

MaCC: WPAN コンフィグレーション機構

滝澤 允[†] 間 博人[†] 斉藤 匡人[†] 戸辺 義人[§] 徳田 英幸^{†‡}

近年, WPAN(Wireless Personal Area Network) という狭範囲無線ネットワークや近距離無線デバイスを搭載したセンサによるセンサネットが現れた. WPAN は PC に加え, 携帯電話や PDA などのモバイル機器や家電などから即時的に構成される. センサネットは高密度に配置されたセンサにより粒度の細かいセンシングを可能にする. 次世代ネットワーク環境ではインターネットとこれら WPAN とセンサネットが融合することが考えられる. インターネットという世界規模の広域ネットワークから WPAN を通じて家電など各機器を制御し, また, WPAN に繋がれたセンサネットに対してクエリを発行する. すなわち, WPAN がインターネットおよびセンサネットと繋がり, これらネットワークが融合する形態である. 本稿では中間に位置する WPAN に焦点を置き, WPAN に適したコンフィグレーション機構 MaCC を提案する. MaCC は WPAN に対してネットワークトポロジ動的構成およびルーティング機能を提供する. 本稿では即時的に構成されたネットワークにおいてルーティングプロトコルとして一般的なフラッディング方式が WPAN に対しては不十分であること, また MaCC はフラッディング方式に比べネットワークに対して低負荷であることを示す.

MaCC: A WPAN Configuration

Makoto Takizawa[†] Hiroto Aida[†] Masato Saito[†]
Yoshito Tobe[§] Hideyuki Tokuda^{†‡}

These days, Wireless Personal Area Network(WPAN) and Sensor networks are emerging. WPAN are short distance wireless ad-hoc networks, composed of PCs, mobile devices such as cell phone and PDA, and consumer electronics. Sensors in sensor networks equipped with short range radio technologies are deployed densely. These sensors are capable of computation, communication and sensing. Sensor networks can make a high level result through merging individual sensor readings and sensing of a fine particle size. In future networks, The Internet will be inter-connected with WPAN and sensor networks. Therefore, we focus on WPAN at the middle position. In this paper, we present Master-driven Connection Control(MaCC), making WPAN configuration. MaCC provides dynamic network-topology formation and routing mechanisms. Furthermore, we show routing protocols by floodings for ad-hoc networks are insufficient for WPAN and MaCC is suitable for WPAN, giving lower load to a whole network.

1. はじめに

通信技術が発展し通信デバイスの小型化, 軽量化が進むことで PC など従来のネットワーク接続機器以外の多くの機器が通信機能を搭載することとなった. PDA などの小型端末や家電, 各種センサが通信機能を有す.

通信端末の増加に伴って, 新たなネットワークが構成される. インターネット, LAN にはじまり, WPAN(Wireless Personal Area Network) やセンサネットが構成される. WPAN は近距離無線による半径十メートル程の無線ネットワークである. PC や PC 周辺機器, 携帯電話や PDA などのモバイル端末に加え家電などから WPAN が構成される.

また, 各種センサがセンサ同士でネットワークを構成し, センサネットとなる. センシング可能範囲が狭いセンサを高密度で配置させることで, 粒度の細かいセンシングを可能とする. 無線通信機能を搭載したセンサとして, UC Berkeley SmartDust⁽⁷⁾, UC Berkeley Mote⁽⁴⁾, Rockwell Science Center による WINS⁽¹²⁾ などがある.

基盤ネットワークであるインターネットと WPAN, センサネットの関係を図示したものが図 1 である. これらネットワークはネットワーク内のノード密度により分類できる. インターネット, WPAN, センサネットの順でノード密度が高くなり, ある範囲に存在する通信端末数が多くなる.

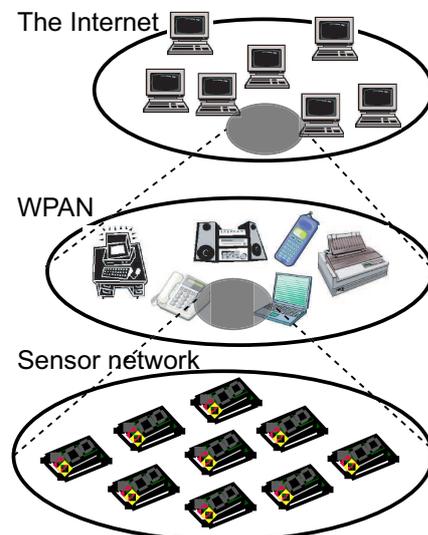


図 1 インターネット, WPAN, センサネットの関係図

ここで重要になってくることは各ネットワークの相互接続である. 各ネットワークの利点を最大限利用するために, 各ネットワークの相互接続が必要になる. インターネットは世界規模での広域コミュニケーションが可能である. WPAN では身の周りのデバイスを相互に接続することで各機器の機能補完が行える. センサの高密度配置により粒度の細かいセンシングがセンサネットとして可能となる. それぞれのネットワークが対象とする範囲の情報をさらに広範囲なネットワークで, または狭範囲のネットワークで応用することで高い不可価値を得ることができる.

[†] 慶應義塾大学大学院 政策・メディア研究科

[‡] 慶應義塾大学 環境情報学部

[§] 東京電機大学 工学部 情報メディア学科

[†] Graduate School of Media and Governance, Keio University

[‡] Faculty of Environmental Information, Keio University

[§] Department of Information Systems and Multimedia Design, Tokyo Denki University

例えば、外出先から携帯電話を使い部屋の空調機を動作させたい場合にまず、部屋にあるセンサに対し室温をセンサさせるクエリを発行する。部屋にあるセンサはそのクエリを受信し室温を測り、結果を返信する。それにより、作動時の設定温度を決定できる。この時空調機は直接外部ネットワークとは接続していない。外部ネットワークに接続された PC などを経由して空調機へ情報が送信される。空調機は周辺に存在する多くのセンサから粒度の細かいデータを取得し、空調の方向を調節する。

また、街中で各ネットワークを融合することで、例えばナビゲーションが実現できる。信号などから WPAN を構成して、携帯電話を即時的に WPAN に組み込む。そこで、携帯電話は周辺のセンサから位置情報を取得して、その位置情報を WPAN 上に伝送する。そうすることで各 WPAN 端末の位置へとナビゲートできる。

本稿では、インターネットとセンサネットワーク間を中継する WPAN に焦点をおく。WPAN により各ネットワークが融合する。各ネットワークに対して拡張を必要とせず、融合を可能にするためには WPAN が柔軟なネットワークである必要がある。動的に即時的にネットワークが構成でき、かつ即時的に構成されたネットワークに対してアクセス透過性を保証する必要がある。しかし、WPAN はネットワークの即時的に構成可能であるが、動的なトポロジ構成やルーティング機能が確立されていない。

そこで、本稿では WPAN コンフィグレーション機構 MaCC を提案する。MaCC は動的なネットワーク構成およびルーティング機能を提供する。オンデマンドで WPAN 上の最短経路を検索し、その最短経路となるネットワークを動的に構成する。

本稿は、第 2 節において具体的な基盤インターネットおよび WPAN、センサネットワークの相互接続の構想を述べる。第 3 節で、本稿で着目する WPAN コンフィグレーション機構 MaCC の設計と実装を示す。また、MaCC のオーバーヘッドに対する分析を示す。最後第 4 節において結論および今後の課題を述べる。

2. 次世代ネットワーク構成

本節では、インターネットと WPAN およびセンサネットワークが繋がったネットワーク構成を述べる。WPAN とセンサネットワーク、WPAN とインターネットのそれぞれが融合したネットワーク構成に加え、WPAN を介してインターネットとセンサネットワークの融合した構成を示す。

2.1 WPAN とセンサネットワーク

多くのセンサを配置してセンサネットワークを構成する場合、WPAN 端末をセンサネットワークにおけるビーコンとして動作させる。センサからのデータは WPAN 端末で収集かつ集約され、集約されたデータは WPAN 上で伝送される。これを図 2 に示す。

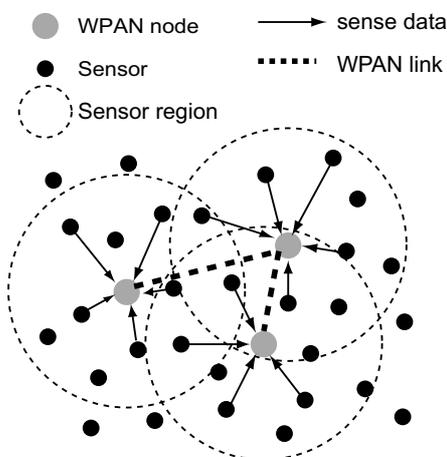


図 2 WPAN とセンサネットワークの融合

センサネットワークでは、そのセンサしたデータに加え位置情報も必

要となる。不規則に配置されたセンサが位置特定を行うアルゴリズムの多くは、位置情報を先験的に保持しているセンサとしてビーコンを必要とする。センサに対して GPS を搭載するなど機能を追加することは逆に小型化などに対して妨げになる。よって、GPS 機能を搭載した携帯電話や位置がある程度固定された家電などの WPAN 端末がビーコンに適している。

また、センサしたデータを処理する端末はセンサ以外であるべきである。センサは小型かつ軽量であり、バッテリー駆動であるため必要以上に電力を消費するデータ処理には適さない。したがって、電力が供給される WPAN 端末へデータを集中させて処理させることは適切である。

よって、センサする各リージョンを WPAN 端末 1 つと対応させる。センサからの情報は WPAN 上で交換される。センサは、センシングおよびデータを近隣の WPAN 端末へ伝送する機能のみ有す。データ処理および利用者へのデータ送信は WPAN 端末が行う。センサにオーバーヘッドとなるデータ処理およびルーティングなどの機能を排除し、WPAN と融合することでその機能を補完できる。

2.2 WPAN とインターネット

WPAN は少数のアクセスポイントを通してインターネットへ繋がる構成となる。

例えば家庭およびオフィス内の機器を外部から制御するために、インターネットとの接続性が必要となる。そこで、ゲートウェイとして動作する 1 台を経由して、各 WPAN 端末へとアクセスすることになる。

ゲートウェイより内側は、WPAN 上でデータが交換される。家庭、オフィス環境では従来のネットワークではなく、WPAN が適切である。例えば、ゲートウェイを無線 LAN アクセスポイントおよび DHCP サーバとしてアドレスを動的に与える環境を考える。その場合、確かに通信端末数の変動や端末の移動に柔軟に対応可能である。しかし、アクセスポイントのようなインフラ機器を必要とする場合、すべての通信がアクセスポイントを経由することになる。家庭およびオフィスでは電波を遮断する障害物も多いため、各通信端末の位置に制約が生じる。しかし、アドホックネットワークとして構成される WPAN はインフラ非依存であるから、物理的な位置の制約を格段に低くできる。また、PC からプリンタへのデータ送信、PDA と PC 間のデータ交換などに対してポイント・ツー・ポイント通信を可能とする WPAN は利便性が高い。インフラ機器に依らず即時的に通信を行えることは重要な要素である。

また、街中で信号や電灯、および各建物へ無線デバイスを搭載することで WPAN を構成できる。ここでも WPAN が有効である。すべてをインターネット端末とするならばインフラ依存なため敷設コストが高くなる。携帯電話の基地局や物理的な配線が必要となり利便性が損なわれる。インターネットへのゲートウェイは携帯電話がある特定の建物となる。

2.3 インターネット、WPAN、センサネットワーク

図 3 にインターネットと WPAN およびセンサネットワークが融合した形態の概念図を示す。図 3 では家庭内を想定している。

各ネットワークに対し拡張を行うのではなく、それらを融合させることで例えばセンサからの情報をより広範囲で利用できる。センサはデバイスの制限上、処理量を低く抑える必要がある。そこで、WPAN と融合することでデータ処理、ルーティングを含む機能を WPAN へ委託する。また、さらにインターネットにより世界規模で情報を利用できるために、WPAN とインターネットを融合させる。融合の形態は WPAN がインターネットとセンサネットワークそれぞれと繋がる形となる。街中ネットワークでも同様にインターネットおよびセンサネットワークは WPAN を中継してそれぞれに到達する。

WPAN は、センサからのデータを広範囲に伝送するために、また外部ネットワークからの情報を WPAN 内へ伝送するために、ルーティング機能が必要となる。また、WPAN 上のルーティングは WPAN のトポロジに依存するため、どの端末間のコネクショ

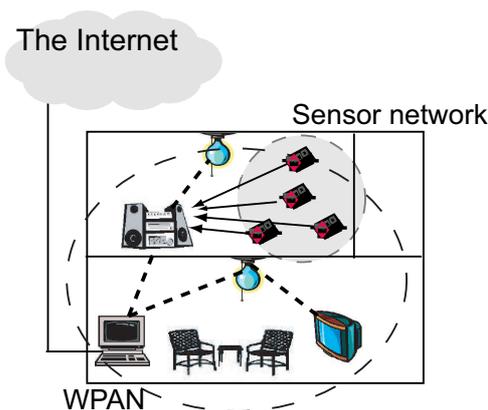


図 3 次世代ネットワーク構想

ンを確立すべきかという問題もある。

3. WPAN コンフィグレーション機構

本稿では WPAN に焦点を絞り、WPAN のネットワークポロジの動的構成およびルーティング機能を提案する。本節において、提案する WPAN コンフィグレーション機構 MaCC の設計および実装を述べる。また、MaCC に対する性能分析を示す。

3.1 設計

本小節において MaCC の設計を述べる。はじめに WPAN の特徴と対象とする WPAN 環境を述べ、次に動作概要と MaCC の 3 機能を概説する。その後各 3 機能に対して詳説する。

WPAN は 1 台の管理ノード (マスタ) に対し 1 台以上のノード (スレーブ) が属する形をとり、以降本稿でマスタ駆動型ネットワークと呼ぶ。1 台のマスタに対し 1 つのネットワークが存在し、それがネットワーク最小単位となる。これをピコネットと呼ぶ。ピコネット内の通信はすべてマスタにより管理される。それにより通信品質の維持およびデータの衝突回避を実現している。WPAN におけるコネクション状態とはノード間でマスタ/スレーブ関係を構築した状態である。ノード間の通信はコネクション状態へ移行してはじめて可能となる。

本稿で対象とする WPAN は家庭およびオフィスなどの環境で構成される。デスクトップ PC、ノート PC、携帯電話、家庭用電話機など従来の通信機器に加え、ライトなどの機器、およびプリンタ、テレビ、ビデオデッキ、各種オーディオ機器がネットワークを構成する。そこで、家庭およびオフィスでは WPAN は静的な端末が比較的多くなる。よって MaCC においてノードは静的であると前提をおく。しかし、ノート PC や PDA、携帯電話などは電源オフや移動することがある。そのため MaCC は少数ノードの電源オフや移動によるネットワークの変動に対応する再構成機能を有す。

動作概要

MaCC では、各ノードが自ノードから 1 ホップ通信可能であるノードの一覧 (以降、隣接情報と表記) を保持し、その隣接情報がネットワーク内で階層型集中管理される。各ノードはその保持する隣接情報をもとに通信相手との最短経路を割り出す。もし自ノードで保持する隣接情報から経路が見つからなければ他ノードに経路要求を送信する。その結果得た最短経路を用いてソースルーティングにより、通信を開始する。MaCC では、1) はじめにリレーションネットワークというネットワークを構成する。2) その上であるノードから経路検索要求が発行されると、リレーションネットワークに従って要求パケットが流れ、結果が返信される。3) 結果として経路を取得後、通信経路を確立する。以下これら 3 機能を概説する。

- (1) リレーションネットワーク構成機能
リレーションネットワークというネットワークを木構造で構成す

る。隣接情報の階層型集中管理のために、ネットワークを階層型の木構造にする。リレーションネットワークでは各ノード間はコネクション状態ではなく、リレーションのみ確立している。リレーションとは、非コネクション状態とコネクション状態の中間の状態であり、概念的には図 4 となる。リレーションは MaCC 独自で定義した用語であり、また後述する。

- (2) 最短経路検索機能
リレーションネットワーク内において、ある 2 点間で通信要求が発生した時点でその 2 点間最短経路を検索し、経路要求元へ検索結果を与える。この時最短経路とは最少ホップ数経路を指す。
- (3) コネクション確立機能
経路検索機能から通信元が経路を取得し、通信元が宛先へリンクを確立するパケットを送信する。パケットに含まれる経路情報から各ノードがコネクションを確立する。

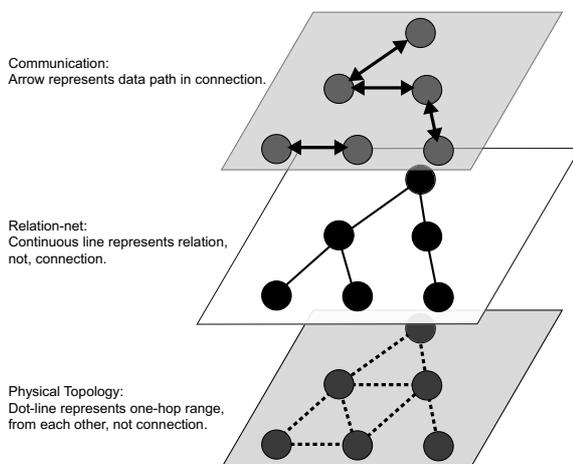


図 4 物理的トポロジとリレーションネットワーク、およびデータリンクの関係

リレーションネットワーク構成機能

本機能はリレーションネットワークとして階層型木構造ネットワークを構成する。本稿ではリレーションを「2 つのノードにおいて親-子関係を構築し、互いのアドレスを保持している状態」と定義する。リレーション確立段階では、互いにアドレスを通知して親-子関係を構築するのみであり、かつコネクション状態以前であり通信はできない。以降本稿で親ノードを PARENT、子ノードを CHILD と表記する。リレーションネットワークとはネットワーク上で隣り合うノードがリレーション状態のネットワークである。

リレーションネットワークの例を図 5 に示す。実線で直接結ばれた 2 つのノードはリレーションを確立している。リレーションネットワークでは、他ノードとリレーションを確立していないノードは FREE、木構造の根 (ルート) にあたるノードは ROOT、末端ノードは LEAF であり、それら以外を BRANCH とする。図 5 では J が FREE であり、A が ROOT、C、E、F、G、I が LEAF、B、D、H が BRANCH である。リレーションネットワーク毎に ROOT が 1 つのみ存在する。階層構造であるリレーションネットワークにおいて ROOT を最上位層ノードとし、各ノードに対する直接の上位層ノードは PARENT である。リレーションネットワーク内では ROOT 以外のノードすべてがただ 1 つの PARENT を有す。図 5 では F は B を、G は D を PARENT とし、A は B、C、D の PARENT である。

次に MaCC がリレーションネットワークを木構造に構成するアルゴリズムを述べる。MaCC は、Bluetooth¹⁰⁾ ネットワークのフォーマションプロトコルである TSF (Tree Scatternet Formation)⁹⁾ のアルゴリズムを利用する。表 1 にノードの各状態 (FREE、ROOT、LEAF、BRANCH) におけるリレーション確立規則を示す。Non-ROOT は LEAF および BRANCH である。表 1 のリレーション確立規則により木構造のネットワークを構成できる。以下に本ア

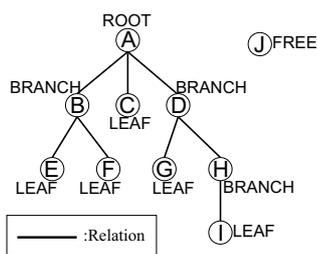


図 5 リレーションネットワーク構成例

ルゴリズムが木構造を構成することを証明する。

証明

木 T 内で閉路が存在しないことを示す。表 1 から、Non-ROOT 間はリレーション確立不可である。また単一の木には ROOT は 1 つである。よって T は閉路を含まない。

MaCC によるリレーションネットワークにおいて、頂点の数が $n_1 (\geq 1)$ の木を T_1 、また頂点の数が $n_2 (\geq 1)$ の木を T_2 とする。 $E(T)$ は木 T の辺の集合とする。 T_1, T_2 は木構造であるから、各々の辺の数は $n_1 - 1, n_2 - 1$ である。この時、 T_1 と T_2 がリレーションを確立する場合を考える。 $n_1 = n_2 = 1$ の時、閉路が存在しないことは自明である。 $n_1 \geq 2$ かつ $n_2 \geq 2$ の場合、表 1 の規則により ROOT 間でリレーションを確立する。よって、 T_1 と T_2 がリレーションにより繋がった木を T_3 とすると、 T_3 の辺の数が、

$$|E(T_3)| = (n_1 - 1) + (n_2 - 1) + 1 = (n_1 + n_2) - 1$$

となる。よって、MaCC により構築されるリレーションネットワークは木構造である。(証明終り)

表 1 リレーション確立規則 (1: 可, 0: 不可)

State	ROOT	Non-ROOT	FREE
ROOT	1	0	0
Non-ROOT	0	0	1
FREE	0	1	1

図 6 はノードの 4 つの状態の状態遷移図である。ノードの状態はリレーション確立後に遷移し、次の状態はそのリレーションにおける PARENT か CHILD によって決まる。状態遷移後、各状態に応じてノードの動作変更を行う。具体的にはより下位層のプロトコルに対し、リレーション確立規則を満たす設定を施すか、または MaCC での処理動作を変更する。下位層プロトコルへ設定を行うことで MaCC の処理を省くことが可能となる。破線は、リレーションネットワーク再構成時における状態遷移を示す。再構成の状態遷移は、すべての CHILD を失うか、または PARENT を失うことで起こる。

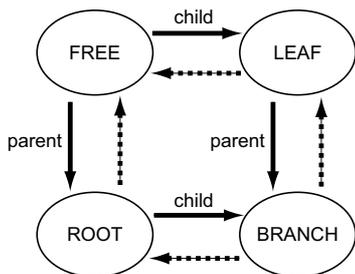


図 6 状態遷移図

以上がリレーションネットワーク構成機能の動作内容である。本機能により WPAN で階層型木構造のリレーションネットワークが構成される。

最短経路検索機能

MaCC 最短経路検索機能は、リレーションネットワーク内において、任意の 2 点間の通信開始時に通信ノード間を繋ぐ最少ホップ数経路を与える。以下、本機能を詳説する。

経路検索では各ノードが保持する隣接情報を使用する。リレーションネットワークではこの隣接情報は階層型集中管理される。その階層型集中管理の例を図 7 に示す。各ノードは一定の時間間隔で 1 ホップで到達可能なノードを探し、その結果を隣接情報として保持する。隣接情報は図 7 にあるように隣接リスト表現されて保持される。隣接リストは先頭ノードとそれに隣接する (1 ホップ到達可能) ノードから成る。各ノードは隣接リストを自ノードの PARENT に送信する。図 7 で例えると、E は E に対する PARENT である B へ送信し、B は送信されてきた隣接リストをマージし、A へ送信する。A は PARENT が存在しない ROOT であるため、隣接リストを他ノードへ送信しない。各ノードが PARENT へ隣接情報を送信することで、結果 ROOT ノードはネットワーク全体の隣接リストを保持することになる。リレーションネットワークは各ノード間にコネクション状態ではないことを先に述べた。そのため隣接リストを送信する場合は該当ノード間で一時的にコネクションを確立する。

最短経路検索機能はさらに 2 つに分けることができる。1) 隣接リストから最短経路を算出するアルゴリズム部と、2) 経路要求 (応答) メッセージのネットワーク内での流れである。先に最短経路アルゴリズムについて述べ、次に経路要求 (応答) 手順を述べる。

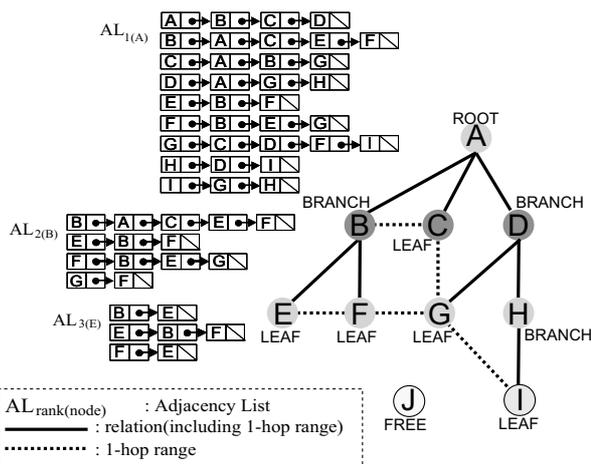


図 7 隣接情報 (隣接リスト) の階層型集中管理

最短経路アルゴリズム

MaCC は最短経路アルゴリズムとしてグラフ理論における Dijkstra のアルゴリズム¹¹⁾を用いた。Dijkstra のアルゴリズムは隣接リストを入力値とすることが仮定されている。そのため MaCC は隣接リストとして情報を保持している。各ノードが保持する隣接情報を Dijkstra のアルゴリズムに投入し、最短経路およびその経路のホップ数を得る。

経路要求 (応答) 手順

経路要求は通信を開始するノードが発行する。経路要求後、検索結果である最短経路は通信を開始するノードのみ取得する。経路上のノードへは通知されない。

経路要求および経路応答の流れは以下ようになる。通信を開始するノードがはじめに自身の隣接リストから経路検索を行う。経路検索が失敗すると、そのノードは PARENT へ経路要求を送信する。経路要求を受信したノードは経路検索を行い、成功であれば結果を返信し、失敗すればさらにそのノードの PARENT へ経路要求を送信する。このように経路要求は再帰的に送信される。経路検索が成功した時点で応答として

経路結果が通信開始ノードへ返信される。また、FREE および ROOT は要求を送信すべきマスタがないため、自ノードで保持する隣接リストから経路が発見できなければ、経路は存在しないというエラーを発行する。経路応答も同様に再帰的に送信される。各ノードは直接要求を送信してきた CHILD へ経路応答を返信する。以上が経路要求メッセージおよび経路応答メッセージの流れである。

MaCC では経路要求 (応答) メッセージおよび隣接リストの送受信は上位層へまたは下位層へ行われる。その際、層を越えることはない。すべて1つ上位、1つ下位へとメッセージは流れ、複数の層に渡るメッセージであっても必ず中間の層を経由することになる。

例として図 7 において E が D へ通信を開始する場合をあげる。はじめ E は保持する $AL_{3(E)}$ から経路検索を行う。検索に失敗し、1つ上位に位置する B へ経路要求を送信する。次に経路要求を受信した B は $AL_{2(B)}$ から経路検索を行う。そこでまた経路が発見できない場合、さらに上位の A へ経路要求を送信し、A が $AL_{1(A)}$ から経路検索を行う。経路結果が算出されると A は B へ結果を返信し、それを受信した B は E へ検索結果を返信する。以上の手順を経て、E は D への最短経路を得る。図 8、図 9 はそれぞれ経路要求、経路応答のフローである。

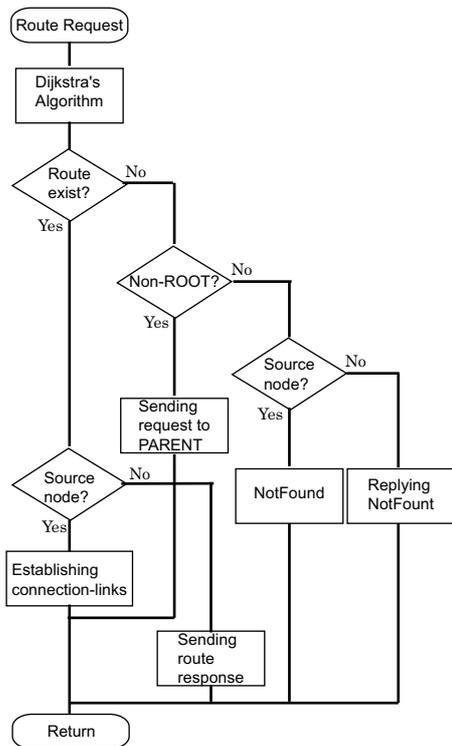


図 8 経路要求フロー

コネクション確立機能

コネクション状態にないリレーションネットにおいて経路検索から得た最短経路を繋ぐノード間のコネクションを確立する。最短経路は各ノードから1ホップ到達可能なノードから成る隣接リストをもとに割り出されるため、コネクションを確立すべき経路はリレーションネットの木構造ネットワークに依らない。

MaCC では通信経路は通信開始ノードへのみ返信されるため、実際の通信においてはソースルーティングを行う。経路上のノードは通信パケットを含む経路情報から次に転送すべきノードを決定する。その際各ノードは転送先ノードとコネクションを確立しているかを調べる。コネクションが確立していなければコネクシ

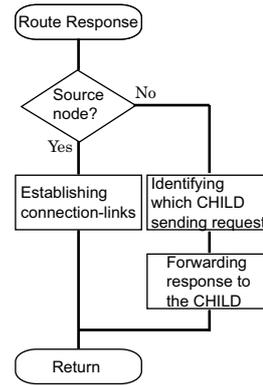


図 9 経路応答フロー

ョン確立を行うということ各ノードで繰り返す。すなわち、コネクション確立用のパケットを用意せず、通信パケットがコネクション確立を兼ねる。

WPAN においてコネクションを確立することはピコネットを形成することである。単一ピコネットで十分でなければ、複数ピコネットがノードを共有してネットワークを拡張させる。共有ノードが異なるピコネットに属するノード間通信の中継を行う。共有するノードは各ピコネットにおいてスレーブとして動作しているノードとする。このようにスレーブノードを共有することでネットワークを拡張した際のピコネット数を最小限に抑えることができる。ピコネット間転送のオーバーヘッドはピコネット内転送のオーバーヘッドより大きいいため、ピコネット数は少なく抑えることで経路上のオーバーヘッドを抑えることができる。

例として、[E-F-G-D] という最短経路が見つかった場合を図 10 に示す。通信元 E はピコネット 1 のスレーブ、ピコネット 1, 2 それぞれのマスタが F, D となり、G がピコネット 1, 2 両方のスレーブとなる。このように通信元がスレーブとしてはじまり、以降交互にマスタ、スレーブとし、ネットワークにおけるピコネット数を最小限に抑える。

以上が MaCC の有する機能である。MaCC はこの 3 つの機能に関してネットワークの状態に依存しない。すなわちネットワーク内の全ノードがネットワーク構成を把握する必要はない。自ノードで処理を行い、必要に応じて PARENT へメッセージを送信する。すなわち自ノードが保持する情報のみで動作判断できる。よって、ネットワーク内でどのノードがダウンしたとしても MaCC 自体は機能停止しない。これは木構造における ROOT がダウンしたとしても同様である。最適なネットワークシステムとするならば各ノードがネットワーク状態から判断すべきである。しかし、WPAN のように必ずしも一定ではないネットワークでは各ノードが自律分散的に動作する必要がある。よって、MaCC においても最適であることより、自律分散処理を優先した。

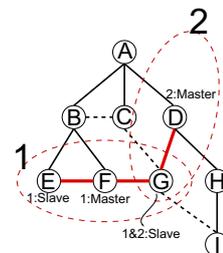


図 10 ED 間経路のピコネット形成例

3.2 プロトタイプ実装

本小節では Bluetooth 上での MaCC のプロトタイプ実装につ

いて述べる．Bluetooth は現状における Bluetooth 規格の確立や Bluetooth 搭載製品の普及から，主要な WPAN の通信技術となることが予想される．よって本稿では WPAN 通信技術として Bluetooth 技術を用いて実装を行った．はじめに実装環境および MaCC プロトタイプ実装の概要を述べる．次に Bluetooth 依存の実装について，および MaCC で用いるメッセージについて述べる．

実装環境および MaCC プロトタイプ

MaCC プロトタイプ実装は，CPU 866 MHz，OS が Redhat7.3(kernel-2.4.18) のノート PC 上で行った．また，Linux 用 Bluetooth プロトコルスタックとして BlueZ⁸⁾ を用い，BlueZ が提供する API をもとに実装を行った．Bluetooth デバイスは，Brain Boxes BL-554 USB デバイスを使用した．

本稿では MaCC をアプリケーション層でのデーモンプロセスとして作成した．アプリケーション層から Bluetooth L2CAP(Logical Link Control and Adaptation Protocol) 層およびベースバンド層へのインタフェースである HCI(Host Controller Interface) を制御する．ルーティングもアプリケーション層で行う．Bluetooth 上での MaCC において，ノード識別子は IEEE802 仕様に準拠した一意の 48 ビット Bluetooth アドレスを用いた．

本プロトタイプ実装では通信を行うアプリケーションと MaCC のインタフェースに UNIX ドメインソケットを用いた．任意のアプリケーションは MaCC が指定する UNIX ドメインソケットにデータを書き込み，MaCC はアプリケーションから受けとったデータを宛先へ送信する．

Bluetooth 上への実装

Bluetooth は通信を行うためにマスタ/スレーブ関係を構築し，物理層でのコネクションを確立する必要がある．このコネクション確立までに 2 つの過程を経る．それは問い合わせ (Inquiry) と呼び出し (Page) である．図 11 と図 12 にそれぞれ問い合わせと呼び出しのシーケンスを示す．

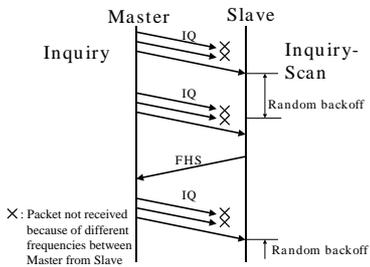


図 11 問い合わせ (Inquiry) 手続きのシーケンス

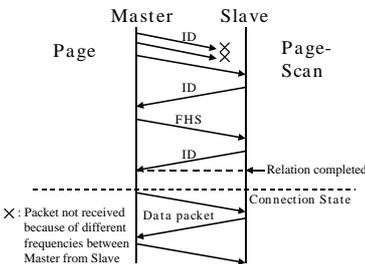


図 12 呼び出し (Page) 手続きのシーケンス

Bluetooth では問い合わせおよび呼び出しにおいて，要求パケットを送信するノードがマスタとなり，要求パケットをスキャンし受信するノードがスレーブとなる．スレーブは問い合わせ要求パケットを受信すると応答パケット (FHS(Frequency Hop Synchronization) パケット) をマスタへ返信する．FHS パケットは Bluetooth アドレスを含んでいるためマスタはスレーブの Bluetooth アドレスを得る．また，問い合わせ要求はブロードキャスト

とされるため，マスタは周辺に存在するスレーブとなるノードすべての Bluetooth アドレスを得る．

一方，呼び出しはユニキャストで行われる．呼び出し要求は問い合わせ段階で FHS パケットを返信したスレーブへ送信される．マスタはスレーブが返信する呼び出し要求パケット受信確認を受け取ると，FHS パケットをスレーブへ送信する．よってスレーブはマスタの Bluetooth アドレスを得る．以上の問い合わせと呼び出しによりマスタとスレーブは互いの Bluetooth アドレスを通知し，互いの Bluetooth アドレスを取得する．したがって，Bluetooth においてリレーション定義を満たすためには，問い合わせ手続きを経て呼び出し手続きを完了する必要がある．

しかし，Bluetooth 標準仕様から呼び出し手続きが独立して提供されておらず，接続手続きに含まれている．それに加えて，MaCC では隣接リストを PARENT へ送信する必要がある．よって各ノードはリレーション確立の判断を隣接リストの要求 (AL_REQ) パケットおよび応答 (AL_RES) パケット受信後に行う．また同様に各ノードのリレーションネットにおける状態遷移も AL_REQ パケットおよび AL_RES パケット受信後に行う．AL_REQ パケットを受信したノードは CHILD としてリレーションを確立し，状態遷移アルゴリズムに従って状態遷移する．次に AL_RES パケットを PARENT となるべきノードへ返信する．AL_RES パケットを受信したノードは PARENT としてリレーションの確立および状態遷移を行う．また受信した隣接リストを既に保持している隣接リストへマージする．また，AL_RES パケット受信後，自ノードに対する PARENT が存在すればその PARENT へマージした隣接リストを含む AL_RES パケットを送信する．PARENT は AL_RES パケット受信後 CHILD とのコネクションを切る．

AL_REQ パケットは，ノードが BlueZ の提供する hci_inquiry 関数によって周辺のスキャン状態ノードの発見後に送信される．発見されたノードからその時点ではじめて発見されたノードのみ AL_REQ パケットを送信する．図 13 は AL_REQ パケットおよび AL_RES パケットのパケットフォーマットである．

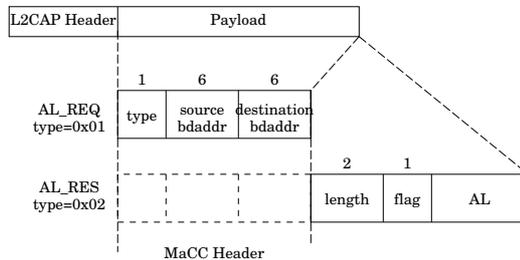


図 13 AL_REQ パケット AL_RES パケットのフォーマット図

MaCC で用いるパケットはすべて L2CAP パケットとして送受信を行う．そこで MaCC は L2CAP ヘッダに続くペイロード部に新たにヘッダとして MaCC ヘッダを加える．MaCC ヘッダはパケットタイプを示す type フィールド，および送信元，宛先の Bluetooth アドレスを含む．MaCC ヘッダは MaCC で用いるパケットすべてに付加されている．

AL_REQ パケットは type フィールドに AL_REQ パケットであることを示す値が格納された MaCC ヘッダのみのパケットである．AL_RES パケットは MaCC ヘッダ以降に，length, flag, AL フィールドを含む．length は AL フィールド長を示す．AL フィールドには隣接リストが格納され，その隣接リストを追加または削除するかを flag フィールドで指定する．

図 14 は経路要求 (R_REQ) パケットおよび (R_RES) パケットのフォーマットである．

R_REQ パケットは MaCC ヘッダの後ろに要求する経路の始点 (from) および終点 (to) の Bluetooth アドレスを含む．R_RES パケットは，flag フィールドが経路検索の成功または失敗を示し，R_REQ パケット同様に経路の始点および終点を含む．weight フィールドは検索結果の経路におけるホップ数を示し，route フィールド

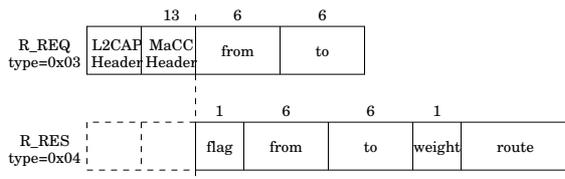


図 14 R_REQ パケット R_RES パケットのフォーマット図

には from および to を除いた経路上の Bluetooth アドレスが経路順に格納されている。

MaCC は通信にソースルーティングを行う。よって、MaCC 上で転送されるパケットには経路情報が含まれる。転送を行う各ノードはこの経路情報を参照し、転送先を決定する。図 15 はデータパケットのフォーマットである。

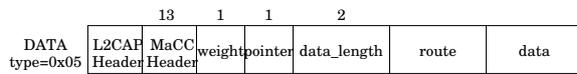


図 15 データパケットフォーマット

MaCC ヘッダの type フィールドにはデータパケットであることを示す値を格納する。以降に続く weight フィールドは終点までのホップ数、pointer フィールドは route フィールドに格納されている Bluetooth アドレス列の中の次ホップ先 Bluetooth アドレスを示す値である。実際には pointer フィールドの値はホップごとに 1 ずつインクリメントされる整数値である。よって次ホップ先の Bluetooth アドレスのポインタを得るには route フィールドの先頭に pointer $\times 6$ (byte) を加える。data_length フィールドは data フィールド長である。route フィールドには経路上の Bluetooth アドレスが始点および終点を除き経路順に配置されている。始点および終点の Bluetooth アドレスは MaCC ヘッダに格納されている。MaCC ヘッダの値はデータパケットにおいては中継ノードによって変更されることはない。

データパケットを受信したノードは pointer と weight フィールドから自ノード宛か転送すべきかを判断し、転送する場合は pointer フィールドから次ホップ先を決定する。

3.3 性能分析

本小節では MaCC の性能を分析する。MaCC のオーバーヘッドとして経路検索に要する実行時間およびネットワーク全体でのデータ送信回数を示す。

一般にアドホックネットワーク (MANET⁵⁾) で用いられるルーティングプロトコル (DSR³⁾, AODV⁶⁾) がフラッディング方式を採用しており、同様に Bluetooth ネットワークに対してもフラッディング方式によるルーティングプロトコル (Routing Vector Method (RVM)¹⁾, Bluetooth Routing Protocol²⁾) が提案されている。しかし、フラッディング方式は WPAN に対して不十分である。そこで加えて、MaCC がより有用であることを示す。

オーバーヘッド分析

オーバーヘッドとして MaCC の経路検索に要する実行時間、経路検索時の全体のデータ送信回数を分析する。実行時間はあるノードの経路要求から経路結果の受信までとする。

経路検索に要する実行時間

MaCC の実行時間は、パケット送受信の通信時間と経路探索アルゴリズム (Dijkstra のアルゴリズム) に要する時間、およびリレーション状態からコネクションに要する時間の合計で表す。通信時間として 1 ホップ往復時間を rtt とする。Dijkstra のアルゴリズム実行時間を $D(|V|)$ とする。 V は頂点の集合である。コネクションに要する時間を C とする。また、結果として得た経路のホップ数を ω とする。

すべてのノードが互いに隣接しているならば、一回の $D(|V|)$ または隣接リストの走査で経路検索が完了する。 $D(|V|)$ は

$O(\frac{3|V|^2-|V|}{2})$ となる。これはネットワークをグラフと考えた場合完全グラフとなり各頂点であるノードの次数が最大の場合である。一般にはつぎの実行時間 T_ω を要する。

$$T_\omega = \frac{\sum_{i=0}^{\omega} \{i \cdot rtt + i \cdot C + \sum_{j=0}^i D(\Theta^{\omega-j}(|V|))\}}{\omega + 1} \frac{\omega + 1}{(rtt + C + D(\Theta^{\omega}(|V|)))} \quad (1)$$

$\Theta^n(v)$ は、経路要求を行ったノードからリレーションネットに従って n 階層上のノードを根とするサブツリーのノード数である。上式は一般化した平均実行時間である。一般に同一ホップ数の経路を検索する場合、どのノードが要求を行うかによって実行時間は異なる。ROOT に近いノードであればあるほど Dijkstra のアルゴリズムにより検索成功率が上がり、通信時間は減少する。例として図 16 をあげる。これは 3 ホップ数の経路を要求した場合のどのノードから要求があるかの全パターンである。下の数字は要求に必要な通信の数である。塗りつぶされたノードにおいて Dijkstra のアルゴリズムが実行される。

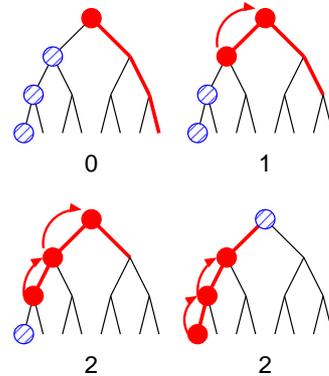


図 16 3 ホップ経路要求例

フラッディング方式と比較した場合、通信時間はより小さくできる。しかし、コネクションにある程度時間を要する。しかし、問い合わせにより得た Bluetooth アドレスなどの情報を利用することで、コネクションに要する時間は問い合わせに比べ十分小さくなる。また、Dijkstra のアルゴリズム上ノード数が増加すると、ほぼ 2 乗スケールで実行時間が増加する。しかし、家庭およびオフィスなどの WPAN 端末数はインターネットのように莫大な数ではないため、数十台程度であれば十分機能すると考えられる。要求に 2 ホップ、応答に 2 ホップ要する検索において検索実行時間は 1 秒から 3 秒ほどであった。 rtt が 20 ms、Dijkstra のアルゴリズムはノード数百台ほどでも rtt 以下であった。問題となる遅延はコネクションに要する時間であり平均 1 秒以上である。BlueZ が開発段階であることもあり、これを改善することで実行時間を短縮できる。

データ送信回数

フラッディング方式は、すべて重複なしに要求パケットがネットワーク全体へ流れたとしても $|V| - 1$ 回データ送信が行われる。それに加え応答時に最低でも経路のホップ数分のデータ送信が起こる。MaCC による経路要求時における全体のデータ送信回数は、結果として得る経路のホップ数を ω とすると、平均 $\sum_{i=0}^{\omega} \frac{i-1}{\omega+1} \times 2$ である。これは、

$$\omega > \frac{\sum_{i=0}^{\omega} i - 1}{\omega + 1} \times 2 \quad \omega^2 + \omega + 2 > 0 \quad (2)$$

となり、フラッディング方式での経路応答時におけるデータ送信回数 ω より小さい。したがって、ネットワーク全体に対してフラッディング方式より非常に少ないデータ送信回数となる。

データ送信回数は WPAN では問題となる。WPAN ではノードはパケット送信に電波を発生。電波は干渉の可能性があり、ネットワーク全体で経路検索時における送信回数が多ければその間それだけ電波干渉の可能性は高くなる。本稿では Bluetooth 上へ実装を行ったが、他の通信規格であれ無指向性の電波を用いる規格には該当する。表 2 は Bluetooth ピコネット間の干渉の影響を示している。実験環境は図 17 である。マシン B からマシン A へパケットを連続で送信し、マシン A が計測する。

表 2 Bluetooth 同士の干渉 (ビット誤り率)

干渉なし	Case 1	Case 2
0.0013	0.0157	0.1529
0.0004	0.0080	0.1293
0.0002	0.0127	0.0801
0.0010	0.0169	0.0615
0.0017	0.0169	0.1063
0.0006	0.0214	0.1497
0.0017	0.0233	0.1751
0.0009	0.0162	0.1829
0.0020	0.0219	0.3519
0.0010	0.0214	0.2553

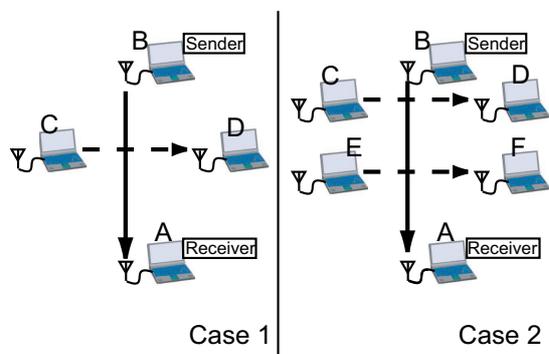


図 17 電波干渉実験環境

電波干渉がない場合は、ビット誤り率は 0.01 未満である。しかし、図 17 の Case 1 のように 1 対の外乱を加える。その結果ビット誤り率が高くなる事が確認できる。Case 2 のように 2 対に増やした場合、さらにビット誤り率が高くなる。ほぼ 0.1 程度、高い場合は 0.3 ほどになる。

フラッド方式により経路検索を行った場合、複数のノードが同時に送信を行う可能性がある。Bluetooth ピコネットではマスタによりピコネット内のデータの衝突は回避される。しかし、複数ピコネットが存在し、異なるピコネット同士ではチャネルアクセスはコントロールされない。そのため通信が同時に行われる可能性がある。また、フラッド方式はある時間内に多数のデータ送信が発生するため、その間電波状況が悪化し他の通信に影響を与える。

一方、MaCC は経路検索時のみであれば同時に送信を行うノードは存在しない。したがって電波干渉により発生した再送などにより遅延が増加する可能性を低くできる。また、経路検索時における送信回数が少なく、他の通信に与える影響を小さく抑えることができる。

4. 結論および今後の課題

本稿では次世代ネットワークにおいて重要な位置を占める WPAN に対するコンフィグレーション機構を提案した。WPAN は次世代環境において、インターネットとセンサネットワークを繋ぐ媒介となり、これら 3 つのネットワークを融合する。WPAN は即時的に形成されるため、動的なネットワーク機能が要求される。そこで本稿では WPAN コンフィグレーションシステム MaCC を提案した。

MaCC はフラッド方式によらず、最短経路を検索できる。また、WPAN に対してフラッド方式は通信品質悪化を引き起こす可能性があることを示し、MaCC は WPAN のような狭範囲ネットワークに有用であることを示した。

しかし、MaCC はその経路検索に Dijkstra のアルゴリズムを利用しているため、スケーラビリティが低い。同時にノード数が増加すると特に ROOT での隣接リストのサイズが肥大化する。ノード識別子として 6byte の Bluetooth アドレスを使用していることもあり、コンパクトにノード識別子を管理することが必要である。今回の実装ではデバイスの制限からマルチホップに対応していない。そこで今後の課題として、実装をマルチホップに対応させる。また、QoS ルーティングの問題がある。本稿では最短経路のみを対象しているが、最短経路が通信リンクの品質などから必ずしも最善であるとは限らない。よって、通信品質を考慮した改善が必要である。

5. 謝 辞

本研究の Bluetooth 上への実装に関しては文部科学省科学技術振興調整費「人間支援のための分散リアルタイムネットワーク基盤技術の研究」の支援による。

参 考 文 献

- 1) P. Bhagwat and A. Segall. A Routing Vector Method (RVM) for Routing in Bluetooth Scatternets. In *1999 IEEE International Workshop on Mobile Multimedia Communications (MoMuC'99)*, 1999.
- 2) B.J.Prabhu and A. Chockalingam. A Routing Protocol and Energy Efficiency Techniques in Bluetooth Scatternets. In *Proc. IEEE ICC'2002, New York*, April 2002.
- 3) Josh Broch, David B. Johnson, and David A. Maltz. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. IETF Internet-Draft [Work in Progress], February 2003.
- 4) UC Berkeley Mote hardware. <http://www.cs.berkeley.edu/~awoo/smartdust/>.
- 5) Mobile Ad-hoc Networks Working Group (MANET). <http://www.ietf.org/html.charters/manet-charter.html>. Charter [Work in Progress], 2002-09-16.
- 6) Charles E. Perkins, Elizabeth M. Royer, and Samir R. Das. Ad Hoc On Demand Distance Vector (AODV) Routing. IETF Internet-Draft [Work in Progress], June 2002.
- 7) Smart Dust Project. <http://robotics.eecs.berkeley.edu/~pister/smartdust/>.
- 8) BlueZ project Official Linux Bluetooth protocol stack. <http://bluez.sourceforge.net/>.
- 9) Godfrey Tan, Allen Miu John Guttag, and Hari Balakrishnan. Forming Scatternets from Bluetooth Personal Area Networks. Technical Report MIT-LCS-TR-826, MIT Laboratory for Computer Science, October 2001.
- 10) The Bluetooth Special Interest Group. <http://www.bluetooth.com>.
- 11) Ronald L. Rivest Thomas H. Cormen, Chales E. Leiserson and Clifford Stein. *Introduction to Algorithms, Second Edition*. MIT Press, 2001.
- 12) Wireless Intergrated Network Systems(WINS). <http://wins.rsc.rockwell.com/>.