

WLAN と ZigBee の共存に向けた AP-Assisted CTS-Blocking に関する研究

佐伯 良光

平成 27 年 2 月

修士課程

情報知能工学専攻

社会情報システム工学コース

概 要

現代社会において、無線ネットワークによる通信は生活に不可欠な存在である。特に近年、無線 LAN (Wireless LAN, WLAN) 通信はモバイル端末の普及などの要因もあり、広く一般に利用されている。一方、別のネットワーク通信として ZigBee が注目されている。ZigBee 端末は他の無線ネットワーク端末に比べ安価で省電力であるため、センサネットワークを用いた M2M (Machine to Machine) 技術に利用される。

しかしながら、WLAN と ZigBee では同一の周波数帯を利用するため、干渉が発生し通信が阻害されるため共存が難しい。そこで、同一周波数帯を利用する WLAN と ZigBee の共存に向けて容易に構築可能な衝突回避方式の研究を行っている。衝突回避方式の実現に向けては既存の機器をそのまま利用することが必要となることから、OS やハードウェアの改変を必要としないシンプルな方式の実現が重要となる。

本論文ではこのような観点で設計された AP-Assisted CTS-Blocking (AA CTS-Blocking) 方式を示す。AA CTS-Blocking 方式では RTS/CTS 方式を応用することで WLAN の通信を抑制し、ZigBee 通信と WLAN 通信の衝突を回避させる。RTS/CTS 方式は WLAN の標準に定められた衝突回避機構であり、所定の手順を踏んで RTS/CTS フレームを送信することで OS やハードウェアの改変を行うことなく WLAN と ZigBee の衝突を抑制することができる。AA CTS-Blocking 方式を適用した ZigBee データ収集システムを実装し、実証評価を通じて AA CTS-Blocking 方式の衝突回避効果により ZigBee 通信の成功率が向上したことを確認した。

目次

第1章	はじめに	1
第2章	WLAN 及び ZigBee におけるネットワーク・プロトコル	6
2.1	TCP/IP	6
2.2	TCP/IP 参照モデル	7
2.3	Media Access Control	10
2.3.1	多元接続制御方式	10
2.3.2	WLAN における MAC プロトコル	12
2.3.3	ZigBee おける MAC プロトコル	13
第3章	関連研究及びその比較	18
第4章	AP-Assisted CTS-Blocking 方式の提案	21
4.1	概要	21
4.2	AP 選択アルゴリズム	22
4.3	スケジューリング手法	23
第5章	設計及び実装	26
5.1	ハードウェア	26
5.1.1	ZED	26

5.1.2	ZC 及び制御 PC	27
5.2	ZigBee ノードによるデータ収集システムの設計	28
5.2.1	nesC	30
5.2.2	ステートマシン図	32
5.2.3	コンポーネント図	34
5.3	AA CTS-Blocking 方式の実装	38
第 6 章	評価	40
6.1	予備実験	40
6.2	評価環境	42
6.3	マクロな ZigBee 通信成功率の比較	43
6.4	ミクロな ZigBee 通信成功率の比較	43
6.5	考察	44
第 7 章	おわりに	48
	謝辞	49
	参考文献	50
	発表論文	52

第 1 章

はじめに

現代社会において，無線ネットワークによる通信は生活に不可欠な存在である．特に近年，無線 LAN（Wireless LAN，WLAN）通信は Wi-Fi 等で広く一般に普及している．例えば PC とプリンタを，自宅の WLAN アクセスポイント（Access Point，AP）に接続し印刷を行うことは珍しくない．それだけにとどまらず，更に HDD，テレビ，レコーダーなど，様々なデジタル機器を WLAN ネットワークに接続して利用するホームネットワークという概念も登場している．近年では，WLAN 通信端末の主体がスマートフォンやタブレットで代表されるモバイルワイヤレス端末に移行しつつある．コンビニエンスストアや駅など，不特定多数の人々がゲストとしてアクセス可能な WLAN AP の設置も進んでいる．特定の場所に限らず，あらゆる地域・場所で WLAN ネットワークに接続して通信することが求められるようになってきている．

他方で，別のネットワーク通信として ZigBee が注目されている．ZigBee とは，IEEE 802.15.4 に準拠したセンサーネットワークを主目的とする近距離無線通信規格の 1 つであり，WLAN に比べ通信距離が短く通信速度も低速であるが，安価で省電力である．それ故に，ネットワークに繋がれた機器同士が人間を介在せずに通信し，サービスを提供する M2M（Machine to Machine）技術によく利用される．M2M の応用範囲は非常に幅広く，

多様なサービスを提供可能である．屋内に ZigBee ネットワーク接続されたセンサを配置して，電力やガスの消費量や，温度を最適に自動制御するスマートハウスも M2M の 1 つである．

ネットワーク技術の進歩に伴い人々の暮らしが益々便利になる一方で，通信フレームの衝突による干渉という課題も存在する．例えば，ZigBee センサにより構築されたスマートハウスを利用する場合，屋内に WLAN AP が存在すれば干渉が発生する可能性は高い．干渉とは，電波技術 A に対応した受信機 Tx が電波 A と電波 B の混在した電波を受信した際，正常に電波をデコードできないことを指す．これは ZigBee が，WLAN と同じ 2.4GHz 帯を利用するためである．図 1.1 に，WLAN 及び ZigBee の周波数チャンネルを示す．ZigBee のすべてのチャンネルは WLAN の複数のチャンネルと重なっており，ZigBee と WLAN の中心周波数が近いところでは相互に干渉が生じることが報告されている [1]．WLAN は ZigBee に比べて 10～100 倍程度送信電力が大きいので，干渉による通信への影響は ZigBee 側が大きい．

WLAN, ZigBee にはアクセス制御方式として CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance) 機構を具備している．しかし，WLAN の CSMA/CA では WLAN のみの通信環境，ZigBee の CSMA/CA では ZigBee のみの通信環境を想定しているため，WLAN 通信と ZigBee 通信が混在した環境において互いの干渉を回避することは困難である．

WLAN 通信と ZigBee 通信が混在した環境内における WLAN と ZigBee の共存を目的とした干渉回避方式として，文献 [2] では WLAN のアクセス制御である RTS/CTS (Request to Send / Clear to Send) 方式を利用した CTS-Blocking が提案されている．図 1.2 に，CTS-Blocking の概要を示す．CTS-Blocking では，制御 PC から CTS フレームを直接送信することで周囲の WLAN 端末の通信を一時的にブロックする．WLAN 通信において，送信端末以外の端末が CTS フレームを受信すると CTS フレーム内の Duration フィールドに記載された時間だけ送信を控える．CTS-Blocking ではこれを利用して周囲の WLAN 端末

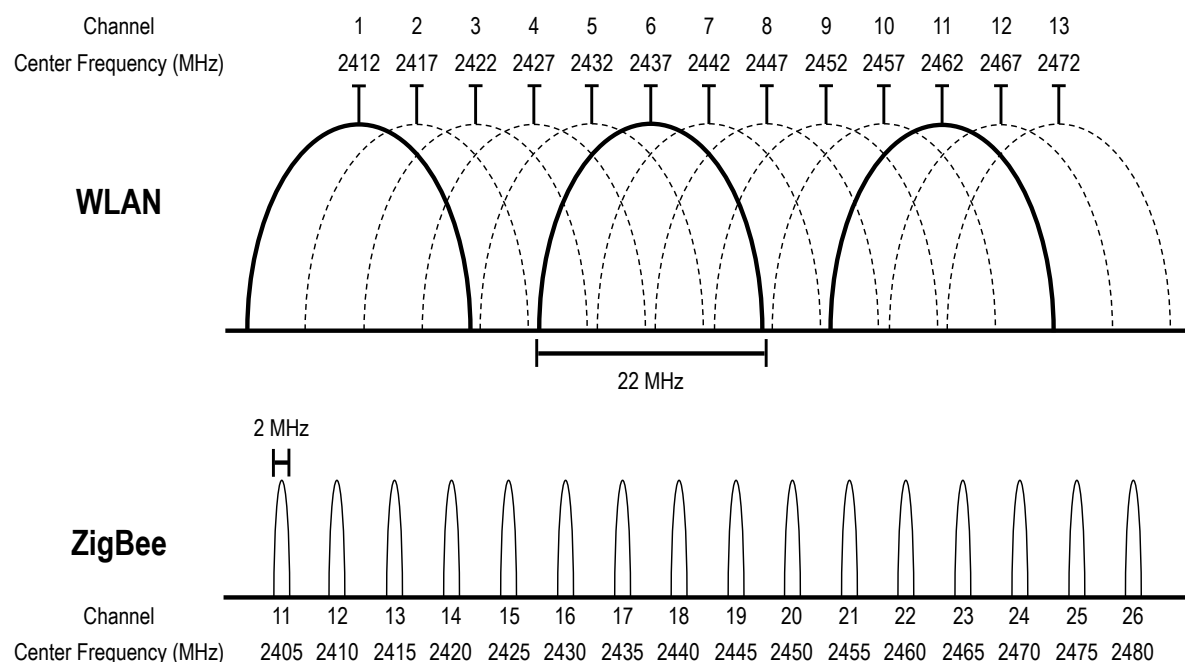


図 1.1: WLAN 及び ZigBee の周波数チャネル

の通信を抑制し，WLAN と ZigBee の干渉を回避する．しかしながら，現在の OS・無線 LAN モジュールでは通信の公平性確保の観点から CTS フレームの直接送信が禁止されており，CTS-Blocking の実現に向けてハードウェアや OS の改変が必須となる．また，送信電力制御の影響により CTS フレームの到達範囲が狭くなる可能性があり，隠れ端末問題の影響を受けやすい．

WLAN と ZigBee の共存のためには，ハードウェアや OS を改変することなく，既存のモジュールをそのまま用いて WLAN からの干渉を受けない ZigBee 通信を実現することが重要である．これに向け，筆者らは既存の RTS/CTS 方式をそのまま利用した AP-Assisted CTS-Blocking (AA CTS-Blocking) 干渉回避方式の開発を進めている．本稿では，AA CTS-Blocking 方式について述べ，その有効性の検証に向けた実証評価について報告する．

本論文の構成は以下の通りである．2 章では WLAN 及び ZigBee の基礎技術について触れ，アクセス制御方式などの説明を行う．3 章では WLAN と ZigBee の共存に向けた干渉回

避方式に関する研究について述べ、既存システムの課題を示す。4 章では筆者らが提案する AA CTS-Blocking 方式の概要及び設計について示す。5 章では AA CTS-Blocking 方式を適用した ZigBee ノードによるデータ収集システムの実装について述べ、6 章で WLAN 通信環境下における ZigBee ノードによるデータ収集システムについて、AA CTS-Blocking 方式の有無による通信成功率の差を比較・考察する。最後に 7 章でまとめとする。

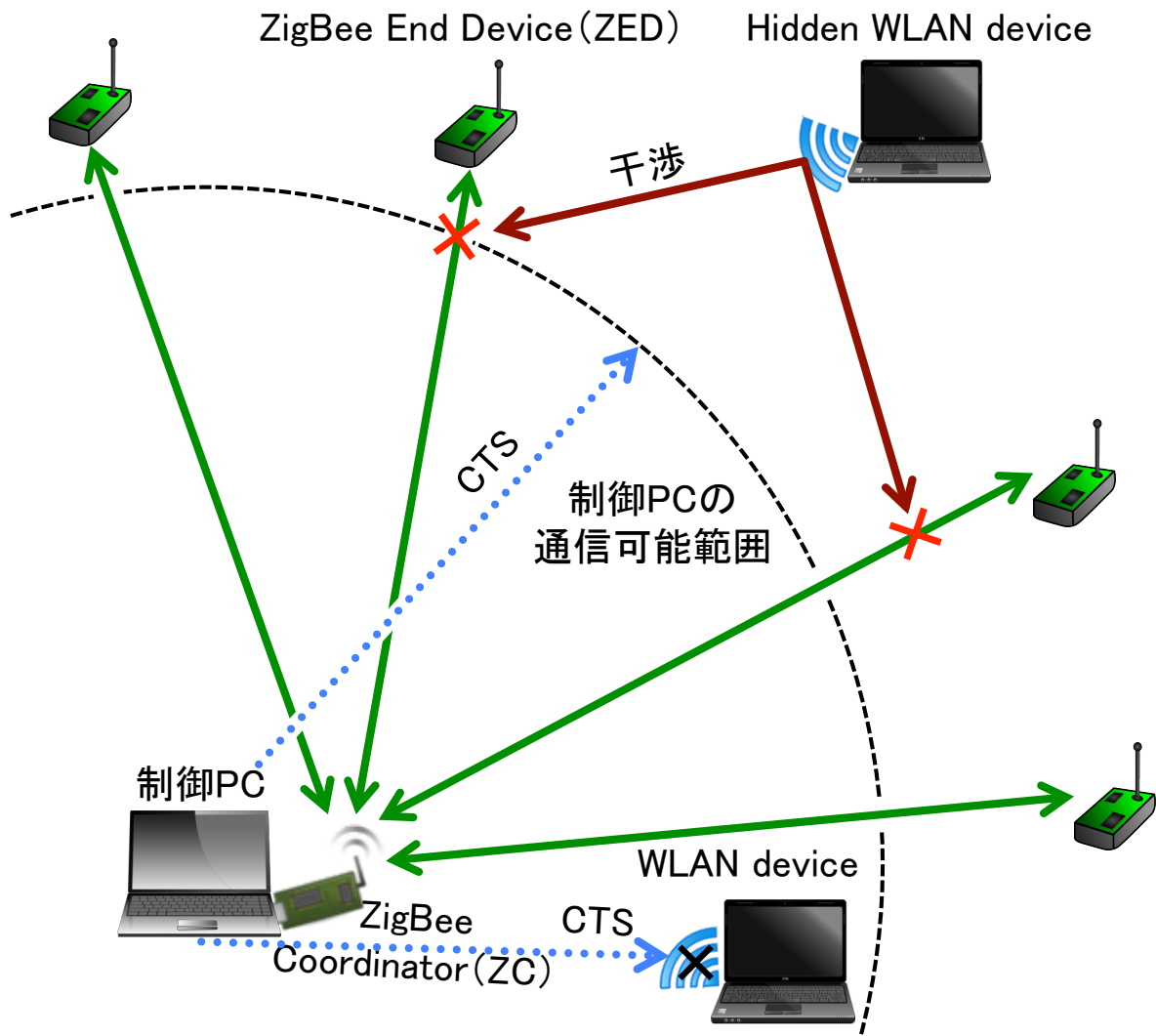


図 1.2: CTS-Blocking と隠れ端末問題

第 2 章

WLAN 及び ZigBee におけるネットワーク・プロトコル

本章における研究対象を明確に示すため、ネットワーク上での通信に関する規約を定めたものであるネットワーク・プロトコルについて説明を行う。2.1 節では現在の標準である TCP/IP プロトコルについて簡単に触れる。2.2 節では TCP/IP のうちリンク層における MAC について示し、2.3 節では WLAN の MAC を策定している IEEE 802.11 及び、ZigBee の MAC を策定している IEEE 802.15.4 (-2003) を説明する。2.4 節では本研究に深く関連する、IEEE 802.11 及び IEEE 802.15.4 で規定されている具体的なアクセス制御方式について紹介する。

2.1 TCP/IP

TCP/IP とは、インターネットなどで標準的に用いられる通信プロトコルであり、TCP (Transmission Control Protocol) と IP (Internet Protocol) を含むインターネット標準のプ

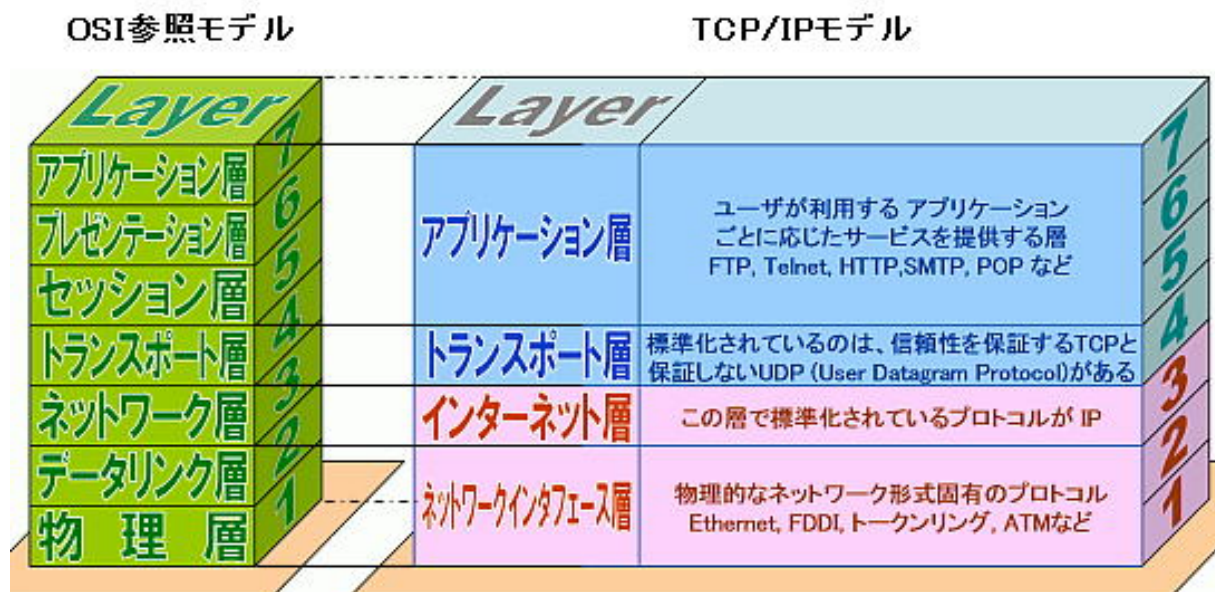


図 2.1: TCP/IP 参照モデル及び OSI 参照モデル

ロトコル群全体の総称である。IP は複数のネットワークを繋ぎあわせて相互に通信可能にするプロトコルである。現代社会において欠かすことのできないインターネットも、IP を用いて世界中の様々な組織の管理するネットワークを相互接続してできたものである。TCP は IP を基盤にその上層で利用されるプロトコルで IP ネットワーク上の 2 地点間で信頼性の高い通信を可能にする。

2.2 TCP/IP 参照モデル

TCP/IP プロトコル群は階層の一式として見る事ができる。図 2.1 に TCP/IP 参照モデル、OSI 参照モデル及びそれに関連したプロトコルなどを示す。TCP/IP 参照モデルは 4 つの階層で構成される。

実際に TCP/IP プロトコル群を利用して通信を行う時、各階層ではデータにヘッダを付加する。データは付加されたヘッダの情報をもとに、送信先や送信先で引き渡すアプリ

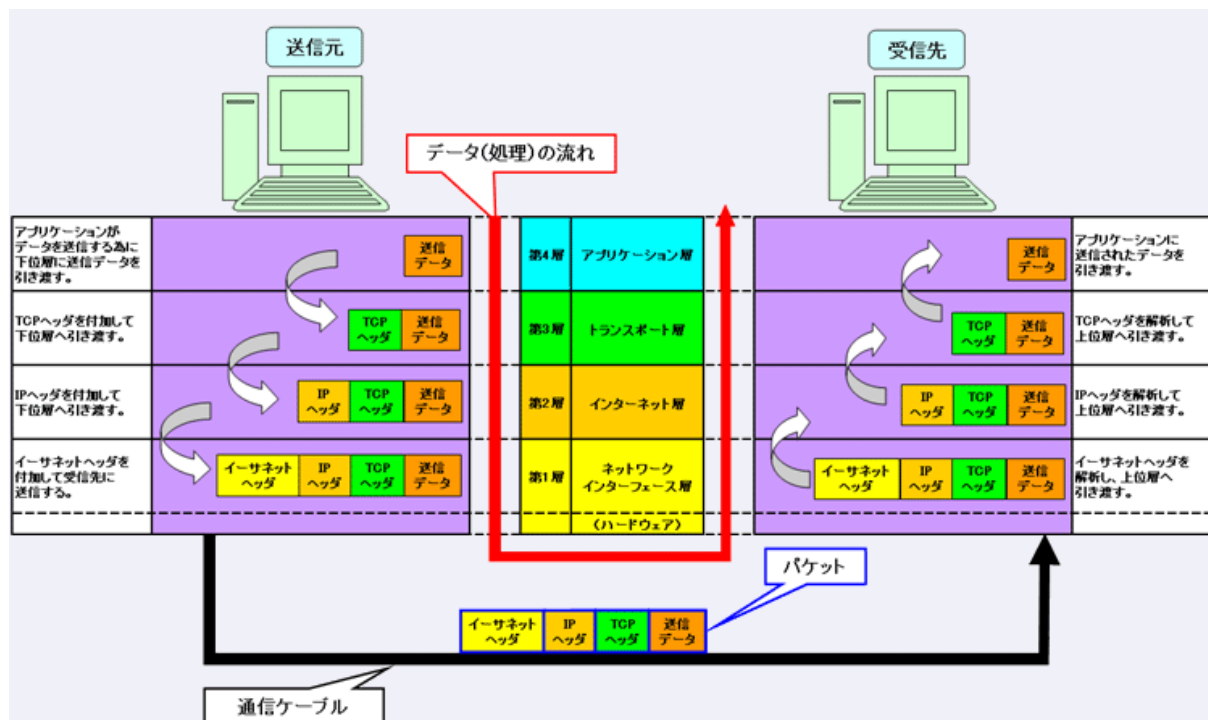


図 2.2: TCP/IP 参照モデルによる通信の説明

ケーションなどの判断を行う。ヘッダとは～である。具体的な通信の流れとして、図 2.2 に TCP/IP 参照モデルを用いた通信例の説明を示す。データの流れとしては各層へ逐次的に通信を行い、ヘッダを付加する。送信元ではアプリケーション層からトランスポート層、インターネット層を通してネットワークインターフェース層まで通信を行う。ネットワークインターフェース層ではハードウェアである通信ケーブルへデータパケットを送信し、受信先では送信元とは逆順でネットワークインターフェース層からインターネット層、トランスポート層を通りアプリケーション層へと至る。以下で TCP/IP 参照モデルにおける各層の役割について詳細を示す。

データリンク層はネットワークコミュニケーション層とも呼ばれ、データリンクプロトコルを利用して通信をするためのインターフェースとなる階層である。具体的なイメージとしては、PC やスマートフォン等のハードウェアと OS 間でデータの橋渡しをするような

役割を持つ。データリンクプロトコルの例として、イーサネット、IEEE 802.11、WLAN 等があり、本研究とも関連が深い。

ネットワーク層は、IP プロトコルを使用して付加する IP アドレスをもとにしてデータパケットの転送を行う階層である。IP にはデータリンクの特性を隠す役割もあり、通信したいホスト間の経路がどのようなデータリンクプロトコルを用いていたとしても通信が可能である。故にインターネットを利用して通信を行うことが可能であり、インターネットに接続する機器では必ず IP の機能が実装されている。

トランスポート層はアプリケーションプログラム間の通信を実現する階層である。PC の内部では複数のプログラムが同時に動作しているため、どのプログラムとどのプログラムが通信しているかを識別する必要がある。トランスポート層におけるプロトコルでは、アプリケーションプログラムを識別するために、ポート番号と呼ばれる識別子が使われる。TCP/IP 参照モデルにおいては、前述したネットワーク層及びトランスポート層は OS に組み込まれることが想定されている。

アプリケーション層はトランスポート層より上の階層であり、アプリケーションプロセスのための共通アプリケーションサービスへ直接接続して実行したり、OSI 参照モデルの第 6 層であるプレゼンテーション層に対して要求も行うなど、役割の幅が広い。共通アプリケーション層サービスとは関連するアプリケーションプロセス間の意味的な変換を提供するものであり、一例としては、仮想ファイル、仮想端末、ジョブの転送及び操作プロトコルなどである。アプリケーション層に属するプロトコルの例としては、ブラウザ - サーバー間の通信で使われる HTTP (HyperText Transfer Protocol) や暗号や認証の技術を利用して、安全にリモートコンピュータと通信するための SSH (Secure Shell) などがある。

各層はデータ転送に伴い生じる一連の問題を解決し、下位層プロトコルのサービスを使用する上位層プロトコルに明確なサービスを提供する。上位層は利用者と論理的に近く、より理論的なデータを処理する。また最終的に物理的に転送できる形式へデータを変換するため、下位層プロトコルに依存する。本研究で取り扱うのは最下層であるデータリンク

層である。次節で、WLAN 及び ZigBee の準拠する IEEE 802 における通信プロトコルである、データリンク層に属する MAC (Media Access Control) について説明する。

2.3 Media Access Control

Media Access Control は日本語で媒体アクセス制御またはメディアアクセス制御といい、IEEE 802 における TCP/IP モデルの第 1 層にあたるデータリンク層の一部を成す通信プロトコルである。MAC は、データリンクに接続しているノードを識別するために利用するアドレスを指定する。これを MAC アドレスといい、各ネットワークカードに割り当てられた固有の番号であり、サブネットワーク（イーサネットなどのルーターを経由しない物理ネットワーク）内の通信相手にパケットを送信可能とする。この様に、MAC 層は論理リンク制御とネットワークの物理層の間のインタフェースとして動作し、物理アドレスあるいは MAC アドレスによるアドレッシング機構を提供する。

また、MAC は複数の無線ネットワーク端末が電波帯域を共有して情報を送る多元接続を制御する機構を提供する。従って複数のターミナルやネットワークノードをネットワーク上で通信可能とし、一般に Local Area Network (LAN) や Metropolitan Area Network (MAN) で使用される。2.3.1 項でこの多元接続の制御方式について説明し、2.3.1 項で WLAN における MAC プロトコルを、

2.3.1 多元接続制御方式

無線ネットワークでは、周波数帯域は限られたリソースであるため多元接続に対する制御方式が必要になる。多元接続制御方式として代表的なのは、図 2.3 に示す周波数分割多重方式 (Frequency Division Multiple Access, FDMA)、時分割多重方式 (Time Division Multiple Access, TDMA)、時分割多重方式 (Code Division Multiple Access, CDMA)、

の 3 つである。以下にその詳細を示す。

FDMA は、電波の周波数帯を複数の帯域に分割することで多元接続を行う無線通信技術である。各無線局にはそれぞれ異なる搬送周波数が割り当てられる。要求割当多元接続 (DAMA : Demand Assigned Multiple Access) の場合、接続確立時と終了時になんらかの制御機構を必要とする。その結果、任意の部分帯域をどの地球局にでも割り当て可能となる。FDMA には要求割当の他に固定割当 (PAMA : Pre-Assigned Multiple Access) もある。要求割当は、統計的割当処理を使って一時的に搬送周波数を割り当てることで全てのユーザーに対してトランスポンダの帯域への連続なアクセスを可能とする。

TDMA は、同一周波数の電波の固定タイムスロットをそれぞれの無線局に割り当て多元接続を行う無線通信技術である。特徴としては、時間ダイバシティが使用できるためフェージング耐性があり、回線交換接続が可能で遅延時間や通信速度が保証できる。但し、無線局間の同期が必要不可欠である。TDMA における周波数利用効率・回路規模は、FDMA と CDMA の中間程度である。時間軸圧縮を行っているため音声の場合遅延時間があるが、通信衛星を利用する場合には FDMA と比較して地球局から受信する電波の電力の差に対応しやすい。

CDMA は、複数の発信者の信号にそれぞれ異なる符号を乗算し、すべての信号を合成して 1 つの周波数を使って送信し多元接続を行う無線通信技術である。主に直接拡散符号分割多重接続 (DS/CDMA, DS : Direct Sequence), 周波数ホッピング (FH : Frequency-Hopping) の方式に分けられる。

CDMA では、Cell Breathing という CDMA 基地局の通信範囲（セル）の大きさが呼吸しているように収縮変化する特有の現象が存在する。Cell Breathing は、FDMA, TDMA では見られない。CDMA の場合、セル内の端末は同一周波数で通信するため常に干渉（お互い混信している）状態にあるが、セル内の端末数が増えて容量の限界に近づいてくるとエリア端に近い局は電波伝播損失が大きいため、基地局に近い局よりも干渉量が大きくなり基地局との通信が出来なくなる。この現象をマクロで見ると、あたかもセルの大きさが

収縮しているように見えるため Cell Breathing という。

2.3.2 WLAN における MAC プロトコル

WLAN の規格である IEEE 802.11b や IEEE 802.11g ではアクセス制御方式としての通信プロトコルがいくつかあるため、本稿ではそれらについて説明する。IEEE 802.11 は、米国電気通信学会 IEEE (The Institute of Electrical and Electronics Engineers, Inc.) により策定された広く普及している WLAN 関連規格の 1 つであり、1997 年に初めて規格統一された WLAN 規格である IEEE 802.11 をはじめとして現在では??に示す 8 つの規格が存在する。このうち、IEEE 802.11a、IEEE 802.11b 及び IEEE 802.11g において利用されているのが CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance, 搬送波感知多重アクセス/衝突回避方式) である。そのシーケンスを図 2.4 に示す。CSMA/CA では、まず通信を開始する前に一度受信を試みることで現在通信をしているホストが他にあるかどうか確認する、搬送波感知 (Carrier Sense) を行う。WLAN 環境中は複数の WLAN 端末が同じ回線を共用しているため、多重アクセス (Multiple Access) 状態である。搬送波感知の結果、他者が通信をしていなければ自分の通信を開始する。ここで搬送波感知の段階で通信中の WLAN 端末が存在した場合、通信終了と同時に送信を試みると衝突する可能性が高い。従って、他の WLAN 端末の送信終了を検知した場合は自分が送信を開始する前にランダムな長さの待ち時間をとる。なお、永久に送信できない事態を防ぐため、この待ち時間は徐々に短くされてゆく。これが衝突回避 (Collision Avoidance) 部分にあたる。

しかし、CSMA/CA では図 2.5 に示す隠れ端末問題を解決できない。隠れ端末問題とは、無線通信において互いに相手の通信を検知できない関係にある複数の機器が同じ機器に向けて同時に信号を送信してしまい、信号の衝突 (コリジョン) が発生してしまう問題である。

IEEE 802.11 においては、RTS (Request to Send) /CTS (Clear to Send) 方式を採用しており、これを用いて隠れ端末問題による干渉の影響を軽減している。RTS/CTS 方式のシーケンスを図 2.6 に示す。WLAN 環境における RTS/CTS 方式では送信電力の大きく通信範囲の広い AP を利用するため周囲の多くの WLAN 端末に CTS フレームが届き、CSMA/CA 方式より確実なアクセス制御を行うことが可能であるが、送信前に必ず待ち時間が入るため、オーバーヘッドが生じる。

2.3.3 ZigBee おける MAC プロトコル

ZigBee の規格である IEEE 802.15.4 では IEEE 802.11 同様、アクセス制御方式としての通信プロトコルは CSMA/CA を利用している。図 2.7 にその動作の詳細を示す。

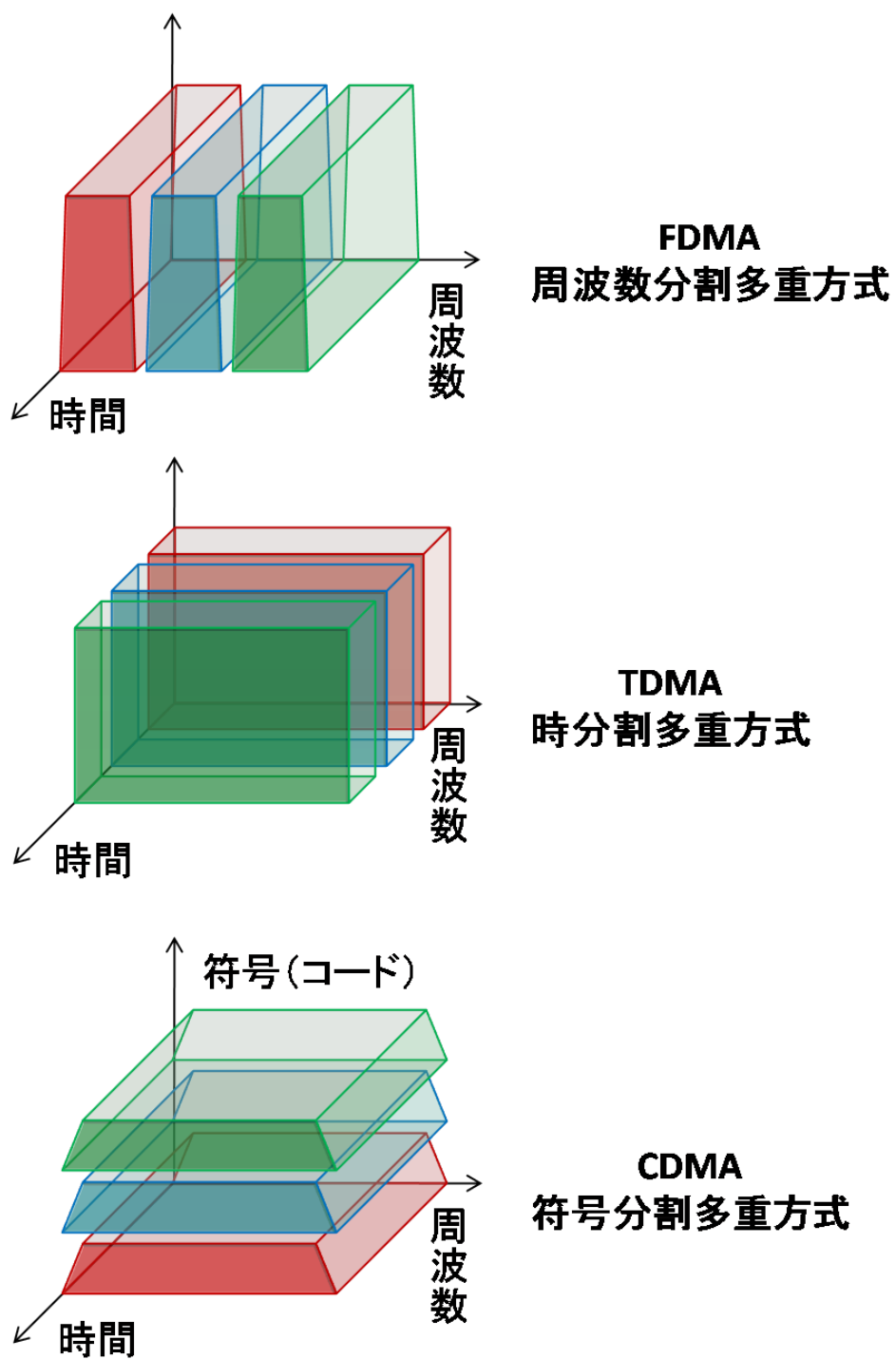


図 2.3: 多元接続制御方式

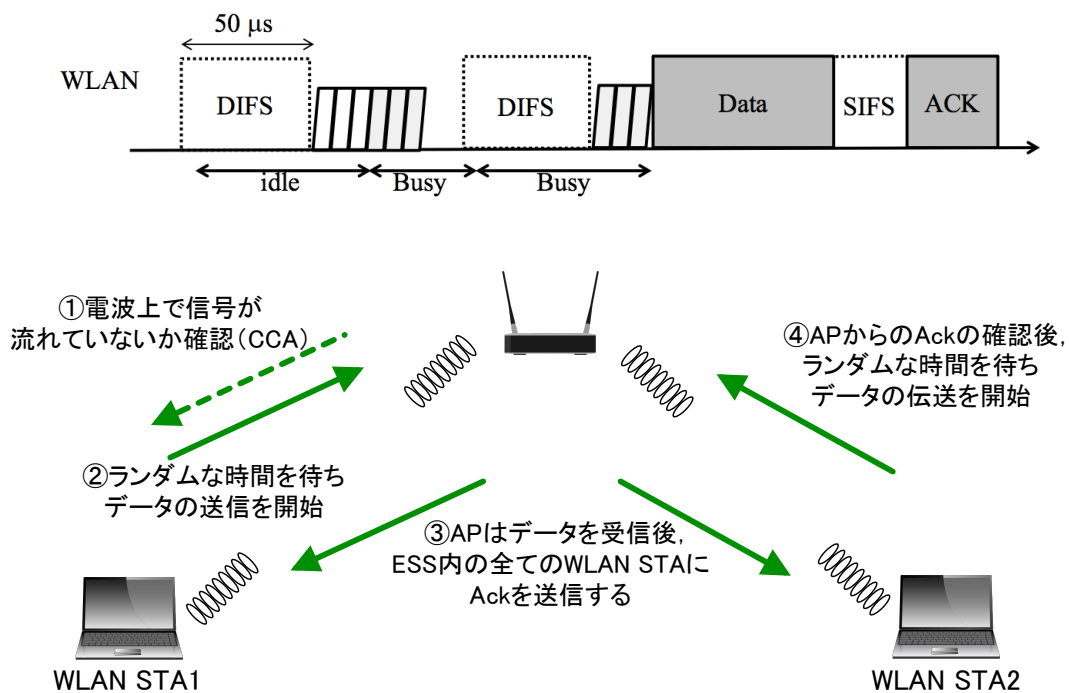


図 2.4: WLAN 環境における CSMA/CA 方式

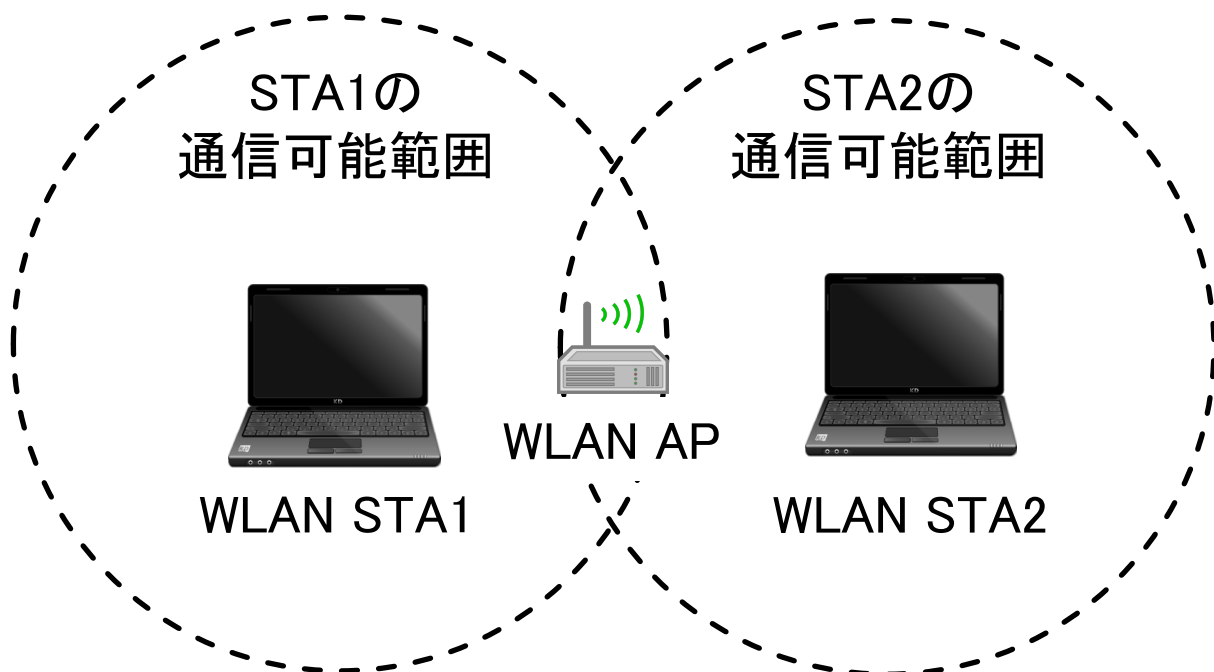


図 2.5: 隠れ端末問題

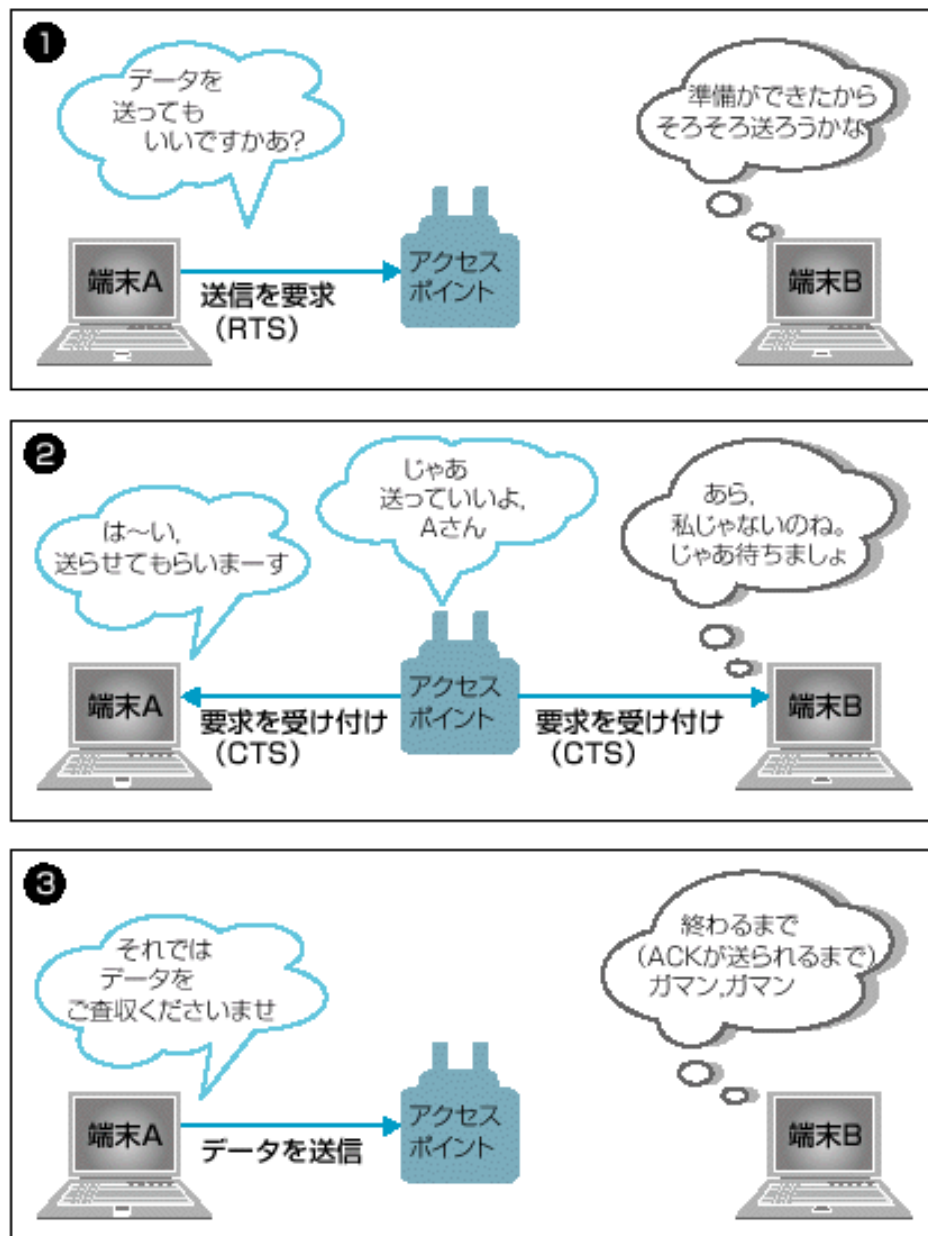


図 2.6: WLAN 環境における RTS/CTS 方式

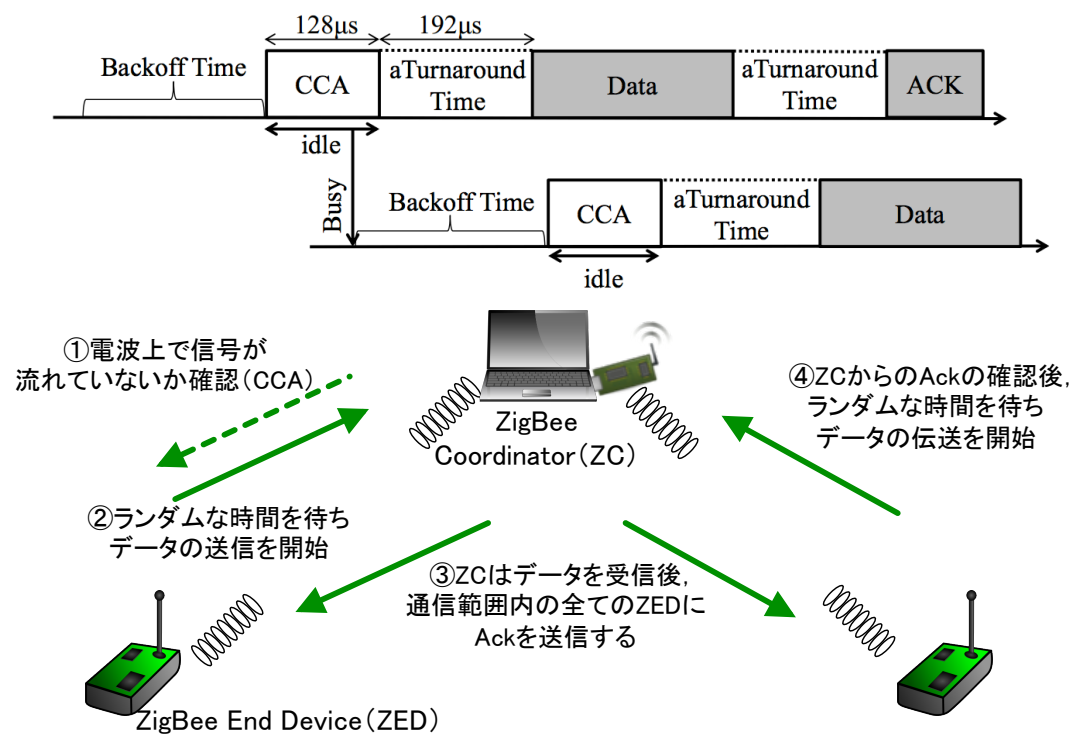


図 2.7: ZigBee 環境における CSMA/CA 方式

第3章

関連研究及びその比較

干渉回避方式に関する研究は膨大に存在するが，大別すると

1. WLAN/ZigBee の既存のアクセス制御方式を利用した方式 [2-6]
2. 既存のアクセス制御方式以外の技術を利用した方式 [7,8]

に分類される．

WLAN の CSMA/CA 機構を利用した干渉回避方式として CBT [3] が挙げられる．CBT では，signaler と呼ばれる ZigBee ノードを WLAN 機器の近くに配置し，ZigBee 通信時に signaler から長いフレームを送信することで WLAN 端末に ZigBee が通信状態にあることを通知する．WLAN 端末では signaler の送信信号により CSMA/CA 機構が働き，送信が抑制されるため WLAN と ZigBee の衝突を回避できる．しかしながら，スマートフォンなどの移動体も含めて考えると全ての WLAN 機器の近くに signaler を配置することは非現実的である．

また，村田ら [4] は干渉の原因を ZigBee の aTurnaroundTime（送受反転時間）が通信アイドル時の待機時間である DIFS（DCF Inter Frame Space）よりも長いことと捉え，

DIFS を拡大する方式を提案している．DIFS を aTurnaroundTime に比べ長くすることで衝突を回避する．しかしながら，DIFS を拡大するためには予め AP にアクセスして設定しなければならず，実現のためには周囲の AP 全てを管理下に置かなければならない．

ZigBee が準拠している IEEE 802.15.4 のビーコンモードによるアクセス制御を利用した干渉回避方式も提案されている [5]．Dynamic GTS では，ビーコンモードの 1 つである ZigBee ノードに Guaranteed Time Slot (GTS) を割当て優先的に通信できる期間 CFP (Contention Free Period) を利用し，この CFP を拡大することで衝突を回避する．CFP の拡大は，GTS を動的に配置しそのビーコンを検知した WLAN 端末が CTS フレームを送信することで達成可能である．しかしながら，??で示した CTS-Blocking と同様にハードウェアや OS の改変が必須である点，隠れ端末問題の影響を受けやすい点が問題である．CACCA [6] においても同様の問題が考えられる．

既存のアクセス制御方式以外の技術を利用した方式として，無線通信の通信空き時間を有効活用して通信を行うための「White Space」技術を応用した干渉回避方式が提案されている [7]．WISE では，WLAN 通信における White Space を予測し，その White Space 内で通信が完了するように ZigBee フレーム長の動的制御を行う．例えば，WLAN 通信の混雑時には予測される White Space の長さは短くなるため，ZigBee ノードにおいてデータを分割して送信する．これにより，ZigBee フレームが WLAN 通信によって破壊される確率を低減させることが可能となる．しかしながら，WLAN 通信が混雑している時には WLAN 通信の空き時間が少ないために White Space の予測が外れ，通信が衝突する確率が高くなる．

フレーム衝突について分析し，分析結果を干渉回避方式とした研究もある．BuzzBuzz [8] では，WLAN と ZigBee のフレーム衝突により通信が阻害される原因として IEEE 802.15.4 フレームのヘッダが破壊されることと捉え，ヘッダを 2 重にして送信することで ZigBee のパケットロス率を減少させている．BuzzBuzz はこれまでに紹介した関連研究群とは違い，既存の方式やモジュールの改造及び特別なハードウェア等の利用が必要なく，干渉回避方

式の構築コストについて考慮されている。これは、本稿で提案する AA CTS-Blocking 方式とは違うアプローチで ZigBee の通信を保証しており、AA CTS-Blocking と組み合わせることでさらなる干渉回避効果が期待される。

第 4 章

AP-Assisted CTS-Blocking 方式の提案

WLAN AP は，WLAN 端末に比べ送信電力が大きく，RTS/CTS 制御方式においても利用されるため，これを利用して干渉回避方式である AP-Assisted (AA) CTS-Blocking を提案する．

4.1 概要

図 4.1 に，AA CTS-Blocking 方式の概要を示す．本システムは，環境内に配置された複数の ZigBee End Device (ZED) 及び ZigBee ネットワーク制御を行う ZigBee Coordinator (ZC)，制御 PC から構成される．ZigBee 基地局と制御 PC は有線接続されている．AA-CTS Blocking では，周辺にある WLAN AP から CTS フレームを送信させることで WLAN 通信をブロックし干渉の影響を低減した ZigBee ノード間の通信を実現する．

図 4.2 に AA CTS-blocking 方式の通信シーケンスを示す．制御 PC は，あらかじめ周囲に存在する WLAN AP のビーコンフレームを受信し，チャンネル，受信信号強度 (RSSI) 等を収集しておく．ZigBee ノード間の通信を行う場合，まず (1) 制御 PC は AP を 1 つ選択し，(2) 制御 PC は選択した AP に向けて RTS フレームを送信する．次に (3) RTS フレームを

受信した AP は周囲の WLAN 端末に向け CTS フレームをブロードキャストする。CTS フレームを受信した WLAN 端末は CTS フレーム内の **Duration** フィールドに記載された時間だけ送信を控えるため、WLAN 通信が一時的にブロックされる。WLAN 通信 Block 期間 (Duration of Block WLAN Communication, DBW) がスタートした時、(4) CTS フレームを受信した制御 PC は、有線接続された ZC に向けて通信開始信号 (Communication Start Signal, CSS) を送信することで ZigBee ノード間の通信を開始する準備が整う。ZigBee ノード間の通信は ZC を用いた制御により DBW 内に完了させる。

このような AA CTS-Blocking を実現する上では 2 つの疑問が生じる。

1. RTS 送信先 AP をどのように選ぶか

実環境には多くの WLAN AP が存在するため、制御 PC は RTS の送信先 AP を選択する必要がある。4.2 において、周囲に存在する WLAN AP のビーコンフレームを用いた AP 選択アルゴリズムを示す。

2. DBW において、ZigBee ノード間通信をどのようにスケジューリングするか

DBW は有限であるため、ZigBee ノード間で通信の衝突が起きないようにアクセス制御を行う必要がある。??において、1 台の ZC と複数の ZED との通信におけるスケジューリング手法を検討する。

4.2 AP 選択アルゴリズム

AA CTS-Blocking においては、RTS フレームの送信先 AP の選択は干渉回避性能に大きな影響を及ぼす。例えば、制御 PC の通信可能範囲ギリギリの位置に存在する AP を選択した場合 RTS フレームが AP に届かない可能性や AP からの CTS フレームを受信できない可能性がある。RTS フレーム送信先 AP の選択を最適化することにより、より確実に多くの WLAN 通信をブロックできる。

本研究では AP と制御 PC の物理的距離が重要になると考え、位置推定 [9] などにも用いられる RSSI を AP 選択アルゴリズムのキーとして採用した。RSSI の取得は特殊なパラメータや追加のデバイスを必要とせず、既存の AP から発せられる WLAN フレームを受信するだけで収集可能である。図 4.1 に示すように制御 PC が ZC に有線接続されており、ZED は ZC を中心に周囲に配置されていることも考慮に入れると、制御 PC 周辺の WLAN 通信をブロックすることが非常に有効であると考えられる。

4.3 スケジューリング手法

AA CTS-Blocking 方式においては、ZigBee ノード間の通信は AP から送信された CTS フレーム内の Duration フィールド（最大値は 32 ms [10]）に記載された時間内に終了させる必要がある。ZED の台数が増えた場合通信回数が増え、1 回の RTS/CTS 送信での通信が難しくなるため、スケジューリング等のアクセス制御方式が必須となる。

これに向け、本研究では既存の MAC プロトコルを利用する。MAC プロトコルは種類を問わず利用可能である。ZED の台数が 32 ms 間で通信できない台数に増加した場合には、ZED をグループ化し、ZC からの制御によって RTS/CTS の 1 回の送信に対して 1 グループに通信を許可すれば良い。ただし、グループ間での通信に関しては問題が残る。これについては今後の研究課題とする。

一例として、TDMA (Time Division Multiple Access) 方式の MAC プロトコルを利用する場合、??で示した ZigBee ノード間の通信シーケンス部分は以下のようにすれば良い。制御 PC から CSS を受信した ZC は、周囲の ZED ヘデータ要求フレームをブロードキャストする。データ要求フレームを受け取った ZED は、データ要求フレームを Slot 同期信号として Time Slot (TS) を用いたアクセス制御を開始し、ZED 毎にあらかじめ定められた Slot において ZC に対してデータフレームを送信する。

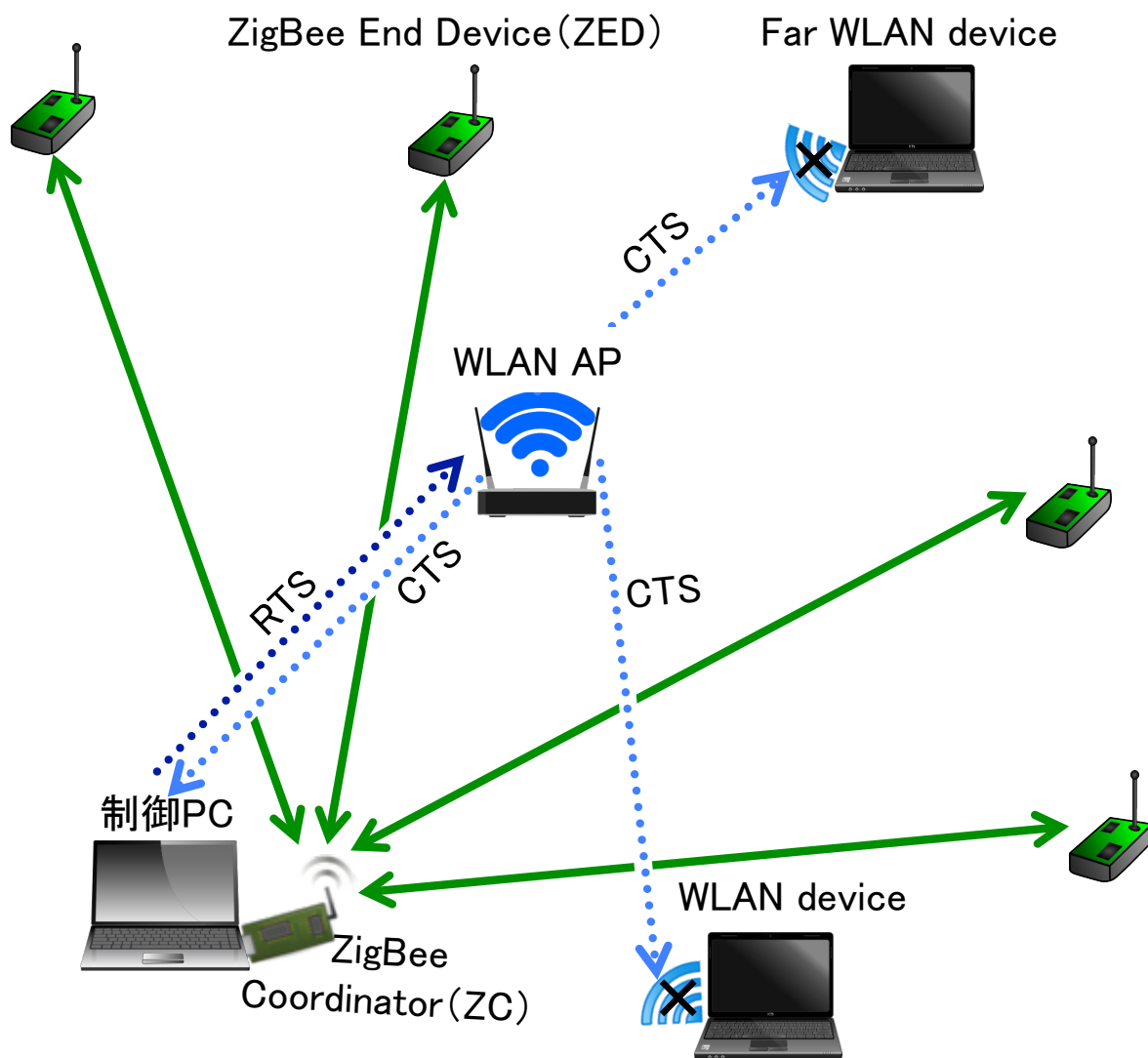


図 4.1: AA CTS-Blocking

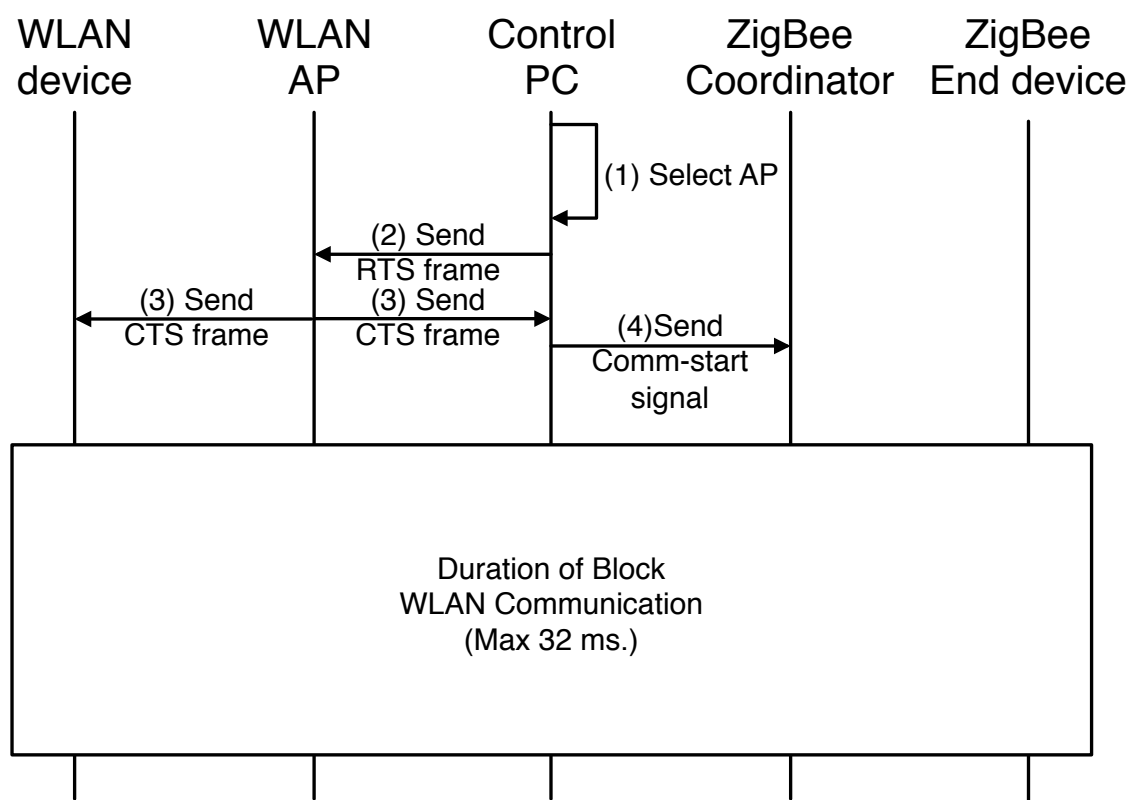


図 4.2: AA CTS-blocking 方式の通信シーケンス

第5章

設計及び実装

本章では，AA CTS-Blocking 方式を適用した ZigBee ノードによるデータ収集システムの設計及び実装について述べる．はじめに，ZigBee ノードによるデータ収集システムを構成するハードウェア及び OS について触れ，UML を用いた設計について説明する．その後，データ収集システムの実装について述べ，制御 PC による AA CTS-Blocking 方式の制御アプリケーション実装を説明する．

5.1 ハードウェア

5.1.1 ZED

データ収集システムにおける ZigBee ノードは日本国内で入手しやすく，センサノードとして一般的な Crossbow 社の MICAz MPR2600J [11] を用いた．図 5.1 にその外観を示す．MICAz は無線機能，CPU，メモリ等を有したノード部分と各種センサを搭載したセンサ基板部分から構成されるセンサノードであり，MPR2600J(RF 周波数帯 2405MHz～2480MHz) は Chipcon CC2420，IEEE 802.15.4 準拠，Atmega128L マイクロコントローラと統合された ZigBee 対応無線周波数トランシーバを使用しており，最大約 50m の通信



図 5.1: MICAz MPR2600J

を行うことができる。安定動作電圧は 3.3V，消費電流は通信時 60mA，スリープ時 $20\ \mu\text{A}$ である。電源は単三乾電池 2 本から供給可能である。端の一方には ON/OFF スイッチが，もう一方にはアンテナが付属している。

5.1.2 ZC 及び制御 PC

データ収集システムにおける ZC は，前述した ZED で使用する MICAz と，MICAz に対応した Crossbow 社の MIB520 [?] を用いた。図 5.2 にその外観を示す。I/O インターフェースとして USB A タイプ（オス）を備え，電源は USB バスを通じて PC から供給する。MIB520 はオンボードで ISP（in-system programming）に対応しており，MICAz を接続したままシリアル通信及びプログラミングが可能である。出力コンソールとして 3 色（赤，緑，黄）の LED を利用できる。MICAz を MIB520 に装着し，オスメス USB A-A



図 5.2: ZC (MICAz MPR2600J + MIB520)

コネクタを用い制御 PC と接続して ZC の機能を果たさせた。また、制御 PC は Debian GNU/Linux の動作する dynabook UX/28LWHEM を用いた。

5.2 ZigBee ノードによるデータ収集システムの設計

データ収集システムの概要を図 5.3 に示す。ZC を中心とするスタートポロジのネットワークを構築し、各 ZED から ZC にデータを収集する。DBW における ZigBee ノード間の通信は TDMA 方式のスケジューリングを行い、各 ZED 毎に定められた Time Slot (TS) を設けた。TS は ZED の Node ID と対応付けており、最大の TS でも DBW を超えないように定めた。

制御 PC から CSS を検出した ZC は周囲の ZED に向けて Broadcast Frame (BF) を送信する。BF を受信した ZED は、BF を Slot 同期信号として ZED 毎に定められた Time

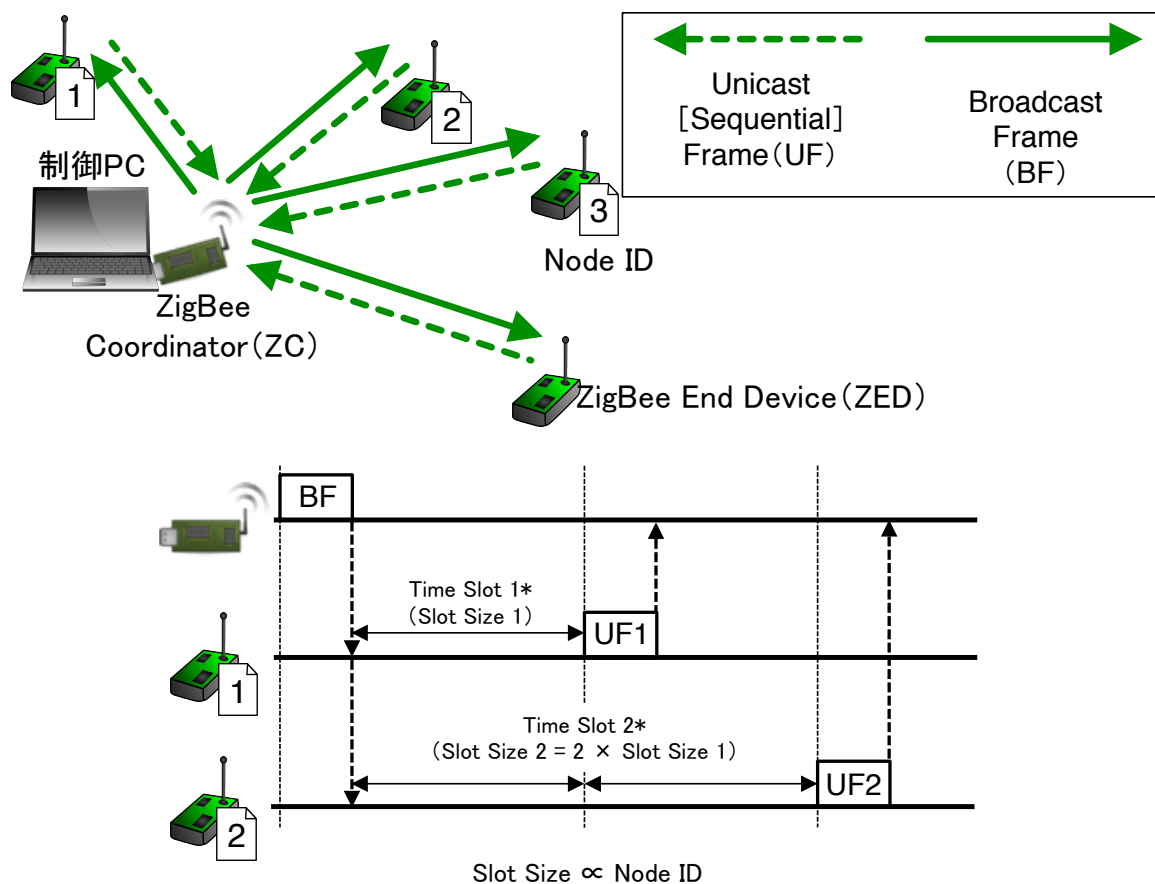


図 5.3: ZigBee ノードによるデータ収集システム

Slot に従い、Slot Size (SS) の時間だけ待機する。SS の時間が経過した後、ZED は ZC の Node ID を送信先アドレスとして Unicast Frame (UF) を送信する。

ZED, ZC に利用した MICAz において、ユーザーが定義可能なアプリケーション層の実装はセンサネットワークノード向けのイベント駆動 OS としてオープンソースで開発されている TinyOS を用いた。TinyOS における実装は nesC によるイベント駆動型プログラミングとなり、C 言語の拡張である。nesC は一般的なプログラミング言語ではないため、5.2.1 項で nesC について説明を行う。

5.2.1 nesC

本節では ZigBee ノードによるデータ収集システム実装のために学習した，nesC の基礎的な文法について説明を行う．nesC においては C 言語の if 文等の文法は同じように利用できるが，プログラミングパラダイムはイベント駆動型プログラミングであるため記述方法は大きく異なる．以下で，nesC アプリケーションの構築に必須である構文を示す．

- `implementation` : 実装を意味し，`implementation` 構文の中に実装部分の中身を記述する．
- `event` : プログラムの実行に際し，データを受信した，起動していたタイマが完了した等，何らかのアクションが発生した際にプログラムに発信される信号をイベントという．`event` 構文の中には，何らかの条件を満たした時に実行したいアクションを記述する．
- `components` : アプリケーションの 1 機能をコンポーネントといい，`components` 構文はその名前を表す．
- `interface` : コンポーネントの接続口をインターフェースといい，`interface` 構文はその名前を表す．
- `module` : `module` 構文の中には，使用する `interface` を全て記述する．

具体的な nesC ファイルのイメージを図 5.4 に示す．

nesC によるアプリケーションは慣例的に，2つのファイルで記述される．`event` 構文を使用して実際の動作を記述する（アプリケーション名）`C.nc` ファイルと，`components` 文を使用して利用するコンポーネントを示し，インターフェース間の接続がわかるように記述された（アプリケーション名）`AppC.nc` ファイルの 2つが存在する．nesC における実装は上記の通りであり，アプリケーションの要であるアクションはイベント駆動である事

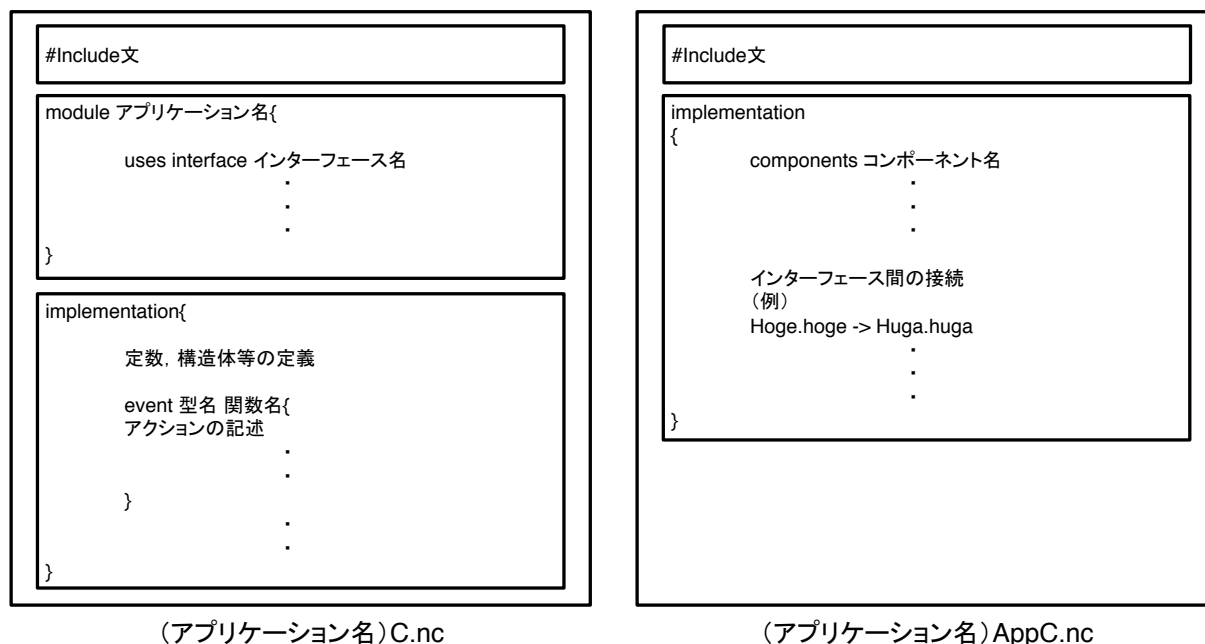


図 5.4: nesC 実行ファイル

がわかるイベント駆動型プログラミングでは、起動すると共にイベントを待機し、起こったイベントに従って処理を行う。イベントを待機している間、MICAzは何らかの状態を持つ。このような状態遷移のフローをを記述するのに最適なのが、UMLのステートマシン図である。UML(Unified Modeling Language)は、抽象化したシステムをグラフィカルな記述でモデル化し、汎用的なプログラム設計図を与える。UMLで表現されるモデルには、システム実装を補助するために多くのダイアグラムが存在する。データ収集システムの設計では、システムを構成するZED, ZCのアクション及び状態遷移を記述するためにステートマシン図を利用し、コンポーネント間の接続を正しく把握するためにコンポーネント図を利用した。以下でZED, ZCそれぞれのステートマシン図及びコンポーネント図について示す。

5.2.2 ステートマシン図

ステートマシン図は無償 UML モデリングツールである astah* community を利用して作成した。ステートマシン図の遷移は矢印で表されており、説明はイベント [ガード条件] / アクションで記述される。ガード条件は直前に発生したイベントの評価をするための条件である。評価値は真もしくは偽の値を持ち、ガード条件に対して真であるときのみ遷移が許される。アクションは遷移が起こると同時に実行される動作である。以下で、各機器毎に詳細を説明する。

ZED は ZC から BF を受信するイベントが発生した時、ZC へ UF を送信するアクションをとる。状態を S_n (n は添字)、イベントを E_n 、ガード条件を G_n 、アクションを A_n と表記し ZED の状態遷移を考慮すると、以下の通りになる。

- S_1 : BF 受信待機状態
- E_1 : ZC から BF を受信する
- G_1 : BF の受信に成功する
- A_1 : SS の時間分待機するための TS_n (n はノード番号) タイマを起動する
- S_2 : UF 送信準備状態
- E_2 : TS_n タイマが完了する
- A_2 : ZC に向けて UF_n を送信する
- S_3 : UF 送信完了状態
- A_3 : TS_n タイマをリセットする

AA CTS-Blocking 方式により ZigBee 通信を保証する時間は、4.1 で示した DBW に等しい。この時間はわずかであり、送信失敗した場合にタイムアウトによる再送の仕組みを

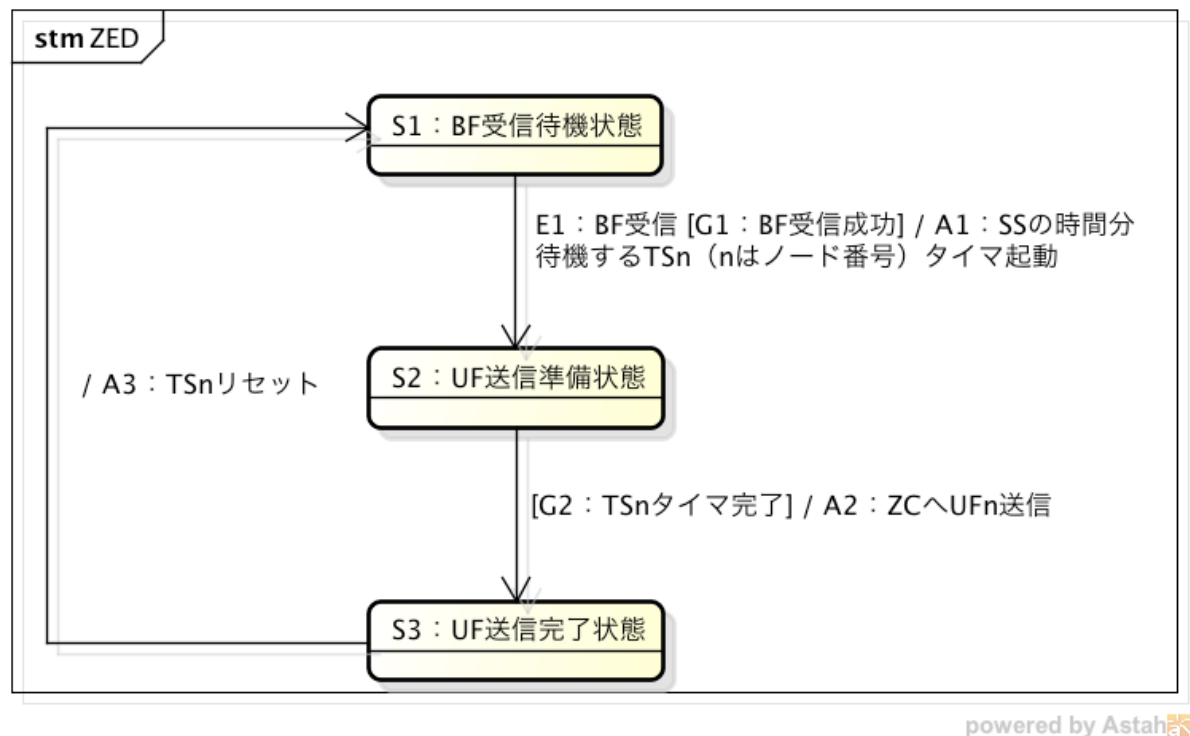


図 5.5: ZED のステートマシン図

設ける余裕が無い。従って、送信成功かどうかの判定は行わない。これらを総合すると、ZigBee ノードのステートマシン図は図 5.5 の様に描画できる。

次に、ZC のステートマシン図を考える。ZC は 1. 制御 PC から CSS を受信するイベントが発生した時、DBW の測定を開始し、ZED へ BF を送信するアクションをとり、2. ZED から UF を受信するイベントが発生した時、制御 PC へ UF を送信するアクションをとる。ZED の場合と同様に、状態を S_n 、イベントを E_n 、ガード条件を G_n 、アクションを A_n と表記し ZC の状態遷移を考慮すると、以下の通りになる。

- S1 : CSS 受信待機状態
- E1 : 制御 PC から CSS を受信する

- G1 : CSS の受信に成功する
- A1 : DF 記載時間待機する DBW タイマを起動する
- S2 : BF 送信準備状態
- A2 : ZED へ BF を送信する
- S3 : UF 受信待機状態
- E3 : ZED_n (n はノード番号) から UF_n を受信する
- G3 : UF_n の受信に成功する
- A3 : 制御 PC へ UF_n を送信する (制御 PC でのデータ取得のために)
- E3' : DBW タイマが完了する
- A3' : DBW タイマをリセットする

ZED と同様送信失敗した場合にタイムアウトによる再送の仕組みを設ける余裕が無い
ため、S3 は S2 から A2 の動作を伴い自動遷移するようにした。そして、A1 で起動させた
タイマが終了した際は、S3 から S1 に遷移する事とした。これらを総合すると、ZigBee 基
地局のステートマシン図は図 5.6 の様に描画できる。

5.2.3 コンポーネント図

コンポーネント図もステートマシン図と同様、無償 UML モデリングツールである astah*
community を利用して作成した。コンポーネントの接続は提供インターフェース及び要求
インターフェースで表される。提供インターフェース側のコンポーネントで実際のアプリ
ケーション機能が実現されており、要求インターフェース側のコンポーネントでそれを継

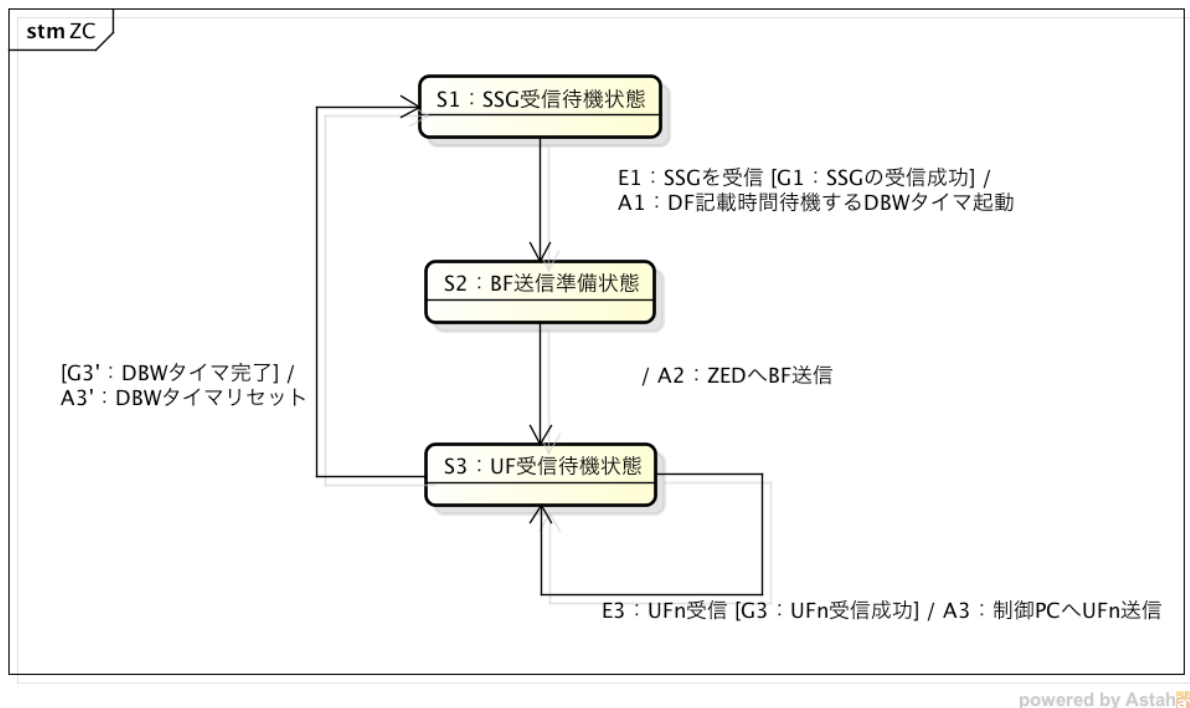


図 5.6: ZC のステートマシン図

承する．astah では提供インターフェースは丸に線が繋がった形で，要求インターフェースは丸を受け取る皿のような形で表現される．

nesC によるアプリケーションにおいては，慣例的に app と記述される．ユーザーが作成した app コンポーネントはすべて他の既存コンポーネントの機能を継承するため，コンポーネント間の接続はすべて要求インターフェースになる．また，5.2.2 項で示したステートマシン図より，ZED 及び ZC に必要な機能がわかる．

これを踏まえ，図 5.5 に示す ZED のステートマシン図を参照すると，ZED で必要な機能とそれに対応した既存コンポーネントは以下の通りである．

- TinyOS の起動：MainC
- BF フレームの受信：AMReceiverC

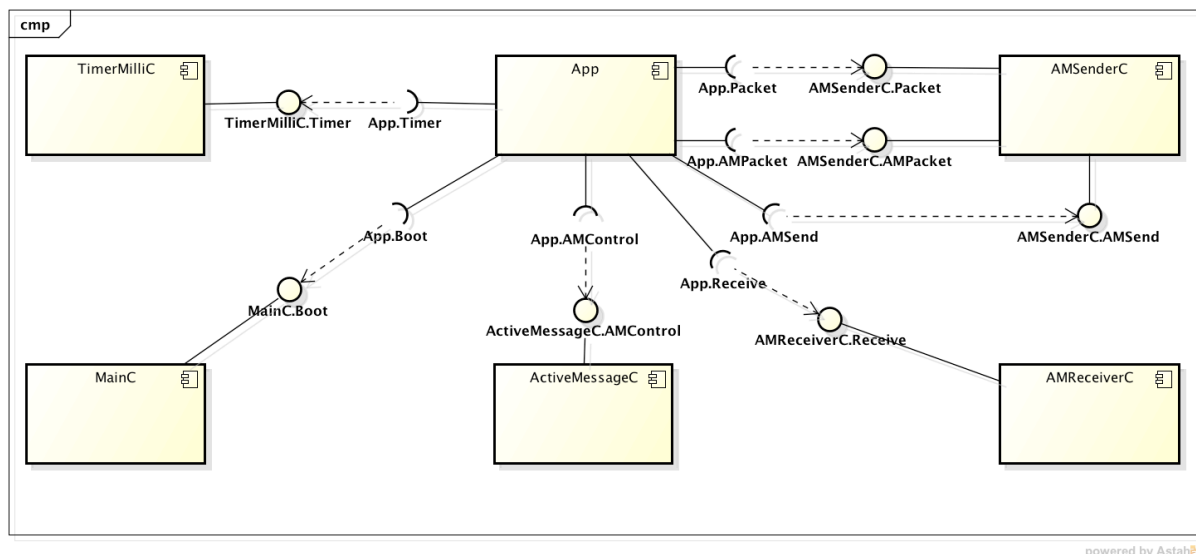


図 5.7: ZED のコンポーネント図

- TSn タイマの起動：TimerMilliC
- UF フレームのユーザー定義：ActiveMessageC
- フレームヘッダ等の生成：AMsenderC
- UF フレームの生成：AMsenderC
- UF フレームの送信：AMsenderC

TinyOS では Active Message (AM) layer を提供しており、これによって ZigBee ノードが送受信するデータについてユーザーが自由に定義することが可能である??。従って、ZED のコンポーネント図は図 5.7 の様に描画できる。

同様に、ZC のコンポーネント図を考える。図 5.6 に示す ZC のステートマシン図を参照すると、ZC で必要な機能とそれに対応した既存コンポーネントは以下の通りである。

- TinyOS の起動：MainC

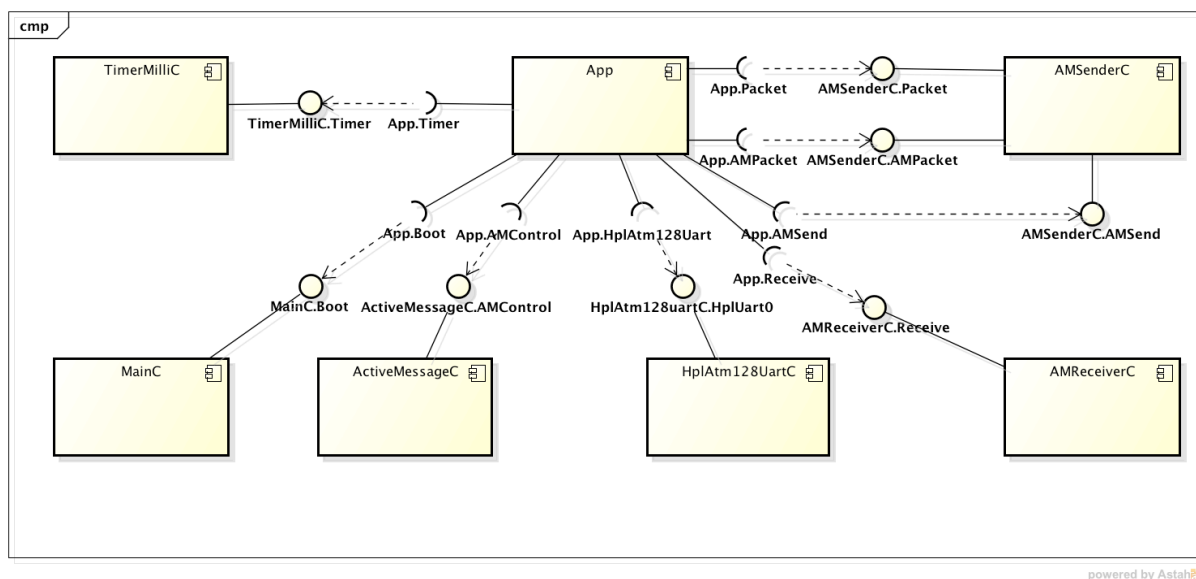


図 5.8: ZC のコンポーネント図

- シリアル通信による CSS 検出 : HplAtm128UartC
- DBW タイマの起動 : TimerMilliC
- BF フレームのユーザー定義 : ActiveMessageC
- フレームヘッダ等の生成 : AMsenderC
- BF フレームの生成 : AMsenderC
- BF フレームの送信 : AMsenderC
- UF フレームの受信 : AMReceiverC

従って，ZC のコンポーネント図は図 5.8 の様に描画できる。

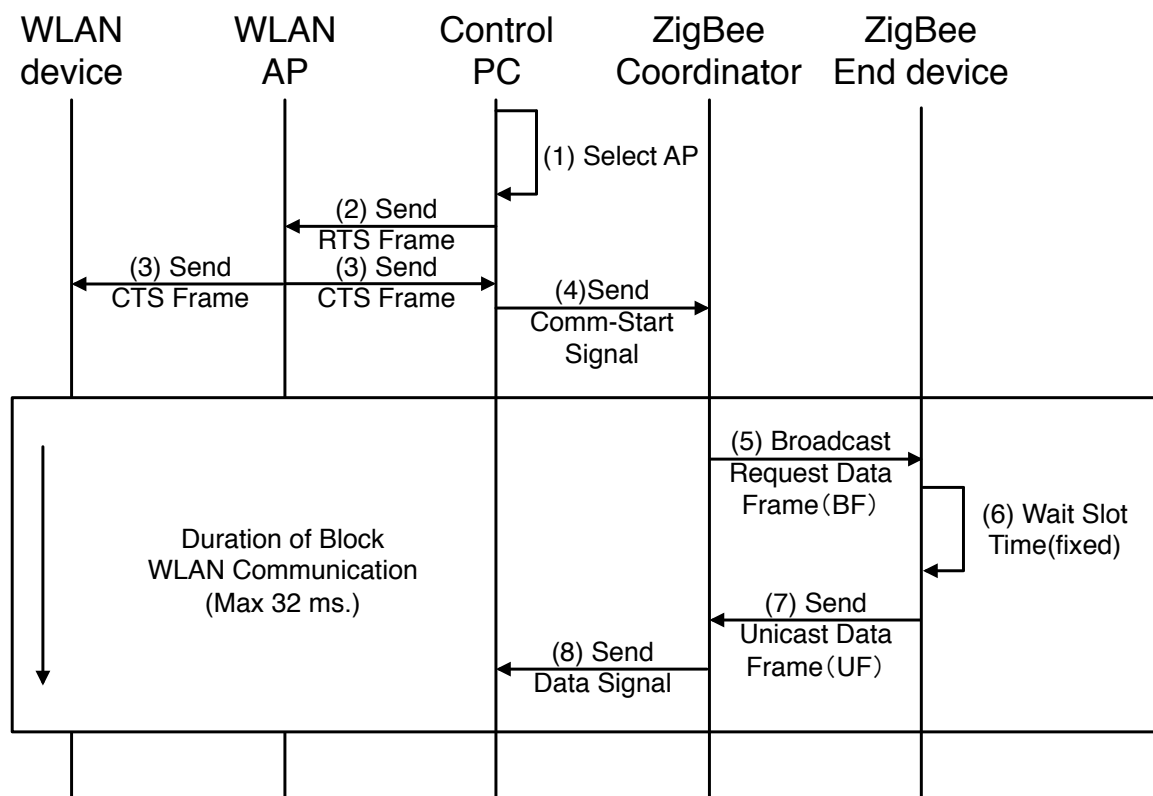


図 5.9: ZC のコンポーネント図

5.3 AA CTS-Blocking 方式の実装

AA CTS-Blocking 方式の動作の実証と基本性能の評価に向け、5.2 節で作成したステートマシン図、コンポーネント図及びシーケンス図を参考に、PC 及び ZigBee ノード用いて、AA CTS-Blocking を適用した ZigBee ノードによるデータ収集システムを実装した。

図 5.9 に AA CTS-Blocking 方式を適用したデータ収集システムのシーケンス図を示す。シーケンス図は、オブジェクト間の相互作用を時系列に沿って表現する UML ダイアグラムである。シーケンス図での時間は、ライフラインに沿って上から下に進む。ここでは、システム全体の処理の流れが確認できる。

AA CTS-blocking 方式を実現する制御アプリケーションは C 言語で実装した。モニターモードの無線 LAN インタフェースを用い、libpcap ライブラリを利用することで環境中に

流れるフレームを傍受し，RTS/CTS フレームのキャプチャを行う．また，RTS フレームの送信にも libpcap ライブラリを利用した．

RSSI による AP 選択アルゴリズムにおいても libpcap ライブラリを利用した．制御アプリケーションの開始時に Radiotap ヘッダが付加された IEEE 802.11 ビーコンフレームを 1.5 秒間収集した．収集したフレームを解析し，RSSI がもっとも大きい AP を RTS 送信先として選択した．

第 6 章

評価

提案する AA CTS-Blocking 方式の有効性を検証するために評価実験を行った。まず、予備実験を行ってデータ収集システムを動作させるパラメータである Slot Size を決定した。次に、AA CTS-Blocking 方式が WLAN の混雑度に関わらず高い干渉回避効果を有することを示すために、ZigBee 通信の成功率の評価を行った。

6.1 予備実験

??で述べた Time Slot の Slot Size を決定するために、データ収集システムの予備実験を行った。適切な Slot Size 決定に向けては、ZigBee ノード間の通信を保証した上で通信が失敗しないことを確認する必要があるため、予備実験では電波暗箱内でデータ収集システムの動作を評価した。図 6.1 に、電波暗箱内に配置したデータ収集システムを示す ZigBee ノード間で送受信するデータは、サイズの変化による影響をなくすため送信元アドレス及びシーケンス番号のみが記録されたダミーデータとした。サイズはヘッダを含めて 18 バイトと一定である。18 バイトのデータ送信に要する時間は、ZigBee の公称通信速度 250 kbps [12] から計算すると 0.576 ms である。

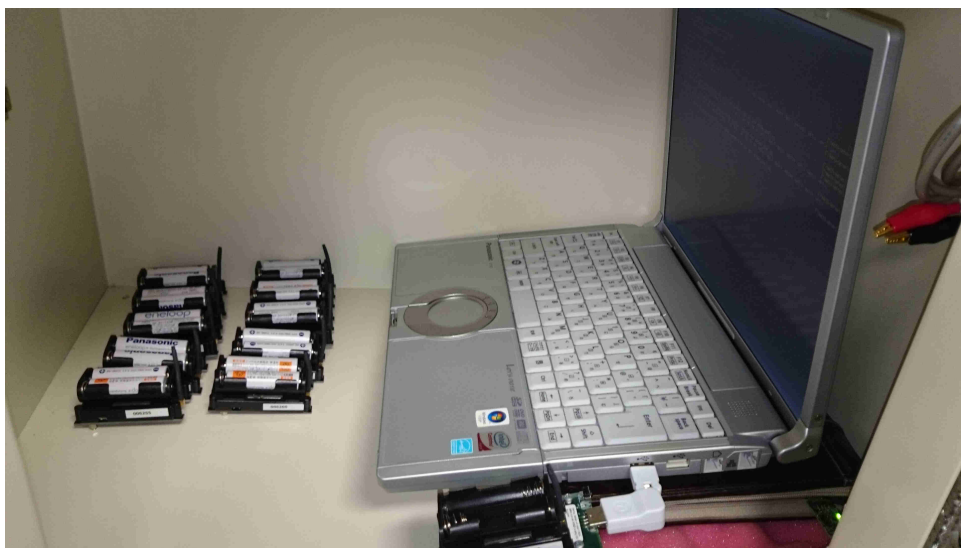


図 6.1: 電波暗箱内の様子

本システムにおける Slot Size は、データの送信時間に加えて MICAz が送受信の切り替えに要する時間及びガード時間を含めたものを想定している。ZC は全 10 台の ZED からダミーデータを収集する。予備実験ではデータ収集の試行を 200 回行った。??で述べた通り、ZigBee ノード間の通信時間が最大 32 ms であり、データの送信時間が 0.576 ms であることを考慮に入れ、Slot Size を 1 ms～3 ms で変化させながらデータ収集予備実験を行った。このとき Packet sniffer を用いてエラーパケットの有無を監視し、通信成功台数を測定した。通信成功台数とは、1 回のデータ収集試行で ZED が ZC へ送信成功した台数であり最大値は 10 台である。200 回試行中の ZED の延べ通信成功台数を 200 回試行中 ZED が全て通信成功した台数である 2000 で割り通信成功率に変換した。

結果、Slot Size 1 ms では 49.8%であったのが 2 ms では 99.95%、3 ms では 99.93%と 100%に限りなく近い値となった。Slot Size は短いほど DBW に対し余裕が持てるため、本評価実験では Slot Size=2 ms と決定した。

6.2 評価環境

??で示した AA CTS-Blocking の有効性を検証する評価として、WLAN 通信環境下において??で実装したデータ収集システムを動作させる通信実験を行った。実験環境として、WLAN ネットワークが多く存在する筆者の研究室を選択した。図 6.2 に実験環境の概略を示す。研究室内に 10 台のノードを分散して配置する。WLAN ネットワーク側には 5 台の WLAN 端末を用いて通信させ、約 5 Mbps の通信負荷を常時発生させた。通信負荷値の測定にはパケット解析ツールである Wireshark を用いた。評価実験の際には、WLAN の平均トラフィックが大幅に変化しないことを確認した。

ZigBee 通信時間は Duration フィールドの最大値 32 ms を参考に、30 ms を確保した。本実験では、??で実装したデータ収集システムの通信シーケンスを利用して全 10 台のノードからダミーデータを収集する動作を 1 サイクルと定義する。1 サイクルの周期は 200 ms とし、データ収集を 1000 サイクル実施した。

比較対象として、(A) Normal: 何もせずに ZigBee 通信を行った場合、(B) CTS-Blocking: 制御 PC から周囲の WLAN 端末へ直接 CTS を送信した場合、(C) AA CTS-Blocking: 制御 PC から AP へ RTS を送信した場合のそれぞれについて実験を行った。

評価には 2 つの軸を設けた。1 つ目の軸ではマクロな ZigBee 通信成功率を知るために、1000 サイクル全体での ZED の平均通信成功台数を評価した。通信成功台数は 6.1 での定義と同様に、1 回のデータ収集サイクルで ZED が ZC へ送信成功した台数であり最大値は 10 台である。2 つ目の軸では、ミクロな ZigBee 通信成功率を知るために、1 サイクル毎の ZED の通信成功台数を評価した。

6.3 マクロな ZigBee 通信成功率の比較

図 6.3 に、実験種別 (A) Normal, (B) CTS-Blocking, (C) AA CTS-Blocking の場合における ZigBee 通信効率を示す。縦軸の ZigBee 通信効率は、1000 サイクル中の ZED の延べ通信成功台数を最大の総通信成功台数で除した百分率である。図 6.3 より、以下の 2 つのことがわかる。

1. (C) AA CTS-Blocking は (A) Normal に比べ、通信効率が向上している。(C) AA CTS-Blocking と (A) Normal では約 8 % 程度の差がある。これは、WLAN 通信をブロックしない状況に比べて AA CTS-Blocking 方式による WLAN 通信のブロック効果が大きく出ているためであると考えられる。
2. (C) AA CTS-Blocking は (B) CTS-Blocking に比べ、通信効率が向上している。(C) AA CTS-Blocking と (B) CTS-Blocking では約 5 % 程度の差がある。これは、制御 PC よりも送信電力の大きい AP に CTS フレームを送信させていることに加え、RSSI による AP の選択アルゴリズムが、AA CTS-Blocking による周辺の WLAN 通信のブロック効果を強め CTS-Blocking の問題点であった隠れ端末問題が改善されていると考えられる。

以上より、AA CTS-Blocking による通信成功台数は平均的に向上していることが確認できた。

6.4 ミクロな ZigBee 通信成功率の比較

6.3 では、AA CTS-Blocking 手法によりマクロな ZigBee 通信成功率が向上したことを示したが、平均だけでは ZED の通信成功台数が 1 サイクル毎にどう変化しているかは確認できない。そこで、局所的な ZigBee 通信成功率も AA CTS-Blocking 手法で向上できることを検証するために、各サイクルにおける通信成功台数を評価した。

図 6.4 に，実験種別 (A) Normal, (B) CTS-Blocking, (C) AA CTS-Blocking の場合におけるサイクル毎の ZigBee 通信成功台数分布を示す．縦軸の度数割合は，1000 サイクル中における ZED の各通信成功台数のサイクル数を，1000 サイクルで除した百分率である．図 6.4 より，以下の 2 つのことがわかる．

1. 通信成功台数が 5 台以下の度数割合は，(A) Normal, (B) CTS-Blocking, (C) AA CTS-Blocking であまり変化がない．これは，AA CTS-Blocking 方式を適用しても WLAN 通信をブロックする効果をなさないほど WLAN 通信が混雑していたと考えられる．
2. (C) AA CTS-Blocking は (A) Normal, (B) CTS-Blocking に比べて通信成功台数が 7 台，8 台の時の度数割合が大幅に減少しており，代わりに 9 台，10 台の時の度数割合が大幅に増加している．これは，(A) Normal, (B) CTS-Blocking を利用した場合では WLAN 通信による干渉で ZigBee ノード間通信が失敗していた場合が，AA CTS-Blocking 方式を利用することで成功するためと考えられる．

以上より，AA CTS-Blocking による通信成功台数の変化は，1000 サイクルの平均といったマクロな視点だけでなく 1 サイクル毎といったミクロな視点でも改善していることが確認できた．

6.5 考察

hogehoge

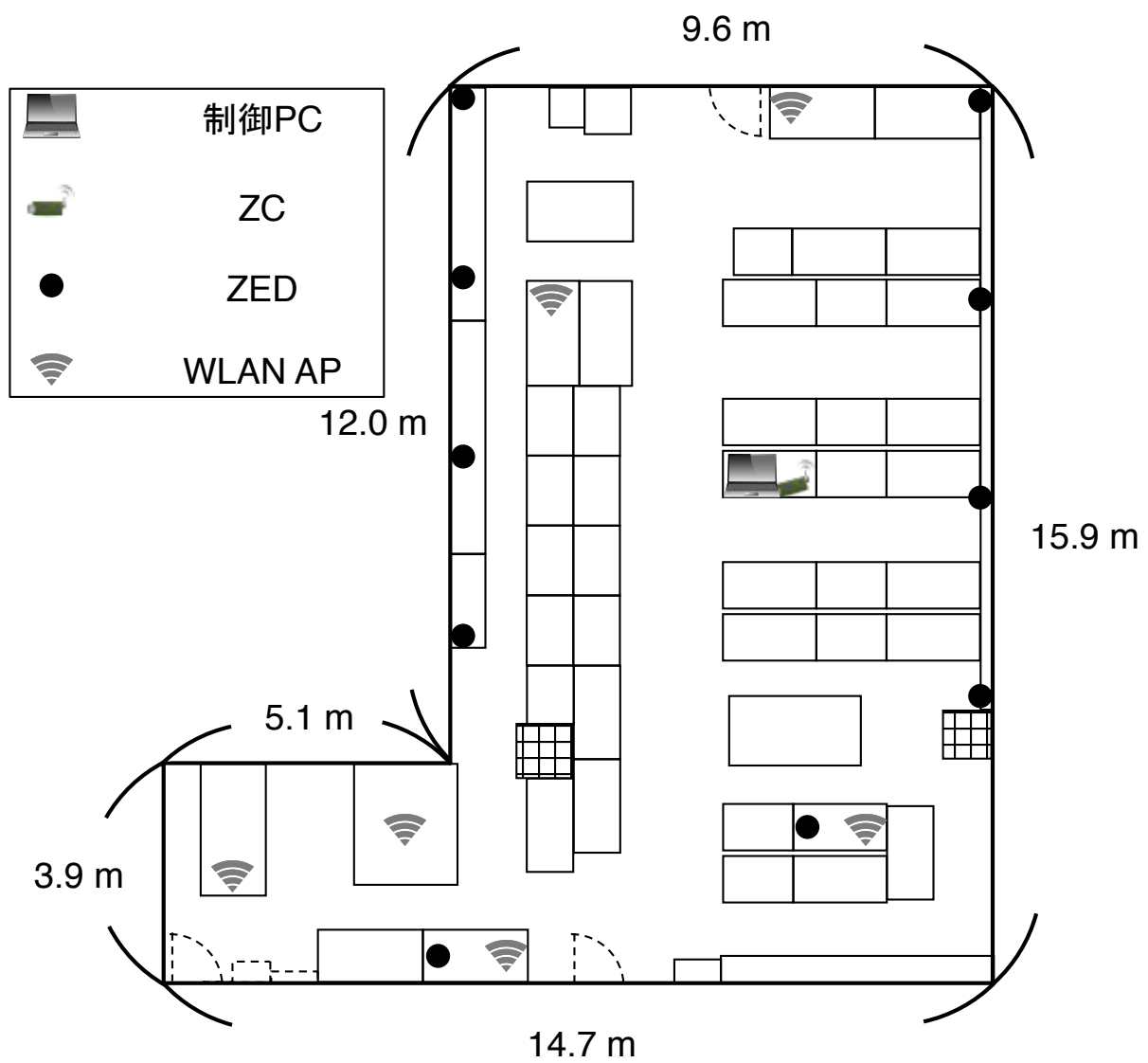


図 6.2: 実験環境の概略

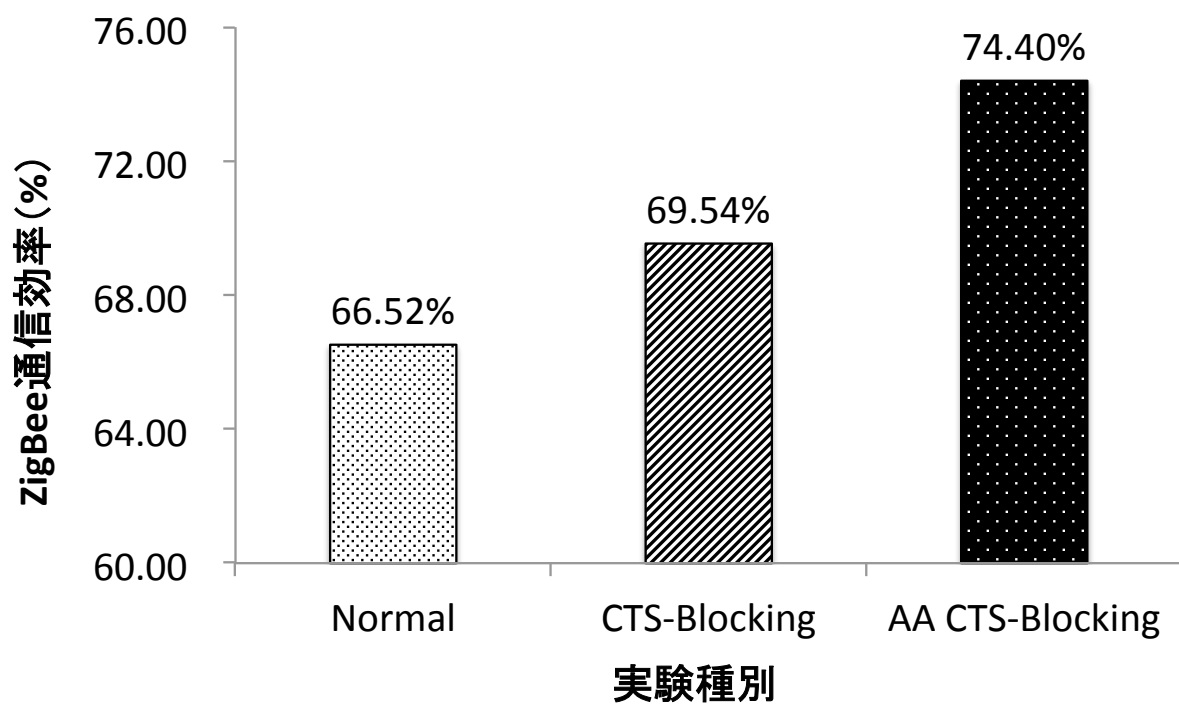


図 6.3: ZigBee 平均通信成功率

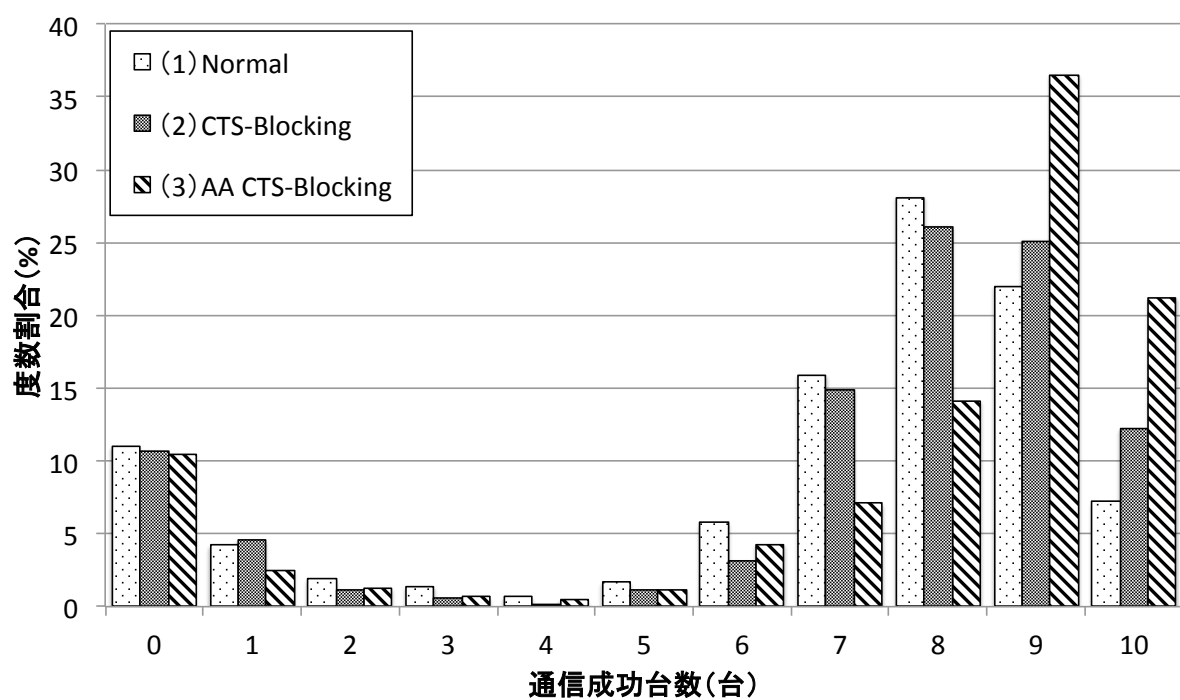


図 6.4: サイクル毎の ZigBee 通信成功台数分布

第 7 章

おわりに

本稿では，WLAN と ZigBee の共存に向けた AA CTS Blocking 方式を示した．AA CTS Blocking を利用したデータ収集システムを実装し，実証評価を通じて AA CTS Blocking の有効性を検証した．この結果，既存手法よりも通信成功率を 5% 改善できることを確認した．今後，通信成功率のさらなる向上に向けた RTS 送信先 AP の選択手法を検討する予定である．

謝辞

筆者に本研究の機会を与えていただき、様々にご指導を頂きました九州大学大学院システム情報科学研究院の福田晃教授に深い感謝の意を表します。また、本研究を進めていく上で多くの意見を頂き、さまざまな相談に応じて頂きました九州大学大学院システム情報科学研究院の石田繁巳助教に心より感謝いたします。本研究に対して多くのご助言とご指摘を頂きました関西大学総合情報学部の田頭茂明准教授に深くお礼を申し上げます。議論の場で筆者に有益なご助言を頂きました九州大学大学院システム情報科学研究院のアシル・アハマッド准教授に深く感謝いたします。また、日々の研究活動において様々な助言と励ましを頂きました、九州大学システム LSI 研究センターの久住憲嗣准教授に心よりお礼を申し上げます。最後に、日頃の研究活動において様々な協力を頂きました、九州大学大学院システム情報科学研究院福田・久住・アシル研究室諸氏に深く感謝し、お礼を申し上げます。

参考文献

- [1] K. Shuaib, M. Boulmalf, F. Sallabi, and A. Lakas, “Co-existence of Zigbee and WLAN, a performance study,” Proc. IEEE Wireless Telecommunications Symposium (WTS), pp.1–6, April 2006.
- [2] J. Hou, B. Chang, D.-K. Cho, and M. Gerla, “Minimizing 802.11 interference on Zigbee medical sensors,” Proc. Int. Conf. Body Area Networks (BodyNets), pp.1–8, April 2009.
- [3] X. Zhang and K.G. Shin, “Enabling coexistence of heterogeneous wireless systems: Case for ZigBee and WiFi,” Proc. MobiHoc, pp.1–11, May 2011.
- [4] 村田亮介, 高木洋介, 小川将克, 奥川雄一郎, 高谷和宏, 秋山佳春, “無線 LAN と ZigBee 間の干渉回避方式の実験評価,” 信学技報, pp.49–53, モバイルネットワークとアプリケーション研究会 (MoNA2013-57), Jan. 2014.
- [5] M.L. Huang and S.-C. Park, “A WLAN and ZigBee coexistence mechanism for wearable health monitoring system,” Proc. Int. Symp. Communications and Information Technology (ISCIT), pp.555–559, Sept. 2009.
- [6] L. Tytgat, O. Yaron, S. Pollin, I. Moerman, and P. Demeester, “Avoiding collisions between IEEE 802.11 and IEEE 802.15.4 through coexistence aware clear chan-

- nel assessment,” EURASIP J. Wireless Communications and Networking, vol.2012, no.137, pp.1–15, April 2012.
- [7] J. Huang, G. Xing, G. Zhou, and R. Zhou, “Beyond co-existence: Exploiting WiFi white space for ZigBee performance assurance,” Proc. IEEE Int. Conf. Network Protocols (ICNP), pp.305–314, Oct. 2010.
- [8] C.-J. Liang, N.B. Priyantha, J. Liu, and A. Terzis, “Surviving Wi-Fi interference in low power ZigBee networks,” Proc. ACM SenSys, pp.309–322, Nov. 2010.
- [9] 和泉 晃, 三輪直樹, 田頭茂明, 荒川 豊, 福田 晃, “ネットワーク側測位における端末固有の RSSI の特性を用いた測位精度向上手法の提案,” 信学技報, pp.1–6, モバイルマルチメディア通信研究会 (MoMuC2012-40), March 2013.
- [10] IEEE Standards Association, “IEEE standard for local and metropolitan area networks — specific requirements: Part 11 (IEEE 802.11-2007),” June 2007. <http://standards.ieee.org/>.
- [11] 住友精密工業, “無線センサネットワーク MOTE ZigBee 無線タイプ・センサネットワーク MICAz”. <http://www.xbow.jp/zigbee-smartdust.html>.
- [12] IEEE Standards Association, “IEEE standard for local and metropolitan area networks — specific requirements: Part 15.4 (IEEE 802.15.4-2006),” Sept. 2006. <http://standards.ieee.org/>.

発表論文

1. 佐伯良光, 石田繁巳, 田頭茂明, 福田晃, “WLAN と ZigBee の共存に向けた AP-Assisted CTS-Blocking の評価,” 電子情報通信学会技術報告, 情報ネットワーク研究会, Vol.2014-IN-●●, No. ●●, p. ●-●, 2015.
2. 佐伯良光, 石田繁巳, 田頭茂明, 福田晃, “WLAN と ZigBee の共存に向けた AP-Assisted CTS-Blocking の初期的評価”, 電子通信情報学会 2015 年総合大会講演論文集 ●-●-●, p. ●, 2015.