

SmartMesh IP Embedded Manager API ガイド

目次

1	本書について	6
1.1	関連資料	6
1.2	表記規則	8
1.3	改訂履歴	9
2	はじめに	10
2.1	通信	10
3	プロトコル	11
3.1	データ表現	11
3.1.1	共通データ・タイプ	11
3.1.2	整数表現	12
3.1.3	伝送順序	13
3.2	パケット形式	13
3.2.1	フレーミング	13
3.2.2	データ・ペイロード	13
3.3	マネージャとクライアント間の通信	15
3.3.1	通信の開始	15
3.3.2	信頼できる(アクノリッジ対象)通信	17
3.3.3	ベスト・エフォート(アクノリッジ対象外)通信	17
3.3.4	前方互換性を備えたクライアントの作成ガイドライン	18
4	コマンド	19
4.1	clearStatistics(0x1F)	20
4.2	deleteACLEntry(0x29)	21
4.3	deleteMote(0x45)	22
4.4	exchangeMoteJoinKey(0x21)	23
4.5	exchangeNetworkId(0x22)	24
4.6	getIPConfig(0x43)	25
4.7	getLicense(0x37)	26
4.8	getLog(0x2B)	27
4.9	getManagerStatistics(0x35)	28
4.10	getMoteConfig(0x2F)	29
4.11	getMoteConfigById(0x41)	30

4.12	getMoteInfo (0x3E)	31
4.13	getMoteLinks (0x46)	33
4.14	getNetworkConfig (0x3F)	35
4.15	getNetworkInfo (0x40)	37
4.16	getNextACLEntry (0x28)	38
4.17	getNextPathInfo (0x31)	39
4.18	getPathInfo (0x30)	41
4.19	getRadiotestStatistics (0x26)	42
4.20	getSystemInfo (0x2E)	43
4.21	getTime (0x17)	44
4.22	pingMote (0x2A)	45
4.23	radiotestRx (0x25)	46
4.24	radiotestTx (0x23)	47
4.25	reset (0x15)	49
4.26	restoreFactoryDefaults (0x3D)	50
4.27	sendData (0x2C)	51
4.28	sendIP (0x3B)	53
4.29	setACLEntry (0x27)	56
4.30	setAdvertising (0x32)	57
4.31	setCLIUser (0x3A)	58
4.32	setCommonJoinKey (0x42)	59
4.33	setDownstreamFrameMode (0x33)	60
4.34	setIPConfig (0x44)	61
4.35	setLicense (0x38)	62
4.36	setNetworkConfig (0x1A)	63
4.37	setTime (0x36)	65
4.38	startNetwork (0x2D)	66
4.39	subscribe (0x16)	67
5	通知 (0x14)	68
5.1	ブリンク通知	69
5.2	data通知	70
5.3	event通知	71
5.3.1	commandFinishedイベント	72
5.3.2	pathCreateイベント	73
5.3.3	pathDeleteイベント	74

5.3.4	pingイベント	75
5.3.5	networkTimeイベント	76
5.3.6	networkResetイベント	77
5.3.7	moteJoinイベント	78
5.3.8	moteCreateイベント	79
5.3.9	moteDeleteイベント	80
5.3.10	moteLostイベント	81
5.3.11	moteOperationalイベント	82
5.3.12	moteResetイベント	83
5.3.13	packetSentイベント	84
5.3.14	invalidMICイベント	85
5.3.15	joinFailedイベント	86
5.4	healthReport通知	87
5.4.1	健全性レポートのメッセージ	87
5.5	ipData通知	90
5.6	log通知	92
6	定義	93
6.1	パケットおよびコマンド・タイプ	93
6.2	通知タイプ	94
6.3	サブスクリプション・フィルタ	95
6.4	イベント・タイプ	95
6.5	レスポンス・コード	95
6.6	フレーム・プロファイル	96
6.7	アドバタイズメント・ステート	97
6.8	下リフレーム・モード	97
6.9	ネットワーク・ステート	97
6.10	モート・ステート	97
6.11	リセット・タイプ	97
6.12	バックボーン・フレーム・モード	98
6.13	パス・フィルタ	98
6.14	パスの方向	98
6.15	パケット優先順位	98
6.16	コマンド終了結果	98
6.17	送信出力	98
6.18	CCAモード	99

6.19	リンク・フラグ	99
6.20	近隣モート・フラグ	99
6.21	CLIユーザ・ロール	99
6.22	無線テスト・タイプ	99
6.23	参加失敗の理由	100

1 本書について

1.1 関連資料

SmartMesh IPネットワーク向けに以下の資料が提供されています。

スタータ・キットのクイック・ガイド

- [SmartMesh IP Easy Start Guide](#) - 基本的なインストール方法とネットワークの動作確認テストについて説明しています。
- [SmartMesh IPツール・ガイド](#) - インストールのセクションではシリアル・ドライバのインストール手順について説明しており、Easy Start Guideやその他のチュートリアルで使用されるサンプル・プログラムも含まれています。

ユーザ・ガイド

- [SmartMesh IPユーザ・ガイド](#) - ネットワーク概念についての説明と、モートおよびマネージャのAPIを使用して特定のタスク（データ送信や統計情報の収集など）を実行する方法について説明しています。この資料は、APIガイドを使用するための予備知識を提供します。

デバイスの対話操作インターフェース

- [SmartMesh IP Manager CLI Guide](#) - ユーザがマネージャとやり取りするために使用します（クライアントの開発中やトラブルシューティングなど）。このガイドは、CLIの接続とそのコマンド・セットについて説明しています。
- [SmartMesh IP Manager API Guide](#) - プログラムを使用してマネージャとやり取りするために使用します。このガイドは、APIの接続とそのコマンド・セットについて説明しています。
- [SmartMesh IP Mote CLI Guide](#) - ユーザがモートとやり取りするために使用します（センサー・アプリケーションの開発中やトラブルシューティングなど）。このガイドは、CLIの接続とそのコマンド・セットについて説明しています。
- [SmartMesh IP Mote API Guide](#) - プログラムを使用してモートとやり取りするために使用します。このガイドは、APIの接続とそのコマンド・セットについて説明しています。

ソフトウェア開発ツール

- [SmartMesh IPツール・ガイド](#) - [SmartMesh SDK](#)に含まれる各種の評価および開発サポート・ツールについて説明しています。モートおよびマネージャAPIの使用とネットワークの視覚化のためのツールを含みます。

アプリケーション・ノート

- [SmartMesh IPアプリケーション・ノート](#) - SmartMesh IPネットワーク固有の各種トピックと、SmartMeshネットワーク全般に当てはまるトピックが含まれています。

新規設計の開始時に役立つ資料

- [LTC5800-IPM SoC](#)またはこれに基づく[モジュール](#)のデータシート。
- [LTC5800-IPR SoC](#)またはこれに基づく[組込みマネージャ](#)のデータシート。
- モート／マネージャSoC用または[モジュール用のハードウェア統合ガイド](#) - 設計にSoCまたはモジュールの統合を盛り込むためのベスト・プラクティスを提供しています。
- 組込みマネージャ用の[ハードウェア統合ガイド](#) - 設計する際に組込みマネージャを統合させるためのベスト・プラクティスを提供しています。

- [Board Specific Integration Guide](#) - SoCモートおよびマネージャに対するデフォルトのI/O設定方法と、「ヒューズ表」を使用した水晶発振器のキャリブレーション情報について説明しています。
- [Hardware Integration Application Notes](#) - SoC設計チェックリスト、アンテナ選定ガイドなどを含みます。
- [ESP Programmer Guide](#) - DC9010 Programmer Boardと、デバイスへのファームウェアのロードに使用するESPソフトウェアのガイドです。
- ESPソフトウェア - モートまたはモジュールにファームウェア・イメージをプログラミングするために使用します。
- Fuse Tableソフトウェア - [Board Specific Configuration Guide](#)で説明されているヒューズ表を作成するために使用します。

その他の役立つ資料

- SmartMesh技術文書で使用するワイヤレス・ネットワークの用語については、[SmartMesh IPユーザ・ガイド](#)を参照してください。
- [よくある質問](#)の一覧。

1.2 表記規則

本書では、以下の表記規則を使用します。

コンピュータ・タイプ (Computer type) は、URLの指定など、ユーザが入力する情報を示します。

太字は、ボタン、フィールド、メニュー・コマンド、デバイス・ステート、モードを示します。

斜体は、新しい用語やAPIとそのパラメータを示します。

✔ ヒントは、製品に関して役立つ情報を提供します。

i 情報テキストは、背景や前後関係の理解に役立つ追加情報を提供します。

⚠ 注記は、概念についてより詳しい説明を提供します。

⚡ 警告！警告は、データ損失やハードウェアまたはユーザへの物理的な損害を引き起こす可能性のある動作をユーザに知らせます。

コード・ブロックは、コード例を示します。

1.3 改訂履歴

リビジョン	日付	説明
1	2012/07/18	初期リリース
2	2012/08/10	radiotestTx APIの更新
3	2013/03/18	多数の軽微な変更
4	2013/10/22	radioTxフィールドの修正、RCハンドリングの明確化、その他の軽微な変更
5	2014/04/04	無線テスト・コマンドの更新および明確化
6	2014/10/23	タイトルへのコマンドIDの追加、maxMotes設定の明確化、その他の軽微な変更
7	2015/04/22	autostartコマンドの廃止、setTimeコマンドの廃止、キュー占有率の修正、その他の軽微な変更
8	2015/12/03	ソフトウェア・ライセンス・コマンドの廃止、組込みマネージャに焦点を合わせるためのガイド名の変更、その他の軽微な変更
9	2016/11/07	joinFailedおよびinvalidMICイベントの追加、RC_UNSUPPORTEDレスポンス・コードの追加、RSSIレポートの説明の追加、getMoteInfoに対するフィールドの追加、ブリンク通知の追加

2 はじめに

本書では、シリアル・アプリケーション・プログラミング・インターフェース (API) を介した SmartMesh IP Manager との通信で使用するコマンドについて説明します。API はマシン間通信 (ホストプログラムとマネージャの通信など) での使用を意図したものです。

コマンド・ライン・インターフェース (CLI) は、開発中やインタラクティブなトラブルシューティングなどで、ユーザがマネージャとやり取りするために使用します。詳細については、[SmartMesh IP Manager CLI Guide](#) を参照してください。

2.1 通信

メッセージは、パケット・ベースのプロトコルを使用して送信されます。本書で触れるマネージャ機能の概要については、[SmartMesh IP ユーザ・ガイド](#) を参照してください。

シリアル API を使用するため、クライアントは最初に、[マネージャとクライアント間の通信](#) の記載に従ってマネージャとのセッションを確立します。セッションが確立されると、クライアントは一連の [コマンド](#) をマネージャに送信することで、ネットワーク操作を制御するか、ネットワーク内のモートにデータを送信します。また、[subscribe](#) コマンドを使用して、マネージャからの [通知](#) を要求します。

シリアル API には、HDLC シリアル・プロトコルを使用して、シリアル・ポートから直接アクセスできます。それぞれのコマンドのリクエスト、レスポンス、通知をまとめて HDLC フレームが形成されます。このシリアル API で使用されるプロトコルについては、[プロトコル](#) のセクションを参照してください。

このシリアル API で記述される構造は以下のとおりです。

- [コマンド](#): コマンドは、クライアントが開始する RPC 形式のリクエストで、マネージャからレスポンスが返されます。
- [通知](#): 通知は、マネージャからクライアントに送信される非同期メッセージです。

マネージャのシリアル接続が有効なのは、物理的に 1 つのクライアント・エンドポイントに対してのみですが、シリアル API マルチプレクサ (「Serial Mux」) を介したシリアル API へのアクセスも可能です。Serial Mux を使用すると、同時に複数のクライアントからマネージャに接続できます (ユーザ GUI と統計ロガーなど)。基盤となるコマンドおよび通知構造は同じですが、Serial Mux は異なるパケット形式を使用してメッセージをカプセル化します。このメッセージ形式については、Serial Mux の技術文書を参照してください。

3 プロトコル

SmartMesh IP ManagerシリアルAPIプロトコルは、パケット・ベースの通信プロトコルで、非同期シリアル・ポートを介した通信を可能にします。デフォルトでは、このプロトコルはRS232を介した115kbps、8ビット、パリティなし、ストップ・ビット1の通信です。RTS/CTSハードウェア・ハンドシェイクを使用しますが、本書ではこれについて説明しません。

- **パケット形式**のセクションでは、双方から送信されるパケットのHDLC形式を説明します。
- **データ表現**のセクションでは、データ・タイプのシリアライズを説明します。
- **マネージャとクライアント間の通信**のセクションでは、マネージャとクライアントの間でどのように通信が開始されるかを説明します。

3.1 データ表現

3.1.1 共通データ・タイプ

このAPIガイドでは、データを表すために以下のデータ・タイプを使用します。

タイプ	長さ(バイト)	注釈
INT8U	1	符号なしバイト。
INT16U	2	短い符号なし整数。
INT32U	4	長い符号なし整数。
INT8S	1	符号付きバイトまたは文字。
INT16S	2	短い符号付き整数。
INT32S	4	長い符号付き整数。
INT8U[n]	n	固定サイズ・アレイ。固定サイズ・アレイは常に[n]個の要素を含みます。有効な値が[n]個より少ない場合、残りがデフォルト値で埋められます。
INT8U[]	可変	可変長アレイ。可変長アレイのサイズは、パケットの長さによって決まります。可変長アレイは、常に、パケット構造の最後のフィールドになります。
IPV6_ADDR	16	INT8U[16]バイト・アレイとして表されるIPV6アドレス。
ASN	5	絶対スロット番号 (ASN) は、ネットワークが起動してからのタイム・スロット数で、5バイトの整数で表されます。
UTC_TIME	8	UTC時間は、1970年1月1日午前0時 (UTC) 以降の秒数およびマイクロ秒数です。シリアライズされた形式は以下のとおりです。 • INT32U - seconds - 1970年1月1日午前0時以降の秒数 • INT32U - microseconds - 現在の秒が始まってからのマイクロ秒数
UTC_TIME_L	12	Long UTC時間は、1970年1月1日午前0時 (UTC) 以降の秒数およびマイクロ秒数です。シリアライズされた形式は以下のとおりです。 • INT64 - seconds - 1970年1月1日午前0時以降の秒数 • INT32 - microseconds - 現在の秒が始まってからのマイクロ秒数

タイプ	長さ(バイト)	注釈
MAC_ADDR	8	INT8U[8]バイト・アレイとして表されるEUI-64識別子またはMACアドレス。
SEC_KEY	16	INT8U[16]バイト・アレイとして表されるセキュリティ・キー。
BOOL	1	True(= 1)、False(= 0)。ブール・フィールドは1バイトを占めます。
APP_VER	5	アプリケーションのバージョン。シリアルライズされた形式は以下のとおりです。 - INT8U - major - メジャー・バージョン - INT8U - minor - マイナー・バージョン - INT8U - patch - パッチ・バージョン - INT16U - build - ビルド・バージョン

3.1.2 整数表現

マルチバイトの数値フィールドは、最上位ビットから始まる順序のオクテット文字列として表されます。全てのオクテットは、最上位ビット(MSB)で始まるバイナリ文字列として表現されます。符号付き整数は、2の補数形式で表されます。

INT8S、INT8U

ビット7	...	ビット0
MSB		LSB

INT16S、INT16U

ビット15～ビット8	ビット7～ビット0
------------	-----------

INT32S、INT32U

ビット31～ビット24	ビット23～ビット16	ビット15～ビット8	ビット7～ビット0
-------------	-------------	------------	-----------

ASN

ビット39～ビット32	ビット31～ビット24	ビット23～ビット16	ビット15～ビット8	ビット7～ビット0
-------------	-------------	-------------	------------	-----------

3.1.3 伝送順序

本書に示す構造はすべて、伝送される順序で左から右へ（表形式の場合は上から下へ）と示します。

3.2 パケット形式

3.2.1 フレーミング

それぞれのパケットは、RFC 1662の規定に従いフラグで区切られたHDLCフレームであり、2バイトのフレーム・チェック・シーケンス(FCS)が付加されます。

フラグ	データ	フレーム・チェック・シーケンス	フラグ
1バイト	Nバイト	2バイト	1バイト

フラグ

このプロトコルでは、0x7E（「フラグ・シーケンス」）を使用してパケットの境界を示します。すべてのパケットの最初と最後にこのフラグが付けられます。フラグ・シーケンスのインスタンスは、フレーミング・エラーを防ぐために再エンコードする必要があります。この「オクテット・スタフティング」手法については、RFC 1662を参照してください。

フレーム・チェック・シーケンス(FCS)

FCSフィールドは、受信したパケットの正当性をチェックするために使用します。このフィールドは、各パケットの情報部分の全バイトに対して計算されますが、非同期伝送で追加されることのあるスタート／ストップ・ビットは除外されます。特に、フラグ・シーケンスまたはFCSフィールド自体は含まれません。FCSフィールドは、RFC 1662に規定されたアルゴリズムに従って計算されます。

3.2.2 データ・ペイロード

すべてのパケットのデータ・セクションには、以下のフィールドが含まれます。

コントロール	パケット・タイプ	シーケンス番号	ペイロード長	ペイロード
1バイト	1バイト	1バイト	1バイト	0～Nバイト（タイプによって異なります）

コントロール、パケット・タイプ、シーケンス番号、ペイロード長のフィールドが、シリアルAPIヘッダとなります。

コントロール

コントロールは必須のビットマップ・フィールドで、すべてのパケットの最初に含まれます。現時点で定義されているフィールドは以下のとおりです。

ビット0 (DATA/ACK)	0 = データ・パケット (非ACK) 1 = ACKパケット
ビット1 (SERVICE_TYPE)	0 = アクノリッジ対象外 1 = アクノリッジ対象
ビット2～7	予約ビット、0にセットされます。

ビット0は、このパケットがデータ・パケットまたはACKパケットのどちらであることを示します。リクエストはデータ・パケットで、レスポンスはACKパケットです。

ビット1は、パケットにアクノリッジが必要かどうかを示します。セッション確立時のハンドシェイク・パケット ([hello](#)、[mgrHello](#)、[helloResponse](#)) はアクノリッジ対象外 (ACKは不要) です。

パケット・タイプ

パケット・タイプは必須フィールドで、その後に続くパケット・ペイロードの構造を示します。各メッセージに割り当てられるパケット・タイプ値の一覧は、[パケットおよびコマンド・タイプ表](#)を参照してください。アプリケーション・ペイロードを含まないACKのみのパケットの場合、パケット・タイプにはアクノリッジされるパケットと同じ値を設定します。

シーケンス番号

シーケンス番号は、信頼性を備えたデータ通信を実現するためだけに使用されるフィールドです。パケットがアクノリッジ対象外として指定されている場合、このフィールドは無視されます。標準 (ACK以外) のパケットでは、送信者に固有の信頼できるパケット番号が含まれます。ACKパケットでは、アクノリッジされるパケットのシーケンス番号が含まれます。

ペイロード長

ペイロード長は、パケットのペイロード部分のサイズ (バイト) を示します。マネージャが生成および終了するパケットの最大サイズは128バイトです。これにはAPIヘッダおよびペイロードが含まれますが、HDLCフラグ、エスケープ、フレーム・チェックサムは含まれません。

ペイロード

ペイロードが含まれるかどうかと、そのコンテンツを決めるのはタイプ・フィールドです。ペイロードは、標準 (非ACK) パケットとACKパケットのどちらにも含まれる可能性があります。

[コマンド](#)および[通知](#)の構造については、それぞれ該当するセクションを参照してください。

3.3 マネージャとクライアント間の通信

クライアントは、マネージャのAPIシリアル・ポートを介してSmartMesh IP ManagerシリアルAPIと通信します。ここでは、クライアントがマネージャとのセッションを開始する方法を説明します。

3.3.1 通信の開始

*hello*パケットの交換により、クライアントとマネージャ間で新しいセッションが開始されます。このハンドシェイクを使用して、以前の設定をクリアし、プロトコル・バージョンに合意し、シーケンス番号を規定することで、信頼性を備えた次の通信が確立されます。いったん確立されたセッションは、以下のいずれかの状況になるまで持続します。

- マネージャがリセットを実行する
- クライアントが通知をアクノリッジしない
- クライアントが新しい*hello*パケットを送信する

MgrHelloパケット

*mgrHello*パケットはマネージャによって送信され、クライアントとの間に新しいセッションを開始する準備が整っていることを示します。*mgrHello*パケットはまた、クライアントとの既存セッションをマネージャが切断したことを示します。マネージャとクライアントの間にセッションが確立されていないとき、マネージャは定期的に*mgrHello*パケットを送信します。*mgrHello*パケットのペイロードは次のとおりです。

パラメータ	タイプ	説明
version	INT8U	プロトコルのバージョン。本書では、プロトコル・バージョン4について説明します。
mode	INT8U	互換性維持のために予約されています。0をセットします。

helloパケット

*hello*パケットはクライアントによって送信され、マネージャとの間に新しいセッションを開始することを示します。*hello*パケットのペイロードは次のとおりです。

パラメータ	タイプ	説明
version	INT8U	クライアントにサポートされるプロトコルのバージョン。マネージャはこのフィールドをチェックして、クライアントとの互換性を判断します。マネージャがこのプロトコル・バージョンをサポートしていない場合、 <i>helloResponse</i> にサポート対象外バージョンを示すエラー・コードが含まれ、versionフィールドにサポート対象バージョンが含まれます。本書では、プロトコル・バージョン4について説明します。
cliSeqNo	INT8U	クライアント・シーケンス番号は、この <i>hello</i> パケットを示す一意の番号で、通信の信頼性を確保するために使用されます。両側で一意の番号を使用することで、信頼性できるメッセージの重複を検出します。 <i>helloResponse</i> を受信した後、クライアントからの信頼できるコマンドは、「次の」シーケンス番号で開始する必要があります。
mode	INT8U	互換性維持のために予約されています。0をセットします。

helloResponseパケット

*helloResponse*パケットは、クライアントの*hello*への応答として、マネージャからクライアントに送信されます。*responseCode*がOKの場合、マネージャとクライアントの間にセッションが確立されており、コマンドの交換を開始できます。それ以外の場合、クライアントはリトライを実行します。*helloResponse*パケットのペイロードは次のとおりです。

パラメータ	タイプ	説明
responseCode	INT8U	レスポンス・コードは、マネージャがセッションの確立結果を示すために使用します。このコマンドに対するレスポンス・コードの値を次の表に示します。
version	INT8U	このマネージャによってサポートされるプロトコル・バージョン。 <i>responseCode</i> としてOKが返された場合、プロトコル・バージョンが <i>hello</i> メッセージのバージョンと一致しています。
mgrSeqNo	INT8U	マネージャ・シーケンス番号は、マネージャの信頼できるデータ・ストリームに対してシーケンス番号を規定します。この後にマネージャが送信する信頼性を備えたメッセージでは、次のシーケンス番号が使用されます。
cliSeqNo	INT8U	<i>hello</i> メッセージで受信したクライアントの信頼できるシーケンス番号を確認します。クライアントが送信する信頼できるコマンドは、「次の」シーケンス番号で始まる必要があります。
mode	INT8U	互換性維持のために予約されています。0をセットします。

レスポンス・コード

レスポンス・コード	Value	説明
OK	0	セッション確立済み
unsupportedVersion	1	クライアントのプロトコル・バージョンがサポートされていない
invalidMode	2	シリアルAPIをサポートしないモードでマネージャが実行されている

3.3.2 信頼できる(アクノリッジ対象)通信

マネージャとクライアントの間にセッションが確立されると、すべてのリクエスト／レスポンス通信が信頼できる通信になります。つまり、すべてのリクエスト・パケットが、受信側によって明確にアクノリッジされる必要があります。レスポンス・パケットにはリンク層のアクノリッジが含まれますが、オプションでアプリケーション層のレスポンスを含めることもできます(タイプによる)。レスポンスに含まれるアプリケーション層のメッセージは、個別にアクノリッジされません。

前のパケットがアクノリッジされるまで、送信側は次のパケットを送信しないようにします。送信側は、各送信パケットのシーケンス番号を保管します。リクエスト・パケットが短い時間枠内でアクノリッジされない場合、送信側は同じシーケンス番号でパケットを再送する必要があります。何度かリトライした後、送信側はセッションが切断されている可能性を考慮して、*hello*パケットの交換によるセッションの再確立を行います。

受信側でシーケンス番号を管理するとき、以下の要件に従う必要があります。

1. 受信側は、各(アクノリッジ対象)パケットのシーケンス番号を保管します。受け取るシーケンス番号は、送信するシーケンス番号とは別に保管します。
2. 受信パケットのシーケンス番号が以前の(保管済み)シーケンス番号とは異なる場合、受信側はこのパケットを処理します。
3. 受信パケットのシーケンス番号が一致する場合、受信側は重複パケットを処理せずにアクノリッジします。
4. *hello*パケットの交換中に受信したシーケンス番号は、以前のパケットのシーケンス番号として扱われるため、セッションの最初の信頼できるパケットには次のシーケンス番号が含まれることになります。セッションを開始するときのシーケンス番号には任意の値を指定できますが、後続の値はモジュロ256ずつインクリメントします。

 マネージャは、それぞれの間に200ミリ秒の遅延をはさんで、アクノリッジ・パケットの送信を最大3回試行します。有効な応答を受信しない場合、マネージャは接続を終了して未処理パケットを破棄します。接続が破棄された後、新しいクライアントが新しい*hello*パケット交換を完了する必要があります。

3.3.3 ベスト・エフォート(アクノリッジ対象外)通信

クライアントがマネージャの通知をサブスクライブするとき、オプションで、一部の通知をアクノリッジ対象外(ベスト・エフォート)で送信するようにリクエストできます。こうすることで、ネットワークから受信するデータ・トラフィックの総合的なパケット・スループットを高める効果があります。ベスト・エフォートで送信されるパケットは、マネージャによって「アクノリッジ対象外」というマークが付けられます(コントロール・フィールドを参照)。このようなパケットに対して、クライアントは明示的な応答を送り返す必要はありません。

3.3.4 前方互換性を備えたクライアントの作成ガイドライン

シリアルAPIプロトコルは、ソフトウェア・リリースが新しくなってもクライアントの互換性を維持できるように設計されています。将来的なソフトウェアの改定によって以下の変更が生じる可能性を考慮に入れる必要があります。

- 新しいフィールドを追加するために、ペイロード構造が拡張される場合があります。新規フィールドは、最後または予約バイトの位置に追加されます。
- フィールドが廃止としてマークされる場合がありますが、フィールド自体はペイロード構造から削除されません。
- 新しいコマンドおよび通知が追加される場合があります。
- 新しいアラームおよびイベントが追加される場合があります。
- 新しいレスポンス・コードが追加される場合があります。

互換性を維持するには、クライアントが以下のルールに従う必要があります。

- 受信したレスポンス・ペイロードが想定より長い場合、余分なバイトは無視し、既知のバイトのみを処理します。
- 認識されない通知タイプを持つパケットを受信した場合、RC_OKを使用してアクノリッジします。
- 認識されないアラームまたはイベントを受信した場合、RC_OKを使用してアクノリッジします。
- 予約フィールドの値は決して使用せず、無視します。
- リクエスト・ペイロードで未使用または予約としてマークされているフィールドでは、特別な記載のない限り値を0にセットします。
- 認識されないレスポンス・コードを受信した場合、一般的なエラー・レスポンス・コードとして扱います。

上記以外の互換性のない方法でプロトコルが変更された場合、プロトコル・バージョンが変更されます。

4 コマンド

セッションを確立したクライアントは、マネージャにコマンドを送信できます。コマンドは常に、クライアントからマネージャに対して送信され、信頼性を備えています。コマンドはマネージャによってレスポンスでアクノリッジされます。

シリアルAPIヘッダ ([パケット形式](#)を参照) にはパケット・タイプが含まれており、受信側はこれを使用してペイロードに含まれる構造を識別できます。シリアルAPIに定義されたコマンドおよび通知構造は、Serial Mux APIからも使用されます。Serial Muxは異なるヘッダ構造を使用しますが、リクエストおよびレスポンス構造は同じです。

リクエスト構造は、シリアルAPIパケットのペイロード・セクション内でシリアライズされています。パラメータのないリクエストは空のペイロードになります。

シリアルAPIヘッダ	リクエスト構造、0～Nバイト(各コマンドの説明を参照)
------------	-----------------------------

レスポンスのパケット・タイプは、リクエストのパケット・タイプと同じです。ペイロードの1バイト目には常にレスポンス・コード(RC)が含まれ、その後に、残りのシリアライズされたレスポンス構造が続きます。

シリアルAPIヘッダ	レスポンス構造、1～Nバイト(各コマンドの説明を参照)
------------	-----------------------------

すべてのコマンドで、コマンドが成功した場合はRC_OK(0)が返され、失敗した場合はゼロ以外のRCが返されます。現在定義されているゼロ以外のRCは、各コマンドの説明に記載されています。ゼロ以外の数字でエラー・コードのような設定をしておけば、RCの内容によってエラーの原因が分かります(ただしエラー・コードは既存コマンドリストとコンフリクトしないように設定します)。

各コマンドに独自のパケット・タイプ値が割り当てられており、これがシリアルAPIヘッダに含まれます。値の一覧は、[パケットおよびコマンド・タイプ](#)表を参照してください。

ここからは、定義されているコマンドについて説明します。

4.1 clearStatistics (0x1F)

説明

clearStatistics コマンドは、蓄積されたネットワーク統計情報をクリアします。パス品質またはモート統計はクリアしません。

リクエスト

パラメータ	タイプ	一覧	説明
-------	-----	----	----

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

4.2 deleteACLEntry(0x29)

説明

`deleteACLEntry` コマンドは、指定されたモートをアクセス制御リスト(ACL)から削除します。`macAddress` パラメータがすべて0xFFまたは0x00に設定されている場合、ACL全体をクリアします。これは永続的な変更です。

リクエスト

パラメータ	タイプ	一覧	説明
macAddress	MAC_ADDR		モートのMACアドレス

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_NOT_FOUND	指定されたモートがACLに見つからない
RC_WRITE_FAIL	フラッシュ書き込みエラー、新しい設定を保存できない

4.3 deleteMote(0x45)

説明

`deleteMote` コマンドは、マネージャのリストからモートを削除します。削除できるのは、*Lost* または *Unknown* ステートのモートのみです。これは永続的な変更です。

リクエスト

パラメータ	タイプ	一覧	説明
macAddress	MAC_ADDR		モートのMACアドレス

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_NOT_FOUND	指定されたモートが見つからない
RC_INV_STATE	モートのステートがLostではないか、モートがアクセス・ポイントである
RC_WRITE_FAIL	フラッシュ書き込みエラー、新しい設定を保存できない

4.4 exchangeMoteJoinKey(0x21)

説明

`exchangeMoteJoinKey` コマンドはマネージャをトリガすることで、指定されたモートに新しい参加鍵を送信し、このモートに対するマネージャのACLエントリを更新します。レスポンスには `callbackId` が含まれます。処理が完了すると、この `callbackId` を含む `commandFinished` イベント通知が送信されます。これは永続的な変更です。

リクエスト

パラメータ	タイプ	一覧	説明
macAddress	MAC_ADDR		モートのMACアドレス
key	SEC_KEY		新しいモートの参加鍵

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
callbackId	INT32U		コールバックID

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_NOT_FOUND	指定されたMACアドレスを持つモートが見つからない
RC_INV_STATE	モートのステートがoperationalではない
RC_NACK	ユーザ・コマンド・キューがフル状態
RC_WRITE_FAIL	フラッシュ書き込みエラー、新しい設定を保存できない

4.5 exchangeNetworkId(0x22)

説明

`exchangeNetworkId` コマンドはマネージャをトリガすることで、ネットワーク内のすべてのモートに新しいネットワークIDを配布します。レスポンスで `callbackId` が返されます。処理が完了すると、この `callbackId` を含む `commandFinished` 通知が送信されます。これは永続的な変更です。

リクエスト

パラメータ	タイプ	一覧	説明
id	INT16U		ネットワークID

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
callbackId	INT32U		コールバックID

レスポンス・コード

コード	説明
RC_OK	コマンドを受信した
RC_NACK	ユーザ・コマンド・キューがフル状態
RC_IN_PROGRESS	コマンド保留中。リトライする前に、前のコマンドに対する <code>commandFinished</code> 通知を受信するまで待ちます。
RC_WRITE_FAIL	フラッシュ書き込みエラー、新しい設定を保存できない

4.6 getIPConfig(0x43)

説明

getIPConfigコマンドは、IPv6アドレスとマスクを含むマネージャのIP設定パラメータを返します。

リクエスト

パラメータ	タイプ	一覧	説明
-------	-----	----	----

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
ipv6Address	IPV6_ADDR		IPv6アドレス
mask	INT8U[16]		サブネット・マスク

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

4.7 getLicense(0x37)

説明

 `getLicense` コマンドは、1.3.0以上のマネージャでは廃止されています。モート数が32を上回るネットワークでのライセンス使用は不要です。

`getLicense` コマンドは現在のライセンス・キーを返します。

リクエスト

パラメータ	タイプ	一覧	説明
-------	-----	----	----

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
license	INT8U[13]		現在のソフトウェアのライセンス・キー

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

4.8 getLog(0x2B)

説明

getLogコマンドは、マネージャまたはMACアドレスで指定されたモートから診断ログを取得します。

リクエスト

パラメータ	タイプ	一覧	説明
macAddress	MAC_ADDR		モートのMACアドレス

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_NOT_FOUND	指定されたモートが見つからない
RC_INV_STATE	モートのステートがoperationalではない

4.9 getManagerStatistics (0x35)

説明

`getManagerStatistics` コマンドは、マネージャAPIに関する動的な情報と統計を返します。[clearStatistics](#)を使用すると、統計カウントと現在のすべての統計情報がクリアされます。

リクエスト

パラメータ	タイプ	一覧	説明
-------	-----	----	----

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
serTxCnt	INT16U		シリアル・ポートで送信したパケット数。この値をクリアしないと適宜更新 (rollover) される場合があります。
serRxCnt	INT16U		シリアル・ポートで受信したパケット数。この値をクリアしないと適宜更新 (rollover) される場合があります。
serRxCRCErr	INT16U		CRCエラーの数
serRxOverruns	INT16U		検出されたオーバーランの数
apiEstabConn	INT16U		確立されたシリアルAPI接続の数
apiDroppedConn	INT16U		破棄されたシリアルAPI接続の数
apiTxOk	INT16U		シリアルAPIで送信されたリクエスト・パケットのうち、ACK-OKを受信したパケットの数
apiTxErr	INT16U		シリアルAPIで送信されたリクエスト・パケットのうち、ACKエラーを受信したパケットの数
apiTxFail	INT16U		アクノリッジが返されなかったパケットの数
apiRxOk	INT16U		受信およびアクノリッジされたリクエスト・パケットの数
apiRxProtErr	INT16U		受信されたが、無効なパケット形式により破棄されたパケットの数

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

4.10 getMoteConfig(0x2F)

説明

getMoteConfigコマンドは、レスポンスとして単一モートの設定を返します。このコマンドは、引数としてMACアドレスとフラグの2つを受け取ります。フラグは、MACアドレスが要求対象のモートであるか、マネージャのメモリ内の次のモートであるかを示します。このコマンドを使用して、マネージャが認識しているすべてのモートの繰り返し処理を実行することができます。はじめに、macAddress/パラメータに0をセットし、nextにtrueをセットします。次に、レスポンスで返されたMACアドレスを次のコールの入力として使用します。

モートのMACアドレスはすべての問い合わせコマンドで使用されますが、近隣モートの健全性レポートでは、スペースに制約があるため、識別子としてモートIDを使用する必要があります。このため、モート構造には両方の識別子が含まれています。

リクエスト

パラメータ	タイプ	一覧	説明
macAddress	MAC_ADDR		モートのMACアドレス
next	BOOL		次のモートを対象とする場合はtrue、このMACアドレスを対象とする場合はfalse

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
macAddress	MAC_ADDR		モートのMACアドレス
moteld	INT16U		モートID (健全性レポートで使用)
isAP	BOOL		これがマネージャのアクセス・ポイントであることを示します
state	INT8U	モート・ステート	モートのステート
reserved	INT8U		予約済み。値は無視してください
isRouting	BOOL		このモートをネットワーク内で非リーフ・ノードとして使用できるかどうかを示します

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_NOT_FOUND	指定されたモートが存在しない
RC_END_OF_LIST	リスト内の最後のモートに達した (next = trueの場合)

4.11 getMoteConfigById(0x41)

説明

`getMoteConfigById` コマンドは、レスポンスとして単一モートの設定を返します。このコマンドは引数として、モートの短縮アドレス(モートID)を受け取ります。返されるレスポンスの構造は、[getMoteConfig](#) コマンドと同じです。

リクエスト

パラメータ	タイプ	一覧	説明
moteId	INT16U		モートID

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
macAddress	MAC_ADDR		モートのMACアドレス
moteId	INT16U		モートID(健全性レポートで使用)
isAP	BOOL		これがマネージャのアクセス・ポイントであることを示します
state	INT8U	モート・ステート	モートのステート
reserved	INT8U		予約済み。値は無視してください
isRouting	BOOL		このモートをネットワーク内で非リーフ・ノードとして使用できるかどうかを示します

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_NOT_FOUND	指定されたモートが存在しない

4.12 getMoteInfo(0x3E)

説明

getMoteInfoコマンドは、指定されたモートの動的な情報を返します。

リクエスト

パラメータ	タイプ	一覧	説明
macAddress	MAC_ADDR		モートのMACアドレス

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
macAddress	MAC_ADDR		モートのMACアドレス
state	INT8U	モート・ステート	モートのステート
numNbrs	INT8U		このモートの範囲内にあるモートの数(現在接続中のモートと接続可能性のあるモート両方)
numGoodNbrs	INT8U		このモートに対して良好なパス品質(> 50)を持つ近隣モートの数
requestedBw	INT32U		モートに要求された帯域幅(パケットあたりミリ秒)
totalNeededBw	INT32U		モートおよびその子モートに要求された総帯域幅(requestedBwを含む、パケットあたりミリ秒)
assignedBw	INT32U		現在割り当てられている帯域幅、パケットあたりミリ秒
packetsReceived	INT32U		マネージャがモートから受信したパケットの数
packetsLost	INT32U		モートから送信されたが、マネージャで紛失したパケットの数。紛失パケットの数はセキュリティ・カウンタを使用して計算されるため、パケット・ウィンドウのサイズごとにリアルタイムで追跡されます(紛失パケットは、WINDOW_SIZE分のパケットが追加で伝送されるまで記録されません)。
avgLatency	INT32U		パケットがモートで生成されてからマネージャに届くまでの平均時間(ミリ秒)
stateTime	INT32U		モート・ステートが最後に変ってから経過した時間(秒)。(Manager 1.3.0で追加)
numJoins	INT8U		このデバイスのこれまでの参加回数。このフィールドの最大値は255です。(Manager 1.4.1で追加)
hopDepth	INT8U		マネージャからのホップ数に10を掛けた数。ホップ数が25を上回る場合でも、このフィールドの値は255を超えません。(Manager 1.4.1で追加)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_NOT_FOUND	指定されたモートが存在しない

4.13 getMoteLinks (0x46)

説明

`getMoteLinks` コマンドは、モートに割り当てられたリンクの情報を返します。レスポンスには、モートに対するN番目のリンクから始まるリストが含まれます。ここでのNは、リクエスト内の`idx`パラメータとして指定されたインデックスです。デバイス上のすべてのリンクを取得するには、前のレスポンスで返されたリンクの数だけインクリメントした`idx`を指定してこのコマンドをコールし、`RC_END_OF_LIST` レスポンス・コードが返されるまで繰り返します。モートに割り当てられるリンクが、APIコール間で変わる場合があります。

リクエスト

パラメータ	タイプ	一覧	説明
macAddress	MAC_ADDR		モートのMACアドレス
idx	INT16U		返されるリンクの開始インデックス

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
idx	INT16U		返される最初のリンクの開始インデックス(ゼロベース)
utilization	INT8U		このモートに対するすべてのリンクの帯域幅使用状況(0~31)
numLinks	INT8U		このレスポンスに含まれるリンクの数
links	links[]		「numLinks」個のリンクを含むアレイ(構造については下の表を参照)

レスポンスで返されるリンクの構造

パラメータ	タイプ	一覧	説明
frameId	INT8U		フレームID
slot	INT32U		スロット番号
channelOffset	INT8U		チャンネル・オフセット
moteId	INT16U		ピア・モートID
flags	INT8U	リンク・フラグ	リンク・フラグ

レスポンス・コード

コード	説明
RC_NOT_FOUND	指定されたモートが存在しない
RC_INV_STATE	モートのステートがoperationalではない
RC_END_OF_LIST	要求されたインデックスが、リンク数を超えている

4.14 getNetworkConfig (0x3F)

説明

getNetworkConfigコマンドは、ネットワークID、帯域幅関連パラメータ、モート数などの、一般的なネットワーク設定パラメータを返します。

リクエスト

パラメータ	タイプ	一覧	説明
-------	-----	----	----

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
networkId	INT16U	なし	ネットワークID
apTxPower	INT8S	送信出力	アクセス・ポイントの送信出力
frameProfile	INT8U	フレーム・プロファイル	フレーム・プロファイルは、ネットワーク構築中と通常動作時のスロットフレームの長さです。レガシー・フレーム・プロファイルはすべて、標準IPマネージャ・プロファイルであるプロファイル1（高速構築、中速動作）にマッピングされます。
maxMotes	INT16U	なし	ネットワークで許容されるモートの最大数
baseBandwidth	INT16U	なし	ベース帯域幅は各参加モートに割り当てられるデフォルト帯域幅です。
downFrameMultVal	INT8U	なし	下りフレーム乗数は最初の下りスロットフレームの長さに対する乗数です。
numParents	INT8U	なし	各モートに割り当てられる親の数
ccaMode	INT8U	CCAモード	ネットワーク内のクリア・チャンネル・アセスメント(CCA)のモードを示します。
channelList	INT16U	なし	ホワイトリストに含まれ通信に使用されるチャンネルのビットマップ。その他はすべてブラックリストに含まれます。ビット0×0001がチャンネル0に相当し、ビット0×8000がチャンネル15に相当します(0 = 未使用、1 = 使用)。
autoStartNetwork	BOOL	なし	自動開始ネットワーク・フラグは、デバイスが起動され次第すぐにネットワークを開始するかどうかをマネージャに指示します(廃止済み)。
locMode	INT8U	なし	予約済み
bbMode	INT8U	バックボーン・フレーム・モード	バックボーン・フレーム・モード
bbSize	INT8U	なし	バックボーン・フレーム・サイズ
isRadioTest	INT8U	なし	マネージャが無線テスト・モードであるかどうかを示します。

パラメータ	タイプ	一覧	説明
bwMult	INT16U	なし	帯域幅プロビジョニング乗数(%, 100~1000)
oneChannel	INT8U	なし	ワン・チャンネル・モード用のチャンネル番号。 0xFF = ワン・チャンネル・モードはオフ

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

4.15 getNetworkInfo(0x40)

説明

getNetworkInfoコマンドは、動的なネットワーク情報と統計情報を返します。

リクエスト

パラメータ	タイプ	一覧	説明
-------	-----	----	----

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	説明
numMotes	INT16U		「Operational」ステートのモートの数(アクセス・ポイントは含まない)
asnSize	INT16U		ASNサイズはタイム・スロット時間(ミリ秒)です
advertisementState	INT8U	アドバタイズメント・ステート	アドバタイズメント・ステート
downFrameState	INT8U	下りフレーム・モード	現在の下りフレーム長に乗数が掛けられているかどうかを示します
netReliability	INT8U		ネットワークの信頼性(%)
netPathStability	INT8U		パスの安定性(%)
netLatency	INT32U		平均遅延(ミリ秒)
netState	INT8U	ネットワーク・ステート	現在のネットワーク・ステート
ipv6Address	IPV6_ADDR		システムのIPv6アドレス
numLostPackets	INT32U		紛失したパケットの数(Manager 1.3.0で追加)
numArrivedPackets	INT64U		受信したパケットの数(Manager 1.3.0で追加)
maxNumbHops	INT8U		ネットワーク内の最大ホップ数 * 10(Manager 1.3.0で追加)。モートの上りパケットが通過する平均ホップ数が、すべてのモートの中で最大となる場合の平均ホップ数。

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

4.16 getNextACLEntry(0x28)

説明

getNextACLEntryコマンドは、アクセス制御リスト(ACL)内の次のモート・エントリに関する情報を返します。検索を開始する(ACL内の最初のモートを見つける)には、MACアドレスとしてゼロ(0000000000000000)を送信します。

 特定のモートのACLエントリを読み取る方法はありません。このコマンドは繰り返し実行されます。
getNextACLEntryの引数としてモートAを渡すと、ACL内の次のモートであるモートBのACLエントリが返されます。

リクエスト

パラメータ	タイプ	一覧	説明
macAddress	MAC_ADDR		モートのMACアドレス

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
macAddress	MAC_ADDR		モートのMACアドレス
joinKey	SEC_KEY		参加鍵の読み取りは許可されていません。0が返されます。

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_END_OF_LIST	ACLの最後に達した
RC_NOT_FOUND	ACL内に指定されたモートが見つからない

4.17 getNextPathInfo(0x31)

説明

`getNextPathInfo` コマンドは、特定のモートに接続されたパスを横断して繰り返し処理を実行します。`pathId` パラメータは反復処理での前回の値を示します。`pathId` に 0 をセットすると最初のパスが返されます。`pathId` を一意のパス識別子として使用することはできません。この ID は特定のモートに関連付けられている間のみ有効です。

リクエスト

パラメータ	タイプ	一覧	説明
macAddress	MAC_ADDR		モートの MAC アドレス
filter	INT8U	パス・フィルタ	モートに接続されたすべてのパスに対して繰り返し処理を実行するか、または上りパスのみにするかを指定します。
pathId	INT16U		パス ID

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
pathId	INT16U		パス ID
source	MAC_ADDR		送信元モートの MAC アドレス
dest	MAC_ADDR		宛先モートの MAC アドレス
direction	INT8U	パスの方向	パスの方向
numLinks	INT8U		リンクの数
quality	INT8U		伝送パケット数に対する受信パケット数の移動平均に基づいて、パス品質を内部的に測定した値。この値には意味のある単位はありません。値が高いほど品質も高くなります。範囲は、0 (最低) から 100 (最高) です。使用パスの場合、品質はパス安定性に相当し、未使用パスの場合、RSSI に基づきます。
rssSrcDest	INT8S		送信元モートから宛先モートへのパスに対する最新の RSSI または 0 (データがない場合)。最初の健全性レポートの後でのみ、パスに対して値が計算されます。
rssDestSrc	INT8S		宛先モートから送信元モートへのパスに対する最新の RSSI または 0 (データがない場合)。最初の健全性レポートの後でのみ、パスに対して値が計算されます。

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_NOT_FOUND	指定されたパスIDが存在しない
RC_END_OF_LIST	リクエストで指定されたパスIDがリストの最後

4.18 getPathInfo(0x30)

説明

getPathInfoコマンドは、要求されたパスのパラメータを返します。

リクエスト

パラメータ	タイプ	一覧	説明
source	MAC_ADDR		送信元モートのMACアドレス
dest	MAC_ADDR		宛先モートのMACアドレス

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
source	MAC_ADDR		送信元モートのMACアドレス
dest	MAC_ADDR		宛先モートのMACアドレス
direction	INT8U	パスの方向	パスの方向
numLinks	INT8U		上りフレームに対するモート間のリンク数
quality	INT8U		伝送パケット数に対する受信パケット数の移動平均に基づいて、パス品質を内部的に測定した値。この値には意味のある単位はありません。値が高いほど品質も高くなります。範囲は、0(最低)から100(最高)です。使用パスの場合、品質はパス安定性に相当し、未使用パスの場合、RSSIに基づきます。
rssSrcDest	INT8S		送信元モートから宛先モートへのパスに対する最新のRSSIまたは0(データがない場合)。最初の健全性レポートの後でのみ、パスに対して値が計算されます。
rssDestSrc	INT8S		宛先モートから送信元モートへのパスに対する最新のRSSIまたは0(データがない場合)。最初の健全性レポートの後でのみ、パスに対して値が計算されます。

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_NOT_FOUND	指定されたモート間にパスが存在しない

4.19 getRadiotestStatistics (0x26)

説明

このコマンドは、以前に実行した`radiotestRx`コマンドから統計情報を取得します。マネージャが無線テスト・モードで起動されている場合のみ実行できます(`setNetworkConfig`コマンドを参照)。

リクエスト

パラメータ	タイプ	一覧	説明
-------	-----	----	----

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
rxOk	INT16U		正しく受信されたパケットの数
rxFail	INT16U		エラー付きで受信されたパケットの数

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_IN_PROGRESS	無線テストの進行中
RC_INVALID_COMMAND	無線テストが開始されていない

4.20 getSystemInfo (0x2E)

説明

getSystemInfoコマンドは、ハードウェアおよびソフトウェア・バージョンに関するシステム・レベルの情報を返します。

リクエスト

パラメータ	タイプ	一覧	説明
-------	-----	----	----

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
macAddress	MAC_ADDR		MACアドレス
hwModel	INT8U		ハードウェア・モデル
hwRev	INT8U		ハードウェア・リビジョン
swMajor	INT8U		ソフトウェア・バージョン (メジャー)
swMinor	INT8U		ソフトウェア・バージョン (マイナー)
swPatch	INT8U		ソフトウェア・バージョン (パッチ)
swBuild	INT16U		ソフトウェア・バージョン (ビルド)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

4.21 getTime(0x17)

説明

getTimeコマンドは、現在のマネージャのUTC時刻と現在の絶対スロット番号 (ASN) を返します。このコマンドから返される時刻の値は、キューイングとシリアル接続経由の伝送にかかる時間により遅延します。より高い精度を得るには、外部アプリケーションでTIMEピンを使用してnetworkTime通知をトリガする必要があります。

リクエスト

パラメータ	タイプ	一覧	説明
-------	-----	----	----

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
uptime	INT32U		レスポンスが生成された時刻 (秒、アップタイム)
utc	UTC_TIME_L		レスポンスが生成された時刻 (UTC)
asn	ASN		絶対スロット番号 (ASN)
asnOffset	INT16U		ASN内のオフセット (マイクロ秒)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

4.22 pingMote (0x2A)

説明

`pingMote` コマンドは、MACアドレスで指定されたモートにping (エコー・リクエスト) を送信します。一意の `callbackId` が生成され、レスポンスと共に返されます。モートからレスポンスを受信すると、マネージャは、測定したラウンド・トリップ遅延およびその他のパラメータと共に `pingResponse` 通知を生成します。リクエストはアクノリッジ対象外の伝送を使用して送信されるため、モートがリクエストを受信することは保証されません。

リクエスト

パラメータ	タイプ	一覧	説明
<code>macAddress</code>	<code>MAC_ADDR</code>		モートのMACアドレス

レスポンス

パラメータ	タイプ	一覧	説明
<code>rc</code>	<code>INT8U</code>	<code>レスポンス・コード</code>	レスポンス・コード
<code>callbackId</code>	<code>INT32U</code>		コールバックID

レスポンス・コード

コード	説明
<code>RC_OK</code>	コマンドが正しく完了した
<code>RC_NOT_FOUND</code>	指定されたモートが見つからない
<code>RC_INV_STATE</code>	モートのステータスが <code>operational</code> ではない
<code>RC_NO_RESOURCES</code>	ユーザ・コマンド・キューがフル状態
<code>RC_IN_PROGRESS</code>	指定されたモートに対する前回のエコー・リクエスト・コマンドが未処理

4.23 radiotestRx (0x25)

説明

`radiotestRx` コマンドは、それまでに収集したすべての統計情報をクリアし、指定されたチャンネルで無線受信を開始します。マネージャが無線テスト・モードで起動されている場合のみ実行できます (`setNetworkConfig` コマンドを参照)。テスト中、デバイスで受信したパケット数に関する統計情報は維持されます (エラーあり／なしの両方)。テスト結果を取得するには、`getRadiotestStatistics` コマンドを使用します。

i ステーションIDの値はユーザが選択できます。トランスミッタで使用されているステーションIDと同じ値を設定してください。ステーションIDは、同じ無線空間で複数のテストが実行されている場合に、トラフィックを分離するために使用します。

⚠ チャンネル番号は0～15で、IEEE 2.4GHzチャンネルの11～26に相当します。

リクエスト

パラメータ	タイプ	一覧	説明
mask	INT16U		テストで使用するRFチャンネルのマスク。1チャンネルのみを選択します。
duration	INT16U		テスト時間(秒)
stationId	INT8U		このデバイスの一意のステーションID (0～255)。送信側のステーションIDと一致しなければなりません。送信側のステーションIDを無視する場合、値に0を使用します。

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_IN_PROGRESS	無線テストの進行中
RC_INVALID_ARGUMENT	マスクの値が無効

4.24 radiotestTx(0x23)

説明

*radiotestTx*コマンドを実行すると、ユーザが無線伝送テストを開始できるようになります。マネージャが無線テスト・モードで起動されている場合のみ実行できます(*setNetworkConfig*コマンドを参照)。以下の4種類の伝送テストがサポートされています。

- パケット伝送
- 連続変調(CM)
- 連続波(CW、非変調信号)
- クリア・チャンネル・アセスメント(CCA)を有効にしたパケット伝送(Manager 1.3.x以降で使用可能)

パケット伝送テストでは、デバイスで`repeatCnt`個のパケット・シーケンスが生成されます。各シーケンスは最大10個のパケットで構成され、サイズと遅延を設定できます。それぞれのパケットは、PHYプリアンブル(5バイト)、PHY長フィールド(1バイト)、データ・ペイロード(最大125バイト)、802.15.4 CRC(2バイト)の順で構成されます。ペイロードのバイト0には、送信側のステーションIDが含まれます。バイト1および2にはパケット番号(ビッグエンディアン形式)が含まれ、この番号は伝送パケットごとにインクリメントされます。バイト3～Nにはカウンタ(0～N-2)が含まれ、このカウンタはペイロード内のバイトごとにインクリメントされます。伝送は、`chanMask`で定義され、疑似ランダムな順序で選択された一連のチャンネルで実行されます。

連続変調テストでは、指定されたチャンネルを中心に、連続した疑似ランダム変調信号がデバイスで生成されます。デバイスをリセットすると、テストが停止されます。

連続波テストでは、指定されたチャンネルを中心に、非変調トーンがデバイスで生成されます。デバイスをリセットすると、テスト・トーンが停止されます。

CCAを有効にしたパケット伝送テストでは、パケット伝送テストとまったく同じようにデバイスが設定されます。ただし、デバイスは各伝送の前にクリア・チャンネル・アセスメントを実行し、チャンネルがビジーの場合はパケットをアボートします。

 チャンネル番号は0～15で、IEEE 2.4GHzチャンネルの11～26に相当します。

 ステーションIDの値はユーザが選択できます。パケット・テストでこの値を使用することで、同じ無線空間で複数のテストが実行される可能性があるケースでも、受信側(*radiotestRx*を参照)がこのデバイスからのパケットを識別できます。このフィールドは、CMまたはCWテストでは使用しません(Manager 1.3.0以上で使用可能)。

リクエスト

パラメータ	タイプ	一覧	説明
testType	INT8U	無線テスト・タイプ	伝送テストの種類
chanMask	INT16U		テスト用に有効化されたチャンネル(0～15)のマスク。ビット0がチャンネル0に相当します。連続波および連続変調テストでは、1つのチャンネルのみを有効化してください。
repeatCnt	INT16U		パケット・シーケンスを繰り返す回数(0 = 中断なし)。パケット伝送テストのみに適用されます。

パラメータ	タイプ	一覧	説明
txPower	INT8S		送信出力 (dBm)。有効な値は、0および8です。
seqSize	INT8U		各シーケンスに含まれるパケットの数。このパラメータはパケット・テストのみで使用されます。
sequenceDef	seqDef[]		seqSizeシーケンス定義 (最大10) のアレイには、各パケットの長さとおケット伝送後の遅延を指定します。このパラメータはパケット・テストのみで使用されます。 各シーケンス定義の形式は以下のとおりです。 INT8U pkLen; /* パケットの長さ (2~125バイト) */ INT16U delay; /* パケット伝送後の遅延 (マイクロ秒) */
stationId	INT8U		パケットに含まれる一意 (1~255) の識別子で、送信側を識別します。このパラメータはパケット・テストのみで使用されます。(1.1.0以上のモードで使用可能)。

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_IN_PROGRESS	無線テストの進行中
RC_INVALID_ARGUMENT	「channel」または「txPower」の値が無効

4.25 reset(0x15)

説明

reset コマンドは各種のオブジェクトをリセットするために使用します。このコマンドの引数はオブジェクト・タイプです。オブジェクトがモートの場合、MACアドレスを指定する必要があります(それ以外の場合、引数は無視されます)。

リクエスト

パラメータ	タイプ	一覧	説明
type	INT8U	リセット・タイプ	リセットするオブジェクトのタイプ
macAddress	MAC_ADDR		モートのMACアドレス

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
macAddress	MAC_ADDR		モートのMACアドレス

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_NOT_FOUND	指定されたMACアドレスを持つモートが見つからない
RC_INV_STATE	モートのステートがoperationalではない
RC_NACK	ユーザ・コマンド・キューがフル状態(モート・リセットの場合)
RC_INVALID_ARGUMENT	リセット・タイプの値が無効

4.26 restoreFactoryDefaults (0x3D)

説明

`restoreFactoryDefaults` コマンドはデフォルト設定を復元し、ACLをクリアします。これは永続的な変更です。

Managerバージョンが1.3.0未満でライセンスが必要な場合、オプション機能の有効化に使用されるライセンスは、復元中も保持されます。

リクエスト

パラメータ	タイプ	一覧	説明
-------	-----	----	----

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_WRITE_FAIL	フラッシュ書き込みエラー、新しい設定を保存できない

4.27 sendData(0x2C)

説明

`sendData`コマンドはネットワーク内のモートにパケットを送信します。レスポンスには`callbackId`が含まれます。マネージャはネットワークにパケットを送出するとき、`packetSent`通知を生成します。顧客は、モートのアプリケーション層でレスポンスを送信する必要があります。また、予想されるレスポンスをマネージャが受信しない場合、顧客のアプリケーション層でタイムアウトを実行する必要があります。

`sendData`コマンドは、マネージャと直接通信するアプリケーションで使用します。エンドtoエンド(アプリケーションからモートまで)のIP接続が必要なアプリケーションでは、`sendIP`コマンドを使用してください。この違いについての詳細な説明は、SmartMesh IP Network User Guideを参照してください。

リクエスト

パラメータ	タイプ	一覧	説明
macAddress	MAC_ADDR		宛先モートのMACアドレス。すべてのモートにブロードキャストする場合、0xFFFFFFFFFFFFFFFFを使用します。
priority	INT8U	パケット優先順位	パケットの優先順位
srcPort	INT16U		送信元ポート
dstPort	INT16U		宛先ポート。これは、モート上の特定のサービスにパケットを伝送するときに使用します。
options	INT8U		optionsフィールドは将来的な使用のために予約されています。0をセットしてください。
data	INT8U[]		パケットのペイロード・データ。データ長はメッセージ全体の長さから計算されます。

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
callbackId	INT32U		コールバックID

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_NOT_FOUND	指定されたモートが見つからない
RC_INV_STATE	モートの状態がoperationalではない
RC_NACK	ユーザ・コマンド・キューがフル状態であるか、ペイロードにメモリ・バッファを割り当てられない
RC_INVALID_ARGUMENT	ペイロード・サイズが最大許容値を超えている

ペイロード・サイズの制限

送信元／宛先ポート	ペイロード・サイズ(バイト)
F0Bx	82
その他	79

4.28 sendIP (0x3B)

説明

`sendIP` コマンドはネットワーク内のモートに6LoWPANパケットを送信します。レスポンスには `callbackId` が含まれます。マネージャはネットワークにパケットを送出するとき、`packetSent` 通知を生成し、この `callbackId` を付加します。アプリケーション側で、有効な6LoWPANパケットを構築する必要があります。パケットはベスト・エフォートでモートに送信されるため、アプリケーション側でレスポンスとタイムアウトを処理する必要があります。

エンドtoエンドのIP接続が必要なアプリケーションでは、`sendIP` コマンドを使用する必要があります。エンドtoエンドのIP接続を必要としないアプリケーションでは、`sendData` コマンドが提供するインターフェースの方が簡単で、6LoWPANのカプセル化について理解する必要もありません。この違いについての詳細な説明は、[SmartMesh IP Network User Guide](#) を参照してください。

リクエスト

パラメータ	タイプ	一覧	説明
macAddress	MAC_ADDR		宛先モートのMACアドレス。すべてのモートにブロードキャストする場合、0xFFFFFFFFFFFFFFFFを使用します。
priority	INT8U	パケット優先順位	パケットの優先順位
options	INT8U		将来的な使用のために予約されています。optionsフィールドには0をセットしてください。
encryptedOffset	INT8U		データの暗号化部分のオフセット。0xFF - データは暗号化されていません。
data	INT8U[]		完全な6LoWPANパケット。データ・フィールド長はコマンド全体の長さから計算されます。

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
callbackId	INT32U		コールバックID

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_NOT_FOUND	指定されたモートが見つからない
RC_INV_STATE	モートの状態がoperationalではない
RC_NACK	ユーザ・コマンド・キューがフル状態であるか、ペイロードにメモリ・バッファを割り当てられない
RC_INVALID_ARGUMENT	ペイロード・サイズが最大許容値を超えているか、6LoWPANパケットが無効

ペイロード・サイズの制限

送信元／宛先ポート	ペイロード・サイズ(バイト)*
F0Bx	86
F0xx	84
その他	83

*アプリケーションで付加する必要のある6LoWPANヘッダによって、5バイトが使用されます

6LoWPANパケット形式

セクション	説明	長さ(バイト)	エンコーディング
6LoWPAN IP ヘッダ	LOWPAN_IPHC	2	011.11.1.10:C.S.ss.M.D.dd 011 - LOWPANヘッダを表します 11 - トラフィック・クラスおよびフロー・ラベルの省略 1 - 次のヘッダ・フィールドの圧縮 10 - ホップ・リミットの圧縮(上限64) - C - コンテキスト識別子(0 = 現在のバージョンのLBRでは省略) - S/D - 送信元／宛先の圧縮(1) - s/d - 送信元／宛先のモード: 00 - フル(16B - 外部送信元でのみ使用) 11 - 省略／圧縮(M = 1の場合、宛先のみ)(0/1B) - M - マルチキャスト(0)
	コンテキストIDの 拡張	0/1	SSSS.DDDD S、D - 送信元／宛先
	送信元アドレス	0/16	メッシュ送信元(マネージャなど)の場合は省略、送信元がメッシュ外の場合は16バイト
	宛先アドレス	0/1/16	メッシュ宛先(モートなど)の場合は省略、マルチキャストの場合は1バイト(現在のバージョンのLBRでは未使用)
UDP	ヘッダ	0/1	111101.S.D S/D - 送信元／宛先ポートの圧縮(1)

セクション	説明	長さ(バイト)	エンコーディング
	ポート	1/3/4	UDPヘッダのSDが11の場合、両ポートはF0Bxの範囲内にあり、ポートは1バイトのssss.ddddとしてエンコードされます s、d – 送信元／宛先 SD = 10の場合、送信元ポートはF0Bxの範囲内にあり、1バイトとしてエンコードされます。2バイトの宛先ポートはインラインで伝送されます SD = 01の場合、宛先ポートはF0xxの範囲内にあり、1バイトとしてエンコードされます。2バイトの送信元ポートはインラインで伝送されます SD = 00の場合、2バイトの各ポートはインラインで伝送されます
ペイロード	ユーザ・ペイロード	可変	

4.29 setACLEntry (0x27)

説明

setACLEntryコマンドは、アクセス制御リスト(ACL)に新規エントリを追加するか、リスト内で既存のエントリを更新します。これは永続的な変更です。エントリの最大数は1,200です。

リクエスト

パラメータ	タイプ	一覧	説明
macAddress	MAC_ADDR		モートのMACアドレス
joinKey	SEC_KEY		参加鍵

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_NO_RESOURCES	ACLがリストフル状態(新規エントリを追加する場合)
RC_WRITE_FAIL	フラッシュ書き込みエラー、新しい設定を保存できない

4.30 setAdvertising(0x32)

説明

setAdvertisingコマンドは、アドバタイズメントの有効化、無効化、または低速使用をマネージャに指示します。レスポンスはcallbackIdです。コマンドの伝搬が完了すると、callbackIdを含むcommandFinished通知が生成されます。

バージョン1.4.1より前のバージョンのモートでは、アドバタイズメントをオンまたはオフに切り替えることができます。主に1.4.1以降のモートで構成されたネットワークを形成する場合、アドバタイズメントを「低速」に設定すると電力を節約できることがあります。INIパラメータで、advtimeoutの値(ミリ秒)を設定し、このコマンドは0に設定します。

例えば、デフォルトのフル・アドバタイズメント頻度は約2秒に1回です。推奨値であるadvtimeout = 20000に設定すると、アドバタイズメントの頻度が20秒に1回になるため、アドバタイズメント・コストのうち90%の電力を節約できます。

 ネットワーク内のアドバタイズメントを無効にすることは危険です。アドバタイズメントを無効にした場合、新しいモートがネットワークに参加できず、リセット後にモートが再参加することもできません。アドバタイズメントの無効化は主に電力節約を目的としており、ネットワークにモートを参加させないようにするような特殊なユースケースでも役立つ場合がありますが、たいていの場合、アドバタイズメントはマネージャの制御下に置いたままにすることを推奨します。

リクエスト

パラメータ	タイプ	一覧	説明
activate	INT8U	アドバタイズメント・ステート	アドバタイズメント・ステート

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
callbackId	INT32U		コールバックID

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_IN_PROGRESS	コマンド保留中。リトライする前に、前のコマンドに対するcommandFinished通知を受信するまで待ちます。

4.31 setCLIUser(0x3A)

説明

setCLIUserコマンドは、特定のユーザ・ロールでコマンド・ラインにログインするためのパスワードを設定します。ユーザ・ロールは以下のとおりです。

- ビューア - 非機密情報への読み取り専用アクセス
- ユーザ - 読み取り／書き込みアクセス

これは永続的な変更です。

リクエスト

パラメータ	タイプ	一覧	説明
role	INT8U	CLIユーザ・ロール	ユーザ・ロール(上記を参照)
password	INT8U[16]		パスワード

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_WRITE_FAIL	フラッシュ書き込みエラー、新しい設定を保存できない

4.32 setCommonJoinKey(0x42)

説明

setCommonJoinKeyコマンドは、共通参加鍵に新しい値を設定します。共通参加鍵は、ACLが空の場合に限り、参加メッセージの解釈に使用されます。

リクエスト

パラメータ	タイプ	一覧	説明
key	SEC_KEY		共通参加鍵

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

4.33 setDownstreamFrameMode (0x33)

説明

setDownstreamFrameModeコマンドは、下りスロットフレームを短く、または長くするようにマネージャに要求します。「標準速度」の場合、ベースのスロットフレーム長にdownFrameMultValが掛けられます。「高速」の場合、下りスロットフレームの長さはベースの長さになります。いったんこのコマンドが実行されると、マネージャは手動モードに切り替わるため、この後にスロットフレーム・サイズが自動的に変更されることはありません。レスポンスはcallbackIdです。コマンドの伝搬が完了すると、callbackIdを含むcommandFinished通知が生成されます。

リクエスト

パラメータ	タイプ	一覧	説明
frameMode	INT8U	下りフレーム・モード	下りスロットフレーム・モード

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
callbackId	INT32U		コールバックID

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_IN_PROGRESS	コマンド保留中。リトライする前に、前のコマンドに対するcommandFinished通知を受信するまで待ちます。
RC_INVALID_ARGUMENT	downFrameMultVal (setNetworkConfigで設定) の値が1であるため、下りフレーム・モードの変更は無効です。

4.34 setIPConfig(0x44)

説明

setIPConfigコマンドは、メッシュ・ネットワークのIPv6プレフィックスを選択します。関係するのはIPv6アドレスの上位8バイトのみで、下位8バイトは無視されます。また、マスク・フィールドの下位8バイトは予約済みのため、0をセットする必要があります。これは永続的な変更です。

リクエスト

パラメータ	タイプ	一覧	説明
ipv6Address	IPV6_ADDR		IPv6アドレス
mask	INT8U[16]		サブネット・マスク

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_WRITE_FAIL	フラッシュ書き込みエラー、新しい設定を保存できない

4.35 setLicense (0x38)

説明

 *setLicense* コマンドは、Manager 1.3.0以降では廃止されています。モート数が32を上回るネットワークでのライセンス使用は不要になりました。

setLicense コマンドは、フラッシュ内に格納されたソフトウェアのライセンス・キーを検証して更新します。ライセンス・キーの変更による機能の有効化または無効化は、デバイスの再起動後に有効になります。*license* パラメータにすべて0x0がセットされた場合、マネージャはデフォルトのライセンスに戻します。これは永続的な変更です。

リクエスト

パラメータ	タイプ	一覧	説明
license	INT8U[13]		ソフトウェアのライセンス・キー

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_VALIDATION_ERROR	ライセンス・キーが有効ではない
RC_WRITE_FAIL	フラッシュ書き込みエラー、新しい設定を保存できない

4.36 setNetworkConfig (0x1A)

説明

setNetworkConfigコマンドは、ネットワーク設定パラメータを変更します。変更が正しく適用されたかどうかを、レスポンスコードが示します。これは永続的な変更です。

通常、ネットワーク設定の変更が有効になるのはマネージャの再起動後ですが、以下の例外があります。

- **最大モート数**: 新規モートがネットワークへの参加を試みると、すぐに新しいmaxMotesの値が使用されますが、numMotesよりも小さい数を設定した場合、モートがネットワークから削除されることはありません。

リクエスト

パラメータ	タイプ	一覧	説明
networkId	INT16U		ネットワークID
apTxPower	INT8S	送信出力	アクセス・ポイントの送信出力
frameProfile	INT8U	フレーム・プロファイル	フレーム・プロファイルは、ネットワーク構築中と通常動作時のフレームの長さです。レガシー・フレーム・プロファイルはすべて、標準IPマネージャ・プロファイルであるプロファイル1(高速構築、中速動作)にマッピングされます。
maxMotes	INT16U		ネットワークで許容されるモートの最大数(APを含む)。マネージャの製品番号とインストールされたライセンス(Managerバージョンが1.3.0より低い場合)によって、設定できる値は1~33または1~101のどちらかになります。
baseBandwidth	INT16U		ベース帯域幅は各参加モートに割り当てられるデフォルト帯域幅で、パケット間の予想間隔(ミリ秒)で定義されます。0 = 割り当てなし。
downFrameMultVal	INT8U		下りフレーム乗数は最初の下りフレームの長さに対する乗数です。有効な値は、1、2、4です。
numParents	INT8U		各モートに割り当てられる親の数(1~4)
ccaMode	INT8U	CCAモード	ネットワーク内のクリア・チャンネル・アセスメント(CCA)のモードを示します。
channelList	INT16U		ホワイトリストに含まれ通信に使用されるチャンネルのビットマップ。その他はすべてブラックリストに含まれます。ビット0×0001がチャンネル0に相当し、ビット0×8000がチャンネル15に相当します(0 = 未使用、1 = 使用)。チャンネル数の制限については、 SmartMesh IP User's Guide のChannel Blacklistingセクションを参照してください。
autoStartNetwork	BOOL		廃止済み - 使用しないでください
locMode	INT8U		予約済み
bbMode	INT8U	バックボーン・フレーム・モード	バックボーンのフレーム・モード(0 = オフ、1 = 上りのみ、2 = 双方向)
bbSize	INT8U		バックボーンのフレーム・サイズ(タイム・スロット数、bbmode = 1の場合、bbsize = 1、2、4、8。bbmode = 2の場合、bbsize = 2)
isRadioTest	INT8U		マネージャを無線テスト・モードで起動するかどうかを決めます。
bwMult	INT16U		帯域幅のオーバープロビジョニング乗数。値/100でオーバープロビジョニングします(100~1000)。

パラメータ	タイプ	一覧	説明
oneChannel	INT8U		ワン・チャンネル・モード用のチャンネル番号。(0～15、255 = オフ)。このモードはRFテストのみに使用されます

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_INVALID_ARGUMENT	ネットワーク・パラメータの検証に失敗した
RC_WRITE_FAIL	フラッシュ書き込みエラー、新しい設定を保存できない

4.37 setTime(0x36)

説明

 このコマンドは廃止されているため、新しい設計では使用しないでください。
Managerを再起動すると、2002年7月2日20:00:00(UTC)からカウントが開始されます。

`setTime`コマンドは、マネージャのUTC時刻を設定します。このコマンドを実行できるのは、ネットワークが稼働していないときだけです。`trigger`フラグが`false`の場合、マネージャは`setTime`コマンドの受信後すぐに指定された時刻を設定します。TIMEピン・トリガを受信した場合、現在の時刻を一時的に保存します。トリガの受信後の短い時間内に`setTime`リクエストを受信した場合、マネージャはトリガ後の遅延を計算し、指定された時間にトリガを受信したように時刻を調整します。

リクエスト

パラメータ	タイプ	一覧	説明
<code>trigger</code>	INT8U		0 = 即座に時刻を設定、1 = 直前のTIMEピン・トリガを使用して時刻を設定
<code>utcTime</code>	UTC_TIME_L		マネージャに設定する時刻(UTCマイクロ秒)

レスポンス

パラメータ	タイプ	一覧	説明
<code>rc</code>	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した。マネージャが時刻設定の準備を完了した
RC_INVALID_ARGUMENT	いずれかのパラメータが無効
RC_VALIDATION_ERROR	ネットワークが稼働中のため、 <code>setTime</code> コマンドは無効

4.38 startNetwork (0x2D)

説明

`startNetwork` コマンドは、マネージャに指示して、ネットワーク形成を開始できるようにします（デバイスからの参加リクエストの受け入れを開始する）。`autoStartNetwork` が設定されていない場合（[setNetworkConfig](#) を参照）、外部アプリケーションは `startNetwork` コマンドを実行する必要があります。

 このコマンドは廃止されているため、新しい設計では使用しないでください。

リクエスト

パラメータ	タイプ	一覧	説明
-------	-----	----	----

レスポンス

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_IN_PROGRESS	ネットワークは既に開始されている

4.39 subscribe (0x16)

説明

`subscribe` コマンドは、指定された通知を外部アプリケーションに送信するようにマネージャに指示します。このコマンドには以下の2つのフィルタ・フィールドが含まれます。

- `filter` はフラグのビットマスクで、クライアントが受信を要求する通知タイプを示します。
- `unackFilter` では、`filter` で指定した通知のうちのどれにアクノリッジが必要かをクライアントが指定できます。「アクノリッジ対象」として通知が送信されると、その後の通知パケットはレスポンスを待つ間、キューに追加されます。

サブスクリプション・リクエストは毎回、前回のリクエストを上書きします。アプリケーションでデータをサブスクライブしており、イベントもサブスクライブすることにした場合、データとイベントの両方のフラグをセットして、`subscribe` コマンドを送信する必要があります。すべてのサブスクリプションをクリアするには、フィルタに0をセットして、このコマンドを送信します。マネージャとクライアントの間にセッションが開始されると、サブスクリプション・フィルタがゼロに初期化されます。

`subscribe` ビットマップには、通知タイプの列挙値を使用します。一部の値は、以前のバージョンのAPIに対して後方互換性を提供するために未使用になっています。

リクエスト

パラメータ	タイプ	一覧	説明
<code>filter</code>	INT32U	サブスクリプション・フィルタ	サブスクリプション・ビットマップ
<code>unackFilter</code>	INT32U	サブスクリプション・フィルタ	<code>filter</code> で設定された通知ビットマップで、アクノリッジ不要な通信を行う通知タイプを示す。 <code>filter</code> ビットマップに対応するビットが設定されていない場合、 <code>unackFilter</code> 内のビットは無効になる

レスポンス

パラメータ	タイプ	一覧	説明
<code>rc</code>	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_INVALID_ARGUMENT	サブスクリプション・フィルタの値が無効

5 通知(0x14)

すべての通知パケットは、パケット・タイプに`Notification`が設定されています(パケット・タイプはシリアルAPIヘッダに含まれます)。通知の構造は、シリアルAPIパケットのペイロードに含まれます。各通知構造の最初にあるタイプ・フィールド([通知タイプ](#)を参照)により、通知の種類と残りのペイロードをデシリアライズする方法が示されます。

シリアルAPIヘッダ	通知タイプ、1バイト	通知構造、Nバイト(各通知の説明を参照)
------------	------------	----------------------

Controlフィールドで「アクノリッジ対象」と指定された通知に対しては、クライアントからアクノリッジを送信する必要があります。このプロパティの設定には、[subscribe](#)コマンドを使用します。認識できない「アクノリッジ対象」の通知タイプを受け取ったクライアントは、アクノリッジを送信して、通知を無視します。通知のアクノリッジには、アクノリッジ対象レスポンスであることを示す**Control**フィールド、`Notification`、パケット・タイプ、受信した通知と同じシーケンス番号、レスポンス・コードOKを含む1バイトのペイロードが含まれます。

シリアルAPIヘッダ	レスポンス・コードOK、1バイト = 0x00
------------	-------------------------

- [データ通知](#)には、モートによって送信された上りデータが含まれます。
- [IPデータ通知](#)には、モートによって送信された上り6LoWPANデータが含まれます。
- [イベント通知](#)には、マネージャによって送信されたネットワークまたはシステム通知が含まれます。
- [健全性レポート通知](#)には、モートからの健全性レポートが含まれます。
- [ログ通知](#)には、診断ログが含まれます。

5.1 ブリンク通知

マネージャによって生成されるブリンク通知は、標準のデータ通知です。ブリンク通知では、ユーザ・ペイロードは、下表に示すような、コマンドID、データ長、その他構造化されたデータ値からなるリスト内にカプセル化されます。

 ブリンク通知を送信できるのは、マネージャのACLに含まれるデバイスに対してのみです。

フィールド	タイプ	値	説明
command id	INT8U	0x94	ブリンク・ペイロードのコマンドID
length	INT8U		ユーザ・ペイロードの長さ
data	INT8U[]		ブリンク・コマンドに渡されたユーザ・ペイロード

検出された近隣モート・フラグに1をセットしてブリンク・コマンドをコールする場合、ブリンク・ペイロードに以下の構造も含まれます。

フィールド	タイプ	値	説明
command id	INT8U	0x95	検出された近隣モートの短縮コマンドID
length	INT8U		検出された近隣モート・データの長さ
numNeighbors	INT8U		検出された近隣モートの数
neighbors	dscv_neighbor[]		検出された近隣モートの短縮構造のリスト(下記を参照)

各近隣モートは、*dscv_neighbor*構造で表されます。

フィールド	タイプ	値	説明
moteld	INT16U		検出された近隣モートのモートID
rssi	INT8S		検出された近隣モートのRSSI

5.2 data通知

説明

*data*通知には、ヘッダとバイナリ・データの可変長アレイが含まれます。データの長さは、通知の長さに基づいて決定されます。マネージャのIPアドレスとマネージャ以外のポートで受信したパケットはすべて、*data*通知としてマネージャから転送されます。

通知の構造

フィールド	タイプ	一覧	説明
notifType	INT8U	通知タイプ	データ・ペイロード (<i>data</i>)
timestamp	UTC_TIME_L	なし	モートでパケットが生成された時刻
macAddress	MAC_ADDR	なし	生成モートのMACアドレス
srcPort	INT16U	なし	送信元ポート
dstPort	INT16U	なし	宛先ポート
data	INT8U[]	なし	データ・ペイロード

5.3 event通知

説明

イベントは、システム・レベルの変更とネットワーク・トポロジでの各種変更を示します。*event*通知を構成する要素は以下のとおりです。

- 一意のイベント識別子
- イベント・タイプ
- イベント・タイプに固有のイベント詳細構造

通知の構造

フィールド	タイプ	一覧	説明
notifType	INT8U	通知タイプ	通知タイプ (<i>event</i>)
eventId	INT32U	なし	イベントID
eventType	INT8U	イベント・タイプ	イベント・タイプ
eventData	----	なし	イベント・タイプに固有のイベント詳細構造

5.3.1 commandFinishedイベント

説明

`commandFinished`通知は、指定されたコールバックIDに関連付けられたコマンドの実行が終了したときに送信されます。

通知の構造

フィールド	タイプ	一覧	説明
callbackId	INT32U	なし	コールバックIDは、対応するコマンドのレスポンス・パケットの一部として返されます。
rc	INT8U	コマンド終了結果	コマンド終了結果コード

5.3.2 pathCreateイベント

説明

この通知は、マネージャが2つのモート間に接続(パス)を作成したときに送信されます。

通知の構造

フィールド	タイプ	一覧	説明
source	MAC_ADDR	なし	送信元モートのMACアドレス
dest	MAC_ADDR	なし	宛先モートのMACアドレス
direction	INT8U	パスの方向	パスの方向

5.3.3 pathDeleteイベント

説明

この通知は、マネージャが2つのモート間の接続(パス)を削除したときに送信されます。

通知の構造

フィールド	タイプ	一覧	説明
source	MAC_ADDR	なし	送信元モートのMACアドレス
dest	MAC_ADDR	なし	宛先モートのMACアドレス
direction	INT8U	パスの方向	パスの方向

5.3.4 pingイベント

説明

この通知は、モートへのpingの応答を受信したときに送信されます。

通知の構造

フィールド	タイプ	一覧	説明
callbackId	INT32U	なし	pingモート・リクエストに関連付けられたレスポンス・パケットの一部として返されたコールバックID
macAddress	MAC_ADDR	なし	pingを送信したモートのMACアドレス
delay	INT32U	なし	ラウンド・トリップ遅延(ミリ秒)、または-1 (pingタイムアウトの場合)
voltage	INT16U	なし	モートから報告された電圧(ミリボルト)
temperature	INT8S	なし	モートから報告された温度(摂氏)

5.3.5 networkTimeイベント

説明

*time*通知は、クライアントによるTIMEピンのアサートまたは*getTime*コマンドの呼び出しによってトリガされます。この通知には、TIMEピンがアサートされた(または*getTime*コマンドが処理された)時刻が含まれ、以下の形式で表されます。

- *ASN* - 絶対スロット番号(UTCがマネージャで設定されている場合は2002年7月2日午後8:00:00(PST)以降のタイム・スロット数、それ以外は1970年1月1日以降のタイム・スロット数)
- *Uptime* - デバイス起動後に経過した秒数
- *Unix time* - 1970年1月1日(UTC)以降の秒数およびマイクロ秒数

通知の構造

フィールド	タイプ	一覧	説明
uptime	INT32U	なし	パケットが生成された時刻(秒、アップタイム)
utcTime	UTC_TIME_L	なし	パケットが生成された時刻(UTC)
asn	ASN	なし	絶対スロット番号
asnOffset	INT16U	なし	ASNオフセット(マイクロ秒)

5.3.6 networkResetイベント

説明

この通知は、マネージャがネットワークを開始したときに送信されます。このイベントには`eventData`フィールドはありません。

通知の構造

フィールド	タイプ	一覧	説明
-------	-----	----	----

5.3.7 moteJoinイベント

説明

この通知は、モートがネットワークに参加したときに送信されます。

通知の構造

フィールド	タイプ	一覧	説明
macAddress	MAC_ADDR	なし	モートのMACアドレス

5.3.8 moteCreateイベント

説明

このイベントは、モートがはじめてマネージャに参加したときに送信されます。

通知の構造

フィールド	タイプ	一覧	説明
macAddress	MAC_ADDR	なし	モートのMACアドレス
moteId	INT16U	なし	モートID

5.3.9 moteDeleteイベント

説明

この通知は、*moteDelete*コマンドの結果としてモートが削除されたときに送信されます。

通知の構造

フィールド	タイプ	一覧	説明
macAddress	MAC_ADDR	なし	モートのMACアドレス
moteId	INT16U	なし	モートID

5.3.10 moteLostイベント

説明

この通知は、モートのステートがLostに変わったときに送信されます。このステートは、モートが下りメッセージに応答していないことを示します。

通知の構造

フィールド	タイプ	一覧	説明
macAddress	MAC_ADDR	なし	モートのMACアドレス



モートがネットワークに再参加したとき、*moteLost*イベントは生成されませんが、*motest*トレースがオンになっている場合、モートの参加リクエストを受信したときに、マネージャがLostから**Negotiating1**への遷移にマークを付けます。

5.3.11 moteOperationalイベント

説明

この通知は、ネットワークに参加したモートのステートが稼働中になったときに送信されます。

通知の構造

フィールド	タイプ	一覧	説明
macAddress	MAC_ADDR	なし	モートのMACアドレス

5.3.12 moteResetイベント

説明

この通知は、ユーザが開始したリセットをマネージャが実行したときに送信されます。

通知の構造

フィールド	タイプ	一覧	説明
macAddress	MAC_ADDR	なし	モートのMACアドレス

5.3.13 packetSentイベント

説明

*packetSent*通知は、クライアントのパケットがマネージャのキューから削除され、ワイヤレス・ネットワークに送信されたときに生成されます。

通知の構造

フィールド	タイプ	一覧	説明
callbackId	INT32U	なし	コールバックIDは、対応するコマンドのレスポンス・パケットの一部として返されます。
rc	INT8U	なし	結果コード

5.3.14 invalidMICイベント

説明

invalidMICイベントは、マネージャがネットワーク内のモートから受信したパケットの解読に失敗したときに生成されます。この通知は、バージョン1.4.1以上のManagerで使用できます。

通知の構造

フィールド	タイプ	一覧	説明
macAddress	MAC_ADDR	なし	モートのMACアドレス

5.3.15 joinFailedイベント

説明

joinFailedイベントは、モートがマネージャに参加リクエストを送信したが、リクエストが認められなかったときに生成されます。この通知は、バージョン1.4.1以上のManagerで使用できます。

通知の構造

フィールド	タイプ	一覧	説明
macAddress	MAC_ADDR	なし	モートのMACアドレス
reason	INT8U	参加失敗の理由	参加に失敗した理由

5.4 healthReport通知

説明

healthReport通知には、デバイスから受信した健全性レポートの未処理ペイロードが含まれます。このペイロードには、特定の健全性レポートのメッセージが1つ以上含まれます。各メッセージに、識別子、長さ、可変サイズのデータが含まれます。個々のhealthReportのメッセージ構造は下記のとおりです。

通知の構造

フィールド	タイプ	一覧	説明
notifType	INT8U	通知タイプ	通知タイプ (healthReport)
macAddress	MAC_ADDR	なし	健全性レポートを送信したモートのMACアドレス
payload	INT8U[]	なし	1つ以上の健全性レポート・メッセージの可変長ペイロード

5.4.1 健全性レポートのメッセージ

デバイス健全性レポート

デバイス健全性レポートは、前回のデバイス健全性レポート以降に蓄積されたデバイス統計情報のレポートです。

フィールド	タイプ	Value	説明
id	INT8U	0x80	デバイス健全性レポート識別子
length	INT8U	0x18	デバイス健全性レポートの残りの部分の長さ
charge	INT32U		ライフタイムの消費電気量 (mC)
queueOcc	INT8U		平均および最大キュー占有率。ビット0~3が平均キュー占有率で、ビット4~7が最大キュー占有率です。
temperature	INT8S		モートの温度 (摂氏)
batteryVoltage	INT16U		モートのバッテリー電圧 (mV)
numTxOk	INT16U		NETからMACに送信されたパケット数
numTxFail	INT16U		混雑またはパケット割り当て失敗により、送信されなかったパケット数
numRxOk	INT16U		受信したパケット数
numRxLost	INT16U		紛失した (様々なエラーによりNET層で破棄された) パケット数
numMacDropped	INT8U		MACによって破棄されたパケット数 (原因はリトライ回数、生存期間、ルートなしのいずれか)
numTxBad	INT8U		リンク不良の伝送失敗カウンタ
badLinkFrameId	INT8U		最後の健全性レポート期間で最も性能の低いリンクのフレームID
badLinkSlot	INT32U		最後の健全性レポート期間で最も性能の低いリンクのスロット
badLinkOffset	INT8U		最後の健全性レポート期間で最も性能の低いリンクのオフセット

フィールド	タイプ	Value	説明
numNetMicErr	INT8U		読解に失敗し、このモートで終了した受信パケットの数 注: モート・バージョン1.4.1以降
numMacMicErr	INT8U		認証に失敗した受信パケットの数 注: モート・バージョン1.4.1以降
numMacCrcErr	INT8U		MAC層のCRCエラーを含む受信パケットの数。このフィールドの目的は、ネットワークの妨げとなる異常なトラフィックや妨害の存在を示すことです。 注: モート・バージョン1.4.1以降

近隣モートの健全性レポート

近隣モートの健全性レポートには、モートと各近隣モートとの通信に関する現在の統計情報レポートが含まれます。

フィールド	タイプ	値	説明
id	INT8U	0x81	近隣モートの健全性レポート識別子
length	INT8U		残りのメッセージの長さ
numItems	INT8U		このメッセージに含まれるneighborHRData構造の数
neighbors	neighborHRData[]		numItems個のneighborHRData構造のシーケンス

neighborHRData構造

フィールド	タイプ	値	説明
neighborId	INT16U		近隣モートID
neighborFlag	INT8U		近隣モート・フラグ を参照
rssI	INT8S		近隣モートのRSSI
numTxPackets	INT16U		伝送されたパケット数
numTxFailures	INT16U		失敗した伝送の数
numRxPackets	INT16U		受信したパケット数

検出された近隣モート健全性レポート

検出された近隣モートの健全性レポートには、この健全性レポート間隔内に検出された近隣モートのリストが含まれます。

フィールド	タイプ	値	説明
id	INT8U	0x82	検出された近隣モート健全性レポート識別子
length	INT8U		残りのメッセージの長さ
numJoinParents	INT8U		親モートの数
numItems	INT8U		このメッセージに含まれる検出された近隣モート構造の数
discoveredNeighbors	discoveredNeighborData[]		numItems個のdiscoveredNeighborData構造のシーケンス

discoveredNeighborData構造

フィールド	タイプ	値	説明
neighborId	INT16U		近隣モートID
rssI	INT8S		近隣モートのRSSI
numRx	INT8U		近隣モートが検出された回数

拡張健全性レポート

一部の健全性レポートは、拡張健全性レポート通知として分類されます。これらの健全性レポートはマネージャによって処理されず、拡張健全性レポート通知のサブタイプとして、特別なペイロードが転送されます。拡張健全性レポートの一般的な形式は次のとおりです。

フィールド	タイプ	値	説明
id	INT8U	0x91	拡張健全性レポート識別子
length	INT8U		残りのメッセージの長さ
extType	INT8U		個別の拡張健全性レポート・タイプ
extLength	INT8U		個別の拡張健全性レポート・ペイロードの長さ
extPayload	INT8U[]		健全性レポートのペイロード

RSSIレポート

RSSIレポートには、RSSIと伝送統計情報を含むチャンネルのリストが含まれます。RSSIレポートを生成できるのは、バージョン1.4以上のIP Stackです。

フィールド	タイプ	値	説明
id	INT8U	0x91	拡張健全性レポート識別子
length	INT8U		残りのメッセージの長さ
extType	INT8U	1	拡張健全性レポート・タイプ、RSSIレポート
extLength	INT8U	75	RSSIレポート・ペイロードの長さ
rssIReport	rssIReport[15]		RSSIレポートのペイロード

rssIReportの構造

フィールド	タイプ	値	説明
idleRssi	INT8U		アイドル・リスニング中に測定された平均RSSI
txUnicastAttempts	INT16U		当該チャンネルでのユニキャスト試行回数
txUnicastFailures	INT16U		当該チャンネルでアクノリッジされなかった回数

5.5 ipData通知

説明

*ipData*通知には、モートから送信されたIPパケット全体(6LoWPANヘッダ、UDPヘッダ、UDPペイロードを含む)が含まれます。マネージャは、宛先にマネージャ自身のIPアドレス以外が指定されたパケットをモートから受信すると、この通知を生成します。*data*フィールドのサイズは、通知パケット全体のサイズから固定ヘッダ・サイズ(*macAddress*まで)を引くことで算出されます。

通知の構造

フィールド	タイプ	一覧	説明
notifType	INT8U	通知タイプ	通知タイプ(<i>ipData</i>)
utcTime	UTC_TIME_L	なし	モートでパケットが生成された時刻のUTCタイムスタンプ
macAddress	MAC_ADDR	なし	生成モートのMACアドレス
data	INT8U[]	なし	6LoWPANパケット

6LoWPANパケット形式

セクション	説明	長さ(バイト)	エンコーディング
6LoWPAN IP ヘッダ	LOWPAN_IPHC	2	011.11.1.10:C.S.ss.M.D.dd 011 - LOWPANヘッダを表します 11 - トラフィック・クラスおよびフロー・ラベルの省略 1 - 次のヘッダ・フィールドの圧縮 10 - ホップ・リミットの圧縮(上限64) - C - コンテキスト識別子(0 = 現在のバージョンのLBRでは省略) - S/D - 送信元/宛先の圧縮(1) - s/d - 送信元/宛先のモード: 00 - フル(16B -宛先がSmartMeshネットワーク外の場合のみ使用) 11 - 省略/圧縮(M = 1の場合、宛先のみ)(0/1B) - M - マルチキャスト(0 - マルチキャスト・アドレスは、現在のバージョンのLBRでは未使用)
	コンテキストIDの 拡張	0/1	SSSS.DDDD S、D - 送信元/宛先
	送信元アドレス	0/16	メッシュからのパケットは常に省略
	宛先アドレス	0/1/16	メッシュ内のデバイスは常に省略、宛先がメッシュ外の場合は16バイト

セクション	説明	長さ(バイト)	エンコーディング
UDP	ヘッダ	0/1	111101.S.D S/D - 送信元／宛先ポートの圧縮(1)
	ポート	1/3/4	UDPヘッダのSDが11の場合、両ポートはF0Bxの範囲内にあり、ポートは1バイトのssss.ddddとしてエンコードされます s、d - 送信元／宛先 SD = 10の場合、送信元ポートはF0Bxの範囲内にあり、1バイトとしてエンコードされます。2バイトの宛先ポートはインラインで伝送されます SD = 01の場合、宛先ポートはF0xxの範囲内にあり、1バイトとしてエンコードされます。2バイトの送信元ポートはインラインで伝送されます SD = 00の場合、2バイトの各ポートはインラインで伝送されます
ペイロード	ユーザ・ペイロード	可変	

5.6 log通知

説明

log通知は、[getLog](#)コマンドに対する応答として生成されます。それぞれのlog通知に、モートのログからのメッセージが含まれます。

通知の構造

フィールド	タイプ	一覧	説明
notifType	INT8U	通知タイプ	通知タイプ(log)
macAddress	MAC_ADDR	なし	通知元のMACアドレス
logMsg	INT8U[]	なし	ログ・メッセージ

6 定義

このセクションでは、API構造で使用する定数および事前定義値の一覧を示します。

6.1 パケットおよびコマンド・タイプ

それぞれのメッセージにパケット・タイプが割り当てられます。最初のいくつかのパケット・タイプは、シリアルAPIプロトコルのハンドシェイクで使用されます。指定のないパケット・タイプは、互換性または将来的な使用のために予約されています。

名前	値	説明
(Null)	0x00	Nullパケット(予約済み)
Hello	0x01	クライアントのHelloメッセージ
HelloResponse	0x02	クライアントのHelloに対するマネージャのレスポンス
MgrHello	0x03	マネージャのHelloメッセージ
Notification:	0x14	通知パケット
reset	0x15	システム、ネットワーク、またはモートをリセットします
subscribe	0x16	通知をサブスクライブします
getTime	0x17	現在のマネージャのUTC時刻と絶対スロット番号(ASN)を返します
setNetworkConfig	0x1A	ネットワーク設定パラメータをセットします
clearStatistics	0x1F	蓄積されたネットワーク統計情報をクリアします
exchangeMoteJoinKey	0x21	新しい参加鍵をモートに送信します
exchangeNetworkId	0x22	新しいネットワークIDをモートに送信します
radiotestTx	0x23	無線テスト用のコマンド
radiotestRx	0x25	無線テスト用のコマンド
getRadiotestStatistics	0x26	無線テスト用のコマンド
setACLEntry	0x27	新しいACLエントリの追加、または既存エントリの更新
getNextACLEntry	0x28	ACLエントリの繰り返し処理
deleteACLEntry	0x29	ACLからモートを削除します
pingMote	0x2A	ping(エコー・リクエスト)をモートに送信します
getLog	0x2B	モートから診断ログを取得します
sendData	0x2C	データ・パケットをモートに送信します
startNetwork	0x2D	ネットワーク形成を開始します(廃止済み - 使用しないでください)
getSystemInfo	0x2E	ハードウェアおよびソフトウェアに関するシステム・レベルの情報を返します
getMoteConfig	0x2F	モート設定パラメータを取得します
getPathInfo	0x30	2つのモート間の通信に関する情報を取得します
getNextPathInfo	0x31	モートの近隣モートに対する繰り返し処理

名前	値	説明
setAdvertising	0x32	アドバタイズメントを有効化または無効化します
setDownstreamFrameMode	0x33	下リスロットフレームを短縮または伸長します
reserved	0x34	reserved
getManagerStatistics	0x35	マネージャAPIの統計情報を取得します
setTime	0x36	マネージャのUTC時刻を設定します
getLicense	0x37	現在のライセンス・キーを返します(1.3.0以上のManagerでは廃止)
setLicense	0x38	ソフトウェア・ライセンス・キーを更新します(1.3.0以上のManagerでは廃止)
setCLIUser	0x3A	CLIログインを更新します
sendIP	0x3B	モートにIPデータを送信します(6LowPAN使用)
reserved	0x3C	予約済み
restoreFactoryDefaults	0x3D	デフォルト設定を復元し、ACLをクリアします
getMoteInfo	0x3E	モート統計情報を取得します
getNetworkConfig	0x3F	ネットワーク設定パラメータを取得します
getNetworkInfo	0x40	ネットワーク統計情報を取得します
getMoteConfigByld	0x41	モートIDで指定されたモートの設定を取得します
setCommonJoinKey	0x42	共通参加鍵に新しい値を設定します
getIPConfig	0x43	マネージャのIP設定パラメータを返します
setIPConfig	0x44	マネージャのIP設定パラメータを設定します
deleteMote	0x45	Managerのデータベースからモートを削除します
getMoteLinks	0x46	モートのリンクに関する情報を取得します

6.2 通知タイプ

名前	値	説明
event	1	イベント通知
log	2	ログ通知
data	4	データ・ペイロード通知
ipData	5	6lowpanパケット通知
healthReport	6	健全性レポート通知



マネージャAPIには通知コマンド・タイプが1つ(0x14)しかなく、最初の通知フィールドで通知の種類を識別します。モートAPIでは、各通知に固有のコマンド・タイプがあります。イベント通知の最初のフィールドで、イベントの種類を識別します。

6.3 サブスクリプション・フィルタ

フィルタ	値
イベント	0x02
ログ	0x04
データ	0x10
IPデータ	0x20
健全性レポート	0x40

6.4 イベント・タイプ

名前	値	説明
moteReset	0	モートのリセット
networkReset	1	ネットワークがリセットされた
commandFinished	2	コマンドの実行が完了した
moteJoin	3	モートがネットワークに参加した
moteOperational	4	新しいモートが設定され、現在稼働中である
moteLost	5	ネットワーク内でモートが通信を停止した
networkTime	6	ネットワーク・アップタイムを含む(getTime コマンドに対する応答として)
pingResponse	7	モートへのpingから応答を受信した
reserved	9	予約済み
pathCreate	10	パスが作成された
pathDelete	11	パスが削除された
packetSent	12	パケットが送信された
moteCreate	13	モートが作成された
moteDelete	14	モートが削除された
joinFailed	15	参加リクエストを処理できなかった
invalidMIC	16	モートからのパケットの解読に失敗した

6.5 レスポンス・コード

名前	値	説明
RC_OK	0	アプリケーション層でコマンドが正しく処理された
RC_INVALID_COMMAND	1	無効なコマンド
RC_INVALID_ARGUMENT	2	無効な引数
RC_END_OF_LIST	11	イタレーションが完了したところで「End of list」を返す
RC_NO_RESOURCES	12	最大アイテム数に達した

名前	値	説明
RC_IN_PROGRESS	13	処理の進行中
RC_NACK	14	否定のアクノリッジ
RC_WRITE_FAIL	15	フラッシュの書き込みに失敗した
RC_VALIDATION_ERROR	16	パラメータの検証エラー
RC_INV_STATE	17	オブジェクトのステートが不適切
RC_NOT_FOUND	18	オブジェクトが見つからない
RC_UNSUPPORTED	19	操作がサポートされていない

6.6 フレーム・プロファイル

名前	値	説明
Profile_01	1	高速ネットワーク構築、中速ネットワーク動作

6.7 アドバタイズメント・ステート

名前	値	説明
on	0	アドバタイズメントは有効
off	1	アドバタイズメントは無効または低速*

* 低速アドバタイズメントは、1.4.1以上のモートで使用可能です。アドバタイズメント速度は、INIパラメータadvtimeoutで設定します。

6.8 下りフレーム・モード

名前	値	説明
normal	0	標準の下り帯域幅
fast	1	高速の下り帯域幅

6.9 ネットワーク・ステート

名前	値	説明
operational	0	ネットワークは正常に稼働している
radiotest	1	マネージャは無線テスト・モードになっている
notStarted	2	startNetwork APIコマンドを待機中
errorStartup	3	起動時に予期しないエラーが発生した
errorConfig	4	起動時に無効な(またはライセンスのない)設定が見つかった
errorLicense	5	起動時に無効なライセンス・ファイルが見つかった

6.10 モート・ステート

名前	値	説明
lost	0	モートは現在ネットワークに参加していない
negotiating	1	モートがネットワークに参加中
operational	4	モートは稼働中である

6.11 リセット・タイプ

名前	値	説明
resetSystem	0	システムのリセット
resetMote	2	モートのリセット

6.12 バックボーン・フレーム・モード

名前	値	説明
off	0	バックボーン・フレームは無効
upstream	1	バックボーン・フレームは上りフレームに対して有効
bidirectional	2	バックボーン・フレームは上りフレームと下りフレームの両方に対して有効

6.13 パス・フィルタ

名前	値	説明
all	0	すべてのパス
upstream	1	上りパス

6.14 パスの方向

名前	値	説明
なし	0	パスなし
unused	1	パスは未使用
upstream	2	上りパス
downstream	3	下りパス

6.15 パケット優先順位

名前	値	説明
Low	0	デフォルトのパケット優先順位
Medium	1	より高いパケット優先順位
High	2	最も高いパケット優先順位

6.16 コマンド終了結果

名前	値	説明
OK	0	コマンドが正しく完了した
nack	1	コマンドがアクノリッジされていない
commandTimeout	2	コマンドがタイムアウトした

6.17 送信出力

送信出力は符号付きバイト(INT8S)で、値は0または8です。

6.18 CCAモード

名前	値	説明
off	0	CCAが無効
energy	1	電力の検出
carrier	2	搬送波の検出
both	3	電力の検出および搬送波の検出

6.19 リンク・フラグ

フラグ	値
送信	0x01
受信	0x02
共有	0x04
予約済み	0x08
参加	0x10
アドバタイズメント	0x20
検出	0x40
パス障害の検出なし	0x80

6.20 近隣モート・フラグ

フラグ	値
既存パスの障害状況	0x01

6.21 CLIユーザ・ロール

名前	値	説明
viewer	0	ビューア・ロールのユーザには、非機密情報への読み取り専用アクセスが割り当てられます
user	1	ユーザ・ロールのユーザには、読み取り／書き込み権限が割り当てられます

6.22 無線テスト・タイプ

名前	値	説明
packet	0	パケット伝送
cm	1	連続変調
cw	2	連続波
pkcca	3	クリア・チャンネル・アセスメント(CCA)を有効にしたパケット・テスト

6.23 参加失敗の理由

名前	値	説明
counter	0	参加パケットで、既に使用済みの参加カウンタが再利用された
notOnACL	1	モートがACLに含まれていない
authentication	2	参加リクエストを解読できなかった。これは通常、リクエストの暗号化に使用された参加鍵が、ACL内の鍵と一致しない場合発生
unexpected	3	参加リクエストの処理中に、予期しないエラーが発生した

商標

Eterna、Mote-on-Chip、SmartMesh IPは、Dust Networks, Incの商標です。Dust Networksロゴ、Dust、Dust Networks、SmartMeshは、Dust Networks, Incの登録商標です。LT、LTC、LTM、 は、アナログ・デバイセズの登録商標です。第三者のブランド名および製品名は各社の商標であり、情報提供のみを目的として使用されています。

著作権

本書は、米国著作権法、国際著作権法、その他の知的財産法および産業財産法によって保護されています。本書はアナログ・デバイセズおよびその実施許諾者によって専有されており、制限付きライセンスに従って配布されます。アナログ・デバイセズの書面による事前の認可なく、本書の全部または一部を使用、複製、変更、逆アセンブル、逆コンパイル、リバース・エンジニアリング、配布、再配布することは、その形式、手段にかかわらず禁じられています。

制限付き権利: 米国政府による使用、複製、開示は、FAR 52.227-14(g) (2) (6/87) および FAR 52.227-19(6/87)、または DFAR 252.227-7015(b) (6/95) および DFAR 227.7202-3(a) ならびにこれに準ずる法律および規制と後継の法律および規制に規定された制限の対象となります。

免責事項

本書は現状のまま提供され、明示、暗示を問わず一切の保証を行わないものとします。かかる保証には、特定目的に対する商品性または適合性の黙示的保証が含まれますが、これに限定されません。

本書には技術的な誤りやその他の間違いが含まれる場合があります。訂正と改善は、新しいバージョンの文書に取り入れられる可能性があります。

アナログ・デバイセズは、製品やサービスの適用または使用により発生する責任を負いかねます。また、間接的あるいは偶発的損害を含むがそれに限定されない、いかなる責任も負わないものとします。

アナログ・デバイセズの製品は、誤動作がユーザの深刻な人身傷害につながると合理的に予想できる生命維持装置、デバイス、またはその他のシステムでの使用、またはその機能不全により生命維持装置またはシステムの故障あるいはその安全性や有効性に影響すると合理的に予想できる生命維持装置またはシステムの重要な部品としての使用を目的として設計されていません。このような用途での使用を目的としてこれらの製品を使用または販売しているアナログ・デバイセズの顧客は、顧客自身の責任でそれを行い、このような意図しないまたは不正な使用に関連する人身傷害または死亡に直接または間接的に起因するすべての主張、費用、損害、支出、および妥当な額の弁護士費用、また、かかるクレームでアナログ・デバイセズに該当製品の設計または製造に関わる過失があったと主張される場合でも、これを完全に補償し、アナログ・デバイセズとその役員、従業員、子会社、関連会社、および販売代理店に何ら損害を与えないことに同意するものとします。

アナログ・デバイセズは、いつでも製品またはサービスに対する修正、変更、拡張、改良、その他の変更を行う権利を保有し、製品またはサービスを予告なく中止する権利を有します。顧客は、発注の前に最新の関連情報を入手し、その情報が最新で完全であることを確認する必要があります。すべての製品は、注文承諾時または販売時に提供される、販売に関するDust Networkの契約条件に従い販売されます。

アナログ・デバイセズは、アナログ・デバイセズの製品またはサービスが使用される組み合わせ、マシン、またはプロセスに関連するアナログ・デバイセズの特許、著作権、マスクワーク権、その他のアナログ・デバイセズの知的所有権に従って、明示か黙示かにかかわらず、ライセンスが付与されることを保証または主張するものではありません。第三者の製品またはサービスに関してアナログ・デバイセズが公開した情報は、その製品またはサービスを使用するためのアナログ・デバイセズからのライセンス提供、あるいはその保証または推奨を意味するものではありません。このような情報を使用する場合、第三者の特許または他の知的所有権に従って第三者からのライセンスが必要になるか、またはアナログ・デバイセズの特許または他の知的所有権に従ってアナログ・デバイセズからのライセンスが必要になります。

Dust Networks, Incは、アナログ・デバイセズの完全所有子会社です。

© Analog Devices, Inc. 2012-2016 All Rights Reserved.