

卒業論文

2002年度(平成14年度)

MaCC: WPAN コンフィギュレーション機構の設計と実装

指導教員

慶應義塾大学環境情報学部

徳田 英幸

村井 純

楠本 博之

中村 修

南 政樹

慶應義塾大学環境情報学部

滝澤 允

MaCC: WPAN コンフィグレーション機構の設計と実装

本論文では WPANs(Wireless Personal Area Networks) に適したルーティング機能を提案する。WPANs は、近年の無線通信技術に発展に伴い、新しいネットワークとして注目を集めている。WPANs は人を中心としたネットワークであり、身の周りの多くのデバイスからなるネットワークである。必要に応じた柔軟なネットワークを容易に構成できる。

ネットワークはアクセス透過性であることが要件となっている。インターネットではネットワークの内部構成を利用者は意識せずにあらゆるホストへ到達可能である。よってインターネットはアクセス透過性を実現していると言える。WPANs に対しても利用者がネットワークがどのようにデバイス間を繋いでいるかを意識せずに通信できるアクセス透過性を実現するシステムが必要である。

しかし、従来のインターネットで用いられるシステムをそのまま適用することはできない。従来のインターネットで用いられる IP(Internet Protocol) は厳格な管理下にあるアドレス体系が前提にあるシステムである。時と場所を限定しない WPANs に対してある管理組織が必要となるならば、その柔軟さ容易さが損なわれる。

また、WPANs はネットワーク 1 つに対しネットワーク内の通信を管理するマスタが必要なマスタ駆動型ネットワークである。しかしマスタはマスタになるべきデバイスが予め決まっているわけではない。周辺にあるデバイス同士がその中でマスタとなるデバイスを選出する。そこで従来のシステムを WPANs に適用した場合、マスタに負荷が集中する可能性があり、マスタ自身が行う通信帯域も狭められ、結果ネットワークのスループットが低下する可能性がある。

そこで本研究では、マスタ駆動型ネットワークである WPANs に適したシステムとして MaCC を提案する。MaCC は WPANs 利用者にアクセス透過性を実現するルーティング機能を提供する。MaCC によって、固定化されない WPANs 上の通信に対しオンデマンドで最短経路の確立が可能である。

本論文では MaCC のプロトタイプ実装、およびそれに対する評価測定結果を示す。そこで測定結果から MaCC の性能に加えて Bluetooth 依存の問題点を示し、Bluetooth 依存の問題点が大きな処理遅延に繋がるため、このことへの対応および消費電力やネットワーク全体からの評価など今後取り組むべき課題を述べる。

Abstract of Bachelor's Thesis

MaCC: The Design and Implementation of WPAN Configuration

In this thesis, we present a configuration mechanism for Wireless Personal Area Networks(WPANs), which includes formation and routing protocol.

WPANs address short range wireless networks, formed in a space about a person that typically extends to 10m in all directions easily and flexibly.

A routing protocol is necessary for WPANs as for Internet, providing access transparency for users. Internet Protocol(IP) provides access transparency, but IP assumes IP addresses are under control authoritatively. WPANs do not assume node identifiers are under control, because WPANs is not fixed. Thus, existing protocol can not be applied to WPANs. WPANs are formed dynamically. Hence, Formation Protocol is required to form WPANs effectively.

And WPANs have a characteristic of being Master-driven networks. So, Data traffic may concentrate to Master if WPANs use existing routing mechanism. That leads to throughput degradation of networks.

There, we present Master-driven Connection Control(MaCC), which provides the minimum hop-count routes for each communication dynamically when communications occur between nodes. MaCC has several features: autonomous distributed process, symmetric protocol, and simple mechanism.

We have implemented MaCC as prototype version and measured performance of it. We show that delay of MaCC and the delay to complete connection using Bluetooth. At last, we conclude this thesis and describe future work.

Makoto Takizawa

**Faculty of Environmental Information
Keio University**

目次

第1章	序論	1
1.1	本研究の背景	1
1.2	本研究の目的と意義	2
1.3	本論文の構成	3
第2章	WPANs(Wireless Personal Area Networks) 環境	5
2.1	IEEE802.15 Working Group for WPANs	5
2.2	通信規格	6
2.2.1	Bluetooth	7
第3章	関連研究	9
3.1	Bluetooth における関連研究	9
3.1.1	スキヤッタネット構成プロトコル	9
3.1.2	ルーティングプロトコル	10
3.2	グラフ理論:単一始点最短路アルゴリズム	11
3.2.1	グラフ理論	11
第4章	MaCC の設計	15
4.1	対象 WPAN 環境	15
4.2	MaCC の動作概要	16
4.3	MaCC の3機能	17
4.3.1	リレーションネット構成機能	17
4.3.2	最短経路検索機能	19
4.3.3	コネクション確立機能	22
第5章	Bluetooth 上での MaCC プロトタイプ実装	24
5.1	実装環境および MaCC プロトタイプ	24
5.2	リレーションネット構成機能部	24
5.2.1	MaCC メイン・ループ部	26
5.2.2	状態遷移部	28
5.3	最短経路検索機能部	29
5.3.1	隣接リストの実装	29
5.3.2	最短経路アルゴリズム部	30

5.3.3	経路要求および経路応答部	30
5.4	コネクション確立機能部	31
第6章	評価	34
6.1	測定環境	34
6.1.1	Bluetooth 通信の RTT 値	34
6.2	測定	35
6.2.1	測定方法	36
6.2.2	測定結果	36
6.3	フラッディング方式との比較	38
第7章	結論	43
7.1	本論文のまとめ	43
7.2	今後の課題	43

目 次

1.1	無線規格	1
2.1	Bluetooth ネットワーク構成例 ノード 1:ピコネット 1 およびピコネット 2 のスレーブ ノード 2:ピコネット 2 のスレーブかつピコネット 3 のマスタ	7
3.1	グラフとその 2 種類の表現方法 (a).5 頂点と 7 辺のグラフ (b).(a) の隣接 リスト表現 (c).(a) の隣接行列表現	11
3.2	幅優先探索 (BFS)	12
3.3	緩和操作	13
3.4	Dijkstra のアルゴリズム	13
3.5	Bellman-Ford のアルゴリズム	14
4.1	対象 WPAN 環境	15
4.2	リレーションネット構成例	17
4.3	状態遷移アルゴリズム	19
4.4	隣接情報 (隣接リスト) の階層型集中管理	20
4.5	経路要求の流れ図	21
4.6	経路応答の流れ図	22
4.7	ED 間経路のピコネット形成例	23
5.1	問い合わせ (Inquiry) 手続きのシーケンス	25
5.2	呼び出し (Page) 手続きのシーケンス	25
5.3	MaCC メイン・ループ	26
5.4	受信パケット処理部	27
5.5	AL_REQ パケット AL_RES パケットのフォーマット図	28
5.6	隣接リスト用構造体	29
5.7	R_REQ パケット R_RES パケットのフォーマット図	30
5.8	経路要求関数	31
5.9	経路応答関数	32
5.10	データパケットフォーマット	32
5.11	MaCC シーケンス	33
6.1	パケットタイプごとの RTT 値	35
6.2	RDTSC マクロ	36

6.3	MaCC 実行例および処理時間	37
6.4	Dijkstra 実行時間 (完全グラフ)	38
6.5	Dijkstra 実行時間 (木グラフ:次数 ≤ 2)	39
6.6	3 ホップ経路の要求例	41
6.7	MaCC およびフラッディング方式による経路検索所要時間	41

表 目 次

2.1	無線通信規格	6
2.2	ACL パケット	8
4.1	リレーション確立規則 (1 : 可 , 0 : 不可)	18
6.1	マシン環境	34
6.2	Bluetooth デバイス	35
6.3	コネクション確立遅延測定結果	38

第1章 序論

1.1 本研究の背景

無線通信技術の発展に伴い，無線通信端末機器が普及してきた．インターネットの一般利用者が，電話線などの伝送線を用いてインターネットへ接続する時代から無線アクセスポイントを設置し，無線によるインターネット接続を行うまでに至った．また現在までに各種用途に合わせた無線規格が数多く登場し，無線端末機器の普及を促している．図 1.1 にいくつかの無線規格とその用途を示す．

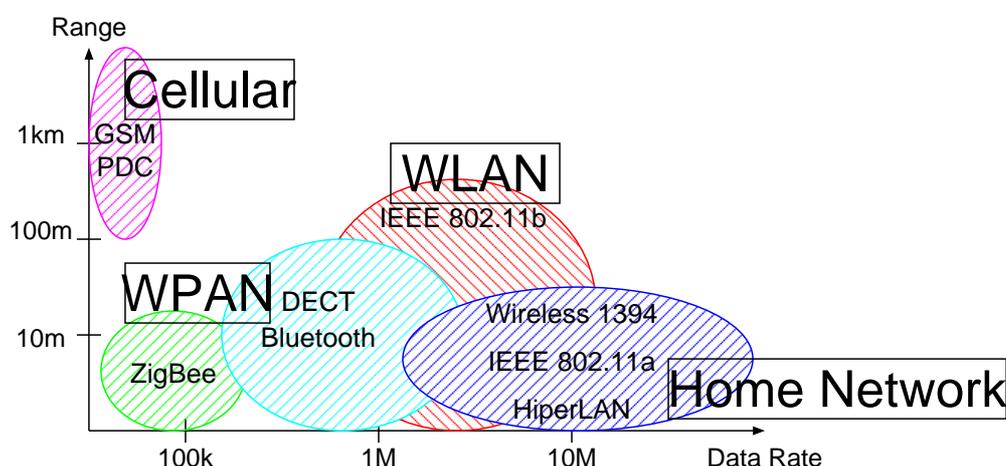


図 1.1: 無線規格

無線通信普及の背景にはその容易さがある．従来の有線接続ではケーブル設置の手間がかかり，またケーブルなどが必要であるために通信端末の場所にも制約が生じる．一方無線を用いれば，設定の煩わしさが解消され，物理的な位置の制限が格段に低くなる．たとえば LAN(Local Area Network) を構成する場合に，無線であれば柔軟にかつ容易に組むことができる．

LAN をはじめ一般にネットワークを無線により容易に構成できることが，多くの機器に通信機能を搭載させることに繋がった．従来の通信端末であるデスクトップ PC およびノート PC や携帯電話に加え，PDA などの小型端末，家庭電化製品や各種センサも無線通信機能を搭載している．今後，身の周りの多くの機器間で無線による通信が可能となる．

そこでネットワークのあり方も従来のインターネットに留まらない．インターネッ

トという大規模なネットワークが存在し、そこへアクセスポイントを経由することでネットワークに加わる従来の形式に加え、新しいネットワークが現れる。その1つが WPANs(Wireless Personal Area Networks) である。WPANs は人を中心としたネットワークである。身近にある家電や AV 機器に携帯電話、PDA などの機器同士で無線によりネットワークを構成し、それぞれが通信端末となる。小規模なネットワークが人の周りに構成されるのである。その特徴は、インターネットのように固定化したネットワークではなく、必要に応じて、かつ容易に柔軟にネットワークを構成できることである。

ネットワークにおける要件の1つは各通信端末へのアクセス透過性である。インターネット利用者はネットワークをどのように通り相手の通信端末へ到達しているかを意識する必要はない。すなわち、ネットワークの経路制御(ルーティング)機能が利用者へ利便性を提供している。WPANs もネットワークである以上、通信が行われ、ルーティング機能が必要となる。網状のネットワークにおいて通信端末間で経路が確立されてはじめて通信が可能となる。

しかし、WPANs に対して従来のインターネットで使用されているルーティング機能をそのまま適用することはできない。インターネットは IP(Internet Protocol) がルーティング機能を実現している。IP は各端末の識別子である IP アドレスをもとに経路判断を行う。IP アドレスはインターネット上のアドレス資源の標準化および割り当てを行う ICANN(Internet Corporation for Assigned Names and Numbers) により管理され、実際の割り当てはその下部組織にあたる各国の NIC(Network Information Center) が行っている。IP は端末識別子が厳格に管理されたもとで機能するシステムである。WPANs においても識別子を管理することが不可能とは限らないが、通信端末数が変動するなど、固定化されない WPANs に対して従来のシステムは適さない。

そこで本研究では、必要に応じて一時的に構成され、かつ固定化していない WPANs に対しルーティング機能を提供する MaCC(Master-driven Connection Control) を提案する。MaCC は任意の通信端末間に対し、オンデマンドでネットワーク内最短経路を提供することを目的としたシステムである。

1.2 本研究の目的と意義

WPANs にはその特性上、従来のルーティング機能は十分ではない。したがって WPANs に特化したルーティング機能を提供する必要がある。そこで以下に WPANs の特徴を述べ、従来のルーティング方式が十分ではないことを示す。

WPANs はネットワークを構成する際、ネットワーク管理の権限のある通信端末(ノード)を必要とする。他のノードはこのネットワークを管理するノード(マスタ)と交渉し、そのネットワークに加わる。本論文ではコネクション状態は、あるマスタの管理下に入るかまたは他ノードに対しマスタとなった状態のことを示す。すなわち、コネクション状態とはネットワークを構成し通信可能な状態を表す。WPANs では、1つのマスタに対して1つのネットワークがあり、これが最小単位のネットワークである。異

なるマスタは異なるネットワークに存在し、各々のマスタに属すノードも異なるネットワークに存在することになる。しかし、ノードは複数のマスタに属することも可能であり、そのノードを中継ノードとすることで複数ネットワークを接続できる。

マスタは主にネットワーク内ノードの送信権限を管理する。従来のネットワークでのアクセス制御である CSMA/CD(Carrier Sense Multiple Access with Collision Detection) や CSMA/CA(Carrier Sense Multiple Access with Collision Avoidance) は自律分散型の制御である。一方 WPANs では、マスタがそのマスタに属するノードの送信権利を管理し、同一ネットワーク内の複数のノードが同時に通信することを防ぐ。このようなネットワークはマスタ駆動型 (Master-driven) のネットワークである。

従来のルーティング方式はマスタ駆動型ネットワークに対し考案されたものではない。従来の方式としては、常に経路情報をルーティングテーブルとして保持し、それをもとに経路決定を行う方式と、アドホックネットワークで用いられるフラッディングを利用し経路を発見する方式がある。

ルーティングテーブルを用いる方式は、全体が複数ネットワークから構成される場合、マスタがルータの役割をすることが可能ではあるが、ノード識別子が IP アドレスのように階層化されていなければ扱う情報量が多くなる。また、マスタに負荷が集中する可能性がある。マスタがルータの代わりであれば、そのマスタのネットワーク外への通信および外からの通信がマスタを経由することになる。

フラッディングを用いる方式は、ルーティングテーブルによる方式が抱える問題を解決する。自律分散型機能であり、オンデマンドで経路の検索を行うため予め情報を保持する必要がないからである。しかし、フラッディングを行う場合、効率の点で問題が生じる。フラッディングはネットワーク全体に負荷を与える。また、WPANs はノードが密集する可能性が高く、そのため、複数のノードが同時にデータ送信を行った場合に電波状態が悪化し通信品質が低下する可能性がある。同一ネットワーク内ノードであればマスタが送信管理を行うが、異なるネットワークの場合は、その全体は管理されない。

そこで、本研究で提案する MaCC はノードに負荷が集中することを軽減し、かつネットワーク全体に負荷を与えない、自律分散型ルーティング機能を提供することを目的とする。MaCC はマスタ駆動型ネットワークに対し、マスタ駆動型という特性を利用したルーティングを行う。マスタは必然的にそのネットワークに属するノードの情報を保持する。よって、マスタの情報を利用すればフラッディングのようにネットワーク全体に負荷を与えることを防ぐことができる。また、MaCC は通信ノード間に最短経路を与えることで、転送にかかるオーバーヘッドの軽減を図る。

1.3 本論文の構成

本論文の構成は以下ようになる。第 2 章で WPANs の標準化および使用される通信方式について述べる。第 3 章は関連研究である。関連研究では WPANs で用いられる通信規格の 1 つである Bluetooth に関するネットワークトポロジおよびルーティング

プロトコルの先行研究をあげ，それに加えてグラフ理論についても触れる．グラフ理論はネットワークに応用可能であり，グラフにおける最短路アルゴリズムが既に存在する．MaCC が利用するマスタの情報は最短路アルゴリズムの入力値となり，MaCC はそれにより最短路を得る．第 4 章に MaCC の設計を述べ，第 5 章で MaCC のプロトタイプ実装について述べる．実装した MaCC プロトタイプに対する評価を第 6 章で述べ，最後第 7 章で本論文の結論とする．

第2章 WPANs(Wireless Personal Area Networks) 環境

本章ではWPANsについて述べる。WPANsを扱うIEEE802.15 WG(Working Group) [6]について述べ、次にIEEE802.15のTG(Task Group)をあげる。またWPANsで使われる通信規格を提示する。

2.1 IEEE802.15 Working Group for WPANs

802.15 WGはWPANsにおける標準化を行っている。WPANsはPCやPC周辺機器、携帯やPDAに加え家電などの機器間で近距離無線により形成したネットワークのことである。WPANsの特徴は、小規模ネットワーク、低消費電力、低コスト、POS¹でのデバイス間通信等がある。802.15 WGはWPANsにおける標準化に加え、他の無線規格との共存や相互接続性についても扱っている機関である。以下に各TG(Task Group)を提示する。

Task Group 1 - WPAN/Bluetooth™

IEEE802.15.1はIEEE-SA(Standards Association)に2002年3月21日に承認された。IEEE802.15.1はBluetooth SIG [12]からライセンスを取得し、Bluetooth仕様の一部を変更および利用している。802.15.1規格はBluetooth v1.1と完全互換である。

Task Group 2 - Coexistence

IEEE802.15.2はWPANs(802.15)とWireless Local Area Networks(WLANs) [5](802.11)の共存を図るTGである。同じ周波数帯を使用する規格同士は干渉によりスループットが低下するという問題が指摘されている。そこでTG 2は共存モデルや共存メカニズムを提示している。共存モデルはデータトラフィックモデル、MAC層モデル、物理層モデル、RFチャンネルモデルに分かれている。共存メカニズムには送信電力コントロールやBluetoothと802.11を合わせた時分割多重方式の提案等がある。

¹Personal Operating Space. 人の静止と移動に関わらず、人を中心として半径10メートル程度の範囲。

Task Group 3 - WPAN High Rate

IEEE802.15.3は20Mbps以上の高転送速度のWPANsの標準化を進めている。802.15.3は周波数帯としてISMバンドである2.4GHz帯を用い、5つの転送レートをサポートする。802.15.3MAC仕様はPNC(Piconet Controller)とDEV(Device)からなるトポロジ、ピア・ツー・ピア通信、QoSおよびセキュリティと認証を定めている。PNC-DEVネットワークは1つのPNCに対し複数のDEVから成るスター型トポロジとなる。PNCがネットワーク内ノード間の通信を制御する。

Task Group 4 - WPAN Low Rate

IEEE802.15.4は転送レートが240kbps、20kbps程度の低転送レートのWPANsにおけるMACおよびPHYの標準化を行っている。802.15.4も同様に2.4GHz帯を使用している。802.15.4MAC仕様はデバイスを2種類に分けている。それはFFD(Full function device)とRFD(Reduced function device)である。FFDはネットワークコーディネータとしてネットワークを管理でき、どのデバイスとも直接通信可能である。一方、RFDはネットワークコーディネータの管理下のもと、ネットワークコーディネータとのみ通信可能である。よって、FFDとRFDはスター型ネットワークを構成する。

2.2 通信規格

本節は現在でのWPANs用の通信規格について述べる。表2.1に各通信方式を3つあげる。Bluetooth、UWB、ZigBeeはそれぞれIEEEにおける802.15.1、802.15.3a、802.15.4に対応する。802.15.3aは802.15.3のより高速な物理層規格を扱っている。本稿ではBluetooth上へMaCCの実装を行ったので、Bluetoothを取りあげて小節2.2.1にて概説する。

表 2.1: 無線通信規格

規格名	Bluetooth	UWB	ZigBee
最大伝送レート	1Mbps(実効値 721kbps)	100Mbps	10kbps - 250kbps
周波数帯域	2.4GHz 帯	3.1GHz - 10.6GHz*帯	2.4GHz 帯, 915MHz 帯
変調方式	FD-SS(最大 1600hop/秒)	1 次変調のみ	DS-SS
最大伝送距離	10m(1mW,0dBm), 100m(100mW,+20dBm)	10m	75m
最大消費電力	100mW	10nW	受信時 21mW, 送信時 30mW

*FCC(Federal Communication Commission:(米)連邦通信委員会)による。[4]

2.2.1 Bluetooth

本小節では Bluetooth 技術を概説する。Bluetooth は WPANs での使用が想定されている。その技術要件は、モバイル端末搭載に最適である、近距離の無線接続を廉価に実現できる、1 対多のアドホックネットワークを構築できる、音声およびデータを双方向に伝送できる、送信パケットを選択可能である、誤接続防止および盗聴防止のセキュリティ機能を有す、という項目である。Bluetooth はこれらの技術要件を満たしている。

Bluetooth では、通信端末は一意的な Bluetooth アドレスによって識別される。Bluetooth アドレスは 48 ビットの IEEE802 仕様に準拠している。

以下に Bluetooth が構築するネットワークについて述べる。Bluetooth は通信を行うにあたり、各ノードにマスタまたはスレーブの役割を与える。1 つのマスタに対し複数のスレーブ (≤ 7 台) が接続し、ピコネットというネットワークを構築する。通信帯域はピコネット内で共有される。ピコネット内通信はすべてマスタによって管理され、スレーブが任意に送信することはできない。また、Bluetooth ではマスタとスレーブ間通信のみ可能であり、スレーブ同士は直接通信できない。

ピコネットは 1 つのマスタと 7 つのスレーブ計 8 つが最大となる。しかし、複数のピコネットがノードを共有することで複数ピコネットを接続してより大きなネットワークとしてスカッターネットを構築できる。図 2.1 に Bluetooth ネットワークの構築例を示す。

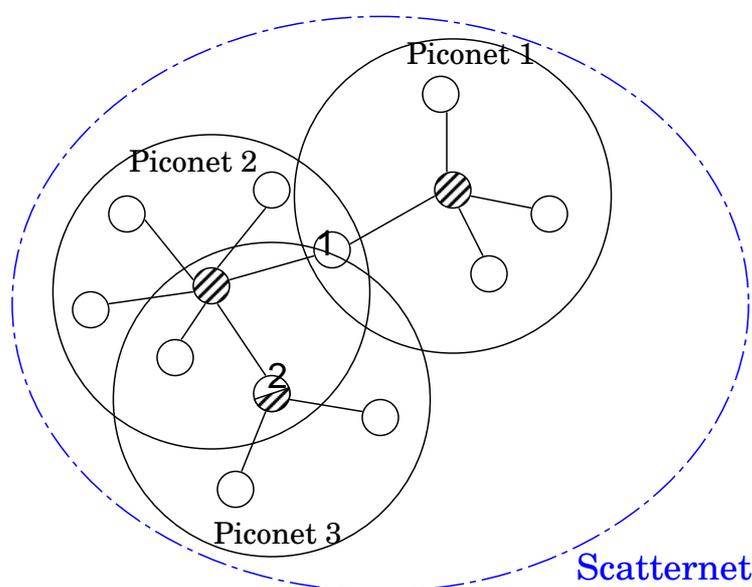


図 2.1: Bluetooth ネットワーク構成例 ノード 1:ピコネット 1 およびピコネット 2 のスレーブ ノード 2:ピコネット 2 のスレーブかつピコネット 3 のマスタ

Bluetooth の通信方式は周波数ホッピングスペクトル拡散方式である。79 チャンネル (1MHz/チャンネル) を毎秒 1600 回でホッピングさせ、79MHz 帯域に拡散変調する。そ

ここで通信を行うためには通信ノード間で送受信に同一周波数帯を使用するようホッピングパターンを一致させる必要がある。Bluetoothにおけるコネクション状態とは通信ノード間で周波数ホッピングパターンを一致させた状態である。この状態で通信が可能となる。ピコネットでは、スレーブがマスタの周波数ホッピングパターンに合わせることで、周波数ホッピングパターンが共有される。

1チャンネルあたり625マイクロ秒の時間帯をスロットと呼び、スロットごとにスロット番号がある。マスタは偶数番号スロットで送信し、逆にスレーブは奇数番号スロットで送信する。

次にBluetoothの通信リンクについて説明する。Bluetoothは技術要件として音声およびデータの双方向通信を満たしていると先述した。そこでBluetoothは音声などのリアルタイム性が要求される通信に対しSCO(Synchronous Connection-Oriented)リンクを、それ以外のデータ通信に対しACL(Asynchronous Connection-Less)リンクを提供している。

SCOリンクはリアルタイム性が必要な通信に対してマスタとスレーブで1対1のポイント・ツー・ポイントの接続リンクを提供する。一定間隔でデータを送信し、通信速度は64kbpsで一定である。

一方、ACLリンクは非同期の通信に使用される。ACLリンクはさらに3つに分けることができる。それは非同期通信方式(Asynchronous Transfer)、アイソクロナス(Isochronous Transfer)、同報通信方式(Broadcast Transfer)である。非同期通信方式は通常の packets 送受信である。通信速度は約100kbps~700kbpsである。アイソクロナス通信方式ではある一定間隔内に必ず packets を送信する。同報通信方式はマスタからそのピコネットに存在するすべてのスレーブに packets を送信する。表 2.2 に ACL packets の packets タイプおよび各々の特徴を示す。

表 2.2: ACL packets

packets タイプ	ペイロードヘッダ	ユーザデータ	FEC	CRC	最大伝送速度 (kbps)
DM1	1	0-17	rate 2/3	有	108.8
DH1	1	0-27	無	有	172.8
DM3	2	0-121	rate 2/3	有	258.1
DH3	2	0-183	無	有	390.4
DM5	2	0-224	rate 2/3	有	286.7
DH5	2	0-339	無	有	433.9

第3章 関連研究

本章は本研究の関連研究をあげる．現在 WPANs における関連研究は Bluetooth 通信を利用した研究が中心である．そこで本章で Bluetooth に関する先行研究として，Bluetooth ネットワークであるスキヤッタネットの構成プロトコルおよびルーティングプロトコルの先行研究をあげる．また MaCC が最短経路を求めるために用いた最短路アルゴリズムについて述べる．本研究で利用したグラフ理論，およびグラフ理論における主要な最短路アルゴリズムについて概説する．

3.1 Bluetooth における関連研究

3.1.1 スキヤッタネット構成プロトコル

本小節では Bluetooth ネットワークにおけるスキヤッタネット構成プロトコルの関連研究をあげる．本小節であげる先行研究はネットワークトポロジにのみ重点があり，ルーティング機能には触れていない．

BTCP

Bluetooth Topology Construction Protocol(BTCP) [10] は Bluetooth スキヤッタネット構成プロトコルである．BTCP は 3 段階で構成される．第 1 段階はネットワークを構成するすべてのノードからネットワークトポロジを決定する 1 つのノード(コーディネータ)を選出する．第 2 段階では，コーディネータがネットワーク内すべてのノードにマスタかスレーブか役割を決定する．最後，第 3 段階でコーディネータは各ノードに割り当てた役割でのコネクション確立を命令し，ピコネットまたはスキヤッタネットを構成する．BTCP はネットワーク全体におけるピコネット数を最小にする．すなわち，マスタとして動作するノードができるだけ多くのスレーブを保持することでネットワーク構成が非常に簡潔になる，という特徴を有す．

しかし，BTCP はコーディネータ選出に時間がかかる．またネットワーク全体の情報をもってコーディネータはネットワークトポロジを決定するため，再構成を行う場合にもオーバーヘッドが大きくなる．また，ネットワークトポロジは静的であり，新たにノードが加わった場合にネットワーク内に取り込むことができない．

BTCP の特徴の 1 つにピコネット数を最小に抑えることがあげられる．しかし，ピコネット数が最小であることが必ずしも通信のスループットを向上させることには繋がらない．Bluetooth での通信はすべてマスタとスレーブ間で行われるため 1 つのマ

スタに対するスレーブ数が多いとマスタに負荷が集中する可能性がある．また BTCP はルーティングについては触れておらず，ネットワークトポロジにのみ着眼している．

TSF

Tree Scatternet Formation(TSF) [11] は BTCP と同様にスキヤッタネット構成プロトコルである．TSF の特徴は木構造スキヤッタネットを構成することである．TSF は厳密な接続規則を設定し，各ノードが自律的にどのノードとコネクションを確立するかを判断する．よって，BTCP はコーディネータ選出のために互いにメッセージを交換する必要があるが，TSF は木構造スキヤッタネット構成に対し他ノードとの情報交換を必要としない．また，TSF はスキヤッタネット構成終了段階を設けていない．したがってノードの移動やノード数の変動に対し柔軟に対応可能である．

しかし，TSF もルーティングに関しては触れていない．ネットワークはトポロジのみでなく，ネットワーク内の通信をサポートする機能が不可欠である．

3.1.2 ルーティングプロトコル

本小節では Bluetooth ネットワークにおけるルーティングプロトコルの関連研究をあげる．本小節であげるルーティングプロトコルはフラッディング方式を用いている．また，あらかじめスキヤッタネットが構成されていることを前提としている．すなわち，ネットワークのノード間はコネクション状態にある．

RVM

Routing Vector Method(RVM) [1] はフラッディングを用いるルーティングプロトコルとして設計されている．RVM は OSI モデル第 3 層で動作する．第 3 層プロトコルのヘッダに経路情報を含めてソースルーティングを行う．経路情報は，転送先ピコネット識別子とその識別子のピコネット内で転送すべきノードのアドレスとの組みとして付加される．RVM はピコネット識別子としてマスタの 48 ビットの Bluetooth アドレスを用いない．複数ピコネットに属するノードが各々のピコネットにローカルに 7 ビットの識別子を割り当てることでデータ量を減らしオーバーヘッドを小さくしている．

Bluetooth Routing Protocol

ここであげるルーティングプロトコル [2] も経路発見にフラッディングを行う．このプロトコルはスキヤッタネットが構成され，ピコネット識別子がマスタに与えられていることを前提としている．経路情報としてマスタはピコネット識別子を加え，スレーブは Bluetooth アドレスを加える．結果となる経路情報はピコネット識別子と Bluetooth アドレスが交互に表れる．ここであげたルーティングプロトコルはフラッディングに

よって複数の経路が見つかった場合，エネルギー効率を指標として選ぶことを特徴としている．すなわち，可能な限りバッテリー残量が多いノードを経由しネットワーク全体としてエネルギー効率を高めることを目的としている．

3.2 グラフ理論:単一始点最短路アルゴリズム

本節ではネットワークにおける最短経路を求めるアルゴリズムをあげる．はじめにグラフ理論について概説し，次に単一始点最短路問題として扱われるアルゴリズムについて述べる．本節では MaCC が用いる Dijkstra のアルゴリズムに加え，幅優先探索アルゴリズム，Bellman-Ford のアルゴリズムをあげる．

3.2.1 グラフ理論

本小節ではグラフ理論を概説する．グラフとは頂点と呼ばれる集合 V ，および辺と呼ばれる頂点の非順序対の集合 E で構成される．グラフ G は一般に， $G(V, E)$ で表される．グラフにおいて，辺 u, v が存在する場合，頂点 u, v は隣接する (adjacent) という．図 3.1 にグラフの 1 例とそのグラフの隣接リストによる表現および隣接行列による表現を示す．

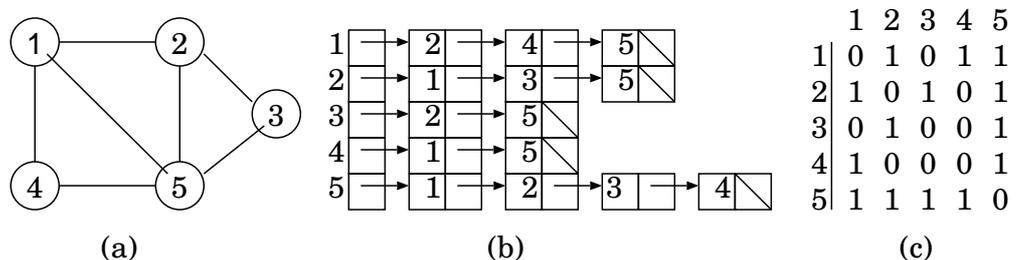


図 3.1: グラフとその 2 種類の表現方法 (a).5 頂点と 7 辺のグラフ (b).(a) の隣接リスト表現 (c).(a) の隣接行列表現

次にあげるアルゴリズムはグラフ探索アルゴリズムである．ある頂点 s から他の頂点への経路を求めるアルゴリズムである．経路は辺の列 $(e_s, e_x, e_y, \dots, e_d)$ かまたは頂点の列 $(v_s, v_x, v_y, \dots, v_d)$ ，頂点と辺の交互列 $(v_s, e_x, v_x, e_y, v_y, \dots, e_d, v_d)$ として表される．

幅優先探索 (Breadth-first search)

幅優先探索は始点 s から到達可能なすべての頂点を発見する．また，始点 s を根とする幅優先探索木を求める．幅優先探索木は始点 s から各頂点への最短路を示す．

幅優先探索アルゴリズムでは各頂点が 3 段階の過程を経る．一般的に白，灰色，黒とされ，はじめはすべての頂点は白である．頂点は順に灰色，黒へと変化する．白以

外へ変化するのは発見された時である．そして黒への変化はその頂点に隣接する頂点がすべて発見された，すなわち白以外へ変化した時に起こる．すべての頂点が黒に変化した時点でアルゴリズムの終了である．図 3.2 に幅優先探索アルゴリズムを示す．

```

BFS( $G, s$ )
1: for  $u \in V[G] - \{s\}$ 
2:   color[ $u$ ]  $\leftarrow$  WHITE
3:    $d[u] \leftarrow \infty$ 
4:    $\pi[u] \leftarrow \text{NIL}$ 
5: color[ $s$ ]  $\leftarrow$  GRAY
6:  $d[s] \leftarrow 0$ 
7:  $\pi[s] \leftarrow \text{NIL}$ 
8:  $Q = \{s\}$ 
9: while  $Q \neq \emptyset$ 
10:   $u \leftarrow \text{head}[Q]$ 
11:   for each  $v \in \text{Adj}[u]$ 
12:     if color[ $v$ ] = WHITE
13:       color[ $v$ ]  $\leftarrow$  GRAY
14:        $d[v] \leftarrow d[u] + 1$ 
15:        $\pi[v] \leftarrow u$ 
16:       ENQUEUE( $Q, v$ )
17:   DEQUEUE( $Q$ )
18:   color[ $u$ ]  $\leftarrow$  BLACK

```

図 3.2: 幅優先探索 (BFS)

$d[u]$ は始点 s から頂点 u までの距離 (辺の数)， $\pi[u]$ は u の先行頂点 (単一起点木における始点側の隣接頂点) を示す．アルゴリズム終了後に得た π 配列を順番に辿ることで頂点の列としての経路を得る． Q は灰色の頂点を扱う FIFO キューである．

BFS の動作を以下に述べる．1-4 行目が初期化部分である．すべての頂点を白にし，始点 s からすべての頂点への距離を ∞ ，すべての頂点の先行頂点を NIL に初期化する．次に s を灰色へ， s への距離を 0 にする．8 行目で Q へ s を挿入する．ループ 1 回目は Q は s のみであるので， s の隣接頂点を灰色にし， Q へ挿入する．隣接頂点をすべて調べ終わったら Q から出して，黒へ変化させる．これをすべての頂点が黒に変化するまで繰り返す． Q は灰色の頂点のみ扱うので Q が空になった時点ですべての頂点が黒へ変化したことを示す．

ここで，幅優先探索アルゴリズムの実行時間を考える．すべての頂点の隣接頂点の走査は，重複した頂点の走査は存在しないため辺の数に等しく，これの実行時間は $O(E)$ である．それに初期化に要する時間が $O(V)$ であり，BFS の実行時間は $O(V + E)$ となる．

Dijkstra のアルゴリズム

幅優先探索は単一始点最短路を求めることが可能である。しかし、幅優先探索はすべての辺が単位重みであるグラフを対象とするアルゴリズムである。一方で、Dijkstra のアルゴリズムは各辺に重みがあるグラフに対し、経路上の辺の重みの合計が最小となる最短路を求めるアルゴリズムである。すなわち、Dijkstra のアルゴリズムは経路の重みが最小となる単一最短路木を求める。

Dijkstra のアルゴリズムは緩和法を用いている。緩和法とは始点 s から v への経路の重みの上界を $d[v]$ として管理し、この上界をさらに低くできる場合は $d[v]$ を更新する。図 3.3 に緩和操作を示す。

```
RELAX( $u, v, w$ )
1: if  $d[v] > d[u] + w(u, v)$ 
2:    $d[v] \leftarrow d[u] + w(u, v)$ 
3:    $\pi[v] \leftarrow u$ 
```

図 3.3: 緩和操作

$w(k, l)$ は辺 (k, l) の重みを返す関数である。この操作は、もとの上界 $d[v]$ と u を v の先行頂点とした場合の重みを比べ、もとの上界より低ければ更新を行う。

次に図 3.4 に Dijkstra のアルゴリズムを示す。

```
DIJKSTRA( $G, w, s$ )
1: INITIALIZE-SINGLE-SOURCE( $G, s$ )
2:  $S \leftarrow \emptyset$ 
3:  $Q \leftarrow V[G]$ 
4: while  $Q \neq \emptyset$ 
5:    $u \leftarrow \text{EXTRACT-MIN}(Q)$ 
6:    $S \leftarrow S \cup \{u\}$ 
7:   for each  $v \in \text{Adj}[u]$ 
8:     RELAX( $u, v, w$ )
```

図 3.4: Dijkstra のアルゴリズム

INITIALIZE-SINGLE-SOURCE は始点 s からすべての頂点への経路の重みの上界を ∞ とし、すべての頂点の先行頂点を NIL に初期化する。その後始点 s の重み $d[s]$ に 0 を設定する。 Q は d の値をキーとするプライオリティキューである。

以下に Dijkstra のアルゴリズムの動作を説明する。初期化操作後、プライオリティキュー Q へ値を代入する。5 行目は Q から重みが最小の頂点を取り除き、 u へ代入する。7,8 行目で u に隣接する頂点すべてに対し緩和操作を行う。この操作を Q からす

すべての頂点を取り除くまで繰り返す．最短路は幅優先探索と同様に π 配列から取得できる．

Dijkstra の実行時間を考える．4 行目の while ループは $|V|$ 回であり，1 回の EXTRACT-MIN の実行時間は $O(V)$ であるから，全体で EXTRACT-MIN には $O(V^2)$ の実行時間を要する．次に，while ループ中全体での 7 行目隣接リスト走査で各辺は 1 度のみ調べられるため，全体で辺の数 $|E|$ 回となる．よって Dijkstra のアルゴリズム全体の実行時間は $O(V^2 + E)$ となる．

Bellman-Ford のアルゴリズム

Bellman-Ford のアルゴリズムも Dijkstra のアルゴリズム同様に経路の重みが最小となる単一始点最短路を解く．また，Dijkstra のアルゴリズムと同様に緩和法を用いている．Dijkstra のアルゴリズムは辺の重みは非負であるグラフを対象としているが，Bellman-Ford のアルゴリズムは辺の重みが負であってもよく，より一般化された単一始点最短路アルゴリズムである．

以下の図 3.5 に Bellman-Ford のアルゴリズムを示す．Bellman-Ford のアルゴリズムは解が存在するか否かを判定する．それは負の重みの閉路が存在するかを調べ，存在する場合は解が存在しないことを告げる．

```
BELLMAN-FORD( $G, w, s$ )
1: INITIALIZE-SINGLE-SOURCE( $G, s$ )
2: for  $i = 1$  to  $|V[G]| - 1$ 
3:   for each  $(u, v) \in E[G]$ 
4:     RELAX( $u, v, w$ )
5: for each  $(u, v) \in E[G]$ 
6:   if  $d[v] > d[u] + w(u, v)$ 
7:     return FALSE
8: return TRUE
```

図 3.5: Bellman-Ford のアルゴリズム

1 行目の初期化は Dijkstra のアルゴリズムにおける初期化と同様の操作である．2-4 行目のループですべての辺に対する緩和走査を $|V| - 1$ 回行っている．次の 5-7 行目は負の重みの閉路の有無を調べ，存在する場合は FALSE を返す．解が存在する場合は TRUE を返し，アルゴリズムは終了する．

Bellman-Ford のアルゴリズムの実行時間は $O(VE)$ である．初期化に $O(V)$ ，各辺への緩和操作 $O(E)$ 時間が $|V| - 1$ 回である．それに加え 5-7 行目のループが $O(E)$ 時間を要する．

第4章 MaCCの設計

本章では，WPANにおけるマスタ駆動型ネットワークに適したネットワークコンフィグレーション機構 MaCCの設計について述べる．MaCCはマスタ駆動型ネットワークという特徴を利用し，WPAN内での通信に対し最短経路を提供する．MaCCでは各ノードが等価な機能を有し自律分散的に動作する．

はじめに MaCC が対象とする WPAN 環境を述べ，MaCC の動作概要を述べる．次に MaCC の各機能を詳説する．

4.1 対象 WPAN 環境

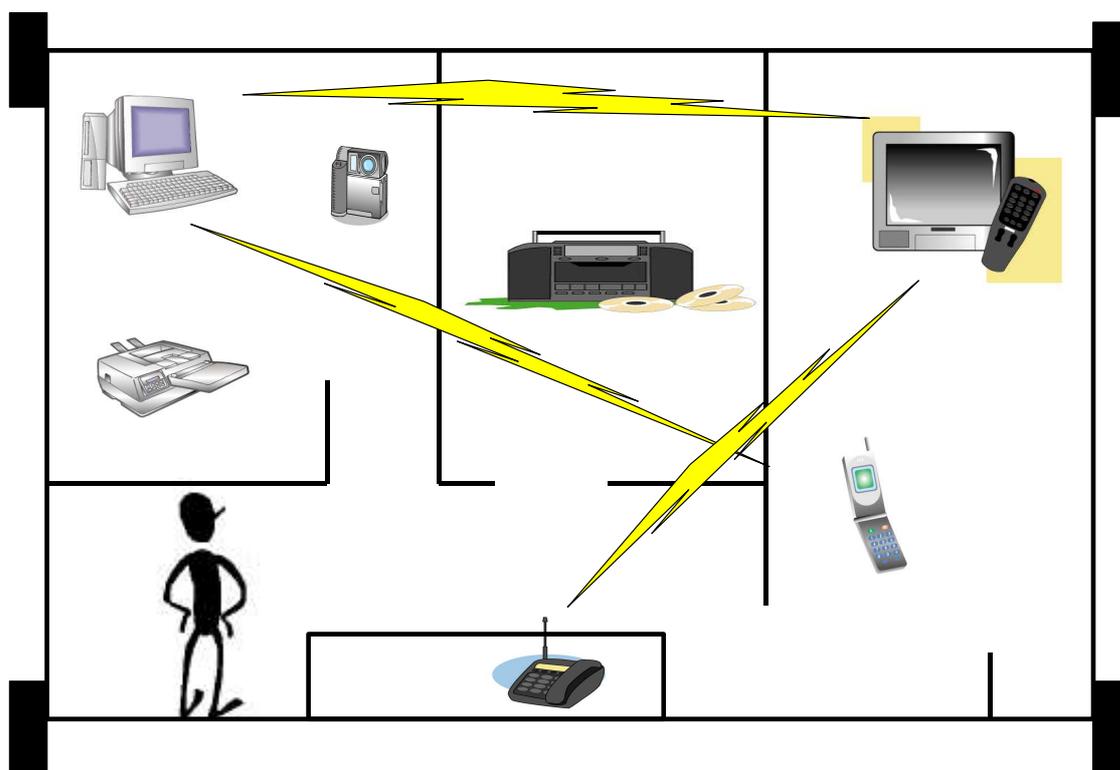


図 4.1: 対象 WPAN 環境

MaCC は家庭内で構築される WPAN を対象としている．デスクトップ PC，プリン

タ、ノートPC、携帯、家庭用電話器など従来の通信機器に加え、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MaCCの前提条件を、現在のインターネットのような莫大なノード数から成る大規模ネットワークではなく、数十台前後の小規模ネットワークとする。

また、家庭内ネットワークを想定することで、頻繁にコネクションが切り替わる移動は考えない。比較的狭い小規模ネットワークであり、各ノードが他ノードの通信範囲を頻繁に出入りするような移動は起こらないとする。しかし、通信ノード数の変動は想定する。従来のような通信機器の種類の制限がなくなり今後も通信機能を有した機器の増加が予想される。よって家庭内ネットワークにおいても通信ノード数は従来のインターネットに比べ頻繁に変動すると想定される。また各機器の電源オン/オフによるネットワーク内ノード数の変動も十分考えられる。

以上の前提条件を満たすWPANにMaCCを適用する。

4.2 MaCCの動作概要

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算出する。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を行い、通信を開始する。

MaCCは3機能を有し、以下にこの3機能を概説する。

1. リレーシオンネット構成機能: リレーシオンネットというネットワークを木構造で構成する。これによりネットワークを階層型にする。リレーシオンネットはMaCCで独自に定義した用語であり、4.3.1小節で詳述する。
2. 最短経路検索機能: リレーシオンネット内において、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: 経路検索機能から得た経路に沿ったコネクションを確立する。

リレーシオンネット構成機能により隣接情報を階層型集中管理を行うための階層型木構造ネットワークを構成する。その上で最短経路検索を行う。自ノードで経路が割出せなかった場合、上位層ノードへ経路検索を要求する。この経路要求は再帰的に行われる。これら手続きから得た経路をもとに、コネクション確立機能が通信経路を確立する。

4.3 MaCCの3機能

本節ではMaCCが有する3機能を詳説する。

4.3.1 リレーションネット構成機能

本機能はリレーションネット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、マスタは一方をスレーブと認識し、スレーブは一方をマスタであると認識し、互いに他方のアドレスを保持している状態”と定義する。リレーション状態はマスタ/スレーブ関係を構築するが、コネクション状態以前の状態であるため通信はできない。ここで混同を避けるため、リレーション状態におけるマスタおよびスレーブをRL_MASTER, RL_SLAVEとする。リレーションネットとはネットワークとして隣り合うノードがリレーション確立状態にあるネットワークである。

リレーションネットの例を図4.2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4.2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をPARENTとする。リレーションネット内ではROOT以外のノードすべてがただ1つのPARENTを有す。図4.2ではFはBを、GはDをPARENTとし、AはB,C,DのPARENTである。

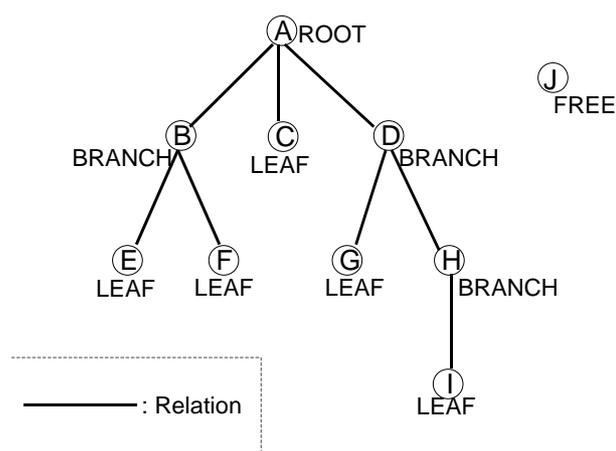


図 4.2: リレーションネット構成例

次にMaCCがリレーションネットを木構造に構成するアルゴリズムを述べる。MaCCは関連研究で述べたTSFの接続規則を用いる。表4.1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 4.1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。

証明

木 T 内で閉路が存在しないことを示す。表 4.1 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。 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, 図 4.1 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,

$$T_3 \text{の辺の数} = (n_1 - 1) + (n_2 - 1) + 1 = (n_1 + n_2) - 1$$

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

表 4.1: 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 4.3.1 はノードの 4 つの状態の状態遷移アルゴリズムである。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 RL_MASTER か RL_SLAVE かによって決まる。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。

以上がリレーションネット構成機能の動作内容である。本機能により WPAN で階層型木構造ネットワークとなるリレーションネットが構成される。

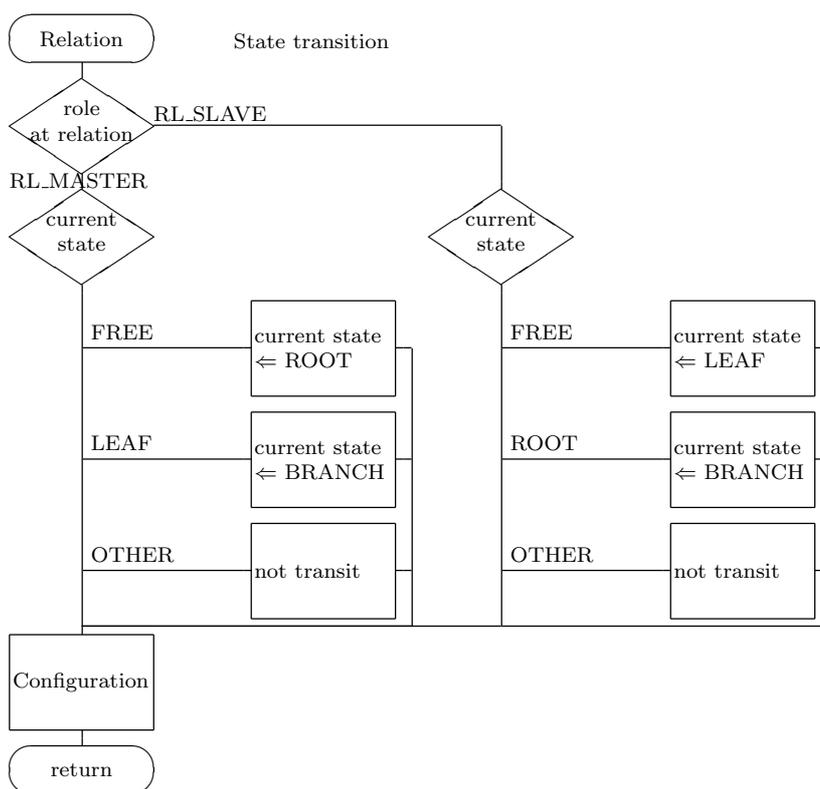


図 4.3: 状態遷移アルゴリズム

4.3.2 最短経路検索機能

本小節では最短経路検索機能について述べる。MaCC 最短経路検索機能は、構成したリレーシオンネット内において、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。以降で本機能を詳説する。

経路検索では各ノードが保持する隣接情報を使用する。リレーシオンネット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 4.4 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 4.4 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 4.4 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。リレーシオンネット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つは隣接リストから最短

経路を算出するアルゴリズム部であり，もう1つは経路要求(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．

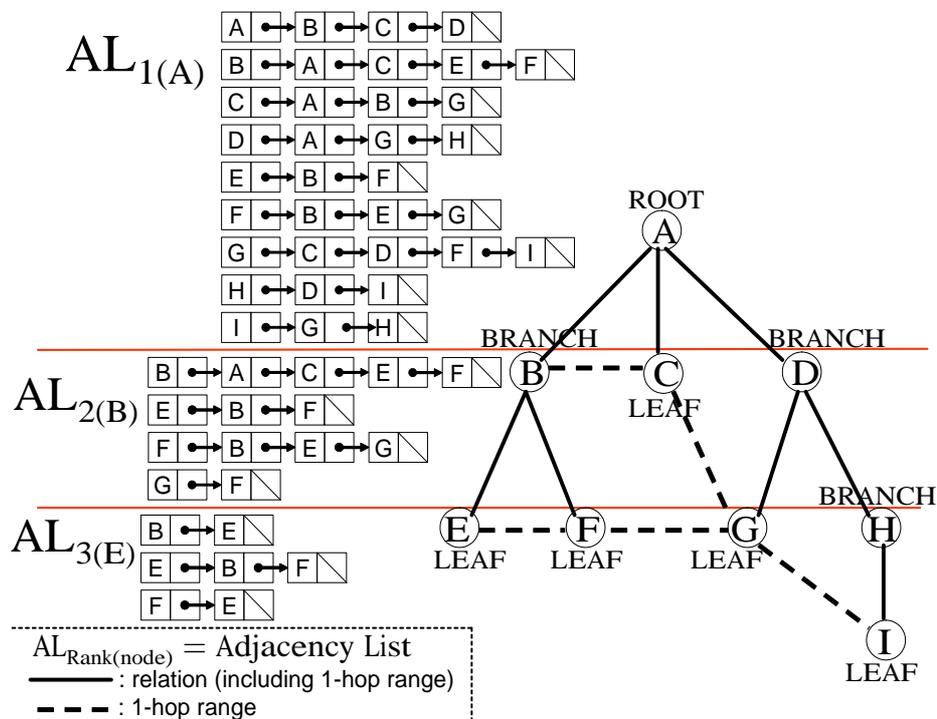


図 4.4: 隣接情報 (隣接リスト) の階層型集中管理

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．Dijkstra 自体は関連研究で述べたので本節ではその内容には触れない．

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に対する 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ノード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と行われる．その際，層を越えることはない．すなわち，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 4.4 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．そこで経路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 4.3.2，図 4.6 はそれぞれ経路要求，経路応答の流れ図である．

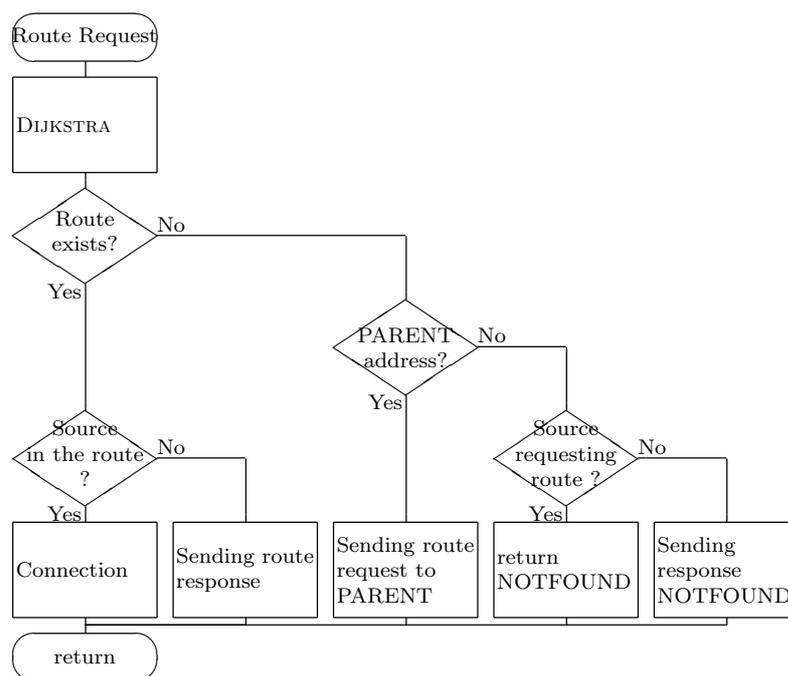


図 4.5: 経路要求の流れ図

経路検索の関連研究として，経路検索に MANET(Mobile Ad-hoc Networks) [7] で用いられるルーティングプロトコル (DSR [3]，AODV [8]) と同様にフラッディングにより経路検索を行う RVM(Routing Vector Method) がある．しかし，フラッディング

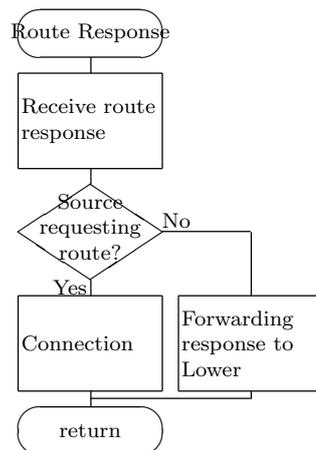


図 4.6: 経路応答の流れ図

による経路検索はネットワーク全体へ負荷を与えるという問題点がある．MaCC は経路検索要求を階層構造に従って行い，ネットワーク全体へメッセージ送信はせずネットワークへの負荷を抑えることができる．

4.3.3 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ッケージが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ョン確立を行うということを各ノードで繰り返す．すなわち，コネクション確立用のパッケージを用意せず，通信パッケージがコネクション確立をも兼ねる．

WPAN においてコネクションを確立するという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 4.7 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．

ここでリレーションネットおよびコネクション時のマスタ/スレーブ関係について述べる．リレーションネットとコネクション時におけるマスタ/スレーブ関係は一致しない．リレーションネットから見た経路が必ずしも最短経路とは限らないからである．

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．

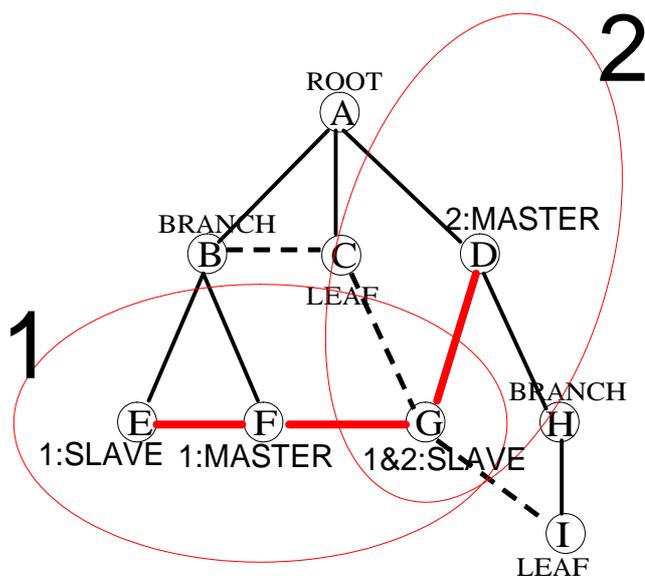


図 4.7: ED 間経路のピコネット形成例

第5章 Bluetooth上でのMaCCプロトタイプ実装

本章では Bluetooth 上での MaCC のプロトタイプ実装について述べる。Bluetooth は現状での Bluetooth 規格の確立や Bluetooth 搭載製品の普及から、主要な WPAN の通信規格となることが予想される。よって本研究では WPAN 通信規格として Bluetooth を用いて実装を行った。本章の構成は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、以降順に、リレーションネット構成機能部、最短経路検索機能部、コネクション確立機能部の実装について述べる。

5.1 実装環境および MaCC プロトタイプ

本節では MaCC の実装環境および MaCC プロトタイプの概要を述べる。MaCC プロトタイプ実装は Redhat7.3(kernel-2.4.x) 上で行った。また、Linux 用 Bluetooth プロトコルスタックとして BlueZ [9] 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。

本研究では MaCC をアプリケーション層でのデーモンプロセスとして実装を行った。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。

5.2 リレーションネット構成機能部

本節ではリレーションネット構成機能部の実装について述べる。しかし実装について具体的な説明を行う前に Bluetooth におけるリレーション確立状態を定義する。本研究ではリレーション確立は Bluetooth 依存の実装となる。

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 5.1 と図 5.2 にそれぞれ問い合わせと呼び出しのシーケンスを示す．

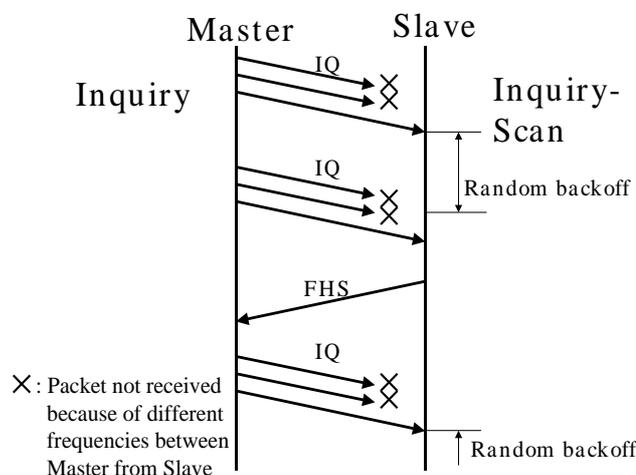


図 5.1: 問い合わせ (Inquiry) 手続きのシーケンス

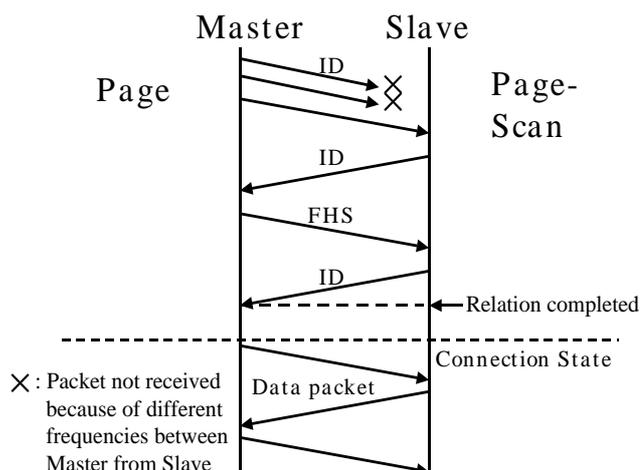


図 5.2: 呼び出し (Page) 手続きのシーケンス

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 において小節 4.3.1 での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。

以降で具体的なリレーション ネット構成機能の実装について述べる。次小節 5.2.1 で MaCC メイン・ループ部の実装について述べ、次々小節 5.2.2 にリレーション ネット構成におけるリレーション 接続規則および状態遷移部実装を述べる。

5.2.1 MaCC メイン・ループ部

MaCC はメイン・ループにおいて問い合わせ送信と問い合わせ スキャンを一定間隔で交互に行う。送信およびスキャンを交互に行うことですべてのノードが RL_MASTER および RL_SLAVE 両方になる可能性を有す。図 5.3 に問い合わせ 交互実行を行う MaCC メイン・ループ部疑似コードを示す。

```
/* MaCC main loop */
1: for (;;) {
2:     FD_SET(l2cap_socket, &readfds);
3:     FD_SET(timer, &readfds);
4:
5:     select(FD_SETSIZE, &readfds, NULL, NULL, NULL);
6:
7:     if (FD_ISSET(l2cap_socket, &readfds)) {
8:         /* receive l2cap packet */
9:         receive(l2cap_socket, buffer);
10:        packet_handler(buffer);
11:    }
12:
13:    if (FD_ISSET(timer, &readfds)) {
14:        discard(timer, trash);
15:        start_inquiry();
16:    }
17: }
```

図 5.3: MaCC メイン・ループ

問い合わせをトリガするタイマは、パイプを利用した親子間プロセス通信で実現している。問い合わせ要求送信中以外はスキャン状態となる。このスキャン状態時、他ノードからのパケットを待ち受ける。

図 5.4 はスキャン状態時にパケットを受信した後に呼ばれ、パケットの種別ごとの処理を行う関数の 1 部である。

```
1: packet_handler(buffer)
2: {
3:     struct message *msg;
4:     msg = (struct message *)buffer;
5:
6:     switch (msg->type) {
7:     AL_REQ:
8:         state_transition(RL_SLAVE, msg->src_bdaddr);
9:         send_al(parent, AL_RES);
10:        break;
11:    AL_RES:
12:        state_transition(RL_MASTER, msg->src_bdaddr);
13:        merge_al(msg->al);
14:        disconnect(msg->src_bdaddr);
15:        if (parent)
16:            send_al(parent, AL_RES);
17:        break;
18:        ----- 以下省略 -----

```

図 5.4: 受信パケット処理部

リレーション確立には呼び出しを行う必要があることを先に述べた。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 RL_SLAVE 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 RL_MASTER 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 RL_MASTER 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、自ノードの PARENT が存在すれば PARENT へ AL_RES パケットを転送する。リレーション状態はコネクション状態へ移行する前であるが、一旦コネクションを確立し、AL_RES パケット受信後 RL_MASTER が切断しコネクション前に戻る。

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新たに発見されたノード

ドにのみ AL_REQ パケットを送信する。

最後に 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を図 5.5 に示す。

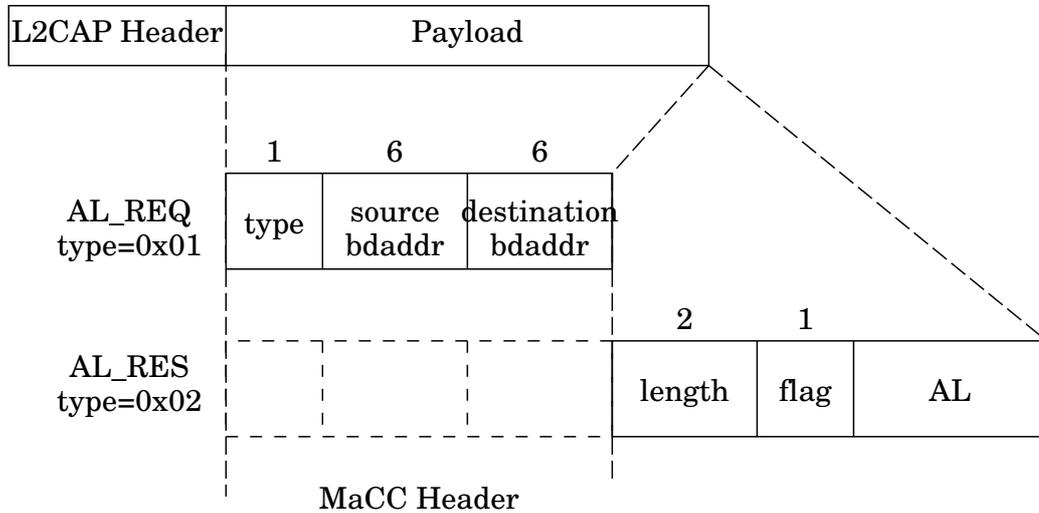


図 5.5: AL_REQ パケット AL_RES パケットのフォーマット図

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。

MaCC は本節であげた 2 種類のパケットの他に 3 種類のパケットを使用する。それについては以降の 5.3 節、5.4 節で述べる。

5.2.2 状態遷移部

本小節ではリレーション確立規則の実装上の実現について述べる。また、先研究である TSF との比較を加えて述べる。

MaCC は TSF の方法を用いて ROOT ノード同士のみリレーション確立が可能であるという規則を実現した。Bluetooth 標準仕様には問い合わせが 2 種類 (GIAC (General Inquiry Access Code) と DIAC (Dedicated Inquiry Access Code)) ある。GIAC はデフォルトの問い合わせである。DIAC を用いることでノードをグループ化することが可能

である。DIAC を設定した場合、その DIAC が設定されたノード間のみ問い合わせ要求 (応答) できる。よって、MaCC は ROOT へ遷移したノードに DIAC を設定することで、ROOT 間のみリレーション確立可能であることを実現した。

リレーションネット内の ROOT 以外は FREE とのみリレーション確立可能とする方法を以下に述べる。この方法は MaCC と TSF では異なる。TSF では、LEAF と BRANCH 状態にあるノードは問い合わせ送信のみ行い、問い合わせスキャンは行わない。すなわち、TSF はネットワーク内で送信とスキャンで対となる 2 つのノードの存在を許さない。これにより TSF は接続規則を実現している。一方、MaCC は MaCC デーモンで処理を行っている。MaCC は AL_REQ パケットおよび AL_RES パケットの送受信の段階でリレーション確立規則を満たすかどうかを調べ、リレーション確立規則を満たす動作を決定する。MaCC は可能な最短経路算出を目的とするため、周辺に存在するノードはネットワーク内外によらず検知する必要がある。

5.3 最短経路検索機能部

本節ではリレーションネット構成後の通信ノード間最短経路検索機能部の実装について述べる。本節の構成はまず隣接リストの実装上での表現を述べ、次に隣接リストから最短経路を求めるアルゴリズム部、最後に経路要求と経路応答のメッセージの流れを説明する。

5.3.1 隣接リストの実装

リレーションネットでの隣接リストは図 5.6 に示す 2 つの構造体で表されている。これらの構造体は単一リストである。

<pre>struct adjacent { bdaddr_t bdaddr; struct adjacent *next; };</pre>	<pre>struct adjacent_list { struct adjacent *adjct; struct adjacent_list *next; };</pre>
---	--

図 5.6: 隣接リスト用構造体

adjacent 構造体は 1 ノードの隣接情報である。adjacent 構造体からなる単一リストには、その単一リストの先頭ノードに隣接するノードの Bluetooth アドレスが含まれる。そして各ノードの隣接情報をさらに単一リストとして adjacent_list 構造体で保持する。adjacnet 構造体および adjacent_list 構造体に情報の追加またはマージを行い、隣接リストを管理する。

5.3.2 最短経路アルゴリズム部

MaCC では最短経路算出アルゴリズムとして単一始点最短路アルゴリズムの1つである Dijkstra のアルゴリズムを用いた。単一始点最短路アルゴリズムは始点 s からの全域木を求める。よって、そこから必要な2点間の最短路を抽出し、結果とする。また、始点から終点までのホップ数も返す。

5.3.3 経路要求および経路応答部

本小節では経路要求パケットおよび経路応答パケットの送受信部の実装を述べる。はじめに経路要求 (R_REQ) パケットおよび経路応答 (R_RES) パケットのパケットフォーマットを図 5.7 に示す。

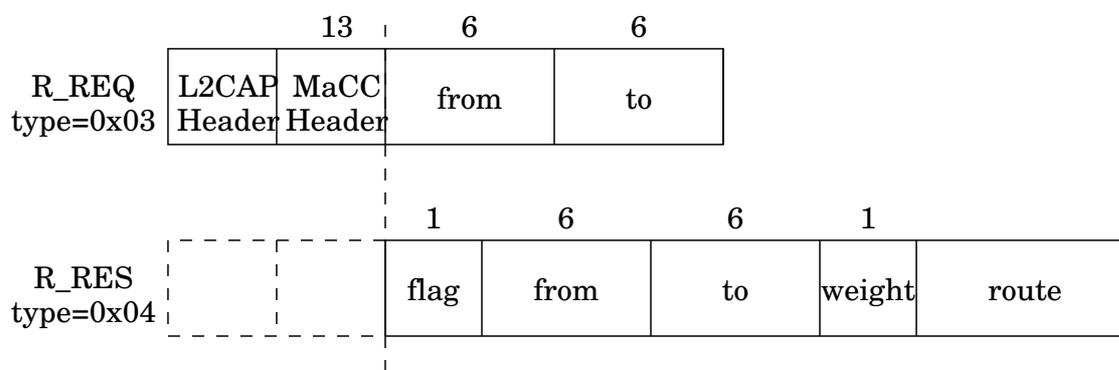


図 5.7: R_REQ パケット R_RES パケットのフォーマット図

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。

図 5.8 は経路要求関数であり、図 5.9 は経路応答関数である。経路要求関数は通信開始直後および R_REQ パケット受信後に呼び出され、経路応答関数は R_RES パケット受信後に呼び出される。

経路要求を行うノードははじめに自ノードが保持する隣接リストから最短路算出を試みる。そこで最短路が求まらなければ PARENT ノードへ R_REQ パケットを送信する。R_REQ パケット送信前に要求内容をキャッシュする。キャッシュする情報は、child:1 階層下の R_REQ パケット送信元ノード、from:要求する経路の始点 Bluetooth アドレス、to:要求する経路の終点 Bluetooth アドレスである。これにより連続して同一の経路要求が起こった際重複した R_REQ パケットの送信を防ぐ。キャッシュされた情報は対応する R_RES パケット受信後に廃棄する。R_RES パケットは ROOT での経路検索が失敗した場合にも flag が NOROUTE に設定され、返信される。

```

1: route_request (child, from, to)
2: {
3:     if (Dijkstra != NULL) {
4:         if (child == from)
5:             return GOTROUTE;
6:         else {
7:             route_response(child, from, to, route, flag);
8:             return RESROUTE;
10:        }
11:    }
12:    else {
13:        cache_route_request(child, from, to);
14:        send_route_request(parent, from, to);
15:    }
16:    return REQUESTING;
17: }

```

図 5.8: 経路要求関数

経路応答関数は自ノードが要求経路の始点であるか否か进行检查し、さらに R_RES パケットを転送する必要がある場合は 1 階層下の R_REQ パケット送信元ノード (child) へ転送する。自ノードが経路の始点であれば取得した経路のキャッシュを行う。これにより、同一経路の要求が起こった際の効率を上げる。また、この関数は対応する経路要求情報のキャッシュがあればそれを廃棄する。

5.4 コネクション確立機能部

本節ではコネクション確立機能部の実装について述べる。MaCC はコネクションを確立し、また経路選択も行う。パケットに含まれた経路情報をもとに次ホップ先を決定し転送する。また、コネクション確立機能部はアプリケーションからのデータを処理し、またアプリケーションへの転送も行う。アプリケーションから送信すべきデータを受け取ると MaCC は送信バッファ (sbuf) へ格納する。経路要求を行い、経路結果を得ると送信バッファにあるデータをすべて送信する。この送信時に経路情報をパケットに書き込む。

図 5.10 に MaCC で扱われるデータパケットのパケットフォーマットを示す。

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 アドレスから転送時に次ホップ先の Bluetooth アドレスを示すよう設定されるポインタである。実際には pointer フィールドの値

```

1: route_response (child, from, to, route, flag)
2: {
3:     if (route_req = exist_rreq_cache())
4:         delete(route_req);
5:     if (child == mybdaddr) {
6:         cache_route(route);
7:         return GOTROUTE;
8:     }
10:    else {
11:        send_route_response(child, route);
12:        return RESROUTE;
13:    }
14: }

```

図 5.9: 経路応答関数

		13	1	1	2		
DATA type=0x05	L2CAP Header	MaCC Header	weight	pointer	data_length	route	data

図 5.10: データパケットフォーマット

はホップするごとに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。

経路のキャッシュはデータ伝送路が切断していた場合に廃棄される。コネクション機能がコネクションを確立できずタイムアウトした場合、data_length フィールド値を負の値に設定し、コネクション確立が経路中で停止した地点から route フィールドの値を反転させ、データ送信元へ戻される。データパケットを受信し、data_length フィールドが負であったならばそのパケットを廃棄および該当する経路のキャッシュを削除する。

コネクション確立としてピコネットおよびスキヤッタネットを構築した場合、ピコネット数が最小となるべく構成することを第 4 章の設計で述べた。しかし、BlueZ で

は L2CAP からマスタかまたはスレーブかを設定できない．コネクションを要求するノードがマスタとなる．また，本研究で使用した Bluetooth デバイスはスキヤッタネットワークを構築できないため，プロトタイプ実装もマルチホップ対応ではない．スキヤッタネットワークを構築可能な Bluetooth デバイス上への MaCC 実装は今後の課題とする．

最後に，通信を行うアプリケーションがデータを送信して，それを受けて MaCC が経路を確立し，宛先アプリケーションが受信するまでの全体の流れを図 5.11 に示す．これは X:LEAF, Y:BRANCH, Z:ROOT の 3 台での X が Z へ通信する場合の MaCC による各々の処理を示した例である．また，Y では経路を割り出せず，Z まで経路要求の送信が必要ある場合である．楕円で囲ってあるものは伝送されているパケットの種類を示し，矢印が各処理を示す．

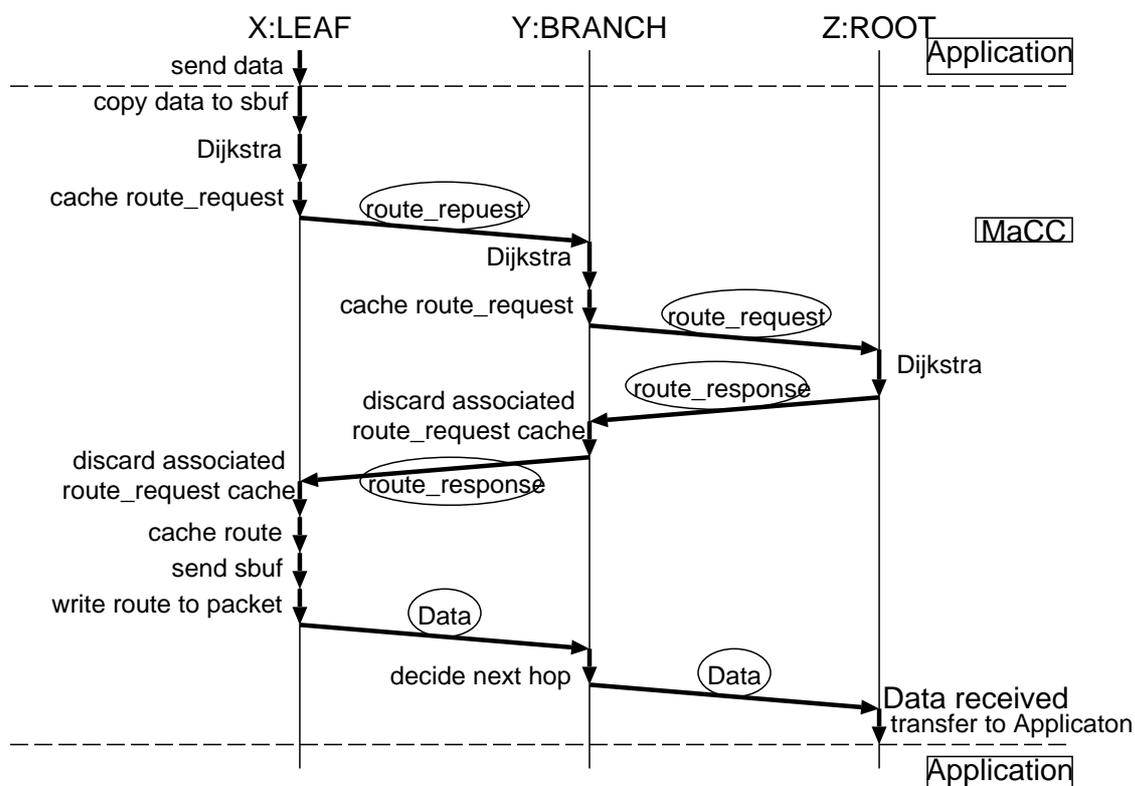


図 5.11: MaCC シーケンス

第6章 評価

本章では，MaCC プロトタイプ実装の評価を述べる．はじめに測定環境を示し，加えて Bluetooth 通信での 1 ホップの RTT(データ伝送往復時間) 値の測定結果を示す．次に測定項目をあげ，測定方法および測定結果を述べる．また測定結果をもとにフラッシング方式のルーティングプロトコルとの比較を示す．

6.1 測定環境

本研究で行った測定にはノート PC3 台 A,B,C を使用し，それぞれに ROOT , BRANCH , LEAF を割り当てた．表 6.1 に各ノート PC の環境を示し，表 6.2 に本研究中使用した Bluetooth デバイスをあげる．表 6.2 には各 Bluetooth デバイスが Bluetooth ベースバンド層のいくつかの機能に対してサポートしているか否かの比較も示す．一般に CF 用 Bluetooth カードは USB タイプに比べ機能が少ない．また，スキャッタネットに関してはほとんどのデバイスがサポートしていないのが現状である．本研究での測定には Brain Boxes BL-554 Bluetooth USB アダプタを用いた．

表 6.1: マシン環境

マシン	A	B	C
CPU	Pentium 3 866MHz	Celeron 750MHz	Pentium 4500MHz
メモリ	380MB	128MB	192 MB
OS	RedHat Linux 7.3	RedHat Linux 7.3	RedHat Linux 7.3
カーネルバージョン	2.4.18-17	2.4.18-3	2.4.18-3
Bluetooth stack	bluez-kernel-2.3	bluez-kernel-2.3	bluez-kernel-2.3

6.1.1 Bluetooth 通信の RTT 値

図 6.1 は Bluetooth のパケットタイプごとの RTT 値である．RTT 値は BlueZ が提供する `l2ping` を使用し，300 回の要求と応答を行い，`l2ping` が出力する RTT 値の平

* ノードは同時に異なるピコネットでマスタとスレーブにはなれるが，スレーブとしてのみ複数ピコネットに属することはできない

表 6.2: Bluetooth デバイス

製品名	BL-554	3CREB96	Bluetooth Starter Kit	BL4508-309	IbCARD
Bluetooth バージョン	1.1	1.1	1.1	1.1	1.1
チップセット	CSR	CSR	Ericsson	CSR	Ericsson
マルチスロットパケット					
暗号化					
マスタ/スレーブ変換					×
省電力モード				×	×
SCO リンク				×	
パワーコントロール				×	×
スキップネット	×	×	*	×	×

均値を求めた。

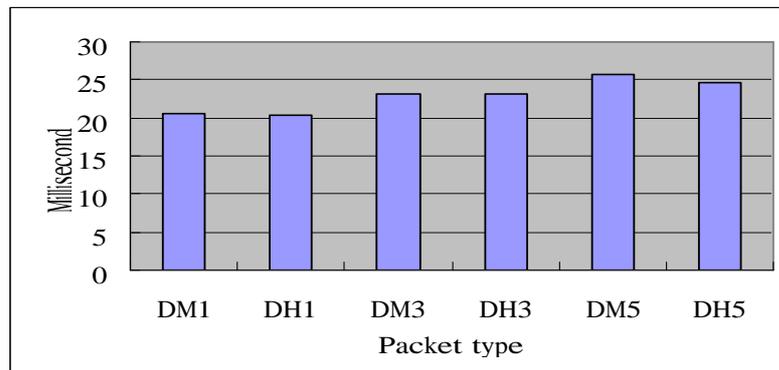


図 6.1: パケットタイプごとの RTT 値

6.2 測定

本節では MaCC プロトタイプの測定項目，測定方法および測定結果を示す．本研究では MaCC の各処理における処理所要時間を測定した．そこから Bluetooth 自体による処理時間と MaCC のみの処理に要する時間を分けた．

6.2.1 測定方法

本小節は測定方法を示す。MaCC が行う各処理の開始時点，終了時点に Pentium Counter の現在値を取得し，それらの差をその処理にかかる所要時間，すなわちオーバヘッド値とする。図 6.2 は Pentium Counter の値を取得し，各処理にかかるクロック数を取得する操作を示す。取得したクロック数は，CPU クロック周波数で割り，単位を時間へ変換した。

```
#define RDTSC(x) asm("rdtsc" : "=A" (x))

unsigned long long int count1, count2;

RDTSC(count1);
/* 計測する処理 */
RDTSC(count2);
printf("process time = %lld\n", count2-count1);
```

図 6.2: RDTSC マクロ

6.2.2 測定結果

本小節で，測定した MaCC の各処理の所要時間 (ミリ秒) を示す。図 6.3 は MaCC の処理のシーケンスである。これは A が LEAF，B が BRANCH，C が ROOT としてリレーシオンネットを構成した場合である。3 台でリレーシオンネット構成後，A が B からは発見できないノードへの経路を要求し，その検索要求が ROOT である C へ届き C が経路応答を返信して A が受信する，という試行を 20 回行なった。楕円で囲まれた数字は各処理の所要時間 (ミリ秒) である。

所要時間が示されている処理は 0.01 ミリ秒以下であり RTT 値に比べても十分小さい。また，キャッシュ情報検索にかかる実行時間は $O(V)$ である。したがって所要時間が小さく，また MaCC における主要な処理部ではないため，問題となるオーバヘッドとは考えない。そこで MaCC によってオーバヘッドとなる部分を斜線部の処理とする。斜線部は，Dijkstra のアルゴリズムを実行する経路検索部とコネクション確立処理部を示す。以降で経路検索部における所要時間とコネクション確立遅延についての測定結果を示す。

経路検索部所要時間

MaCC が経路検索時に実行する Dijkstra のアルゴリズムの所要時間計算結果を図 6.4 および図 6.5 に示す。両図ともノード数 2 台から 20 台まで変化させたシミュレーション

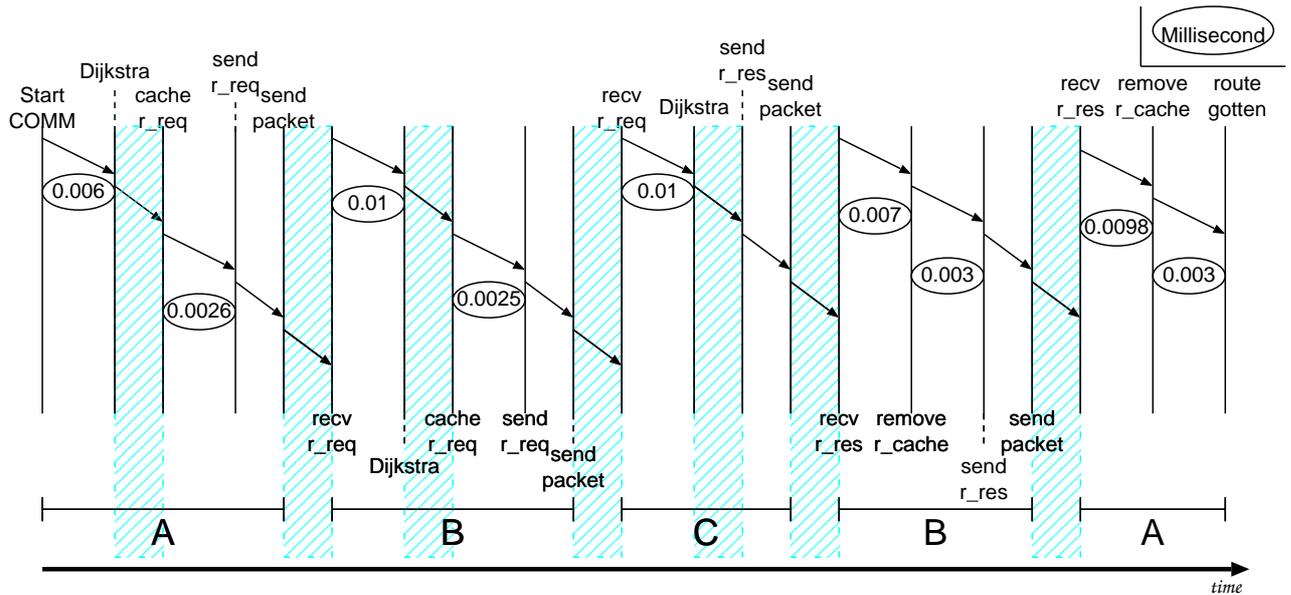


図 6.3: MaCC 実行例および処理時間

ンでの各ノード数におけるアルゴリズム実行時間である．図 6.4 は各ノード数のネットワークが完全¹な場合を示し，図 6.5 は LEAF および ROOT が各々1つであり，他のノードは BRANCH であるネットワークに対する測定である．Dijkstra のアルゴリズムの実行時間は $O(\text{ノード数}^2 + \text{辺の数})$ であるので，図 6.4 は実行時間が $O(2 * \text{ノード数}^2)$ と最も処理に時間を要する場合である．また，図 6.5 は $O(\text{ノード数}^2 + (\text{ノード数} - 1))$ となり最小の実行時間となる場合である．

コネクション確立遅延

表 6.3 はコネクション要求コマンドの発行からコネクション確立を通知するイベント取得までの所要時間を 100 回測定した結果である．

Bluetooth はコネクション確立までに問い合わせと呼び出しを行う必要がある．ここで示しているのは，呼び出しからコネクション状態へ遷移するまでの時間である．呼び出し段階までは周波数ホッピングパターンに送信側と受信側で同期がとれておらず，送信側と受信側とで周波数帯が一致させるための処理に時間がかかる．Bluetooth では周波数ホッピングパターンを一致させるために必要な制御パケットの送受信を成功させるまでが大きなオーバーヘッドとなっている．

¹すべてのノード間が隣接している状態

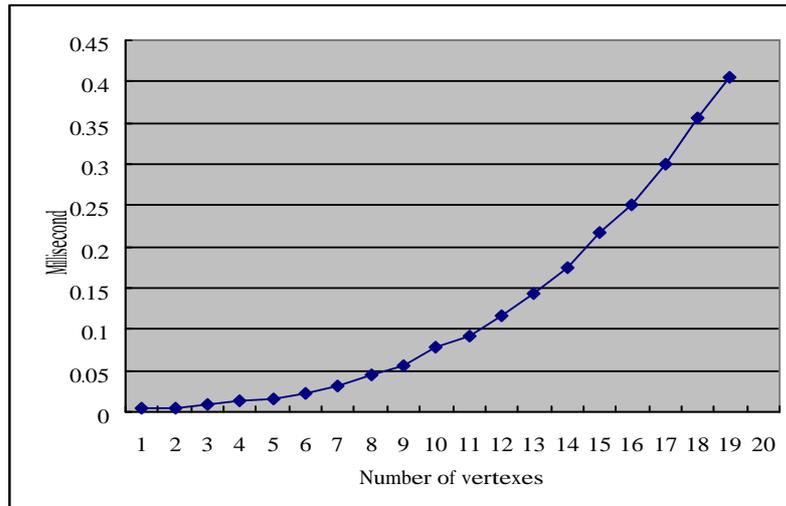


図 6.4: Dijkstra 実行時間 (完全グラフ)

表 6.3: コネクション確立遅延測定結果

平均値	標準偏差	最大値	最小値
1.6357(秒)	0.84611	4.4091(秒)	0.1413(秒)

6.3 フラッディング方式との比較

本節では MaCC プロトタイプ の各処理にかかる所要時間の測定結果をもとに、フラッディング方式で経路検索を行った場合を仮定してそれとの比較について述べる。

ネットワークのモデルとして以下 3 つを用いる。MaCC に対するベストケース、ワーストケース、アベレージケースである。

- 完全グラフ (ベストケース)
- 次数 ≤ 2 の木 (ワーストケース)
- 完全 2 分木 (アベレージケース)

完全グラフ

MaCC

ノードが保持する隣接リストが含むノードの数 V に対する Dijkstra のアルゴリズム実行所要時間を、 $D(V)$ とおく。完全グラフのとき MaCC は自ノードが保持する隣接リストですべての宛先を発見できるため、経路発見に要する時間は、

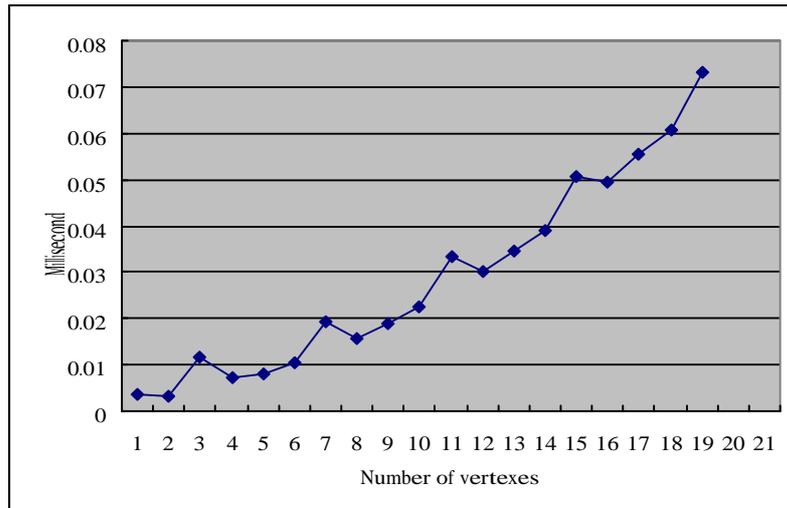


図 6.5: Dijkstra 実行時間 (木グラフ: 次数 ≤ 2)

$$T_{\omega} = T_1 = D(V)$$

となる .

フラッディング方式

フラッディング方式ではすべてのノードが 1 ホップで到達できるため , 経路検索に要する時間は ,

$$T_{\omega} = T_1 = rtt$$

となる . フラッディング方式ではネットワークはコネクション確立しているものと仮定する . 先行研究にあるフラッディング方式はコネクションを確立したネットワークを前提にしている . また , 転送にかかるオーバーヘッドを考えない . すなわち , マルチホップする場合であってもデータ伝送に要する時間のみを考える .

図 6.4 のグラフの近似式を用いるとノード数約 110 までは , $D(V) < 25(ms)$ となり MaCC の方がより経路検索にかかる時間が少ない .

次数 ≤ 2 の木

次数 ≤ 2 の木は , 一本のパスのみからなる . リレーシオンネットでは , 次数 1 のノードを LEAF および ROOT , 次数 2 のノードすべてを BRANCH とする . ここで , LEAF から ROOT への経路要求を LEAF が行う場合を考える . C はコネクション確立遅延である . ω ホップの経路検索所要時間は以下となる .

MaCC

$$T_{\omega} = rtt \times (\omega - 1) + C \cdot (\omega - 1) + \sum_{i=1}^{\omega} D(i)$$

フラッディング方式

$$T_{\omega} = rtt \times \omega = rtt \times (\omega - 1) + rtt$$

C は平均約 1.6 秒であり，フラッディング方式が有効であることは明らかである．ここで仮に C を 0 として比較を行う．その場合ノード数約 300 程度までは MaCC の経路検索所要時間がより少ない．Bluetooth ではコネクション確立遅延が大きなオーバーヘッドであることが分かる．

完全 2 分木

完全 2 分木は左右対称の構造であり，木構造としてその構造が簡潔であるため，ネットワークモデルとして用いた．完全 2 分木の根が ROOT であり，階層構造を有している．そのため，上位層のノードであれば経路上のホップ数に対し，経路要求ではより少ないホップ数で可能である．したがって， T_{ω} は以下のようなになる．

MaCC

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

ただし，

$$\Theta^n(v) = \underbrace{f \circ f \circ f \circ \dots \circ f}_n(v), \quad \Theta^0(v) = v$$

$$f(v) = 2v + 1$$

ω ホップの経路を検索する場合は $\omega + 1$ 通りあり，その平均を T_{ω} とした． $\Theta^n(v)$ は，基準ノード数 v から n 階層上位のノードが保持する隣接リストが含むノード数である．1 階層上位へいくごとにノード数はもとのノード数の 2 倍に 1 加えた数になる．完全 2 分木において例として 3 ホップの経路を検索する 4 通りを図 6.6 に示す．各 2 分木の下にある数字は経路要求時に必要なホップ数である．塗りつぶされたノードでは Dijkstra のアルゴリズムが実行される．太線は得られる経路の 1 例である．

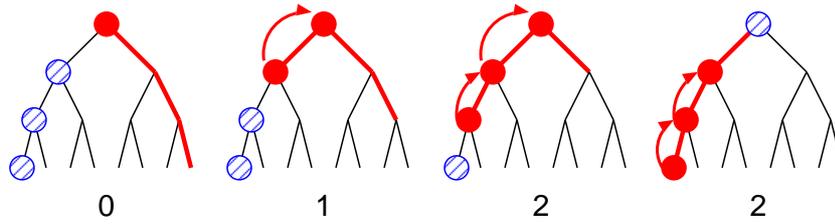


図 6.6: 3 ホップ経路の要求例

フラッディング方式

$$T_{\omega} = rtt \times \omega$$

C が大きい場合フラッディング方式による経路検索所要時間がより小さい．そこでまた C を 0 と仮定する．その場合の MaCC およびフラッディング方式の経路検索所要時間を図 6.7 に示す．このとき，RTT 値は 20ms とした．

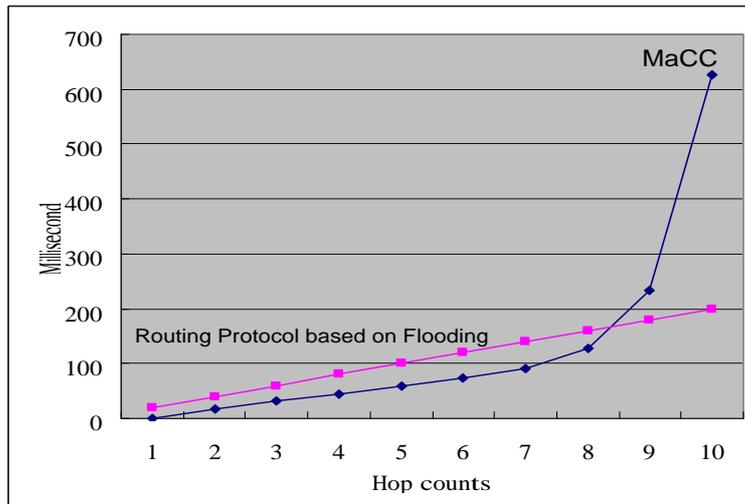


図 6.7: MaCC およびフラッディング方式による経路検索所要時間

図 6.7 によると 8 ホップまでの経路を検索する場合は MaCC の処理時間がより小さい．すなわち，完全二分木で約 500 台までのネットワークであれば MaCC が勝る．よって，MaCC のみから生じるオーバーヘッドは十分小さい．

しかし，実際はコネクション確立遅延がありフラッディング方式が勝る．Bluetooth 依存のコネクション確立遅延が最も大きなオーバーヘッドである．

また，Bluetooth を用いる場合コネクション確立遅延に加え，周辺に存在するノードの検知にもコネクション確立遅延同等かそれ以上の時間を要する．よって，Bluetooth

通信を利用するシステムはこれら Bluetooth 特有の遅延を考慮して設計を行う必要があることが分かる．これに関してはまた 7.2 節で述べる．

第7章 結論

本章では、本論文のまとめを述べ、最後に今後取り組むべき課題を述べる。

7.1 本論文のまとめ

本論文では、WPANsの説明とその特性を示し、従来のルーティング方式はWPANsに適さないことを述べ、WPANsに特化したルーティングを提供するMaCCの設計および実装と評価について述べた。

WPANsはマスタ駆動型ネットワークであるため、マスタをルータとするルーティング方式はマスタに負荷を集中させることに繋がる。WPANsは構成する通信端末機の種類を制限しない。多くの機器が自由にネットワークを構成できる。よって負荷が集中するマスタにその負荷に耐えうる機器を用意したとしても、WPANsの柔軟性を損ねることになる。フラッディング方式は完全自律分散型な手法であるためWPANsにおいても有用である。しかし、ネットワーク全体へ負荷を与えることになる。WPANsではその特性上マスタがネットワークの情報を保持している。よって、その情報を利用すれば必ずしもフラッディングを行う必要はない。

そこで本研究ではマスタ駆動型のWPANsに特化したルーティング機能を有するMaCCを提案した。MaCCはマスタのネットワーク内の隣接情報を、複数ネットワークが結ばれ拡張されたインターネットワーク内で階層型集中管理させ、その隣接情報から最短経路を求める。MaCCは経路要求をネットワーク全体に行わず、かつマスタの負荷を軽減する。

本研究ではMaCCのプロトタイプ実装を行い、それに対する評価を行った。今回の評価では処理時間の計測を行った。フラッディング方式による場合をRTT値をもとに仮定し、それに対してMaCCを適用するネットワークモデルを3つ用意し、それぞれのモデルごとにおける比較および考察を述べた。

7.2 今後の課題

隣接情報の更新と再構成に関する問題

現段階では例えばあるBRANCHが停止したとしても、そのPARENTは停止したBRANCHを認識できるが、そのBRANCHより以下の階層の情報を失ったと判断できない。すなわち、上位層のノードはどの隣接情報がどの下位層のノードからの情報で

あるかを識別できない．これは隣接リストの構造体に識別子としての要素を加えることで対応できる．

リレーションネットがあるノードの欠落により分解する可能性がある．その際の状態遷移，および新たなリレーションネットの再構成を実装する必要がある．

QoSルーティング

通信の種類は一様ではない．ストリーミングのような通信継続時間が長い通信，ファイル転送などの通信，スイッチのオン/オフなど1パケットで可能な通信等がある．そこで1パケットのみの通信に対してはMaCCの経路検索はオーバーヘッドでしかない．一方ストリーミングは継続し，帯域も必要とする．よってストリーミングを行う通信経路が重複した場合，互いに帯域を狭めてしまう．その場合，仮に最短経路でなくとも，重複を回避した経路が必要である．しかし，MaCCは最短経路を求めるのみでどのような通信であるかを考慮していない．Bluetoothであればプロトコル・サービス多重識別子(PSM:Protocol Service Multiplexer)でTCP/UDPにおけるポートのようにサービスを識別できる．これを利用すれば通信の種類を大別できる．しかし，通信の種類，継続時間を特定は困難であり，さらに検討する必要がある．

ノード数の問題

MaCCは家庭内で構成されるWPANsを想定し，そのノード数も数十台とした．しかし，家庭内でも温度計や赤外線センサ等の各種センサも同様にネットワークを構成するとノード数がさらに多くなる可能性がある．ROOTが保持する情報量が増加し，MaCCが用いるDijkstraのアルゴリズムはノード数の自乗で実行時間が増加する．よってMaCCに対し，ノード数増加に対するスケーラビリティを実現する改良が必要である．

未実装部

MaCCプロトタイプ実装はキャッシュに対しタイムアウトを設定していない．明示的なキャッシュを削除するイベントが発生しない限り半永久的にキャッシュは残る．しかし，必ずしもイベントが発生するとは限らない．よってタイムアウトによって廃棄される部分を加え，有効なタイムアウト値を設定する必要がある．

またMaCCはネットワーク機能を提供するためカーネルに組み込むことが最善と考えられる．本研究ではプロトタイプとしてデーモンプロセスとして実装を行ったが今後カーネルモジュールとして実装を進める．

評価

本研究では評価の基準として経路検索にかかる時間のみオーバーヘッドとして測定した。しかし、消費電力測定可能なデバイスを用いて消費電力についても測定する必要がある。また、ネットワーク全体にかかる負荷をシミュレータなどを利用して評価を行う。経路検索時にどの程度のデータがネットワークに流れるかの測定を行う。

マルチホップ通信

本研究で使用した Bluetooth デバイスはスキッタネットを構築できず、マルチホップ通信を行えない。そこでプロトタイプ実装でもマルチホップ通信に対応していない。今後スキッタネット構築可能なデバイス上で MaCC 実装を行う。

謝辞

本研究を進めるにあたり，研究指導をはじめあらゆる面でご協力して下さった慶應義塾大学環境情報学部徳田英幸教授に深く感謝致します．また，東京電機大学工学部情報メディア学科戸辺義人助教授から多大なる御指導および御助言を頂き，深く感謝致します．また，日頃の研究および論文作成にあたり，常に御指導・御助言をして下さった慶應義塾大学大学院政策メディア研究科修士間博人氏および同政策メディア研究科修士齊藤匡人氏に感謝の意を表します．

慶應義塾大学徳田研究室 ECN 研究グループの諸氏，および徳田・村井・楠本・中村・南研究室の方々より数多くの御助言を頂きましたこと拝謝致します．

最後に，研究の日々を共にした，青木俊氏，青木基衣氏，門田昌哉氏，志和木愛子氏，鈴木源太氏，高橋元氏，高橋ひとみ氏，田丸修平氏，中山裕佳子氏，村上朝一氏，峰松美佳氏，守分滋氏，その他多くの友人に感謝致します．

2003年2月22日

滝澤 允

発表論文

- [1] 滝澤 允, 間 博人, 斉藤 匡人, 戸辺 義人, 徳田 英幸. "通信状況に応じた WPAN
コンフィグレーションシステムの提案". 情報処理学会マルチメディア通信と分散
処理研究報告 No.110 - 012 . 2002/11/21-22

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], March 2001.
- [4] Federal Communication Commission. *Revision of Part 15 of the Commission's Rules Regarding Ultra-Wideband Transmission Systems*, April 2002.
- [5] IEEE 802.11 Standard (IEEE Computer Society LAN MAN Standards Committee). *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, June 1999.
- [6] IEEE 802.15 Working Group for WPAN. <http://grouper.ieee.org/groups/802/15>.
- [7] Mobile Ad-hoc Networks Working Group (MANET). <http://www.ietf.org/html.charters/manet-charter.html>. Charter [Work in Progress], 2002-09-16.
- [8] 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.
- [9] BlueZ project Official Linux Bluetooth protocol stack. <http://bluez.sourceforge.net/>.
- [10] Theodoros Salonidis, Pravin Bhagwat, Leandros Tassiulas, and Richard LaMaire. Distributed Topology Construction of Bluetooth Personal Area Networks. In *Proceedings of IEEE INFOCOM*, April 2001.

- [11] 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.
- [12] The Bluetooth Special Interest Group. <http://www.bluetooth.com>.