

SmartMesh IPユーザー・ガイド

1 目次

1	本書について.....	4
1.1	関連文書.....	4
1.2	使用される表記規則	6
1.3	改訂履歴.....	6
2	SmartMesh用語集.....	7
3	SmartMesh IPネットワーク.....	11
3.1	はじめに	11
3.1.1	ネットワークの概要.....	11
3.1.2	SmartMesh IPについて.....	13
3.2	ネットワーク形成	15
3.3	帯域幅とレイテンシ	16
3.3.1	NumParentsパラメータ.....	16
3.3.2	BWMultパラメータ.....	16
3.3.3	BaseBWパラメータ.....	17
3.3.4	サービス	17
3.3.5	カスケード・リンク.....	17
3.3.6	低レイテンシのバックボーン	18
3.3.7	ノーマル/高速ダウンストリーム・モード.....	18
3.4	データ・トラフィック	19
3.4.1	シリアル接続の「ゲートウェイ」アプリケーション	19
3.4.2	エンド・ツー・エンドIP.....	20
3.5	ネットワーク・セキュリティ.....	21
3.5.1	セキュリティ層	21
3.5.2	セキュリティ・モード.....	22
4	SmartMesh IPマネージャ.....	23
4.1	はじめに	23
4.1.1	マネージャ・クライアント・アプリケーション設計の手順.....	23
4.2	マネージャへのアクセス	24
4.2.1	コマンド・ライン・インタフェース (CLI)	24
4.2.2	アプリケーション・プログラミング・インタフェース (API)	24
4.3	設定と使用法	25
4.3.1	ネットワークID.....	25
4.3.2	ネットワーク時間.....	25
4.3.3	モートとの通信	26
4.3.4	ライセンス.....	26
4.4	ネットワーク・アクティビティ	27
4.4.1	ネットワークの構造と形成.....	27
4.4.2	ネットワークの健全性	27
4.4.3	健全性レポート	29
4.4.4	最適化	29
4.5	アクセス制御.....	29
4.5.1	ACL管理.....	29
4.6	無線プログラミング (OTAP)	30
4.7	マネージャの工場出荷時デフォルト設定の復元	31

5	SmartMesh IP モート	32
5.1	はじめに	32
5.1.1	モート設計の手順	32
5.2	モート状態マシン	33
5.3	参加	34
5.3.1	プログラムからの参加	34
5.3.2	自動参加	35
5.3.3	隣接ネットワークへの参加	36
5.4	サービス	36
5.4.1	帯域幅の要求	36
5.4.2	バック・オフのメカニズム	37
5.5	通信	38
5.5.1	ソケット	38
5.5.2	UDPポート割り当て	38
5.5.3	データの送信と受信	39
5.6	イベントとアラーム	39
5.7	工場出荷時のデフォルト設定	39
5.8	電力に関する考察	40
5.8.1	電源情報	40
5.8.2	ルーティング・モード	40
5.8.3	ジョイン・デューティ・サイクル	41
5.9	マスタとスレーブ	41
5.9.1	モード動作	41

1 本書について

1.1 関連文書

SmartMesh IP ネットワーク向けに、以下の文書が用意されています。

Starter Kit のクイック・ガイド

- [SmartMesh IP Easy Start Guide](#)—基本的なインストールとネットワークが稼動していることを確認するために必要なテストをいくつか説明します。
- [SmartMesh IP Tools Guide](#)—「インストール」のセクションには、シリアル・ドライバのインストールについての説明、および Easy Start Guide や他のチュートリアルで使用されるプログラム例が記載されています。

ユーザー・ガイド

- [SmartMesh IP ユーザー・ガイド](#)—ネットワークの概念について説明し、モートおよびマネージャ API を活用してデータ送信や統計データの収集などを実施する方法について説明します。本書では API ガイドを使用するために必要な予備知識を提供します。

デバイス間の対話用インタフェース

- [SmartMesh IP Manager CLI Guide](#)—本書は、クライアント・ソフトウェアの開発やトラブルシューティングの際に、マネージャをコマンドラインで制御するための CLI とそのコマンド・セットについて説明します。
- [SmartMesh IP Manager API Guide](#)—本書は、ホスト・アプリケーションからマネージャを制御するための API とそのコマンド・セットについて説明します。
- [SmartMesh IP Mote CLI Guide](#)—本書は、センサ・アプリケーションの開発やトラブルシューティングの際にモートをコマンドラインで制御するための CLI とそのコマンド・セットについて説明します。
- [SmartMesh IP Mote API Guide](#)—本書は、センサ・アプリケーションからモートを制御するための API とそのコマンド・セットについて説明します。

ソフトウェア開発ツール

- [SmartMesh IP Tools Guide](#)—[SmartMesh SDK](#)に含まれる開発支援ツールについて説明します。これには、モート API やマネージャ API の実行およびネットワークの視覚化のためのツールが含まれます。

アプリケーション・ノート

- [SmartMesh IP Application Notes](#)—SmartMesh IP ネットワークの構築に関するトピックおよび一般的に SmartMesh ネットワークに関連したトピックについて説明します。

新規設計に役立つ文書

- [LTC5800-IPM SoC](#) およびそれに基づく [モジュール](#) のデータシート。
- [LTC5800-IPR SoC](#) およびそれに基づく [埋め込みマネージャ](#) のデータシート。
- モート/マネージャ SoC または [モジュール](#) 用 [Hardware Integration Guide](#)—SoC またはモジュールを組み込んだ設計を行う際のベスト・プラクティスについて説明します。

- 埋め込みマネージャ用 [Hardware Integration Guide](#)—埋め込みマネージャを組み込んだ設計を行う際のベスト・プラクティスについて説明します。
- [Board Specific Integration Guide](#)—SoC モードおよびマネージャ用に、デフォルトの IO 構成および「ヒューズ表」を用いた水晶振動子調整情報の設定方法について説明します。
- [Hardware Integration Application Notes](#)—SoC 設計チェック・リスト、アンテナ選択ガイドなどが含まれます。
- [ESP Programmer Guide](#)—デバイス上でファームウェアをプログラムするために使用される DC9010 Programmer Board および ESP ソフトウェアのガイド。
- ESP ソフトウェア—モードまたはモジュールにファームウェア・イメージを書き込むために使用されます。
- Fuse Table ソフトウェア—Board Specific Integration Guide で説明されているヒューズ表を構築するために使用されます。

その他の役立つ文書

- SmartMesh 文書で使用する無線ネットワークの用語集は、[SmartMesh IP ユーザー・ガイド](#)にあります。
- [よくある質問](#)の一覧

1.2 使用される表記規則

本書では、以下の表記規則が使用されます。

コンピュータ・タイプは、URL の指定など、ユーザー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、およびデバイスの状態とモードを示します。

*斜体*は、新しい用語の紹介およびAPIとそのパラメータへの参照のために使用されます。

 ヒントは、製品に関する有用な情報を提供します。

 情報テキストは、背景およびコンテキストに関する追加情報を提供します。

 注意は、概念に関するより詳細な情報を提供します。

 **警告!**警告は、データの損失やハードウェアまたは人体に物理的損傷を引き起こす可能性がある行動について通知します。

コード・ブロックは、コード例を表示します。

1.3 改訂履歴

リビジョン	日付	説明
1	07/17/2012	初期リリース
2	03/18/2013	多数の小規模な変更

2 SmartMesh 用語集

アクセス・ポイント (または AP) — 無線ネットワークと有線ネットワークの橋渡しをするデバイス。無線 MAC パケットから有線 Net パケットへの変換、またはその逆を行います。

確認応答 (または ACK) — パケットの受信に対して送信されるフレーム。そのパケットが正しく受信されたことを確認します。ACK には、パケット受信者と送信者の時間差が含まれます。

通知 — 他のデバイスがネットワークと同期できるようにするために送信され、新しいモートが参加するために必要なリンク情報を含むフレーム。

ASN — 絶対スロット番号 (Absolute Slot Number)。ネットワークの開始 (WirelessHART)、または 2002 年 7 月 2 日 20:00:00 UTC (ネットワークを開始する前に UTC が設定される場合は IP) から経過したタイムスロットの数。

認証 — 受信したパケットが改変されていないこと、および要求元のネット層の送信者から発信されたものであることを確認する暗号プロセス。

バックボーン — モートが高速スーパーフレームを共有して、低レイテンシ・アラーム・トラフィックを利用できるようにする SmartMesh ネットワークの機能。

帯域幅 — データを送信するためのモートのデータ転送容量。通常はパケット/秒単位。

ベース帯域幅 — ネットワーク内の各モートが、サービスを要求しなくても予め得られる帯域幅。

CCM* — CBC-MAC 付きカウンタ・モード (Counter mode with CBC-MAC)。Dust 製品で使用する認証/暗号化方式。詳細は、アプリケーション・ノート「SmartMesh Security」を参照してください。

チャンネル — PHY が使用する中心周波数のリストのインデックス。802.15.4 では、2.4 ~ 2.4835GHz の産業科学医学用 (ISM) バンドに 16 チャンネルあります。

チャンネル・ホッピング — スロット間のチャンネル変更。Bluetooth など、メッセージ内でチャンネルを変更する PHY のホッピングと区別するために「スロー・ホップ」とも呼ばれます。

チャンネル・オフセット — この ASN で使用するチャンネルを選択するためにチャンネル計算機能で使用するリンク固有の数値。

子 — 他のモートから時間情報を受信するデバイスはそのモートの子です。子は、その親を通してデータを転送します。

CSMA — 搬送波感知多重アクセス (Carrier Sense Multiple Access)。非同期の送信機が、送信を行う前に他の送信機が送信しているかどうかを事前に感知する通信アーキテクチャ。

コミッショニング — 一般的に、ネットワーク ID、ジョイン・キー、およびその他の結合パラメータを設定することにより、ネットワークへの参加に向けてモートを起動すること。

探索 — モートが近隣にあるモート、あるいはアクセス・ポイントを探し、その情報をマネージャに報告するプロセス。

ダウンストリーム — マネージャまたは有線側のアプリケーションからメッシュ・ネットワークへの方向。

暗号化 — 対象の受信者だけがペイロード情報を解読できるように、ランダム・ノイズと区別ができない形式に変換する暗号プロセス。

フレーム (またはパケット) —PHY 層により送信される情報。通常は、ホップごと (MAC) やエンド・ツー・エンド (Net) の層のヘッダ、およびペイロードを含みます。

ゲートウェイ—WirelessHART メッシュ・ネットワークから有線 HART データ・モデルを抽出する WirelessHART ネットワーク内のデバイス。

グラフ—各パケットの送信先をモートに知らせる WirelessHART ネットおよび IP メッシュ層ヘッダに含まれる、番号付きルーティング要素。スーパーフレームとスロットフレームにもグラフ ID があります。

グラフ・ルート—グラフ ID のみがパケット・ヘッダに指定されているルート。パケットは、複数のモートを通してその着信先までグラフ・ルートを辿ります。ソース・ルートと比較してください。

ガード時間—デバイスの同期が不完全なモート間での通信を可能にするために、予想されるパケット到着時間の前または後で、デバイスが受信する時間。

健全性レポート—モートが、その内部の状態および近隣パスの品質を伝達するために送信するパケット。健全性レポートは、マネージャがネットワークの最適化に使用します。

アイドル・リスン—受信デバイスはパケットを受信するためにウェイクアップしているが、送信デバイスがパケットを送信していないスロット。一般的なネットワークにおいて、全ての受信スロットのうち 2/3 がアイドル・リスンになります。

ジョイン (参加)—新しいモートをネットワークに組み入れるためにモートとマネージャの間で行われる一連のハンドシェイク。最初にモートが暗号化された要求を提出し、リンクおよび実行時のセキュリティ資格が割り当てられて終了します。

キープ・アライブ—データがパス上で送信されていない状態においても、デバイス間の同期を保つために一定時間に一度送信される空のパケット。このタイムアウト時間は、通常 WirelessHART で 30 秒、SmartMesh IP で 15 秒です。

レイテンシー—パケットが生成された時間と最終着信先に到着した時間の差。

リーフ—現在、アップストリームの RX リンクがなく、したがって子がないモート。リーフ・ノードまたはリーフ・モートとも呼ばれます。

低消費電力ボーダー・ルーター (LBR) —6LowPAN (IPv6 for Low power Wireless Personal Area Networks、RFC 4944) パケットを IPv6 パケットに変換する論理または物理デバイス。エッジ・ルーターとも呼ばれます。

MAC—メディア・アクセス制御 (Medium Access Control) 層。技術的には、データ・リンク層 (OSI 層の第 2 層) の副層だが、通常 Dust 文書では同じ意味で使用されます。

マネージャー—ネットワークの確立と保守を担当するデバイスまたはプロセス。

マスタ (モード) —モートの動作モードのひとつ。モートは設定されたネットワークに自動的に参加しようとします。このモードではシリアル API が無効になり、常駐アプリケーションからオンボード I/O の一部を制御できます。

メッシュ—各モートが 1 つまたは複数のモートに接続されるネットワーク・トポロジ。

モート—フィールド・デバイスがセンサまたは他のデータを送信するために、無線通信を提供するデバイス。ネットワークを構築するための基本ブロック。

マルチホップ—1 つまたは複数のモートがアクセス・ポイントへの直接のパスを持たないようなネットワーク。データ・パケットは、送信元から送信先まで複数のホップを経由する場合があります。

マルチパス—複数の反射信号の重ね合わせからなる無線信号。マルチパス・フェージングなど、わずかな環境の違いで信号レベルが大幅に変化する現象を説明するために使用されます。

NACK—パケットの受信に際して送信される特別なACKフレーム。パケットは正しく受信されたが、転送することは拒否されたことを示します。

近隣—当該モートの無線伝搬範囲内にあるモート。

非ルーティング—他のモートの参加を促すための通知をしないように設定されているモート。非ルーティング・モートは、子を作ることはなく、子からのパケットを受け取って転送することはありません。

最適化—ネットワークの全てのモートから健全性レポートのパケットを受信し、そこに含まれる情報を利用して、エネルギー消費量とレイテンシが最小になるようにネットワークを再構築し続けるマネージャのプロセス。

パケット—フレームとも呼ばれる、可変サイズのデータ転送単位。

親 (または時間親)—時間同期の際の基準として機能するデバイス。Dust ネットワークでは、モートの親は、アクセス・ポイントに近い方のモートです。親は、子のデータをマネージャに転送します。

パス—2つのモート間で取ることが可能な接続の組み合わせです。リンクを割り当てられているパスは、使用中のパスであり、検出されたがリンクを割り当てられていないパスは、未使用のパスです。

パス安定性—2つのモート間で通信したパケットの数と確認応答を受けたパケットの数の比率。2つのモートは、A->BおよびB->Aの両方向のパス安定性のカウントを保持します。2つの方向でパス安定性が大幅に異なるパスは、非対称のパスと呼ばれます。

パイプ—1つのターゲット・モートとの間でレイテンシの低い高速な送受信を可能にする SmartMesh WirelessHART ネットワークの機能。

プロビジョニング—リンクの接続安定性が悪い場合のことを想定して、冗長性を持たせるために、モートが生成するパケット毎にマネージャが割り当てるリンクの数。デフォルトのプロビジョニングは3xです。これは、モートがパケットを待ち受けキューに追加する前に、パケットが正常に送信されるチャンスが平均3回あることを意味します。

発行レート (バースト・レート)—モート・アプリケーションがアップストリーム・データを送信する頻度。WirelessHARTでは、バースト・レートという用語は発行レートと同じです。

PHY—無線機などの物理層。

信頼性—生成されたパケット数に対して、正しく受信されたパケット数の比率。

ルート—送信元と送信先の間でパケットが通過するモート。例えば、モート3からのパケットはルート3-2-6-APを使用します。アップストリームで使用するグラフ・ルーティングのため、同一のモートでのパケット発信はランダムにさまざまなルートを取ります。

スケジュール—特定の目的のために設定される、ネットワーク内のスーパーフレームとリンクの集まり。

サービス (またはタイムテーブル)—タスクが必要とする帯域幅を要求して、それに対する割り当てを受け取る、または拒否されるプロセス。

スレーブ (モード) —モートの動作モードの一つ。センサ・アプリケーションは、APIを用いてモートがネットワークに参加するように指示する必要があります。

スロットフレームまたはスーパーフレーム—特定の繰り返し周期を持ち、グラフIDのラベルが付けられたタイムスロットの集まり。

ソース・ルート—パケット・ヘッダに送信元と送信先の間のホップが全て明示的に指定されているルート。Dust ネットワークは、ダウンストリーム・パケットにはソース・ルーティングのみを使用します。

スター—ネットワーク・トポロジの一つ。全てのモートは、アクセス・ポイント (ZigBeeにおけるPANコーディネータ) へ接続されているだけであり、ルーティング機能を持ちません。

スター・メッシュ—ネットワーク・トポロジの一つ。モートはルーターを中心とするスターを構成します。ルーターは1つまたは複数の近隣との間でメッシュを構成し、それを通してデータを転送します。

統計情報—モートからの健全性レポートから抽出された、ネットワーク・トポロジとその性能に関する情報。

TDMA—一時分割多重アクセス (Time Division Multiple Access)。パケット化された情報の交換が、1組のデバイスに明示的に割り当てられたタイムスロットおよびチャネル・オフセットでのみ発生する通信アーキテクチャ。

タイムスロット—1組のモートが最も長いパケットを送信して、その確認応答を受け取るために十分な長さの決められた時間幅。ネットワーク内の時間は同期したタイムスロットに分割されます。

アップストリーム—メッシュ・ネットワークからマネージャまたは有線側のアプリケーションへの方向。

ZigBee—単一チャネルのCSMAネットワーク・プロトコル。

3 SmartMesh IP ネットワーク

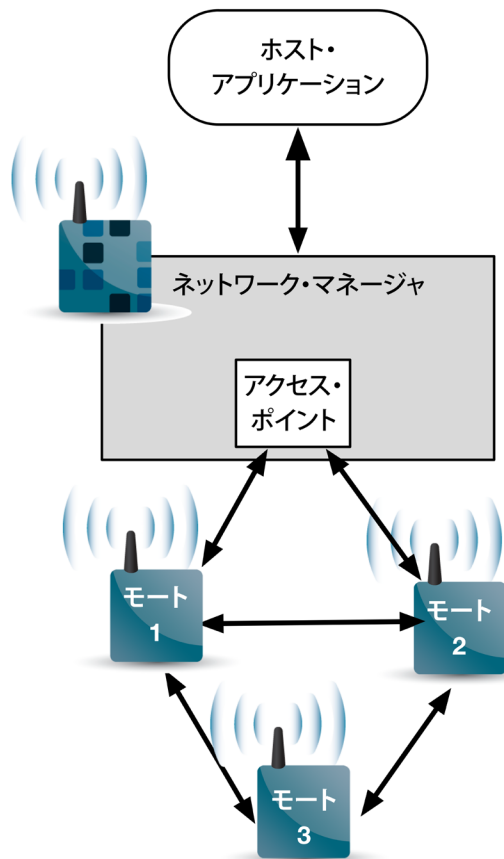
3.1 はじめに

ネットワーク・ユーザー・ガイドは、ネットワークとデバイスの動作と機能に関する重要な内容を、高いレベルで説明することを目的としています。この中で参照するAPIの詳細な内容については、各 SmartMesh IP のマネージャ・ガイドおよびモート・ガイドをご覧ください。

3.1.1 ネットワークの概要

SmartMesh® ネットワークは、自動的にマルチホッピングのネットワークを構築して、データを収集しそれを中継するモートと呼ばれるノード群と、ネットワーク全体の性能とセキュリティを管理し、ホスト・アプリケーションとの間でデータを交換するネットワーク・マネージャで構成されます。

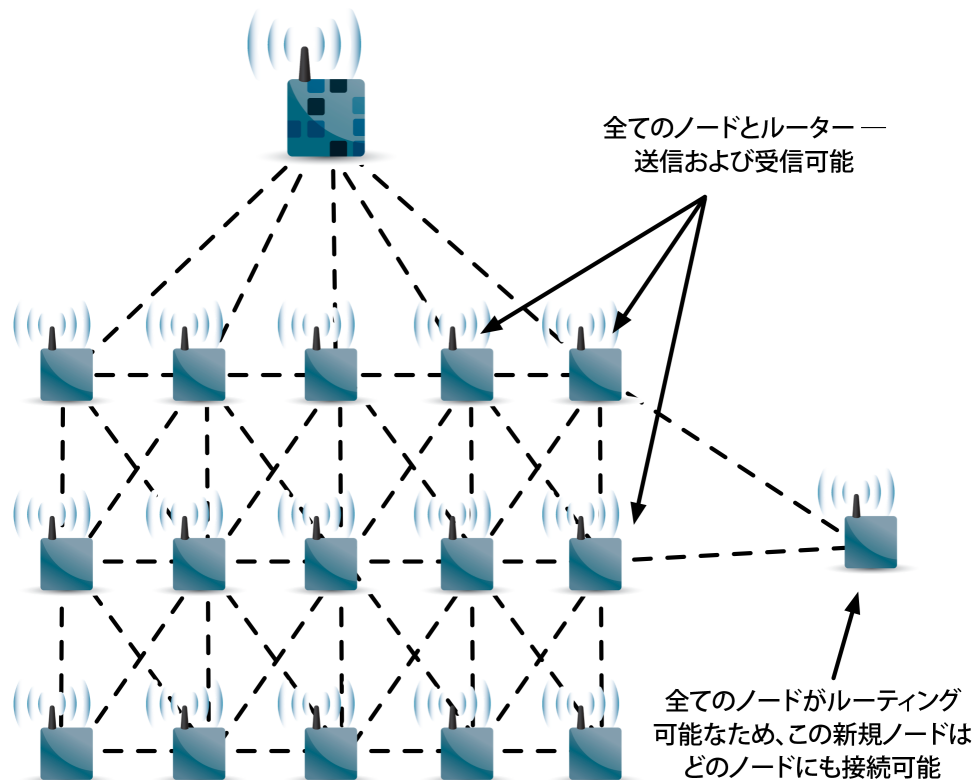
SmartMesh ネットワークは、リニアテクノロジーの Dust Networks グループが開発した、タイム・スロット・チャネル・ホッピング (TSCH : Time Slotted Channel Hopping) リンク層を使って通信します。TSCH ネットワークでは、ネットワーク内の全モートが1ミリ秒以下で同期されます。ネットワーク内の時間は、タイムスロットとして管理されています。これにより、衝突のないパケット交換と送信毎のチャネル・ホッピングが可能になります。SmartMesh ネットワークでは、各デバイスに、干渉、物理的遮断、またはマルチパス・フェージングによる通信の中断を克服するための冗長パスを提供する1つまたは複数の親があります (例えば、モート3には、その親としてモート1とモート2があります)。パケット送信があるパスで失敗すると、次の再送信は異なるパスおよび異なるチャネルで行われます。十分な冗長性を持ったネットワークを構築するためには、単純な配置ガイドラインに従うだけで構いません。これは、アプリケーション・ノート「Planning a Deployment」に記載されています。



ネットワーク・マネージャが内蔵のアクセス・ポイント無線機から、デバイスがネットワークに参加するために必要な情報を含む通知パケットの送信を行うことにより、ネットワークの形成が開始します。この際のメッセージ交換は、マネージャやアプリケーションとの間で暗号化された通信を確立するセキュリティ・ハンドシェイクの一部でもあります。モートがネットワークに参加した後は、パケットの確認のたびに時間を修正することにより同期を維持します。

検出プロセスを継続することにより、ネットワークは無線の状態が変化すると新しいパスを継続して検出できます。更に、ネットワーク内の各モートは、使用中のパスの通信品質や代替パスのリストなどの統計情報を含んだ健全性レポートと呼ばれるパケットを定期的にネットワーク・マネージャに送信します。ネットワーク・マネージャは、健全性レポートを使ってネットワークを継続して最適化し、最も過酷な無線環境でも99.999%を超えるデータの信頼性を維持します。

TSCHを使用することにより、SmartMeshのデバイスはスケジュールされた通信の合間にスリープ状態になり、この状態では消費電力を非常に少なく抑えることが可能です。モートは、送信または受信するようスケジュールされたタイムスロットでのみ動作するため、通常でのデューティ・サイクルは1%未満になります。ネットワーク・マネージャの最適化ソフトウェアは、このスケジュールを自動的に管理します。Eterna低消費電力無線ICを用いたSmartMeshネットワークにおける全てのモートは、頻繁にルーティングを行うものでも、数年間バッテリーで稼動できます。デフォルトの設定では、ネットワーク内の全てのモートが他のモートからのトラフィックをルーティングできるようになっています。これにより、ルーターとそれに接続される非ルーティング・ノードの配置関係を考える必要がなくなり、配置計画が大幅に簡略化されます。また、特定のモートの消費電力を更に大幅に下げのために、あるモートを非ルーティングに設定することもでき、これによりいろいろなネットワーク・トポロジを構築することができます。

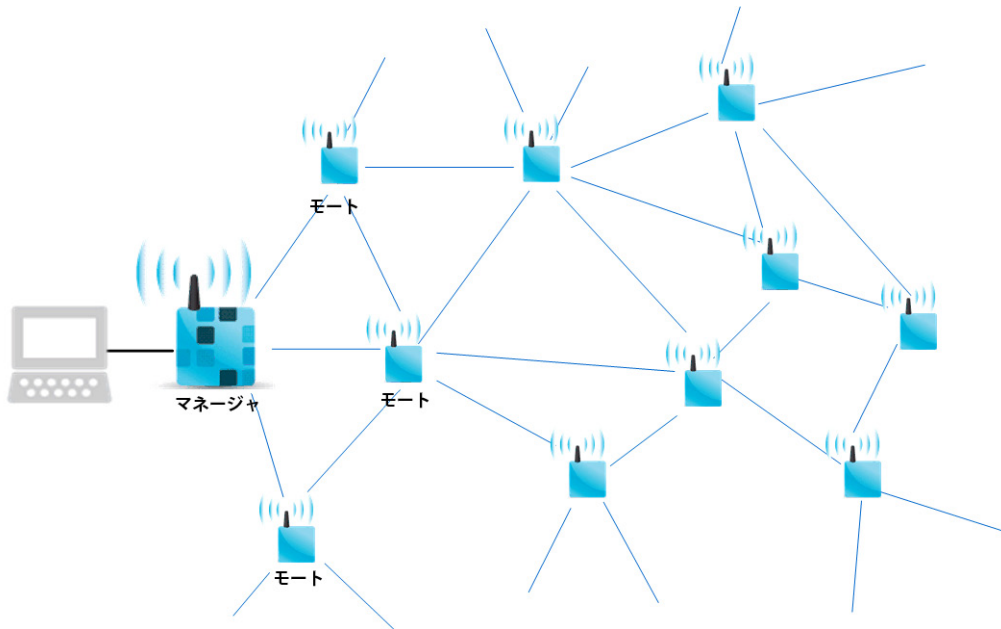


SmartMeshのモートとネットワーク・マネージャを構成する際の中心となるデバイス、Eterna IEEE 802.15.4e System-on-Chip (SoC) です。このICには、高度に集積、統合された低消費電力無線回路およびSmartMeshネットワーキング・ソフトウェアを実行するARM® Cortex™-M3 32ビット・マイクロプロセッサを搭載しています。SmartMeshソフトウェアは完全にコンパイルされた状態で供給され、ホスト・アプリケーションがネットワークを制御するためのアプリケーション・プログラミング・インタフェース (API) が提供されています。APIには、デバイスへのデータ送信、各モートのデータ発行頻度の設定、ネットワークの状態や性能の状況監視などの豊富な機能が用意されています。データ発行頻度については、ネットワーク全体で共通の値に設定することもできますし、各モートの必要とするデータ転送量に合わせて頻度を下げることや、または1秒当たり1回以上の速さに設定することもできます。

3.1.2 SmartMesh IPについて

SmartMesh IPは、信頼性と超低消費電力を両立させたインターネット・プロトコル (IP) 層と物理層 (PHY) を備え、広い範囲の応用分野に、堅牢で標準規格に準拠した、配置計画の容易なソリューションを提供します。リニアテクノロジーのEterna™ 802.15.4e SoCを用いることにより、SmartMesh IPネットワーク内の全てのノードは、バッテリーで10年間、環境発電でほぼ永久的に稼働します。任意の場所にケーブル配線なしでセンサを置けるため、現場担当者の工数を削減し、インフラ利用者の利便性を下げることなくネットワークを設置できます。メッシュ・ネットワークの自動的形成と自己修復を可能にするアルゴリズムにより、SmartMesh IPシステムは、無線技術の専門知識を持たない現場担当者でも容易に設置できます。SmartMesh IPネットワークは、少なくとも1つのSmartMeshマネージャ、および組み込みSmartMesh Eternaチップを含む最大32台または100台 (外部RAMを装備したマネージャ上) のデバイスで構成されます。各マネージャは、24パケット/秒または36パケット/秒 (外部RAM使用時) の総データ量をサポートします。より大きい規模のネットワークを実現するため、またはより多くのデータ量が必要な場合には、オーバーラップした多重ネットワークも容易に構築することができます。

SmartMesh IP Starter Kit ([DC9021A](#)) は、総務省の技適を取得済みの1つのSmartMeshマネージャと5つのモートで構成されています。キットで提供されるWindowsベースのグラフィカル・ユーザー・インタフェースであるStargazerを使って、PCからネットワークを構成し管理することができます。このキットに含まれるモートは、スタンドアロンで動作可能な**マスタ・モード**で出荷されます。このモードには、自動的にネットワークに参加し、内蔵センサの値を読み取ることができるサンプル・アプリケーションが搭載されています。量産時のデバイスのデフォルト設定である**スレーブ・モード**は、センサ・プロセッサがAPIを用いてモートを制御するアプリケーションを想定しています。詳細は、本ガイドの「モート」のセクションで説明します。



SmartMesh IP ネットワーク

SmartMesh IP の機能

SmartMesh IP システムには、以下の機能があります。

- **超低消費電力ネットワーク**—ネットワークは、バッテリー、環境発電、またはライン給電で動作できます。
- **ネットワークの高信頼性**—過酷な無線環境でも 99.999% を越えるネットワークの信頼性を誇ります。
- **IPv6 アドレス指定可能**—6LoWPAN を IEEE 802.15.4e と組み合わせています。
- **低レイテンシ・モード**—ライン給電によるバックボーンを活用することによって、通信頻度の低いネットワークにおいてレイテンシを大幅に下げることができます。
- **包括的なセキュリティ管理**—応用分野が必要とするセキュリティを持ったネットワークを構成できます。

「SmartMesh IP ネットワーク・ユーザー・ガイド」は、ネットワークの重要な動作および機能を、高いレベルで説明することを目的としています。ここで参照するAPIの詳細については、「[SmartMesh IP Manager API Guide](#)」または「[SmartMesh IP Mote API Guide](#)」をご覧ください。

3.2 ネットワーク形成

モートがネットワークに参加する場合、既にネットワークにあるモートまたはアクセス・ポイント (AP) からの通知を受信することにより、ネットワーク内の他のデバイスと時間同期する必要があります。マネージャが AP に通知を送信するように指示すると、ネットワークの構築が始まります。モートは AP からの通知を受信して参加した後、自分自身からも通知の送信を始めます。このプロセスは、他の全てのモートがネットワークに参加するまで、繰り返されます。通知は、同期とリンク情報を含む IEEE 802.15.4e Beacon フレームです。通知には新しいデバイスの時間同期に加えて、新しいデバイスがネットワークへの参加要求をいつ送信すればよいのか、またそれに対する回答をいつ受信すればよいのかが指定されています。これにより、新しいモートに暫定的なリンクが割り当てられます。この暫定リンクは、そのモートがマネージャから正式なリンクを割り当てられるまでの間使用されます。

SmartMesh IP ネットワークでは、アクセス・ポイントによる通知は起動時に自動的に、または *startNetwork* API により明示的に発生します。起動動作は、*autostart* 不揮発性設定パラメータを介して設定されます。SmartMesh IP マネージャは、256 のスロットフレーム当たり 4 回、平均で 500 ミリ秒ごとに 1 回通知します。AP の無線範囲内にあるモートは、これらの通知の 1 つを受信した後で参加します。5% のデューティ・サイクルおよび一般的な 80% のパス安定性での 1 ホップ・モートの平均同期時間は、次のようになります。

```
Synch time = adv rate per device * #channels / (#advertisers * path stability * join duty cycle)
             = 0.5 s * 15 / (1 * 0.8 * 0.05)
             = 188 s
```

モートのジョイン・デューティ・サイクルは、*joinDutyCycle* モート・パラメータで設定できます。100% のデューティ・サイクルを設定した場合の平均同期時間は、10 秒未満になります。より高いデューティ・サイクルを設定すると、モートの消費電力は増えますが、より早く同期できるようになります。ネットワークに 1 ホップ・モートがたくさんある場合には、低いデューティ・サイクルを設定しても平均同期時間が大幅に長くなることはありません。消費電力が非常に厳しく制限されたアプリケーションの場合、デューティ・サイクルは 0.5% 程度に設定することもできます。

同期後、モートは、他のモートからの通知を受け取ることを目的に受信を続けながら、1 スロットフレームの間待機します。モートはその後その持つ電源容量とルーティング能力に関する情報、および待機中に受信した近隣のモートのリストからなるジョイン要求を送信します。このメッセージ交換は、マネージャやアプリケーション、モートの間で暗号化された通信を確立するためのセキュリティ・ハンドシェイクの一部を構成し、共有の秘密ジョイン・キーを使って暗号化されます。マネージャは、実行時 MAC 層認証キー、マネージャへのセッション、これ以降の全ての通信にモートが使用するショート・アドレス、およびマネージャへのルートを知らせます。さらにパケットをやり取りすることによって、ブロードキャスト・セッションを開くためのルート、スロットフレームおよびリンクを割り当て、時間親を設定します。この際、WirelessHART とは異なり、暫定リンクから正式なリンクに移行するように指示するアクティベート・コマンドを送信します。

この時点で、デバイスはデータ発行用に追加の帯域幅を要求することができるようになります。完全にネットワークに参加したモートは、マネージャの指示に従って通知を出力します。このように、ネットワークは、AP、1 ホップ・モート、2 ホップ・モートという順に、内から外へ構成されていきます。モートが参加するためにマネージャの無線範囲にある必要はありません。ネットワークに含まれたいくつかのモートの無線範囲内にあれば大丈夫です。

参加する途中にモートが近隣モートを受信するだけでなく、検出プロセスが継続して行われます。決められた検出間隔に従って、1 つのモートが送信し、その時間に他のモートは受信します。モートは、それが受信した近隣モートの情報を、定期的な健全性レポートでマネージャに伝えます。この仕組みにより、マネージャはネットワークの最適化に必要なパスの品質に関する情報を継続して集めることができます。

3.3 帯域幅とレイテンシ

各モートが取得できるアップストリーム・リンクの数に影響を与える設定可能なパラメータは複数あります。これらは相互に関連を持つ場合があります。システムは、多数のノードからなるネットワークを柔軟に構築でき、電池駆動でも長期間稼働できる超低消費電力を実現できるように設計されています。ネットワークの全てのモートで同じ量のデータ生成を行う場合には、マネージャで *basebw* パラメータを使用して基本の帯域幅を割り当てることを推奨します。モートごとに要求が異なる場合には、各モート上でサービス・モデルを個別に使用することを推奨します。それ以外のパラメータについては、特別な場合以外にはデフォルト値から変えないことを推奨します。他の制約に関わらず、全てのデバイスに少なくとも2つのアップストリーム・リンクが与えられ、これはモートが一つの親のみを持つ場合も同じです。

一般的にモートのリンクを増やすことには次の効果があります。

- レイテンシの低下
- パケット/秒のスループットの増加
- 消費電力の増加

モートに割り当てられた全ての帯域幅を実際に使用する必要はありません。低いレイテンシが要求されるアプリケーションでは、実際に必要となる帯域幅よりも多くの帯域幅を要求することにより、アップストリームのレイテンシを下げるができます。パス安定性には不確実性が伴うため、ネットワークはレイテンシの上限を保証しません。場合によっては、以下で述べる低レイテンシのバックボーンを使用できることもあります。

3.3.1 NumParents パラメータ

マネージャの *numParents* パラメータ (デフォルト = 2) は、モート毎に持つことが望ましい親の数を設定します。このパラメータはネットワーク全体に適用されます。2に設定すると、タイミングとデータ・ループを避けるために、常に一つのモートだけ単一の親を持ちます。同様に、3に設定すると、常に一つの親を持つモートと二つの親を持つモートが一つずつできます。ネットワークを最適化する際に、マネージャは、一部のモートに対して新しい親へのリンクをテストするため、一時的に *numParents* の値よりも多くの親を持たせる場合があります。劣悪な環境により頻繁にパス障害が発生するようなネットワークの場合、*numParents* を3または4に設定することにより、消費電力の増加と引き換えに切断の可能性を引き下げることができます。

3.3.2 BWMult パラメータ

マネージャの *BWMult* パラメータ (デフォルト = 300%) は、ネットワークの安全マージンを設定します。このパラメータはネットワーク全体に適用されます。モートが12秒ごとにパケットを生成するように要求し、*BWMult* が300%に設定されている場合、モートには少なくとも4秒あたり1リンクが割り当てられます。大多数のDustの実例で何年にもわたり300%という値を使用してきて問題がないことを確認しているため、この値を300%未満に下げ場合には特に注意してください。300%を使用した場合、概念的には、パス安定度33%まで下がっても、パケット・キューが一杯になることは避けられることになります。設定値を300%より下げ場合は、マルチホップ・ネットワークで消費電力量を最小限に抑えなければならないケースに限るようにしてください。

3.3.3 BaseBW パラメータ

マネージャの *BaseBW* パラメータ (デフォルト=9000 ミリ秒) は、全モートでパケットが平均的に生成される間隔です。このパラメータはネットワーク全体に適用されます。9000 ミリ秒の設定は、全てのコマンドと診断パケットを送るのに十分であるだけでなく、各モートのアプリケーションが 10 秒ごとに 1 パケットを生成するためにも十分です。上記の他のパラメータがデフォルト設定に設定されている場合、*BaseBW* を 3100 ミリ秒以上に設定していれば、リンクの数はデフォルトの 2 を越えることはありません。ネットワークの全てのモートで同じデータ生成レートの場合、*BaseBW* のみを使用して帯域幅の設定を行うことを推奨します。

3.3.4 サービス

モートにより異なるデータ・レートを設定したり、動的にデータ・レートを変更する必要がある場合には、サービス・モデルを使用する必要があります。このモデルでは、アプリケーションが希望するデータ・レートを要求する役割を持ちます。あらかじめセンサ・プロセッサが固定のデータ・レートを要求するか、アプリケーション・メッセージを使って動的に変更する方法が使えます。これは、必要に応じて OEM インテグレータが決めることになります。目的の送信先に対してソケットを開いた後、センサ・プロセッサは *requestService* API を使用して、希望するパケット間隔と送信先を要求します。センサ・プロセッサは、サービス・レベルが確立または変更された時に *serviceChanged* 通知を受け取ります。センサ・プロセッサは、常にどれだけの帯域幅が割り当てられているかを把握しているとともに、ネットワークが混雑した場合のバック・オフの処理を実装していることが望ましいです。

サービスに対して要求よりも低いレベルが割り当てられる場合があります。*serviceChanged* 通知を受け取る以外に、センサ・プロセッサは *getServiceInfo* API を使用して、保留中の要求のステータスを問い合わせることができます。センサ・プロセッサは、必要とするサービス・レベルが割り当てられていないことをホスト側のアプリケーションに連絡することができます。これらのアプリケーション・レベルのメッセージは、OEM インテグレータが必要に応じて決めます。ホスト側のアプリケーションも定期的に *getMotelInfo* マネージャ API を使って、要求した帯域幅と割り当てられた帯域幅の差異を確認できます。この情報により、サービス拒否の理由が、例えば既存の 1 ホップ・モートのリンク数が少なすぎるのが問題であると判れば、更にリピータが増設するなどの判断が可能になります。

3.3.5 カスケード・リンク

上記のパラメータは全て、モートがそれ自身のトラフィックに必要なリンクの数を決めるためのものです。モートに子がある場合、ネットワーク・マネージャはその子孫全てのトラフィックを転送するために必要なリンクをそのモートに追加します。この計算方式はカスケード・リンクと呼ばれます。マネージャは、そのモートへのアップストリーム RX リンクの数調べて、それとモート自身のトラフィックに必要なリンクの数を加算して、そのモートのアップストリーム TX リンクの数を決めます。マルチホップ・ネットワークでは、ネットワーク内のトラフィックが多くない場合でも、1 ホップ・モートにはリーフ・モートよりも多くのリンクがあります。

3.3.6 低レイテンシのバックボーン

SmartMesh IPネットワークは、ライン給電のモートを含むネットワーク用に、低レイテンシのバックボーン・スーパーフレームをサポートします。バックボーンは、全てのモートが共有するアップストリーム・リンクを提供する短い競合ベースのスーパーフレームです。帯域幅が共有されるため、サービスの割り当てによる方式よりも少ない消費電力で、全てのモートが低いレイテンシでの通信が可能です。帯域を共有することのデメリットは、複数のモートが同時に帯域を使用しようとすると、衝突が発生し、どちらとっても効果的に使用することができなくなることです。バックボーンは、その速度がネットワークの平均的なサービス・レベルよりもずっと高速な場合に効果があり、特に低いトラフィック向けに設計されたネットワークにおいて、稀に起きるアラームの配信を高速に行いたい場合などに有効です。

bbmode 設定は、マネージャの不揮発性パラメータです（デフォルト = 0、オフ）。アクティブ時には、アップストリームのみまたは双方向のいずれかになります。このパラメータの変更は、ネットワークをリセットした後に有効になります。バックボーンの長さは、*bbsize* パラメータを変更することで変更できます。数字を大きくすると、スーパーフレームの時間を長くして消費電力を節約できますが、レイテンシが増加します。

アップストリーム・バックボーンが有効な場合、電力容量が十分に大きいモートだけが、バックボーン内で子（リスン・リンクなど）を持つことができます。全てのモートは電力容量に関係なくアップストリーム・バックボーン内で送信リンクを持ちます。アップストリーム・バックボーンは、パケット衝突の可能性が低いような低いトラフィックの用途で、ZigBeeのようなライン給電のルーター・インフラを持つネットワークを構築して、センサ・レイテンシを最小化することを目的としています。

双方向バックボーンを有効にした場合、電力条件に関係なく全てのモートが参加するため、電池駆動のデバイスの電池がすぐに消耗してしまう可能性があります。全てのモートが、常に1mAを越える電源を供給されることが保証されている場合にのみ使用してください。

アクセス・ポイントがライン給電されていて、使用可能なリンクの1/3未満しか使用しておらず、バックボーンを使う用途が他にない場合には、アップストリーム・バックボーンを有効にしておくことを推奨します。デフォルトのモート電源情報では、モートはバックボーンの子を持つことはできないため、バックボーンRXリンクを持つデバイスはアクセス・ポイントだけとなり、バックボーンTXリンクを持つデバイスは1ホップ・モートだけとなります。つまり、どのモートの電力消費も増やすことなく、1ホップ・モートからAPへのレイテンシが大幅に短縮できます。全ての専用リンクは引き続き正常に動作しますし、アクセス・ポイントが専用リンクにサービスを提供していない時は、1ホップ・モートからの受信を続けます。アクセス・ポイントがライン給電されている場合には、この方法にデメリットはありません。

3.3.7 ノーマル/高速ダウンストリーム・モード

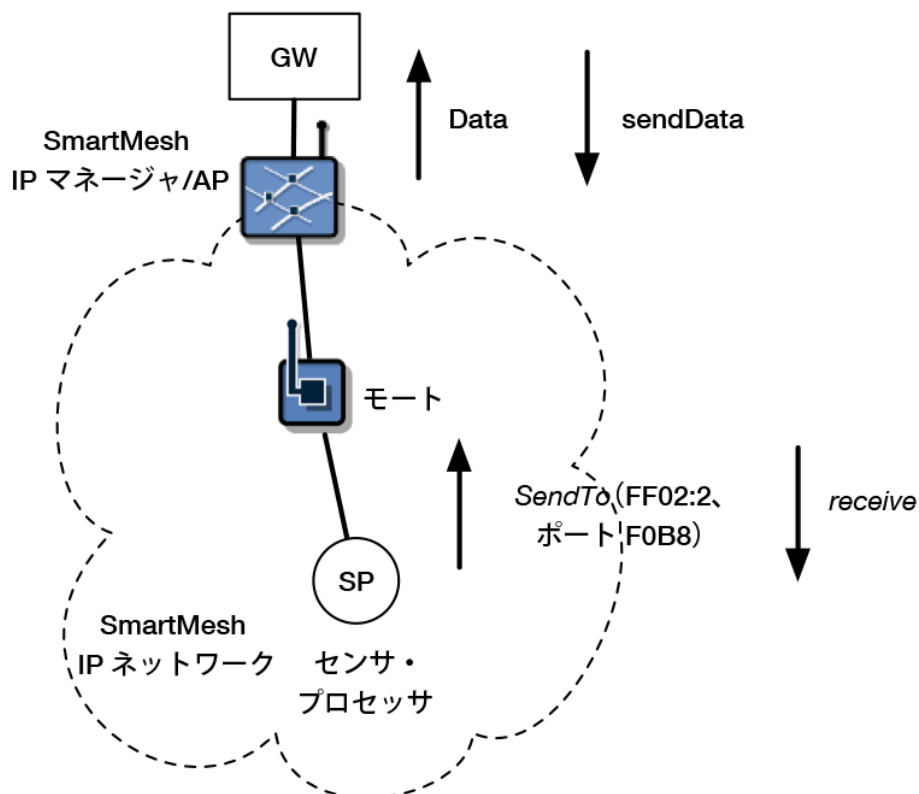
主にセンサ・データ収集のために使用されるネットワークでは、ネットワークの形成時には大量のダウンストリーム制御トラフィックが発生し、通常の稼働中は最適化と修復のための少ないトラフィックしか発生しない傾向があります。SmartMesh IPマネージャは、「高速」ダウンストリームのフレーム構築フェーズと「ノーマル」ダウンストリームのフレーム稼働フェーズとの間で自動的な遷移を行います。アップストリーム・フレームとダウンストリーム・フレームのデフォルトのフレーム・サイズは256スロットです。ユーザーは、CLIを介して*dnfr_mult* 不揮発性「設定」パラメータ（デフォルト=1）を2または4に変更して、「低速」をそれぞれ512または1024スロットに増やすことができます。また、プログラムから*downFrameMultVal* パラメータを使って*setNetworkConfig* APIを呼び出すことにより、変更することもできます。これにより、モートの平均電流はそれぞれ約4μAと8μA下げることができます。この機能を使った場合、マネージャがネットワーク内の問題を修正する際の速度が遅くなるため、デフォルト設定でのモート電流が許容できない場合にのみ、1を越える*downFrameMultVal* を使用してください。ユーザーは、*SetDownstreamFrameMode* コマンドを介して、マネージャにノーマルと高速のダウンストリーム・モード（*downFrameMultVal* が決定した *framelength*）を切り換えさせることもできます。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換わり、自動的にはフレーム・サイズを変更しなくなります。

3.4 データ・トラフィック

モートは、ユーザー・データに対して信頼性の低い転送 (UDP) を使用しますが、データが失われる可能性があるということではありません。ダウンストリームで、8 ホップ未満のモートに送信されるメッセージは、ホップ当たり複数の再試行を行うソース・ルーティングを使用します。この8 ホップにはAPを含めます。8 ホップのソース・ルート制限よりも深いモートに送信される場合、パケットが溢れてしまって送信先に到達しない可能性もあります。これが許容されない場合は、アプリケーション層の確認応答を使用して、確実にダウンストリームへ配信されたことを確認できます。

アップストリームで、パケットは確認応答されるまでリンク・レベルで無制限に再試行が行われます。まれに (通常 0.001% 未満)、転送のためにモートが受け入れたデータが、モートがリセットする時に失われることがあります。これが許容されない場合は、確認応答がダウンストリームで送信される速度 (1.9 秒当たり 1 回) まで発行を遅らせることになりますが、アプリケーション・レベルの確認応答を使用して、全てのパケットを検証することができます。

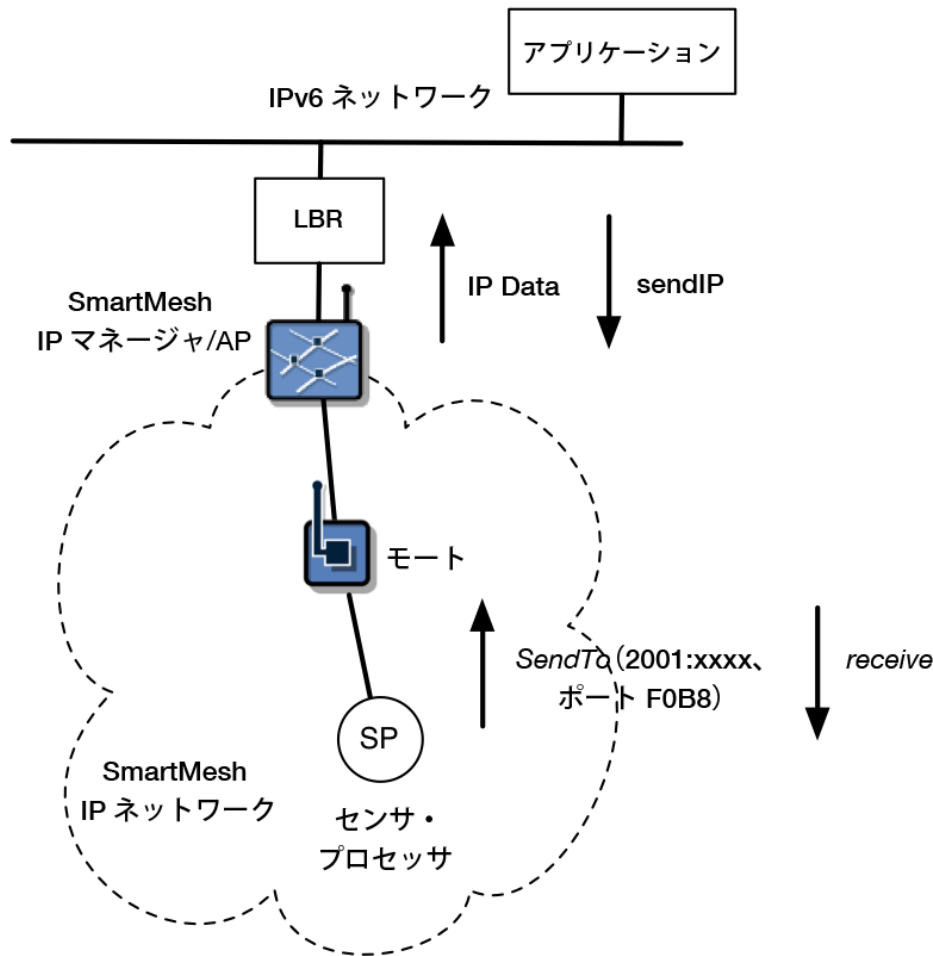
3.4.1 シリアル接続の「ゲートウェイ」アプリケーション



マネージャに直接接続するアプリケーションは、*sendData* API を使って、モート上の任意の UDP ポートにパケットを送信することができます。*sendData* API では、アプリケーションは送信先、ポート、およびペイロードの提供のみを担当します。この場合の送信先アドレスはモートの EUI-64 で、その IPv6 アドレスの下位 8 バイトでもあります。API が呼び出されると、*callbackId* が返されます。この *callbackId* は、このパケットがネットワークに送り込まれる時の *packetSent* 通知に含まれます。モートでパケットが受信されると、UDP ペイロードの他に IPv6 ソース/ポート情報を含む受信通知が生成されます。省略されるため、UDP チェックサム情報は含みません。

センサ・プロセッサは、モートで *sendTo* API を使用して、マネージャを介してゲートウェイにパケットを送信することができます。これらは、モートの EUI-64 アドレス、ポート情報、タイムスタンプ、および UDP ペイロードを含む *data* 通知になります。

3.4.2 エンド・ツー・エンドIP



IPv6ネットワークに接続される低出力境界ルーター (LBR、「エッジ・ルーター」と呼ばれることもある) は、メッシュおよび圧縮された6LowPANヘッダを生成し、*sendIP* マネージャ API を使用して通信します。*sendIP* API は、パケットがネットワークに送り込まれる時に *callbackId* を返します。モートでパケットが受信されると、UDP ペイロードの他に IPv6 ソース/ポート情報を含む *receive* 通知が生成されます。省略されるため、UDP チェックサム情報は含みません。

センサ・プロセッサは、モートで *sendTo* API を使用して、任意の目的地にパケットを送信します。パケットにマネージャ以外のアドレスが指定されている場合、パケットは *IP data* 通知になり、モートの EUI-64 アドレスとタイムスタンプを含みますが、UDP ペイロードに加えて 6LowPAN ヘッダが保存されます。

3.5 ネットワーク・セキュリティ

3.5.1 セキュリティ層

SmartMesh ネットワーク内のパケットは全て各層で認証され、エンド・ツー・エンドで暗号化されます。

- 認証—メッセージが示された送信者からのメッセージであること、および変更または再生されていないことを確認します
- 暗号化—ペイロードを秘密にしておきます

SmartMesh IP には複数のセキュリティ層があります。

- リンク層—パケットは、実行時キーと時間ベースのカウントを使って各ホップで認証されます。これにより、同期が確立し、マネージャによってネットワークへの参加を許可されたモートのみがメッセージを送信できるようになります。
- エンド・ツー・エンド—パケットは、実行時セッション・キーと共有カウントを使って認証され、エンド・ツー・エンドで暗号化されます。これにより、対象の受信者だけがメッセージを解読できるようになり（データ・プライバシー）、再生、データ破損、または中間者攻撃を避けることができるようになります（データ・セキュリティ）。

ネットワークに参加する時、モートは、マネージャが知っている共有の秘密ジョイン・キーを使ってネットワーク・マネージャに参加要求を送信します。キーの選択は、以下に説明するセキュリティ・モードにより決定されます。モートは、そのジョイン・キーで参加要求を暗号化し、マネージャはデータ・トラフィックのエンド・ツー・エンド暗号化用セッション・キーで応答します。これはセキュリティ・ハンドシェイクと呼ばれています。通知はリンク層の特殊な例です。新しいモートが認証できるようによく知られたキーで認証されます。正しいジョイン・キーを使ってモートが参加すると、そのモートはネットワーク・データを暗号化するために使用する4つのセッション・キーを受け取ります。

- ネットワーク管理トラフィックに使用するモート固有のセッション・キー
- アプリケーション・トラフィックに使用するモート固有のセッション・キー
- ネットワーク管理のために全てのモートが使用するブロードキャスト・セッション・キー
- アプリケーション・トラフィックのために全てのモートが使用するブロードキャスト・セッション・キー

この4つのキーを使って、通常データの発行は全て、生成元のモートにより暗号化され、マネージャでのみ解読されます。盗聴者やルーティング・モートのいずれもパケット・データを解読できません。同様に、マネージャのみが、マネージャ・コマンドに対する応答を読み取ることができます。特定のモートに対するダウンストリームのコマンドが、ルーティング・モートや非対象の受信者によって知られることはありません。ネットワーク内の全てのモートに明示的に送信されるコマンドのみ誰もが知ることができ、適切なブロードキャスト・キーを使って解読されます。

3.5.2 セキュリティ・モード

SmartMesh IPでは、マネージャが参加要求を解読する方法を決定するセキュリティ・モードを選択できます。マネージャは、次の3通りの方法で解読できます。

- 共通キー：最も緩いセキュリティ・モードは、共通のジョイン・キーを受け取ることです。このモードでは、マネージャは、ネットワーク全体で共通のジョイン・キーで暗号化された参加要求を出すモートを受け入れます。Dust Networksのネットワーク・マネージャは、公に知られているデフォルトのジョイン・キーおよびネットワークIDを設定して出荷されます。ネットワークIDと共通のジョイン・キーが変更されないままであれば、セッション・キーを割り当てるパケットの盗聴および解読は、困難ですが技術的には可能です。したがって、共通のジョイン・キーを秘密のジョイン・キーに変更することを強く推奨します。
- アクセス制御リスト (ACL)：マネージャは、アクセス制御リストにあるモートのみを受け入れるように設定することもできます。マネージャは参加要求を受け取って、最初に参加するモートのシリアル番号を探し、次に関連付けられたジョイン・キーで要求を解読します。両手順が成功した場合、モートはネットワークに受け入れられます。ACLは全てのモートに対して個別のジョイン・キーを設定することができます。これは最も安全なモードですが、マネージャと全てのモートの情報を設置の前に設定する必要があるため、ネットワークの構築の準備作業に手間がかかります。これらのデバイスが既に正しく設定されている場合、現場担当者は何もする必要はありません。モートは通知を受信すると参加します。ACLは共通キーで設定できます。参加できるようにするにはデバイスのMACアドレスをマネージャが知っている必要があるため、共通キー単独場合と比べてセキュリティが強化されます。
- 共通キー -> ACL：共通キーでネットワークを形成し、その後ACLを作成して、無線で各モートに一意のキーを割り当てることもできます。

4 SmartMesh IP マネージャ

4.1 はじめに

SmartMesh IP マネージャは、SmartMesh IP ネットワークの「頭脳」です。マネージャは、以下を担当します。

- キーや一時的カウンタなどのセキュリティ情報を管理しネットワークに配信します
- 常に時間同期が維持され、データ・サービス・レベルが適正となるように、全てのモートのリンク・スケジュールを決定します
- ネットワークの状態を継続して把握するために健全性レポートを収集します
- トポロジの変更に対応します
- モートの消費電力を最小にし、トラフィックを分散するようネットワークを最適化します
- ネットワーク・ホストにユーザー・インタフェースを提供します

SmartMesh IP マネージャは、同一のデバイスに管理機能とアクセス・ポイント (AP) 機能を組み込んだ単一チップのソリューションです。モジュラ LTP5902-IPR と LTP5901-IPR マネージャは、最大 32 のモートで構成されるネットワークをサポートします。LTC5800-IPR は、それ自身で最大 32 のモートで構成されるネットワーク、または外部 RAM を追加した場合には最大 100 のモートで構成されるネットワークをサポートします。また、外部 RAM を使用すると、より多くのメモリをリンク記憶領域に使用できるため、アップストリームのデータ・スループットが 30% 増加します。

マネージャ	AP RX リンク数	秒当たりの AP RX 数	Pkt/s @ 3x prov	Pkt/s @ 2x prov
LTP5800/5901/5902-IPR	150	72.9	24.3	36.4
LTP5800-IPR + 外部 RAM	223	108.3	36.1	54.2

4.1.1 マネージャ・クライアント・アプリケーション設計の手順

マネージャにはネットワークを管理するために必要な多くのタスクがありますが、クライアントは比較的わずかです。最小限のタスクは以下の通りです。

- マネージャへの接続
- ネットワーク構築の前に必要なパラメータの設定 (*networkID* など)
- モートの状態を監視しデータを収集するための通知に加入

「[SmartMesh IP Manager API Guide](#)」は、ACL の使用を含むセキュリティの設定や、健全性レポートの通知からの詳細な統計情報の収集など、マネージャを制御するためのコマンドについて説明します。「[SmartMesh IP Manager CLI Guide](#)」は、モートの状態やデータの収集を含むマネージャの活動を制御するコマンド・ライン・インタフェースの使い方について説明します。

4.2 マネージャへのアクセス

4.2.1 コマンド・ライン・インタフェース (CLI)

SmartMesh IP マネージャは、端末プログラム (PuTTY など) を通してアクセス可能なコマンド・ライン・インタフェースを提供します。CLI は、開発の過程でユーザーがマネージャと対話することを目的としています。ネットワーク ID などのシステム・パラメータは、CLI を通して設定できます。システムとネットワークのステータス情報も、このインタフェースを介して取得できます。CLI のアクセスは、ユーザーが変更できるユーザー名 / パスワードで保護されています。

✔ CLI への接続と使用可能なコマンドの詳細は、「[SmartMesh IP Manager CLI Guide](#)」を参照してください。

4.2.2 アプリケーション・プログラミング・インタフェース (API)

SmartMesh IP Manager API は、プログラムからネットワークを制御するためのインタフェースを提供します。ホスト・アプリケーションは API を使用してマネージャと通信します。API ポートは、115,200 ボー、8 データ・ビット、パリティなし、1 ストップ・ビットで動作します。

マネージャ API を使ってマネージャと通信するためのツールがいくつかあります。

- Python Developer SDK は、API を使って一般的なタスクを実行するためのツールを提供します。
- Serial API Multiplexer (Serial Mux) を使用すると、一つまたは複数のクライアントが TCP 接続を通してマネージャ API にアクセスできるようになります。
- Stargazer UI は、グラフィカルな無線ネットワークの表示および定期的なセンサ・データ・レポートを設定するためのインタフェースを提供します。

✔ API への接続と使用可能なコマンドの詳細は、「[SmartMesh IP Manager API Guide](#)」を参照してください。

4.3 設定と使用法

4.3.1 ネットワークID

複数のネットワークを空間的にオーバーラップして配置することは、それぞれのネットワーク（のモートとマネージャー）にそれぞれ異なるネットワークIDを割り当てることにより可能になります。モートは、異なるネットワークIDを持つモートやマネージャとは通信しません。マネージャ設定に保存されたネットワークIDは、APIの`setNetworkConfig`またはCLIの`set config`コマンドにより更新できます。新しいネットワークIDを使用するには、マネージャをリセットする必要があります。

`exchangeNetworkId` コマンドを使って、特定のネットワーク内にある全てのモートのネットワークIDを変更できます。`exchangeNetworkId` コマンドは、ネットワーク内にある全てのモートに新しいネットワークIDを確実に渡します。モートがネットワークID交換操作の確認応答に失敗した場合、そのモートはネットワークから削除されます。`exchangeNetworkId` 操作は、完了するまでに数分かかる場合があります。

4.3.2 ネットワーク時間

マネージャは、ネットワーク内の時間同期に責任を持ちます。デフォルトの設定では、マネージャは起動時にネットワークの構築を自動的に開始するようになっています。UTC時間を設定するためには、自動開始を無効にしておく必要があります。また、アプリケーションは、`setTime` コマンドで時間を設定した後、ネットワークの構築を開始する`startNetwork` コマンドを発行する必要があります。ネットワークの構築が始まると、ネットワーク内の時間は、ネットワーク内の全てのデバイス上で共通して同時にドリフトしていきます。マネージャの将来のリリースでは、ネットワーク時間は外部時間と随時同期できるようになります。

アプリケーションがマネージャ上の現在時刻を問い合わせる方法は2つあります。

1. アプリケーションが、TIME GPIOピンをトリガします。マネージャは、トリガを検出すると、Time通知を送信します。
2. アプリケーションが、`getTime` コマンドを呼び出します。マネージャは、`getTime` 要求を受信すると、Time通知を送信します。シリアル・ポートI/Oとコマンド処理の遅延のため、この方法はTIME GPIOをトリガするよりも精度が劣ります。`getTime` APIは、シリアル・ポートAPIを通してのみ通信するシステムとの互換性のために提供されます。

networkTime通知には、マネージャの現在時刻が以下の複数の形式で含まれています。

- **ASN**：絶対スロット番号。ASN=0（2002年7月2日20:00:00 UTC）以降のスロット数。
- **稼動時間**：マネージャの稼動時間（秒単位）
- **Unix (UTC) 時間**：1970年1月1日以来の秒とミリ秒

4.3.3 モートとの通信

外部ネットワーク側のアプリケーションは、モートおよびそれと接続されたセンサ・プロセッサとの通信を担当します。アプリケーションは *sendData* API を使ってパケットをモートに送信し、データ通知によりモートからのアップストリーム・データを受信します。

ネットワーク内の各パケットには、送信元と送信先のアドレスとポートが含まれます。ポートは、特定サービス宛のメッセージの効率的なフィルタリングを提供するために TCP/UDP で使用されます。アプリケーションでどのポートを使うかについては、「[SmartMesh IP ユーザー・ガイド](#)」の「[UDP ポート割り当て](#)」のセクションを参照してください。

アップストリームAPIの使用例の詳細は、SmartMesh SDKのUpStreamとSensorDataReceiverツールを参照してください。

4.3.4 ライセンス

マネージャには、特定のオプション機能を有効にするライセンスが含まれています。マネージャ・ライセンスは、そのデバイスの MAC アドレスと関連づけられています。追加機能を有効にするために、CLI または API インタフェースを通してライセンスを更新可能です。ライセンスを更新した後、新しい機能を使用可能にするために、マネージャをリスタートする必要があります。

CLI で以下を実行します。

```
> set config license=00:00:00:00:00:00:00:ff:94:63:7d:32:59
```

ライセンスがそのデバイスに対して有効でない場合、マネージャはそのライセンスを拒否します。

ライセンス・コマンドの詳細は、「[SmartMesh IP Manager CLI Guide](#)」および「[SmartMesh IP Manager API Guide](#)」を参照してください。

4.4 ネットワーク・アクティビティ

4.4.1 ネットワークの構造と形成

SmartMesh IP マネージャは、自動的にネットワーク管理を実行して、ネットワークの健全性を維持し、低消費電力と高信頼性のためにネットワークを最適化します。またマネージャは、干渉、サービス要求の追加や削除、およびデバイスの追加や削除など、パスの安定性やネットワークの状態の変化に応じてメッシュに対する変更を動的に行います。顧客アプリケーションは、ネットワークの管理面に関わる必要は全くありません。

モートがマネージャの近くで起動された場合、ほぼすぐに参加を開始します。アクセス・ポイントは平均して秒当たり2回の通知を行い、いったん参加したモートは2秒ごとに1回の通知を行います。「非ルーティング」と明示的に指定されたモートを除いて、全てのモートが通知を行います。マネージャは通知を停止しません。必要な場合は、アプリケーションはモートによる通知を停止することができ、モートの消費電力を下げるすることができます。このようなアプリケーションは、モートが紛失した場合にそれを検出し、再度接続するために通知を再開できる必要があります。

マネージャのネットワークに参加するために、モートはマネージャが使用するネットワークIDを割り当てられる必要があります。マネージャとモートは、セキュリティ・キーと適切なネットワーク・ルーティングを確立するために、参加プロセスにおいて数回メッセージを交換します。モートは、参加プロセス中の数回の状態変化を経て、**稼動状態**になります。マネージャは、モートがネットワークを離れたことを検出して、モートの状態を**紛失**に更新します。

外部アプリケーションは、イベント通知を受けることによりモートのネットワークへの参加と退出を把握できます。

4.4.2 ネットワークの健全性

モートとマネージャはそれぞれ、ネットワークとデバイスの統計情報を記録します。

以下のAPIコマンドを通して統計情報を照会できます。

- *getNetworkInfo* は、ネットワーク全体の統計情報を提供します。
- *getMotelInfo* は、モートとの通信から集められた統計情報を提供します。
- *getManagerStatistics* は、マネージャ APIからの統計情報を提供します。

CLIからの `show stat` コマンドは、これらのコマンド全てを実行します。

```
> show stat
Manager Statistics -----
established connections: 1
dropped connections    : 0
transmit OK           : 7971
transmit error        : 0
transmit repeat       : 0
receive OK            : 40
receive error         : 0
acknowledge delay avrg : 19 msec
acknowledge delay max  : 112 msec
Network Statistics -----
reliability: 100% (Arrived/Lost: 7998/0)
stability:   95% (Transmit/Fails: 5663/290)
latency:    1000 msec
Motes Statistics -----
Mote Received Lost Reliability Latency Hops
#2          1726  0    100%         990   1.0
#3          3143  0    100%        1190   1.0
#4          3020  0    100%         800   1.0
```

- *established connections* : 確立されたAPI接続の数
- *dropped connections* : 破棄されたAPI接続の数
- *transmit OK* : クライアント・アプリケーションにより確認応答されたAPIパケット (通知など)
- *transmit error* : 確認応答されていないAPIパケット
- *transmit repeat* : 再試行されたAPIパケット (何度か再試行すると送信エラーより大きい数になり得る)
- *receive OK* : マネージャが確認応答したAPIコマンド
- *receive error* : マネージャが確認応答できなかったAPIコマンド (フレーミングや他のエラーにより)
- *acknowledge delay avrg* : APIパケットの応答を受け取る平均時間。
- *acknowledge delay max* : APIパケットの応答を受け取る最長時間。
- *reliability* : ネットワークの信頼性は、モートが生成しマネージャが正常に受信したパケットの割合として計算されます。健全なネットワークでは、この数字は99.99%以上になります。この統計情報は、パケットが入信した時にパケットのセキュリティ・カウンタを監視することによって、マネージャ上でカウントされます。
- *stability* : ネットワーク・パスの安定性は成功した全てのMAC送信の割合で示されます。この数字は、各モートの配置に応じて異なります。ネットワークは、50%の安定性でも100%の信頼性を達成するように設計されています。安定性の値が低い時、より多くのパケットを再試行する必要があるため、モートはより多くの電力と帯域幅を使用します。この統計情報は、モートから受信した健全性レポートに基づき計算されます。マネージャは新しい健全性レポートが到着する度に、送信と失敗の合計を積算します。
- *latency* : マネージャが受信したパケットの平均アップストリーム・レイテンシ。マネージャは生成タイムスタンプ (ASN) のパケット・ヘッダを確認し、これを現在のASNと比較して、個別のパケット・レイテンシを計算します。
- *Mote Statistics* : モートの統計情報には、そのモートに関わるネットワーク統計情報の内訳が示されます。ネットワーク全体について示される統計情報に加えて、アップストリーム・パケットがモートからマネージャまでに経由したホップの平均数も出力されます。これは、各ホップで1ずつ減るメッシュ・ヘッダのTTLフィールドを用いて計算されます。

マネージャの統計情報は、*clearStatistics* APIまたはCLIの `exec clearStat` コマンドを使ってリセットできます。

4.4.3 健全性レポート

モートは、健全性レポートでマネージャに統計情報を定期的にレポートします。マネージャは健全性レポートを使って、ネットワークを最適化するために、モートの近隣の状態、およびパスの安定性を判断します。マネージャは、いくつかの統計情報を累積しますが、個別の健全性レポートは保存しません。

健全性レポートはAPIの通知として受信できるため、マネージャに接続された外部アプリケーションは、健全性レポートの通知に加入し、長期間にわたる詳細なネットワーク統計情報を解析することができます。

4.4.4 最適化

健全性レポートの情報に基づき、マネージャは継続してネットワークの改善に取り組みます。マネージャは、モートごとに、既存の親と、他に検出した近隣のモートを比較します。ある既存の親が別の親よりも適していると思われる場合、または検出されたが未使用の近隣が最適な親になりそうな場合、マネージャはモートから最適な親へのリンクを追加します。これは、**最適化追加サイクル**と呼ばれます。親の比較とは、パスの安定性、アクセス・ポイントまでのホップの数、およびモート上のリンクの相対数を考慮した評価関数の比較です。マネージャは、各パケットの送信を少なくして安定性を高めることによりネットワークにおけるエネルギー消費を減らそうとし、また、最も忙しいモートの負荷を分散しようとしています。追加サイクルの間、マネージャは、どのパスがネットワークにとって最適になるかを推測しなければならない場合があり、各モートにリンクを追加することによってパスを試します。

1時間後、マネージャは新しい親を有するモートから4回分の健全性レポートを受け取っており、どの親が最適かを判断するための十分なデータを得ます。この前の段階でリンクを追加されたモートは、同じ評価関数による比較を行い、最も不適切な親へのリンクを削除します。これは、**最適化削除サイクル**と呼ばれます。ネットワークは、その存続の間、1時間おきに追加と削除を繰り返し続けます。このプロセスを継続して行うことにより、いかなる環境の変化にもネットワークが対応できるようになります。

4.5 アクセス制御

4.5.1 ACL 管理

モートは、ネットワークに参加する際の最初の参加要求を暗号化するために、ジョイン・キーを使用します。モートが使用するジョイン・キーがマネージャに設定されたモートのジョイン・キーと一致しない場合、マネージャは参加要求を解読できず、モートはネットワークへの参加を許可されません。デフォルトでは、モートとマネージャは、共通のジョイン・キーがデフォルト値に設定された状態で出荷されます。セキュリティの詳細は、本ガイドの「[SmartMesh IP ネットワーク](#)」のセクションを参照してください。

マネージャは、モートのMACアドレスをモート固有のジョイン・キーに割り当てるアクセス制御リスト (ACL) を管理します。いったんACLエントリが設定されると、マネージャは、ACL内に存在しないモートをネットワークに参加させません。ネットワーク内のモートのジョイン・キーは、`exchangeMoteJoinKey` コマンドを使ってマネージャとモート上で更新されます。これは、マネージャがモートに新しいジョイン・キーを渡す際に行われます。モートが更新を確認応答しないと、マネージャはそのACLを更新しません。モートは、APIの`setACL` コマンドまたはCLIの`set acl` コマンドを通して手動でACLに追加できます。

ネットワーク内の全モートのACLを設定する最も簡単な手順は、以下の通りです。

- 共通のジョイン・キーを使ってモートをネットワークに参加させます。
- ネットワーク内のモートごとに、`exchangeMoteJoinKey` コマンドを使って、ネットワーク内のモートごとにACLエントリを設定します。
- 全てのモートが正常にジョイン・キーの交換操作を完了し、ACLに記載された後、マネージャをリセットしてモートの参加を確認します。

4.6 無線プログラミング (OTAP)

外部アプリケーションはマネージャを通してモートと通信し、デバイスのファームウェアをアップグレードできます。これはOver-The-Air-Programming (OTAP) と呼ばれます。SmartMesh SDKは、OTAPを行うために使用可能な、OTAP Communicatorと呼ばれる参考アプリケーションを提供します。詳細は、「[OTAP Communicator documentation](#)」を参照してください。

OTAP操作の基本的な手順は、以下の通りです。

1. 更新されたファームウェアが .otap2 ファイル形式で提供されます。
2. 全モートとのハンドシェイクにより、この更新 (受信リスト) を受け入れるモートが決定し、更新の準備が整えられます。
3. OTAP ファイルがパケットに分割され、全てのデバイスに送信されます。受信リストに記載されたデバイスのみがファイル进行操作します。
4. プロセスは、データが受信され有効であることを確認するために、受信リストに記載された全てのモートを照会します。全て (対象) のモートがファイルを受信するまで、手順2と3を繰り返します。
5. ファイルを受信した全てのモートにコミット・メッセージを送信します。これにより、全モートのフラッシュが新しいファームウェアで書き換えられます。
6. モートがリセットされ、新しいファームウェアを使ってネットワークに再度参加します。

4.7 マネージャの工場出荷時デフォルト設定の復元

マネージャは、CLIコマンドの `exec restore` または API コマンドの `restoreFactoryDefaults` を使用して、工場出荷時の設定に戻すことができます。以下の表に、工場出荷時の設定に戻すことにより影響を受ける CONFIG パラメータのデフォルト値を示します。復元することにより、アクセス制御リストもクリアされます。

パラメータ	コメント	デフォルト値
netid	ネットワーク ID	1229
txpower	AP TX 電力 [dBm]	8
frprofile	フレーム・プロファイル ID	1
maxmotes	モートの最大数 (AP を含む)	33 (100 モート・マネージャはこの状態に戻ります。maxmotes を手動で 101 に設定する必要があります)
basebw	ベース帯域幅—パケット間の期間 [msec]	9000
dnfr_mult	ダウンストリーム・フレーム乗数	1
numparents	親の最小数	2
cca	CCA 値	0
channellist	チャンネル・リスト	7FFF (hex)
autostart	自動開始オン/オフ	1 (オン)
locmode	予約済み	0
bbmode	バックボーン・モード	0 (オフ)
bbsize	バックボーン・フレーム・サイズ	1
pwdviewer*	ビューワ・ユーザーのパスワード	viewer
pwduser*	通常ユーザーのパスワード	user
commjoinkey*	共通結合キー	44 55 53 54 4E 45 54 57 4F 52 4B 53 52 4F 43 4B (hex)
ip6prefix	IP6 アドレス	::
ip6mask	IP6 マスク	FFFF:FFFF:FFFF:FFFF:: (hex)
radiotest	無線テスト	0 (オフ)
bwmult	帯域幅プロビジョニング乗数 [%]	300
onechannel	1 チャンネル・ネットワーク	FF (hex)

5 SmartMesh IP モート

5.1 はじめに

SmartMesh IP モートは、SmartMesh IP ネットワークの「本体」です。モートは、以下を担当します。

- ネットワークとの同期を維持します
- 子孫からのデータを上流に転送します
- マネージャがネットワークの状態を継続して把握するために健全性レポートを生成します
- 切断したパスなどの失敗を示すアラームを生成します
- センサ・アプリケーションにユーザー・インタフェースを提供します

[LTC5800-IPM](#)は、顧客の設計に組み込まれることを目的とする単一チップのソリューションです。[LTP5902-IPM](#)と[LTP5901-IPM](#)は、モジュール方式で認証されているため、無線認証の必要なしで使用できます。

5.1.1 モート設計の手順

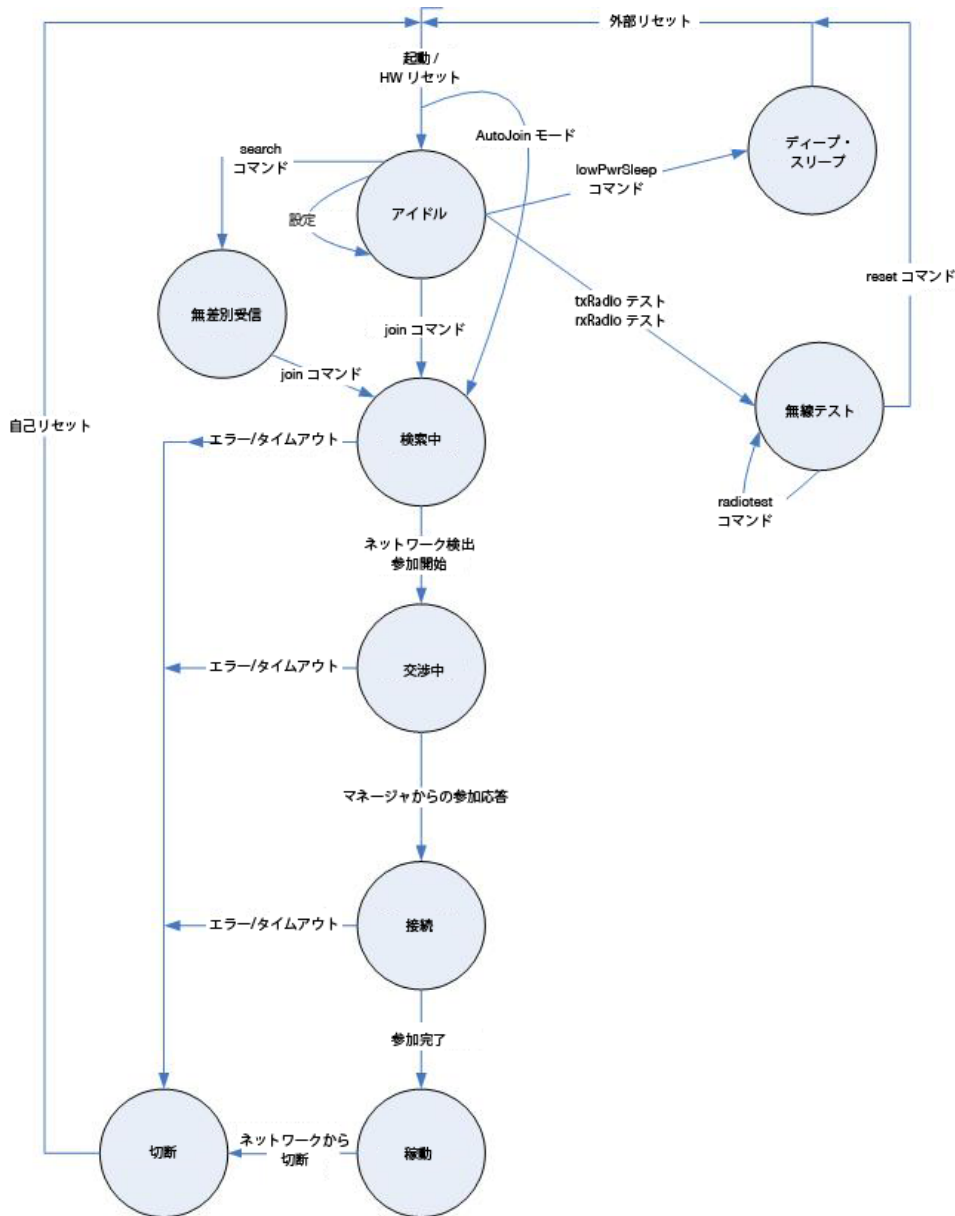
マネージャと同様、モートは多くのネットワーク・タスクを実行しますが、センサ・アプリケーションが行う必要があるタスクは、以下のように比較的わずかです。

- モートの起動イベントに確認応答します
- 参加の前に必要なパラメータ (*joindutycyle* など) を設定します
- *join* API を使用して、モートがネットワークを検索するようにします
- モート状態を監視して、データの受け入れ準備が整っているかを確認します
- データを発行するためにサービスを要求します
- データを送信し、アプリケーション層メッセージに対して応答します。

「[SmartMesh IP Mote API Guide](#)」はモートを制御するその他のコマンドを説明します。「[SmartMesh IP Mote CLI Guide](#)」は、モートのアクティビティを監視するコマンド・ライン・インタフェースについて説明します。

5.2 モート状態マシン

次の状態マシンは、モートの一般的な動作を示しており、ユーザーの参考のために提供されます。一般に、アプリケーションだけが、ネットワークに入るための参加コマンドを発行し、データを送信するAPIコマンドを発行する必要があります。



モートの状態は、以下の通りです。

- **アイドル**—この状態の間は、モートは設定コマンドを受け付けます。この状態は、モートが自動参加に設定されているとスキップされます。
- **ディープ・スリープ**—取り付けられたシリアル・プロセッサから *lowPowerSleep* コマンドを受信すると、モートはディープ・スリープ状態になります。この状態では、デバイスはシリアル・コマンドに応答できなくなり、通常動作を再開するためにはリセットする必要があります。電力消費情報については、モート製品のデータシートを参照してください。
- **無制限受信**—特殊な検索状態。 *search* コマンドにより起動されます。モートは任意のネットワークIDからの通知を受信し、受信した通知を報告します。モートはネットワークに参加しようとはせず、join コマンドを待ってそれが発行されると検索中状態になります。

- **検索中 (非同期検索)** — モードは、ネットワークIDに一致するネットワークを検索しています。その無線受信の時間を設定可能なジョイン・デューティ・サイクルに留めます。
- **同期検索 (非表示)** — 検索の終わりの短時間。デバイスは通知を受信し、ネットワークに同期しています。その無線受信の時間を設定可能なジョイン・デューティ・サイクルに留め、近隣を受信します。
- **交渉中** — デバイスがネットワークへの参加を開始。
- **接続** — デバイスはマネージャからの参加応答を受信し、その応答により設定されています。
- **稼動** — デバイスはネットワークへの参加を完了し、データを送信する準備が整っています。
- **切断** — デバイスはネットワークから切断されました。
- **無線テスト** — この状態では、デバイスはradiotestコマンドを実行しています。通常動作に戻すには、リセットする必要があります。

5.3 参加

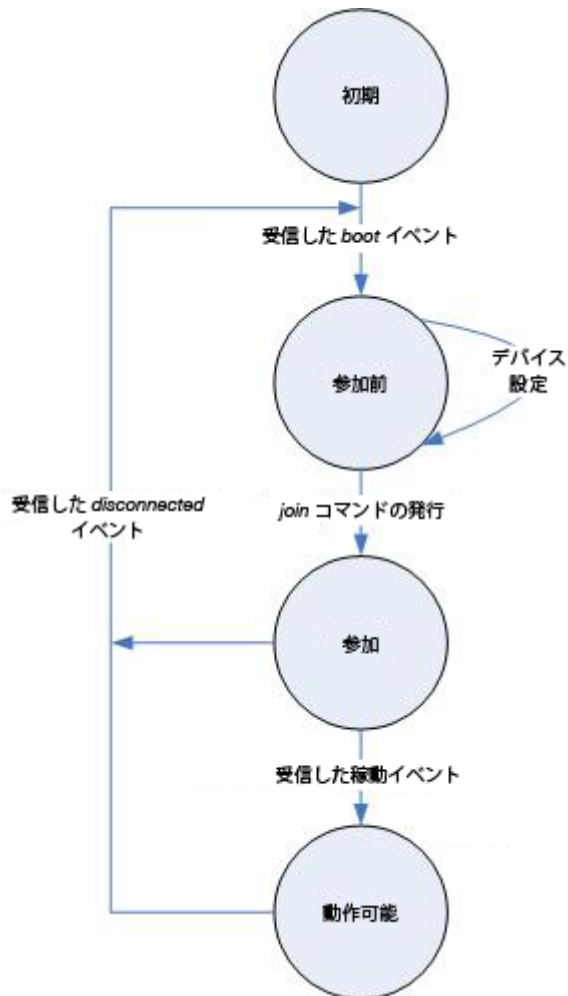
5.3.1 プログラムからの参加

ネットワークに参加するには、デバイスをネットワークに結びつける以下のパラメータをデバイスに設定する必要があります。

- *networkId*
- *joinKey*

これらのパラメータは永続的で、コミッショニング中に一度設定されます。更に、*joinDutyCycle*、*routingMode*、*powerSourceInfo*などの他のパラメータは全て、参加動作に影響し、オプションで設定できます。

クライアント・アプリケーションは、以下の状態マシンを使って、ネットワークに参加する必要があります。



この図では、以下の状態が想定されます。

- **初期**—この状態では、アプリケーションが起動し、初期化しています。低消費電力構成ではありませんが、アプリケーションがモートと通信する準備を終えるまで、モートはハードウェア・リセット状態にしておきます。
- **参加前**—モートが起動したことを *boot* イベントで通知されると、アプリケーションは、複数の *setParameter* API を呼び出すことにより、パラメータの設定を行います。この状態の終わりに、アプリケーションは *join* コマンドを発行します。
- **参加中**—この状態では、モートはネットワークに参加しています。参加に成功したことは、稼働イベント通知を介して報告されます。参加に失敗した場合は、*joinFail* イベント通知を介して報告されます。
- **動作可能**—この状態では、デバイスは参加を完了し、ネットワークの一部となっています。アプリケーションは、帯域幅を要求し、データを送信します。

5.3.2 自動参加

モートは、起動後にネットワークを自動で検索し参加を開始するように設定できます。この設定は、永続的な *autoJoin* パラメータを介して制御されます。この場合、明示的な *join* コマンドは必要ありません。ネットワークID、電源などのパラメータは全て、事前に設定しておく必要があります。

 デバイスを自動参加するように設定すると、*radiotest* 機能は実行できません。

5.3.3 隣接ネットワークへの参加

場合によっては、ネットワークIDが事前に知らされていないこともあります。どのネットワークがデバイスの近くにあるかを知るには、APIコマンドの`search`を使用します。このコマンドは、デバイスを無差別受信モードにします。このモードでは、受信した通知は全て、`advReceived`通知を介して報告します。正しいネットワークIDが確立された後、ユーザーは`setParameter<networkId>`を使ってネットワークIDを設定し、続いて`join`コマンドを実行します。

以下は、モートの起動に続いて行われる手順のまとめです。

1. モートを無差別受信モードにします (`search` コマンド)
2. `advReceived` 通知を処理します。
3. ネットワークIDを設定します (`setParameter<networkId>`)
4. 参加を開始します (`join` コマンド)

5.4 サービス

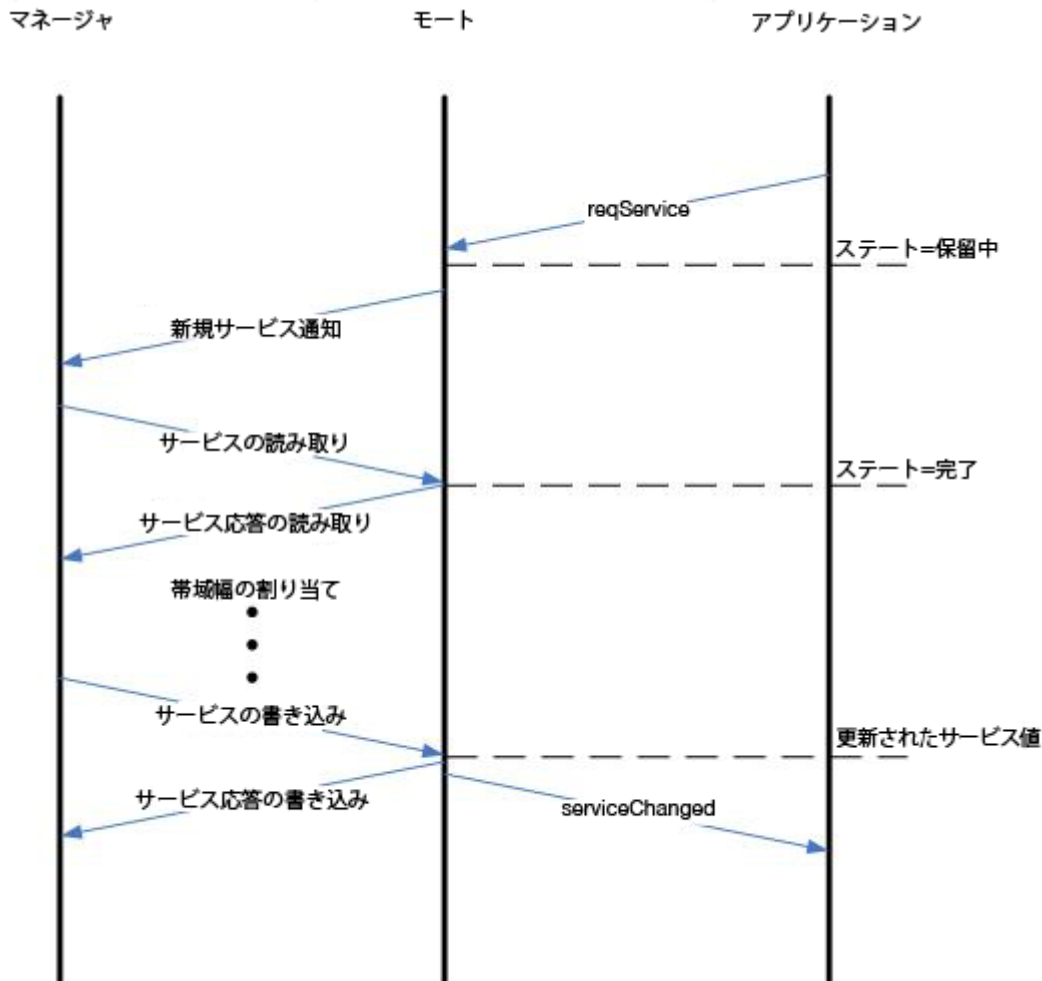
5.4.1 帯域幅の要求

モートは、ネットワークに参加して稼働状態になると、データ・パケットの送信を開始できるようになります。最初、全てのモートは、あらかじめ決められたアップストリーム帯域幅（ベース帯域幅）が与えられます。この帯域幅がモートの発行レートに対して足りない場合、アプリケーションはネットワーク帯域幅の追加を要求することができます。これは、サービスの要求と呼ばれます。割り当てられている帯域幅を知るために、アプリケーションは`getServiceInfo` APIを使用します。モートは、特に明示的に要求しなくてもマネージャとの通信用のベース帯域幅が与えられます。

サービスはメッシュ内の送信先によって区別され、モートが必要とする合計の帯域幅を含みます。送信先ごとに許可されるサービスは1つのみであるため、異なるレートでパケットを生成するアプリケーションは、最大のレートに対応した一つのサービスを要求する必要があります。現在、モートがデータを送信できるメッシュ内送信先はマネージャだけです（メッシュ・アドレス 0xFFFE）。

センサ・アプリケーションが`requestService` APIを使用した時に、モートは保留中のサービス要求についての通知をマネージャに送信します。この時点で、APIを介して返されたサービス状態は保留中になります。マネージャがモートからのサービス要求に応答すると、サービス状態は、完了に変わります。モートからサービス要求を取得した後、マネージャは要求された最新の値を保存し、帯域幅の割り当てが変わるとモートに通知します。アプリケーションは`serviceChanged` 通知を受信します。このような通知は、ネットワーク状態が変化すると、いつでも何度でも発生することに注意してください。アプリケーションは必要な時に`requestService` を使ってサービス要求を変更できます。マネージャは最後の要求を最新のものとして扱います。

次のトランザクション図は、サービス要求においてアプリケーション、モート、およびマネージャの間で何が起きているかを示しています。



5.4.2 バック・オフのメカニズム

アプリケーションは、サービスを要求した後、*sendTo* APIを介してただちにデータの送信を開始できますが、ネットワークが受け付けることの出来る範囲のレートで送信する必要があり、そのためにバック・オフの手続きが必要になります。複数のデータが同じ送信先に送られている場合、アプリケーションは必要なサービスを合計したものを一つのサービスとして要求する必要があります。

次のバックオフ・アルゴリズムを推奨します。

- RC_NO_RESOURCESを受信した場合、パケットの間隔を2倍にします。再度受信した場合、3倍、4倍と順に255倍まで増やします。
- パケット送信が正常に行われるようになった場合には、キューに空きがあることを意味しますので、パケットの間隔を逆に5倍、4倍、...、1倍のように縮めていきます。

5.5 通信

5.5.1 ソケット

ソケットは、モートと他の IP デバイスの間の通信フローのエンドポイントです。モートのソケットは、Berkeleyソケット標準にほぼ基づいています。通信するために、アプリケーションはソケットを開き、それをポートにバインドする必要があります。複数のポートを持つアプリケーションは複数のソケットを開くことができます。

ソケット使用の通常シーケンスは次の通りです。

1. `openSocket` コマンドを呼び出し、通信ソケットを開きます。これによりソケットの `socketID` が得られます。現在のところ、UDPソケットのみがサポートされています。
2. `bindSocket` コマンドを呼び出し、ソケットを `sendTo` コマンドで使用するポート `destPort` にバインドします。
3. `sendTo` コマンドを使って、データを送信します。ペイロードとソケット情報に加えて、`serviceType`、`priority`、`packetId` を指定する必要があります。詳細については、Mote API Guideの `sendTo` を参照ください。現在、帯域幅サービスがサポートされていますが、レイテンシ・サービスはサポートされていません。オープン状態のソケットに対して、`sendTo` は繰り返し行うことができます。
4. 送信先にデータを送信する必要がなくなった時、`closeSocket` コマンドを呼び出します。これにより、ポートへのバインドが外され、ソケットに割り当てられていたメモリが解放されます。各パケット送信の後にソケットを閉じる必要はありません。

5.5.2 UDP ポート割り当て

0xF0B0 ~ 0xF0BF の範囲にある UDP ポートは、メッシュ内部で最も効率よく圧縮されるので、出来る限りこの範囲を使用することによって、使用可能なペイロードが最大化できます。

モート上では、次のポート割り当てが使用されます。

ポート	説明
0xF0B0	マネージャとモートとの間の管理トラフィック
0xF0B1	予約済み (OTAP が使用)
0xF0B2 ~ 0xF0B7	予約済み
0xF0B8 ~ 0xF0BF	アプリケーションが使用可能

マネージャ上では、次のポート割り当てが使用されます。

ポート	説明
0xF0B0	マネージャとモートとの間の管理トラフィック
0xF0B1	予約済み (OTAP が使用)
0xF0B2	予約済み
0xF0B3 ~ 0xF0B7	予約済み
0xF0B8 ~ 0xF0BF	アプリケーションが使用可能

他のポートも使用可能ですが、その場合ペイロードが 3 バイトだけ短くなります。

5.5.3 データの送信と受信

ソケットが作成されると、センサ・アプリケーションは、*sendTo* APIを使用して、マネージャを含むIPv6デバイスにデータを送信することができます。マネージャのIPv6アドレスはFF02::02です。マネージャに送信されたパケットは、マネージャのシリアルAPI上で*data*通知に変換されます。他のIPv6アドレスに送信されたパケットは、マネージャのシリアルAPI上で*ipData*通知に変換されます。

アプリケーションが送信する無線データは高い信頼性（通常は99.9%以上）を有しますが、UDPですのでエンド・ツー・エンドの配信は保証されません。アプリケーションがパケットの紛失を許容できない場合、アプリケーション層の信頼できるメッセージングは顧客のアプリケーションにより提供される必要があります。

アプリケーションは、ソケットが開かれ特定のポート番号にバインドされているポートからのみパケットを受信します。その場合、モートは*receive*通知を使って、そのポートで受信した全てのパケットをセンサ・アプリケーションに転送します。

5.6 イベントとアラーム

モートAPIでは、センサ・アプリケーションがモートの状態を把握するために役に立つ、イベントとアラームが定義されています。アラームは、不揮発性メモリでのエラーやバッファ不足などの現在の状況を表しています。現在のアラームを読み取るために、アプリケーションは*getParameter<moteStatus>* APIを使用します。

これに対して、イベントはモートまたはネットワークの動作における状態の変化を知らせます。イベントの例には、モート起動イベントやアラーム状態の変化があります。アプリケーションは、*setParameter<eventMask>* API呼び出しを使って、どのイベントの通知を受け取るかを指定できます。

5.7 工場出荷時のデフォルト設定

工場出荷時のモートのデフォルト設定は下記の通りです。CLIコマンドの*restore mote*またはモートAPIコマンドの*clearNV*を使用して、工場出荷時の設定に戻すことができます。

パラメータ	デフォルト値
Network Id	1229
Transmit Power	+8dBm
Join Key (16バイト、Hex)	44 55 53 54 4E 45 54 57 4F 52 4B 53 52 4F 43 4B
OTAP Lockout	0 (許可)
Routing Mode	0 (有効)
Join Duty Cycle	DC9003Bの場合、0xFF (100%)。その他は0x40 (25%)
maxStCurrent	0xFFFF (無制限)
minLifetime	0 (無制限)
currentLimit	0 (なし)
Auto Join	off

5.8 電力に関する考察

5.8.1 電源情報

詳細については、`setParameter<powerSrcInfo>` APIを参照してください。

`powerSrcInfo` 文書に記載されたパラメータのうち `maxStCurrent` だけが、SmartMesh IP マネージャの判断に使用されます。リンクを割り当てる際、マネージャはRXとTXリンクがそれぞれ最大長のパケットを受信すると仮定します。そしてマネージャが追加のリンクを割り当てるとモートの消費電流が `maxStCurrent` を超えると判断した場合には、リンクの割り当ては行いません。ネットワークの動作に最小限必要なリンク数があり、`maxStCurrent` をこのしきい値未満に設定しても、モートは最小リンク構成を取るため、この場合に限り `maxStCurrent` が無視されることになります。このしきい値は、ダウンストリーム・フレーム乗数とランダムに選択されたベース・フレーム・サイズにより異なりますが、約30μAになります。

`maxStCurrent` は、バックボーンの動作にも影響します。アップストリーム・バックボーンの場合、電流制限のないライン給電モート（つまり、`maxStCurrent=0xFFFF` のモート）のみがバックボーン・フレーム内でアップストリームRXリンクを持つことができます。ライン給電モートを持つことが、マルチホップ・アップストリーム・バックボーンを構築する唯一の方法です。双方向バックボーンの場合、全てのモートはバックボーンを受信を行う必要があるため、全てのモートに2スロットごとにRXリンクがあります。これは、電力設定に関係なく起こります。

他の電力パラメータは、現在のマネージャでは使われていませんが、将来のバージョンに対応するために、適切に設定しておくことをお勧めします。`minLifetime` パラメータは `maxStCurrent` を補うために使用され、マネージャはどちらかより厳しい方に従います。例えば、`minLifetime` によりモートが10リンク/秒に設定されたが、`maxStCurrent` の設定により12リンク/秒に制限されている場合、10リンク/秒が使用されます。このパラメータは、異なる種類のバッテリーを持つデバイスが混在するネットワークで最も有効です。更に3種類の一時的な電流制限が用意されています。これらは `currentLimit_0`、`dischargePeriod_0`、および `rechargePeriod_0` から成ります。ここでは、`currentLimit_0` を `maxStCurrent` よりも高くすることを目的にしています。例えば、環境発電を利用したデバイスはそのライフタイム全体で40μAの電流を調達できますが、オンボードのコンデンサは同時に1分間に100μAの電流を提供するために使用でき、その後充電に10分かかります。この場合、`currentLimit_0 = 100`、`dischargePeriod_0 = 60`、`rechargePeriod_0 = 600` と設定します。

これらの3つのパラメータにより、モートの電源構成に応じた臨時サービスと帯域幅の設定を可能にする、非常に柔軟な電力プロファイルを記述できます。

5.8.2 ルーティング・モード

電力設定に関わらず、各モートには、`setParameter<routingMode>` APIを介して変更できるルーティング・モードを持っています。

モード `=non-routing` に設定すると、ルーティングを行わなくなります。非ルーティング・モートは子を割り当てられません。通知リンクまたは検出RXリンクが与えられないため、モートの消費電流をその分だけ下げることができます。このモードの設定は、モートが `maxStCurrent=0xFFFF` に設定されている場合でも有効です。全てのモートを非ルーティングに設定すると、必然的にネットワークはアクセス・ポイントを唯一の親とする1ホップのスター・トポロジになります。

モード `=routing` に設定すると、ルーティングが有効になります（デフォルト）。つまり、モートは子を割り当てられます。モートは通知を送信し、他のモートからの検出RXリンクを受信します。最適なネットワークのために全てのモートが子を必要とするとは限らないため、ルーティングを有効にしたモートが全て子を持つとは限りません。子を持たないルーティング可能モートもリーフと呼ばれます。全てのモートがルーティング可能なネットワークでも、少なくとも1つのリーフがあります。

5.8.3 ジョイン・デューティ・サイクル

joinDutyCycle パラメータによって、マイクロプロセッサはジョイン・デューティ・サイクル、つまりモートがネットワークを検索している間のアクティブな受信時間とスリープ時間（低消費電力状態）の比率を制御できます。デフォルトのデューティ・サイクルによって、モートはバッテリー電力を使いすぎることなく妥当な速度でネットワークに参加します。電力消費を増やしてでも参加までの時間を縮めたい場合には、*setParameter<joinDutyCycle>* コマンドを使用して、ジョイン・デューティ・サイクルを 100% まで上げることができます。*setParameter<joinDutyCycle>* コマンドは次の参加にのみ影響することに注意してください。電力消費に関する情報については、モート製品のデータシートを参照してください。このコマンドは、参加プロセス中に何回でも発行して構いません。このコマンドは、モートが**アイドル**および**検索中**状態の場合にのみ有効です。

5.9 マスタとスレーブ

5.9.1 モード動作

モートには参加動作とコマンド処理を制御する2つのモードがあります。

- **マスタ・モード**。このモードでは、モートがコマンド処理と参加動作を担当するアプリケーションを実行します。アプリケーションは、内蔵のアナログ入力および温度センサの読み取りと汎用デジタル入出力ができ、サンプル・パケット生成機能が含まれています（WirelessHART モートでは温度読み取りとパケット生成のみ可能です）。デフォルトでは、アプリケーションは 30 秒ごとに温度レポートを送信します。「[SmartMesh IP Tools Guide](#)」に記載されているように、顧客のアプリケーションは OAP プロトコルを介して**マスタ・モード**のアプリケーションと対話できます。スターター・キット内のモートは全て、デフォルトで**マスタ・モード**に設定されています。**マスタ・モード**では、API は使用できません。
- **スレーブ・モード**。このモードでは、シリアル接続されたセンサ・プロセッサがコマンド処理と参加動作を制御することになります。デフォルトでは、モートは自分からはネットワークに参加しようとはしません。**スレーブ・モード**では API が使用でき、API Explorer などのシリアル・アプリケーションが使用されることを想定しています。SmartMesh IP では、*setParameter* を介して *autojoin* が有効にされている場合、**スレーブ・モード**のモートは、シリアル・アプリケーションが *join* コマンドを発行しなくても自動的にネットワークに参加します。

モードは CLI の *set* コマンドを通して設定でき、リセットされるまで持続します。



マスタ・モードの SmartMesh IP モート (**DC9003**) の場合、モートは自動的に検索を開始するため、起動するとすぐに STATUS_0 LED が点滅し始めます。**スレーブ・モード**では、LED は点灯しません。空のバッテリーと間違えないようにしてください。

SmartMesh WirelessHART モートの場合、STATUS_0 LED はソフトウェアでは使用されません。

DC9003 ボードの LED は、LED_EN ジャンパが短絡されている場合にのみ点灯することに注意してください。

スレーブ・モードへの切り換え

デフォルトでは、スタータ・キット (DC9000 および DC9007) のモートは、**マスタ・モード**に設定されています。現在の設定を読み取るには、USB ケーブルを介してモートをコンピュータに接続し、モート CLI コマンドの `get` を使用します。モートを **スレーブ・モード**に設定するには、モート CLI コマンドの `set` を使用します。

以下のように `get mode` コマンドを使って、現在のモードを表示します。

```
> get mode
master
```

以下のように `set mode` コマンドを使って、**スレーブ・モード**に変更します。

```
> set mode slave
> reset
```

 モード変更を有効にするには、モートをリセットする必要があります。いったん設定すると、モードはリセットされるまで持続します。

マスタ・モードへの切り換え

現在の設定を読み取るには、USB ケーブルを介してモートをコンピュータに接続し、CLI コマンドの `get mode` を使用します。モートを **マスタ・モード**に設定するには、CLI コマンドの `set mode` を使用します。

以下のように `get mode` コマンドを使って、現在のモードを表示します。

```
> get mode
slave
```

以下のように `set mode` コマンドを使って、モートを **マスタ・モード**に設定します。

```
> set mode master
> reset
```

 `set mode` コマンドを有効にするには、モートをリセットする必要があります。いったん設定すると、モードはリセットされるまで持続します。

商標

Eterna、Mote-on-Chip、SmartMesh IP は Dust Networks, Inc. の商標です。Dust Networks のロゴ、Dust、Dust Networks、SmartMesh は Dust Networks, Inc. の登録商標です。LT、LTC、LTM、および  はリニアテクノロジー社の登録商標です。第三者の会社名および製品名は、それぞれ所有者の商標であり、情報提供の目的でのみ使用されています。

著作権

本文書は米国および国際著作権、その他の知的所有権法と工業所有権法により保護されています。リニアテクノロジー社とそのライセンサにより専有されており、限定的ライセンスに基づいて頒布されています。リニアテクノロジー社からの書面による許可なく、本製品またはその一部を、いかなる形態または手段によっても使用、複製、修正、逆アセンブル、逆コンパイル、リバース・エンジニアリング、頒布、または再頒布を行ってはなりません。

限定的権利：米国政府による使用、複製、または開示は、FAR 52.227-14(g) (2)(6/87)と FAR 52.227-19(6/87)、または DFAR 252.227-7015 (b)(6/95)と DFAR 227.7202-3(a)、全ての同等および後継の法令と規制による制限の対象となります。

免責事項

本文書は、明示、黙示を問わず、商品性、特定の目的への適合性に関する暗黙の保証を含む一切の保証なしで「そのまゝの状態」で提供されるものとします。

本書には技術的に不正確な記述またはその他の誤り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ます。

リニアテクノロジー社は、アプリケーション、または製品やサービスの使用により発生する責任を負いません。特にまた、間接的あるいは偶発的損害を含むがそれに限定されず、いかなる責任を負わないものとします。

リニアテクノロジー社の製品は、故障がユーザーへの重大な傷害につながる可能性がかなり高い生命維持装置、デバイス、その他のシステムでの使用、またはその機能停止により生命維持装置や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可能性がかなり高い生命維持装置の重要なコンポーネントとしての使用を目的としていません。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リニアテクノロジー社の顧客は、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直接または間接的に起因する全ての主張、費用、損害、支出、および妥当な額の弁護士費用、また、かかるクレームでリニアテクノロジー社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、リニアテクノロジー社とその職員、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。

リニアテクノロジー社は、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。全ての製品は、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。

リニアテクノロジー社は、リニアテクノロジー社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リニアテクノロジー社の特許、著作権、回路配線権、またはその他のリニアテクノロジー社の知的所有権に従って、明示、黙示を問わず、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リニアテクノロジー社が公開した情報は、その製品またはサービスを使用するためのリニアテクノロジー社からのライセンス、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、またはリニアテクノロジー社の特許または他の知的所有権に従ってリニアテクノロジー社からのライセンスが必要になります。

Dust Networks, Inc は、リニアテクノロジー社の完全所有子会社です。

© Linear Technology Corp. 2012-2013 All Rights Reserved.