		
				MLF23	N		
				MLF23.1	N		

※1：表 5-4　MAC sublayer function の item number に対応

※2：表 5-5　MAC frame の item number に対応

※5：子機は使用しない

※9：子機も使用可能(参照規格にない FD2 規定を明確化)

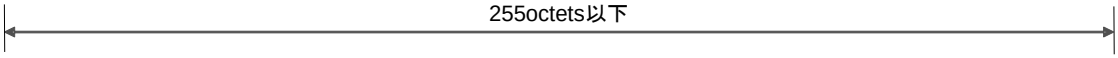
※10：PSDU サイズが 255 オクテット以下では、16bit 誤り検出を使用

5.9.3.2.　MAC フレームフォーマット

[802.15.4] 5.2 MAC frame formats をベースとし、本仕様の MAC フレームフォーマットについて記述する。

5.9.3.2.1.　DATA フレーム

本仕様で使用する DATA フレームフォーマットを図 5-10 に示す。([802.15.4e] 5.2.2.2 Data frame format を参照し、本仕様での使い方を明確化)



Octets:2	1	2	2/8	8	0/6	Variable	2
Frame Control	Sequence Number	Destination PAN Identifier	Destination Address	Source Address	Auxiliary Security Header	Frame Payload	FCS
		Addressing fields					
MHR						MAC Payload	MFR

図 5-10:　DATA フレームフォーマット

(1) Frame Control フィールド

Frame Control フィールドの内容を表 5-23 に示す。

表 5-23Frame Control（DATA フレーム）

bit	内容	備考
2-0	Frame Type	DATA フレームを示す"001"を設定する
3	Security Enable	セキュリティ無効の場合は"0"、セキュリティ有効の場合は"1"を設定する
4	Frame Pending	使用しないので"0"を設定する
5	AR(Ack Request)	ACK リクエスト無し (ブロードキャスト)の場合は"0"、 ACK リクエスト有り (ユニキャスト)の場合は"1"を設定する
6	PAN ID Compression	[802.15.4e] Table 2a に従い"0"を設定する
7	Reserved	原則として"0"を設定するが、Don't Care とする
8	Sequence Number Suppression	Sequence Number フィールドを使用するので、″0″を設定する
9	IE List Present	IEs は使用しないので"0"を設定する
11-10	Destination Addressing Mode	ユニキャストアドレスの場合は 64-bit 拡張アドレスを使用するため"11"を設定する。ブロードキャストアドレスの場合は 16-bit ショートアドレスを使用するため”10”を設定する。
13-12	Frame Version	拡張フォーマットの ACK を使用するため"10"を設定する　※1、※2
15-14	Source Addressing Mode	64-bit 拡張アドレスを使用するので"11"を設定する

※1：enhanced acknowledgment フレームによる応答を想定し、802.15.4-2003/2006 との非互換性を示すため、常に 0b10 を設定する。

※2：以下の仕様とする:

a) 本仕様の装置は Frame Version フィールドが 10b の Beacon、DATA、ACK、command フレームを受信できるものとする。

b) 本仕様の装置は Frame Version フィールドが 00b または 01b の Beacon、DATA、ACK、command フレー

ムを受信できてもよい。

c) 本仕様の装置は Beacon、DATA、ACK、command フレームを生成する場合は Frame Version フィールドを 10b に設定するものとする。

(2) Sequence Number フィールド

[802.15.4] 5.2.1.2 Sequence Number field 参照。

(3) Addressing フィールド

Source Address は、64-bit 形式の MAC アドレスであり、Destination Address は、64-bit 形式の MAC アドレスまたは 16-bit ブロードキャストアドレス(0xffff)のいずれかである。これらのアドレスフィールドは、least significant octet から送出し、各オクテットは least significant bit(LSBit)から順に送出する。

Source PAN Identifier は Addressing フィールドに含まれない。PAN Identifier は、16bit の数値として扱い、LSBit から順に送出する。

(4) Auxiliary Security Header

フレームを暗号化する場合に使用する Auxiliary Security Header の内容を表 5-24 に示す。

表 5-24: Auxiliary Security フィールド

octet	bit	内容		備考
1	b2-b0	Security Control	Security Level	ENC-MIC-32 を使用するので"101"を設定する
	b4-b3		Key Identifier Mode	1 オクテットの鍵 ID を使用するので"01"を設定する
	b7-b5		Reserved	-
4	-	Frame Counter		
1	-	Key Identifier		

5.9.3.2.2. ACK フレーム

本仕様で用いる ACK フレームのフォーマットを図 5-11 に示す。([802.15.4e] 5.2.2.3 Acknowledgment frame format を参照し、本仕様での使い方を明確化)

Octets:2	1	2	8	2
Frame Control	Sequence Number	Destination PAN Identifier	Destination Address	FCS
		Addressing fields		
MHR				MFR

図 5-11: ACK フレームフォーマット

(1) Frame Control フィールド

Frame Control フィールドの内容を表 5-25 に示す。

表 5-25：Frame Control(ACK フレーム)

bit	内容	備考
2-0	Frame Type	ACK フレームを示す"010"を設定する
3	Security Enable	セキュリティは無効とするので"0"を設定する
4	Frame Pending	使用しないので"0"を設定する
5	AR(Ack Request)	"0"を設定する
6	PAN ID Compression	[802.15.4e] Table 2a に従い"0"を設定する
7	Reserved	"0"を設定する
8	Sequence Number Suppression	Sequence Number フィールドを使用するので、″0″を設定する
9	IE List Present	IEs は使用しないので"0"を設定する
11-10	Destination Addressing Mode	64-bit 拡張アドレスを使用するので"11"を設定する
13-12	Frame Version	拡張フォーマットを使用するので"10"を設定する
15-14	Source Addressing Mode	Source Address は使用しないので"00"を設定する

(2) Sequence Number フィールド

[802.15.4] 5.2.1.2 Sequence Number field 参照。ACK フレームでは、送達応答対象の受信 DATA フレームの値を設定する。

(3) Addressing フィールド

Destination Address は、送達応答対象の受信フレームの Source Address を設定する。本仕様 5.9.3.2.1 節の DATA フレームの Addressinng フィールドの項を参照。

5.9.3.2.3. Enhanced Beacon フレームフォーマット

本仕様で用いる Enhanced Beacon フレームフォーマットを図 5-12 に示す。（[802.15.4e] 5.2.2.1 Beacon frame format を参照し、本仕様での使い方を明確化）

<table>
<tr><td>Octets:2</td><td>1</td><td>2</td><td>8</td><td>8</td><td>Variable</td><td>2</td></tr>
<tr><td rowspan="2">Frame Control</td><td rowspan="2">Sequence Number</td><td>Destination PAN Identifier</td><td>Destination Address</td><td>Source Address</td><td rowspan="2">Payload IE</td><td rowspan="2">FCS</td></tr>
<tr><td colspan="3">Addressing fields</td></tr>
<tr><td colspan="5">MHR</td><td>MAC Payload</td><td>MFR</td></tr>
</table>

図 5-12：Enhanced Beacon フレームフォーマット

(1) Frame Control フィールド

Frame Control フィールドの内容を表 5-26 に示す。

表 5-26: Frame Control（Enhanced Beacon フレーム）

bit	内容	備考
2-0	Frame Type	Beacon フレームを示す"000"を設定する
3	Security Enable	セキュリティは無効とするので"0"を設定する
4	Frame Pending	使用しないので"0"を設定する
5	AR(Ack Request)	ACK リクエスト有り (ユニキャスト)のため"1"を設定する
6	PAN ID Compression	[802.15.4e] Table 2a に従い"0"を設定する
7	Reserved	原則として"0"を設定するが、Don't Care とする
8	Sequence Number Suppression	Sequence Number フィールドを使用するので、″0″を設定する
9	IE List Present	IEs を使用する場合は"1"、 IEs を使用しない場合は"0"を設定する
11-10	Destination Addressing Mode	64-bit 拡張アドレスを使用するので"11"を設定する
13-12	Frame Version	拡張フォーマットを使用するので"10"を設定する
15-14	Source Addressing Mode	64-bit 拡張アドレスを使用するので"11"を設定する

(2) Sequence Number フィールド

[802.15.4e] 5.2.2.1.1 Beacon frame MHR fields を参照し、装置が保持しているシーケンス番号（macEBSN）の値を設定する。

(3) Addressing フィールド

Destination Address は Enhanced Beacon Request の Source Address を設定する。本仕様 5.9.3.2.1 節の DATA フレームの Addressinng フィールドの項を参照。

Destination PAN Identifier は本フレームを送信する装置の Source PAN Identifier を設定する。

(4) Payload IE

Enhanced Beacon Request に設定された IEs フィールドの情報と同じ情報を設定する。

5.9.3.2.4. Enhanced Beacon request command フレームフォーマット

本仕様で用いる Enhanced Beacon request command フレームのフォーマットを図 5-13 に示す。（[802.15.4e] 5.3.7.2 Enhanced beacon request を参照し、本仕様での使い方を明確化）

<table>
<tr><td>Octets:2</td><td>1</td><td>2</td><td>2</td><td>8</td><td>Variable</td><td>1</td><td>2</td></tr>
<tr><td rowspan="2">Frame Control</td><td rowspan="2">Sequence Number</td><td>Destination PAN Identifier</td><td>Destination Address</td><td>Source Address</td><td rowspan="2">Payload IE</td><td rowspan="2">Command Frame Identifier</td><td rowspan="2">FCS</td></tr>
<tr><td colspan="3">Addressing fields</td></tr>
<tr><td colspan="5">MHR</td><td colspan="2">MAC Payload</td><td>MFR</td></tr>
</table>

図 5-13: Enhanced Beacon request command フレームフォーマット

(1) Frame Control フィールド

Frame Control フィールドの内容を表 5-27 に示す。

表 5-27: Frame Control（Enhanced Beacon request command フレーム）

bit	内容	備考
2-0	Frame Type	MAC command フレームを示す"011"を設定する
3	Security Enable	セキュリティは無効とするので"0"を設定する
4	Frame Pending	使用しないので"0"を設定する
5	AR(Ack Request)	ACK リクエスト無し (ブロードキャスト)のため"0"を設定する
6	PAN ID Compression	[802.15.4e] Table 2a に従い"0"を設定する
7	Reserved	"0"を設定する
8	Sequence Number Suppression	Sequence Number フィールドを使用するので、″0″を設定する
9	IE List Present	IEs を使用する場合は"1"、 IEs を使用しない場合は"0"を設定する
11-10	Destination Addressing Mode	16-bit 拡張アドレスを使用するので"10"を設定する
13-12	Frame Version	拡張フォーマットを使用するので"10"を設定する
15-14	Source Addressing Mode	64bit-拡張アドレスを使用するので"11"を設定する

(2) Sequence Number フィールド

[802.15.4] 5.2.1.2 Sequence Number field 参照。

(3) Addressing フィールド

5.9.3.2.1 節の DATA フレームの Addressinng フィールドの項を参照。

(4) Payload IE

本仕様 5.9.6.1.1 節　データリンク層の設定を参照。

(5) Command Frame Identifier

[802.15.4e] Table 5 を参照し、"0x07"を設定する。

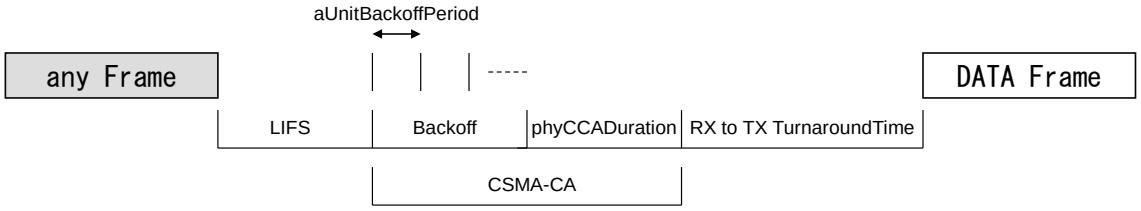
5.9.3.3. 主な MAC 機能の記述

本仕様の主な MAC 機能について記述する。

5.9.3.3.1. 送信タイミング規定

(1) DATA フレームの送信タイミング規定

DATA フレームの送信タイミング規定を図 5-14 に示す。（[802.15.4] 5.1.1.4 CSMA-CA algorithm の記述、[802.15.4g] Table 51 に基づき、本仕様でのタイミング規定を明確化）



パラメータ（注１）	算出式	値[μsec](nominal)（注 2）
LIFS	aTurnaroundTime	1000
aUnitBackoffPeriod	phyCCADuration＋aTurnaroundTime	1130
phyCCADuration	－	130
RX to TX TurnaroundTime	－	300 以上 1000 以下

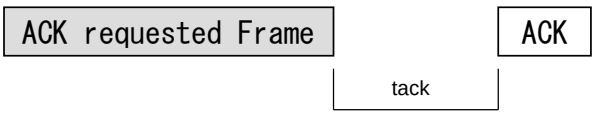
注１：5.9.3.3.5 参照

注２：値の誤差範囲については[802.15.4]、[802.15.4e]、[802.15.4g]を参照のこと

図 5-14：DATA フレームの送信タイミング規定

(2) ACK フレーム送信タイミング規定

ACK フレームの送信タイミング規定を図 5-15 に示す。（[802.15.4] 5.1.1.3 Interframe spacing (IFS)の記述に基づき、tack の下限を規定し、本仕様でのタイミング規定を明確化）



パラメータ（注１）	算出式	値[μsec]
tack	RX to TX TurnaroundTime	300 以上 1000 以下 (注 2)

注 1：5.9.3.3.5 参照

注 2：TX to RX TurnaroundTime ＜ 300μs であること。

図 5-15: ACK フレームの送信タイミング規定

5.9.3.3.2.　CSMA-CA

再送を含めた CSMA-CA アルゴリズムを図 5-16 に示す。（[IEEE802.15.4e] 5.1.1.4 CSMA-CA algorithm をベースに、本仕様での再送を含めた CSMA-CA アルゴリズムを明確化）

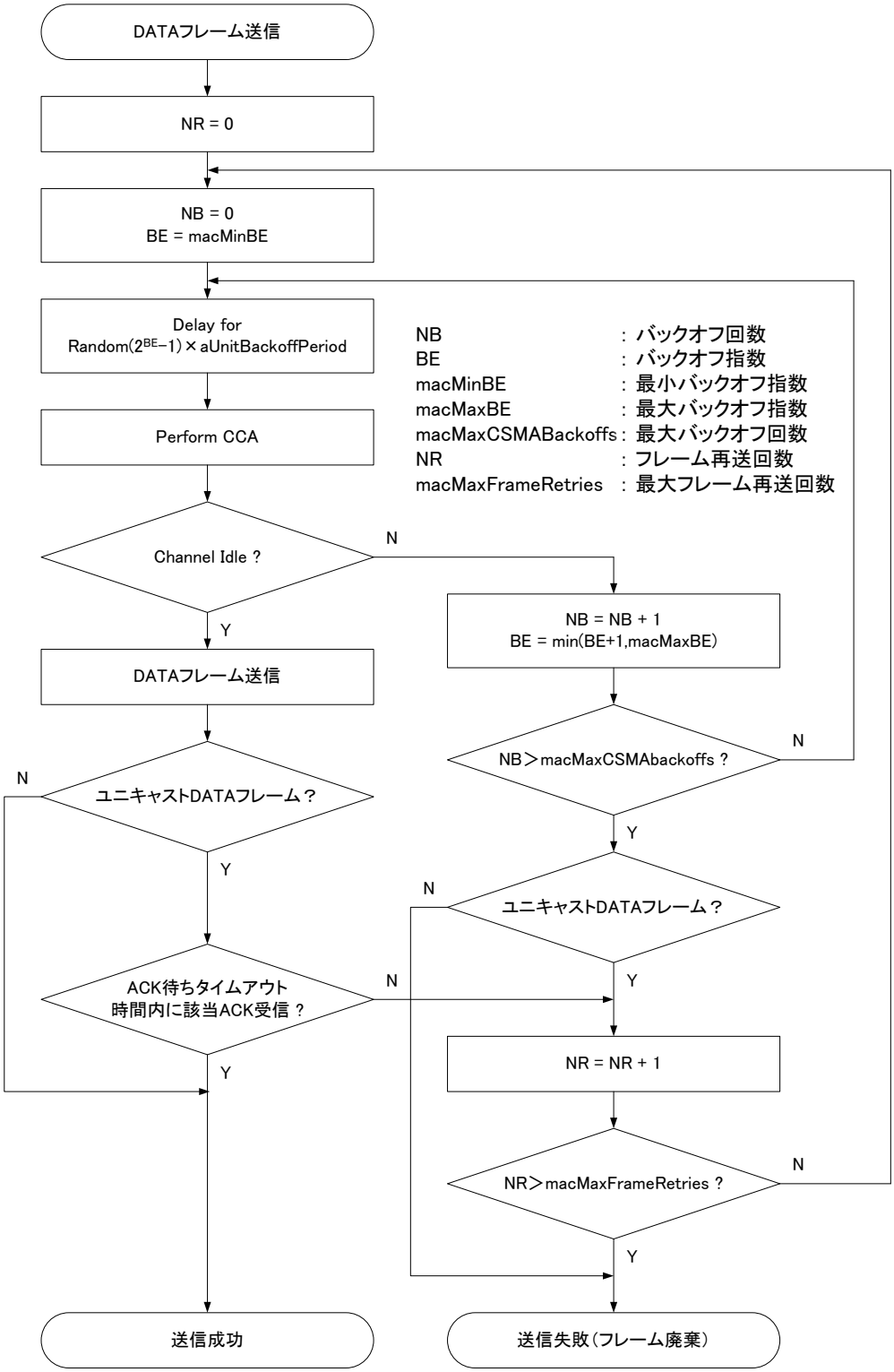
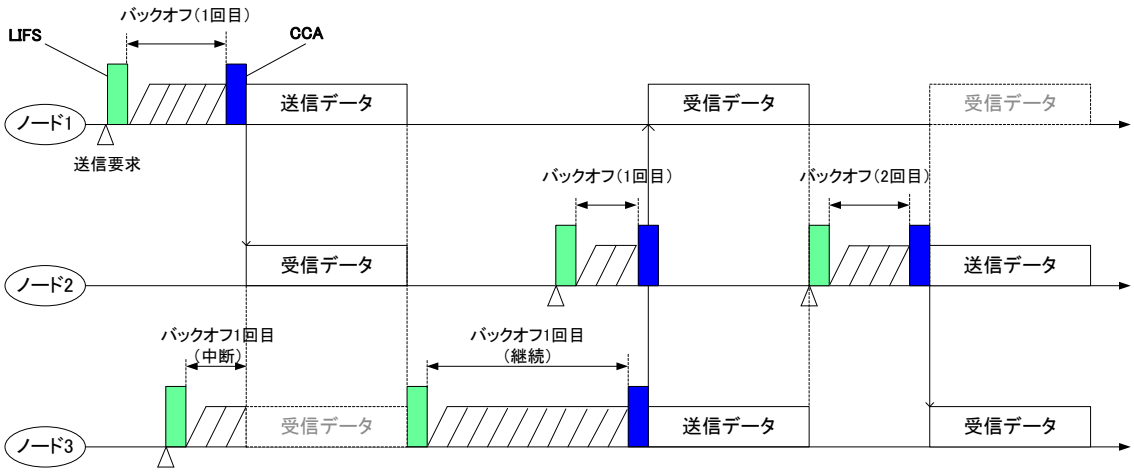


図 5-16: DATA フレーム送信における再送を含めた CSMA-CA アルゴリズム

5.9.3.3.3.　バックオフ動作

本仕様のバックオフ動作を図 5-17 に示す。（[802.15.4] 5.1.1.4 CSMA-CA algorithm の記述に基づき、動作を明確化）



No	送信動作	説明
1	ノード１送信動作	バックオフ（１回目）後の CCA で Idle →送信
2	ノード２送信動作	バックオフ（１回目）後の CCA で busy →Idle に遷移するまで待つ（受信可能な場合は、データ受信する）（注１） →バックオフ２回目後の CCA で Idle →送信
3	ノード３送信動作	バックオフ（１回目）中にデータ受信 →データ受信後 Idle 遷移 →バックオフ（１回目）継続（残バックオフ時間消化）後の CCA で Idle →送信

本図では ACK フレームを省略。

注１：CCA 期間中にビジーを検出した場合、データ受信を行うか否かは、採用 PHY に依存する。

図 5-17: バックオフ動作

5.9.3.3.4.　送信時間管理機能

(1) 休止時間管理

[T108]の規定に基づき、休止時間を設けること。

(2) 送信時間総和管理

[T108]では、DATA フレームの１時間あたりの送信時間の総和が 360[s]以内と規定されているため、遵守出来る機能を有すること。

5.9.3.3.5. MAC 定数と変数

(1) MAC 定数

本仕様の MAC 定数について表 5-28 に示す。（[802.15.4g] Table 51, Table 71 を参照し、nominal 値を規定）

表 5-28: MAC 定数

定数名	説明[単位]	値(nominal)（注）	備考
phyCCADuration	キャリアセンス時間[μsec]	130	
aTurnaroundTime	送受信切り替え時間[μsec]	1000	
RX to TX TurnaroundTime (=tack)	受信→送信切り替え時間[μsec]	300以上1000以下	
TX to RX TurnaroundTime	送信→受信切り替え時間[μsec]	300 未満	
macMinLIFSPeriod	LIFS 最小値[μsec]	1000	5.9.3.3.1参照
aUnitBackoffPeriod	バックオフ単位時間[μsec]	1130	5.9.3.3.1参照
macAckWaitDuration	DATA フレーム送信終了後から ACK フレームの受信を待つ時間[ms]	5	5.9.3.3.1参照

注：値の誤差範囲については[802.15.4]、[802.15.4e]、[802.15.4g]を参照のこと

(2) MAC 変数

本仕様の MAC 変数について表 5-29 に示す。（[802.15.4] Table 52 を参照し、デフォルト値を規定）

表 5-29: MAC 変数

変数名	説明	範囲	デフォルト値	備考
macMaxBE	最大バックオフ指数	3～15(注)	8	
macMinBE	最小バックオフ指数	0～macMaxBE	8	
macMaxCSMABackoffs	最大バックオフ回数	0～5	4	
macMaxFrameRetries	フレーム再送回数	0～7	3	

注：待ち時間の幅を広げるため 15 としている（ただし、デフォルト値は規格範囲内の 8）。

5.9.4. インタフェース部

5.9.4.1. 概要

シングルホップスマートメーター・HEMS 間推奨通信仕様における インタフェース部は、以降の各項で特に記載のない限り 5.5 章に準拠すること。

5.9.4.2. アダプテーション層

スマートメーター、及び HEMS は、5.5.3 節に準拠すること。

5.9.4.2.1. Fragmentation

スマートメーター、及び HEMS は、5.5.3.1 項に準拠すること。

5.9.4.2.2. Header compression

スマートメーター、及び HEMS は、5.5.3.2 項に準拠すること。

5.9.4.2.3. Neighbor discovery

スマートメーター、及び HEMS は、5.5.3.3 項に記載のとおり、IPv6 ベースの Neighbor discovery を使用するため、6LoWPAN-ND に基づく 5.5.3.3.項の 5.9.4.2.3. Neighbor discovery はサポートしないものとする。IPv6 ベースの Neighbor discovery については次項のネットワーク層を参照のこと。

5.9.4.3. ネットワーク層

スマートメーター、及び HEMS は、5.5.4 節に準拠すること。

5.9.4.3.1. IP アドレッシング

スマートメーター、及び HEMS は、5.5.4.1 項に準拠すること。

5.9.4.3.2. 近隣探索

スマートメーター、及び HEMS は、5.5.4.2 項に準拠すること。

5.9.4.3.3. マルチキャスト

スマートメーター、及び HEMS は、5.5.4.3 項に準拠すること。

5.9.4.4. トランスポート層

スマートメーター、及び HEMS は、5.5.5 節に準拠すること。

5.9.4.5. アプリケーション層

スマートメーター、及び HEMS は、5.5.6 節に準拠すること。

5.9.5. セキュリティ処理

5.9.5.1. 概要

本仕様では、5.6 に従ったセキュリティ処理を行う。

5.9.5.2. 認証

5.6.2 に従う。本仕様ではスマートメーターが PAA となり、HEMS が PaC となる。

5.9.5.2.1. PANA

5.6.2.1 に従う。

5.9.5.2.2. EAP

5.6.2.2 に従う。

5.9.5.3. 鍵更新

5.6.3 に従う。

5.9.5.3.1. PANA 鍵導出関数

5.6.3.1 に従う。

5.9.5.3.2. EAP-PSK 鍵導出関数

5.6.3.2 に従う。

5.9.5.3.3. MAC 層鍵導出関数

MAC 層で使用するセキュリティ鍵は EAP-PSK のネゴシエーションの結果導出される EMSK を用いて導出する。まず MAC 層用鍵を生成するためのマスター鍵 SMMK を USRK 導出関数[USRK]より生成し、SMMK を使用してスマートメーターと HEMS 間の MAC 層鍵 SMK-SH を導出する。

SMMK = KDF(EMSK, “Wi-SUN JP Route B” | “¥0” | optional data | length)
- optional data = NULL(0x00)
- length = 64

SMK-SH = KDF(SMMK, “Wi-SUN JP Route B” | “¥0” | optional data | length)
- optional data = EAP ID_P | EAP ID_S | IEEE802.15.4 Key Index
- length = 16

KDF として PANA の鍵導出関数と同じもの、つまり PRF_HMAC_SHA2_256 を用いた prf+()を使う。SMMK と SMK-SH 生成に必要な optional data 内の length は符号無し 8 ビット整数とする。IEEE802.15.4 の Key Index は SMMK の KEY ID（つまり当該 PANA セッション中で PAA より示された Key-Id AVP 中の 32 ビット MSK Identifier）の下位 8 ビットとする。このため PAA は同一 PaC に対して下位 8 ビットが連続して同じ値となるような MSK Identifier を割り当ててはならない。

尚、この MAC 層用鍵（SMK-SH）は、PANA による認証成功の結果として、スマートメーターと HEMS のみで共有されるマスター鍵（EMSK）から導出される。このため、スマートメーターと HEMS 間は 1:1 接続の構成となる。

5.9.5.4. 暗号化と改ざん検知

5.6.4 に従う。

5.9.5.5. リプレイアタック対策

5.6.5 に従う。

5.9.6. ネットワーク推奨設定

スマートメーターと HEMS は、長さ 8 オクテットのネットワーク識別子を用いる。本 ID はスマートメーターと HEMS との関連付けに用いられる。本仕様では、当該 ID はスマートメーターと HEMS 上であらかじめ設定されているとする。さらに、PANA/EAP のために必要な NAI と認証鍵も同様にスマートメーターと HEMS 上であらかじめ設定されているとする。

スマートメーターは、次の手順に従い、ネットワーク形成のための使用無線チャネルと PAN ID を決定する。

1-1: データリンク層の設定(スマートメーター)

無線チャネル選定と、PAN ID 検出は Energy Detection Scan(ED Scan)及び、Enhanced Active Scan を通じて行われる。無線チャネルと PAN ID の選択基準は本プロファイルでは規定しない。

1-2: ネットワーク層の設定(スマートメーター)

スマートメーターは、[SLAAC]の記載に従い、自身の IPv6 リンクローカルアドレスを決定する。スマートメーターがコーディネータとなるネットワーク構築後、HEMS は次のデータリンク層処理及びネットワーク層処理により、自宅のスマートメーターに接続処理を行う。

2-1: データリンク層の設定(HEMS)

HEMS は、Enhanced Active Scan により接続対象となるスマートメーターを検出する。

2-2: ネットワーク層の設定(HEMS)

HEMS は、[SLAAC]の記載に従い、自身の IPv6 リンクローカルアドレスを決定する。

HEMS は、スマートメーターからの Enhanced Beacon の送信元 MAC アドレスより、スマートメーターの IPv6 リンクローカルアドレスを算出し、事前に共有されている NAI と認証鍵を元に、PANA によるネットワーク認証を要求する。スマートメーターは、HEMS との PANA セッションを確立し、事前に共有されている NAI と認証鍵を元に、認証の許可または拒否の判断をする。また、認証成功時には、スマートメーターと HEMS 間で一意となる通信用の鍵情報の交換が行われる。共有した通信用の鍵情報は、MAC 層の暗号化用の鍵として利用する。

スマートメーターと HEMS 間の暗号化通信が確立すると、スマートメーターと HEMS の通信が暗号化メッセージを介して開始される。HEMS は、ECHONET Lite プロトコルを用いたサービス探索を実行し、スマートメーターは HEMS に対し、メータ検針値を 30 分毎に通知する。

5.9.6.1. 新しいネットワークの形成

スマートメーターは電源が入ると、本プロファイルに準拠する新規ネットワークを形成する。本手順は、5.6.2 節と同じである。ネットワーク形成と、本ネットワークへの参加の手順を図 5-18 に示す。

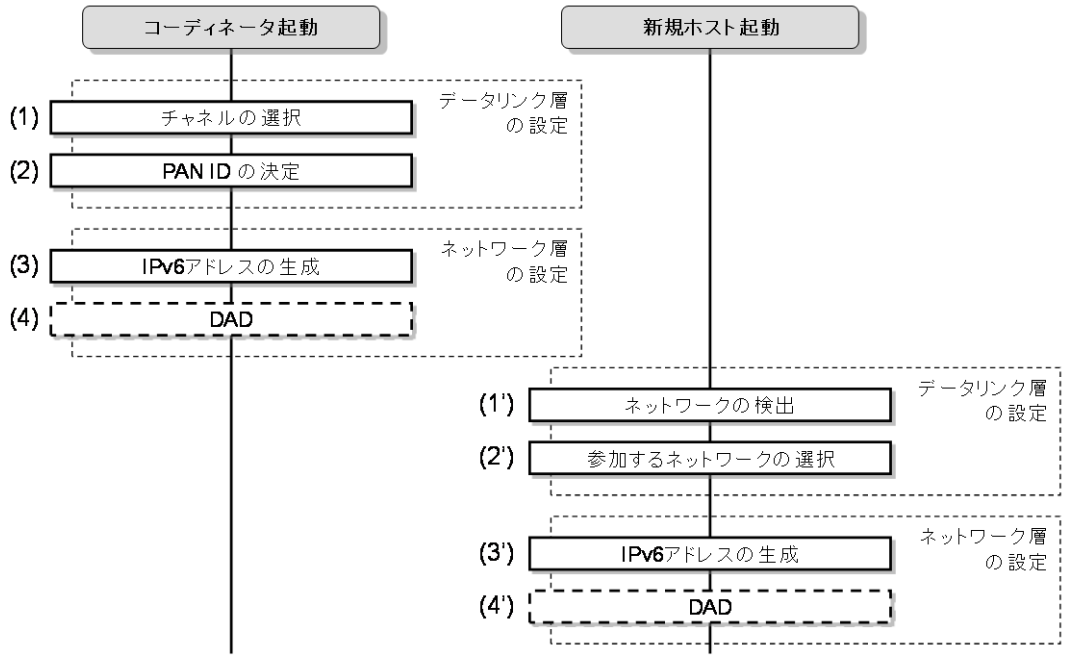


図5-18：ネットワーク形成および参加手順概要

5.9.6.1.1. データリンク層の設定

コーディネータのデータリンク層設定は 5.8.2.1 節と同じである。ただし、スマートメーターは Enhanced Active Scan を用い、Information Element field には情報を有しない。

スマートメーターのネットワークを検出するために、HEMS は Enhanced Active Scan を用い、Information Element field に MLME IE を保持する。HEMS からの Enhanced Beacon Request コマンドに対して、スマートメーターは同一の MLME IE を Information Element field に保持する Enhanced Beacon を返す。アソシエーションの手順は省略される。HEMS の、これ以外のデータリンク層設定は 5.8.3.1 節と同じである。

本設定に関する追加情報を表 5-30 に示す。

表5-30: MLME IE の Sub-ID 割当て

Sub-ID value	Content length	Name	Description
0x68	Variable	Unmanaged (ネットワーク識別子)	HEMS が対象メータを特定する為に利用される Sub-ID。本推奨通信仕様で定義。

5.9.6.1.2. ネットワーク層の設定

スマートメーターは、IPv6 リンクローカルアドレスのみを用いる。スマートメーターの、これ以外のネットワーク層の設定は、5.8.2.2 節と同じである。

HEMS も同様に、IPv6 リンクローカルアドレスのみを用いる。HEMS の、これ以外のネットワーク層の設定は、5.8.3.2 節と同じである。

認証手順は、5.9.6.3 節で述べる。

5.9.6.2. IP アドレス検知

PANA による認証手順の前に、HEMS はスマートメーターの IPv6 アドレスの算出を行う。相互のアドレス解

決の方法として、HEMS は、スマートメーターからの Enhanced Beacon の MAC アドレスから、IPv6 リンクローカルアドレスを推定する。
MAC アドレスからの判断であるため、[ND]による Neighbor Discovery は実施しなくてもよい。

5.9.6.3. 認証･鍵交換処理

HEMS は、データリンク層およびネットワーク層での設定の後に、セキュリティの設定を実施する。すなわち、PaC の役割を持つ HEMS が、PAA の役割を持つスマートメーターに対し、PANA のセッションを開始する。

5.9.6.4. アプリケーション

5.5.6 節の記述のとおり、アプリケーションとして ECHONET Lite が用いられ、複合データフォーマットがサポートされる。詳細は[SMHEMSIF]を参照のこと。

5.9.7. クレデンシャルの取扱い(補足)

国内の B ルート(スマートメーター・HEMS 間)ネットワークでは、B ルートに特化するクレデンシャル(表 5-35)が定義されている。この見地から、本節では通信プロトコルにおける当該クレデンシャルの取扱いを述べる。

表5-35: B ルートクレデンシャル

名称	説明
B ルート認証 ID	特定のスマートメーターと HEMS を結びつけるために使用されるユニークな ID。0～9、A～F の ASCII 文字で構成される 32 桁の文字列（32 オクテット長）とする。本仕様書では後述するルールにより PANA（EAP-PSK）で使用する ID（[NAI]形式）やネットワーク識別子に変換される。
（B ルート認証用）パスワード	B ルート認証 ID に結びつけられたパスワード（0～9、a～z、A～Z の ASCII 文字で構成される 12 桁の文字列）。後述するルールにより、[EAP-PSK]で用いる PSK を生成するために使用される。

5.9.7.1. B ルート認証 ID の EAP 認証情報への変換

32 桁の B ルート認証 ID をもとに以下のルールで EAP Identifier（ID_S、ID_P）で使用する NAI を生成する。

【NAI 生成ルール】
スマートメーター側 NAI（EAP ID_S）：”SM”＋”B ルート認証 ID”（34 オクテット）
HEMS 側 NAI（EAP ID_P）：”HEMS”＋”B ルート認証 ID”（36 オクテット）

例：
B ルート認証 ID が「0023456789ABCDEF0011223344556677」の場合、
スマートメーター側 NAI（EAP ID_S）：「SM0023456789ABCDEF0011223344556677」
HEMS 側 NAI（EAP ID_P）：「HEMS0023456789ABCDEF0011223344556677」

5.9.7.2. パスワードの PSK への変換

EAP－PSK で使用する PSK は以下のルールで生成する。

【PSK 生成ルール】
B ルート認証 ID に結びついたパスワードをもとに次の PSK 生成関数（PSK_KDF）を使用して 16 オクテットの PSK を生成する。

PSK = PSK_KDF(パスワード)

= LSBytes16(SHA-256(Capitalize（パスワード))
(パスワード文字列を大文字化し、SHA-256 でハッシュした出力の下位 16 オクテット)

例：
パスワードが「0123456789ab」の場合
PSK = LSBytes16(SHA-256(“0123456789AB”))
= 0xf58d060cc71e7667b5b2a09e37f602a2

5.9.7.3. B ルート認証 ID のネットワーク識別子への変換

HEMS は、自宅のスマートメーターを検出するため、IEs フィールドを用いた Enhanced Active Scan を実施する。HEMS が送信する Enhanced Beacon Request の Payload IEs フィールドに MLME IE(Group ID=0x1)を利用、Sub-ID=0x68(Unmanaged)の IE Contents に、自身が所持する B ルート認証 ID の下位 8octets（ネットワーク識別子）を含めて送信する。スマートメーターは、受信したネットワーク識別子が、自身が持つネットワーク識別子と一致する場合に、Enhanced Beacon を返すことで応答とする。この場合の Enhanced Beacon はユニキャスト送信され、HEMS からの Enhanced Beacon Request と同じ情報を、Enhanced Beacon の Payload IEs フィールドに含める。以上のやり取りで同じ ID を持った装置同士であることを HEMS とスマートメーター間で確認した後に、HEMS は PANA セッションをスマートメーターに対して開始する。(図 5-19)

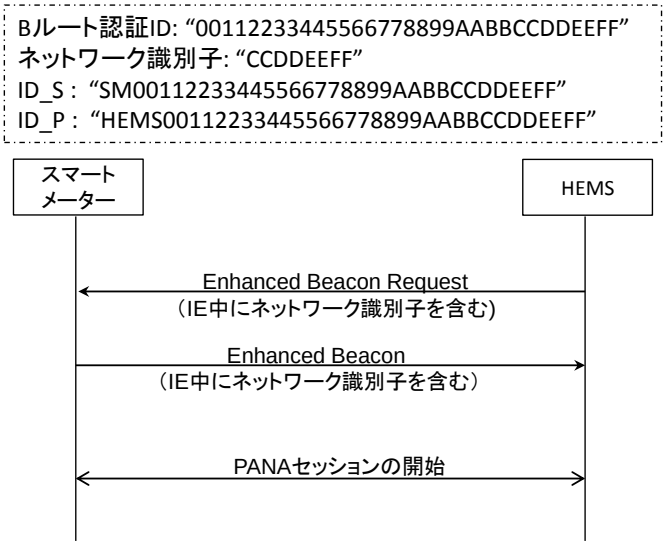


図5-19：スマートメーター探索手順

5.9.8. 推奨仕様を実現するためのデバイス/物理層/MAC 層の仕様

5.9.2、5.9.3を参照のこと。

6. 方式 B

本章には、[ZIP]で規定する 920MHz 用の IPv6 に対応した ZigBee IP 仕様を記載する。なお、ZigBee の公式な認証を取得する場合は、[ZIP]も参照すること。ZigBee IP は以下、ZIP と表記する場合もある。

方式 B では、ZIP コーディネータ, ZIP ルータ, ZIP ホストの 3 種類のノード種別を規定する。ZIP コーディネータがネットワークを管理する役割をする。ZIP ルータはマルチホップネットワークの転送機能を有する。ZIP コーディネータと ZIP ホストでネットワークを構成するとスター型のシングルホップネットワーク、ZIP ルータを加えると、マルチホップネットワークを構成することができる。これら 3 種類のノード種別が実装するプロトコルは[ZIP]で規定されていて、利用者は用途に応じて組み合わせて利用することができる。ベンダ毎にネットワーク形態に応じてプロトコルスタックをカスタマイズする必要はなく、高い相互接続性を提供する。

920MHz 無線は電波到達性が高いため、エリアの小さいホームでは、シングルホップネットワークで構成できるケースもある。ただし、組み込み用の小型アンテナを使用したり、家電等の金属製品の後ろや中に設置する場合や、屋外設置に組み込んで設置したりする場合など、装置の設置場所や設計条件によってはシングルホップでは届かないケースも報告されている。このようなケースでは、マルチホップネットワークに対応可能な方式 B が有効である。

また方式 B はセキュリティやリンクの安定性向上させる機能が既に規定されており、さらには 3 方式の中で唯一グローバルアドレスを使用するため、家の中の他の IP システムや外部の IP ネットワークとの接続性が高い。

さらに方式 B では、ネットワーク層は IETF 規定を単に参照するだけでなく、相互接続性を確実にするために規定の追加を行っているため、本章に従って実装すれば相互接続性が高まる。ZigBee Alliance にて認証を取得することで他システムとの相互接続性が保証される。

6.1. 概要

6.1.1. 目的

ZigBee IP仕様の目的は、IEEE802.15.4を使った無線マルチホップネットワークを使用するために、IETFで定義されたネットワークプロトコルを用いて、標準・相互運用可能なプロトコルスタックを定義することである。

6.1.2. 適用範囲

本章には、ECHONET Lite で使用するための ZigBee IP プロトコルスタックの仕様が含まれている。

本標準は、IETF と IEEE の仕様を利用しており、これらの仕様からの変更部分（「必須機能をオプション機能とする」、「オプション機能を必須機能とする」など）を記載する。

6.1.3. プロトコルスタック概要

ZigBee IP プロトコルスタックは、以下の図で説明されている。

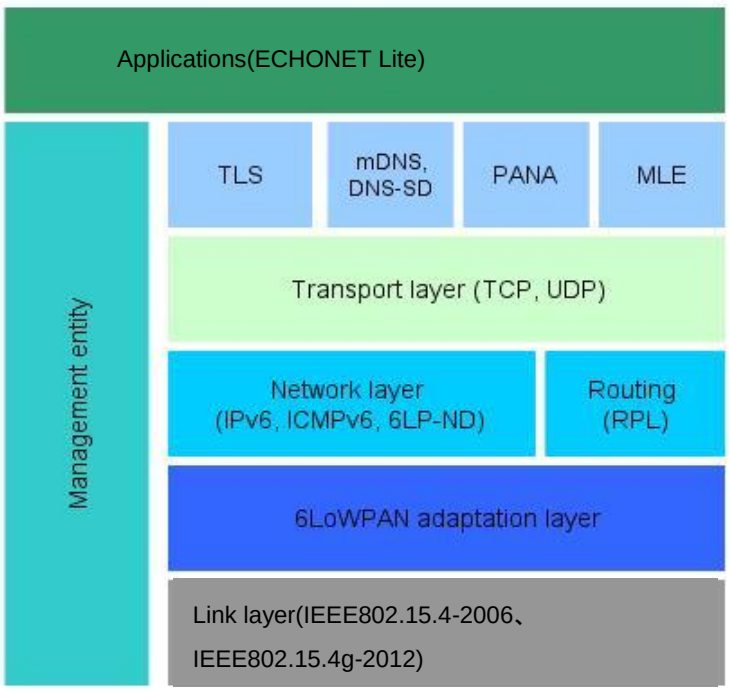


図6-1:ZigBee IP プロトコルスタック

データリンク層は、次のサービスを提供する。

- 無線到達距離内における、IEEE802.15.4 PAN の発見
- MAC ペイロードの最大サイズ(別途規定)の転送。各フレームの実際の MAC ペイロードは、モード、セキュリティオプション 及び アドレッシングによって異なる。
- フレームのバッファリングとポーリングを使用して、眠っているデバイスへのフレーム伝送のサポート。
- 暗号化・認証・再送保護を含むフレームセキュリティ。鍵管理はこの層で実行されていないことに注意すること。

6LoWPAN を利用するアダプテーション層は、次のサービスを提供する。

- IPv6 および UDP ヘッダのヘッダ圧縮と解凍。
- リンクレイヤフレームの中で可能な最大ペイロードを超える IPv6 パケットのフラグメンテーションおよび再構成。

ネットワーク層は、次のサービスを提供する。

- IPv6 アドレス管理とパケット・フレーミング
- ICMPv6 メッセージ
- ルータ 及び 近隣探索
- IPv6 ステートレスアドレス自動設定 及び 重複アドレス検出(DAD)
- 6LoWPAN 構成(コンフィグレーション)情報の伝播
- RPL プロトコルを使用しての経路計算とメンテナンス
- IPv6 のパケット転送

・　サブネット内の IPv6 マルチキャストフォワーディング

トランスポート層は、次のサービスを提供する。
・　保証されたパケットと保証されていないパケットの配信サービス
・　複数アプリケーションのためのパケットの多重化

マネジメント・エンティティは、ノードによって希望された運用上の動作を達成するために、様々なプロトコルを起動させ管理することに責任を持つ概念的な機能である。これは以下について責任を負っている：
・　ノードのブートストラップのプロセス
・　ノードのパワーマネージメント
・　重要なネットワークパラメータの不揮発保存と回復
・　PANA プロトコルを使用して認証とネットワークアクセス制御
・　PANA プロトコルを使用してネットワーク全体への鍵配布
・　MLE プロトコルを使用してネットワーク・コンフィグレーション・パラメータの伝播

6.1.4.　ドキュメントの構成

ドキュメントの残りの部分は次のように構成されている。第6.2節は、ZigBee の IP プロトコルの仕様を記述している。それは、それぞれ必須 および オプション機能を詳細に定義された ZigBee の IP 実装によってサポートされる必要がある、様々な IEEE と IETF 標準プロトコルについて説明する。第6.3節では、ネットワークオペレーションの様々な段階での ZigBee IP ノードの機能動作について説明する。第6.4節では、有益な材料と、この仕様の実装に役に立つかもしれないプロトコルのメッセージ交換の例が含まれている。

920MHz 対応のための変更条件は、第6.6節にまとめて記載した。また、物理層とデータリンク層の実装規定については、第6.7節に記載した。

また外部ドキュメントの参照については、「3 章参照規格・参考文献」に記述されている[802.15.4]のように、この章で規定されている文書を参照するものと、[RFC 4944] のように IETF などの機関のよく知られている文書番号を直接参照するものがある。

6.2.　プロトコル仕様

6.2.1.　物理層

ZigBee IP ノードは、IEEE802.15.4-2006 および[802.15.4]および IEEE802.15.4g-2012 で定義されている物理層の仕様に準拠する物理インタフェースを、少なくとも一つサポートしなければならない。

本標準書では、単一物理インタフェースのみをサポートし、マルチ物理インタフェースは未サポートとする。

6.2.2.　データリンク層

ZigBee IP ノードは IEEE802.15.4-2006 [802.15.4] データリンク層の仕様を実装しなければならない。ZIP ホストは最低でも RFD (reduced function device)機能を実装しなければならない、 一方 ZIP ルータと ZIP コーディネータは FFD (full function device)機能を実装しなければならない。

ZIP ノードは利用できるすべてのデータリンク機能を実装する必要はない。具体的には、ビーコンモードと保障タイムスロット(GTS)機能は、ZigBee IP ネットワークでは必要ない。Association と Disassociation コマンドフレームはサポートする必要はない。

ZIP ノードは、本ドキュメントの第6.4節で説明されるデータリンク層セキュリティ機能をサポートする必要がある。

ZigBee IP ノードは 64-bit と 16-bit のデータリンク層アドレッシングモードをサポートしなければならない。EUI-64 アドレスが製造時に各デバイスに設定されていなければならない。このアドレスはグローバルにユニークであり、デバイスに対して生涯固定であることが期待されている。16 ビットのショートアドレスはネットワーク参加が完了した後、各デバイスに割り当てられなければならない。このアドレスは特定の IEEE802.15.4 PAN 内でユニークである。

6.2.3. アダプテーション層

6LoWPAN を利用するアダプテーション層は、IETF の 6LoWPAN Working Group で制作された規格で定義されている。

[6LOWPAN] と [6LPHC]の指定に従って、IPv6 パケットの IEEE802.15.4 フレームへのカプセル化が実行されなければならない。mesh addressing header はサポートする必要がない。ZigBee IP は[6LOWPAN]に記載されているリンク層のメッシュアンダールーティングを使わず、その代わりにルートオーバーの構成(コンフィグレーション)をとるためである。

6.2.3.1. 6LoWPAN フラグメンテーション

6LoWPAN のフラグメンテーション方式は、[6LOWPAN]で定義されており、サポートされなければならない。

単一の IP データグラムを構成するフラグメントは、データグラムオフセットの増加順で送信されなければならない。さらに、1 つのデータグラムのフラグメント送信は同じ宛先において、フラグメント化された他のデータグラムと混合(インタリーブ)されてはならない。([6LOWPAN]は、フラグメントパケットを任意の順序で送信することができるが、フラグメントが順番に到着すること、インタリーブの心配なしで再構築できること、はデータの再構成と失われたフラグメントの検出の両方の処理を簡素化する。ZigBee の IP で使用される物理層とデータリンク層は、パケットの順序を変更しない。よって、上記の制限は順序が決まったパケットの到着を十分保証出来る。)

6LoWPAN インタフェースのリンク MTU は 1280 オクテット(例外については6.2.4.3を参照)に設定されなければならない。

6.2.3.2. ヘッダ圧縮

ヘッダ圧縮[6LPHC]に定義されている 6LoWPAN ヘッダ圧縮方式は、ZigBee IP ノードによってサポートされなければならない。ZigBee IP ノードは、[6LPHC]で定義されているすべての圧縮モードをサポートしなければならない。IPv6 パケットを送信する場合、最も効果的な圧縮方式が送信パケットのサイズを最小限にするために使用されるべきである。ノードは、ヘッダが[6LPHC]で定義された書式を使用してエンコードされている限り、どんな、あるいはヘッダ圧縮無しの場合でも、IPv6 パケットを受信できるべきである。

[6LPHC]は、IPv6 アドレスの圧縮の目的のために、事前に定義されたコンテキスト識別子を使用することを規定している。これらのコンテキスト識別子は 6LBR で定義され、ルータ通知を介してネットワーク内のノードに伝達される[6LPND]。

ZigBee の IP ネットワーク内の 6LBR は、IP ヘッダ圧縮の目的のために MIN_6LP_CID_COUNT 以上のコンテキスト識別子を定義してはいけない。第6.1節で定義されているように、6LBR は、デフォルトのコンテキスト識別子(コンテキストゼロ)を定義し、6LoWPAN に割り当てられた IPv6 プレフィックスをその値に設定しなければならない。

他のすべての ZIP ノードは、IPv6 ヘッダ圧縮の目的のために、そのコンフィグレーションと少なくとも MIN_6LP_CID_COUNT 個のコンテキスト識別子の使用をサポートしなければならない。

6.2.3.3. 近隣探索

6LoWPAN neighbor discovery 仕様 [6LPND]で定義されている近隣探索プロトコルが実装されなければならない。

ZigBee IP ノードは、prefix とコンテキスト情報のマルチホップ配布のために、[6LPND]で定義されたオプションメカニズムをサポートしなければならない。

ZigBee IP ノードは、マルチホップでの重複アドレス検知のために、[6LPND]で定義されたオプションメカニズムをサポートしなければならない。

後述のセクションで規定される上位層プロトコルに、新しい近隣ノードの検出に加え、近隣ノードの双方向到達可能性を検出する定期的なパケット送信が含まれているので、Zigbee IP ノードは近隣不到達プローブを抑制するべきである。 ただし、全てのノードは近隣不到達プローブに適切に応答しなければならない。

6.2.4. ネットワーク層

ZigBee IP ノードは IPv6 プロトコル[IPv6]をサポートしなくてはならない。

ZigBee IP ノードは、認証ヘッダ(AH)および Encapsulating Security Payload(ESP)の IPv6 拡張ヘッダをサポートする必要はない。よってこの動作モードに関する記述は本標準書にはない。

ZigBee IP ノードは、フラグメント IPv6 拡張ヘッダをサポートする必要はない。

ZigBee IP ノードは、ICMPv6 プロトコル[ICMP6]をサポートしなければならない。ノードはエコー要求とエコー応答メッセージに加えて、ICMPv6 エラーメッセージをサポートしなければならない。

6.2.4.1. IP アドレッシング

全ての ZigBee IP ノードは[IP6ADDR]で規定される IPv6 アドレッシング・アーキテクチャーをサポートしなければならない。

ZigBee IP ネットワークは、1 つ または 複数の /64 ビット プレフィックスを割り当てられる。これは、6LoWPAN 内全体を通して、プレフィックスとしてアナウンスされる。([6LPND]参照)。これらのプレフィックスは、ULA[ULA] または GUA のどちらかのプレフィックスで良い。また一つのノードは最低でも MIN_6LP_PREFIX 個のプレフィックスをサポートする能力がなくてはならない。[ND]、[6LOWPAN] 及び 他の規格との整合性を保つために、6LoWPAN プレフィックスは、常に/64 ビットでなくてはならない。6LoWPAN ノードは、[6LOWPAN]のセクション 6 で定義されているように、インタフェース識別子を得るために、EUI-64 アドレス または、16 ビットのショートアドレスのいずれでも使用することができる。インタフェース識別子を作成するために 16 ビットショートアドレスを使用する場合は、[6LPHC]で述べられている方法に従わなければならない。16 ビットのショートアドレスに基づいてヘッダ圧縮モードに適用した場合、デフォルトコンテキストからの/64 ビットプレフィックスと 64-bit IID が 16bit ショートアドレスから変換されるところの追加の 48 ビットが、圧縮されたアドレス中から省略される。

ZigBee IP ノードは IEEE802.15.4 インタフェースを少なくとも以下のアドレスで構成しなければならない。

- [SLAAC]と[6LOWPAN]で記述されているように、well-known link-local prefix FE80::0/64 を使用して、インタフェースの識別子としてノードの EUI-64 から 128-bit link-local IPv6 address を構成する。[6LPHC]を使ってこのタイプのアドレスが圧縮される場合、ステートレスの圧縮が考慮されなければならない。このタイプのアドレスは LL64 として知られている。

- [SLAAC]と[6LOWPAN]で記述されているように、well-known link-local prefix FE80::0/64 と、ノードの 16-bit short address から作られるインタフェース識別子から 128-bit link-local IPv6 address を構成する。[6LPHC]を使ってこのタイプのアドレスが圧縮される場合、ステートレスの圧縮が考慮されなければならない。このタイプのアドレスは LL16 として知られている。

- 一つあるいはそれ以上の 128-bit ユニキャスト IPv6 アドレス。アドレス設定のために使われるインタフェース識別子は、ノードの 16-bit ショートアドレスである。ULA または、GUA プレフィックスは、ルータ通知にある 6LoWPAN Prefix information option (PIO)から得られる ([6LPND])。複数のグローバルプレフィクスがアドバタイズ(通知)されている場合、ノードはローカル･ノードのポリシーに基づいてそれらのいずれか、またはすべてのアドレスを構成することを選べる。[6LPHC]を使ってこのタイプのアドレスが圧縮される場合、ステートフルのコンテキストベースの圧縮が考慮さるべきだ。このタイプのアドレスは GP16 として知られている。

加えて、全てのノードが[ND]で要求されている適切なマルチキャストアドレスに参加しなければならない。

DAD は[6LPND]で推奨されているように、EUI-64 インタフェース識別子から構成されるアドレスで実行されてはならない。16 ビットのショートアドレスから構成されている GP16 アドレスはユニークかどうかを DAD メカニズム[6LPND]でテストされなければならない。

6.2.4.2. ルーティング・プロトコル

全ての ZigBee IP ルータは RPL ルーティング・プロトコル[RPL]を実装しなければならない。RPL は DODAG ルートと呼ばれるルートノードに向かって宛先指向の有向非巡回グラフ(DODAG)を確立する。パケットはこのグラフを使って、ルートに向かって DODAG を上るよう指示される。パケットは、Destination Advertisement Object (DAO)によって確立された経路を使って、ルートから DODAG を下るよう指示される。以下のサブセクションは RPL が ZigBee IP でどのようにデバイス間の互換性を確保するのに使われているか述べる。

ZigBee IP ネットワークは同時に複数の RPL インスタンスを動作させても良い。グローバルインスタンスだけが使用されるべきである。LBR ノードは RPL を開始しなければならない。他の ZigBee IP ルータは、外部ネットワークの接続を提供している場合、又は、その様に役割を与えられている場合は、独自の RPL インスタンスを開始しても良い。このケースでは、その RPL インスタンス識別子は、既存の識別子と衝突しないように選択されるべきである。これはルータが最初にネットワークに参加し、独自 RPL を開始する前に、既存 RPL インスタンスを検索するすべきということである。異なる DODAG id フィールドを持っているが、同じ値のインスタンス id フィールドを持った DIO の存在はインスタンス id の重複を示している。DODAG ルートがインスタンス id の衝突を検出した場合、異なるインスタンス id で DODAG を再形成するべきである。

ZigBee IP ルータは最低でも MIN_RPL_INSTANCE_COUNT 個の RPL インスタンスの参加が可能であることが必須であり、メモリ制約に従いネットワークで利用可能なすべての RPL インスタンスに参加すべきである。

もしノードが RPL インスタンスとの接続を RPL_INSTANCE_LOST_TIMEOUT 秒以上失ったとき、(つまり、有限ランクの親を見つけられなかった時)インスタンスを削除すべきである。これは例えば、インスタンスのルートが置き換えられた時起きるかもしれない。

各 DODAG ルートは経路情報オプション(RIO)に 0 個以上の prefix を含むよう構成されても良い。もしルートがデフォルト経路(prefix 0::)を通知したい場合は RIO にそれを含むことが必須になることに注意する。すべての RIO prefix が存在しない場合、その DODAG がそのルートノードに向けてのみルーティングできることを示す。もし DODAG ルートが Authoritative Border Router[6LPND]である場合には、RPL DIO パケットとルータ通知パケットの両方に PIO 情報を含めなければならない。

ZigBee IP ネットワークでは、一つの RPL インスタンスは一つのルートにおいての中で一つの DODAG を含まなければならない。DODAG ルートは常に接地(grounded)されている。フローティング DODAG が使用されてはならない。

RPL コントロールメッセージは RPL セキュリティモードの “unsecured” を使って送られる。セキュリティ要件を満たすためにリンク層セキュリティが使われる。

ZigBee IP ネットワークでは、non-storing RPL mode の操作のみ使われる。non-storing mode では、全ての下向き経路は DODAG ルートによってソース経路として管理される。ルータは下向き経路情報を含んだ DAO メッセージを、DAO-ACK (‘K’) flag を有効にしてルートに直接送る。DAO メッセージは各ホップで遅延されない([RPL] section 9.5 参照)。DAO メッセージは複数ノードが同時にルートへ送ることを避ける為に、発信元ルータによってジッタを持たされるべきである。マルチキャスト DAO メッセージは ZigBee IP ネットワークでは使われない。

ルートでない全てのルータは DODAG 一つに対して、ルータ自身による上向きルーティングと、ルートによる下向きルーティングに使われる最低 RPL_MIN_DAO_PARENT 個の親を持つことがサポートされるべきである。

Metric Container と RPL Target Descriptor オプションは RPL コントロールメッセージには含まれてはならない。

6.2.4.2.1. ホストの RPL への参加

ZIP ホストは、RPL プロトコルに参加しない。

6.2.4.2.2. Objective Function

objective function は RPL インスタンス内の経路選択目標を定義している。objective function は DODAG configuration option の objective code point (OCP)フィールドによって確認される。

ZigBee IP ルータは metric containers を使わずに ETX metric を使用する MRHOF objective function [RPL-MRHOF]を実装しなければならない。

Zigbee IP ルータは近隣ルータへのリンクの ETX を決定する為に Mesh Link Establishment protocol [MLE]を使わなければならない。ルータは近隣テーブル内の各近隣ノードについて受信成功率を評価する。評価方式は実装依存である。受信成功率の逆数が、MLE Neighbor TLV を介して近隣へ伝達される。リンクの ETX は順方向と逆方向の受信成功率の逆数の積に等しい。

MRHOF パラメータは以下のように設定されなければならない。

MAX_LINK_METRIC: 16 * MinHopRankIncrease.

MAX_PATH_COST: 256 * MinHopRankIncrease.

MIN_PATH_COST: 0.

PARENT_SWITCH_THRESHOLD: 1.5 * MinHopRankIncrease.

PARENT_SET_SIZE: 2.

ALLOW_FLOATING_ROOT: 0.

6.2.4.2.3. RPL 構成

このセクションでは RPL 構成と ZigBee IP での使用される RPL コントロールメッセージを指定する。指定されない設定は全て[RPL]定義の通り使われる。

DODAG ルートは DIO を介していくつかの情報を設定する権限があり、その情報はリーフノードに向かって伝播中に変更されない。この情報を以下に示す。

1. RIO(複数可)

2. DODAG configuration option

3. PIO(複数可)、 'R'フラグが設定されている場合を除き、Prefix フィールド内の IPv6 アドレスの最後の 2 オクテット(リンク層ショートアドレス)は変更される。

4. RPLInstanceID

5. DODAGID

6. DODAGVersionNumber

7. Grounded flag

8. Mode of operation field

6.2.4.2.3.1. DODAG Information Solicitation(DIS)フレームフォーマット

DIS のメッセージは、Pad1、PadN または Solicited Information (要請情報)オプションを含めても良い。

ZIP ルータは、特定の RPL インスタンスへの DIO 応答を制限するために、Solicited Information オプションとインスタンス ID 述部を持つ DIS メッセージを送信しても良い。

6.2.4.2.3.2. マルチキャスト DODAG Information Object(DIO)フレームフォーマット

マルチキャスト DIO メッセージは DIO ベースオブジェクトと RIO オブジェクトを含む。

DIO ベースの設定は以下の通り。

- RPLInstanceID は[0x00, 0x7F]の範囲の値でグローバルインスタンスに設定されるべきである。
- Version Number は 初期値 0xF0 に初期化されるべきである。
- Grounded (G) : DIO の Grounded flag は常にセットされなければならない。ZIP ノードは floating DODAG を作ってはならない。
- Mode of Operation (MOP) :DIO の動作モード(MOP)フィールドは 0x01 に設定しなければならない。これは、RPL の non-storing モードを示す。
- DODAGPreference ： DODAGPreference フィールドは 0 に設定するべきである。 ZIP ルータは、このフィールドに基づいて DODAG プリファレンスを実装する必要はない。
- ディスティネーション・アドバータイズメント・トリガ･シーケンス番号(DTSN)－ ルートノードは、DODAG バージョン番号をインクリメントせずにネットワークから新しい DAO メッセージを受信したい場合、その DIO の DTSN フィールドをインクリメントする。ZIP ルータは、DTSN カウンタに親ルータと同じ値をセットする必要があり、親ルータがその値を更新した場合はいつでも同じ値に更新しなければならない。こうして、ルートノードはその DTSN フィールドをインクリメントし、その変化を DoDAG 全体に伝播することができる。

RIO の設定は以下の通り：

- プレフィックス長はその経路が通知されているプレフィックスの長さに設定すべきである。
- 経路優先度(Prf)の値は 0(中度 medium)優先か管理者のコンフィギュレーションに設定するべきである。
- プレフィックスはその経路が通知されている値に設定するべきである。

RPL は、その Root を通して到達可能な外側の経路を通知するために、DIO のフレームに複数の RIO オプ

ションを含むことをその Root に許可する。RPL root として動作している ZIP ノードは、DIO パケットに含まれる RIO オプションの数を RPL_MAX_RIO 個以下に制限するべきである。これによってすべての ZIP ルータが必要な経路情報を処理できるようにする。同様に、RPL　Root は DIO パケットに含まれる PIO オプションの数を RPL_MAX_PIO 個以下に制限するべきである。

6.2.4.2.3.3.　ユニキャスト DODAG Information Object(DIO)フレームフォーマット

ユニキャスト DIO メッセージは DIO ベース、RIO、PIO と DODAG configuration option を含んでいる。ユニキャストで使われる DIO ベースと RIO は、マルチキャストメッセージと同じフォーマットを持っている。

PIO の設定は以下の通り：

- プレフィックス長が 64 ビットであることを示す 0x40 に設定されなければならない。
- 'L'フラグ(オン・リンク・フラグ)はセットしてはいけない。([6LPND]6.1 参照)
- 'A'フラグ(自律アドレス設定フラグ)は、そのプレフィックスがステートレスなアドレス自動コンフィギュレーション用になれる場合は、セットされなければならない
- 'R'フラグ(ルータアドレスフラグ)は、もしそのノードがこのプレフィックスでアドレスを生成していた場合、セットしなければならない。それ以外は設定してはいけない。.
- Prefix フィールドは、ソースノードのルーティング可能な IPv6 アドレスを含めなくてはならない。

DODAG configuration option の設定は以下の通り　：

- 認証有効(A)フラグはセットしてはならない。ZigBee IP は、RPL セキュリティを使用ぜず、代わりにデータリンク層のセキュリティを利用する。
- パスコントロールサイズ(PCS)フィールドは 2 を設定されなければならない。これは、DAO の親の数と ZIP ノードに設定されている下向きの経路の数を制御する。
- DIO 転送を支配するトリクル･パラメータは、RPL root によって設定されるべきである。このパラメータは、参加の開始時間に対するトリクルタイマリセットによって生成されるトラフィック量のバランスをとるために設定されるべきである。以下のパラメータ値が推奨される。
    - DIOIntervalDoublings 値は 12 に設定されるべきである。
    - DIOIntervalMin 値は 9 に設定されるべきである。
    - DIORedundancyConstant 値は 3 に設定されるべきである。

    ZIP ルータは、受信した DODAG 設定オプションに基づいて内部 DIO トリクルタイマのパラメータを設定しなければならない。上記の推奨値をハードコーディングしてはいならない。
- MaxRankIncrease フィールドは 0 以外の値に設定するべきである。 MaxRankIncrease は、ローカル修復への支援の許容ランクの増加を設定するために使用される。もし 0 に設定すると、ローカル修復は無効になる。このフィールドへの典型的な設定値は約 16 であり、より多くのホップ数のネットワークではより大きな値にすべきである。.
- MinHopRankIncrease フィールドは 0x80 に設定するべきである。

- Object Code Point(OCP)は[RPL-MRHOF]で割り当てられた値に設定しなければならい。

6.2.4.2.3.4. Destination Advertisement Object(DAO)フレームフォーマット

ユニキャスト DAO リクエストは下向きの経路を確立する為に DODAG ルートノードに送られる。このリクエストは DAO ベース、RPL target option と Transit information option で構成される。

DAO ベースの設定は以下の通り ：

- RPLInstanceID : [0x00, 0x7F]の範囲の値を持つグローバル RPLInstanceID でなければならない。
- ‘K’ flag ： セットされるべきである。このフラグは DODAG ルートが DAO-ACK を送り返すことを期待されていることを示す。
- ‘D’ flag : クリアされなければならない。ローカル RPLInstanceIDs は使用しないため。
- DAOSequence は初期に 0xF0 にセットされ、その後“lollipop” fashion にインクリメントされるべきである。DAO-ACK が無かったために DAO を再送するときは、ノードは DAO シーケンス番号をインクリメントするべきである。

最低でも一つの RPL target option が DAO リクエストに存在しなければならない。RPL target option は、DODAG ルートにターゲット IPv6 アドレスまでの経路が存在することを知らせるために使われている。

RPL target option の設定は以下の通り ：

- IPv6 アドレスがターゲットのプレフィックスに存在しているので、プレフィックス長は「0x80」に設定するべきである。
- ターゲットのプレフィックスは、DAO ルータ に送ろうとしている ZIP ルータの IPv6 アドレス、または、そのルータから直接到達可能な ZIP ホストの IPv6 アドレスにのいずれかに設定するべきである。

Transit information option は、DODAG ルートへの DODAG 親の通知に使われる。Transit information option の設定は以下の通り：

- External (E) フラグは、ターゲットプリフィックスが DAO パケットを送ろうとしている ZIP ルータの IPv6 アドレスを含む時 0 にセットしなければならない。それ以外は 1 にセットされなければならない。
- Path Control フィールドは、DAO リクエストに含まれる DODAG 親の数を制限するためと、それらの間の優先順位を設定するために使われる。
- Path Sequence は、それぞれ新しい DAO パケットに対して更新されるべきである。
- Path Lifetime は、DAO 親が有効である存続期間が設定されなければならない。 それは ZIP ルータが DODAG ルートでその下方ルーティングテーブルエントリから既存の DAO の親を削除したい場合はゼロに設定しなくてはならない。
- 親アドレスは、Transit Information オプションに一つ存在するべきであり、DODAG 親の IPv6 アドレスまたは、DAO をホストの代わりに送信されたときに要求を生成したノードの IPv6 アドレスが含まれなければならない。複数の親アドレスが、複数の Transit オプションを用いて伝達されても良い。

RPL ルートは、ソースルートエントリを更新する前に DAO パケットから受け取るルーティング情報の新鮮さを確認する。DAO がホストノードのルート情報を伝えるとき、'E'フラグの設定によってそれが示されるが、ルートはその新鮮さを示す time-of-delivery を使わなければならない。すなわち、時間的に後に到着する DAO はもっと新しい経路情報を持っているとみなされる。それ以外の場合は、ルートは time-to-delivery、DAO シーケンス、パスシーケンスの値の組み合わせを用いて、自由に判断出来る。

6.2.4.2.3.5. Destination Advertisement Object ACK (DAO-ACK)フレームフォーマット

DAO-ACK リクエストは、DAO リクエストを生成するノードに DODAG ルートから送信される。ルートはそのシーケンス番号にかかわらず受信した DAO パケットそれぞれにアクノリッジしなければならない。

DAO-ACK の設定は以下の通り ：

- RPLInstanceID フィールドはそのインスタンスに設定されなけれななならない。
- 'D'フラグは、ローカル RPL インスタンスが使用されていない時、ゼロに設定するべきである。
- TDODAGID フィールドは"D"フラグがゼロであるときは存在しない。

6.2.4.3. IP トラフィック転送

宛先ノードが直接到達可能であると分かっている場合は、ZIP ルータが宛先に直接ユニキャストパケットを転送することができる。それ以外の場合は、RPL プロトコルで定義された転送ルールを使用したユニキャストパケットを転送するべきである。

RPL プロトコルでは、RPL ドメインで転送されるすべてのデータパケットは、RPL オプション[RPL-OPT]、または RPL ソースルート[RPL-HDR]ヘッダのいずれかを含まなければならない。

ソースルーティングヘッダは、RPL インスタンスの DODAG ルートによって、挿入のみされてもよい。ソースルーティングは、①DODAG の外部で生成され DODAG ルートを通して配信される P2MP(point to multipoint)トラフィックと、②送信元から DODAG を上りの向きにルートへ転送されその後、DODAG を宛先まで下りの向きに転送される、P2P(point to point)トラフィックに使用される。DODAG ルートは、DODAG 内のノードに IPv6 トラフィックを転送するために、RPL DAO パケットに含まれる情報により生成される、ノード特有の経路情報を使用する。DODAG ルートが送信を開始するか、または DODAG 内のノードのいずれかの宛先アドレスを持つ IPv6 データグラムを受信すると、ルートは、[RPL-HDR]に従って、IPv6 データグラムにソースルーティング情報を追加する。

DODAG ルートは、自らが IPv6 パケットの送信元であり、送信先が RPL ドメイン内にある（すなわち、同じプレフィックスを持つ ZIP ルータである）場合にのみ直接、ソースルーティングヘッダを挿入するべきである。その他の全ケースでは、″IPv6-in-IPv6 トンネリング″を使用しなければならない。そのノードが RPL ドメイン内であれば、トンネルの出口点は最終の宛先アドレスに設定しなければならない。それ以外の場合は、宛先の親のアドレスに設定しなければならない。DODAG ルートは、宛先アドレスに対応するターゲット･オプションを持つ DAO パケット内のトランジットインフォメーションオプションから親のアドレスを決定する。

ユニキャスト IPv6 パケットを生成、および RPL プロトコルを介してそれを転送している ZIP ルータは、RPL オプションヘッダを挿入しなければならない。ヘッダは、宛先アドレスがパケットで使用される RPL インスタンスの DODAG ルートである場合を除いて、IPv6 ベースのすべてのケースでトンネリングを使用して挿入しなければならない。その場合、ヘッダはパケットに直接挿入されるか、 または、"IPv6-in-IPv6"トンネリングのどちらかを使用して挿入されてもよい。RPL オプションヘッダがトンネリングを使用して挿入されるとき、トンネル出口ポイントは DODAG ルートへの経路に沿って次のホップアドレスに設定されるべきである。パケットの最終宛先アドレスがパケットで使用される RPL インスタンスの DODAG ルートであ

る場合には、トンネルの出口ポイントはそのアドレスに設定されてもよい。

もしパケットが既に RPL オプションのヘッダまたはソースルーティングヘッダのいずれかも含んでいない場合、別のノードによって発信されたユニキャスト IPv6 パケットを転送するために RPL を使用している ZIP ルータは、RPL オプションヘッダを挿入しなければならない。ヘッダは、″IPv6-in-IPv6″のトンネリングを使用して挿入されなければならない。トンネル出口ポイントは DODAG ルートへの経路に沿って次のホップアドレスに設定されるべきである。パケットの最終宛先アドレスがパケットで使用される RPL インスタンスの DODAG ルートである場合には、トンネルの出口ポイントは、オプションでそのアドレスに設定することも出来る。

ZIP ノードは、RPL の拡張ヘッダが直接かまたは IPv6-in-IPv6 トンネリングかのどちらかで挿入された際に、そのために IPv6 のフラグメンテーションが生じない事を確認しなくてはならない。これは IPv6 ヘッダが一つの RPL 拡張ヘッダを含んでいるパケットに対して、異なった MTU 値を使用することで行われる。RPL トンネルエントリポイントは、その MTU が 6LoWPAN インタフェースの MTU に RPL_MTU_EXTENSION オクテットを加えた値に設定されている別のインタフェースとして考慮されるべきである。

ZIP ホストノードは、そのデフォルトの親ルータにパケットをフォワードすべきである (([6LPND]に記載あるように、そのホストがアドレスを登録したルータとなる)。もし親ルータが RPL の転送ルールを使用してパケットを転送する必要があると判断した場合、前述のルールに従って必要な RPL 拡張ヘッダを挿入する。

6.2.4.4. マルチキャスト転送

マルチキャストスコープ値の 3 [IP6ADDR]は、単一ネットワーク内の全てのリンクと、ZIP ノードの全てのインタフェースを含む「サブネットーローカル」スコープとして定義される。そのようにして、ZIP ネットワークは、スコープ値 3 のサブネットーローカルマルチキャストゾーン[RFC4007]を形成する。

全ての ZIP ノードは、ZIP インタフェース上で、subnet-scope-all-nodes（サブネット上のすべてのノードからなる）マルチキャストグループ (FF03:0:0:0:0:0:0:1)と subnet-scope-all-mpl-forwarders（サブネット上のすべての MPL 転送ノード）に接続しなければならない。全ての ZIP ルータは ZIP インタフェース上で subnet-scope-all-routers （サブネット上のすべてのルータからなる）マルチキャストグループ (FF03:0:0:0:0:0:0:2)に接続しなければならない。ZIP ノードは管理上のコンフィギュレーションに基づいて追加的なサブネットをスコープとしたマルチキャストグループに接続してもよい。

ZIP ノードはマルチキャスト IP パケットを送付するために MPL プロトコル[MPL]を使用する。全ての ZIP ノードはその ZIP インタフェースを MPL インタフェースとして設定する。全ての ZIP ノードは MPL データメッセージを発信したり受信して良く、ZIP ルータはまた他ノードに MPL データメッセージを転送しても良い。

MPL プロトコルはそれぞれの転送ノードに対し少なくとも一つの subnet-scope-all-mpl-forwarders グループによって指定される MPL ドメインに参加することを要求している。加えて、ZIP ノードは ZIP インタフェース上で加入するそれぞれのサブネットをスコープとしたマルチキャストアドレスによって指定される MPL ドメインに参加しなければならない。

ZIP ノードは MPL パラメータを以下のように設定しなければならない。

- PROACTIVE_PROPAGATION フラグは true にセットされなければならない。これは MPL 転送が能動的に行われることを意味する。
- DATA_MESSAGE_IMIN = 512ms
- DATA_MESSAGE_IMAX = 512ms
- DATA_MESSAGE_K = 無限大

- DATA_MESSAGE_TIMER_EXPIRATIONS = ZIP ホストは 0、それ以外は 3

- CONTROL_MESSAGE_TIMER_EXPIRATIONS = 0

DATA_MESSAGE_TIMER_EXPIRATION パラメータを ZIP ホストで 0 に設定することにより MPL データメッセージの転送や再送がディセーブルになることに注意。同様に CONTROL_MESSAGE_TIMER_EXPIRATION パラメータを全ての ZIP ノードで 0 に設定することは MPL control message が一つの ZIP ネットワークでは送信されないことを意味する。

MPL データメッセージは IPv6 Hop-by-Hop ヘッダの中に MPL オプションを含む。ZIP ノードは MPL オプションを以下のように設定しなければならない。

- S フィールドの値は 1 にしなければならない。それは seed-id が 16bit の値であることを示す。

seed-id フィールドの値は MPL データメッセージを発信するノードの MAC ショートアドレスにセットされなければならない。

6.2.5. トランスポート層

6.2.5.1. コネクション型サービス

全ての ZigBee IP ノードは[TCP]で定義される TCP(Transmission control protocol) プロトコルをサポートしなければならない。

6.2.5.2. コネクションレス型サービス

全ての ZigBee IP ノードは[UDP]で定義される UDP (User Datagram Protocol) プロトコルをサポートしなければならない。

6.2.6. PANA

参加ノードとネットワーク認証サーバ間の認証データを運ぶための EAP トランスポートとして、ネットワークアクセス認証プロトコル[PANA]を使わなければならない。このセクションでは、[PANA]と[PANA-ENC]で規定されたものに加え、制約と仕様の定義を明確化する。

6.2.6.1. PRF(擬似乱数生成関数), メッセージ認証および暗号化アルゴリズム

以下のアルゴリズム識別子のみ使用しなければならない:

表6-1:PANA アルゴリズム識別子

アルゴリズム	タイプ	値	コメント
PRF	PRF_HMAC_SHA2_256	5	IKEv2 Transport Type 2
AUTH	AUTH_HMAC_SHA2_256	12	IKEv2 Transport Type 3
Encryption	AES-CTR	1	

提案した PRF と SHA-256 に基づく AUTH ハッシュは[IKEv2]に記載され、[IPSEC-HMAC]に詳細が書かれている。提案した Encryption は[PANA-ENC]で使用される

6.2.6.2. ネットワークセキュリティマテリアル

PANA プロトコルは、ZigBee IP ネットワークにおいて、認証サーバから各認証されるノードへのネット

ワークセキュリティマテリアルを伝送するために使用される。このセキュリティマテリアルは、さらに他のプロトコルのセキュリティを提供するために使用される暗号化キーを導出するために、各ノードで使用される。ネットワークセキュリティマテリアルは、次のパラメータで構成される。

表6-2:ネットワークセキュリティマテリアル

パラメータ	サイズ	コメント
Network Key	16 octets	PANA を使用する際に、認証サーバによってネットワーク内で認証される全 ZIP ノードへ転送される、ネットワーク全体の共通ネットワークキー
Key sequence number	1 octet	ネットワークキーに関連付けられたシーケンス番号
Node Auth Counter	1 octet	各ノードで使用する認証カウンタの値。このパラメータは、ネットワーク内のノードごとに一意である。

ネットワークキーは、ネットワーク認証サーバによって所有され、管理される。各ネットワークキーは、1～255 のシーケンス番号を持つ。ネットワーク認証サーバは、ネットワークキーとネットワークキーに関連付けられたシーケンス番号の更新を管理し、現在どのネットワークキーがアクティブであるかを定義する。

さらに、認証サーバは、ネットワーク内の各ノードの認証カウンタのパラメータを管理する。ネットワークキー、キーシーケンス番号、認証カウンタの組み合わせは、認証サーバにより、単一のエンティティとして各ノードへ伝送される。

6.2.6.3. ベンダ固有 AVP

次に示される ZigBee アライアンスのベンダ固有「PANA AVP」は、ネットワークセキュリティマテリアルの伝送と更新をサポートするために定義される。ベンダ固有 AVP として、この文書で定義される限り、他のドキュメントにて定義または参照されてはならない。

IANA で割り当てられる ZigBee Allianice の PEN(プライベート企業番号)は 37244 である。これらの割当ては、以下のサイトに記載されている。http://www.iana.org/assignments/enterprise-numbers

6.2.6.3.1. ネットワークキーAVP

この AVP の目的は、安全に認証サーバから各ノードにネットワーク･セキュリティ･パラメータを転送することである。

```
struct PANAAVP {
  uint16 code = 1; /* ZigBee Network Key */
  uint16 flags = 1; /* Vendor-specific */
  uint16 length = 18;
  uint16 rsvd = 0;
  uint32 vendor_id = 37244; /* ZigBee Alliance PEN */
  struct ZBNWKKEY {
    uint8 nwk_key[16]; /* NwkKey */
    uint8 nwk_key_idx; /* NwkKeyIdx */
    uint8 auth_cntr; /* AuthCntr */
  };
  struct AVPPad {
    uint8 bytes[2];
  };
};
```

6.2.6.3.2. キーリクエスト AVP

この AVP の目的は、PaC が PAA に新しいネットワークキーや現在のネットワークキーの認証カウンタの更新を伝送要求することを許可する。

```
struct PANAAVP {
   uint16 code = 2; /* ZigBee Key Request */
   uint16 flags = 1; /* Vendor-specific */
   uint16 length = 2;
   uint16 rsvd = 0;
   uint32 vendor_id = 37244; /* ZigBee Alliance PEN */
   struct ZBNWKKEYREQ {
       uint8 nwk_key_req_flags; /* request flags */
       uint8 nwk_key_idx; /* NwkKeyIdx */
   };
   struct AVPPad {
       uint8 bytes[2];
   };
};
```

6.2.6.4. タイムアウト

タイムアウト再送信タイマは、 [PANA]の第 9 章で指定されている。以下の値を使うべきである:

表6-3:PANA タイムアウト値

パラメータ	値	コメント
PCI_IRT	1 sec	初期 PCI タイムアウト
PCI_MRT	120 secs	最大 PCI タイムアウト値
PCI_MRC	5	最大 PCI 再送試行回数
PCI_MRD	0	最大 PCI 再送間隔
REQ_IRT	15 sec	初期 Request タイムアウト
REQ_MRT	30 secs	最大 Request タイムアウト値
REQ_MRC	5	最大 Request 再送試行回数
REQ_MRD	0	最大 Request 再送間隔

6.2.7. EAP

拡張認証プロトコル(EAP)は、(EAP 方式として知られる)複数の認証方式をサポートする認証フレームワークである。このセクションでは[EAP]で指定されたものに加え、制約や仕様を明確に定義する。

ZIP コーディネータは、EAP オーセンティケータとして機能しなければならず、他のすべてのノードは、EAP ピアとして機能しなければならない。

6.2.7.1. EAP Identity

EAP Request/Identity のメッセージはオプションである。しかし、EAP Response/Identity は、Request/Identity

の応答としてクライアントによってサポートされなければならない。平文で送受信される認証の初期トランザクション中にEAPクライアント／ピアに関する情報が暴露されることを防ぐために、EAP Identity(これは、EAP Request/Identityに対する応答メッセージに含められる)は"匿名"でなければならない。文字列はnull終端にしてはいけない。すなわち、9オクテット長にしなければならない。

6.2.8. EAP-TLS

EAP-TLSはEAP方式の特定のタイプを表す([EAP]参照)。このセクションでは[EAP-TLS]で指定されたものに加え、制約や仕様を明確に定義する。

6.2.8.1. マスターシークレットからのEAPキー拡張

[EAP-TLS]はキーイングとIV(Initial Vector: イニシャルベクトル) マテリアルの導出のためのキー拡張を指定する。このセクションでは、使用される暗号スイートとその出力の使用のための固有の拡張を定義する。

```
MSK = PRF(master_secret, "client EAP encryption", ClientHello.random +
ServerHello.random);
```

"クライアントEAP暗号化"の文字列はnull終端にしてはいけない。すなわち、21オクテット長にしなければならない。

MSKの長さは64オクテットであり、SHA-256からのハッシュ出力は32オクテットしかないのでPRF関数は二度繰り返されなければならない。EMSKは使用してはならない。従って生成する必要はない

MSKは[PANA]と[PANA-ENC]で定義されているように、PANA_AUTH_KEYとPANA_ENCR_KEYの生成に使用されなければならない。

6.2.8.2. EAP-TLSフラグメンテーション

[EAP-TLS]セクション2.1.5で記載されるように、EAP-TLSピアとサーバがフラグメンテーションをサポートするのは必須である。 EAPピアとサーバは、EAP-TLSのフラグメンテーションをサポートしなければならない。EAP-TLSフラグメンテーションを実施する際には、単一のEAPパケットのTLSデータの最大サイズはEAP_TLS_MTUオクテットを超えないことをZIPノードは確認しなければならない。しかしZIPノードは、ZigBee IPネットワークの外から伝送されるかもしれないEAPパケットを、MTU最大値まで受信できなければならない。

6.2.9. TLS

Transport Layer Security version 1.2(TLS)は、参加ノードと認証サーバ間の認証を提供するために、PANA、EAPおよびEAP-TLSと組み合わせて使用される。このセクションでは、[TLS]で規定されたものに加えて、制約や仕様を明確に定義する。

6.2.9.1. TLS暗号スイート

6.2.9.1.1. TLS-PSK暗号スイート

[TLS-CCM]で定義されているように、PSK暗号スイートはTLS_PSK_WITH_AES_128_CCM_8でなければならない。

6.2.9.1.1.1. PSKプリマスターシークレットからのマスターシークレットの生成

[TLS-PSK]で、プリマスタシークレットからマスタシークレットの生成が規定される。このセクションでは、使用されるPSK暗号スイートの具体的な生成を規定する。

```
master_secret = PRF(pre_master_secret, "master secret", ClientHello.random +
ServerHello.random);
```

"master secret"の文字列は NULL 終端してはいけない。すなわち、13 オクテット長でなければならない。

master_secret 長が 48 オクテットであり、SHA-256 からのハッシュ出力は 32 オクテットしかないので PRF 関数は二度繰り返されなければならない。

6.2.9.1.2. TLS-ECC 暗号スイート

ECC 暗号スイートは [TLS-ECC-CCM] で定義されているように、TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 でなければならない。

[ECDP]で定義されているように、この暗号スイートでは、secp256r1 曲線(また NIST-P256 曲線として知られている)のみを楕円曲線として使用されなければならない。

この暗号スイートで使用されるハッシュアルゴリズムは SHA-256 でなければならない。

6.2.9.2. マスターシークレットからの TLS キー拡張

[TLS]は、キーイングと IV マテリアルの生成のためのキー拡張を規定する。このセクションでは、使用する暗号スイートとその出力の使用の具体的な拡張を定義する。

```
key_block = PRF(master_secret, "key expansion", ServerHello.random +
ClientHello.random);
```

"key expansion" の文字列は NULL 終端してはならない。すなわち、13 オクテット長でなければならない。

key_block 長は 40 オクテットであり、SHA-256 からのハッシュ出力は 32 オクテットしかないので PRF 関数は二度繰り返されなければならない:

- client_write_MAC_key と server_write_MAC_key 長は、AEAD 暗号使用のため 0
- client_write_key と server_write_key 長は 16 オクテット (SecurityParameters enc_key_length for [TLS-CCM] and [TLS-ECC-CCM])
- client_write_IV and server_write_IV 長は 4 オクテット (SecurityParameters fixed_iv_length for [TLS-CCM] and [TLS-ECC-CCM])
- キーイングマテリアルのため、合計 40 オクテットは以下の通りでなければならない:

    o client_write_key は key_block[0:15]

    o server_write_key は key_block[16:31]

    o client_write_IV は key_block[32:35]

    o server_write_IV は key_block[36:39]

6.2.9.2.1. CCM 入力

TLS シーケンスの中では、唯一つの CCM で保護されたレコードが利用される。このセクションは、[AEAD] のセクション 2.1 で定義された AEAD 暗号のための入力を定義する。

6.2.9.2.1.1. CCM キー入力

TLS シーケンスで使用されるキーは、クライアントまたはサーバが暗号化するかによって client_write_key または server_write_key となる。

6.2.9.2.1.2. CCM nonce 入力

nonce は[AEAD]に規定されているように、12 オクテット長であり、以下の通りでなくてはならない:

表6-4:CCM none 入力値

フィールド	オクテット	値	コメント
IV data	0:3	-	暗号化している方のクライアント IV またはサーバ IV
Explicit nonce	4:11	{0,0,0,0,0,0,0,0}	Finished ハンドシェイクのシーケンスカウンタ

6.2.9.2.1.3. CCM ペイロード入力

ペイロードは、ヘッダを含む TLS レコードでなければならない。

6.2.9.2.1.4. CCM 関連づけられたデータ入力

関連付けられたデータ('A')は、以下の通り 13 オクテット長でなければならない:

表6-5:CCM に関連付けられたデータの入力値

フィールド	オクテット	値	コメント
Explicit nonce	0:7	{0,0,0,0,0,0,0,0}	Finished ハンドシェイクのシーケンスカウンタ
TLS record type	8	22	TLS ハンドシェイク識別子
TLS Protocol Major	9	3	TLS 1.2
TLS Protocol Minor	10	3	TLS 1.2
TLS length MSB	11	-	TLS record MSB の長さ
TLS length LSB	12	-	TLS record LSB の長さ

6.2.9.2.1.5. データリンク層セキュリティ

データリンク層セキュリティマテリアルは、以下のように PANA 認証 または、PANA キーの更新プロセスを介して受信したネットワークセキュリティマテリアルから、各ノード(セクション6.2.6.2を参照)によって導出される。

データリンク層の MAC キーは、下式の計算結果の上位 16 オクテットに設定される。

```
HMAC-SHA256(Network Key,"ZigBeeIP")
```

Key Index は、キーシーケンス番号に設定される

Outgoing frame counter の初期値は、以下の結果に設定される。

```
Node Auth counter || 00 00 00
```

ここで | | は連結演算子であり、ノード認証カウンタは最上位オクテットの位置になっている。このフィールドの値は、紐付いたキーが、メッセージ保護に使用されるたびに、1 ずつインクリメントされなければならない。

データリンク層セキュリティマテリアルは、後述の MAC キーテーブルの KeyDescriptor エントリを作成するために使用される。MAC キーのテーブルが一杯になった場合、アクティブでない既存のエントリを削除し、新しい KeyDesciptor エントリを格納しなければならない。

各 ZIP ノードは、現在アクティブな MAC キーのキーインデックスを含む属性を維持しなければならない。

最初の MAC KeyDescriptor エントリが作成されたときに、アクティブなキーインデックスはそのキーインデックスの値に設定される。アクティブなキーインデックスは、その後にネットワークキーの更新メカニズム(セクション0を参照)を通して更新される。

生成者の IEEE アドレスベースの EUI-64 MAC アドレス、アクティブな MAC キー、アクティブな MAC キーインデックスは、送信するデータリンク層のデータパケットを保護するために使用されなければならない。

[802.15.4]のデータリンク層セキュリティに関するセクション(IEEE802.15.4-2006 の 7.5.8　Frame Security)で規定された手順は、データリンク層セキュリティを適用するために従わなければならない。次のセクションでは、データリンク層のセキュリティに適用される動作モードを示す。

後述のセクションで説明されているデータリンク層のセキュリティ属性のデータは、[802.15.4]の機能仕様を反映していることに注意すること。データの編成は、ストレージ· スペースを考えて最適化されている分けではなく、特定の実装方法を意味するものではない。

6.2.9.2.2.　デフォルトキーソース

参加ノード(join し、認証され、許可されたもの)は以下の設定をもたなければならない。

表6-6:参加ノードの設定

PIB 属性	値	コメント
macDefaultKeySource	0xff00000000000000	MAC キーを表す任意の値。知られてないかもしれないが、ネットワークキー生成者の実際の IEEE アドレスを格納する必要はない

6.2.9.2.2.1.　キー識別モード 1 の使用

キー識別モード 1 は MAC キーと組み合わせて使用しなければならない。これは *macDefaultKeySource* の利用を意味する。MAC キーインデックスと組み合わせて使用されるグローバル MAC キーに対して、ネットワークキーインデックスと組み合わせて *macDefaultKeySource* の値を格納する必要がないため、(MAC キーを特定するために格納する必要がある参照データを MAC キーインデックスのみに減らせることを意味する。このメカニズムは[802.15.4]において、キーID モードの数を制限するために便利なので使用されている。

6.2.9.3.　MAC キーテーブル

[802.15.4]はキーのストレージをデバイス記述子のストレージと切り離し、関連するデバイス記述子をポイントするためにキーストレージのハンドルを利用することに注意すること。

参加ノードは次の設定を持つべきである。一つのアクティブな MAC キーと(MAX_NWK_KEYS - 1)個のバックアップの MAC キーが存在する。

表6-7:参加ノードのキーテーブル

PIB 属性	値	コメント
macKeyTable	KeyDescriptor entries	アクティブな MAC キーに対して一つのエントリ、バックアップの MAC キーに対して追加のエントリ
macKeyTableEntries	MAC_MAX_NWK_KEYS	アクティブな MAC キーに対して一つのエントリ、バックアップの MAC キーに対して追加のエントリ

ZIP ノードは、各 MAC キーに対し、次の KeyDescriptor エントリのセットを持つべきである。:

表6-8：Key 記述子

KeyDescriptor 属性	値	コメント
KeyIdLookupList	One KeyIdLookupList entry	MAC Key のエントリ
KeyIdLookupListEntries	1	MAC Key の 1 つのエントリ
KeyDeviceList	KeyDeviceList entries	MAC デバイステーブルのエントリ
KeyDeviceListEntries	(variable)	MAC デバイステーブルのエントリ数
KeyUsageList	KeyUsageList entries	MAC データフレームの 1 つのキーを使用
KeyUsageListEntries	1	MAC データフレームの 1 つのキーを使用
Key	(variable)	MAC Key の値

KeyIdLookupList エントリは、次のセットを持つべきである。:

表6-9：KeyID lookup 記述子

KeyIdLookupDescriptor 属性	値	コメント
LookupData	*macDefaultKeySource* || KeyIndex	KeyID のみを格納する必要がある。KeyIndex は、この MAC キーに関連付けられた MAC キーインデックス。
LookupDataSize	0x01	サイズ: 9 octets

KeyDeviceList エントリは、デバイス記述子をポイントする。各 KeyDeviceList エントリは、次のセットを持つべきである。

表6-10：KeyDeviceList エントリ

KeyDeviceDescriptor 属性	値	コメント
DeviceDescriptorHandle	Implementation-specific	適切なデバイスディスクリプタのポインタ
UniqueDevice	0	キーはノードごとにユニークではないから
Blacklisted	Boolean	初期値は、FALSE に設定する

ZIP ノードは、MAC キーがデータリンク層のデータフレームに対して使用されるのが有効であると示す 1 つの KeyUsageList エントリを持つべきである。静的なポリシーとなるため、このデータは暗に示すことができ、このデータを保持するためのストレージは必要ない。データリンク層のデータフレームに対するエントリには、次のセットを持たなければならない:

表6-11：MAC データフレームの KeyUsageList エントリ

KeyUsageDescriptor 属性	値	コメント
FrameType	0x02	データリンク層のデータフレーム

6.2.9.4. MAC デバイステーブル

ZIP ノードには、次のセットを持つべきである。

このノードと通信する近隣ノードごとに 1 つの DeviceDescriptor のエントリがある。

ZIP ルータは MAC デバイステーブルに少なくとも MAC_MIN_DEV_TBL 個のエントリを保持する容量を持つべきである。

表6-12:MAC デバイステーブルのエントリ

PIB 属性	値	コメント
macDeviceTable	DeviceDescriptor entries	通信中の各近隣ノードに対して一つのエントリ
macDeviceTableEntries	(variable)	通信中の各近隣ノードに対して一つ

各近隣ノードの DeviceDescriptor エントリには、次の情報が含まれている。

表6-13：参加ノードの device descriptor エントリ

DeviceDescriptor 属性	値	コメント
PANId	2 bytes	近隣ノードの PAN ID。近隣ノードがこのノードと同じ PAN ID を持つため、このデータを含む事が可能であり、保管は不要であることに注意。
ShortAddress	2 bytes	近隣ノードに割り当てられたショートアドレス。

ExtAddress	8 bytes	近隣ノードの IEEE アドレス。
FrameCounter	4 bytes	近隣ノードから最近受信した MAC フレームの受信フレームカウンタ。
Exempt	FALSE	データリンク層でセキュリティポリシ無しの場合に無関係とする除外フラグ。したがって、このデータは含む事が可能であり、保管不要である。

[802.15.4]において、各々KeyDecriptors は個別の KeyDeviceList(DeviceDescriptors のリスト)を持つことが許可されることに注意。それは近接ノードが個々の鍵を使用することが適切であることを示す。ZIP ノードは、MAC デバイステーブルの全エントリから成る、KeyDescriptors の各 KeyDeviceList と同じ内容の DeviceDescriptor リストを維持しなければならない。これは、各キーがその近隣ノードのどれに使用されても有効であることを意味する。

6.2.9.5. セキュリティレベルテーブル

データリンク層でのセキュリティ･ポリシーはない。エンフォースメント・ポイントは、セクション6.3.9.4 の仕様に基づいてポリシングを実行する。したがって、すべての ZIP ノードには、次のセットを持っていなければならない。

表6-14：セキュリティレベルテーブル

PIB 属性	**値**	**コメント**
macSecurityLevelTable	Empty	データリンク層でのセキュリティ･ポリシーはない。
macSecurityLevelTableEntries	0	データリンク層でのセキュリティ･ポリシーはない。

6.2.9.6. 補助セキュリティヘッダフォーマット

MAC frame 補助セキュリティヘッダ([802.15.4] IEEE802.15.4-2006 の Section 7.6.2 参照)は、MAC frame が保護されているときに、セキュリティのための追加データを提供するために使用される。

6.2.9.6.1. Security Control field

Security Control field は以下の値を持たなければならない。

表6-15：Security control field

フィールド	**値**	**コメント**
Security Level	0x05	ENC-MIC-32 が ZigBee IP リンク層セキュリティのデフォルト値
Key Identifier Mode	0x01	鍵は､セキュリティ補助ヘッダの Key Identifier field の 1 オクテット Key Index サブフィールドと macDefaultKeySource で決定される

6.2.9.6.2. Frame Counter field

Frame Counter field は macFrameCounter PIB 属性の値を前提としなければならない。

6.2.9.6.3. Key Identifier field

Key Identifier はアクティブな MAC Key と関連付けられた MAC Key Index でなければならない。

6.2.10. MLE

mesh link establishment protocol [MLE]は UDP プロトコルを使って、メッシュネットワーク内のノードに対し、ノードと近隣ノードとの接続プロパティを交換するメカニズムを提供する。さらに、それは ZigBee ネットワーク内のすべてのノードへのリンクの構成情報を伝播するために使用される。

すべての ZigBee IP ノードは MLE プロトコルを実装しなければならない。

6.2.10.1. MLE リンク設定

すべての ZIP ノードは MLE コンフィギュレーションメッセージの送信と受信をサポートする必要がある。これらは、リンクリクエスト、リンクアクセプト、リンクアクセプト・リクエスト、リンク・リジェクト の各メッセージを含む。これらのメッセージは、IEEE802.15.4 インタフェースプロパティを交換するため、また、近接ノードが使用するフレームカウンタの値を認証するために使用される。これらのメッセージは、ペイロードに、次の TLV オプションを含めることが出来る。

- 送信元アドレス(TLV タイプ= 0)TLV は、16 ビットのショートアドレスと IEEE802.15.4 の 64 ビットの EUI-64 アドレスで通信するためにノードによって使用される。

- モード(TLV タイプ= 1)TLV は、ノードの機能情報を通信するためにノードによって使用される。Value フィールドの長さは 1 オクテットであり、以下のようにフォーマットされなければならない。

表6-16:MLE リンク設定におけるフォーマット

bits: 0	1	2	3	4 - 7
Reserved	FFD	Reserved	RxOnIdle	Reserved

FFD ビットは、ZIP ホストでないすべてのノードは「1」に設定しなければならない。常時無線が ON のノード(すなわち、スリープしないノード)は、RxOnIdle ビットを「1」に設定しなければならない。Reserved ビットは送信時に「0」設定され、受信側では無視されなければならない。

- タイムアウト(TLV タイプ=2)TLV は、ホストがその親ノードと通信できないと判断することができる不活性の期間を通信するために、スリープホストノードによって使用される。スリープホストノードは、この値よりも小さい周期で MAC poll を実行すべきである。

- チャレンジ(TLV タイプ= 3)と応答(タイプ= 4)TLV が互いの MAC フレームカウンタの値を認証するため、ペアとなるノードで使用される。チャレンジ TLV の Value フィールドは 8 オクテットの長さのランダムな値に設定されなければならない。

- リプレイ･カウンタ(TLV タイプ= 5)TLV は、MAC 送信フレームカウンタの値を通信するために使用される。

6.2.10.2. MLE Advertisement

すべての ZIP ルータは MLE Advertisement メッセージの送信と受信をサポートしなければならない。このメッセージは、近隣ルータとの双方向のリンク品質値を交換するために使用される。双方向のリンク品質値は、RPL の親の選択の質を向上させるために使用される。また、このメッセージは、近隣ルータのセットの変更を検出するために使用される。

ネットワークに参加している ZIP ルータは、定期的に MLE Advertisement メッセージを MLE_ADV_INTERVAL 間隔で送信しなければならない。

MLE Advertisement メッセージは､ペイロードにリンク品質(TLV タイプ=6)TLV を含めなければならない。この TLV の近隣レコードには､発信元ノードの MAC デバイステーブル内のノードに関する情報が入っていなければならない。各近接レコードの近接アドレスフィールドには特定の近接ノードの 16bit short アドレスが入っていなければならない。P(priority)フラグは、RPL 親ノードセットに含まれる隣接ノードのために設定されるべきである。これは、それらの隣接ノードがこのノードとのリンクのメンテナンスを優先的に実施するべきであることを指示するためである。

近接レコードを含む近隣ルータからの MLE Advertisement メッセージを MLE_ADV_TIMEOUT 時間、受信しなかった場合、ZIP ルータは近隣ルータの MAC デバイステーブルエントリを削除しなければならない。

6.2.10.3. MLE 更新

ZIP コーディネータは、MLE 更新メッセージの発信をサポートしなければならない。すべての ZIP ノードは、MLE 更新メッセージの受信をサポートしなければならない。

MLE 更新メッセージは、ネットワーク内の様々なリンク層のパラメータの値を設定するために ZIP コーディネータによって使用される。MLE 更新メッセージは、ネットワークパラメータ TLV のインスタンスを一つだけ含まなければならない。この TLV は、次のいずれかのパラメータを含まなければならない。

- ネットワークパラメータのチャネルは、ノードによって使用されなければならないチャネルを設定するために使用される。それは長さ 2 オクテットの Value フィールドでなければならない。Value フィールドの上位オクテットにチャネルページ番号が含まれており、下位オクテットはチャンネル番号が含まれている。各物理層のチャンネルページとチャンネル番号の定義は[802.15.4]にある。

- ネットワークパラメータの PAN ID は、ネットワーク内のノードによって使用される 802.15.4 PANID 値を設定するために使用される。それは新しい PANID を含んだ長さ 2 オクテットの Value フィールドでなければならない。受信ノードはデータリンク層の対応する属性を更新するためにこの値を使用しなければならない。加えて、MAC device descriptor エントリのそれぞれの対応するフィールドを更新しなければならない。(**表 6-13**参照)

- ネットワークパラメータのパーミットジョインは、ノードによって使用されるべき Allow Join フィールドを設定するために使用される。 (セクション6.3.3.1を参照)。それは長さ 1 オクテットの Value フィールドでなくてはならない。ZIP ルータは、ビーコン・ペイロードの Allow Join パラメータをセットする為に、このオクテットの最下位ビットの値を使用しなければならない。Value フィールドの他のビットは送信ではゼロがセットされ、受信では無視されなければならない。

- ネットワークパラメータのビーコン・ペイロードは、ビーコン・ペイロード内の Optional フィールドを設定するために使用される (セクション6.3.3.1を参照)。受信ノードは、現在のビーコン・ペイロード(**表 6-18**参照)の中の全ての Optional フィールドを、このメッセージの Value フィールドの中味により、置き換える。MLE 更新メッセージには、単一パラメータ TLV を含めることができるので、ZIP コーディネータは単一の TLV の全 Optional フィールドの完全な連結されたセットが含まれることを保証しなければならない。ビーコン・ペイロードに Optional フィールドが含まれていない場合、長さがゼロも可能であることに注意。

ネットワークパラメータ TLV フォーマットは Delay フィールドを含んでおり、それは受信ノードが適当なパラメータを設定することを実施するまでの遅延値を規定するために使用される。パラメータがチャネルか PanID のとき、Delay フィールドはネットワーク内にマルチキャストパケットが伝播する時間よりも長くするべきである。それは、それらのパラメータが変更される前に全ノードが MLE 更新パケットを受信することを保証するためである。推奨値は 5 秒である。

ZIP ノードは、前のメッセージをまだ反映していない場合、同じネットワークパラメータ TLV を含む新しい MLE 更新メッセージを無視してもよい。ZIP コーディネータは、ネットワークパラメータを含む連続した MLE 更新メッセージに、このようなシナリオを回避するために十分な遅延を持たせることを保証するべきである。

稀な状態において、チャネルや PanID を含む MLE 更新メッセージが全てのノードに正しく受信されない場合には、ZIP ノードは、取り残された状態になり得る。各ノードにおいて、この状態を検知することは、この仕様書の範囲外である。回復手順は、全てのチャネルでネットワーク探索を行い、ネットワーク再加入を試みる事である。

MLE の更新メッセージはサブネットローカルの全ルータのマルチキャストアドレスに送信されなければならない。

6.2.10.4. MLE メッセージセキュリティ

MLE メッセージは、場合により、ノードがネットワークに参加し近隣ノードとセキュアなリンクを確立する前に、送受信される。したがって、MLE のメッセージはデータリンク層セキュリティに依存することが出来ない場合があり、MLE のプロトコルはそのペイロードを保護するために独自のメカニズムを定義する。

MLE コンフィギュレーションメッセージは MLE 層で保護され、データリンク層で保護されるべきではない。セキュリティを伴わない MLE コンフィギュレーションメッセージは、新しいノードがまだセキュリティマテリアルを取得していない間の、ノードブートストラッププロセスの初期段階の間だけ送受信可能である。それ以降では、ノードはいつも MLE コンフィギュレーションメッセージにセキュリティを適用しなければならない。ZIP ノードは、受信した MLE コンフィギュレーションメッセージが、ＭＬＥセキュリティを伴わない場合には、既存のノードエントリの状態情報を変化させないことを保証しなければならない。送信側はこれらのパケットの送信元アドレスに LL64 IP アドレスを使用しなければならない。

MLE Advertisement メッセージは MLE 層で保護されなくてはならず、データリンク層では保護されずに送信されるべきである。送信側はそれらのパケットの送信元アドレスに LL64 IP アドレスを使用する必要がある。MLE セキュリティがない MLE Advertisement パケットを受信した場合、破棄する必要がある。ノードは、セキュアリンクが確立されたノードからの MLE Advertisement メッセージのフレッシュネスを検証するべきである。

MLE 更新メッセージは、MLE 層で保護されるべきではなく、データリンク層で保護されなければならない。これらのメッセージはすでにネットワークに参加したノードにだけ送信する。つまり、データリンク層

のセキュリティが適応できる。加えて、MLE 更新メッセージはサイトローカルマルチキャストアドレスで送信されるため、MAC セキュリティが使用されなければならないか、もしくはそのパケットは他の ZIP ノードによって転送されない(Section6.3.9.4参照)また、送信ノードと受信ノードのそれぞれが直接無線通信できない距離にいる場合には、両ノード間に保護されたリンク設定を持たないことがあり、それらのパケットのために MLE セキュリティを使用することは不可能である。

6.2.10.5. MLE セキュリティマテリアル

MLE のパケットを保護するために使用される MLE セキュリティマテリアルには、次のパラメータが含まれている。

表6-17:MLE security material

Parameter	Size	Comment
MLE Key	16 octets	MLE キー
Key Index	1 octet	このキーに関連付けられているキーインデックス
Outgoing frame counter	4 octets	このキーを使用して送信する MLE メッセージを保護するために使用されるフレームカウンタの値

MLE セキュリティマテリアルは、以下に記述される PANA 認証 または、PANA キー更新プロセスを介して受信するネットワークセキュリティマテリアル(セクション 7.3.2 を参照)から、各ノードによって導出される。

MLE キーは、HMAC-SHA256(ネットワークキー"ZigBeeIP")の結果の下位 16 オクテットに設定される。

キーインデックスは、ネットワークキーシーケンス番号に設定される。

送信フレームカウンタの初期値は次のように設定される。

ノ ード認証カウンタ ||00 00 00

ここで、| |は連結演算子であり、ノード認証カウンタが最上位オクテットの位置になる。ノード認証カウンタの値は、メッセージを保護するために関連付けられた鍵が使用されるごとに 1 ずつインクリメントされなければならない。

ZIP ノードは、認証サーバで生成された最新 2 つのネットワークセキュリティマテリアルにより算出された MLE セキュリティマテリアルを格納しなければならない。これらは、アクティブ用と代替用の MLE セキュリティマテリアルとして指定される。

認証サーバから新しいネットワークセキュリティマテリアルを受信した場合、それはアクティブな場所が空の場合、そこに格納しなければならない。それ以外(セキュリティマテリアルを既に持っている場合)の場合は、代替用の場所に格納しなければならない。

送信する MLE パケットのセキュリティには、アクティブな MLE セキュリティマテリアルを適用しなければならない。受信 MLE パケットのセキュリティには、受信メッセージの MLE 補助セキュリティヘッダに含まれているインデックスと一致する MLE セキュリティマテリアルを適用しなければならない。

MLE メッセージ補助ヘッダ内のセキュリティ制御フィールドは、データリンク層のセキュリティに使用される値と同じ値を使用しなければならない。セキュリティレベルは 5(4 オクテットの MAC アドレスを持つ CCM 暗号化)でなければならない。また、鍵の識別子モードは、1 でなければならない。CCM ノンスで使用されるアドレスは、ノードの 64 ビット MAC アドレスでなければならない。フレームカウンタは MLE 送信フレームカウンタでなければならない。

6.3. 機能記述

6.3.1. 概要

ZigBee IP ネットワークは、一つの ZIP コーディネータと複数の ZIP ルータと ZIP ホストを含むノードから構成される。これらのノードは、IEEE802.15.4 の観点から、一つの PAN を形成し、 IPv6 の観点から、ノードは共通のプレフィックスを持つ一つのマルチリンクサブネットを形成する。

ZIP コーディネータが IEEE802.15.4 PAN コーディネータとして動作を開始し、その IEEE802.15.4 のインタフェースを IPv6 ルータとして構成したとき、ZigBee IP ネットワークは形成される。

ネットワークが形成されると、他のノードは、その能力に応じて、ZIP ルータ または ZIP ホストのいずれかとしてネットワークに参加することができる。

新しいノードは、ネットワーク探索、ネットワーク参加許可、ネットワーク認証の 3 ステップのプロセスでネットワークに参加できる。詳細は、後述のセクション(6.3.3, 6.3.4, 6.3.5, 6.3.6)で説明する。一度ノードがネットワークに参加し、もしそのノードが ZIP ルータなら、他のノードが自分を介して参加することを許可してもよい。これは ZIP コーディネータの無線到達範囲を超えた無線メッシュネットワークの形成を可能にする。

ZigBee IP ネットワークの一部であるノードは他の暗号キーを導出するためのユニークなネットワークキーを共有する。これによって、リンク層ですべてのパケットを保護する。ノードは初期参加プロセスの間にこのキーを獲得する。これは時間経過と共にアップデートされてもよい。

6.3.2. ネットワーク構成

6.3.2.1. データリンク層のコンフィグレーション

新しい IEEE802.15.4 PAN ネットワークを形成するために管理設定されるノードは次の複数のステップを実施する。

- ノードは MAC energy detect scan を前もって設定されたチャンネルに行い、設定された閾値以下のエネルギーレベルのチャンネルを識別する。スキャンチャンネルリストは管理者によって構成される。
- ノードは前ステップで選択されたチャンネルで、標準ビーコンリクエストを使った MAC active scan を行う。
- ノードは存在する複数の IEEE802.15.4 ネットワークから最小番号のチャンネルを選択する
- ノードは、前ステップで発見された他のネットワークと衝突しない PANID を選び 16 ビットのショートアドレスをランダムに生成する
- ノードは選択されたチャンネルと PANID で IEEE802.15.4 PAN を開始する

6.3.2.2. IP コンフィグレーション

新しい PAN を開始する時に ZIP コーディネータはグローバルユニークか ULA[RFC 4193]の 64-bit IPv6 global prefix(複数可)を用いて 6LoWPAN を構成する為の準備をしなければならない。この prefix は管理者が設定しているかもしれないし、DHCPv6 prefix delegation や他の手段を介して上流ネットワークから取得しているかもしれないが、それは本標準の範囲外である。

6LoWPAN IPv6 prefix が設定された後、ZIP コーディネータは 6LoWPAN prefix とノードの 16 ビット MAC ショートアドレスから作成されたインタフェース識別子から構成される IPv6 address を用いて IEEE802.15.4 インタフェースを構成する。

ZIP コーディネータは IEEE802.15.4 以外のインタフェースを持つかもしれないが、これらのインタフェースの初期化は、本仕様の対象外である。

IPv6 構成が完了すると、ZIP コーディネータは[6LPND]に従って Neighbor Discovery (ND) protocol exchange に参加する。ZIP コーディネータはデフォルトコンテキストを 6LoWPAN 全体で使用するために割り当てられた /64 prefix として構成する。ZIP コーディネータは、デフォルトコンテキストを含む MIN_6LP_CID_COUNT の最大値までのコンテキスト識別子を維持することができる。[6LPND]で定義されたように、ZIP コーディネータはマルチホップのプレフィックスとコンテキストディストリビューションを使用している。

ZIP コーディネータは、新しい RPL インスタンスを開始し、セクション 5.5.4.2.3 から動作パラメータを持つ DODAG を形成する。追加のノードがネットワークに参加するとき、ZIP コーディネータは、[RPL]に従って RPL プロトコル交換を始める。

ZIP コーディネータは PANA の認証サービスを初期化する。ネットワーク･セキュリティ･マテリアル(セクション6.2.6.2を参照)は、ランダム 128 ビットのネットワークキーとキーシーケンス番号 1 から生成される。データリンク層と MLE の層は、ネットワークセキュリティマテリアルから導出したキーマテリアルの使用を開始する。さらに認証サーバは、ZigBee ベンダ固有のネットワークキーAVP(セクション6.2.6.3を参照)を介して広がるネットワークセキュリティマテリアルを設定している。

6.3.3. Network discovery

ネットワークディスカバリの手順は、電波到達範囲内の他の IEEE802.15.4 のネットワークを見つけるのに使われる。それぞれのネットワークにおいて、この過程の中でネットワーク ID とそれに関わるいくつかの情報が見つかる。

MAC ビーコン機能を使い ZigBee IP ノードがネットワークディスカバリを実行する。

全ての ZigBee IP ノードは MAC ビーコン要求コマンドパケットを送信することができなければならない。ZIP コーディネータと 全ての ZigBee IP ルータはビーコン要求コマンドを処理し、応答としてビーコンパケットを送信することができなければならない。

一般的なネットワーク探索を実行するには、ZigBee IP ノードはビーコンリクエストパケットを送信し、すべてのレスポンスを収集する。これは通常、ノードが新しいネットワークを開始することに使われ、その為に存在する PANID とローカルで使用されているチャンネルを識別することが出来る。

ネットワーク探索プロセスはノードが電波レンジ内のルータノードを発見することも許可している。これらのルータの一つがネットワークに参加する為の"親"ルータとして選択される。

6.3.3.1. ビーコン・ペイロード

MAC ビーコンパケットコマンドは、ビーコンリクエストパケットの応答として送信される。ビーコンパケットはアプリケーションが設定可能なネットワークに関する情報を伝える為に使用されるペイロードフィールドを含んでいる。ZigBee IP ルータは以下のようにビーコン･ペイロードを設定しなければならない。

表6-18：Beacon payload format

Octets: 0	1	2 － 17	18 － variable
ZigBee protocol 識別子	Control field	ZIP NetworkID	Optional fields

- octet Protocol ID － これは 0x02 に設定しなければならない、そして ZigBee IP ネットワークに使用される、そして無線到達範囲内に存在する他の IEEE802.15.4 ベースのネットワークから区

別するのに役立つ。

- octet Control field – これは参加デバイスの情報を伝達するのに使用される、その為に適切なネットワークと参加の為の親ルータを選択できる.それは以下のようにフォーマットされている複数のサブフィールドが含まれている。

表6-19:Beacon payload control field format

Bits: 0	1	2	3 – 7
Allow join	Router capacity	Host capacity	Reserved

  - このネットワークが現在新しいノードがネットワークに参加することを許可しているならば、Allow Join ビットが新しく参加するノードにヒントを与える。このネットワークが現在新しい機器の参加を許可しているならば、それは 1 にセットされる。このフィールドの値は上の階層のプロトコル(Section6.2.10.3参照)を利用としたネットワークを介して伝達され、ZIP コーディネータ上でノード管理アプリケーションによって設定される。ZIP ルータは最初にネットワークに参加した後、それが、その親ルータによって使われた同じ値に対してこのフィールドの値を設定する。次にこのフィールドの値は ZIP コーディネータから受けた新着 MLE Update メッセージに基づいて設定される。MLE Update メッセージのロスを防ぐには、このフィールドが MLE_MAX_ALLOW_JOIN_TIME よりも長い時間に設定されていたならば、ZIP ルータは自動的にこのフィールドを 0 に設定しなければならない。

  - Router capacity と Host capacity のビットは、ビーコンパケットの送信元がネットワークを介して参加するために、新しいホスト または、ルータがノードを受入る能力を持っているかどうかを示すために使用される。これらのビットの値は、そのリソースが利用できるか(例えば、近隣キャッシュと MAC デバイス･テーブルのスペースの余裕によって異なる)に応じて各ノードの管理エンティティによって設定される。

  - 予約ビットは、送信するときにゼロに設定されて、受信時に無視されなければならない。

- NetworkID – この 16-octet field は ASCII 文字として解釈され、特定のネットワークをユーザに識別させる為に使われる。このフィールドの値は、ZIP コーディネータに設定管理されている。他の ZIP ルータは、ネットワークを介して親ルータのビーコン・ペイロードからこのフィールドの値を受信する。
- 可変長のオプションフィールドは、タイプ - 長さ - 値形式を使用して、ビーコンのペイロードに含まれても良い。以下に示すように各オプションフィールドがフォーマットされている。

表6-20：Beacon payload optional field format

Octets: 1		2 - Length
Bits: 0 - 3	4 - 7	
Length	Type	Value

- The Type subfield の長さは 4 ビットであり、フィールドのタイプを識別する。次の値が定義される。

表6-21：Beacon payload optional field types

Type	Description
0	4 オクテット値。ネットワークに参加する特定ノードを操縦するノード識別子として使われる。例えば、デバイス証明書のハッシュの一部を使用することが出来る。
1 - 15	Reserved

- The Length subfield の長さは 4 ビットであり、オクテット単位での値のサブフィールドの長さを識別する。

- Value subfield は、フィールドの値が入る。

ノードは、サポートされていない任意のオプションのフィールドを無視しなければならないが、その他の処理は続行される。

6.3.4. ネットワーク選定

ディスカバリー手順によって、無線到達範囲内の複数の ZigBee IP ネットワークを発見することができる。ノードが参加しなければならないネットワークの選択は、アプリケーション固有の手段で行われる。ZigBeeIP の仕様では、ノードが参加しなければならない正しいネットワークにジョインするために使用できるさまざまなツールを提供している。このセクションでは、それらのツールについて以下に記載している。

- "Allow Join" flag indication - このフラグは、すべての ZigBeeIP ルータのビーコン・ペイロードに存在している。参加しようとしているノードは、適切なネットワークを選択するため、すべての隣接 ZigBeeIP のルータのフラグを調べることができる。ネットワーク内のルータは通常ゼロにこのフラグを設定する。新しいノードが(アプリケーションの特定の手段によって決定される)ネットワークに参加することが期待されている場合、このフラグは特定期間 true(1)に設定される。ZIP コーディネータは、ネットワーク内のすべてのルータにフィールドで使用する値を転送する責務がある。

  ただし、このパラメータは、参加予定ノードへのヒントだけであることに注意すること。ZIP ルータの動作は、このフィールドの値に基づいて変更されない。特に、もし ZIP ルータがゼロに設定されたフラグを持っていても、新しいノードがこのノードを介して参加できるようにし続けなければならない。ZIP コーディネータだけは、接続を拒絶してもよい。

- “User selection” - 参加ノードがビーコンスキャンを実行し、その無線の範囲内にすべての ZigBee IP ネットワークを発見する。その後、ネットワークに関する情報を表示し、ユーザが参加するべきネットワークを選択することができるようになる。

- “Preconfigured information” - 参加しようとしているノードは、参加しなければならない特定のネットワークに関する情報によって構成を更新することができる。この情報は、例えば 、ビーコン・ペイロード内の"NetworkID"フィールドなどであるかもしれない。

- ““Device identifier” - 参加しようとしているノードの識別子は、ビーコン・ペイロードに含まれている。参加しようとしているノードの素性を ZIP コーディネータが知っている場合、、ビーコン・ペイロードに識別子を含めることによって、このメソッドを使用してネットワーク内の全ルータにこの情報を伝播することができる。

これは完全なリストではなく、アプリケーションは参加するネットワークを選択するための他の手段を実装することができることに注意すること。これらのメカニズムは、ただ参加しようとしているノードに対してネットワーク選択の "ヒント"を提供することを意図していることに注意するべきである。また、ネットワークを選択し、参加した後にノードはそれが正しいネットワークに参加していることを検証するために、アプリケーションレベルの登録メカニズムを使用することを期待している。ノードがアプリケーションの検証に失敗した場合、管理エンティティは、そのネットワークをブラックリストに載せ、ネットワークセレクションと参加プロセスを繰り返すべきである。

6.3.5. ノード参加

ネットワークディスカバリとセレクションをした後、参加しようとしているノードは、ネットワークへのアクセスを得るためにブートストラップ手順を実行する。一般的な参加シーケンスは、以下の図に示し、次の各項で詳しく説明する。

6.3.5.1. ホストのブートストラップ

ZigBeeIP の Host ノードのブートストラップシーケンスを、以下で説明する。

1. ノードはネットワークディスカバリを実行する。選択手順は、前述のようにする。参加するには適切なネットワークを選択する。

2. 選択したネットワークに属している ZIP ルータを親として選択する。これは通常、ホストを受け入れる能力(available host capacity)(ビーコン・ペイロードのホスト受容能力のサブフィールドを 1 に設定されている)を持ちかつ ビーコン受信したルータの中で最高の LQI(link quality indicator)を持つルータである。

3. ノードは選択したターゲットネットワークの PANID に自分の IEEE802.15.4 MAC PAN 識別子(PAN-ID)を設定する。

4. ノードは、LL64 アドレス形式を使用して、その IEEE802.15.4 インタフェースに IPv6 リンクローカルアドレスを構成する。

5. .ノードがスリープホストである場合、それはスリープデバイスであることと、レイヤ 2 パケット伝送のための MAC ポーリング機能を使用することを親ルータに通知するために MLE のプロトコル交換を使用しなければならない。この情報は、MLE リンク要求パケットのモード TLV オプションに含まれている。

親ルータは、ノードの EUI-64 アドレスの MAC ポーリングを設定する。親ルータはスリープノードを受入る事ができない場合、リンク要求を拒否しなければならない。その場合 参加しようとしているノードは、別の親ルータを選択し、ステップ 2 のプロセスから続けるべきである。

ノードがスリープホストである場合、それがユニークなショートアドレスを構成し、MLE プロトコルを使用して、親ルータにそれを登録するまで、EUI-64 アドレスを使用して MAC ポーリングを実行する必要がある。(このシーケンスのステップ 11 を参照)。

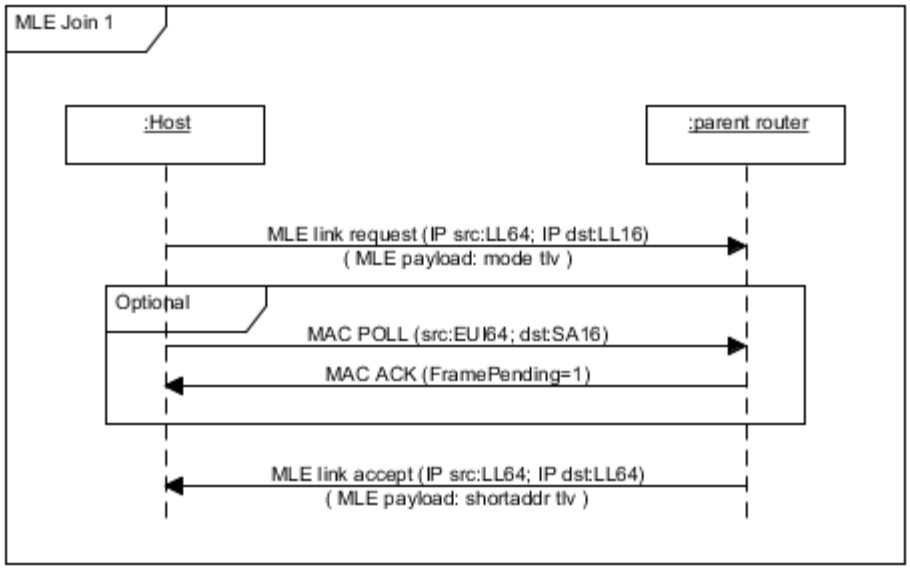


図6-2：Join sequence - MLE 1

6. ノードは、PANA のプロトコルを使用してネットワーク認証を実行する。この手順が正常に完了すると、ノードがネットワークに認証され、ネットワークセキュリティマテリアルを取得する。メッセージシーケンスの例としては、セクション6.5.3.4を参照のこと。

7. ノードは、親ルータとフレームカウンタを同期させる 3 ウェイセキュリティで保護された MLE のハンドシェイクを実行する。この手順の最後にノードは、親ルータのフレームカウンタを知る事ができる。また、親ルータはノードのフレームカウンタを知る事ができる。

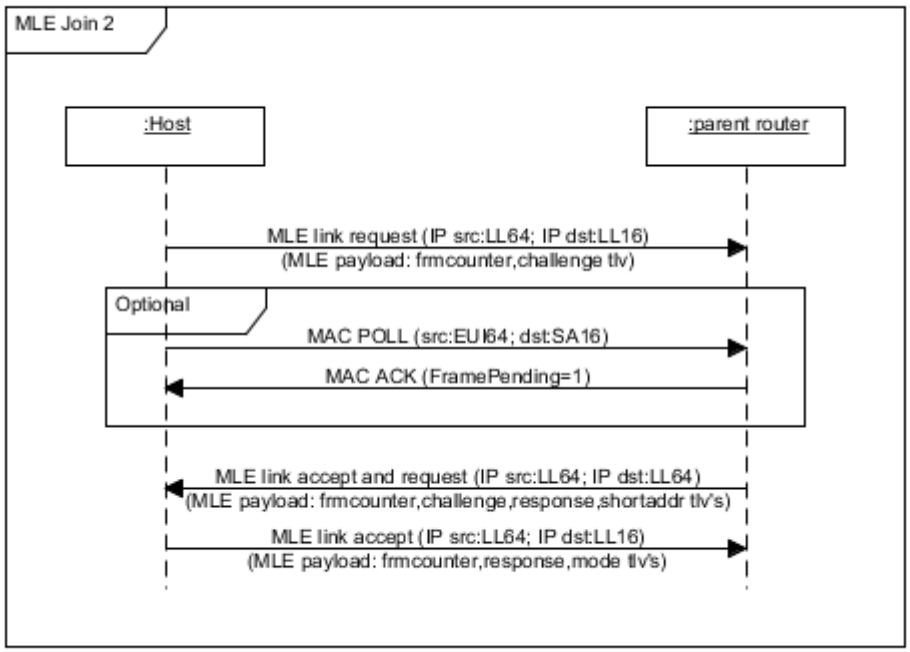


図6-3：Join sequence - MLE 2

8. ノードが、ルータ要請(Router Solicitation)パケットを送信し、応答の Router Advertisement を待つことで、[6LPND]で記述される IPv6 ルーターディスカバリーを実行する。ZigBee の IP ネットワークで使用されている IPv6 プレフィックスは、受信した Router Advertisement パケットの PIO オプションから取り出される。

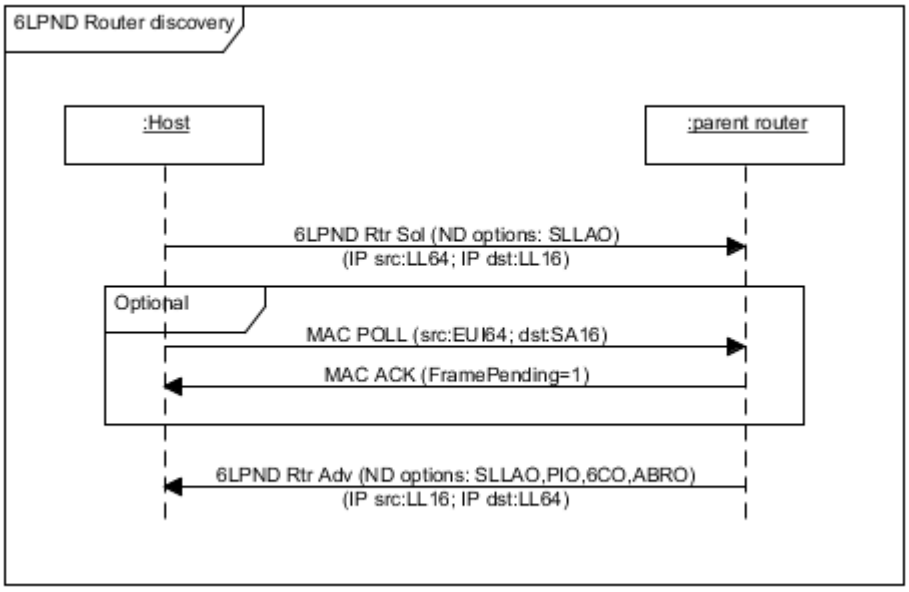


図6-4：Join sequence - Router discovery

9. ノードは、MAC ショートアドレスとしてランダムに生成された 16 ビットのアドレスを設定する。このアドレスは、[802.15.4]仕様に従って、値を 0xFFFE.または、0xFFFF を取ることはできない。ノードは、この 16 ビットの MAC ショートアドレスから形成された IID を使用して、IPv6 グローバルユニキャストアドレス(GP16)と IPv6 リンクローカルアドレス(LL16)を設定する。

10. ノードは、[6LPND]で記述されるとおりにグローバルユニキャストアドレスの DAD(重複アドレス検出)の手順を実行する。親ルータは、ZIP コーディネータと GP16 アドレスを登録し、一意性を確認するために DAR/ DAC パケットを使用する。これはまた、16 ビットの MAC ショートアドレスは、ZigBee の IP ネットワーク内で一意であることを意味することに注意すること。 GP16 アドレスが重複であると判断された場合、そのノードは別の GP16 アドレスを選択し、このプロセスを繰り返す。ノードが 6LoWPAN 近隣探索プロトコルの交換中に、その IPv6 ソースアドレス([6LPND]で必要とされる) および 要求している GP16 アドレスを使用する必要があることに注意のこと。16 ビットの MAC ショートアドレスは、ユニークであることが確認されるまで使用することはできない。したがって、このメッセージ交換は、64/16 アドレッシング･モード混合である。(すなわち、IPv6 アドレスは IID のように 16 ビットの MAC アドレスを使用して形成されているが、使用される MAC アドレスは 64 ビットのアドレスである)

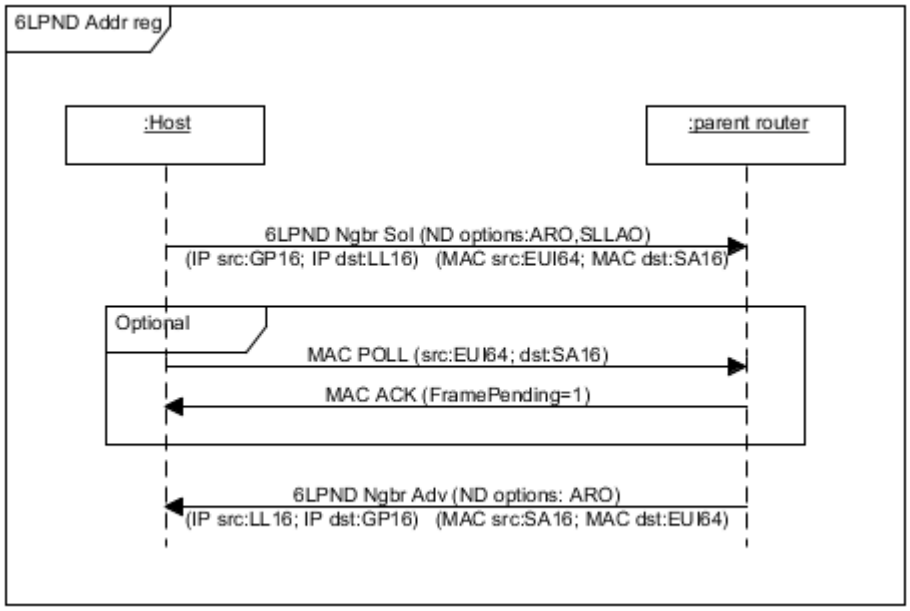


図6-5：Join sequence - Address registration

11. ノードは、3 ウェイ MLE ハンドシェイクを実行し、親ルータとのショートアドレスを交換する。ノードは、ユニークな 16 ビットのショートアドレスをリンク要求(Link Request)またはリンク応答(Link Accept)パケットの MLE ペイロードに含めなければならない。この手順の最後で、ノードは親ルータのショートアドレスを知り、親ルータはノードのショートアドレスを知る事ができる。ノードがスリープホストである場合には、MAC のポーリングを実行するために、親ノードをショートアドレスで更新し次第、すぐに 16 ビットのショートアドレスを使用して開始しなければならない。

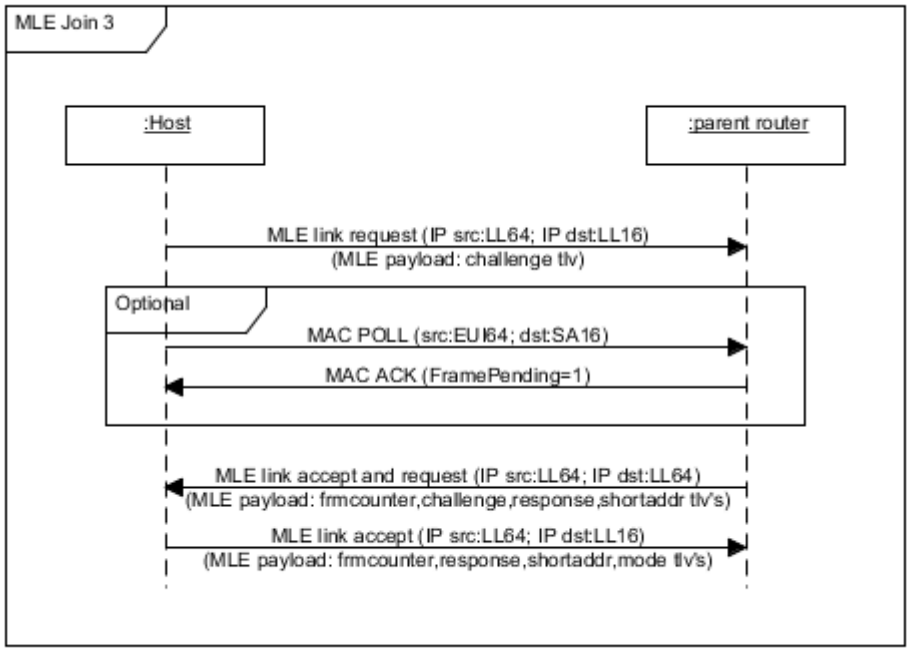


図6-6：Join sequence - MLE 3

12. 親ルータは、新しいノードが ZIP ホストであるかをチェックしなければならない。 MLE メッセージのモード TLV(セクション6.2.10.1を参照)をこの決定を行うために使用するべきである。参加ノードがホストである場合、親ルータは、新しいノードに下方ルートを作成するために DODAG ルーツに RPL DAO メッセージを送信しなければならない。DAO メッセージは、ターゲット Prefix オプションに参加しようとしているノードの GP16 アドレスと、とトランジット･オプションに親ノードの GP16 アドレスを含めなければならない。外部(E)フラグを 1 に設定しなければならない。

ホストのブートストラッピングは以上。これでホストノードは、親ルータを通して IP パケットを送受信することができる。

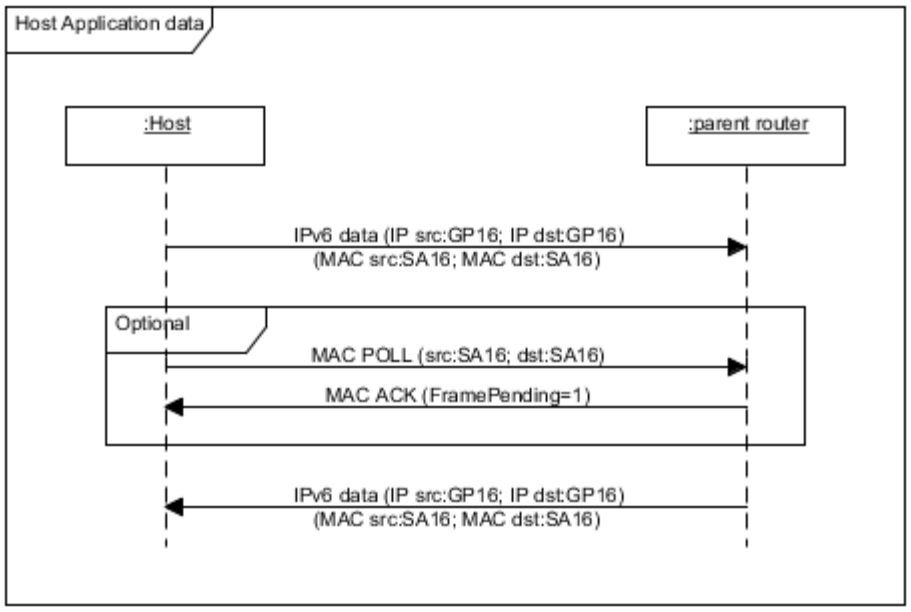


図6-7：Join sequence - Application data

6.3.5.2. ルータのブートストラップ

ZIP ルータのブートストラップシーケンスを、以下に記載する。

1. ZIP ルータは、以下の例外を除いてホストノードで説明したブートストラップシーケンスに従う。ZIP ルータは、提示された使用可能ルータ能力(indicated available router capacity)を持つルータの中から、最初のペアレント・ルータを選ばなければならない。それはビーコン・ペイロード内の router capacity サブフィールドにセットされた 1 で示される。ZIP ルータはスリープノードにすることはできないので、PANA 認証(ホスト手順のステップ 5)の前の最初の MLE 交換はオプションである。それは最終ステップ(ホスト手順のステップ 11)までホスト手順を進め、そして、以下の通りに続ける。

2. ZIP ルータは、近隣 ZIP ルータノードを検出し、安全なレイヤ 2 リンクを構成する。これは、MLE ハンドシェイク交換を使用して行われる。最初の MLE リンク要求パケット(MLE Link request packet)は、MAC ブロードキャストアドレスを使用して送信される。無線範囲内にあるすべての ZIP ルータはこのパケットを受信する。追加のレイヤ 2 リンクを構成する使用可能能力に依存して、MLE リンク accept および MLE リンク要求をもって応答しても良い。(レイヤ 2 リンクを構成する能力は MAC デバイステーブルのサイズによって制限されることに注意)

   参加しようとしているルータは応答した ZIP ルータからサブセットを選択し、それぞれの MLE リンクの確立プロセスを完了する。サブセットの選択はこの仕様書の対象外とする。

これにより、参加しようとしているルータの MAC デバイステーブルに選択した近隣ルータのエントリが取り込まれる。参加しようとしているルータは、この時点で MAC デバイステーブルのキャパシティのすべてを使用していないことを確認するべきである。他のジョインノードが後にネットワークに参加できるようにするために、MAC デバイステーブル内のキャパシティの余裕を持つべきである。

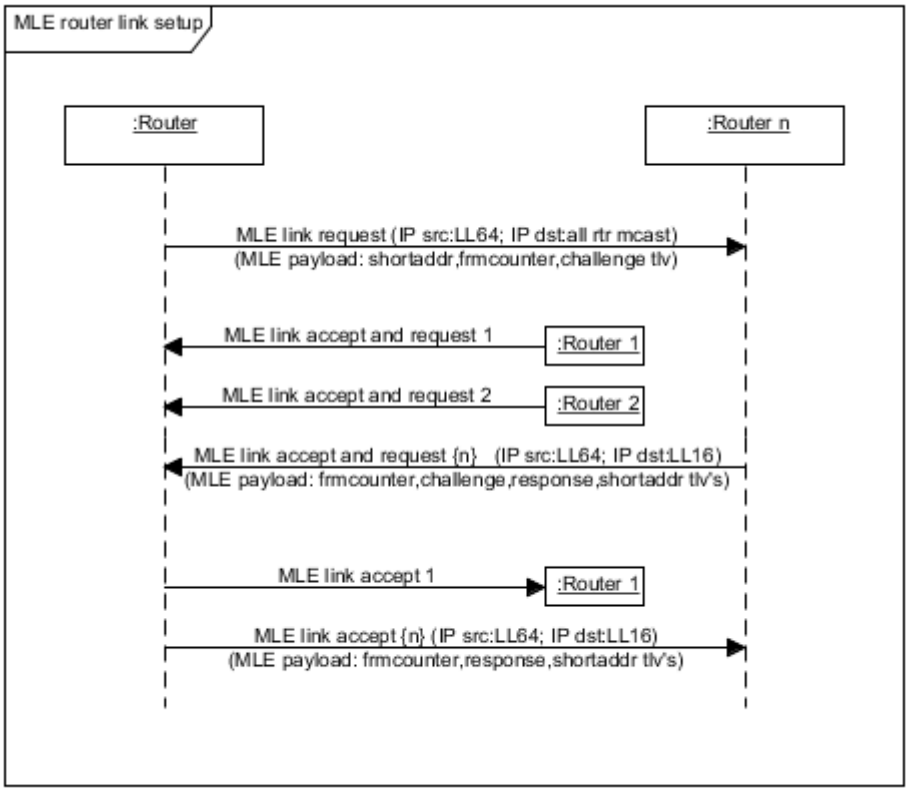


図6-8：Join sequence - Router link setup

3. 次に、ZIP ルータは、RPL のルーティング・プロトコルの構成(コンフィグレーション)を開始する。使用可能なすべての RPL インスタンスを検出するため、マルチキャスト DIS パケットを送信する。ノードは、以下のメッセージのシーケンスを使用して、順番にそれぞれの RPL インスタンスを参加させる。

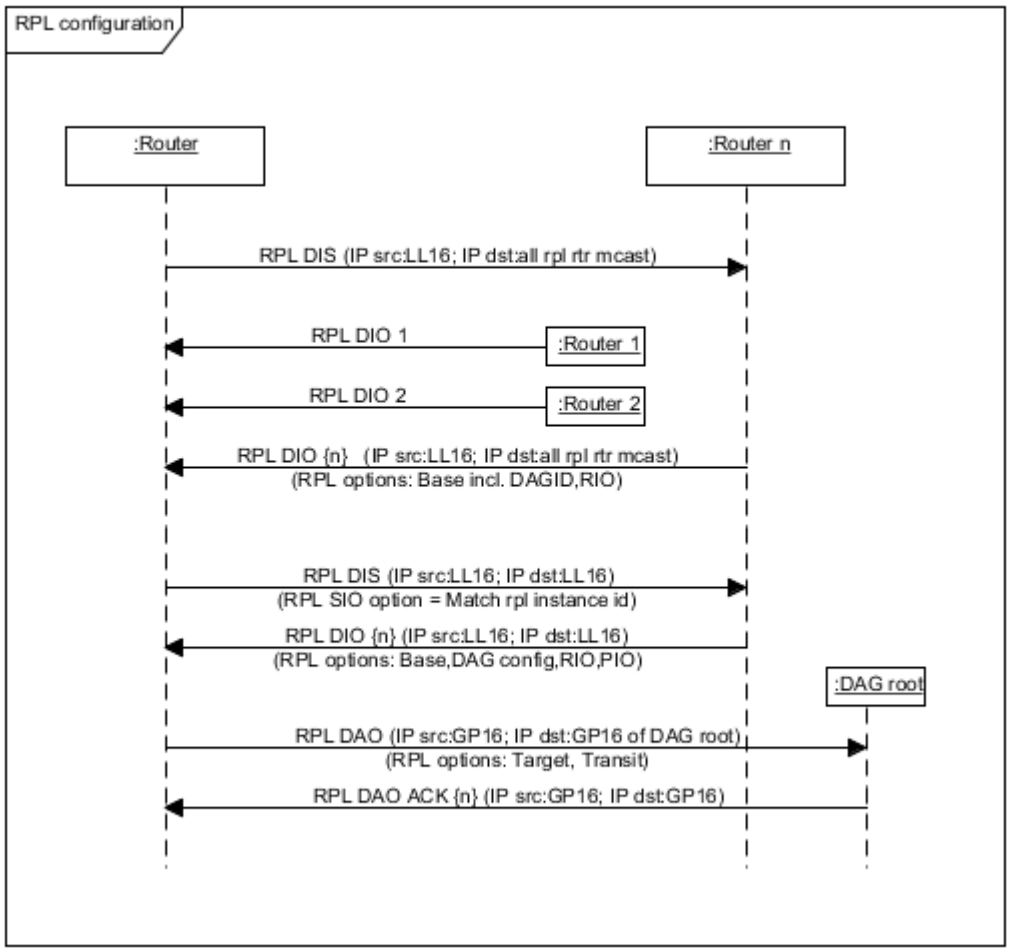


図6-9：Join sequence - RPL configuration

4. ZIP ルータはネットワークの一部となり、完全なコミュニケーション能力を持っている。ブートストラップシーケンスの最後のステップは、ネットワークに新しいノードを認めることができるように、アクセスルータとして機能するために自分自身をコンフィギュアすることである。このためには、セクション6.3.3.1で説明したように MAC ビーコン・ペイロードを設定し、それが受信ビーコン要求パケットに応答して、ビーコンパケットを送信できるように、MAC コーディネータサービスを開始しなければならない。ビーコン中のアソシエーション許可フラグを false に設定しなければならない。そして、PANA のリレーサービスを可能にしなければならない。MLE リンクアドバタイズメントパケットの周期的な送信を開始しなければならない。セクション6.3.9.3.6に述べられているように PANA 認証サーバを新しい GP16 アドレスで更新しなければならない。

6.3.6. ネットワーク認証

新しいノードは、ZigBee の IP ネットワークに参加するときに ZIP コーディネータに自身を認証させ、MAC セキュリティマテリアルへのアクセスを得るために PANA プロトコルを使用する。ノードがネットワークに許可されたら、ネットワーク上のすべての通信機能へのフルアクセスが出来る。

認証サーバは、ネットワークからすでに認証されたノードを拒否することを選択することができる。それはアクセスを無効にしたものを除く、すべてのノードへのキーネットワークの選択的更新を実行することに

よって行うことができる。認証サーバは完全にそのノードのネットワークアクセスを取り消すために、2 回のネットワークキーの更新を実行しなければならない。更新ネットワークキーの詳細については、6.3.10を参照のこと。

6.3.7. 6LoWPAN フラグメントの再統合

ZIP ノードは順番に 6LoWPAN フラグメントを送信し、同じネクストホップノードに別の送信を開始する前に、現在の IP データグラムの送信を完了しなければばらない。これは、受信するノード上で多くの最適化を可能にする。

ZIP ノードは各隣接ノードから送られてくる､フラグメント化された受信メッセージを最大で1つバッファリングするべきである。近隣からフラグメント化されたメッセージの受信をする場合、その近隣から受信した 6LoWPAN パケットが予想される次のフラグメントでない場合、その部分的なメッセージは廃棄されてもよい。また、先頭ではないフラグメントが期待された次のフラグメントではない場合、受信したフラグメントや、部分的なメッセージについても破棄することができる。

6.3.8. スリープノードのサポート

ZigBee IP ネットワーク中のホストは通常バッテリー駆動で､短い時間だけしか無線機を使う事ができない。そのようなホストはスリープホストと呼ばれている。ZIP ルータはスリープする事はできない、常に無線を有効にしなければならない。

Sleepy ホストノードは[802.15.4]に定義されたデータリンク層を使った[indirect transmission scheme]にてデータを受信する。この方式では、送信ノードは、発信 MAC パケットをバッファリングする。スリープホストは、無線をアクティブにすると、親ルータに MAC POLL コマンドパケットを送信し、受信機能を有効にする。親ルータは、MAC POLL コマンドパケットの応答として確認応答パケットを送信し、スリープホストノード宛のパケットがバッファリングされているかを示す。スリープノードは、親ルータがパケットをバッファリングしたことを確認すれば、受信を維持し続けるだろう。これによって、親ルータがアクノリッジメントパケットを送信した直後、バッファリングされたパケットを全てスリープホストに送信することができる。

ZIP ルータは、スリープホストノードへの track(記録)を維持しなければならない。ZIP ルータは MLE メッセージのモード･タイプ･オプションを介してこの情報を取得する。[802.15.4]で定義されているように、これらのノードへのパケット送信は、MAC 間接スキームを使用するべきである。ZIP ルータは、フル IPv6 パケットを少なくとも MAC_MIN_INDIRECT_BUFFER 分、バッファリングする能力を持たなければならない。間接的伝送(indirect transmission)のためにバッファリングしたパケットは正常に送信されるまでか、または MAC_MIN_INDIRECT_TIMEOUT の期間中、キューに入れなければならない。ZIP ルータは、MAC ビーコン・ペイロードのホストキャパシティビットをクリアすることによって、スリープホストがそれらを親ルータとして選択する事を防ぐことができる。ZIP ルータが、確実にサービスを提供できるスリープホストノードの数の内部制限数に達した場合、これを実行するべきである。

スリープホストが動的にスリープの状態を変更する可能性があることに注意すること。スリープホストは親ルータに、そのスリープ状態を変更するたびに、そのステータスを更新しなければならない。これは、MLE メッセージのモード･タイプ･オプションを使用して行われる。例えば、もしスリープホスト上のアプリケーションが大量データストアが必要であると認識した場合 (ノードが新しいファームウェアのアップデートを受信しているケース)、スリープしていないホストに状態を変更し、ダイレクにデータ転送によってパケットを受信することができる。これにより、親ルータのバッファへの負担を軽減し、また、データがより速く、より信頼性の高い転送出来るようになる。

スリープホストデバイスは､通常 アプリケーションレベルのトランザクションのイニシエーター(開始者)

であることが期待されている。スリープホストデバイスは、通常 予期しないパケットを受信すべきではない。スリープホストノードが、パケット受信することが分かっている場合、親ルータがバッファリングしているパケットを正常に受信する確率を向上させる為に、通常より早い速度でポーリングする事ができるべきである。

ZigBeeIP ネットワーク内のスリープホストに対応するために特別な対策が必要である。以下に説明する特別な対策は、ホストが参加プロセス中でも、間接的な伝送(indirect transmission)を使用して通信することができる。

6.3.8.1. スリープホストの参加

ノードがブートストラップする最初のプロセスは、セクション6.3.5.1に示されていて、次のテキストは追加分の詳細が記述されている。

Sleepy ノードはショート MAC の要求なしに帰属プロセスを開始する。まず初めに、MAC にてデータ送信する時のソースアドレスは帰属するホストの 64BIT の MAC アドレスである。

スリープホストは、最初のブートストラッププロセス中に、親ルータにその性質を示すべきである。これは、MLE リンク要求メッセージ(セクション6.3.5.1のステップ 5 を参照)を介して行われる。モード TLV は、リンク要求メッセージに含まれており、値が[802.15.4]で定義されている"機能情報"フィールドが含まれている。

親ルータは MLE リンクメッセージを受け入れるか、拒否すると応答しなければならない。ホストは、MAC 間接的な伝送を使用して、ジョインホストへの応答を送信しなければならなず、そのためにポーリング処理を実行する。新しいリンクを確立するための要件として、指定した時間の間、少なくとも 1 つの IPv6 パケットをバッファリングする機能を持っていない限り、ZIP ルータは、子としてスリープホストを受け入れてはいけない(MAC デバイス内のスペーステーブルなど)。ZIP ルータはスリープホストノードにサービスを提供するために必要な能力を持っていない場合は、MLE のリンク要求に応答してメッセージを拒否しなければならない。

注意：

スリープノードは、セクション6.3.5.1で説明したブートストラップシーケンスのステップ 10(ネイバーディスカバリ)で、ユニークなショートアドレスを確認しているが、ブートストラップシーケンスのステップ 11 の間で親ノードが新しい情報を更新するまでそのショートアドレスをデータリンク層で設定してはならない。そのときまで、ジョインノードはその拡張アドレスを使用してポーリングされる。ノードは、MAC ポーリングのために拡張アドレスを使用し、その後、ショートアドレスを使用しなければばらない。

6.3.8.2. ポーリングレート

ホストは、DEEP スリープと SHALLOW スリープの 2 つのスリープモードを持つ。 スリープホストノードは、2 種類のスリープモードでは、高速ポーリング(fast poll)と低速ポーリング(slow pol)と呼ばれるものがある。2 つのモードの違いは、MAC ポーリングレートである。

高速ポーリング中のスリープノードは、妥当な時間内にそのパケットを受信するため十分な頻度で親ルータにポーリングするべきである。どの程度ポーリング間隔が妥当であるかは、上位レイヤの再送タイマに依存している。例えば、TCP では初期再送タイムアウトから、連続する再送間隔の 3 秒ずつ増加され設定されている。高速ポーリング状態では、不要な再送を無くすため、ホストは少なくとも MAC_MAX_FAST_POLL_TIME 分、親ルータへのポーリングしなければならない。

低速ポーリング状態のスリープホストは、大幅にポーリングレートを遅くすることができる。スリープデバイスは、いつでも低速のポーリング状態へ遷移することができる。デバイスが低速ポーリング状態へ遷移

できるようにしたい場合には、MLE 交換時のタイムアウト TLV を含むことによって、リンクの確立プロセスの中で親に伝えなければならない。タイムアウト TLV は、連続するポーリング(すなわち、低速ポーリング状態時のポーリング周期)の最大間隔を示している。タイムアウトのフィールドの値は MAC_MAX_POLL_TIME 以下の数でなければならない。スリープホストが低速ポーリング状態にあるとき、MAC_MIN_INDIRECT_TIMEOUT の IP パケットをバッファリングする事が親ルータへの要求事項であることに注意のこと。このような理由から、スリープホストノードが低速ポーリング状態の場合は、パケットを受信できない可能性が非常に高くなる。

スリープノードは、パケットを受信するように想定している場合は、高速ポーリングするべきであり、それ以外の場合は、低速ポーリング状態とすることができる。例えば、MDNS または、HTTP 要求を送信し、応答を待機している場合は、高速ポーリング状態であるべきである。

ZIP ノード上で動作するアプリケーションは、ノードが低速ポーリング状態になることがあり、スリープホストノードが常に到達可能で無い事に注意するべきである。それは、ノードが問合せを送信した後に妥当な期間の高速ポーリング状態であることが予想され、スリープホストによって開始された問合せ(例えば、MDN または HTTP)に対応するのが一般的で安全である。

6.3.8.3. データリンク層のデータリクエストコマンドフレームのセキュリティ

データリンク層で暗号化されていない MAC データ要求コマンドフレームは、常に送信される。(すなわちポーリング) 具体的には、親はデータリンク層で子からのセキュリティで保護されていないポーリングを廃棄してはいけない。 リンク確立した子が、存在する場合でも同様である。その理由は、子がネットワークを再ジョイン または、キースイッチ後にキー更新を実行し、現在のネットワークキーを持っていない可能性がある事である。親は、常にセキュリティで保護されていないポーリングを受け入れるので、それらを確保するためにスリープしている子のための理由はない。ネットワークキーを持っている場合でも同じである。

6.3.8.4. スリープホストノードのリンク維持

ノードが Deep Sleep モードである場合、ネットワーク状態が変更されている可能性がある。たとえば、ネットワークキーが更新されて、親ルータとの無線リンクが切断されている可能性がある。このセクションでは、スリープホストノードが、ネットワーク状態を維持する為の診断措置と対処について説明する。

スリープホストノードの通常処理は、定期的にウェイクアップし、親ルータに MAC Poll コマンドパケットを送信し、応答の MAC 確認応答パケットを受信する。また、この時にアプリケーションパケットを送信することができる。アプリケーションが応答を期待している場合、正常に応答受信された場合 または、タイムアウトするまでノードは高速ポーリング状態に遷移するべきである。

スリープホストは、アプリケーションパケットを送信し、応答パケットを受信した場合、そのネットワーク状態が変わらず、正常に動作し続けることの十分な確認となる。

それは、UNAVAILABLE_KEY 状態を持つ内部 MAC COMM-ステータス表示[802.15.4]を介して、ノード上の管理エンティティで検出することができる。 この場合、PANA のネットワークキーの更新プロセスを開始し、認証サーバから新しいセキュリティマテリアルを取得するために、スリープホストノードを起こすべきである。スリープノードは、セクション6.3.10.2に述べられているような周期的な Key Pull オペレーションを行うことにより、新しいセキュリティマテリアルを積極的にチェックすることができる。

中間のデータリンク層が NO_ACK 状態でデータ確認パケットを受信した場合、スリープホストの管理エンティティは、親ルータと無線リンクの損失を検出することができる。この場合、新しい親ルータを探索し、登録するために、スリープホストノードを起こすべきである。セクション6.3.5.1のステップ 2 で説明したようにスリープホストは、MAC ビーコンメカニズムを通して新しい親ルータを発見することができる。親ルータを選択した後、スリープホストは必要なセキュリティマテリアルと IPv6 アドレスの構成情報へのアクセ

ス権を持っていて、新しい親ルータ(セクション6.3.5.1のステップ 10, 11)をアドレス登録し、保護された MLE 交換を実行する。

スリープホストは、長期間 アプリケーションデータパケットを送信しなかった場合、積極的にネットワーク状態を確認する場合がある。例えば、親ルータに ICMPv6 エコー要求を送信することによって実行することができる。期待される ICMPv6 エコー応答 または、エラー表示のいずれかをになるべきである。この処理の利点は、重要な更新などのネットワーク変化の早期発見である。コストは余分パケットの交換。コスト対効果は、実際の展開シナリオに依存するので、アプリケーションに一任されている。

スリープホストが、親ノードからアプリケーションパケット(ICMPv6 エコー要求含む)を送信し、期待される応答 または、MAC エラー表示を受信していない場合は、ネットワーク･セキュリティ･マテリアルが複数回更新されていることを示している。この状態から回復するには、スリープノードは通常の鍵更新の手順を使用することはできない。代わりに、新しい親との最初の MLE 交換(6.3.5.1の手順 5)を実行し、ビーコンを要求することによって、新しい親を探索してネットワークに再接続しなければならない。ネットワークキーを取得するための PANA 認証の代替として、「key pull」を実行し、新しい親(6.3.5.1の手順 11)と保護された MLE 交換を行う。ネットワークへの再参加の手順には、パケット交換を含む。スリープノードは、親ノードとの通信に数回失敗した場合、再参加を行うべきではない。

6.3.9. ネットワーク認証

ネットワークへのジョインプロセス中に、ノードはそれが正しいネットワークであることを確認し、必要なセキュリティ資格証明を取得するためにネットワーク認証を実行する。同様に，ネットワークは、ノードが信頼できるものであり、ネットワークに参加するのに必要なセキュリティ証明書を持っていることを保証することを認証する。

認証手順の目的は、以下の結果をもたらす相互認証を提供することである：

- 適切な資格情報を持たない信頼されていないノードが、信頼された ZigBee IP ネットワークへ参加するのを防ぐ。

- 適切な資格情報を持つ信頼されるノードが、信頼されていない ZigBee IP ネットワークへ参加することを防ぐ。

認証サーバは、ZIP コーディネータに常駐し、ネットワーク上のノードを認証する責任がある。認証に成功した場合、認証サーバは PANA プロトコルを介して参加ノードにネットワークセキュリティマテリアルを送信する。参加ノードは、ZigBee IP ネットワークの参加ノードになり、ネットワーク内の他のすべてのノードと IP パケットを交換することができる。

EAP-TLS サーバが、新しいノードを認証できない場合は、認証は認証サーバ上で失敗しなければならない。これは、EAP-TLS ハンドシェイク中に提示されたセキュリティ資格情報に依存する。

また、認証は本標準の範囲外であるアプリケーション･ロジックによって失敗する可能性がある。このようなアプリケーションロジックの例は、ZIP コーディネータのユーザーボタンである。そこでは、ボタンが押された後、すぐにしなければすべての参加の試みが拒絶される。 そのような状況では、ZIP コーディネータは、ネットワークを外れ、再参加をしようとしているノードからの参加の試みを受け入れるべきであることに留意すべきである。 別のアプリケーションロジックの例は、ノード ID の明示的なホワイトリスト または、ブラックリストである。

参加ノードは、最初 ネットワーク･セキュリティ･マテリアルへのアクセス権を持っていない。したがって、認証プロセス中に交換されるパケットのデータリンク層のセキュリティを適用することはできない。ZIP ルータのエンフォースメント・ポイントのルールは、セクション6. 3. 9. 4で説明され、データリンク層でのセキュリティ保護されていないにもかかわらず、PANA の認証に関与しているパケットが処理されることを

保証する。そのルールは、データリンク層で保護されていない他の受信トラフィックは ZIP ノードによって破棄され転送されないことも保証する。

6.3.9.1. 認証スタック

認証は、一つの層がその上の層をカプセル化するようなプロトコルスタックと見ることができる。ZIP の認証プロトコルは、下図の中に他の関係で示される。

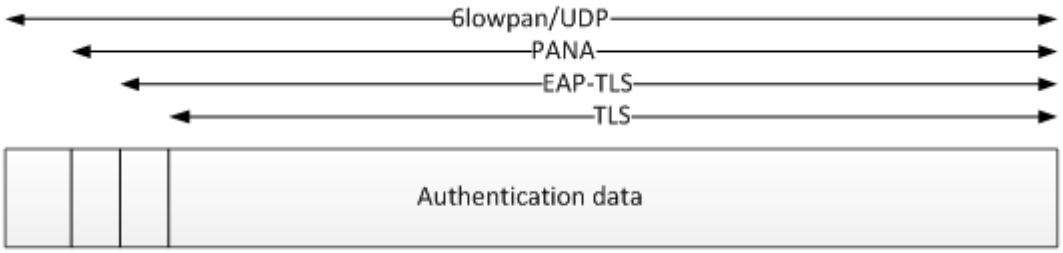


図6-10：ZigBee IP ネットワーク内の認証プロトコルスタック

TLS[TLS]は最上位層で使用され、認証情報を交換しければならない。事前共有鍵に基づいての暗号スイート[TLS-CCM] と ECC[TLS-CCM-ECC]に基づいて暗号スイートがある。

認証プロトコルの TLS レコードを転送するため、EAP-TLS[EAP-TLS]は次の層で使用されなければならない。

拡張認証プロトコル[EAP]は、相互認証のメカニズムを提供するために使用されなければならない。EAP は、参加ノードと認証サーバが常駐するノード間で EAP パケットを転送する方法を必要とする。これらのノードは、お互いの無線範囲内にあるとは限らないので、EAP トランスポート方式においてマルチホップをサポートすることが必要である。PANA プロトコル[PANA]、[PANA-RELAY]は、UDP 上で動作するが、この目的のために使用されなければならない。[EAP]は主要な階層構造を使用して、セッションキーの導き出しを規定する。[PANA]は、それが PANA の認証 および 暗号化キーを設定するために使用され、EAP マスター・セッション・キーを導き出さなければならない。

PANA(RFC5191)[PANA]および PANA リレー[PANA-RELAY]は、次の層で使用されなければならない:

- 参加ノードは、PANA のクライアントとして機能しければならない。 (PaC)
- 親ノードは、認証サーバでないならば、[PANA-RELAY]に従い、PANA リレー(PRE)として機能しければならない。すべての ZIP ルータは PRE の役割で機能することができなければならない。
- 認証サーバノードは、PANA 認証エージェント(PAA)として機能しければならない。
- 認証サーバは、[PANA-RELAY]に従って中継されるパケットの処理ができなければならない。

このネットワークの認証プロセスは、新しいノードと親の間の転送のためにリンクローカル IPv6 アドレスを使用する。親が認証サーバでない場合は、PANA リレーメカニズム[PANA-RELAY]を使用して、認証サーバへ参加ノードからパケットを中継しなければならない。参加ノードは、最初の PANA の認証メッセージ交換のために、送信元アドレスとして LL64 アドレスを使用しなければならない。

6.3.9.2. 適用性宣言(Applicability statements)

適用性宣言では様々な仕様書の関係について記述する。

.

6.3.9.2.1. PSK TLS に対する適用性宣言

[TLS-CCM]は、AEAD の一部が[AEAD] で詳述される、[TLS-PSK-GCM]と非常に類似している AEAD TLS の暗号スイートを含む。 [TLS-PSK-GCM] のリファレンスとしては [TLS-GCM]とオリジナル PSK 暗号スイートドキュメント[TLS-PSK]がある，それらは[TLS]を参照し,[TLS]は TLS 1.2 のメッセージを定義する。

6.3.9.2.2. ECC TLS に対する適用性宣言

[TLS-ECC-CCM]には AEAD の一部が[AEAD]で詳述される[TLS-ECC-GCM]と非常に類似している AEAD TLS の暗号スイートが含まれる。[TLS-ECC-GCM] のリファレンスとしては オリジナル ECC 暗号スイートドキュメント[TLS-ECC](RFC4492)がある。[TLS-ECC](RFC4492)は [TLS]を参照し,[TLS]は TLS 1.2 のメッセージを定義する。

6.3.9.2.3. EAP-TLS と PANA に対する適用性宣言

[EAP-TLS]は、EAP パケットへ[TLS]メッセージをパッケージにするために、[EAP]がどのように使用されるかを規定する。[PANA]は EAP パケットとベンダ仕様の attribute-value pairs (AVPs) にて転送される捕捉設定情報と、[PANA-ENC]とこのドキュメントで定義される暗号化された (AVPs)に対する転送手段を規定する。SHA-256 に基づいて提案された PRF と AUTH のハッシュは、[IKEv2 の](RFC5996)と[IPSEC-HMAC](RFC4868)に詳細が定義される。

6.3.9.3. PANA

6.3.9.3.1. PANA セッション

[PANA]は PANA セッションに対し、いくつかの段階を規定している。ZigBee IP PANA セッションは、Authentication／Authorization のどちらかの段階でなければならない。ZigBee IP PANA セッションは常に PaC によって開始されなければならない。PaC と PAA の間の ZigBee IP PANA セッションは、ネットワークキー更新およびメンテナンス用にオープンしたままにしなければならない。

6.3.9.3.2. PANA セキュリティアソシエーション

[PANA]仕様は、PANA セキュリティアソシエーションが、EAP マスター・セッション・キーから認証キーを生成して、この認証キーが最後の PANA メッセージを認証するために使用される。[PANA-ENC]仕様は、暗号鍵を派生する。それは、ネットワーク転送用暗号化キーおよびノードへ DATA フレームに付属するネットワークキーインデックスに使用されなければならない。

PAA は、[PANA]で指定されたものに加え、安全なアソシエーションの一部として、以下の属性を維持しなければならない。

- PAC の EUI-64。これは、この安全なアソシエーションに関連付けられている PAC の LL64 アドレスから導き出されるべきである。この情報は、一意の PAC を識別し、重複したセッションを防止するために使用されている。.
- ノードの認証カウンタ。これは PAA に保存して、ネットワーク･セキュリティ･マテリアルの一部として、PAC に転送される 1 オクテットの値である。

6.3.9.3.3. 参加ノード(PaC) と親ノード(PRE または PAA)の間の PANA

接続ノードと親ノードの間の PANA メッセージは、次のヘッダ・アドレスの二方向にシングルホップのユニキャスト送信を使用しなければならない:

表6-22：PANA 接続ノードヘッダアドレス

Address	Value	Comment
MAC address	64-bit	IEEE address of the Joining Node
IP address	LL64	Stateless autoconfigured link-local address of joining Node

表6-23：PANA 親ノードのヘッダ・アドレス

Address	Value	Comment
MAC address	16-bit	Short address of the Parent Node
IP address	LL16	Stateless autoconfigured link-local address of parent node

6.3.9.3.4. 親ノード (PRE)と認証サーバの間の PANA

親ノードと認証サーバが同じノードではない場合、親ノードは[PANA-RELAY]に従って接続ノードと認証サーバの間で交換された PANA メッセージを中継しなければならない。中継は接続ノードにとっては透過になる。それに関する限り、それ(接続ノード)は直接、認証サーバとやりとりをすることになる。

親ノードと認証サーバの間の中継された PANA メッセージは、標準ユニキャスト送信を両方向に使用しければならない。中継された PANA メッセージは、リンクレイヤで安全になりそれにより、[PANA-RELAY]のセクション 3 の必要条件を満たし代わりパケット保護の必要性を回避することができる。

6.3.9.3.5. ネットワークセキュリティマテリアルの転送

PANA 認証が成功した場合は、PAA から PAC への最終 PANA 認証要求メッセージにネットワークセキュリティマテリアルを含めて、参加するノードへ送信しなければならない。ネットワークセキュリティマテリアルは、ENCR-ENCAP AVP [PANA-ENC]を使用して、暗号化されているネットワークキーAVP(セクション6.2.6.3を参照)で転送されなければならない。ネットワークキーとインデックスの値は、アクティブなネットワークセキュリティマテリアルを含んでいなければならない。ノードの認証カウンタの値は、そのノードの PANA のセキュアアソシエーションステートから取得しなければならない。

PANA 認証が完了した時点で、PAA はこのノードと重複したセキュアアソシエーションを持っているノードがあるか確認しなければならない。重複したセッション情報をチェックすることを目的として、PAA はノードの EUI-64 MAC アドレスを使用すべきである。このアトリビュートは、PANA 認証時に PAC で使用され、セッション情報の一部として格納されている LL64 アドレスから取得される。

重複したセキュアアソシエーションが見つかった場合、PAA は、重複したセキュアアソシエーションからノードの認証カウンタ値を取得し、インクリメント(ロールオーバでゼロ)して、新しいセキュアアソシエーションにコピーしなければならない。さらに、古いセッション情報を削除しなければならない。それ以外の場合は、PAA はセキュアアソシエーションのノード認証カウンタアトリビュートにゼロを設定すべきである。

6.3.9.3.6. PaC アドレスの更新

ZIP ノードは、PANA の認証プロセス中にリンクローカル IP アドレスを使用する。結果として、各ノードの PAA セキュアアソシエーションは、リンクローカルアドレスを含む。認証が完了した後は、ブートストラッププロセスはグローバルユニキャスト(GP16)の IP アドレスのコンフィギュレーションとなる。[PANA]はノードが PANA 通信に使用する IP アドレスを変更した場合、PAA でそのアドレスを更新しなければならないことを要求する。

ZIP ルータは、ブートストラッププロセスを完了した後、GP16 アドレスを PAA サーバへ更新しなければならない。これは、ソース IP アドレスとして GP16 で PAA に有効な任意の PANA のパケットを送信することによって実現される。通常は、PANA の通知要求メッセージは、この目的のために使用される。PAA でその IP アドレスを更新した後、ノードおよび PAA は、直接グローバルユニキャスト IP アドレスを使用して通信することができる。.

ZIP ホストは、GP16 アドレスを PAA サーバにその IP アドレスを更新するべきではない。ZIP ホストは、通常 スリープデバイスであるため、必ずしも他のノードから到達可能ではない。したがって ZIP ホストは、PAA との通信にローカル IP アドレスを引続き使用すべきである。これらの通信は、その PAA にリレーする親ルータにある PANA リレーエンティティに宛てられなければならない。

6.3.9.4. EP(Enforcement Point)の処理

すべての ZIP ノードはEP(エンフォースメント・ポイント)機能をインプリメントしなければならない。EP は、レイヤ 4、すべての外部のノードからこのように効果的にファイアウォールの通信にすべてのレイヤでノードを起動に入るすべてのトラフィックを取り締まる(policing)ポリシングすることによって作用する。EP は、設定およびパケットの特性に依存しているフィルタリングルールを持つ。フィルタリングルールは以下の通りである。これらの規則の実質的な効果はデータリンク層でセキュリティで保護されていないすべての着信 MAC のデータパケットは、それがノードに属している宛先アドレスを持つ IPv6 パケットが含まれていない限り破棄され、割当てられた PANA ポート番号(716)、または 割当てられた MLE ポート番号に UDP プロトコルを使用して送信されることである。

6.3.9.4.1. データリンク層のフィルタリング

- パケットは、L2 セキュリティ(ネットワークキー)によって保護されている場合は、EP はパケットに'"L2 セキュリティで保護された"としてタグを付け、さらなる処理のためにパケットの通過を許可し、フィルタリング、それ以上のレイヤをバイパスしなければならない。

- パケットは、L2 セキュリティ(ネットワークキー)によって保護されていない場合は、EP は'L2 セキュリティで保護されていない"などのパケットにタグを付け、レイヤ 3 フィルタリングのためにパケットを渡さなければならない。

6.3.9.4.2. ネットワーク層 のフィルタリング

- パケットが'L2 セキュリティで保護されていない"としてタグ付けされ、パケットがこのノード宛の UDP メッセージである場合、EP は第 4 層にパケットを通過しなければならない。 (宛先の IP アドレスのリンクローカルスコープのマルチキャストアドレスを含めて、このノードに割当てられたリンクローカルアドレスである。)

- そうでなければ EP はパケットを破棄しければならない。

6.3.9.4.3. トランスポート層 のフィルタリング

- パケットが"L2 セキュリティで保護されていない"とタグ付けされており、かつ、参加ノード(宛先ポートは割り当てられた PANA のポート番号に設定され、リンクローカルソースアドレスと宛先アドレスを使用した UDP データグラムとして特徴付けられる) または MLE パケット(割り当てられた MLE ポート番号への宛先ポートが設定された UDP データグラムとして特徴付けられる)からの PANA のメッセージである場合、EP はパケットを、それぞれのアプリケーション層に渡さなければならない。

  - MLE メッセージの場合は、"L2 セキュリティで保護されていない" メッセージの取扱いの規程は、さらに、 6.2.10.4で記述されている。PANA のメッセージの場合、プロトコルが低い層のセキュリティに依存しないので、追加の規定が必要ない。

- そうでなければ EP はパケットを破棄しければならない。

6.3.10. ネットワークキーの更新

ネットワークキーは、いつでも認証サーバによって更新することができる。そのような更新の頻度とタイミングは、実装依存である。前のキーからの更新 および 活性化(アクティブ化)が完了するまで、ネットワークキーの更新をしてはいけない。

通常は、認証サーバは、次のいずれかの理由のためにネットワークセキュリティマテリアルを更新する。

- 定期的に標準的な操作手順の一部として MAC フレームのセキュリティを確保するために使用されるセキュリティマテリアルを更新する。

- 現在のネットワークのセキュリティマテリアルを持っているノードへのネットワークアクセスを取り消す。

- 任意の ZIP ノードに対し、ノード認証カウンタが最大値に達するのを見越してセキュリティマテリアルを更新。

更新されたネットワークセキュリティマテリアルは、PANA のプロトコルを介して認証されたノードに配信される。それは "push"または"pull"どちらかのメカニズムで配信することができる。PAA は、すべての ZIP ルータに更新されたネットワークセキュリティマテリアルを "push"する。ZIP ホストは、PAA から更新されたネットワークセキュリティマテリアルを "pull"することが期待される。

認証サーバは、1 日から 1 ヶ月の間に定期的にセキュリティマテリアルを更新することが推奨される。ネットワークセキュリティマテリアルを少なくとの 1 ヶ月に一度更新する理由は、ノードフレームカウンターが、最大値に達しないようにするためである。しかしながら、セキュリティマテリアルが頻繁に更新されると、ネットワークに制御のオーバーヘッドがかかってしまう。また、スリープするホストがキーの更新に失敗し、ネットワークの接続を失うかもしれない。それゆえ、キーの更新は、1 日に一度以上は行わないことが推奨される。

ネットワークキーの更新処理の例は、**図 6-11**に示される。

6.3.10.1. PAA ネットワークセキュリティの更新手続き

ネットワークのセキュリティ更新プログラムは、認証サーバの管理エンティティによって開始される。

新しいネットワークセキュリティマテリアル(セクション6.2.6.2を参照)は、新しい 128 ビットのネットワークキーを生成することによって作成される。このキーのためにシーケンス番号が、1 つずつインクリメント

し、アクティブなセキュリティマテリアルのシーケンス番号を設定すべきである。現在のシーケンス番号が255 の場合は、新しいシーケンス番号は、1 にロールオーバーすべきである。ノード認証カウンタは、すべてのノードで「0」にリセットしなければならない。

新しいセキュリティマテリアルに加えて、管理エンティティは EUI-64 MAC アドレスによって識別されるノードのリストも提供してもよい。それらのノードは、ネットワーク上にあるが、さらにネットワーク･セキュリティ･マテリアルを受け取るべきではない。

新しいネットワークセキュリティマテリアルを取得する場合、PAA サーバは次のアクションを実行する：

1. PAA は、追加のネットワークセキュリティマテリアルを受信する資格のないノードに対応する PANA のセッションを削除する。

2. PAA は、それがセキュアアソシエーションであり、グローバルユニキャスト IP アドレスを持っている各ノードに新しいネットワーク･セキュリティ･マテリアルを "push"する。

3. "push"は、PANA の通知要求メッセージを送信することを含む。PAA は、ENCR-ENCAP AVP [PANA-ENC]を使用して暗号化されているネットワークキーAVP に更新されたネットワークセキュリティマテリアル(セクション6.2.6.3を参照)を含めなければならない。

PAA 管理エンティティは、上記処理が完了した後に新しいセキュリティマテリアルを有効にしてもよい。

新しい鍵の更新、活性化のプロセスが完了するまで、PAA は 2 つのネットワーク･セキュリティ･マテリアルを所持する。これが各ノードのノード認証カウンタの 2 つのコピーが含むことに注意すること。

6.3.10.2. ネットワークキーのプル(pull)

別のノードで新しいセキュリティマテリアルの使用を検出した場合、ZIP ノードはネットワークキーのプル(pull)を開始しなければならない。ノードが、現在キーインデックスよりも大きいキーインデックスを使用して、MAC または MLE 層で保護されているパケットを受信した場合に発生する。

6.3.10.2.1. 要求

ネットワークキーのプル(pull)は、PAA に PANA の通知要求メッセージを送信することによって開始される。ノードは、このメッセージを送信するときに以前の送信元アドレスとして PAA に登録されている IP アドレスを使用すべきである。 (セクション6.3.9.3.6を参照)。 これは、ZIP ホストの場合は、LL64 アドレスで、ZIP ルータの場合は、GP16 アドレスである。

ZIP ホストはこのパケットの送信元アドレスとしてリンクローカル IP アドレスを使用しなければならない。それは、親ルータにパケットを送信しなければならない。親ルータの PANA リレーエンティティは、透過的にこの要求をホスト-PAA 間の応答を中継する。

ZIP ルータは、それ以前にソース IP アドレスとして PAA に登録され、PAA に直接パケットを送信したグローバルユニキャスト IP アドレスを使用しなければならない。

ZIP のノードがキー要求の AVP をサポートする場合、ノードは、PANA 通知要求パケットの中にそれを含めなければならない。 nwk_key_req_flags は 1 の値に設定すべきである。nwk_key_idx フィールドには、現在アクティブなキーインデックスの値が取り込まれるべきである。

6.3.10.2.2. 応答

PANA の通知応答メッセージは、上記の要求に応答して、PAA から ZIP ノードに送信される。

PANA の通知要求の受信メッセージがキー要求 AVP を含まないか、または、PAA がキー要求 AVP をサポートしない場合は、PAA は目下キーの更新が進行中の場合は新しいネットワークセキュリティマテリアルを転

送し、そうでなければ、現在のネットワークセキュリティマテリアルを転送しなければならない。

PANA の通知要求の受信メッセージがキー要求 AVP を含み、PAA がこの AVP をサポートする場合は、PAA は以下のように応答しなければならない。

- nwk_key_req_flags フィールドの最下位ビットが 1 の場合は、
  - nwk_key_idx フィールドがアクティブなキーインデックスに等しいならば、PAA は新しいネットワークセキュリティマテリアルを転送しなければならない。そして、キーの更新が進行中の場合は、空の応答を送らなければならない。
  - nwk_key_idx フィールドがアクティブなキーインデックスに等しくないならば、PAA はアクティブなセキュリティマテリアルを転送しなければならない。
- nwk_key_req_flags フィールドの最下位ビットが 0 の場合は
  - nwk_key_idx フィールドがアクティブなキーインデックスに等しいならば、PAA はアクティブなセキュリティマテリアルを転送しなければならない。
  - そうでなければ、PAA は空の応答を送らなければならない。

PAA は、ENCR-ENCAP AVP [PANA-ENC]を使用して暗号化されたネットワークキーAVP(セクション 6.2.6.3参照)で、現在のまたは新しいネットワークセキュリティマテリアルを転送しなければならない。 新しいセキュリティマテリアルが転送された場合は、ノード認証のカウンタ値は 0 に設定しなければならない。そうでなければ、ZIP ノードに対応する PANA の安全なアソシエーションからの認証カウンタアトリビュートは 1 つ増やさなければならず、その値は、ネットワークキーAVP で使用されなければならない。

PAA がネットワークに参加しようとしている新しいノードにネットワークセキュリティマテリアルを転送する場合(すなわち、PAA から PaC への最終 PANA 認証要求メッセージにおいて)、常に現在のアクティブなネットワークセキュリティマテリアルをノードに転送しなければならないことに注意せよ。.

ZIP ホストは、定期的にそのマテリアルがアクティブになる前に、PAA でのセキュリティマテリアルが更新されているかどうかを確認するためにネットワークキーのプル(pull)手順を実行してもよい。ZIP ホストがキー要求 AVP をサポートする場合は、通知要求メッセージにそれを含み、nwk_key_req_flags の値を 0 に設定しなければならない。しかし、PaC か PAA のいずれかがキー要求 AVP をサポートしない場合は、次のネットワークキーの更新がゼロにリセットするまで、各ネットワークキーのプル(pull)はノード認証のカウンタ値の増分をもたらすので、これは慎重に行うべきである。認証カウンタ値がノードの最大値に達した場合、そのノードのフレームカウンタも上限に達する可能性があり、ノードがネットワーク内で安全に通信ができなくなる。

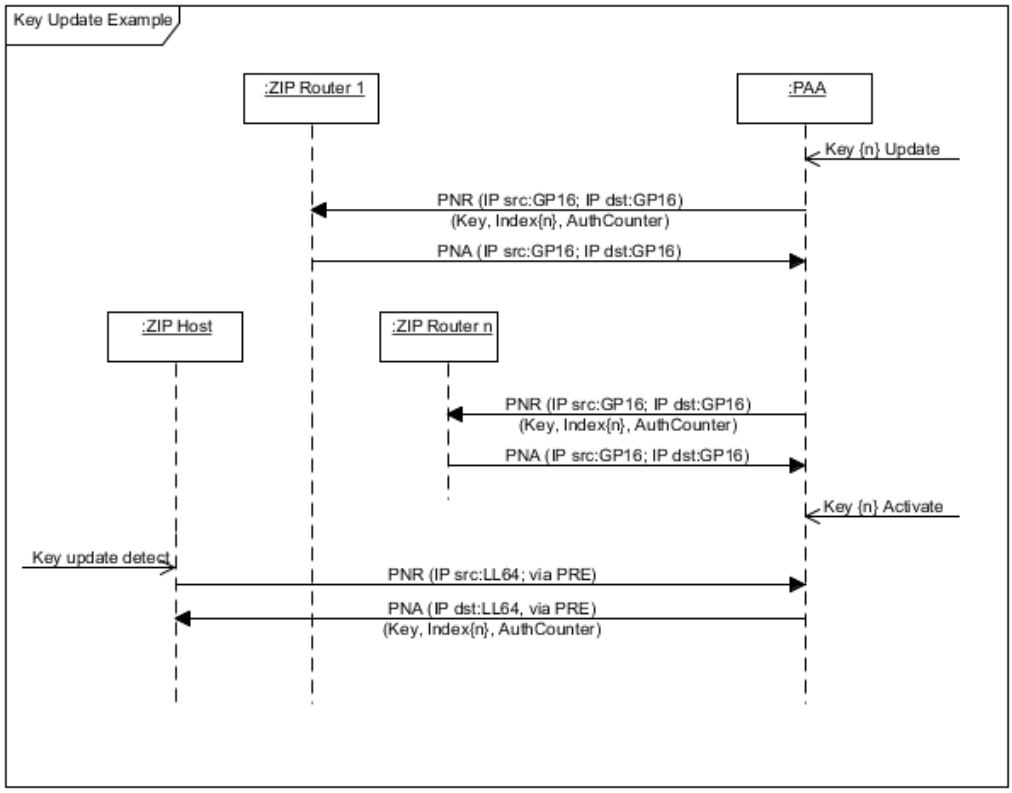


図6-11：ネットワークキーの更新

6.3.10.3. ネットワークキーのアクティブ化

認証サーバの管理エンティティは、新しいネットワークセキュリティマテリアルをアクティブ化する責任がある。

このアクションは、新しいセキュリティマテリアルが、ネットワーク内のすべてのスリープしていないノードによって伝送された後、すぐに実行される事が推奨される。これは、スリープノードが、PAA から新しいセキュリティマテリアルを「pull」する事ができるようにするためである

ネットワーク･セキュリティ･マテリアルのアクティブ化は、アクティブな MAC キーとアクティブな MLE キーがネットワーク･セキュリティ･マテリアルから引き出された時、それらの更新をもたらす。

PAA は、単純にキーがインデックスに新しいネットワークキーのシーケンス番号と一致する MAC と MLE のセキュリティマテリアルをアクティブにする。これは、発信 MAC フレームと新しいキーマテリアルで保護するために PAA から MLE メッセージの原因となる。

ZIP ノードが、現在のアクティブな MLE キーインデックスよりも高いキーインデックスを使用して保護された受信 MLE メッセージを受信した場合、そして、高いキーインデックスが代わりの MLE キーインデックスと等しい場合、アクティブな代わりのセキュリティマテリアルを交換しなければならない。

ZIP ノードは、現在のアクティブな MAC キーのインデックスよりも高いキーインデックスを使用して保護された受信 MAC メッセージを受信し、その高いキーインデックスで MAC KeyDescriptor を有しているた場合、そのアクティブな MAC キーインデックスの値をより高いキーインデックスへ更新する。

ZIP ノードは、MAC または MLE 層のいずれかのアクティブなセキュリティマテリアル更新時に、ノードの管理エンティティは他の層のアクティブなセキュリティマテリアルも更新すべきである。

6.3.11. ノードの診断

ZIP スタックは、データリンク層、アダプテーション層、ネットワーク層に対しノードマネージメントと診断機能を使用可能とする。これらの層のそれぞれについて、以下の情報を利用可能とすべきである。ノードの管理機能は常に利用可能でなければならない。しかし、診断と統計の収集は作動しても、しなくてもよい。

データリンク層は、ノード管理アプリケーションに利用可能な次の属性を実装しなければならない。

- EUI 64 アドレス
- ショートアドレス
- 機能情報
- デバイス PANID

データリンク層は、下記の情報を利用できるようにすべきである。

- 送受信されたパケット
- 送受信されたオクテット
- 送信と受信に失敗したパケット
- 受信のセキュリティエラー
- 確認応答がないことに起因するパケット送信の失敗
- CSMA(チャネルアクセス)障害に起因するパケット送信の失敗
- MAC の再試行回数

アダプテーション層は、下記の情報を利用できるようにすべきである。

- 送受信されたパケット
- 送受信されたオクテット
- 受信時のフラグメンテーションエラー

ネットワーク層は、下記のパラメータを利用できるようにすべきである。

- IPv6 アドレスリスト: ノードの ZigBee IP のインタフェースに割り当てられた IPv6 アドレスのリスト
- RPL インスタンスリスト: ノードが属する RPL インスタンスのリスト
- RPL 送信元経路リスト: ノードで利用可能な各 RPL インスタンスに対しての RPL 送信元経路のリスト
- RPL 親リスト: ノード上の、各 RPL インスタンスに対する RPL の親の集合

管理層は、次のパラメータを利用できるようにすべきである。

- NetworkID: このノードが属する ZigBee IP ネットワークの識別子.
- MLE neighbor table: 近隣のノードのアドレスとそれに関連するリンク品質情報のリスト

6.3.12. 永続的データ

実際に動作しているデバイスは、保守担当者が手動か、またはプログラムによってリセットされてもよいし、局所的、または、ネットワーク全体の電源障害、通常の保守での電池交換、衝撃などを含むどのような理由により、誤ってリセットされてよい。リセットされるデバイスが介入せずに、ネットワークの動作を再起動する必要がある。リセットされるデバイスは、ユーザの介在なし再起動する能力を持つ必要がある。

ZIP ルータと ZIP ホストは、ネットワーク識別子を不揮発性記憶装置に格納すべきである。これにより、ノードは、予定外のリセットからユーザの介在なしに、回復することができる。加えて、ZIP ルータと ZIP ホストは、再接続をより効率的に行うために、PANA セキュリティセッション情報を不揮発性記憶装置に格納すべきである。リセット後、以前のコンフィギュレーションを復元しているノードは、再びユニークさをチェックせずにその前の GP16 の IPv6 アドレス(または MAC ショートアドレス)を再利用するべきではない。

ZIP コーディネータは、リセットの後に ZIP ネットワーク構成を復元するために必要な情報を永続的な記憶装置に格納しなければならない。これには以下のものが含まれる。

- ZIP NetworkID の値
- 認証された各ノードの PANA セキュリティセッション情報
- ネットワークキーマティリアル
- ルータの広告パケットの情報を再現するために必要な情報。これには、ABRO のバージョン、 プレフィックスとコンテキストの情報が含まれる。
- DIO パケットを再現するために必要な情報。これには、RPL のインスタンス ID と DAG のバージョンが含まれる。

データを永続化する方法は、この仕様書の範囲外である。

6.4. 定数と属性

このセクションでは、ZigBee IP プロトコルスイートに必要な定数と属性を指定する。

6.4.1. 属性

ZIP ノードは、以下の属性値を設定しなければならない。

表6-24：ZIP ノードの設定

属性	説明	値
MIN_6LP_CID_COUNT	ノードがサポートする 6LoWPAN ヘッダ圧縮コンテキスト識別子の最小数	4
MIN_6LP_PREFIX	ノードがサポートする 6LoWPAN プレフィクスの最小数	2
MIN_RPL_INSTANCE_COUNT	ZIP ルータが参加可能である RPL インスタンスの最小数	2

MLE_ADV_INTERVAL	ZIP ルータによる連続した MLE アドバタイズメントパケット送信の時間間隔	16 秒
MLE_ADV_TIMEOUT	このノードを近隣ノードとして含む近隣ノードから MLE アドバタイズメントを受け取っていない場合、ZIP ルータが MAC デバイステーブルのノードを削除すべき時間間隔	54 秒
MLE_MAX_ALLOW_JOIN_TIME	ZIP ルータが追加のコマンド無しに Allow Join flag を有効にすべき最大時間	30 分
RPL_INSTANCE_LOST_TIMEOUT	ZIP ルータがインスタンスからそれ自体を削除する前に、RPL インスタンスへの接続を失う可能性がある時間	1200 秒
RPL_MIN_DAO_PARENT	RPL ルータが対応可能であるべき DAO の親の数	2
RPL_MAX_RIO	DIO パケットに含まれるべき経路情報オプションの最大数	3
RPL_MTU_EXTENSION	RPL トンネルインタフェースで送信される IP パケットのためのリンクレイヤ MTU に追加される追加のオクテット数	100 byte
RPL_MAX_PIO	DIO パケットの中に含まれている可能性があるプレフィクス情報オプションの最大数	1
EAP_TLS_MTU	EAP-TLS フラグメントを使用する際の EAP ペイロードの TLS データの最大サイズ	512 octets
MAC_MIN_INDIRECT_TIMEOUT	データリンク層での間接送信のための IPv6 パケットを ZIP ルータがをバッファリングする最小時間	1 秒
MAC_MIN_INDIRECT_BUFFER	データリンク層での間接送信のために ZIP ルータがバッファリングできる IPv6 パケットの最小数	1
MAC_MAX_FAST_POLL_TIME	スリープホストノードの高速ポーリング状態での連続した MAC のポーリング間隔の最大期間	500 ミリ秒
MAC_ MAX_POLL_TIME	ZIP ルータが MAC デバイステーブルからエントリ削除可能後、スリープホストからの非アクティブ最大継続時間	1 日

MAC_MAX_NWK_KEYS	ノードが格納している MAC キーの数	2
MAC_MIN_DEV_TBL	ZIP ルータが MAC デバイステーブルでサポートすべきエントリの最小数	6
MCAST_MIN_TBL_SIZE	ZIP ルータに格納可能なトリクルマルチキャストシーケンス値の最小数	8

6.5. 付属情報－1

このセクションは、仕様の実装に役立つ情報を含んだ説明を網羅している。 そして明確／非明確な標準条件をはっきりさせるための説明がある。すべての標準要求はこの文書の標準セクションに含まれており、仕様はこの文書で述べられている。

6.5.1. PANA [PANA]

6.5.1.1. パケット

PANA パケットは、4 オクテットの倍数の大きさとすべきである。

6.5.1.2. AVPs

PANA AVPs は、最後の AVP である AUTH AVP を除き、任意の順序で現れる可能性がある。オクテット・ストリング AVPs(Auth、EAP-Payload、Nonce)は、パディングをフィールド長に含むことなく 4 オクテットで揃えられなければならない。他の AVPs は、自動的に揃えられる。

6.5.1.3. トランザクション

PANA パケットトランザクションは、EAP パケット転送の基準を作る。PANA トランザクションは PANA クライアント(PaC)と PANA 認証エージェント(PAA)の間で起き、PANA リレー(PRE)を介して中継することができる。中継されたセッションは基本的に同じ EAP と TLS 情報を伝えるが、PANA セッションは 3 つのエンティティの間で伝わる。

EAP 返答は PANA 応答にピギーバックすべきである。しかし、実装では EAP 返答が PAA からの PAN に伴われた PaC によって起きた別の PAR によって代わりに伝えられてもよいことを想定すべきである。

6.5.1.4. PANA 鍵生成

[PANA]および[PANA-ENC]で、PANA_AUTH_KEY と PANA_ENCR_KEY がどのように生成されるかについて明記している。このセクションは補足的なガイダンスである。

```
PANA_AUTH_KEY = prf+(MSK, "IETF PANA", |I_PAR|I_PAN|PaC_nonce|PAA_nonce|Key_ID);
PANA_ENCR_KEY = prf+(MSK, "IETF PANA Encryption Key",
|I_PAR|I_PAN|PaC_nonce|PAA_nonce|Key_ID);
```

PANA_AUTH_KEY と PANA_ENCR_KEY の長さが基になるハッシュ(すなわち 32 オクテット)と同じ場合、PRF 関数は一度だけ繰り返される必要がある。したがって、単に文字列に 0x01 を連結することによって、TLS PRF 関数が使用できる。

```
prf+(K, S) = P_hash(K, S | 0x01)
```

例えば 9 オクテット長を持つので、文字列“IETF PANA”は null で終わらない。また、24 オクテット長を持つので、文字列 “IETF PANA Encryption Key”は null で終わらない。

6.5.1.5. PANA で使われる IKEv2 prf+関数

PANA トランザクションはすべて、[IKEv2](RFC5996)の中で指定されている prf+関数を使用する。下記の中で、”|”は連結を示す。

prf+の定義:

```
prf+ (K,S) = T1 | T2 | T3 | T4 | ...
```

where:

```
T1 = prf (K, S | 0x01)
T2 = prf (K, T1 | S | 0x02)
T3 = prf (K, T2 | S | 0x03)
T4 = prf (K, T3 | S | 0x04)
...
```

要求の鍵を計算するために必要なすべてのデータが prf+から出力されるまでこれを続ける。

使用される PRF は、[IPSEC-HMAC]で記されている IPsec PRF 関数 PRF-HMAC-SHA-256 である。

HMAC キー・サイズ(セクション 2.1.1)で HMAC キー・サイズは基になるハッシュの大きさでなければならないことが記されている点に注意すること。したがって、この場合、PANA_AUTH_KEY サイズは 32 オクテット(SHA-256 からの出力)である。

また、もし出力が常に基になるハッシュ、またはそれ以下のサイズであれば、PRF+関数は一度だけしか反復されてはいけないことにも注意すること。

```
prf+(K, S) ≡ P_hash(K, S | 0x01)
```

6.5.2. TLS

6.5.2.1. TLS PSK

6.5.2.1.1. パラメータシークレット

[TLS-PSK] に記述: 「PSK が N オクテット長の場合、ユニット 16 を N 値(プレーン PSK の場合 N=0 オクテット)と連結し、2 番目のユニット 16 を N 値と PSK 自体で連結する」

Premaster Secret = 00 10 || 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 || 00 10 || CF CE CD CC CB CA C9 C8 C7 C6 C5 C4 C3 C2 C1 C0

ここで||は連結演算子である。

データとの長さの連結が TLS の可変長ベクトル<0..2^16-1>を表す点に注意すること。

6.5.2.1.2. PSK 鍵交換

The TLS PSK 鍵交換は下記の通り。オプションは示されていない。

```
Client                                          Server
------                                          ------

ClientHello                --------->
                                                ServerHello
                           <---------      ServerHelloDone
ClientKeyExchange
ChangeCipherSpec
Finished                   --------->
```

```
                                        ChangeCipherSpec
                                <--------          Finished
    Application Data              <------->    Application Data
```

6.5.2.1.3. PSK データ検証

以下の図で:

- ‘+’ は連結を意味する
- ‘[]’ はデータ発信者に対するデータ受信者を、verify_data の場合、再構築されたデータを意味する
- ‘=>’ は計算を意味する
- メッセージの連結に含まれる最後の Finished メッセージはクリアテキストとして使用される
- 検証はサーバでは SVAL、クライアントでは CVAL で行われる
- verify_data = PRF(master_secret, finished_label, Hash(handshake_messages))
- verify_data_length は 12 オクテットである
- クライアントから送られる Finished メッセージには、文字列 "client finished"が finished_label となる
- サーバから送られる Finished メッセージには、文字列 "server finished" が finished_label となる

データ検証は以下のハンドシェイクメッセージを通して行われる。

```
    Client                                              Server
    ------                                              ------

    C:ClientHello             -------->           [C:ClientHello]
    +                                                     +
    [S:ServerHello]                                S:ServerHello
    +                                                     +
    [S:ServerHelloDone]        <--------          S:ServerHelloDone
    +                                                     +
    C:ClientKeyExchange                       [C:ClientKeyExchange]
    => C:verify_data                            => [C:verify_data]
    +                                                     +
    C:Finished(C:verify_data)   -------->  [C:Finished(C:verify_data)] SVAL
    => [S:verify_data]                            =>  S:verify_data
CVAL  [S:Finished(S:verify_data)] <--------    S:Finished(S:verify_data)
```

6.5.2.2. TLS ECC

6.5.2.2.1. ECC 鍵交換

The TLS ECC 鍵交換は下記の通り。オプションは示されていない。認証は共通しているためこの暗号スイートが使われている場合、TLS サーバはクライアントに認証を要求しなければならない。すなわち、クライアント証明が必須となる。

```
    Client                                              Server
    ------                                              ------

    ClientHello                -------->
                                                  ServerHello
                                                  Certificate
                                            ServerKeyExchange
                                           CertificateRequest
                               <--------     ServerHelloDone
```

```
     Certificate
     ClientKeyExchange
     CertificateVerify
     ChangeCipherSpec
     Finished                  -------->
                                           ChangeCipherSpec
                               <--------           Finished
     Application Data            <------->    Application Data
```

6.5.2.2.2. ECC データ検証

以下の図で:

- ‘+’ は連結を意味する
- ‘[]’ はデータ発信者に対するデータ受信者を、verify_data の場合、再構築されたデータを意味する
- ‘=>’ は計算を意味する
- メッセージの連結に含まれる最後の Finished メッセージはクリアテキストとして使用される
- 検証はサーバでは SVAL、クライアントでは CVAL で行われる
- verify_data = PRF(master_secret, finished_label, Hash(handshake_messages))
- verify_data_length は 12 オクテットである
- クライアントから送られる Finished メッセージには、文字列 "client finished"が finished_label となる
- サーバから送られる Finished メッセージには、文字列 "server finished" が finished_label となる

データ検証は以下のハンドシェイクメッセージを通して行われる。

```
     Client                                              Server
     ------                                               ------

     C:ClientHello             -------->            [C:ClientHello]
     +                                                      +
     [S:ServerHello]                                    S:ServerHello
     +                                                      +
     [S:Certificate]                                    S:Certificate
     +                                                      +
     [S:ServerKeyExchange]                          S:ServerKeyExchange
     +                                                      +
     [S:CertificateRequest]                        S:CertificateRequest
     +                                                      +
     [S:ServerHelloDone]        <--------           S:ServerHelloDone
     +                                                      +
     C:Certificate                                   [C:Certificate]
     +                                                      +
     C:ClientKeyExchange                          [C:ClientKeyExchange]
     +                                                      +
     C:CertificateVerify                         [C: CertificateVerify]
     => C:verify_data                               => [C:verify_data]
     +                                                      +
     C:Finished(C:verify_data)   --------> [C:Finished(C:verify_data)] SVAL
     => [S:verify_data]                             =>  S:verify_data
CVAL  [S:Finished(S:verify_data)] <--------    S:Finished(S:verify_data)
```

6.5.2.3. TLS ECC 追加情報

6.5.2.3.1. ClientHello 拡張

ClientHello には機能拡張がある。それは compression_methods フィールドの後に存在する追加のデータとして確認される。

[TLS-ECC]セクション 5.1 からの機能拡張は下記の通り。

- elliptic_curves (10), size 4:
    - EllipticCurveList length: 2
    - One NamedCurve: secp256r1 (0x0017)
- ec_point_formats (11), size 2
    - ECPointFormatList length: 1
    - One ECPointFormat: uncompressed (0x00)

[TLS]からの機能拡張は下記の通り。

- signature_algorithms (13), size 4:
    - SignatureAndHashAlgorithm length: 2
    - hash sha256 (0x04)
    - signature ecdsa (0x03)

6.5.2.3.2. ServerHello 拡張

ServerHello には機能拡張がある。それは compression_method フィールドの後に存在する追加のデータとして確認される。

[TLS-ECC]セクション 5.2 からの機能拡張は下記の通り。

- ec_point_formats (11), size 2:
    - ECPointFormatList length: 1
    - One ECPointFormat: uncompressed (0x00)

6.5.2.4. TLS CCM パラメータ

[AEAD]で述べられているように、以下のパラメータは TLS-PSK 及び TLS-ECC 暗号スイート中の CCM AEAD 暗号のために使われる。

表6-25：TLS CCM パラメータ

パラメータ	値	説明
M	8	MIC length
L	3	Length length

6.5.3. トランザクション例

トランザクションは通常、階層化される。

- TLS レコード
- EAP パケット
- PANA パケット

PANA セッションは EAP セッションを含み、EAP セッションは TLS ハンドシェイク・トランザクションを含む。

6.5.3.1. 構文

使用される構文は C 構造構文と類似している。すべてのフィールドサイズが明確であり、パケットにフィールド値が固定されたところで、値が定められる。

6.5.3.2. TLS

TLS レコードは一般的にハンドシェイク・トランザクションで説明されている通りに連結される。 各々のレコードは、TLS ハンドシェイクと TLS Change Cipher Spec レコードのためのプレーンテキストデータと、TLS ハンドシェイクレコードのための暗号文データを含む。

6.5.3.3. EAP

EAP パケットは、EAP エンティティ(すなわち Peer と Authenticator)の間で、要求と応答を伝える。EAP プロトコルは、パケットを断片化し再び組み立てるようにする。EAP-TLS は特定の EAP 方式で、EAP プロトコルに TLS レコードをカプセル化し、鍵誘導を定義する。

6.5.3.4. PANA

PANA パケットトランザクションは、上位レイヤパケット転送の基準を作る。PANA トランザクションは PANA クライアント(PaC)と PANA 認証エージェント(PAA)の間で起き、PANA リレー(PRE)を介して中継することができる。

PaC と PAA の PANA セッションは下記のとおりである。中継されたセッションは基本的に同じ EAP と TLS 情報を伝えるが、PANA セッションは 3 つのエンティティの間である。

下のシークエンスは EAP 返答が PANA 応答にピギーバックされることを想定している。いつもこうであることもなく、実装では EAP 返答が PAA からの PAN に伴われた PaC によって起きた別の PAR によって代わりに伝えられてもよいことを想定すべきである。

PANA パケットは、4 オクテットの倍数の大きさであるべきである。

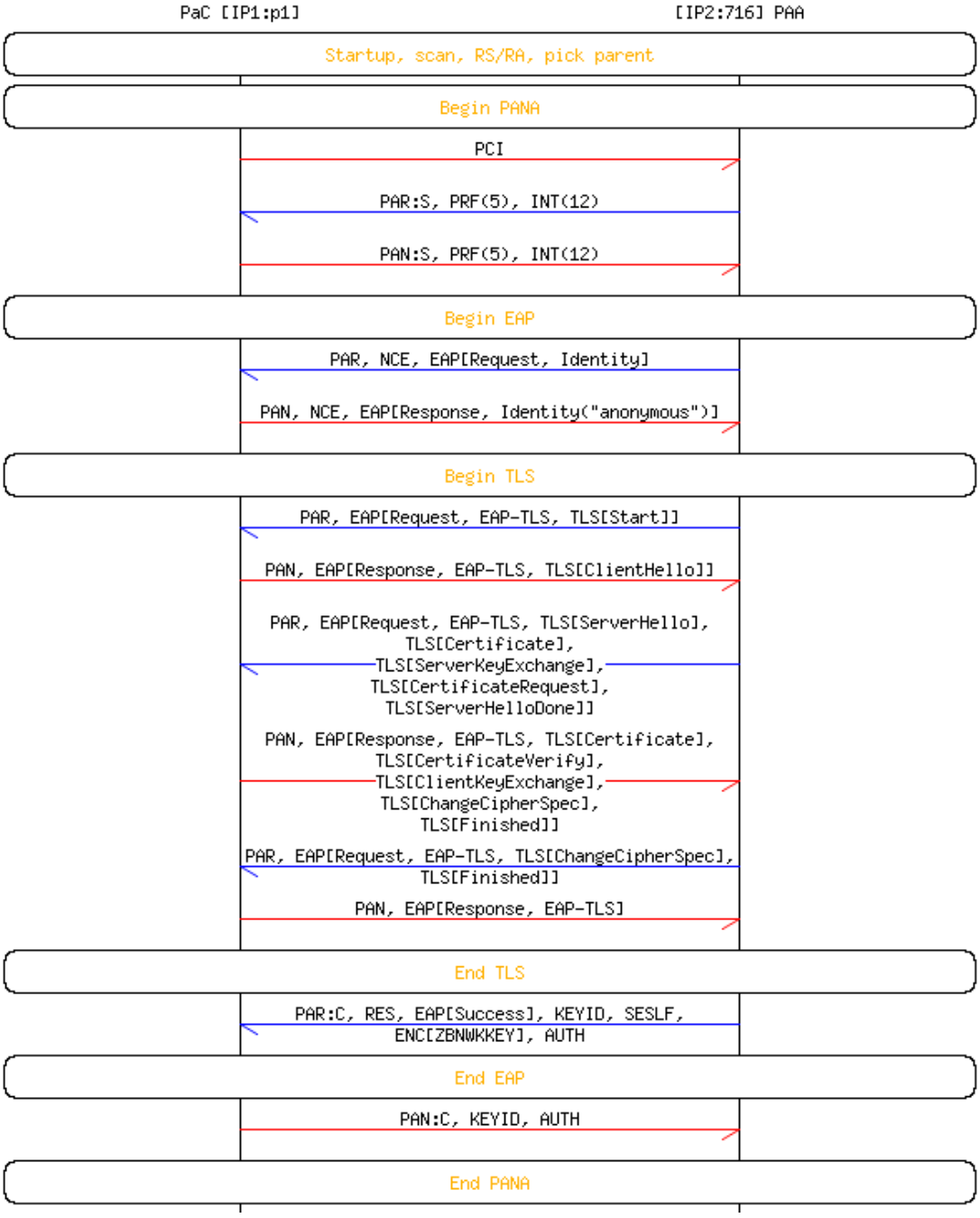


図6-12：ECC PANA 交換

6.5.3.5. PCI (PaC から PAA へ)

```
struct PANA {
    uint16 rsvd = 0;
    uint16 length = 16; /* 16H */
    uint16 flags = 0x0000;
    uint16 type = 1; /* PCI */
    uint32 session_id = 0;
    uint32 seq_no = 0;
};
```

6.5.3.6. PANA 開始 (PAA から PaC へ)

```
struct PANA {
   uint16 rsvd = 0;
   uint16 length = 52; /* 16H + (8H + 4P) + (8H + 4P) + (8H + 4P) */
   uint16 flags = 0xC000; /* Request, start */
   uint16 type = 2; /* PA */
   uint32 session_id = paa_session_id; /* Chosen by PAA */
   uint32 seq_no = paa_seq_no; /* Random number chosen by PAA */
   /* If PRF_HMAC_SHA2_256 is the only PRF, the following AVP may be optional */
   struct PANAAVP {
      uint16 code = 6; /* PRF algorithm */
      uint16 flags = 0;
      uint16 length = 4;
      uint16 rsvd = 0;
      uint32 prf_algorithm = 5;
   }
   /* If AUTH_HMAC_SHA2_256_128 is the only integrity algorithm, the following AVP
may be optional */
   struct PANAAVP {
      uint16 code = 3; /* Integrity algorithm */
      uint16 flags = 0;
      uint16 length = 4;
      uint16 rsvd = 0;
      uint32 integrity_algorithm = 12;
   }
   /* If AES-CTR is the only encryption, the following AVP may be optional */
   struct PANAAVP {
      uint16 code = 12; /* Encryption algorithm */
      uint16 flags = 0;
      uint16 length = 4;
      uint16 rsvd = 0;
      uint32 encryption_algorithm = 1;
   }
};
```

6.5.3.7. PANA 開始 (PaC から PAA へ)

```
struct PANA {
   uint16 rsvd = 0;
   uint16 length = 52; /* 16H + (8H + 4P) + (8H + 4P) + (8H + 4P) */
   uint16 flags = 0x4000; /* Answer, Start */
   uint16 type = 2; /* PA */
   uint32 session_id = paa_session_id; /* Returned by PaC */
   uint32 seq_no = paa_seq_no; /* Returned by PaC */
   /* If PRF_HMAC_SHA2_256 is the only PRF, the following AVP may be optional */
   struct PANAAVP {
      uint16 code = 6; /* PRF algorithm */
      uint16 flags = 0;
      uint16 length = 4;
      uint16 rsvd = 0;
      uint32 prf_algorithm = 5;
   }
   /* If AUTH_HMAC_SHA2_256_128 is the only integrity algorithm, the following AVP
may be optional */
   struct PANAAVP {
      uint16 code = 3; /* Integrity algorithm */
      uint16 flags = 0;
      uint16 length = 4;
      uint16 rsvd = 0;
      uint32 integrity_algorithm = 12;
   }
   /* If AES-CTR is the only encryption, the following AVP may be optional */
```

```
    struct PANAAVP {
        uint16 code = 12; /* Encryption algorithm */
        uint16 flags = 0;
        uint16 length = 4;
        uint16 rsvd = 0;
        uint32 encryption_algorithm = 1;
    }
};
```

6.5.3.8. EAP 識別子要求 (PAA から PaC へ)

```
struct PANA {
    uint16 rsvd = 0;
    uint16 length = 56; /* 16 + (8H + 16P) + (8H + 5P + 3Pd) */
    uint16 flags = 0x8000; /* Request */
    uint16 type = 2; /* PA */
    uint32 session_id = paa_session_id;
    uint32 seq_no = paa_seq_no + 1; /* Increment sequence number */
    struct PANAAVP {
        uint16 code = 5; /* Nonce */
        uint16 flags = 0;
        uint16 length = 16;
        uint16 rsvd = 0;
        uint8 nonce[16];
    }
    /* The following AVP may be optional */
    struct PANAAVP {
        uint16 code = 2; /* EAP Payload */
        uint16 flags = 0;
        uint16 length = 5; /* 5P */
        uint16 rsvd = 0;
        struct EAPReqUnfrag {
            uint8 code = 1; /* EAPReq */
            uint8 identifier = idseq;
            uint16 length = 5; /* inc. 5H + 0P */
            uint8 type = 1; /* EAP-Identity */
        };
        struct AVPPad {
        uint8 bytes[3];
     };
    };
};
```

6.5.3.9. EAP 識別子返答 (PaC からへ)

```
struct PANA {
    uint16 rsvd = 0;
    uint16 length = 64; /* 16H + (8H + 16P) + (8H + 14P + 2Pd) */
    uint16 flags = 0x0000; /* Answer */
    uint16 type = 2; /* PA */
    uint32 session_id = paa_session_id; /* Returned by PaC */
    uint32 seq_no = paa_seq_no + 1; /* Returned by PaC */
    struct PANAAVP {
        uint16 code = 5; /* Nonce */
        uint16 flags = 0;
        uint16 length = 16;
        uint16 rsvd = 0;
        uint8 nonce[16];
    }
    /* The following AVP may be optional */
    struct PANAAVP {
        uint16 code = 2; /* EAP Payload */
        uint16 flags = 0;
        uint16 length = 14;
```

```
        uint16 rsvd = 0;
        struct EAPRspUnfrag {
            uint8 code = 2; /* EAPRsp */
            uint8 identifier = idseq; /* Corresponds to request */
            uint16 length = 14; /* inc. 5H + 9P */
            uint8 type = 1; /* EAP-Identity */
            /* Anonymous NAI */
            uint8 identity[] = "anonymous";
        };
        struct AVPPad {
        uint8 bytes[2];
     };
    };
};
```

6.5.3.10. TLS 開始 (PAA から PaC へ)

```
struct PANA {
    uint16 rsvd = 0;
    uint16 length = 32; /* 16H + (8H + 6P + 2Pd) */
    uint16 flags = 0x8000; /* Request */
    uint16 type = 2; /* PA */
    uint32 session_id = paa_session_id;
    uint32 seq_no = paa_seq_no + 2; /* Increment sequence number */
    struct PANAAVP {
        uint16 code = 2; /* EAP Payload */
        uint16 flags = 0;
        uint16 length = 6;
        uint16 rsvd = 0;
        struct EAPReqUnfrag {
            uint8 code = 1;
            uint8 identifier = idseq + 1;
            uint16 length = 6; /* inc. 6H + 0P */
            uint8 type = 13; /* EAP-TLS */
            uint8 flags = 0x20; /* Start */
        };
        struct AVPPad {
        uint8 bytes[2];
     };
    };
};
```

6.5.3.11. PSK TLS ClientHello (PaC から PAA へ)

```
struct PANA {
    uint16 rsvd = 0;
    uint16 length = 80; /* 16H + (8H + 56P) */
    uint16 flags = 0x0000; /* Answer */
    uint16 type = 2; /* PA */
    uint32 session_id = paa_session_id; /* Returned by PaC */
    uint32 seq_no = paa_seq_no + 2; /* Returned by PaC */
    struct PANAAVP {
        uint16 code = 2; /* EAP Payload */
        uint16 flags = 0;
        uint16 length = 56;
        uint16 rsvd = 0;
        struct EAPRspUnfrag {
            uint8 code = 2;
            uint8 identifier = idseq + 1; /* Corresponds to request */
            uint16 length = 56; /* inc. 6H + (5H + 45P) */
            uint8 type = 13; /* EAP-TLS */
            uint8 flags = 0x00;
            struct TLSPlaintext {
                uint8 type = 22; /* Handshake */
```

```
            uint8 version[2] = {0x03, 0x03}; /* TLS 1.2 */
            uint16 length = 45; /* 4H + 41P */
            struct Handshake {
               uint8 msg_type = 1; /* ClientHello */
               uint24 length = 41; /* 2P + 32P + 1P + 4P + 2P */
               struct ClientHello {
                  struct ProtocolVersion {
                     uint8 major = 0x03;
                     uint8 minor = 0x03; /* TLS 1.2? */
                  } client_version;
                  struct Random {
                     uint32 gmt_unix_time;
                     uint8 random_bytes[28];
                  } random;
                  struct SessionID<0..32> {
                     uint8 length = 0; /* NULL */
                  } session_id;
                  struct <2..2^16-2> {
                     uint16 length = 2;
                     struct CipherSuite {
                        uint8 bytes[2] = {0x00, 0xC6};
                     } cipher_suites[1];
                  };
                  struct <1..2^8-2> {
                     uint8 length = 1;
                     uint8 compression_methods[1] = {0};
                  }
                  /* NOTE: extensions will be needed for public key cipher suite
*/
                  struct { }; /* No extensions */
               };
            };
         };
      };
   };
};
```

6.5.3.12. ECC TLS ClientHello (PaC から PAA へ)

```
struct PANA {
   uint16 rsvd = 0;
   uint16 length = 108; /* 16H + (8H + 82P + 2Pd) */
   uint16 flags = 0x0000; /* Answer */
   uint16 type = 2; /* PA */
   uint32 session_id = paa_session_id; /* Returned by PaC */
   uint32 seq_no = paa_seq_no + 2; /* Returned by PaC */
   struct PANAAVP {
      uint16 code = 2; /* EAP Payload */
      uint16 flags = 0;
      uint16 length = 82;
      uint16 rsvd = 0;
      struct EAPRspUnfrag {
         uint8 code = 2;
         uint8 identifier = idseq + 1; /* Corresponds to request */
         uint16 length = 82; /* inc. 6H + (5H + 77P) */
         uint8 type = 13; /* EAP-TLS */
         uint8 flags = 0x00;
         struct TLSPlaintext {
            uint8 type = 22; /* Handshake */
            uint8 version[2] = {0x03, 0x03}; /* TLS 1.2 */
            uint16 length = 71; /* 4H + 67P */
            struct Handshake {
               uint8 msg_type = 1; /* ClientHello */
               uint24 length = 67; /* 2P + 32P + 1P + 8P + 2P + 22P */
```

```
                struct ClientHello {
                    struct ProtocolVersion {
                        uint8 major = 0x03;
                        uint8 minor = 0x03; /* TLS 1.2? */
                    } client_version;
                    struct Random {
                        uint32 gmt_unix_time;
                        uint8 random_bytes[28];
                    } random;
                    struct SessionID<0..32> {
                        uint8 length = 0; /* NULL */
                    } session_id;
                    struct <2..2^16-2> {
                        uint16 length = 4;
                        struct CipherSuite {
                            uint8 bytes[2] = {0xC0, 0xC6};
                        } cipher_suites[1];
                        struct CipherSuite {
                            uint8 bytes[2] = {0x00, 0xC6};
                        } cipher_suites[1];
                    };
                    struct <1..2^8-2> {
                        uint8 length = 1;
                        uint8 compression_methods[1] = {0};
                    }
                    struct { /* ECC extensions */
                        uint16 length = 22;
                        struct EllipticCurvesExtension {
                            uint16 type = 10; /* elliptic_curves */
                            uint16 length = 4;
                            uint16 eclength = 2;
                            uint16 ec = 23; /* secp256r1 */
                        };
                        struct ECPointFormatsExtension {
                            uint16 type = 11; /* ec_point_formats */
                            uint16 length = 2;
                            uint8 pflength = 1;
                            uint8 pf = 0; /* uncompressed */
                        };
                        struct SignatureAlgorithmsExtension {
                            uint16 type = 13; /* signature_algorithms */
                            uint16 length = 4; /* 2? */
                            struct <2..2^16-2> {
                                uint16 length = 2;
                                struct SignatureAndHashAlgorithm {
                                    uint8 hash = 0x04; /* sha256 */
                                    uint8 signature = 0x03; /* ecdsa */
                                } signature_and_hash_algorithm[1];
                            };
                        };
            };
                 };
              };
           };
        };
        struct AVPPad {
        uint8 bytes[2];
     };
    };
};
```

6.5.3.13. PSK TLS ServerHello, ServerHelloDone (PAA から PaC へ)

```
struct PANA {
```

```
   uint16 rsvd = 0;
   uint16 length = 88; /* 16H + (8H + 61P + 3Pd) */
   uint16 flags = 0x8000; /* Request */
   uint16 type = 2; /* PA */
   uint32 session_id = paa_session_id;
   uint32 seq_no = paa_seq_no + 3; /* Increment sequence number */
   struct PANAAVP {
      uint16 code = 2; /* EAP Payload */
      uint16 flags = 0;
      uint16 length = 61;
      uint16 rsvd = 0;
      struct EAPReqUnfrag {
         uint8 code = 1;
         uint8 identifier = idseq + 2;
         uint16 length = 61; /* inc. 6H + (5H + 50P) */
         uint8 type = 13; /* EAP-TLS */
         uint8 flags = 0x00;
         struct TLSPlaintext {
            uint8 type = 22; /* Handshake */
            uint8 version[2] = {0x03, 0x03}; /* TLS 1.2 */
            uint16 length = 50; /* (4H + 42P) + (4H + 0P) */
            struct Handshake {
               uint8 msg_type = 2; /* ServerHello */
               uint24 length = 42; /* 2P + 32P + 5P + 2P + 1P */
               struct ServerHello {
                  struct ProtocolVersion {
                     uint8 major = 0x03;
                     uint8 minor = 0x03; /* TLS 1.2? */
                  } server_version;
                  struct Random {
                     uint32 gmt_unix_time;
                     uint8 random_bytes[28];
                  } random;
                  struct SessionID<0..32> {
                     uint8 length = 4; /* Arbitrary for now */
                     uint8 bytes[4];
                  } session_id;
                  struct CipherSuite {
                     uint8 bytes[2] = {0x00, 0xC6};
                  } cipher_suite;
                  uint8 compression_method = {0};
                  /* NOTE: extensions will be needed for public key cipher suite
*/
                  struct { }; /* No extensions */
               };
            };
            struct Handshake {
               uint8 msg_type = 14; /* ServerHelloDone */
               uint24 length = 0;
               struct ServerHelloDone { }; /* Empty */
            };
         };
      };
      struct AVPPad {
      uint8 bytes[3];
      };
   };
};
```

6.5.3.14. ECC TLS ServerHello, Certificate, ServerKeyExchange, CertificateRequest, ServerHelloDone (PAA から PaC へ)

```
struct PANA {
   uint16 rsvd = 0;
```

```
uint16 length = 844; /* 16H + (8H + 61P + 3Pd) */
uint16 flags = 0x8000; /* Request */
uint16 type = 2; /* PA */
uint32 session_id = paa_session_id;
uint32 seq_no = paa_seq_no + 3; /* Increment sequence number */
struct PANAAVP {
   uint16 code = 2; /* EAP Payload */
   uint16 flags = 0;
   uint16 length = 820;
   uint16 rsvd = 0;
   struct EAPReqUnfrag {
      uint8 code = 1;
      uint8 identifier = idseq + 2;
      uint16 length = 820; /* inc. 6H + (5H + 50P) */
      uint8 type = 13; /* EAP-TLS */
      uint8 flags = 0x00;
      struct TLSPlaintext {
         uint8 type = 22; /* Handshake */
         uint8 version[2] = {0x03, 0x03}; /* TLS 1.2 */
         uint16 length = 50; /* (4H + 42P) + (4H + 0P) */
         struct Handshake {
            uint8 msg_type = 2; /* ServerHello */
            uint24 length = 78; /* 2P + 32P + 5P + 2P + 1P */
            struct ServerHello {
               struct ProtocolVersion {
                  uint8 major = 0x03;
                  uint8 minor = 0x03; /* TLS 1.2? */
               } server_version;
               struct Random {
                  uint32 gmt_unix_time;
                  uint8 random_bytes[28];
               } random;
               struct SessionID<0..32> {
                  uint8 length = 32; /* Arbitrary for now */
                  uint8 bytes[32];
               } session_id;
               struct CipherSuite {
                  uint8 bytes[2] = {0xC0, 0xC6};
               } cipher_suite;
               uint8 compression_method = {0};
               struct { /* ECC extensions */
                  uint16 length = 6;
                  struct ECPointFormatsExtension {
                     uint16 type = 11; /* ec_point_formats */
                     uint16 length = 2;
                     uint8 pflength = 1;
                     uint8 pf = 0; /* uncompressed */
                  };
               };
            };
         };
         struct Handshake {
            uint8 msg_type = 11; /* Certificate */
            uint24 length = 559;
            uint24 certificates_length = 556;
            uint24 certificate length = 553;
            uint8 certificate[0][553]; /* Single certificate */
         };
         struct Handshake {
            uint8 msg_type = 12; /* ServerKeyExchange */
            uint24 length = 144;
            uint8 server_key_exchange[144]; /* Single certificate */
            struct ServerHelloDone { }; /* Empty */
         };
```

```
            struct Handshake {
               uint8 msg_type = 13; /* CertificateRequest */
               uint24 length = 10;
               struct <2..2^8-1> {
                  uint8 length = 1;
                  uint8 certificate_types = 0x40; /* ecdsa_sign */
               };
               struct <2..2^16-2> {
                  uint16 length = 2;
                  struct SignatureAndHashAlgorithm {
                     uint8 hash = 0x04; /* sha256 */
                     uint8 signature = 0x03; /* ecdsa */
                  } signature_and_hash_algorithm[1];
               };
               struct <2..2^16-1> {
                  uint16 length = 0;
               };
            };
            struct Handshake {
               uint8 msg_type = 14; /* ServerHelloDone */
               uint24 length = 0;
               struct ServerHelloDone { }; /* Empty */
            };
         };
      };
      struct AVPPad {
      uint8 bytes[3];
      };
   };
};
```

6.5.3.15. TLS ClientKeyExchange and ChangeCipherSpec, Finished (PaC から PAA へ)

```
struct PANA {
   uint16 rsvd = 0;
   uint16 length = 88; /* 16H + (8H + 62P + 2Pd) */
   uint16 flags = 0x0000; /* Answer */
   uint16 type = 2; /* PA */
   uint32 session_id = paa_session_id; /* Returned by PaC */
   uint32 seq_no = paa_seq_no + 3; /* Returned by PaC */
   struct PANAAVP {
      uint16 code = 2; /* EAP Payload */
      uint16 flags = 0;
      uint16 length = 62;
      uint16 rsvd = 0;
      struct EAPRspUnfrag {
         uint8 code = 2;
         uint8 identifier = idseq + 2; /* Corresponds to request */
         uint16 length = 62; /* inc. 6H + (5H + (4H + 4P)) + (5H + 1P) + (5H + 32P)
*/
         uint8 type = 13; /* EAP-TLS */
         uint8 flags = 0x00;
         struct TLSPlaintext{
            uint8 type = 22; /* Handshake */
            uint8 version[2] = {0x03, 0x03}; /* TLS 1.2 */
            uint16 length = 4;
            struct Handshake {
               uint8 msg_type = 16; /* ClientKeyExchange */
               uint24 length = 4;
               struct ClientKeyExchange {
                  struct <0..2^16-1> {
                     uint16 length = 2;
                     uint8 bytes[1] = {0x30, 0x00};
                  } psk_identity;
```

```
                } ;
            } ;
        } ;
        struct TLSPlaintext{
            uint8 type = 20; /* ChangeCipherSpec */
            uint8 version[2] = {0x03, 0x03}; /* TLS 1.2 */
            uint16 length = 1;
            struct ChangeCipherSpec{
                uint8 type = 1;    /* ChangeCipherSpec */
            };
        };
        struct TLSCiphertext {
            uint8 type = 22; /* Handshake */
            uint8 version[2] = {0x03, 0x03}; /* TLS 1.2 */
            uint16 length = 32;
            struct GenericAEADCipher {
                struct CCMNonceExplicit {
                    uint64 seq_num;
                };
                struct CCMCipherText { /* inferred from draft-mcgrew-tls-aes-ccm
*/
                    struct Handshake { /* Encrypted */
                        uint8 msg_type = 20; /* Finished */
                        uint24 length = 12;
                        struct Finished {
                            uint8 verify_data[12];
                        ;}
                    };
                    uint8 MAC[8]; /* Using AES_CCM_8 */
                };
            };
        };
    };
    struct AVPPad {
    uint8 bytes[2];
 };
};
};
```

6.5.3.16. TLS ChangeCipherSpec, TLS Finished (PAA から PaC へ)

```
struct PANA {
    uint16 rsvd = 0;
    uint16 length = 134; /* 16H + (8H + 49P + 0Pd) */
    uint16 flags = 0x8000; /* Request */
    uint16 type = 2; /* PA */
    uint32 session_id = paa_session_id;
    uint32 seq_no = paa_seq_no + 4; /* Increment sequence number */
    struct PANAAVP {
        uint16 code = 2; /* EAP Payload */
        uint16 flags = 0;
        uint16 length = 49;
        uint16 rsvd = 0;
        struct EAPReqUnfrag {
            uint8 code = 1;
            uint8 identifier = idseq + 3;
            uint16 length = 49; /* inc. 6H + (5H + 1P) + (5H + 32P) */
            uint8 type = 13; /* EAP-TLS */
            uint8 flags = 0x00;
            struct TLSPlaintext{
                uint8 type = 20; /* ChangeCipherSpec */
                uint8 version[2] = {0x03, 0x03}; /* TLS 1.2 */
                uint16 length = 1;
                struct ChangeCipherSpec{
```

```
                uint8 type = 1;    /* ChangeCipherSpec */
            };
         };
         struct TLSCiphertext {
            uint8 type = 22; /* Handshake */
            uint8 version[2] = {0x03, 0x03}; /* TLS 1.2 */
            uint16 length = 32;
            struct GenericAEADCipher {
               struct CCMNonceExplicit {
                  uint64 seq_num;
               };
               struct CCMCipherText { /* inferred from draft-mcgrew-tls-aes-ccm
*/
                  struct Handshake { /* Encrypted */
                     uint8 msg_type = 20; /* Finished */
                     uint24 length = 12;
                     struct Finished {
                        uint8 verify_data[12];
                     };
                  };
                  uint8 MAC[8]; /* Using AES_CCM_8 */
               };
            };
         };
      };
   };
};
```

6.5.3.17. Final EAP 返答 (PaC から PAA へ)

```
struct PANA {
   uint16 rsvd = 0;
   uint16 length = 30; /* 16H + (8H + 6P + 2Pd) */
   uint16 flags = 0x0000; /* Answer */
   uint16 type = 2; /* PA */
   uint32 session_id = paa_session_id; /* Returned by PaC */
   uint32 seq_no = paa_seq_no + 4; /* Returned by PaC */
   struct PANAAVP {
      uint16 code = 2; /* EAP Payload */
      uint16 flags = 0;
      uint16 length = 6;
      uint16 rsvd = 0;
      struct EAPRspUnfrag {
         uint8 code = 2;
         uint8 identifier = idseq + 3; /* Corresponds to request */
         uint16 length = 6; /* inc. 6H + 0P */
         uint8 type = 13; /* EAP-TLS */
         uint8 flags = 0x00;
      };
      struct AVPPad {
      uint8 bytes[2];
    };
   };
};
```

6.5.3.18. PANA Complete, EAP Success (PAA から PaC へ)

```
struct PANA {
   uint16 rsvd = 0;
   uint16 length = 128; /* 16H + (8H + 4P) + (8H + 4P) + (8H + 4P) + (8H + 4P) +
(8H + (12H + 18P + 2Pd) + (8H + 16P) */
   uint16 flags = 0xA000; /* Request, Complete */
   uint16 type = 2; /* PA */
   uint32 session_id = paa_session_id;
```

```
    uint32 seq_no = paa_seq_no + 5; /* Increment sequence number */
    struct PANAAVP {
       uint16 code = 7; /* Result code */
       uint16 flags = 0;
       uint16 length = 4;
       uint16 rsvd = 0;
       uint32 result_code = 0; /* PANA_SUCCESS */
    };
    struct PANAAVP {
       uint16 code = 2; /* EAP Payload */
       uint16 flags = 0;
       uint16 length = 4;
       uint16 rsvd = 0;
       struct EAPSuccess {
          uint8 code = 3;
          uint8 identifier = idseq + 4;
          uint16 length = 4; /* inc. 4H + 0P */
       };
    };
    struct PANAAVP {
       uint16 code = 4; /* Key ID */
       uint16 flags = 0;
       uint16 length = 4;
       uint16 rsvd = 0;
       uint32 key_id = 0; /* Initial MSK */
    };
    struct PANAAVP {
       uint16 code = 8; /* Session Lifetime */
       uint16 flags = 0;
       uint16 length = 4;
       uint16 rsvd = 0;
       uint32 sess_life = 0xFFFFFFFF; /* -1 = forever (136 years) */
    };
    struct PANAAVP {
       uint16 code = 13; /* Encrypted Encapsulation */
       uint16 flags = 0;
       uint16 length = 32;
       uint16 rsvd = 0;
       struct PANAAVP {
         uint16 code = 1; /* ZigBee Network Key */
         uint16 flags = 1; /* Vendor specific */
         uint16 length = 18;
         uint16 rsvd = 0;
         uint32 vendor_id = 37244; /* ZigBee Vendor ID */
       struct ZBNWKKEY {
          uint8 nwk_key[16];
          uint8 nwk_key_idx;
          uint8 auth_cntr;
       };
         struct AVPPad {
         uint8 bytes[2];
      };
     };
    };
    struct PANAAVP {
       uint16 code = 1; /* Auth */
       uint16 flags = 0;
       uint16 length = 16;
       uint16 rsvd = 0;
       uint8 auth[16]; /* Hash */
    };
};
```

6.5.3.19. PANA Complete (PaC から PAA へ)

```
struct PANA {
   uint16 rsvd = 0;
   uint16 length = 54; /* 16H + (8H + 4P) + (8H + 16P) */
   uint16 flags = 0x2000; /* Answer, Complete */
   uint16 type = 2; /* PA */
   uint32 session_id = paa_session_id; /* Returned by PaC */
   uint32 seq_no = paa_seq_no + 5; /* Returned by PaC */
   struct PANAAVP {
      uint16 code = 4; /* Key ID */
      uint16 flags = 0;
      uint16 length = 4;
      uint16 rsvd = 0;
      uint32 key_id = 0; /* Initial MSK */
   };
   struct PANAAVP {
      uint16 code = 1; /* Auth */
      uint16 flags = 0;
      uint16 length = 16;
      uint16 rsvd = 0;
      uint8 auth[16]; /* Hash */
   };
};
```

6.6. 付属情報－2

この章では 920MHz PHY 向けの実装に必要な各レイヤに関する変更点について解説する。本文にも記載されているパラメータ値などは本章の値をもって上書きする。

6.6.1. 物理層

920MHz PHY 向けには IEEE802.15.4g-2012[802.15.4]で規定される変調方式 FSK 及び通信レート 100kbit/s を必須とし、その他をオプション扱いとする。また、プリアンブル長はダイバーシティアンテナによる受信を考慮し 12 bytes 以上を推奨とする。

PSDU サイズは、254bites を上限とする。IEEE802.15.4g-2012[802.15.4]では、ＣＳＭを利用した送受信が必須となっているが、ここでは CSM はオプションとする。IEEE802.15.4g-2012[802.15.4]では、MR-FSK のデータホワイトニングはオプションとなっているが、ここでは必須とする。

6.6.2. データリンク層

PSDU サイズが 254bites を上限とすることから、2 octet FCS の利用を必須とし、4 octet FCS の利用をオプション扱いとする。

IEEE802.15.4g-2012[802.15.4]では、1%デューティを超える場合にはマルチ PHY 制御（MPM）を用いることが必須となっているが、ここでは MPM はオプションとする。

6.6.3. ネットワーク層

6.6.3.1. マルチキャスト

マルチキャストアドレスは[EL]では FF02:0:0:0:0:0:0:1 を使用するとしているが、下記のアドレスで適時置き換えることとする。

ユニキャストアドレス

FF02:0:0:0:0:0:0:1

FF03:0:0:0:0:0:0:1

FF05:0:0:0:0:0:0:1

FF03:0:0:0:0:0:0:2

FF05:0:0:0:0:0:0:2

[EL]において、プロパティ値通知サービスのようにマルチキャストでの応答が必須である仕様に対して、マルチキャスト送信にて要求が行われた場合、端末数が増加すると通信時のトラフィックが高くなる。

また、端末をホップして通信されることが前提とされる ZigBee IP の場合、より広範囲のスコープへマルチキャスト送信を行いたい場合が想定される。

そのため、マルチキャスト送信先アドレスのスコープを適切に設定することが望ましい。

6.6.3.2. RPL 属性

RPL Instance の最小値 (MIN?RPL_INSTANCE_COUNT) は 1 とする。

6.6.3.3. トランスポート

ECHONET Lite [EL]のアプリケーションデータ通信には、UDP パケットによる送受信を必須とし、TCP はオプションとする。UDP フレームの送信先ポート番号は、[EL]の記載に基づき常に 3610 とする。

6.6.4. アプリケーション層

アプリケーション層としては、ECHONET Lite [EL]を使用する。本方式記載の仕様に基づくノードは、[EL]に規定される必須機能をすべてサポートしなければならない。

また、[EL]は、次のサービスを提供する。

- ネットワーク内の他ノードが保有する機能単位
  (ECHONET オブジェクト)の検出
- 他ノードが有する各種パラメータ・状態
  (ECHONET プロパティ)の取得
- 他ノードの各種パラメータ・状態の設定
- 自ノードが有する各種パラメータ・状態の通知

6.7. 付属情報－3

この章ではサポートすべき IEEE802.15.4/4g の機能を明記する。

「IEEE 規定」欄は IEEE802.15.4/4g における規定を表記しており、M は必須機能、O はオプション機能を表す。「方式 B 使用」欄は、方式 B として使用するか使用しないかを表記しており、Y は必要、N は不要、O はオプションを表す。なお、O.x の表記の場合は、同一の表記の中から一つだけ使用することを表す。

6.7.1. デバイス規定

以下、IEEE802.15.4/4e/4g デバイスに関する使用規定を示す。

表6-26：Functional device types

番号	内容	参考文献の対応節	IEEE 規定	方式 B 使用
FD1	親機	[802.15.4] 5.1	O.1	O.1
FD2	子機	[802.15.4] 5.1	O.1	O.1

FD3	64bit アドレスのサポート	[802.15.4] 5.2.1.1.6	M	Y
FD4	ショートアドレスの割当機能	[802.15.4] 5.1.3.1	FD1 : M	FD1: Y
FD5	ショートアドレスのサポート	[802.15.4] 5.2.1.1.6	M	Y
FD8	15.4g 対応デバイス	[802.15.4g] 8.1	O.3	Y

6.7.2. 物理層規定

以下、物理層に関する使用規定を示す。

表6-27：PHY functions & PHY Packet

番号	内容	参考文献の対応節	IEEE 規定	方式 B 使用
PLF1	電波検出機能	[802.15.4] 8.2.5	FD1 : M	FD1: Y
PLF2	リンク品質表示機能	[802.15.4] 8.2.6	M	Y
PLF3	チャネル選択機能	[802.15.4] 8.1.2	M	Y
PLF4	空チャネル判定機能	[802.15.4] 8.2.7	M	Y
PLF4.1	CCA を電界強度で判定	[802.15.4] 8.2.7	O.2	Y
PLF4.2	CCA をキャリアセンスで判定	[802.15.4] 8.2.7	O.2	N
PLF4.3	1,2 の共用	[802.15.4] 8.2.7	O.2	N
PLP1	PSDU サイズ	[802.15.4g] 9.2	FD8 : M	255byte 程度までを推奨

表6-28：Radio frequency (RF)

番号	内容	参考文献の対応節	IEEE 規定	方式 B 使用
RF12	SUN PHYs			
RF12.1	MR-FSK	[802.15.4g] 16.1	FD8 : M	Y
RF12.2	MR-OFDM	[802.15.4g] 16.2	FD8 : O	N
RF12.3	MR-O-QPSK	[802.15.4g] 16.3	FD8 : O	N
RF12.4	MR-FSK-Generic PHY	[802.15.4g] 8.1.2.7.2	RF12.1 : O	N
RF12.5	共用信号による拡張ビーコンの送受信	[802.15.4g] 8.1a	FD1, 8, MLF15 : M	N
RF12.6	使用周波数の選択	[802.15.4g] 8.1	FD8 : M	920MHz
RF13	SUN PHY operating modes			
RF13.4	920MHz 使用時に 50kbit/s、100kbit/s をサポート	[802.15.4g] 16.1	FD8 : M	100kbit/s を使用
RF13.5	920MHz 使用時に 200kbit/s、400kbit/s をサポート	[802.15.4g] 16.1	FD8 : O	N
RF14	MR-FSK options	[802.15.4g]		
RF14.1	誤り訂正機能	[802.15.4g] 16.1.2.4	O	N
RF14.2	インタリーブ機能	[802.15.4g] 16.1.2.5	O	N
RF14.3	データホワイトニング機能 (スクランブル)	[802.15.4g] 16.1.3	O	Y
RF14.4	パケット単位で伝送速度を変える機能	[802.15.4g] 16.1.4	O	N

表6-29：PHY

項目	方式 B 使用	備考
変調方式	GFSK	
伝送速度	100kbit/s	
送信出力	20mW 以下	
周波数チャネル	ARIB 規定 33～60 チャネルの(奇数＋偶数)チャネル	33～38 チャネルは送信出力 250mW のシステムも利用する
周波数チャネル幅	400kHz(2ch 束ね)	
送信プリアンブル長	12byte 以上	

6.7.3. データリンク層規定

以下、データリンク層に関する使用規定を示す。

表6-30：MAC sublayer functions -1

番号	内容	参考文献の対応節	IEEE 規定	方式 B 使用
MLF1	データ送信	[802.15.4] 6.3	M	Y
MLF1.1	データ送信の途中破棄	[802.15.4] 6.3.4, 6.3.5	FD1 : M FD2 : O	FD1: M FD2: O
MLF2	データ受信	[802.15.4] 6.3	M	Y
MLF2.1	受信処理制御	[802.15.4] 5.1.6.5	FD1 : M FD2 : O	N
MLF2.2	PHY レシーバー制御	[802.15.4] 6.2.9	O	N
MLF2.3	タイムスタンプ機能	[802.15.4] 6.3.2	O	N
MLF3	ビーコン管理	[802.15.4] Clause 5	M	Y
MLF3.1	ビーコン送信	[802.15.4] Clause 5, 5.1.2.4	FD1 : M FD2 : O	FD1: Y FD2: O
MLF3.2	ビーコン受信	[802.15.4] Clause 5, 6.2.4	M	Y
MLF4	チャネルアクセス	[802.15.4] Clause 5, 5.1.1	M	Y
MLF5	帯域保証タイムスロット管理	[802.15.4] Clause 5, 6.2.6,	O	N
MLF5.1	帯域保証タイムスロット管理	[802.15.4] Clause 5, 6.2.6,	O	N
MLF5.2	帯域保証タイムスロット管理	[802.15.4] Clause 5, 6.2.6,	O	N
MLF6	フレーム検証	[802.15.4] 6.3.3, 5.2, 5.1.6.2	M	Y

表6-31：MAC sublayer functions -2

番号	内容	参考文献の対応節	IEEE 規定	方式 B 使用
MLF7	ACK 送信	[802.15.4] Clause 5, 6.3.3, 5.2.1.1.4, 5.1.6.4	M	Y
MLF8	アソシエーション	[802.15.4] Clause 5, 6.2.2, 6.2.3, 5.1.3	M	N※1
MLF9	セキュリティ	[802.15.4] Clause 7	M	Y
MLF9.1	非セキュアモード	[802.15.4] Clause 7	M	Y
MLF9.2	セキュアモード	[802.15.4] Clause 7	O	Y
MLF9.2.1	暗号化	[802.15.4] Clause 7	O.4	Y
MLF9.2.2	改ざん検知	[802.15.4] Clause 7	O.4	Y
MLF10.1	電界強度スキャン	[802.15.4] 5.1.2.1, 5.1.2.1.1	FD1 : M FD2 : O	FD1: Y FD2: O
MLF10.2	アクティブスキャン	[802.15.4] 5.1.2.1.2	FD1 : M FD2 : O	Y
MLF10.3	パッシブスキャン	[802.15.4] 5.1.2.1.2	M	Y
MLF10.4	孤立スキャン	[802.15.4] 5.1.2.1, 5.1.2.1.3	M	O※2
MLF11	スーパーフレーム構成制御	[802.15.4] 5.1.1.1	FD1 : O	N
MLF12	スーパーフレーム構成サポート	[802.15.4] 5.1.1.1	O	N
MLF13	１トランザクションの保管	[802.15.4] 5.1.5	FD1 : M	Y
MLF15	複数 PHY 管理	[802.15.4g] 5.1.9	FD8 : M	N

※1：上位層で代替するので、本機能は使用しない

※2：上位層で必須でないため、本機能は使用しなくてもよい

表6-32：MAC frames

番号	内容	参考文献の対応節	IEEE 規定		方式 B 使用
			送信	受信	
MF1	ビーコン	[802.15.4] 5.2.2.1	FD1 : M	M	Y
MF2	データ	[802.15.4] 5.2.2.2	M	M	Y
MF3	ACK	[802.15.4] 5.2.2.3	M	M	Y
MF4	コマンド	[802.15.4] 5.2.2.4	M	M	Y
MF4.1	アソシエーション要求	[802.15.4] 5.2.2.4, 5.3.1	M	FD1 : M	N※1
MF4.2	アソシエーション応答	[802.15.4] 5.2.2.4, 5.3.2	FD1 : M	M	N※1
MF4.3	切断通知	[802.15.4] 5.2.2.4, 5.3.3	M	M	N※1
MF4.4	データ要求	[802.15.4] 5.2.2.4, 5.3.4	M	FD1 : M	Y
MF4.5	PAN ID 競合通知	[802.15.4] 5.2.2.4, 5.3.5	M	FD1 : M	N
MF4.6	孤立デバイス通知	[802.15.4] 5.2.2.4, 5.3.6	M	FD1 : M	O※2

MF4.7	ビーコン要求	[802.15.4] 5.2.2.4, 5.3.7	FD1 : M	FD1 : M	Y※3
MF4.8	親機再構成	[802.15.4] 5.2.2.4, 5.3.8	FD1 : M	M	Y
MF4.9	GTS 要求	[802.15.4] 5.2.2.4, 5.3.9	MLF5 : O	MLF5 : O	N
MF5	32bit 誤り検出	[802.15.4g] 5.2.1.9	FD6 : M	FD6 : M	Y※4

※1：上位層で代替するので、本機能は使用しない

※2：上位層で必須でないため、本機能は使用しなくてもよい

※3：子機も使用可能(参照規格にない FD2 規定を明確化)

※4：PSDU サイズが 255 オクテット以下であれば、16bit 誤り検出を使用。

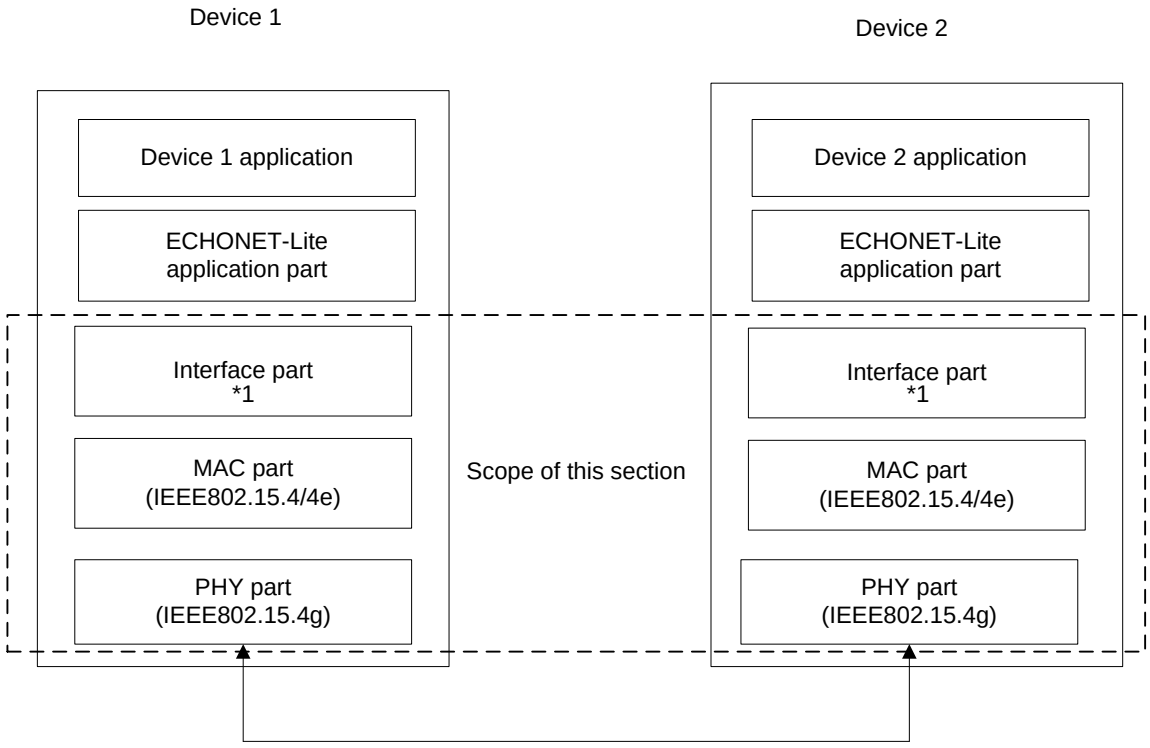
7. 方式 C

7.1. 概要

本章では、コーディネータとホスト間で IEEE802.15.4/4e/4g のみを利用した ECHONETLite 通信に必要となる、物理層部、データリンク層部、およびオプションとして提供するインタフェース部等、について定義する（7.3-7.9）とともに EHONET Lite を用いてシングルホップネットワークを構成する場合の推奨仕様を規定する(7.10)。

物理層部、データリンク層部は、IEEE802.15.4/4e/4g 規格の中で選択された機能で構成されている。一方、インタフェース部は、ECHONET Lite アプリケーション部で指定するアドレス体系と、データリンク層部のアドレス体型が異なる場合を想定し、ECHONET Lite アプリケーション部とデータリンク層、物理層とを直接接続するインタフェースであり ECHONET Lite アプリケーション部からの送信データをデータリンク層、物理層を使用して相手デバイスに送信し、相手装置から受信データを ECHONET Lite アプリケーション部に通知する。**図 7-1**に各部の位置づけを示す。また、**図 7-2**にレイヤ構成について示す。

なお、本章において、”M”は標準化ドキュメント[802.15.4], [802.15.4e], [802.15.4g]として必須機能(マンダトリ)を意味し、”O”はオプション機能、”Y”は ECHONET Lite を動作させる上で必要性がある機能、“N”は必要性がない機能を示している。適合試験仕様、手順および相互接続試験仕様、手順は[Wi-SUN-PHY], [Wi-SUN-MAC], [Wi-SUN-IF], [Wi-SUN-CTEST], [Wi-SUN-ITEST]によって提供される。

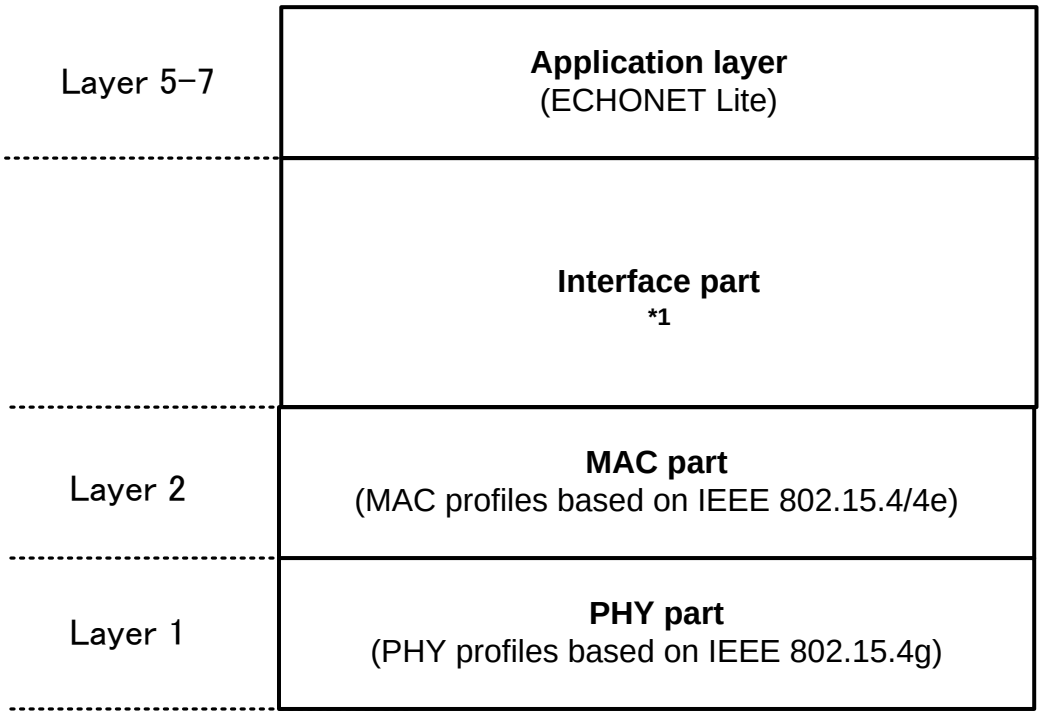


(*1： ECHONET Lite アプリケーション部で指定するアドレス体系とデータリンク層部のアドレス体系が同じ場合ある場合は、必要はない。)

図7-1 本章で対象とする範囲

7.2. プロトコルスタック

本方式が規定するノードが搭載するプロトコルスタックは**図 7-2**のようになる。



(*1： ECHONET Lite アプリケーション部で指定するアドレス体系とデータリンク層部のアドレス体系が同じ場合ある場合は、必要はない。)

図7-2　本章で定義するレイヤ構成

物理層は、本方式で使用する範囲内では次のサービスを提供する。

- 最大 2047 オクテットの PSDU の転送(ただし、システムとしては後述するように 255 オクテット以下を推奨)

データリンク（MAC）層は、本方式で使用する範囲内では次のサービスを提供する。

- 無線到達距離内における、IEEE802.15.4 PAN の発見
- スリープ状態と起床状態を繰り返す省電力ホストのサポート
- 暗号化・改ざん検出・リプレイ攻撃対策の機能を含むセキュリティ機能(鍵管理はこのレイヤで実行されていないことに注意すること)

アプリケーション層は、次のサービスを提供する。

- ネットワーク内の他ノードが保有する機能単位(ECHONET オブジェクト)の検出
- 他ノードが有する各種パラメータ・状態(ECHONET プロパティ)の取得
- 他ノードの各種パラメータ・状態の設定
- 自ノードが有する各種パラメータ・状態の通知

7.3. 物理層部

5.3 を参照のこと

7.4. データリンク層 （MAC 層）部

5.4 を参照のこと

7.5. インタフェース部

7.5.1. 概要

インタフェース部は、ECHONET Lite アプリケーション部で指定するアドレス体系とデータリンク層部のアドレス体系が異なる場合を想定し、ECHONET Lite のアプリケーション部と物理層部/データリンク層部との間に実装し、両部間の通信機能を提供しなければならない。このインタフェースは IP 利用時のオーバーベッドを削減することによりフレーム利用効率を高めることが可能である。

7.5.2. 所要条件

(1) インタフェース部を利用する場合は、独自の送信先アドレスを指定しなければならない。そして、送信元アドレスとInterface Typeを指定することによりECHONET Interface headerヘッダを構築しなければならない。このとき、そのInterface typeは0xEC00でなければならない。

(2) インタフェース部は、MAC部で利用しているアドレス形態を事前に知らなければならない。アドレス形態はEUI-64ビットアドレスである必要がある。

(3) インタフェース部は、MAC部で利用しているアドレス形態に合わせてインタフェース部で指定した独自の送信先アドレスを変換し、MAC部に送信する必要がある。

(4) インタフェース部は、ECHONET Liteアプリケーション部から送信されるMACアドレスがマルチキャストMACアドレスの場合、MAC部に対してブロードキャスト送信を指示しなければならない。

7.6. アプリケーション層

アプリケーション層としては、ECHONET Lite [EL]を使用し、[EL]に規定される必須機能をすべてサポートしなければならない。

7.7. セキュリティ

Non-IP 時のセキュリティを確保する場合、下記の２つの方法があり、いずれか一項目を必須項目として実装しなければならない。

- IEEE802.15.4 MAC部でのデータ暗号化 （インタフェース部を使用しない場合は必須）
- インタフェース部でのデータ暗号化

インタフェース部でのデータ暗号化を行う場合の暗号化方式は、AES-CCM、AES-GCM のいずれかもしくは両方を実装しなければならない[EL], [CMAC], [AES-CCM], [AES-GCM]。AES-CCM/GCM を使用するために、MIC(messageintegrity code)もしくは AAD(Additional Authenticated Data)が使用されなければならない。IEEE802.15.4 MAC 部で AES-CCM のデータ暗号化を行う場合は、その MIC は文献[1]に記載の IEEE802.15.4 MAC フレームに含まれなければならない。一方でインタフェース部でのデータ暗号化を行う場合は、MIC は 4.4.6.5 節に記載のセキュリティヘッダの中に含められなければならない。

7.8. デバイス ID

Non-IP 時のオプション機能として、各 ECHONET Lite 対応デバイスに割り当てられたデバイス ID を使ってもよい。このデバイス ID は、MAC アドレスの初期設定等で用いてもよい。この場合、扱うペイロードは、情報電文(Information payload)と設定電文(setting payload)の 2 種類に分けられる。そして、ECHONET Lite データを送受信するときは情報電文として、デバイス ID を送受信するときは設定電文として利用される。

7.9. フレームフォーマット

本節では本方式のフレームフォーマットを記載する。フレームフォーマットはインタフェース部を使用する場合と使用しない場合とで異なるが、受信ノードにおけるこれらの識別は、別途提供される方式間の共存仕様により行う。

7.9.1. インタフェース部を使用する場合

7.9.1.1. MAC 部でのデータ暗号化を使用する場合

MAC 部でのデータ暗号化を使用する場合のフレームフォーマットの手順のサンプルを**図 7-3**から**図 7-5**に示す。これは、ECHONET Interface header 内の destination および source MAC アドレスが IEEE802.15.4 MAC ヘッダの中の destination および source MAC アドレスが異なる場合を示しており、この 2 つのアドレスは省略することができる。

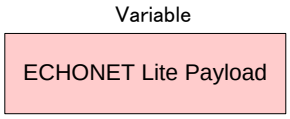


図7-3：ECHONET Lite ペイロード

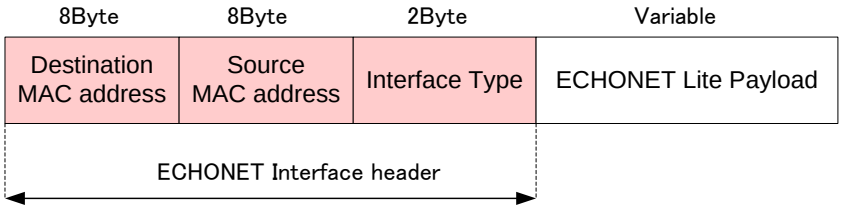


図7-4：インタフェース部で構築されるフレーム

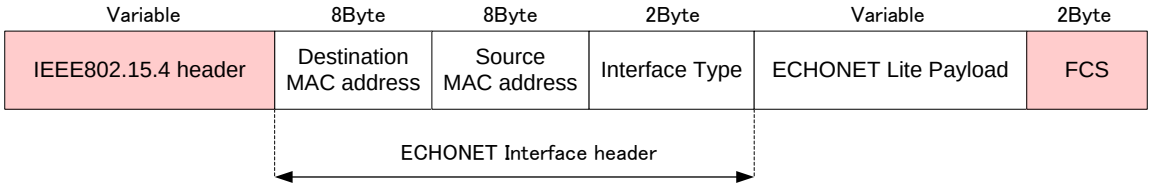


図7-5：MAC 部で構築される IEEE802.15.4 フレーム

7.9.1.2. インタフェース部でのデータ暗号化を使用する場合

インタフェース部でのデータ暗号化を使用する場合のフレームフォーマットの手順のサンプルを**図 7-6**から**図 7-8** に示す。これは、ECHONET Interface header 内の destination および source MAC アドレスが IEEE802.15.4 MAC ヘッダの中の destination および source MAC アドレスが異なる場合を示しており、この 2 つのアドレスは省略することができる。

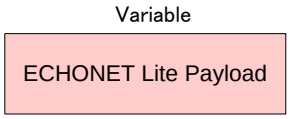


図7-6：ECHONET Lite ペイロード

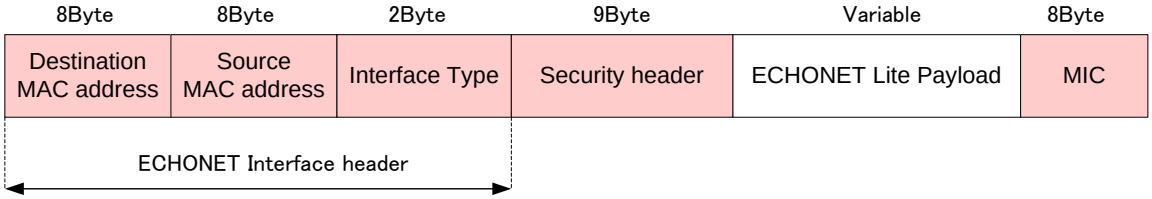


図7-7：インタフェース部で構築されるフレーム

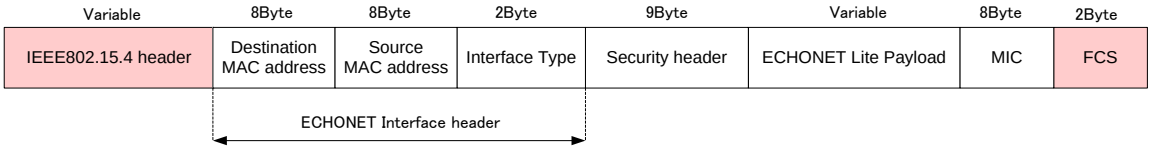


図7-8：MAC 部で構築される IEEE802.15.4 フレーム

7.9.1.3. デバイス ID オプションを使用し、インタフェース部でのデータ暗号化を使用する場合

デバイス ID オプションを使用し、インタフェース部でのデータ暗号化を使用する場合の ECHONET Lite アプリケーションからの情報電文を MAC 部フレームに変換するための手順を示す。これは、ECHONET Interface header 内の destination および source MAC アドレスが IEEE802.15.4 MAC ヘッダの中の destination および source MAC アドレスが異なる場合を示しており、この 2 つのアドレスは省略することができる。

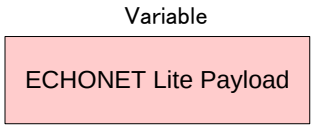


図7-9：ECHONET Lite ペイロード

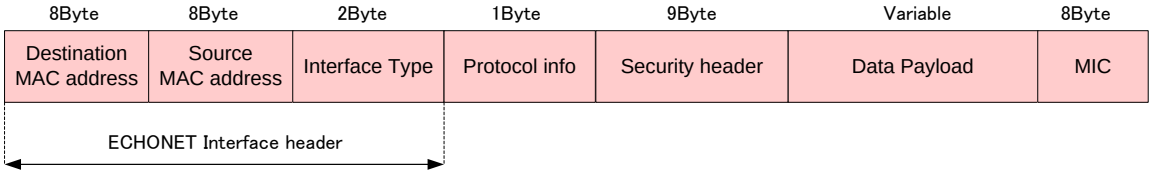


図7-10：インタフェース部で構築されるフレーム

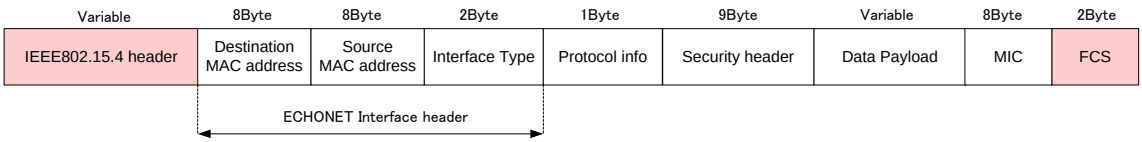


図7-11： MAC 部で構築される IEEE802.15.4 フレーム

7.9.1.4. フレーム構成要素

7.9.1.4.1. ECHONET Lite payload

ECHONET Lite アプリケーション部にて、生成される ECHONET Lite 情報電文。

7.9.1.4.2. ECHONET Interface header

インタフェース部にて、生成される独自ヘッダであり、**図 7-12**にその構成を示す。

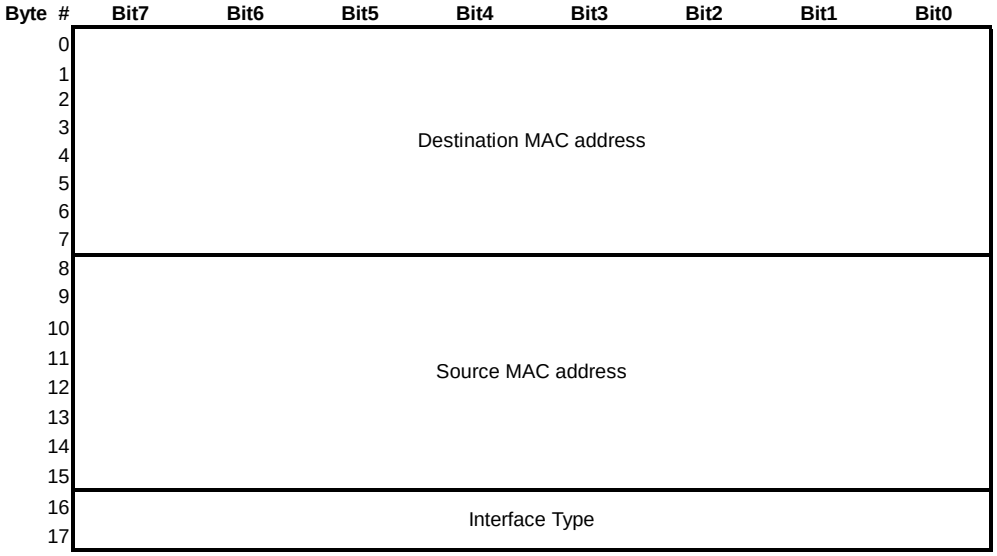


図7-12：ECHONET interface header のフォーマット

(a) Destination address

ECHONET Lite アプリケーション部とインタフェース部が協調して決定する送信先のアドレス。

(b) Source address

送信元の MAC アドレス、インタフェース部にて MAC 部のアドレス形態を基に設定される。

(c) Interface Type

0xEC00： ECHONET Lite 用 Interface Type

7.9.1.4.3. IEEE802.15.4 header

MAC 部にて、生成される送受信用のヘッダ。

7.9.1.4.4. FCS (Frame check sequence)

MAC 部にて、生成されるフレームチェックシーケンス。

7.9.1.4.5. Security header

送信データに対する暗号化に関する情報を定義する。 **図 7-13** にそのフォーマットを示す。

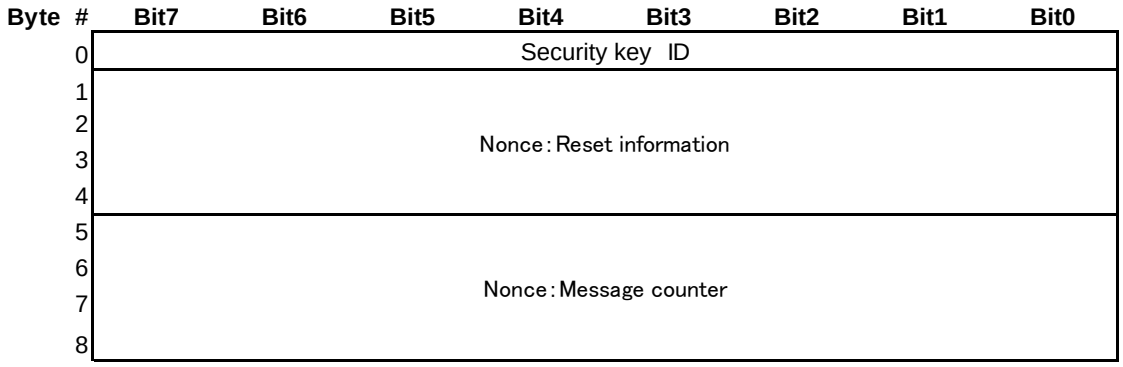


図7-13：Security header のフォーマット

(a) Security key ID

使用する暗号鍵に対応した識別子である。

(b) Nonce (byte# 1-8)

送信するデータごとに独自の値を設定し、データと共に暗号化される。以下の要素からなる。

Reset information (byte# 1-4) : デバイスのリセット毎に増加する値を設定する。

Message counter (byte# 5-8): メッセージ送付数をカウントするカウンタ。

7.9.1.4.6. MIC (Message Integrity Code)

AES-CCM の暗号化に利用されるコード。

7.9.1.4.7. Protocol info

送信するデータのプロトコル種別を示す。独自の機器 ID を定義した場合に利用され、**図 7-14**にそのフォーマットを示す。

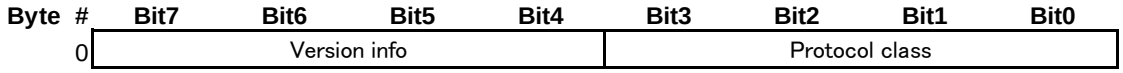


図7-14：Format of protocol info

(a) Version info: 4 ビット用い、16 バージョンまで対応可能である。

(b) Protocol class: 設定用の電文と情報用の電文を識別する。

0000: 情報電文, 0001: 設定電文

7.9.1.4.8. Data payload

情報電文もしくはデバイス ID からなる設定電文のデータを伝送する。そしてこれらは、プロトコル種別により選定される。以下にフォーマットを示す。

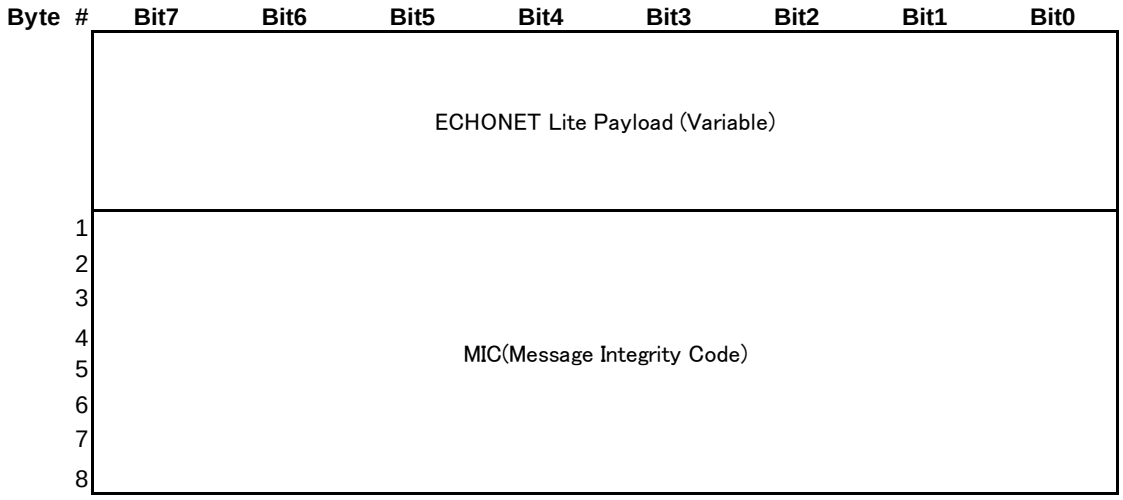


図7-15：情報電文用 payload のフォーマット

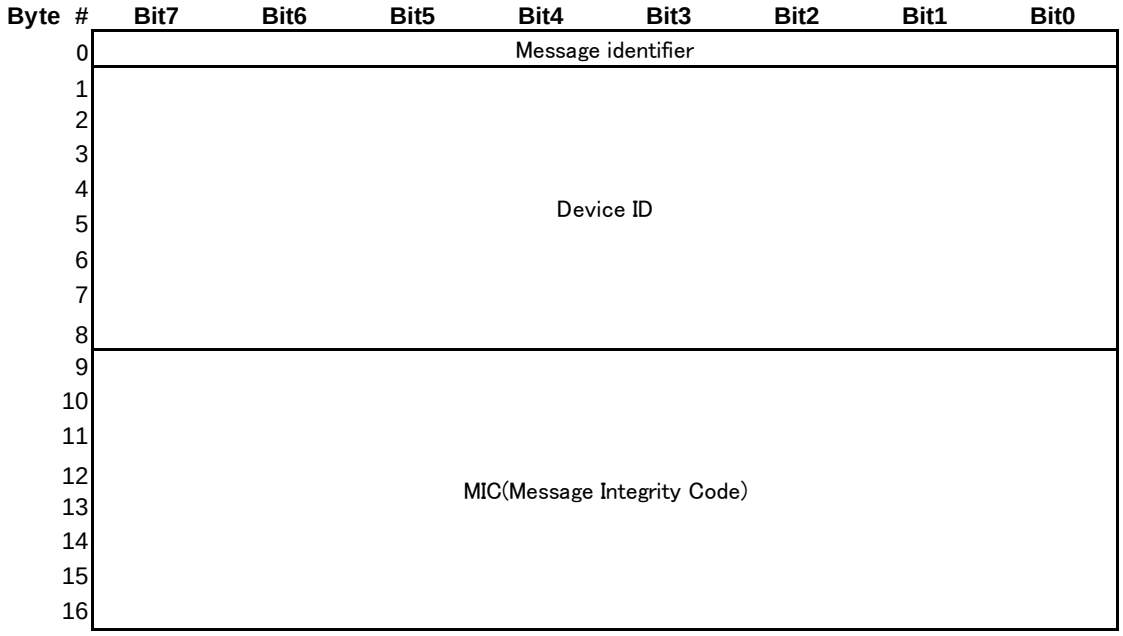


図7-16：設定電文用 payload のフォーマット

(a) Message identifier: 各種設定を要求する場合と、応答する場合で識別を行う。

00000000: 設定要求

00000001: 設定応答

7.9.2. インタフェース部を使用しない場合

ECHONET Lite アプリケーション部が直接 IEEE802.15.4 の MAC アドレスを扱う場合は、インタフェース部が不要である。インタフェース部を使用しない場合のフレームフォーマットのサンプルを**図 7-17**から**図 7-18**に示す。インタフェース部を使用しない場合は ECHONET Lite Payload の前に直接 IEEE802.15.4 のヘッダが配置される。

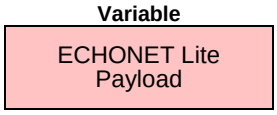


図7-17：ECHONET Lite ペイロード

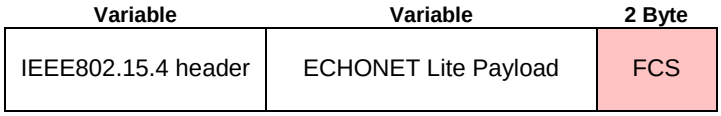


図7-18：MAC 部で構築される IEEE802.15.4 フレーム

7.10. シングルホップネットワークを構成する場合の推奨仕様

7.10.1. 概要

本節では、方式 C を用い、ECHONET Lite を利用するシングルホップネットワークを構築する場合の推奨仕様を示す。ただし、方式 C の範囲内においてこれ以外の仕様を排除するものではない。

本節の仕様に基づくノードは、コーディネータを中心としたシングルホップネットワークを構築する。また、外部ネットワークとの接続方法としてアプリケーションレベルのゲートウェイ接続を想定することで、本方式内に閉じたネットワークを想定している。これらの前提事項により、ECHONET Lite を用いた宅内ネットワークの構築を可能としながらも、実装の容易性を実現している。

7.10.2. 新しいネットワークの形成

コーディネータが起動すると、本方式仕様に基づく新しいネットワークを形成する。ネットワークの形成は、(1) データリンク層の設定、(2)セキュリティの設定の順に実施される。ネットワーク形成手順の概要を、**図 7-19**に示す。

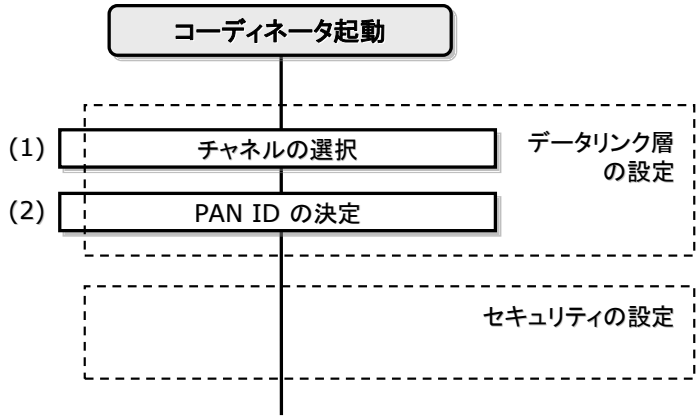


図7-19：ネットワーク形成手順概要

7.10.2.1. データリンク層の設定

コーディネータが起動すると、IEEE802.15.4 PAN を形成する。PAN 形成に関する詳細な手順は以下のとおりである。

コーディネータはまず使用するチャネルの選択を行う。チャネル選択は、ED スキャンやアクティブスキャンを利用して実施する。その際、他システムとの干渉が小さいと想定されるチャネルを選択することが好ましい。(ステップ 1)

最後に、コーディネータはステップ 1 で選択したチャネルにおいて検出されたいずれの PAN も使用して

いない PAN ID を選択し、自身の管理するネットワークにて使用する PAN ID とする。ステップ 1 で選択したチャネルにおいて検出されたいずれの PAN も使用していない PAN ID の中からどのような値を自ネットワークの PAN ID として選択するかについては、本方式では規定しない。(ステップ 2)
以上を実施した後、コーディネータは決定した無線チャネルと PAN ID により PAN の形成を完了する。

7.10.2.2. セキュリティの設定

データリンク層の設定が完了すると、コーディネータは、セキュリティの設定を行う。ここで形成されるネットワークで利用するセキュリティ技術は、アプリケーション要件に応じて適切に選択する。本方式ではコーディネータによるセキュリティ設定の具体的な手順は記載しない。

7.10.3. ネットワークへの参加

新規ホストが起動すると、本方式が規定する既存のネットワークへの参加を試みる。ホストのネットワークへの参加も、コーディネータによるネットワーク形成と同様、(1) データリンク層の設定、(2) セキュリティの設定の手順に大別される。新規ホストが既存のネットワークに参加するための手順の概要を**図 7-20**、に示す。

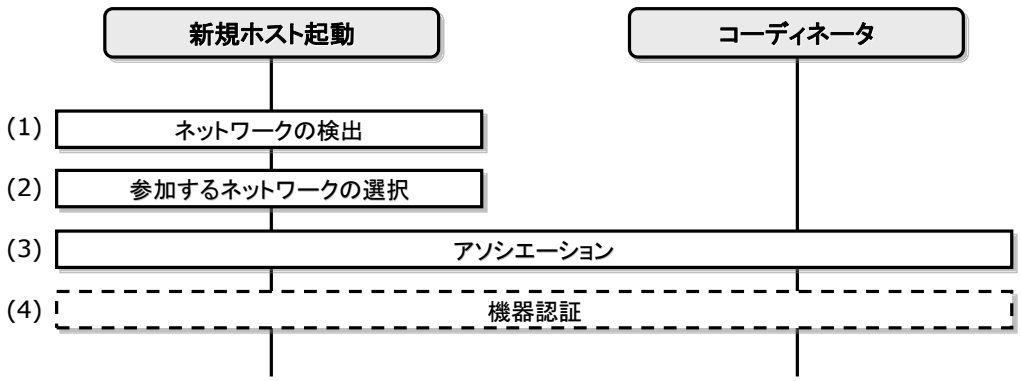


図7-20：ネットワーク参加手順概要

7.10.3.1. データリンク層の設定

新規ホストが起動すると、まず周囲に存在する IEEE802.15.4 PAN の検出を行う。PAN の検出は、新規ホストが[802.15.4]および[T108]で規定される無線チャネルのうち使用可能なすべてのチャネルにおいて[802.15.4]で規定されるビーコン要求コマンドメッセージを送信し、それを受信したコーディネータが応答としてビーコンフレームを送信、新規ホストがこのビーコンを受信することで実現される。また、新規ホストはこれらの手順の結果として、コーディネータが使用する無線チャネルと PAN ID を識別することができる。(ステップ 1)

ステップ 1 において PAN が 1 つのみ検出された場合、その PAN に対して次のステップに進む。複数の PAN が検出された場合はいずれか 1 つの PAN を選択して次のステップに進むが、どの PAN を選択するかは実装依存とする。(ステップ 2)

ステップ 2 において、選択した PAN に対して、新規ホストは IEEE802.15.4 で規定されるアソシエーションを実施する。(ステップ 3)

ここで選択した PAN についてアソシエーションを実施した結果として、コーディネータによる接続拒否などにより、ネットワークへの参加に失敗した場合、新規ホストはステップ 1 もしくはステップ 2 に戻って参加手順を再実施することが推奨される。また、その場合はステップ 2 においては既に参加に失敗したネッ

トワーク以外のネットワークを選択すべきである。

7.10.3.2. セキュリティの設定

IEEE802.15.4 PAN への参加が完了すると、新規ホストは、コーディネータとのセキュリティの設定を行う。ここで形成されるネットワークで利用するセキュリティ技術に関しては本方式の規定範囲外であり、本方式ではネットワークへの参加に伴うセキュリティ設定の具体的な手順は記載しない。

7.10.4. 推奨仕様動作例を実現するためのデバイス/物理層/MAC 層の仕様

5.8.4を参照のこと。