

修士論文

アドホックネットワークにおける  
スループット改善に関する  
MACプロトコルの研究

指導教官

浅野 正一郎教授

2008年2月4日提出

東京大学大学院

情報理工学研究所 電子情報学専攻

66434 西村 康孝



# 目次

<b>第1章</b>	<b>序論</b>	<b>2</b>
1.1	はじめに . . . . .	2
1.2	本論文の構成 . . . . .	3
<b>第2章</b>	<b>標準 MAC プロトコル</b>	<b>4</b>
2.1	物理層 . . . . .	4
2.1.1	パスロスモデル . . . . .	4
2.1.2	SINR . . . . .	5
2.1.3	Transmission Range(送信エリア) . . . . .	5
2.1.4	Carrier Sense Range(キャリアセンスエリア) . . . . .	6
2.2	CSMA/CA . . . . .	6
2.3	隠れ端末問題、さらし端末問題 . . . . .	7
2.4	DCF(Distributed Coordination Function) . . . . .	8
<b>第3章</b>	<b>送信電力制御についての</b>	
	一般動向	<b>11</b>
3.1	送信電力制御手法の目的 . . . . .	11
3.2	シンプルな送信電力制御手法 . . . . .	13
3.3	従来手法 . . . . .	14
3.3.1	考え方 . . . . .	14

---

3.3.2	実現方法	15
<b>第4章</b>	<b>提案手法</b>	<b>17</b>
4.1	従来手法の問題点	17
4.1.1	変更点	17
4.2	mode1	19
4.2.1	mode1-のプロトコル	19
4.2.2	Access Window サイズ	21
4.2.3	RTS2 バックオフ	21
4.2.4	確率制御	22
4.2.5	$P_{post-interfere}$ の予測	24
4.3	RRTS-mode	24
4.3.1	RRTS	26
4.3.2	RRTS-mode のプロトコル	26
4.3.3	RRTS , RTS3 の確率制御	28
4.4	RRTS2-mode	29
4.5	TPC-mode1	31
4.5.1	$P_{T-DATA1}$ の決定方法	32
4.6	TPC-RRTS-mode , TPC-RRTS2-mode	33
4.7	MAC プロトコルの比較	33
4.7.1	同時通信エリアの拡大	33
4.7.2	コントロールパケットサイズ	33
4.7.3	送信電力	34
<b>第5章</b>	<b>シミュレーションモデル</b>	<b>35</b>

---

5.1	シミュレーション条件 . . . . .	35
5.2	定義 . . . . .	35
5.2.1	スループット . . . . .	35
5.2.2	同時通信成功数 . . . . .	35
5.2.3	コントロールパケット効率 . . . . .	36
5.3	マルチホップの影響 . . . . .	37
5.4	端末の移動の影響 . . . . .	38
<b>第6章</b>	<b>シミュレーション結果</b>	<b>39</b>
6.1	隠れ端末の同時受信 . . . . .	39
6.2	400×400 25 端末 固定電力でのシミュレーション . . . . .	41
6.2.1	400×400 25 端末におけるシミュレーションのまとめ . . . . .	49
6.3	400×400 25 端末における 送信電力制御を用いたシミュレーション . . . . .	52
6.3.1	400×400 送信電力制御を用いたシミュレーションのまとめ	54
6.4	600×600 36 端末におけるシミュレーション . . . . .	58
6.5	600×600 36 端末 移動環境におけるシミュレーション . . . . .	60
<b>第7章</b>	<b>結論</b>	<b>63</b>
7.1	まとめ . . . . .	63
7.2	今後の課題 . . . . .	63
7.2.1	公平性 . . . . .	63
7.2.2	コントロールパケット効率の向上 . . . . .	64
7.2.3	フェージング環境での評価 . . . . .	64
<b>第8章</b>	<b>謝辞</b>	<b>66</b>

# 第1章 序論

## 1.1 はじめに

近年，端末の高性能化にともないアドホックネットワークが注目を集めている．アドホックネットワークとは無線通信装置が付いている移動可能な端末から構成されるネットワークである．アドホックネットワークの大きな利点としては，既存のインフラストラクチャーを必要としないことが挙げられる．そのため，アドホックネットワークの利用方法として考えられるのは，砂漠，森林，山地等のインフラストラクチャーの設置が困難な環境や，フェスティバルやイベント会場，会議等で一時的にネットワークを構築する場合などに適している．

しかし，アドホックネットワークは多くの技術的問題点を抱えており，MAC プロトコルに関する研究も盛んに行われている．同じ周波数帯を共有しあう無線環境では，他端末からの干渉が大きな問題となる．特に端末の密度が高い状態では，チャンネルアクセスの競合が起こり，パケットの衝突が頻繁に発生し，伝送特性が劣化するため，ネットワークスループットの改善が必要とされている．アドホックネットワークで広く用いられている CSMA/CA 方式は，分散的なアクセス制御方式であるが，CSMA/CA のアクセス制御を用いた場合，隠れ端末問題，さらし端末問題の影響によりスループットの低下が問題となる．802.11 標準では DCF を用いることにより隠れ端末問題の改善を図っている．しかし，DCF では過剰に周囲端末を待機させるので，スループット改善の余地は多く残されており，さらし

端末問題については未解決である。

DCFに対してスループット向上を図る MAC プロトコルは多く研究されている。送信電力制御により、周波数の空間的利用効率を高める手法や、指向性アンテナを利用した方法、サブチャネルを新しく用意し制御信号用に使用する手法等 [12][13]、様々存在する。文献 [3][4] では新たにハードウェアを加えることなく、MAC プロトコルの改良のみでスループットの向上が達成されている。しかし、問題点が依然として残されており、本論分ではさらなるスループット向上を図る MAC プロトコルの提案を行う。

## 1.2 本論文の構成

本論文の構成は以下の通りである。第2章では、アドホックネットワークにおける代表的な MAC プロトコルについて説明する。第3章では、アドホックネットワークにおける送信電力制御についての研究動向を概観する。第4章では、ネットワークスループットの向上を図る MAC プロトコルの提案を行い、説明する。第5章では、提案したシミュレーションモデルの前提条件および流れを説明する。

第6章では、シミュレーション結果についての評価を行う。シミュレーションは様々な状況に対して行い、それぞれについて結果を記すとともに、その妥当性および傾向について考察を行う。第7章では、シミュレーションの結果をまとめ研究の総括を行う。また今後の課題について記す。

## 第2章 標準MACプロトコル

### 2.1 物理層

#### 2.1.1 パスロスモデル

送信端末から送信された電波は受信端末に到達するまでに次第に減衰していく。本論文では本稿ではパスロスモデルは [3] と同様に two way ground propagation model を仮定する。このモデルでは、送信電力を  $P_t$  とすると受信電力  $P_r$  は次のように表される。

$$P_r = P_t G_t G_r \frac{h_t^2 h_r^2}{d^4} \quad (2.1)$$

ここで、 $G_t$ 、 $G_r$  は送受信アンテナ利得、 $h_t$ 、 $h_r$  は送受信アンテナの高さ、 $d$  は送受信端末間の距離である。本論文ではすべての端末で、 $G_t$ 、 $G_r$ 、 $h_t$ 、 $h_r$  は同じ値とする。すると上述した式は次のように表わされる。

$$P_r = H_{tr} P_t \quad (2.2)$$

ここで  $H$  は送受信端末間のチャネル利得を表す。

### 2.1.2 SINR

パケットを正しく受信するためにはパケットの受信電力  $P_R$  がデコードするために必要な閾値  $P_{R-TH}$  以上であることと, SINR(Signal to Interference and Noise Ratio) が閾値  $SINR_{TH}$  以上であることが必要であり, 次式のように表される.

$$P_R \geq P_{R-TH} \quad (2.3)$$

$$SINR = \frac{P_R}{P_I + P_N} \geq SINR_{TH} \quad (2.4)$$

ここで,  $P_I$  は干渉電力,  $P_N$  は雑音電力である. また, 電波を検知するためには次式を満たすことが必要である.

$$P_R \geq P_{SENSE-TH} \quad (2.5)$$

ここで,  $P_{SENSE-TH}$  は電波を検知するのに必要な最低電力である. また, 送信電力  $P_T$  が既知である場合, 受信電力  $P_R$  からチャネル利得  $H$  は次式のように推定される.

$$H = \frac{P_R}{P_T} \quad (2.6)$$

### 2.1.3 Transmission Range(送信エリア)

Transmission Range とは他の端末からの干渉がない場合, Transmission Range 内の距離の端末はパケットを正しく受信することができる距離を表す. Transmission Range は式 2.3 の条件を満たすことができる距離となる.

### 2.1.4 Carrier Sense Range(キャリアセンスエリア)

Carrier Sense Range 内の距離にいる端末は式 2.5 を満たし、電波を検知することができる。Carrier Sense Range はアンテナの感度によって決まり、Transmission Range よりも大きな値となる。Transmission Range の外にいて、かつ Carrier Sense Range 内の端末はパケットを正しく受信することはできないが、チャンネルがビジー状態であることを検知することができる。

## 2.2 CSMA/CA

アドホックネットワークでは無線周波数帯を各端末で共有しなくてはならない。同じ周波数帯を同時刻、同空間で使用すると、お互いの電波が干渉し合い通信することができない。そのため、各端末間でパケットの衝突を防ぐために MAC(Media Access Control) プロトコルが必要である。無線環境では CSMA/CA(Carrier Sense Multiple Access/Collision Avoidance) 方式という対等分散型のアクセス制御方式が広く用いられている。以下に CSMA/CA 方式の制御方式を説明する。

1. 送信したいパケットを持っている端末はチャンネルが使用されているかどうか確認する。この動作をキャリアセンスと呼ぶ。
2. キャリアセンスの結果、一定時間チャンネルが IDLE 状態であれば、バックオフ状態に移行する。この一定時間を DIFS(Distributed Interframe Space) と呼ぶ。キャリアセンス中、DIFS 時間経過する前にチャンネルが BUSY 状態となった場合は、再び IDLE 状態になるまで待機する。
3. バックオフ状態では、各端末ごとに持っているバックオフカウンタがあり、一定時間 Slot time 経過するごとに 1 つずつ減らしていく。バックオフカウ

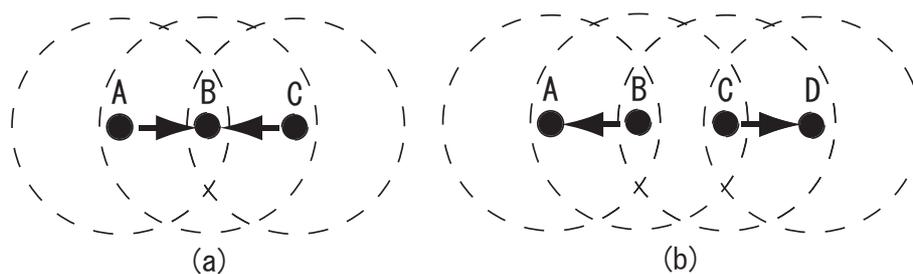


図 2.1: 隠れ端末問題/さらし端末問題

ンタが0になった端末は送信権を与えられ、パケットを送信する。ここで、Slot time は端末がチャネルが IDLE 状態と判断するのに必要な時間である。バックオフカウンタは各端末でランダムな値が用いられることにより、複数の端末が同時にパケットを送信してしまうことを防いでいる。

## 2.3 隠れ端末問題、さらし端末問題

隠れ端末問題について図 2.1 を用いて説明する。マルチホップネットワークでは、ある端末の送信エリアの外に他の端末が存在することが多く、隠れ端末問題は引き起こされる。ここで図の点線は端末の送信レンジを表している。図 2.1(a) に示すように、端末 A が端末 B にパケットを送信しているとき、端末 C が端末 B にパケットを送信したい状況を考える。端末 C は端末 A の電波を検知することができないので、チャネルがアイドル状態であると判断しパケットを送信してしまう。すると端末 B でパケットの衝突が起こり、パケットを再送信する必要が生じてネットワークスループットの重大な低下を招いてしまう。

次にさらし端末問題について説明する。図 2.1(b) に示すように、端末 B が端末 A にパケットを送信しているとき、端末 C が端末 D にパケットを送信したい状況を考える。このとき、端末 C は端末 B の電波を検知しているため、端末 B の送信

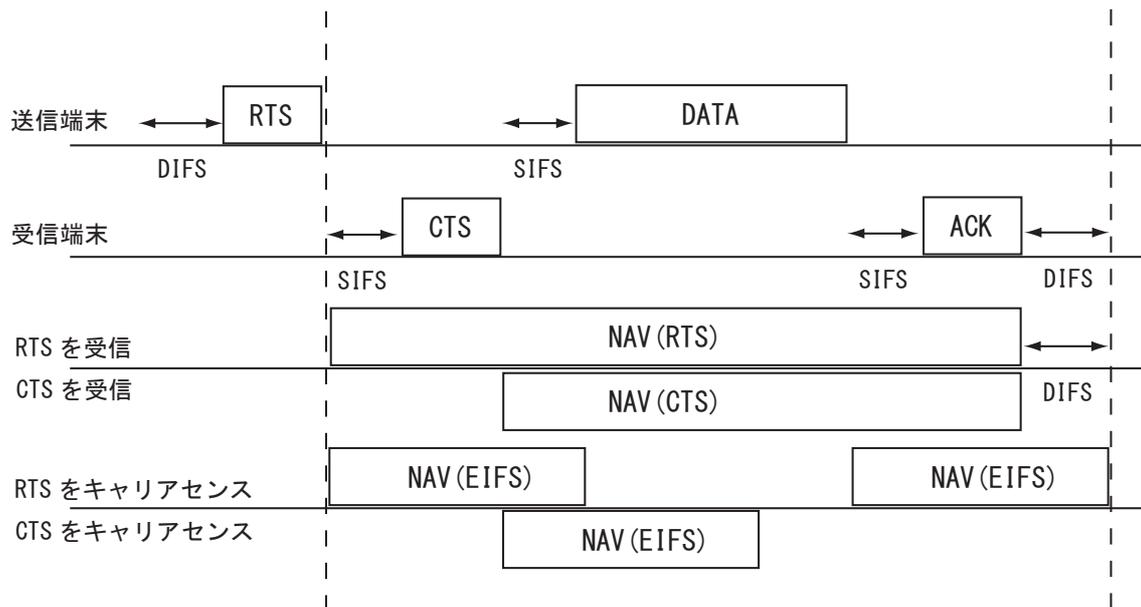


図 2.2: DCF

が終了するまで端末 D への送信を控えてしまうが、端末 C から端末 D へのパケット送信は端末 B から端末 A へのパケット送信を妨害せずに同時に通信を行うことができる。このように、さらし端末問題によってもネットワークスループットは低下してしまう。

## 2.4 DCF(Distributted Coordination Function)

802.11standard では CSMA/CA に加え、DCF が MAC プロトコルとして用いられている。DCF ではデータパケットの送信に先立ち RTS(Request To Send)/CTS(Clear To Send) パケットの交換を行い、隠れ端末問題の影響を低減している。送信端末を A、受信端末を B として以下に DCF の動作を簡単に説明する。

1. CSMA/CA により、アクセス権を得た A は RTS を送信する。
2. RTS を受信した B は SIFS(Short InterFrame Space) 待機した後、CTS を返信

する。

3. CTS を受信した A は SIFS 待機した後，DATA パケットを送信する。
4. DATA パケットを受信した B は SIFS 待機した後，ACK を返信する。
5. A が ACK を受信することにより，1 つのデータパケット送信が終了する。

CTS には送信先の MAC アドレスと NAV(Network Allocation Vector) と呼ばれる ACK の送信が終了するまでの残り時間が含まれており，RTS にはさらに自分の MAC アドレスが含まれる。A,B 以外の端末が RTS，CTS を受信した際，NAV の値を読み取り通信が終了するまで待機する。ここで，SIFS は DIFS よりも小さい値であり，通信の途中で他の端末に割り込まれることのないように優先的にアクセスできるようになっている。また EIFS(Extended Interframe Space) はセンシングした場合の待機時間であり，SIFS，ACK 送信時間，DIFS の和である。

DCF のアクセス制御を図 2.2 に示す。RTS や CTS を受信した端末は NAV の値を読み取り，ACK の送信が終わるまで待機する。RTS や CTS の電波を検知した端末は，チャンネルが IDLE 状態になってから EIFS 待機する。

DCF により隠れ端末問題の影響が低減される。図 2.1(a) の例を用いて説明すると，端末 A は端末 B にデータパケットを送信したいとき，まず端末 A は RTS パケットを端末 B に送信する。端末 B は RTS パケットを受信したら CTS パケットを返信する。この際，端末 C は端末 B からの CTS パケットを受信することにより，将来端末 B がデータパケットを受信することを予測することができる。端末 C は端末 A がデータパケット送信している間待機し，終了してから端末 B へパケットを送信する。このように，RTS/CTS の交換により，パケットの衝突によるスループットの低下と再送信のコストを抑えることができたが，それでも例外的にパケットの衝突が起きてしまう状況も存在する。例えば図 2.1(a) の場合，A から B へ RTS

が送信された後、B から A への CTS と C から B への RTS が同時に送信された場合を考えてみる。A は B からの CTS を受信してデータパケットを送信する。一方、C は B からの CTS を受信することができなかつたので、再び B に対して RTS の再送信を試みた場合、A のデータパケットと C の RTS との衝突が起こる。このように、RTS/CTS の交換が完了すれば周囲の端末を待機させることができるが、RTS/CTS の交換が終わる前に他の端末に邪魔をされてしまうと、パケットの衝突が発生する。しかし、RTS/CTS はデータパケットに比べると非常に小さいので、わずかの時間で RTS/CTS の交換は終了するため、上述したような状況が起こることは確率的に小さく、DCF により隠れ端末問題の影響は低減する。

さらし端末問題に関しては、DCF を用いても改善は見られない。図 2.1(b) を考えると、B と A の RTS/CTS の交換が終了すると、C は B からの RTS により  $B \rightarrow A$  の通信が終了してから D に対して RTS を送信するため DCF の際と比べ変化は見られない。

さらし端末問題やその他の状況でも、お互い妨害することなく同時に通信を行なうことができる場合も多々存在する。DCF は多くの端末に通信を控えさせることにより、データパケットの衝突を大幅に防ぐことが実現できているが、ネットワークスループットの点では改善の余地があると考えられる。本稿では、RTS/CTS を受信した端末でも、同時通信できる可能性を探り、ネットワークスループット向上を図る MAC プロトコルを提案し、シミュレーションにより評価を行う。

## 第3章 送信電力制御についての 一般動向

### 3.1 送信電力制御手法の目的

アドホックネットワークにおける送信電力制御手法は多く研究されている。送信電力制御を使用する目的としては大きく3つに分類される。一つ目は端末の消費電力の低減である。端末はバッテリーによって駆動しているため、端末の消費電力を低減することが重要な問題になってくる。消費電力を低減することにより、端末のライフタイムが長くなり、ネットワークのライフタイムも長くなる。端末の消費電力には、CPUによる電力、計算処理による電力等が考えられるが、パケットの送受信による電力は多くを占める。IEEE802.11Standardでは端末の送信電力は固定となっているが、送受信端末間の距離に関わらず固定電力でパケットを送信することは消費電力の面から考えると無駄が多い。例えば通信相手が近くにいる場合にも大きな送信電力を使用することは、無駄な電力を消費することになる。

2つ目はネットワークトポロジーを動的にコントロールするために送信電力制御を用いる方法である。端末の密度が小さい場合は送信電力を高くすることにより、ネットワークから孤立する端末が生じないようにし、逆に密度が高い場合は、多くの端末に干渉電力を与えないように送信電力を小さくする方法が研究されている [10][11]。

3つ目の目的はスループットの向上である。送信電力を制御することにより、周

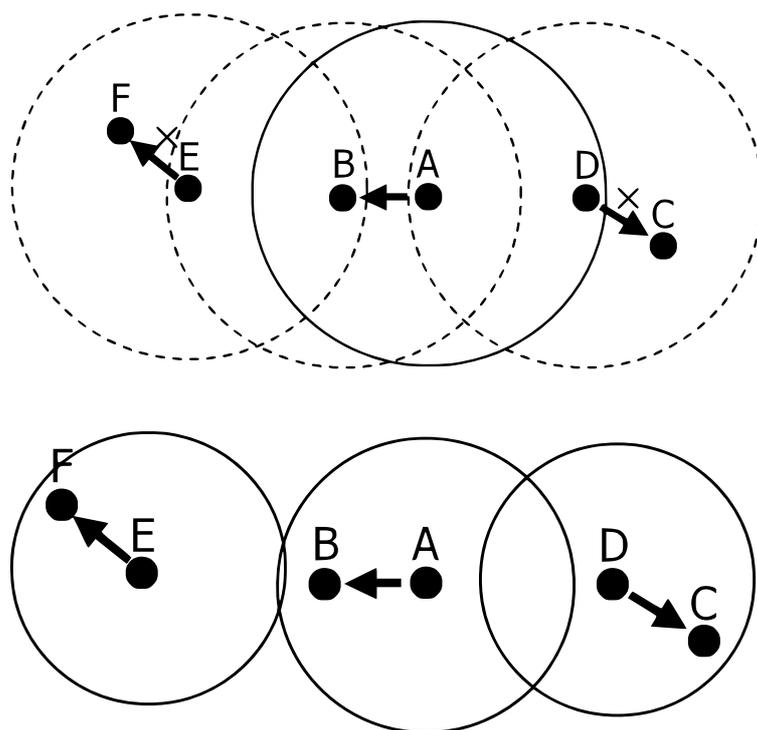


図 3.1: 送信電力制御による空間的周波数利用効率の向上

他の端末への干渉を抑えることができるので、空間的に周波数の再利用が可能となり、ネットワークスループットの向上が期待できる。図 3.1 上図に示すように、各端末が固定電力で送信する場合、周囲への干渉が大きすぎて同時刻に一組の通信しか実行することができない。しかし、図 3.1 下図に示すように、各端末が適度に送信電力を制御することにより、お互いのパケットが衝突を引き起こすことなく同時に通信を行うことができる。これにより、ネットワークスループットの向上が実現される。

本論分では 3 つ目のスループット向上のための送信電力に焦点をあてることとする。

## 3.2 シンプルな送信電力制御手法

文献 [1] 等では簡単な制御で送信電力手法が MAC プロトコルに導入されている。以下この手法を SIMPLE と呼ぶこととする。この手法では、RTS/CTS の交換によって、DATA パケットの送信電力が決定される。まず、RTS/CTS は最大電力  $P_{max}$  で送信する。送信端末が受信端末から CTS を受信した際の受信電力を  $P_{R-CTS}$  とすると、チャンネル利得  $H$  は式 2.6 を用いると次式のように推定される。

$$H = \frac{P_{R-CTS}}{P_{max}} \quad (3.1)$$

送信端末はチャンネル利得を推定した後、DATA パケットの送信電力  $P_{T-DATA}$  を次式のように決定する。

$$P_{T-DATA} = \frac{P_{R-TH}}{H} \quad (3.2)$$

式 3.2 によって送信されたパケットは受信端末に受信電力  $P_{R-DATA}$  で受信される。

$$P_{R-DATA} = H \times P_{T-DATA} = P_{R-TH} \quad (3.3)$$

SIMPLE の問題点は文献 [2] で述べられている。まず従来の IEEE802.11 と同じように、RTS/CTS を受信する端末は NAV を設定し、DATA の送信が終了するまで待機するのでネットワークスループットの向上は期待できない。さらに、DATA パケットの送信電力を小さくすることにより、DATA の受信電力も小さくなり、周囲からの干渉に弱くなり、DATA パケットが受信できないことが多くなる。この問題を避けるために文献 [2] では DATA パケット送信中に送信電力を定期的に最大電

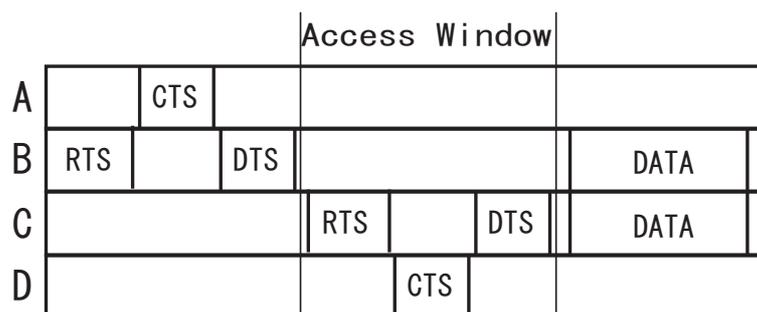


図 3.2: 従来手法の MAC プロトコル

力に切り替えることによりキャリアセンスレンジを拡大し、干渉されるのを防いでいる。

SIMPLE では式 2.3 のみを考慮して送信電力を決定しているため、周囲からの干渉によって妨害されてしまうことがある。よって、式 2.4 も考慮し、干渉に耐えられる送信電力で通信を行う必要がある。

### 3.3 従来手法

#### 3.3.1 考え方

文献 [3][4] では新たにハードウェアや周波数帯を必要とせずに DCF を上回る MAC プロトコルが提案されている。アルゴリズムの簡単な流れを図 3.2 に示す。A が B にデータパケットを送信したい場合、まず、RTS, CTS パケットの交換を行い、ここではさらに DTS(Decide-to-Send) パケットを送信している。DTS については後で説明する。これらのコントロールパケットには、次の Access Window と呼ばれるタイムスロットの数とデータパケットの送信時間の情報が含まれている。A,B の周辺のパケットを送信したい端末は、次の Access Window の間にコントロールパケットを交換する。そして送信電力を調整することにより、C が A が

ら B の通信を妨害しないことが確かめられれば、同時にデータパケットを送信する。ここで同時に通信できるかどうかの判断は受信端末で行われる。受信端末は、送信端末がデータパケットを送信するときの電力を決定し、さらに、周辺端末から受ける干渉電力を推定することによって、必要な SINR の値を確保できるか計算し、同時通信の可否を判断する。図 3.2 の例では、Access Window の間に一組しかコントロールパケットの交換をすることはできないが、例えば Access Window の長さを 2 倍にすれば、Access Window の期間に 2 組の端末がコントロールパケットを交換することができ、通信の妨害が起きないことがわかれば、3 組の端末がデータパケットを同時に通信することができる。

DCF では RTS/CTS パケットを受信した全ての端末は、データパケットのが終了するまで送信を控えていたが、この手法では、RTS/CTS パケットを受信した場合は色々な制限を受けるが、お互いのパケット通信による干渉を考慮することにより、同時に通信できるかどうかの可能性を探る。これによりネットワークスループットの向上が図られている。

### 3.3.2 実現方法

本論分では簡単のため最初の RTS/CTS を RTS1/CTS1 と呼び、2 番目の通信の予約に用いられる RTS/CTS を RTS2/CTS2 と呼ぶこととする。同時通信を実現する手段について簡単に説明する。RTS1 には従来の情報に加えて Access Window の長さを添付する。受信端末が RTS1 を受信したら、現在の干渉電力を考慮し、データパケットの送信電力を決定する。また、式 2.4 と式 2.3 を考慮し、さらにどのくらいの干渉電力が追加されてもデータパケットを正しく受信できるのかを計算する。この 2 つを CTS1 に添付し返信する。送信端末は CTS1 から送信電力を読み取り、DTS に添付し周囲へ伝える。送受信端末は Access Window の間待機する。

周囲の端末で RTS1 を受信した端末は将来データパケットの送信の際に干渉を受けることが分かり、その時刻を保存しておく。CTS1 を受信した端末は、最初通信を妨害しないように送信電力に制限を受ける。DTS を受信した端末は将来受ける干渉電力が計算できる。

Access Window 中に周囲の端末は上述の 2 点を考慮しながら RTS2/CTS2 の交換を試みる。最初に予約された通信を 2 番目の通信が妨害しないことが計算されれば、データパケットを同時に送信する。

## 第4章 提案手法

### 4.1 従来手法の問題点

従来手法のプロトコルでは、最初に通信を予約するために RTS1/CTS1 交換が行われ、周囲の端末で RTS1, CTS1 の少なくとも 1 つを受信した端末が Access Window 中に 2 番目の通信の予約を試みている。つまり、最初に予約した送受信端末の送信エリア内に存在する端末が Access Window 中に RTS2 の送信にチャレンジすることができる。送信エリア外の端末は最初に通信が予約されたことがわからないので、Access Window の開始時刻がわからない。ここで図 4.1 のトポロジーを用いて具体的に考えてみる。A → B, C → D, E → F の 3 つのフローが存在し、A → B の RTS1/CTS1 交換が終了した状況を考える。E は A の RTS により Access Window の開始時刻がわかり、F と RTS2/CTS2 の交換を試みることができる。一方 C は A, B の送信エリア外に存在しているため同時通信の予約を開始できない。

#### 4.1.1 変更点

従来手法で用いられていた DTS は使用しない。従来手法では受信端末にデータパケットの送信電力を決定させ、その電力を DTS を用いて周囲へ伝えるという役目があった。本手法では新たに RRTS パケットを導入するのでコントロールパケットが多くなってしまいうことを避け、DTS を使用しないこととする。そのため、送信端末がデータパケットの送信電力を決定し、RTS1 に送信電力を添付することと

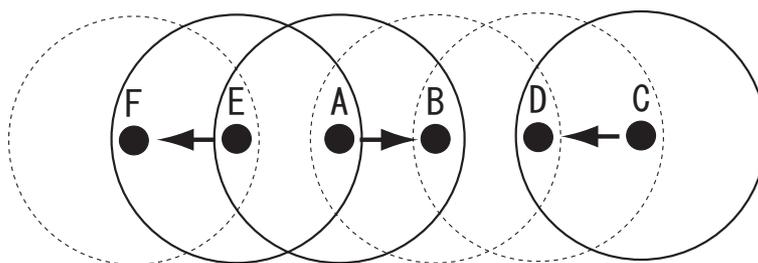


図 4.1: トポロジー 1

する .

従来手法では Access Window の長さを動的に変化させて、状況によっては 2 組、3 組の同時通信が行えるようになっていた。しかし、3 組以上の同時通信が行えるような状況になる確率は小さいので、本手法では Access Window の長さは 1 組が通信の予約を行えるような長さか、Access Window の長さ 0、つまり DCF に切り替える場合の 2 通りとした。

また、ACK について文献 [2] ではデータパケットと同じ送信電力で複数の ACK を同じタイミングで送信していたが、データパケットを同時に送信することができたとしても、ACK を同時に送信してコリジョンが起こらないとは限らないので、ACK のタイミングをずらして送信するように変更した。RTS1/CTS1 から ACK までの流れは図 4.2 となる。

また、本論文では送信電力が固定の場合と送信電力制御を用いる場合の 2 パターンについて評価を行い、電力制御の有用性を示す。まずは固定電力の場合の制御方法について説明する。

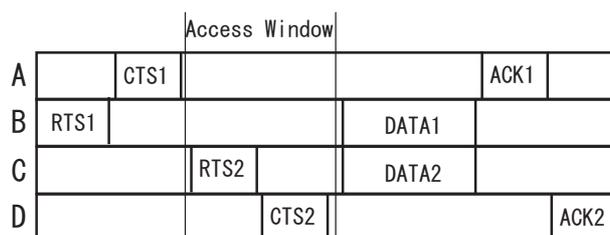


図 4.2: mode1

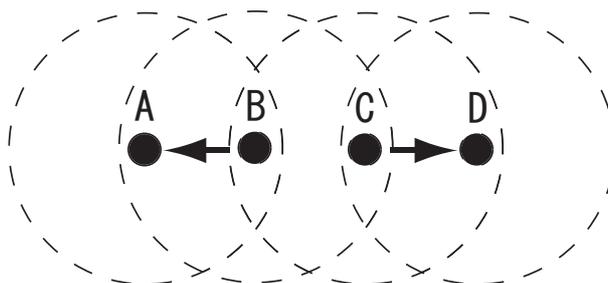


図 4.3: トポロジー 2

## 4.2 mode1

### 4.2.1 mode1-のプロトコル

mode1 は上述したようにいくつか変更点はあるものの、従来手法に近いプロトコルである。図 4.3 を用いて、 $B \rightarrow A$  が最初に通信を予約し、次に  $C \rightarrow D$  の通信が予約され、同時通信が行われる場合を例にプロトコルの流れを説明する。

B は従来の DCF と同様に CSMA/CA によりチャネルアクセス権を得たら RTS1 を送信する。B は A にデータパケットの際の送信電力  $P_{T-DATA}$  を伝える必要があるが、固定電力の場合はすべての端末が既知であるので添付する必要はない。A は RTS1 を受信したら式 2.6 よりチャネル利得  $H_{AB}$  を推定し、耐えられうる干渉電力  $P_{add-interfere}$  を次式のように計算する。

$$P_{add-interfere} = \frac{P_{R-RTS}}{SINR_{TH}} \times MARGIN \quad (4.1)$$

ここで MARGIN は  $(0 < MARGIN \leq 1)$  の定数である．A は  $P_{add-interfere}$  を CTS1 に添付し，B に返信する．B は CTS1 を受信したら Access Window の時間待機する．mode1 の Access Window の長さ  $AW_{mode1}$  はバックオフ時間，RTS2/CTS2 の送信時間，SIFS から成る．本論分では  $MARGIN = 0.9$  とする．

周囲の端末で RTS1 を受信した端末 I は，仮定より， $P_{R-RTS1} = P_{R-DATA}$  であるので，A から将来受ける干渉電力  $P_{post-interfere}$  と時刻を保存する．CTS1 を受信した端末 J は式 2.6 よりチャネル利得  $H_{AJ}$  を推定する．そして  $A \rightarrow B$  の通信が終了するまでのしばらくの間，次式によって送信電力の上限が  $P_{T-allowed}$  に制限される．

$$P_{T-allowed} = \frac{P_{add-interfere}}{H_{AJ}} \quad (4.2)$$

送信電力が固定電力  $P_{max}$  の場合， $P_{max} > P_{T-allowed}$  ならば，データパケットの送信を控えなければならない．Access Window に入ると，C は送信電力を制限されなければ RTS2 の送信を試みる．RTS2 送信の際のバックオフを RTS2 バックオフと呼び，ウィンドウサイズを RTS2-BO-WINDOW-SIZE とする．バックオフが終了してもチャネルがアイドル状態であれば RTS2 を送信する．D は RTS2 を受信したら次式より C から正しく受信できるか判断する．

$$POST\_SINR = \frac{P_{R-RTS2}}{P_{post-interfere}} \geq SINR_{TH} \quad (4.3)$$

ここで，D は B からの RTS1 を受信していれば， $P_{post-interfere}$  の値が保持されているが，A の送信エリア外に存在している場合は A からどれほどの干渉を受けるか予測できない．この場合，同時受信できるかどうか判断できないが，その場合は  $P_{post-interfere}$  を予測することができる．予測方法については後に述べることとする．

式 4.3 を満たす場合は同時通信可能と判断し CTS2 を返信する．満たさない場合

は negative-CTS2 を返信し, C がデータパケットを送信しないようにする. negative-CTS2 を送信しなくても, C はデータパケットを送信することはないが, D が送信エリア外にいると判断することを防ぐために必要と考えられるので送信することとする.

CTS2 を受信した C は, バックオフの残り時間待機する必要がある. Access Window はバックオフウィンドウの時間が含まれているが, C は RTS2 送信の際に全て消費してない場合は残り時間待機する必要がある. 待機時間が終了したら B と C はデータパケットを送信する. A と D はほぼ同じ時刻にデータパケットの受信が完了する. A が先に ACK を送信し, その間 D は待機する. その後 D が ACK2 を送信し, 1 連の通信が終了し, この通信による周囲端末の  $P_{add-interfere}$ ,  $P_{post-interfere}$ ,  $P_{T-allowed}$  の情報はリセットされる.

## 4.2.2 Access Window サイズ

mode1 の Access Window サイズ  $AW\_SIZE_{mode1}$  は次式のように表わされる.

$$AW\_SIZE_{mode1} = BO\_W_{RTS2} + T_{RTS2} + SIFS + T_{CTS2} \quad (4.4)$$

ここで,  $BO\_W_{RTS2}$  は RTS2 のバックオフウィンドウサイズ,  $T_{RTS2}$ ,  $T_{CTS2}$  は RTS2, CTS2 の送信時間を表す.

## 4.2.3 RTS2 バックオフ

通常のバックオフでは 1 から 32 の値をランダムに発生させて, 周囲の端末との競合を防いでいるが, RTS2 のバックオフでは通常に比べて競合する端末数が少ないので, 0 から 3 の値をランダムに選択し, バックオフを行うこととする.  $BO\_W_{RTS2}$

は次式のように表わされる .

$$BO\_W_{RTS2} = 3 \times slottime \quad (4.5)$$

#### 4.2.4 確率制御

##### RTS1 の確率制御

従来手法では周囲の干渉の度合により Access Window の長さを決定していたが , RTS1 を送信 (Access Window が 1) するか RTS(DCF) を送信するかは確率  $PROB_{RTS1}$  により決定することとする .  $PROB_{RTS1}$  は次式のように動的に変化する .

$$PROB_{RTS1-DEFAULT} = 90$$

$$\text{If Success } PROB_{RTS1} = \min(100, PROB_{RTS1} + 10)$$

$$\text{Else If Failure } PROB_{RTS1} = \max(10, PROB_{RTS1} - 20)$$

$PROB_{RTS1}$  の初期値は 90 であり , データパケットが正しく送信できた場合は確率を増加させ , 失敗した場合は減少させる . これは , RTS1 により周囲に同時通信を促した場合 , なんらかの原因でうまく同時通信ができず , 干渉によって自分のパケットが正しく送信できないことが続いてしまうことを避けるためである .

##### RTS2 の確率制御

送信電力が制限されていない端末は Access Window 中に RTS2 を送信する権利を持つが , 1 つの端末しか Access Window 中に RTS2 を送信できないため , 同時通信が成功する確率の高い端末が RTS2 を送るべきである . そこで  $PROB_{RTS1}$  と同

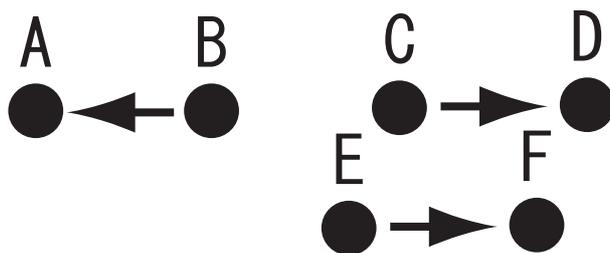


図 4.4: トポロジー 3

様にして  $PROB_{RTS2}$  を次式のように定義する .

$$PROB_{RTS2-DEFAULT} = 50$$

$$\text{If Success } PROB_{RTS2}(K) = \min(100, PROB_{RTS2}(K) + 50)$$

$$\text{Else If False } PROB_{RTS2}(K) = \max(10, PROB_{RTS2}(K) - 10)$$

ここで ,  $PROB_{RTS2}$  の添え字は端末 K の RTS1 により促された RTS2 であることを表す . 図 4.4 において ,  $B \rightarrow A$  の RTS1/CTS1 交換が終了した状況を考えると , E は  $PROB_{RTS2}(B)$  に従い RTS2 を送信する . 位置的にこの同時通信は成功する確率が高いので ,  $PROB_{RTS2}(B)$  は高い値になっていることが望ましい . 一方 ,  $C \rightarrow D$  の RTS1/CTS1 に続いて E が RTS2 を送信する状況を考えると , 位置的にこの同時通信は F において衝突が起きる可能性が高く , F から negative-CTS2 が返ってくる可能性が高い . よって  $PROB_{RTS2}(C)$  は小さい値になっていることが望ましい . このように , どの端末の RTS1 に同期したかにより同時通信できるかどうかは大きく変わってくるので , 端末ごとに  $PROB_{RTS2}$  を保持することが有効であると考えられる .

### 4.2.5 $P_{post-interfere}$ の予測

DがAの送信エリア外に存在する場合、どのようにAからの干渉電力を推定するかを説明する。DはCからのRTS2を受信した際、その時刻からAがどのくらい前にRTS1を送信したか計算できる。ここで、式2.5を満たしていれば、RTS1による電波を検知できるので、その際の受信電力 $P_{RTS1}$ を保存しておくことで、 $P_{post-interfere}$ を予測できる。送信電力が固定の場合は $P_{post-interfere} = P_{RTS1}$ となり、送信電力が固定でない場合でも、 $P_{post-interfere}$ が $P_{RTS1}$ より高くなることはないため安全である。RTS2を受信してからどのくらい前の時刻 $DURATION(RTS2 \rightarrow RTS1)$ の干渉電力を $P_{RTS2}$ とするべきかは次式に示す通りである。

$$DURATION(RTS2 \rightarrow RTS1) = SIFS + T_{CTS1} + BO\_W_{RTS2} + T_{RTS2} \quad (4.6)$$

バックオフや伝搬時間の関係で時刻を正確に特定することはできないが、誤差に閾値を設定すればほとんどの場合特定することが可能である。また、保存すべき過去の干渉電力の情報は長くても数ms前までの情報十分であるため、少ないメモリー量で実現可能である。

## 4.3 RRTS-mode

4.1章で述べた問題点を解決するためのプロトコルを提案する。通常のMACプロトコルは送信端末が主導となりアクションを起こす方法がほとんどであるが、受信端末からRRTS(Request for RTS)を送信することにより通信を開始する手法も研究されている[7]。この考え方を取り入れ、上述の図4.5のようなトポロジーの場合でも同時通信を実現するためにRRTSを新たに導入する。図4.5を例に考えると、 $A \rightarrow B$ のRTS1/CTS1交換が終了した後、CはBからのCTSを受信している

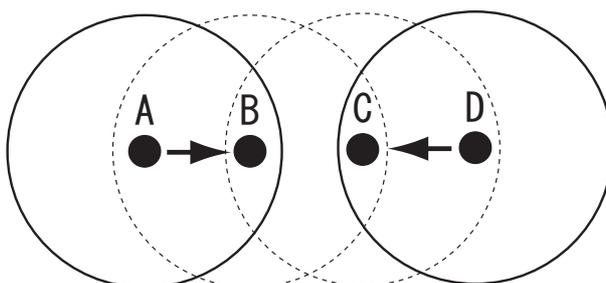


図 4.5: 隠れ端末のトポロジー

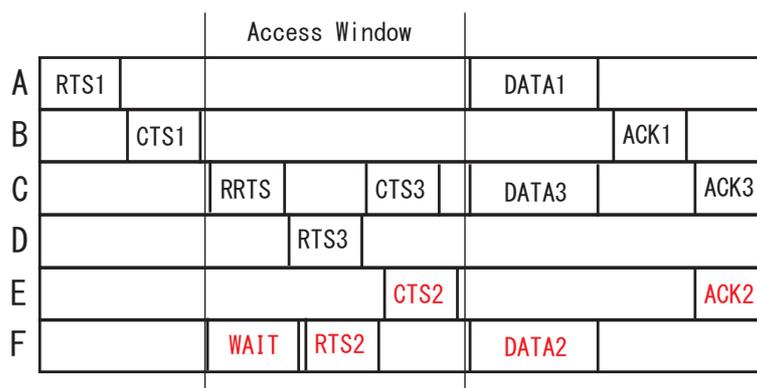


図 4.6: RRTS-mode

ので、将来  $A \rightarrow B$  のデータパケット送信が行われることがわかる。C は同時送信してしまうと B でコリジョンが起きてしまう可能性が高いが、同時受信できる可能性は高い。そこで RRTS パケットを送信し、周囲端末に自分宛の RTS を送信してもらうように促す。RRTS を受信した D は C と RTS/CTS を交換し、二つのデータパケットが同時に送信される。以降 RRTS によって促されたコントロールパケットを RTS3/CTS3、その後のデータパケットと ACK を DATA3/ACK3 と表記する。この流れを図 4.6 に示す。

### 4.3.1 RRTS

RRTS パケットは CTS と似たコントロールパケットであり、違いは CTS は通信相手のアドレスを載せていたのに対し、RRTS では自分のアドレスを添付する。もうひとつの違いは同時受信するにあたり必要な受信電力  $P_{R\text{-requierd}}$  も添付される。 $P_{R\text{-requierd}}$  については後に述べる。図 4.5 のトポロジで、 $A \rightarrow B$  の RTS1/CTS1 交換が終了した状況を考えると、C は B からの CTS1 により送信電力が制限されている可能性が高いのでデータパケットの送信はできないが、A からの RTS1 を受信していないか、受信電力が小さい場合は同時にデータを受信できる可能性が高い。このような場合、積極的に RRTS を送信し、自分宛のデータパケットを所持している周辺の端末に RTS3 の送信を要求する。D のような端末は従来 A, B の送信エリア外に存在しているため、Access Window の開始時刻を検知できなかったが、RRTS により同時通信を試みることができる。

### 4.3.2 RRTS-mode のプロトコル

図 4.5 を用いて RRTS-mode のプロトコルの流れを説明する。 $A \rightarrow B$  の RTS1/CTS1 交換と、それを受信した周辺端末の動作は mode1 と同様である。Access Window に入ったら、過去にデータパケットを受信している端末が RRTS を試みる。本論文では過去 2 秒以内にデータパケットを受信している端末に限り RRTS を送信する権利を与えることとする。RTS2 の際と同様に RRTS 送信に先立ち C はバックオフを開始する。 $BO_{RRTS}$  は次式のように表わされる。

$$BO_{RRTS} = 5 \times slottime \quad (4.7)$$

ここで RRTS の際のバックオフには 0 から 5 の値をランダムで発生させることとする。バックオフが終了した C は、同時受信の際、A からの干渉に打ち勝つために必要な受信電力  $P_{R-reqierd}$  を計算する。RTS1 を受信している場合は  $P_{post-interfere}$  を保持しているはずであり、そうでなければ 4.2.5 章と同様に推定する。 $P_{R-reqierd}$  は次式のように表わされる。

$$P_{R-reqierd} = \max\left(\frac{SINR_{th}}{P_{post-interfere}}, P_{R-TH}\right) \quad (4.8)$$

正しく受信するためには SINR と受信電力の閾値の 2 つを満たす必要があるので、2 つの最大値を選択する。C は  $P_{R-reqierd}$  を RRTS に添付し送信する。

RRTS を受信した D は送信先の C に対してのデータパケットを所持していれば次の段階に進む。RRTS の受信電力  $P_{R-RRTS}$  と式 2.6 からチャネル利得  $H_{CD}$  を推定する。RRTS に添付されている  $P_{R-reqierd}$  より、必要な送信電力  $P_{T-required}$  を次式のように計算する。

$$P_{T-required} = \frac{P_{R-required}}{H_{CD}} \quad (4.9)$$

次に、 $P_{T-required}$  が次式の条件を満たしているか確かめる必要がある。

$$P_{T-required} \leq P_{T-allowed} \quad (4.10)$$

送信電力が固定の場合、CTS1 により  $P_{T-allowed}$  が制限されていなければ、 $P_{T-allowed} = P_{max}$  である。式 4.10 を満たしている場合、RTS3 のバックオフに入る。C の RRTS に対して競合する端末が他に存在する可能性があるのでバックオフは必要である。RRTS バックオフでは 0 から 2 の値をランダムに選択する。

バックオフが終了したら D は RTS3 を C へ送信する。C は RTS3 を受信したら、

RRTS バックオフで使用した値を CTS3 に添付し送信する。CTS3 を受信した D は、mode1 の場合と同様に、Access Window が終了するまで待機し、A と同時にデータパケットを送信する必要があるので、RTS3 のバックオフと、CTS3 に添付された C の RRTS のバックオフの値から残り時間を待機する。データパケット送信からは mode1 と同様の手順に従い通信を終了する。

図 4.5 の F のように通常の RTS2/CTS2 の交換を行いたい端末は図 4.6 に示しているように、Access Window が開始してからしばらく待機する。RRTS/RTS3 の電波を検知しなければ、RTS2 を送信し、同時通信を行うことができる。図 4.6 に示すように RRTS/RTS3/CTS3 の通信予約の方が RTS2/CTS2 よりも時間的に優先されている。

### 4.3.3 RRTS , RTS3 の確率制御

$PROB_{RTS2}$  の場合と同様の理由で、 $PROB_{RRTS}$  ,  $PROB_{RTS3}$  は端末ごとに保持する。

$$PROB_{RRTS-DEFAULT}(K) = 40$$

$$\text{If Success } PROB_{RRTS}(K) = \min(90, PROB_{RRTS}(K) + 70)$$

$$\text{Else If False } PROB_{RRTS}(K) = \max(10, PROB_{RRTS}(K) - 10)$$

$$PROB_{RTS3-DEFAULT}(K) = 70$$

$$\text{If Success } PROB_{RTS3}(K) = \min(100, PROB_{RTS3}(K) + 40)$$

$$\text{Else If False } PROB_{RTS3}(K) = \max(10, PROB_{RTS3}(K) - 20)$$

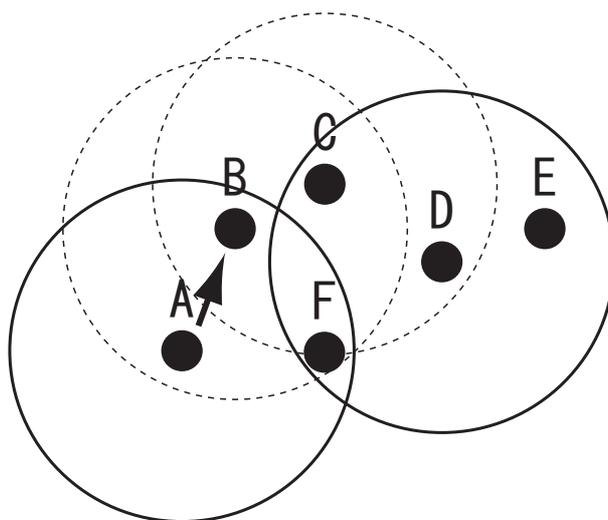


図 4.7: RRTS2-mode のトポロジー

## 4.4 RRTS2-mode

図 4.7 で  $A \rightarrow B$  の RTS1/CTS1 交換が行われ、C が RRTS を送信した状況を考える。RRTS-mode では RRTS を受信した D は C に対して送りたいデータパケットが存在する場合、RRTS に対して RTS3 を送信するように制御していた。ここで、さらに同時通信の可能性を高めるために、D は E に送りたいデータパケットを所持している場合、E に RTS を送信しても良いという変更を行った。RRTS の送信端末 C 以外の端末と交換される RTS/CTS を RTS4/CTS4 とすることとする。この手順の流れを図 4.8 に示す。C の RRTS により A、B の送信エリア外に存在する D、E が同期し同時通信を行えるよう拡張される。

RTS2 の場合と同様にして、RTS4 を受信した E は受信電力  $P_{R-RTS4}$  から次式より同時通信の可否を判断する。

$$POST\_SINR = \frac{P_{R-RTS4}}{P_{post-interfere}} \geq SINR_{TH} \quad (4.11)$$

式 4.11 を満たした場合 CTS4 を返信し、そうでなければ negative CTS4 を送信す

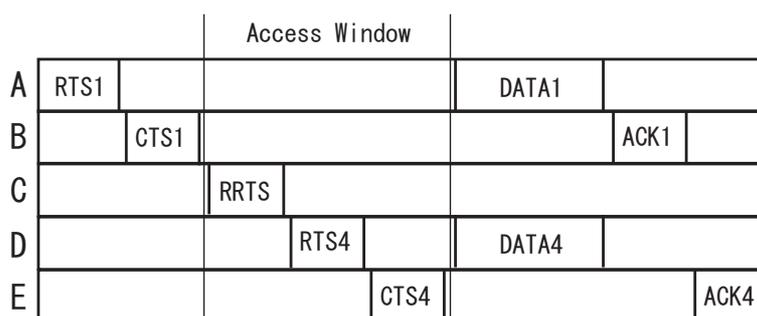


図 4.8: RRTS2-mode のプロトコル

る．その他の処理に関しては RRTS-mode と同様であり，この手法を RRTS2-mode とする．

RRTS2-mode を RRTS-mode と比較すると，同時通信可能な端末の組は増加すると考えられるが，問題点が 2 つ考えられる．1 点目は，RRTS-mode では RRTS に添付された  $P_{R\text{-requierd}}$  の値を参照すれば同時通信可能か判断できる．つまり RRTS を受信した段階で同時通信が不可能と判断した場合，RTS3 を送信しなくて済む．これにより，RTS3 パケットの無駄な送信を避けているだけでなく，RTS3 を送信したい周囲の端末の送信機会を奪うことを回避している．一方 RRTS2-mode では，RRTS を受信した段階では同時通信成功の可否は判断できず，RTS4 を受信した端末により初めて判断される．例えば図 4.7 で C の RRTS に対して D が F に RTS4 を送信した場合を考えると，F は A からの干渉が大きく同時通信不可能と判断することになる．2 点目は，RTS3 と RTS4 の競合が激しくなることが予想される．RRTS2-mode では RRTS に対して反応する端末数が増加するので，RTS3，RTS4 の衝突数が増えると考えられる．

## 4.5 TPC-model1

mode1 に送信電力制御(Transmission Power Control)を導入した手法を TPC-model1 と呼ぶこととし、以下簡単に説明する。TPC-model1 ではデータパケットの送信電力を周囲へ伝える必要があるため、RTS1/RTS2/RTS3 パケットに送信電力の情報を添付する必要がある。そのため、従来の RTS に比べパケットサイズが1バイト大きくなる。それに伴い干渉電力やマージン等の計算式が若干変更されるが、基本的な動作は mode1 と同じである。TPC-model1 の流れを図 4.3 を用いて説明する。B は RTS1 にデータの送信電力  $P_{T-DATA}$  を添付する。RTS1 を受信した A は  $P_{add-interfere}$  を次式より計算する。

$$\begin{aligned} H_{BA} &= \frac{P_{R-RTS1}}{P_{max}} \\ P_{add-interfere} &= P_{T-DATA} \frac{H_{AB}}{SINR_{TH}} \times MARGIN \end{aligned} \quad (4.12)$$

RTS1 を受信した周囲の端末 I は、将来受ける干渉電力  $P_{post-interfere}$  を次式より計算する。

$$\begin{aligned} H_{BI} &= \frac{P_{R-RTS1}}{P_{max}} \\ P_{post-interfere} &= P_{T-DATA} H_{AB} \end{aligned} \quad (4.13)$$

CTS1 を受信した端末 J は式 4.2 より送信電力を制限される。固定電力の場合と違い、送信電力が制限された場合でも Access Window 中に RTS2 の送信を試みることができる。

C は RTS2 に送信電力の上限  $P_{T-allowed}$  を添付し送信する。D は RTS2 を受信する

と同時通信可能か次式より判断する．

$$H_{CD} = \frac{P_{R-RTS2}}{P_{max}}$$

$$POST - SINR = P_{T-allowed} \frac{H_{CD}}{P_{post-interfere}} \geq SINR_{TH} \quad (4.14)$$

式 4.14 を満たしている場合，式 4.8，式 4.9 より計算した  $P_{T-required}$  を CTS2 に添付し送信する．満たしていない場合は negative CTS2 を返信する．

CTS2 を受信した C は  $P_{T-required}$  の値と  $P_{T-allowed}$  から送信電力  $P_{T-DATA2}$  を次のように決定する．

$$P_{T-DATA2} = \max(P_{T-allowed}, P_{T-required}) \quad (4.15)$$

その後の処理は model と同様である．

#### 4.5.1 $P_{T-DATA1}$ の決定方法

RTS1 を送信する端末は DATA パケットの送信電力  $P_{T-DATA1}$  を事前に決定し，RTS1 に添付しなくてはならない．送信電力は離散値をいくつか用意し，前回の通信の成否や，通信相手とのチャンネル利得により動的に変化させることとする．ある通信で DATA パケットを送信した後，ACK が返ってこない場合は DATA パケットが干渉電力により正しく受信されなかったと判断し，送信電力を 1 つ高いレベルにする．逆に，通信の際の CTS，ACK の受信電力から相手端末とのチャンネル利得が推定できるので，チャンネル環境が良好な場合は送信電力を 1 レベル下げ，周囲の同時通信を促す．

表 4.1: MAC プロトコルの比較

MAC プロトコル	性質
mode1	S2 が S1 , R1 の送信エリア内に存在
RRTS-mode	S2 , R2 のどちらかが S1 , R1 の送信エリア内に存在
RRTS2-mode	S2 が S1 , R1 の 2 ホップ以内に存在

## 4.6 TPC-RRTS-mode , TPC-RRTS2-mode

RRTS-mode , RRTS2-mode に送信電力を導入したプロトコルを TPC-RRTS-mode , TPC-RRTS2-mode と呼ぶこととする . RTS1/CTS1 の交換に関しては TPC-mode1 と同様であり , その後の処理に関しても送信電力制御を用いていない場合と同様である .

## 4.7 MAC プロトコルの比較

### 4.7.1 同時通信エリアの拡大

MAC のプロトコルの比較を表 4.1 に示す . 最初に通信を予約した送受信端末組を S1 , R1 , 2 番目に予約する送受信端末組を S2 , R2 とする . mode1 に比べ , RRTS-mode の方が同時通信できる可能性が高まり , RRTS2-mode ではさらに同時通信できる可能性が向上している .

### 4.7.2 コントロールパケットサイズ

固定電力におけるコントロールパケットサイズを表 4.2 に示す . DCF では RTS , CTS がそれぞれ 20 , 14byte である . 他の 3 手法では , CTS1 に式 4.1 の  $P_{add-interfer}$

表 4.2: 固定電力におけるコントロールパケットサイズ (byte)

MAC プロトコル	RTS1	CTS1	RRTS	RTS2	CTS2
DCF	20	14			
mode1	20	15		20	14
RRTS-mode	20	15	15	20	14
RRTS2-mode	20	15	15	20	14

を添付するために 15byte となる . RRTS には式 4.8 の  $P_{R-required}$  を添付するため 15byte となる .

表 4.3: 送信電力制御におけるコントロールパケットサイズ (byte)

MAC プロトコル	RTS1	CTS1	RRTS
TPC-mode1	21	15	
TPC-RRTS-mode	21	15	15
TPC-RRTS2-mode	21	15	15

送信電力制御を導入した場合のコントロールパケットサイズを表 4.3 に示す . RTS1 に  $P_{T-DATA}$  を添付するため 21byte となる .

### 4.7.3 送信電力

本論分では端末の送信電力は 15dBm , 14dBm , 13dBm , 12dBm , 11dBm , 10dBm , 9dBm , 8dBm , 7dBm の 9 値の離散値する . 固定電力の場合は 15dBm のみ使用可能であり , 送信エリアはおよそ 260M である . 送信電力制御を用いる場合は式 4.2 の  $P_{T-allowed}$  を決定する場合は満たしている最大の整数値を選択する . また式 4.9 の  $P_{T-required}$  を決定する場合は満たしている最小の送信電力を選択することとする .

## 第5章 シミュレーションモデル

### 5.1 シミュレーション条件

本論文では提案手法の性能を測定するために Qualnet4.0 [14] を用いて計算機シミュレーションを行った。いくつかのシミュレーションを実行するにあたり共通するシミュレーション条件を表 5.1 に示す。

### 5.2 定義

#### 5.2.1 スループット

本論分で扱うスループットとは全てのフローの end to end スループットの合計値を指すものとする。

#### 5.2.2 同時通信成功数

mode1 , RRTS-mode , RRTS2-mode では同時通信の可能性を向上させるための MAC プロトコルであるので , シミュレーションを通して同時通信成功数をそれぞれカウントする。同時通信の種類は以下の 3 つである。

- DATA2-count:mode1 のように RTS1/CTS1 の後 , RTS2/CTS2 によって予約された同時通信。

表 5.1: 共通のシミュレーション条件

ルーティングプロトコル	AODV
リンクレート	2Mbps
送信レンジ	260m
キャリアセンスレンジ	470m
パケットサイズ	1024Bytes
SINR-TH	6dB
$P_{R-TH}$	-81dBm
$P_{SENSE-TH}$	-91dBm

- DATA3-count:RRTS-mode のように , RTS1/CTS1 の後 , RRTS/RTS3/CTS3 に よって予約された同時通信 .
- DATA4-count:RRTS2-mode のように , RTS1/CTS1 の後 , RRTS/RTS4/CTS4 に よって予約された同時通信 .

同時通信成功数は予約された 2 つの同時通信が両方とも正しく受信され , ACK が返ってきた場合にカウントすることとする . 2 つのデータパケットのうち片方 , もしくは両方が正しく受信されなかった場合にはカウントしない .

### 5.2.3 コントロールパケット効率

コントロールパケット効率  $\eta$  を次式のように定義する .

$$\eta = \frac{\text{RTS, CTS, RRTS の送信数の合計}}{\text{正しく受信されたデータパケット数}} \quad (5.1)$$

5.1 より , DCF の場合 RTS/CTS で予約が行われ , データパケットが送信されるので , 理想的には効率  $\eta = 2$  となる . しかし , RTS や CTS の衝突により , 2 より少し大きい値になることが予想される . mode1 の場合 , RTS1/CTS1 , RTS2/CTS2 の予約により , データパケットが 2 つ送信されるので , 理想的には効率  $\eta = 2$  となる .



図 5.1: フロー A → C

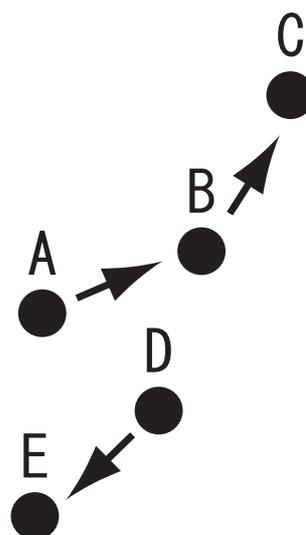


図 5.2: フロー A → C , D → E

ただし、DCFの場合と同様にコントロールパケットの衝突による劣化、さらには RTS2/CTS2 が交換されても同時通信不可能と判断された場合は無駄になってしまうので、効率  $\eta$  は DCF より劣化すると考えられる。RRTS-mode, RRTS2-mode では同時通信を行うために RRTS パケットを新たに導入している。RRTS による同時通信が行われた場合、理想的には RTS1/CTS1, RRTS/RTS3/CTS3 により 2 つのデータパケットが送信されるので  $\eta = 2.5$  となる。RRTS が送信されない場合や通常の同時通信が行われる場合、1 つのデータパケットしか送信されない場合等が考えられるが、RRTS を導入したために mode1 よりも劣化すると考えられる。

### 5.3 マルチホップの影響

マルチホップが与える同時通信への影響に図 5.1 を例に考えてみる。各端末は 1 つのアンテナしか所持していないため、送信と受信を同時に行うことはできない。そのため、 $A \rightarrow B$  と  $B \rightarrow C$  の同時通信が行われることはない。つまり、あるマルチホップのフロー内で同時通信が行われる可能性は低いことが考えられる。

次に図 5.2 を考えてみる．A から C のフローは B を経由して 2 ホップである． $B \rightarrow C$  と  $D \rightarrow F$  が同時通信できることとする．B にパケットが貯まっている場合は同時通信が実現できるが，B がパケットを所持していない場合，同時通信を行うことができない．

## 5.4 端末の移動の影響

Access Window 中のアクセスは確率制御により行われる．過去の同時通信の成否により確率を動的に制御する．そのため，端末が移動して同時通信不可能な位置関係になった場合，しばらくは Access Window 中にアクセスを試みるものの，同時通信できない状況になることが考えられる．逆に端末の移動により新たに同時通信できる位置関係になった場合，同時通信を行うまでにしばらく時間がかかることが考えられる．

## 第6章 シミュレーション結果

### 6.1 隠れ端末の同時受信

まず，図 6.1 について DCF と RRTS-mode の 2 つについて比較を行う．4 つの端末は一直線に並んでおり，A-B，B-D，D-C 間の距離は 200m である． $A \rightarrow B$ ， $C \rightarrow D$  のフローを発生させ，シミュレーション時間は 10 秒とする．スループットと RREQ 送信回数を表 6.1 に示す．スループットに関しては，RRTS-mode の方が 66% 程度高い値になっている．DCF の場合は NAV によりすべての時刻に A，C どちらか一方のデータパケット送信しか行われぬ．一方 RRTS-mode の場合は，RRTS によってうまく二つの通信が予約できた場合は A，C が同時に送信でき，その分だけスループットが向上しているからである．

また RREQ の送信回数については，DCF の場合は RRTS-mode と比べ 10 倍程大きい値となっている．10 秒間で 213 回，1 秒間で 21.3 回も RREQ が送信されている．これは，例えば  $A \rightarrow B$  の RTS1/CTS1 交換が終了した後，D は B の CTS により NAV を設定し全ての送信を控える．その間に C が RTS を送信した場合でも D は CTS を返信せず無視し続ける．通常ロングパケット再送回数の閾値は 4 回がデフォルト値として設定されており，4 回連続で RTS を送信したのに CTS が返ってこなかった場合，リンクが切断したと判断し RREQ を送信する．これは DCF における仮想キャリアセンスの誤機能によるリンク切断の誤検出として知られている [8]．実際通信エリア内に存在するにもかかわらず，リンク切断の誤検出により大

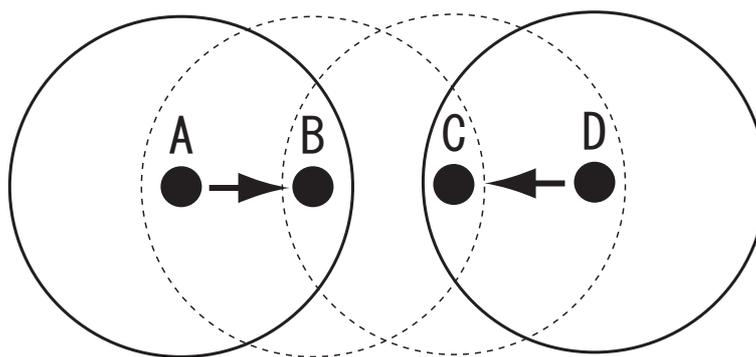


図 6.1: RRTS による隠れ端末の同時受信

量の RREQ が送信されてしまっている。

一方 RRTS モードの場合は 1 秒間で 3.6 回の RREQ が送信されており、決して小さい値ではないが、DCF と比べると約 10 倍少ない値である。C の RRTS により、C は RTS2/CTS2 の交換が行える。もしも D に送りたいパケットがない場合でも、C は D の RRTS によって、 $A \rightarrow B$  のデータパケット、ACK の送信が終了するまで、待機することができる。

以上のように、RRTS は RTS1/CTS1 を直接受信できなかった端末がローカルに同期して同時通信を行えるように手助けをしているだけでなく、RTS1/CTS1 を送信した端末から 2 ホップ以上離れている端末に先行している通信の存在を伝え、無駄な RTS の送信を控えさせるという効果もある。

文献 [8] ではロングパケット再送回数の閾値をデフォルトの 4 回より大きな値に設定することにより伝送特性が改善されることが述べられている。本論分では以降、全ての MAC プロトコルに関して再送回数の閾値を 10 回としてシミュレーションを行うこととする。

表 6.1: 隠れ端末トポロジーでのシミュレーション結果

	DCF	RRTS
スループット (Mbps)	1.34	2.24
RREQ 送信回数	213	36

表 6.2: 400 × 400 25 端末におけるシミュレーション条件

アプリケーション	CBR
エリア	400 × 400
端末数	25
フロー数	4,6,8,10
モビリティ	静止

## 6.2 400 × 400 25 端末 固定電力でのシミュレーション

シミュレーション条件を表 6.2 に示す。400 × 400 のエリアを 25 個の正方形に分割し、正方形内に 1 つの端末をランダムに配置する。フローは 1 ホップ内の通信組をランダムに選択する。

まず 4 フローの場合のスループット結果を図 6.2 に示す。横軸が CBR のパケットの発生間隔であり、縦軸は各フローのスループットの合計値である。発生間隔が 30MS ではすべてのパケットの送信が終了して 4 手法の差は見られない。20MS では同時通信するには十分なパケットが各端末にたまっていないため、Access Window のロスの分だけ DCF 以外のプロトコルはスループットが低下している。10MS、5MS の場合でも DCF に関してはスループットの改善は見られない。これは DCF では同時に 1 組しか通信できないため、限界スループットに達しているためである。mode1 では同時通信できる場合が存在するため、DCF に比べスループットが約 3.7% 程向上している。RRTS-mode ではさらに同時通信できる可能性が高まるため DCF に比べ 8.8% 程の改善が見られる。RRTS2-mode は RRTS の反応して RTS3、RTS4 の送信を試みる端末同士が競合することと、エリアが狭いために DATA4 の

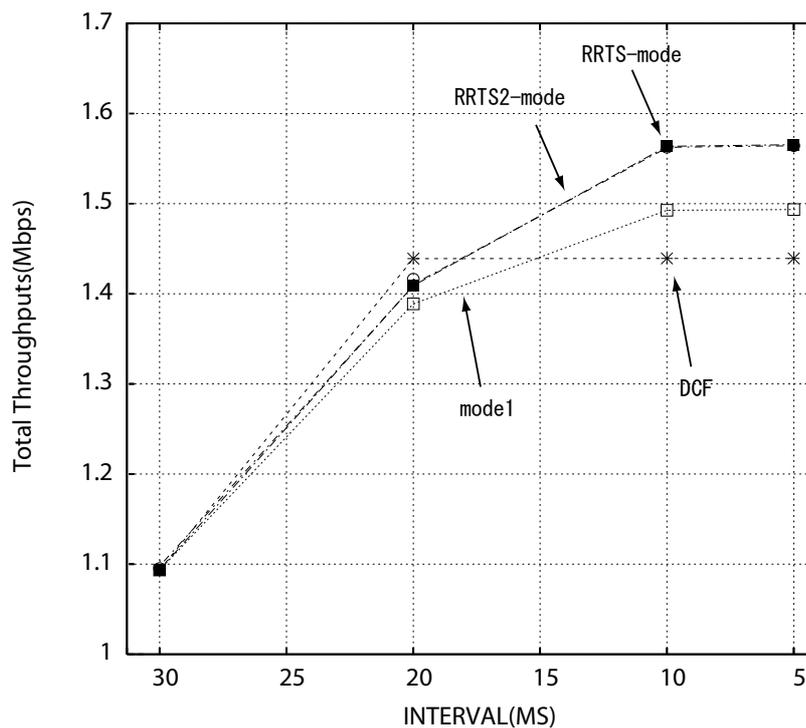


図 6.2: 400 × 400 25 端末 4flow におけるスループット結果

同時通信ができる位置関係に端末が存在する確率が低いいため、RRTS-mode に比べほぼ同じスループット値になっていると考えられる。

次に同時通信成功数を表 6.3 に示す。横軸にパケット発生間隔をとり、縦軸は同時通信成功回数 (回/S) である。各発生間隔における同時通信成功回数を 3 つの MAC プロトコルについてそれぞれ表している。3 手法ともパケット発生間隔が短くなるにつれ同時通信数が増加している。mode1 と RRTS-mode を比較すると、DATA2 の成功数は減少しているが、合計値では増加しており、スループット向上につながっている。RRTS2-mode では DATA3 と DATA4 の合計が RRTS-mode の DATA3 の値とほぼ等しくなっているため、RRTS に対する競合が起こっていると考えられる。

次にコントロールパケット効率を図 6.4 に示す。DCF のコントロールパケット効率は 2 よりわずかに大きい値である。これは、RTS、CTS の衝突により RTS の再送

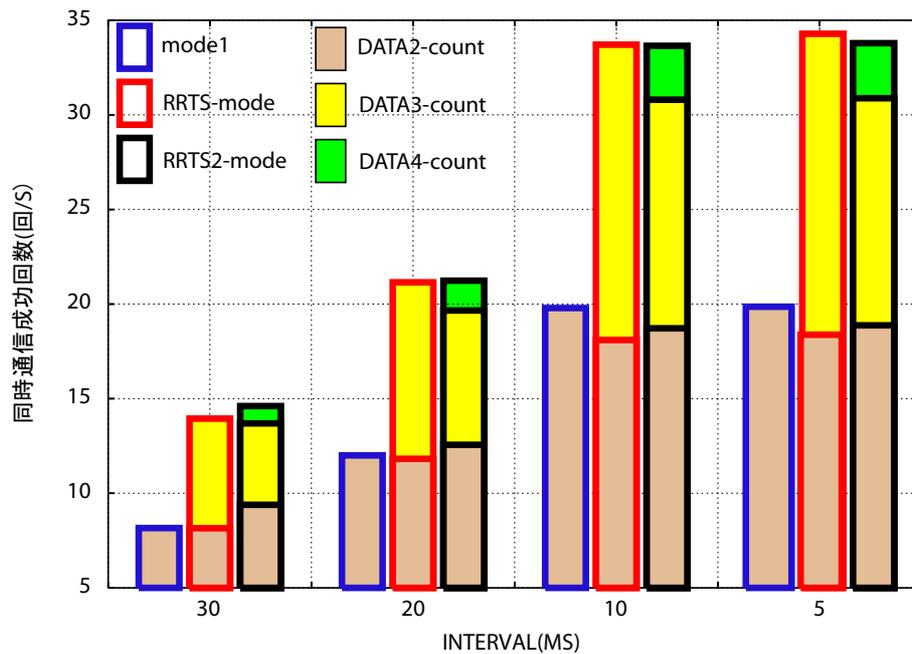


図 6.3: 400 × 400 25 端末 4flow における同時通信成功回数

が必要な場合があるためであると考えられる．mode1 では同時通信が行われた場合は効率 2 であるが，同時通信できない場合でも RTS2/CTS2 の交換が行われる場合があり，DCF に比べて約 6% 効率が劣化している．RRTS-mode では DATA3 の同時通信が行われた場合の効率は 2.5 であるので，さらに効率は悪くなっている．RRTS2-mode では，RRTS に対する競合の激化と，RTS4 は送信した段階では同時通信の可否が判断できないため，無駄になってしまう可能性があるため，RRTS-mode に対してさらに劣化しており，1 つの packet 送信のために平均して 2.4 個程度のコントロール packet が送信されている．

次に 6 フローの場合のスループット結果を図 6.5 に示す．フロー数の増加により，同時通信が可能である状況が多くなることが予想される．DCF に関しては限界値に到達してフローを引いているため 4 フローの場合とほぼ同じ値となっている．一方 mode1 は発生間隔が短くなるにつれてスループットが上昇し，10MS では DCF と比べ 12.6 % 程向上している．RRTS-mode に関してはさらに改善され

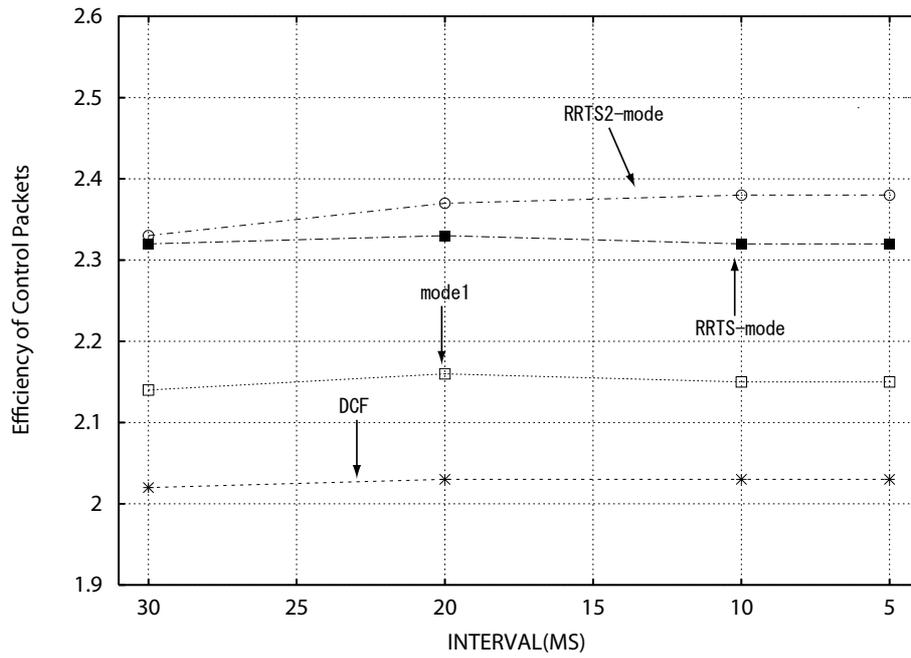


図 6.4: 400 × 400 25 端末 4flow におけるコントロールパケット効率

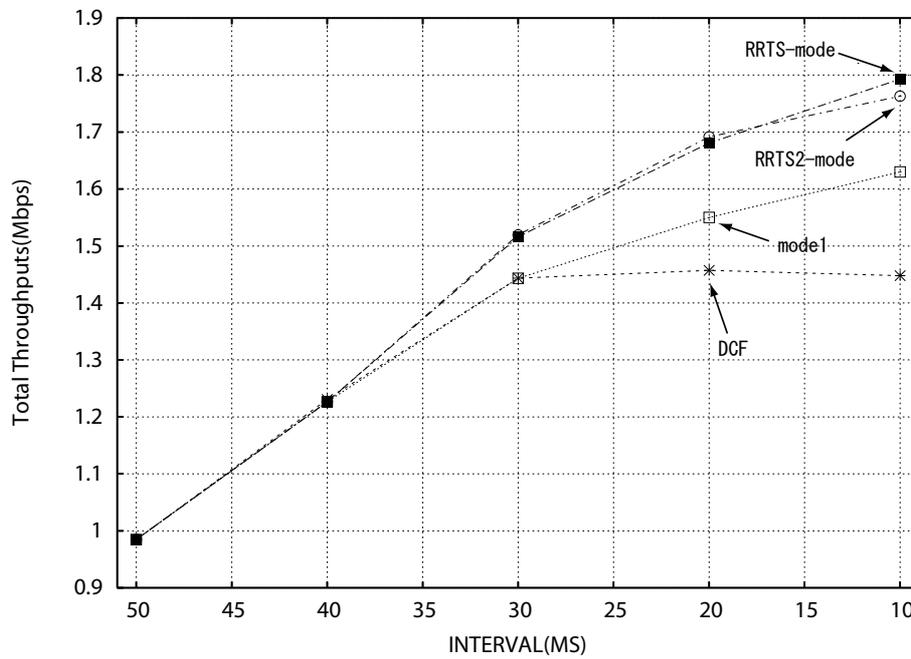


図 6.5: 400 × 400 25 端末 6flow におけるスループット

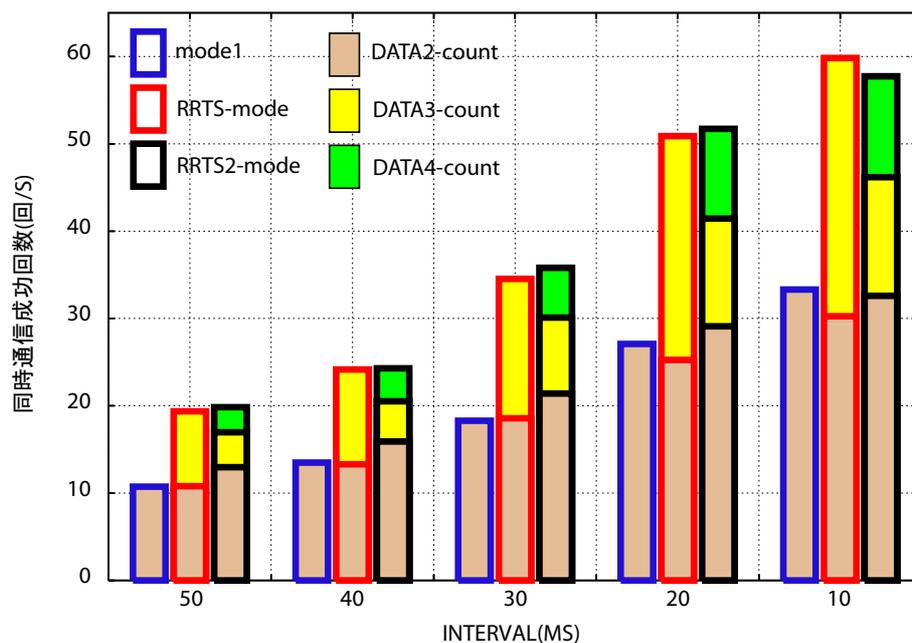


図 6.6: 400 × 400 25 端末 6flow における同時通信成功数

ていて、DCF に比べて 23.8 % 程度高い値となっている。

次に 6 フローの場合の同時通信成功回数を図 6.6 に示す。mode1 ではパケット発生間隔が短くなるにつれて DATA2 の成功数が増加している。RRTS-mode では、DATA2 の成功回数は mode1 の回数を保ち、DATA3 の成功数が加わり、合計数では 2 倍程度多くの成功数を達成している。RRTS2-mode ではほぼ RRTS-mode と同じ値であるが、RTS3、RTS4 の競合から同時通信成功回数が減少している。

6 フローの場合のコントロールパケット効率を図 6.7 に示す、DCF では 4 フローの場合と比べてわずかに劣化している。これはフロー数の増加により競合が激しくなり、コントロールパケットの衝突が増えたためと考えられる。mode1 に関しても同様の理由で劣化が見られる、RRTS-mode、RRTS2-mode では競合の激化に加え、DATA3 の送信数が増加していることから、RRTS の送信数の増加が理由に挙げられる。

同様にして 8 フロー、10 フローにおけるスループット結果を図 6.8、図 6.11 に、

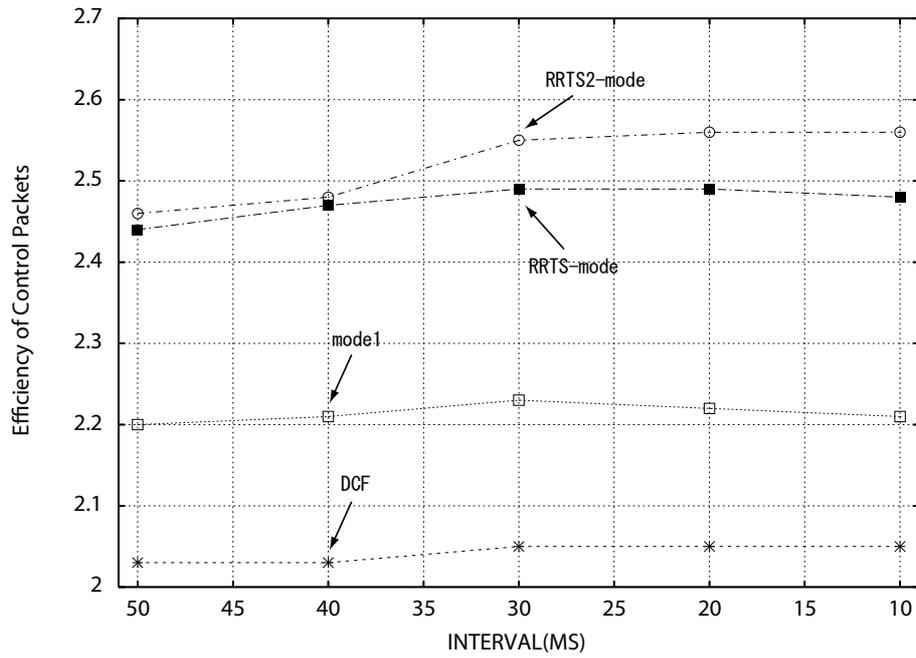


図 6.7: 400 × 400 25 端末 6flow におけるコントロールパケット効率

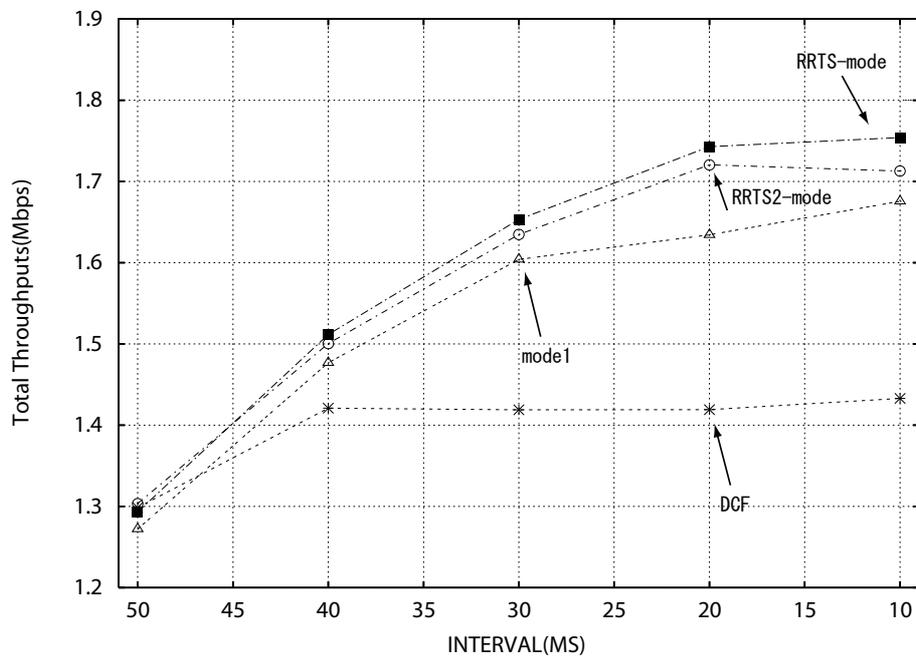


図 6.8: 400 × 400 25 端末 8flow におけるスループット

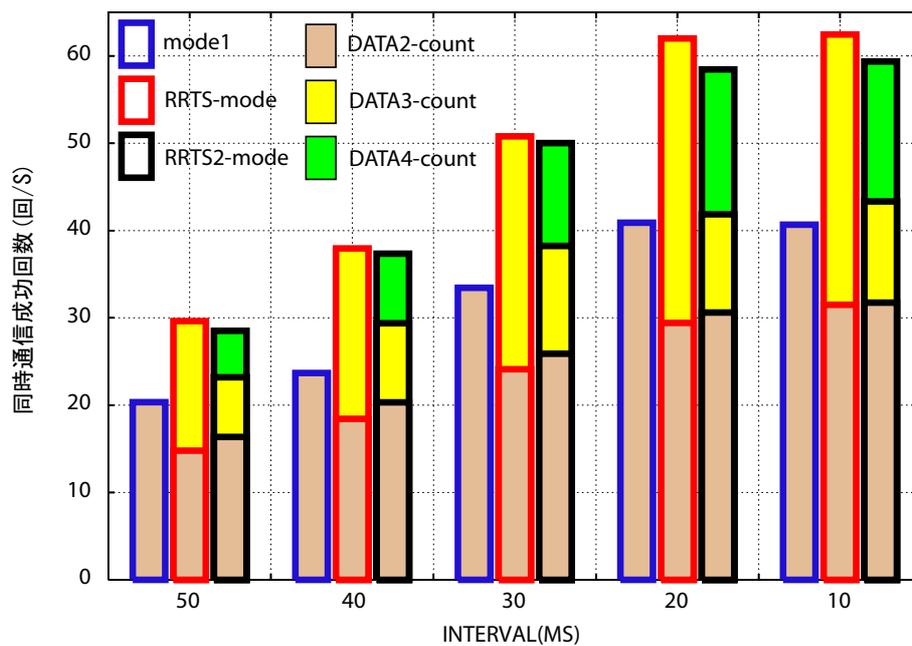


図 6.9: 400 × 400 25 端末 8flow における同時通信成功数

同時通信成功回数を図 6.9, 図 6.12 に, コントロールパケット効率を図 6.10, 図 6.13 にそれぞれ示す.

8 フローのスループットに関しては DCF の場合, 6 フローの値よりわずかに劣化している. これはフロー数の増加により競合が激しくなり, RTS や DATA の衝突が多く発生しているためであると考えられる. 他の 3 手法に関しては, 発生間隔が 30MS までの場合は同時通信できる可能性が高いため, 6 フローの場合よりスループットが向上しているが, 20MS, 10MS となるにつれて, 競合が激しくなるために 6 フローの場合よりも劣化している.

10 フローの場合のスループットに関しては, DCF の場合, さらなる競合の激化により, 8 フローの場合よりさらに劣化している. 他の 3 手法に関しては発生間隔が 50MS, 40MS, 30MS の場合, 6 フロー, 8 フローのスループット値と比較し改善が見られるが, 20MS, 10MS では競合の激化によりフローを引いている.

8 フローの場合の同時通信成功数については, mode1 に関しては 6 フローの場

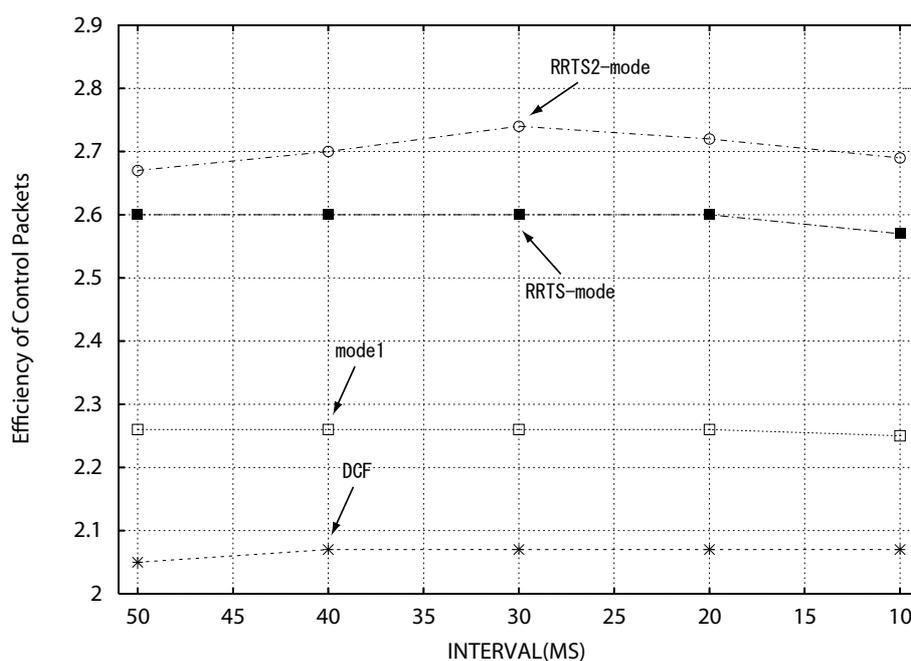


図 6.10: 400 × 400 25 端末 8flow におけるコントロールパケット効率

合と比較して各発生間隔に関して向上が見られる．原因としては6フローの場合から同時通信ができるフローの組が増加したことが挙げられる．一方 RRTS-mode では，発生間隔が長い状況では改善が見られるが，発生間隔が短くなると6フローの場合とほとんど変わらなくなっている．これは Access Window で競合が発生しており，これ以上の同時通信の増加は期待できないことを表している．RRTS2-mode でも同様のことが考えられ，さらに RTS3，RTS4 の競合により RRTS-mode よりわずかに劣化している．

10フローの場合は，mode1 に関しては8フローの場合と比較してほぼ同じ値となっている．一方 RRTS-mode，RRTS2-mode でも8フローの場合と比較して同じか，わずかに減少している．これは上述したことと同様に同時通信の飽和状態にあるからと考えられる．

8フローにおける DCF のコントロールパケット効率は競合の激化により6フローの場合と比べてわずかに劣化している．同様の理由で他の3手法とも6フローの

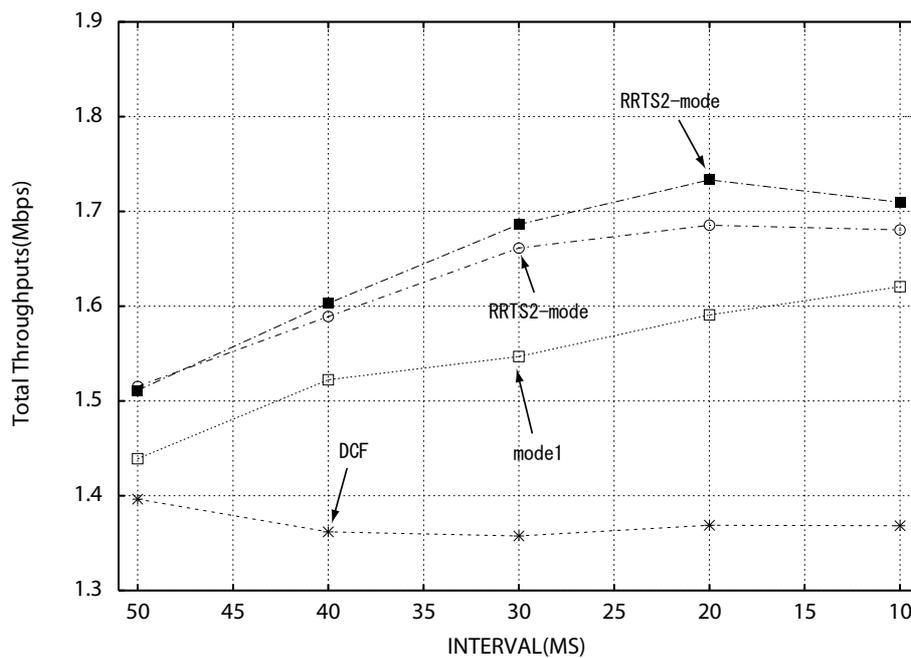


図 6.11: 400 × 400 25 端末 10flow におけるスループット

場合より効率が劣化している．10フローの場合はさらに競合が激化するため，4手法とも8フローの場合に比べて劣化している．

### 6.2.1 400 × 400 25 端末におけるシミュレーションのまとめ

4, 6, 8, 10フローについてシミュレーションを行った結果，DCFに関してはある程度トラフィックが高くなると飽和状態になり，スループットは一定の値を保っていた．さらにフロー数の増加，パケットの発生間隔の減少によりトラフィックが激しくなるとRTSの競合が起こり，スループットはわずかに劣化した．

mode1に関してはパケットの発生間隔が短くなるにつれて同時通信成功数が増加した．これは，常にパケットを所持している端末が増えるため，Access Window中にRTS2の送信を試みる端末が増えるためであると考えられる．また，フロー数の増加により，同時通信可能となる送受信端末組が存在する確率が高くなるの

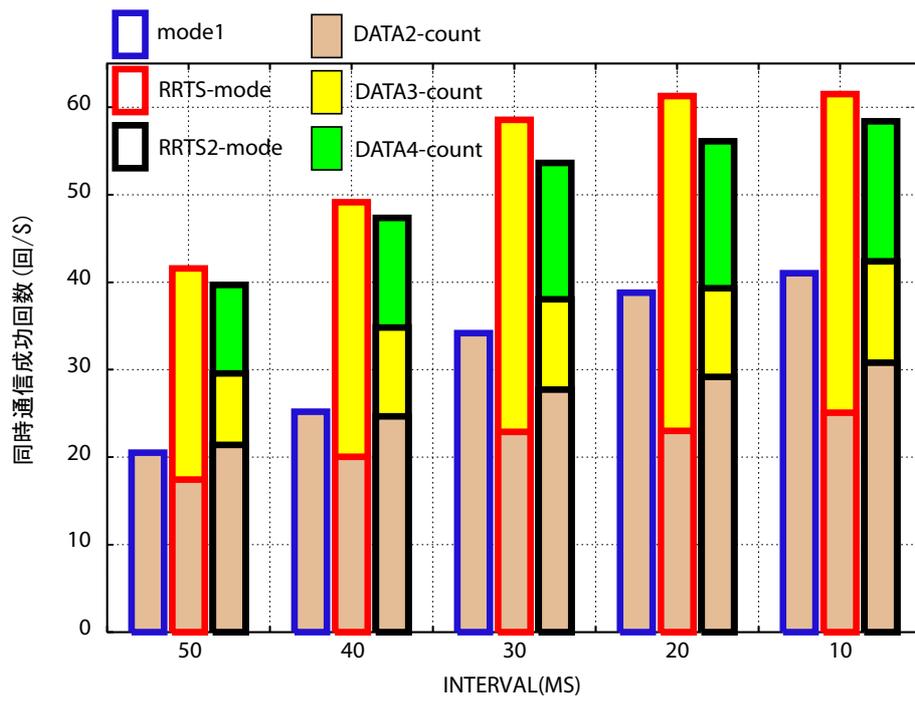


図 6.12: 400 × 400 25 端末 10flow における同時通信成功数

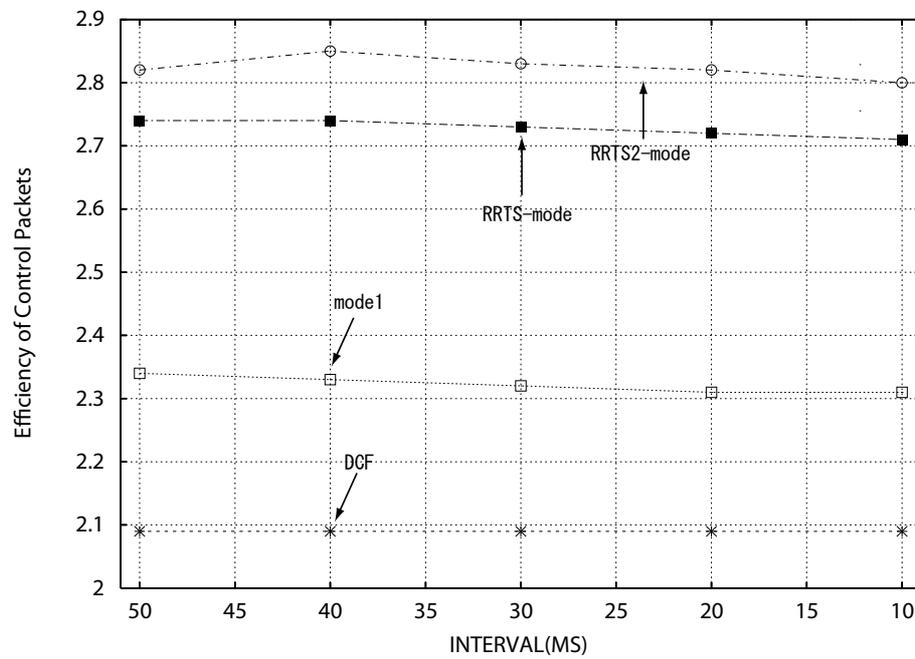


図 6.13: 400 × 400 25 端末 10flow におけるコントロールパケット効率

で、同時通信数が増加し、スループットが向上した。10 フロー、パケット発生間隔 10MS の場合でもスループットは上昇を続けており、飽和状態には至っていないと考えられる。コントロールパケット効率に関しては、理想的には2であるが、Access Window 中に RTS2 が送信された場合、受信端末が negative CTS2 を返信した場合に無駄が発生する。そのため DCF と比較して劣化しており、フロー数の増加により効率は増加しているが、10 フロー、10MS の場合、2.3 個程度のコントロールパケットで1つのデータパケットを送信しており、スループットに関しては DCF に対して約 18.4% 程高い値を達成している。

RRTS-mode では RRTS の効果により同時通信を試みる端末が増加する。そのため、フロー数が少ない場合でも多くの同時通信を行うことが可能となっている。一般的に DATA2 の同時通信は送信端末同士が近い場合に成功する可能性が高く、DATA3 の同時通信は受信端末同士が近い場合に成功しやすいため、2 種類の同時通信はほとんど競合しないと考えられる。そのため、RRTS-mode では mode1 の DATA2 同時通信回数を保ちつつ、DATA3 の同時通信回数が上乘せされるため、スループットで大きく改善していると考えられる。そのため、全ての状況において mode1 のスループット値を上回っている。フロー数の増加とパケット発生間隔の減少により同時通信数は加していくが、ある程度のトラフィック量に達すると、mode1 と異なり飽和状態を迎えている。スループットに関しても同様のことが言える。mode1 と比較して常に高いスループットを達成しているが、トラフィック量が高くなるとフローを引いている。コントロールパケット効率に関しては、RRTS パケットを用いていることにより、mode1 に比べ劣化している。トラフィックが激しくなると、RRTS 同士の衝突や、RTS3 同士の衝突が増加し、効率は大きく劣化していて、10 フロー、10MS では1つのデータパケットを送信するために約 2.7 個のコントロールパケットを送信している。

RRTS2-mode では RRTS-mode と同じような特性を示しているが，RRTS を受信した後，RTS3 を送信したい端末と，RTS4 を送信したい端末の競合が発生している．そのため，RRTS-mode の DATA3 の成功数より RRTS2-mode の DATA3 と DATA4 の成功数の和が少なくなっている．また，400 × 400 のエリアでは狭すぎるため DATA4 の同時通信を行える状況になることは少ないと考えられる．コントロールパケット効率に関しては RRTS-mode の場合に加え，RTS3，RTS4 の競合が起こるので，さらに劣化している．

### 6.3 400 × 400 25 端末における

#### 送信電力制御を用いたシミュレーション

固定電力の場合と同様の環境，シナリオで送信電力制御を導入した場合のシミュレーションを行った．4 フローの場合のスループット結果を図 6.14 に示す．固定電力の場合と比較して 3 手法ともスループットが向上している．これは送信電力制御により 4 フローの場合でも適度に送信電力を調節することにより同時通信を可能にしているためであると考えられる．

次に同時通信成功数を図 6.15 に示す．3 手法とも固定電力の場合と比較して大幅に増加している．特に TPC-RRTS-mode，TPC-RRTS2-mode では固定電力の場合と比べて約 2 倍の同時通信が行われている．

次にコントロールパケット効率を図 6.16 に示す．同時通信が多く行われているため，TPC-RRTS-mode，TPC-RRTS2-mode では固定電力の場合と比較して約 20% 効率が劣化している．

同様にして 6 フロー，8 フローにおけるスループット結果を図 6.17，図 6.20 に，同時通信成功回数を図 6.18，図 6.21 に，コントロールパケット効率を図 6.19，図

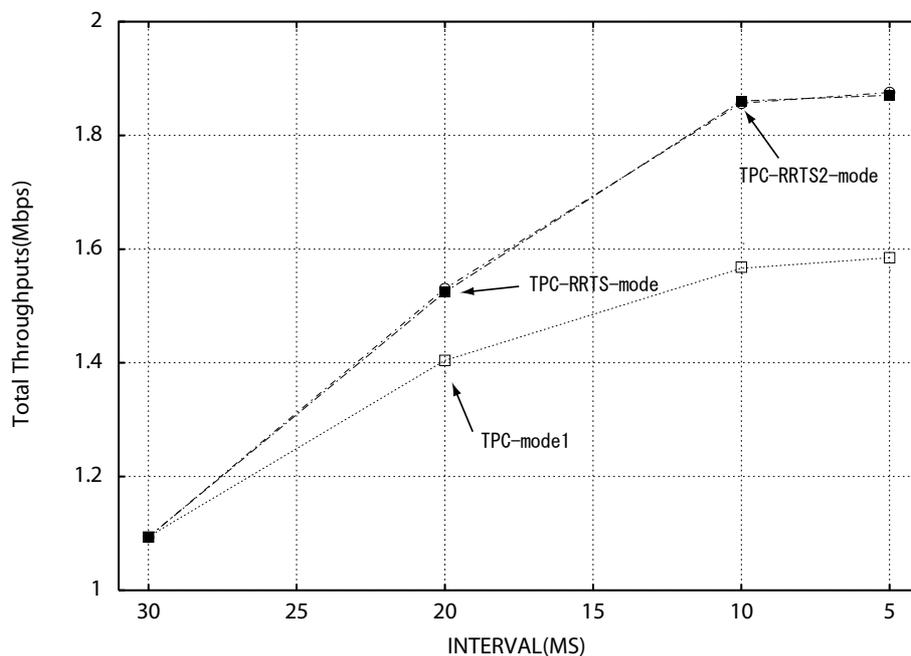


図 6.14: 400 × 400 25 端末 4flow 送信電力制御を用いた場合のスループット

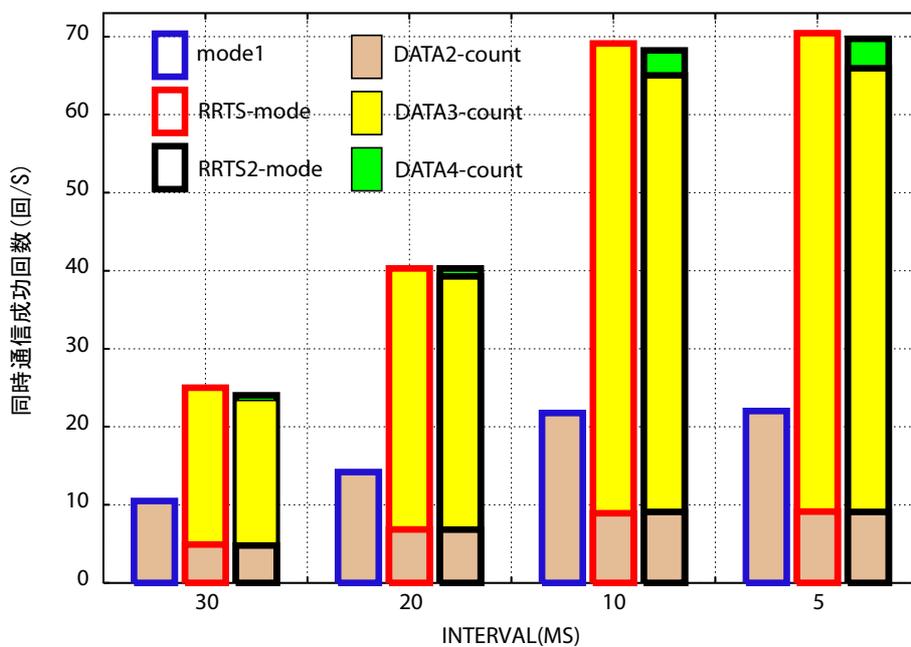


図 6.15: 400 × 400 25 端末 4flow 送信電力制御を用いた場合の同時通信成功数

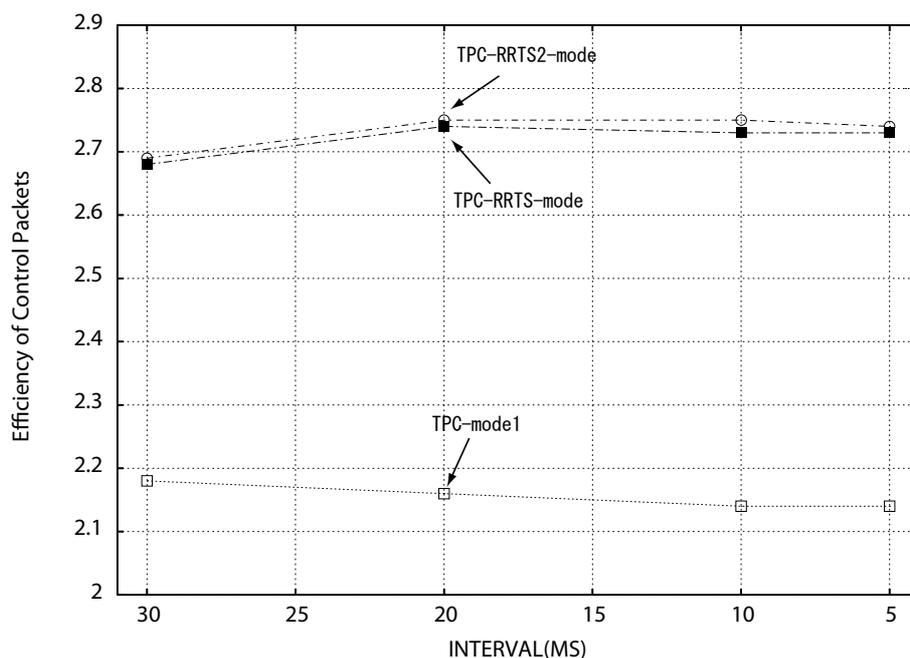


図 6.16: 400 × 400 4flow 送信電力制御を用いた場合のコントロールパケット効率

6.22 にそれぞれ示す。

スループットに関しては3手法とも6フロー，8フローになるに従い向上している。しかし，パケット発生間隔が20MS，10MSになるにつれて，スループット改善の度合いは小さくなり，飽和状態に近づいていると考えられる。

### 6.3.1 400 × 400 送信電力制御を用いたシミュレーションのまとめ

送信電力制御を導入することにより，あらゆる状況において同時通信成功回数が増加している。これは，図3.1に示したように送信電力をお互いが適度に調節することにより同時通信を可能にしているためである。同時通信成功回数の増加にともないスループットも向上している。同時通信成功回数の増加により，Access Window中に多くのコントロールパケットの交換が行われるために，コントロールパケット効率が劣化している。

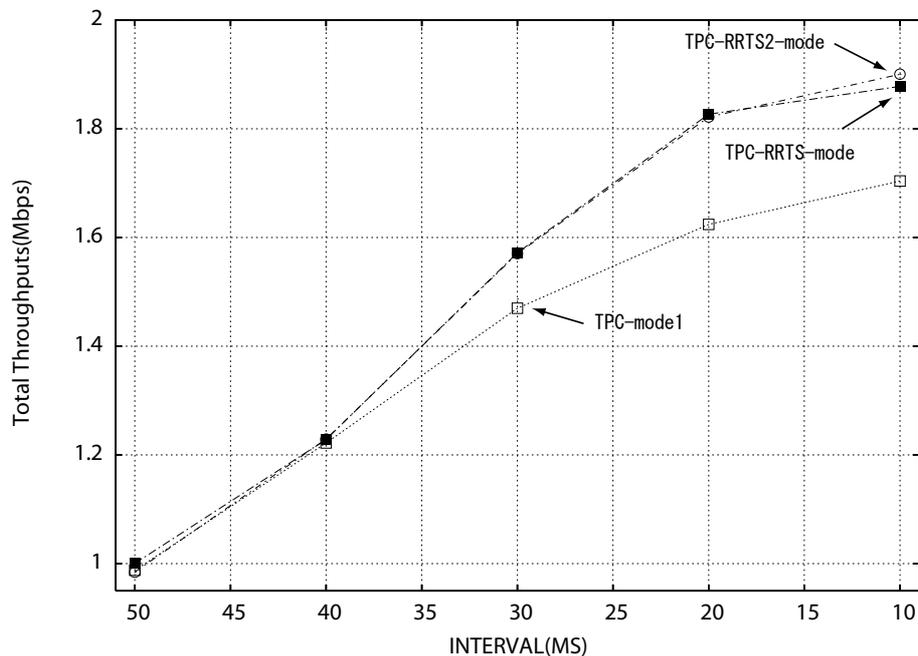


図 6.17: 400 × 400 6flow 送信電力制御を用いた場合のスループット

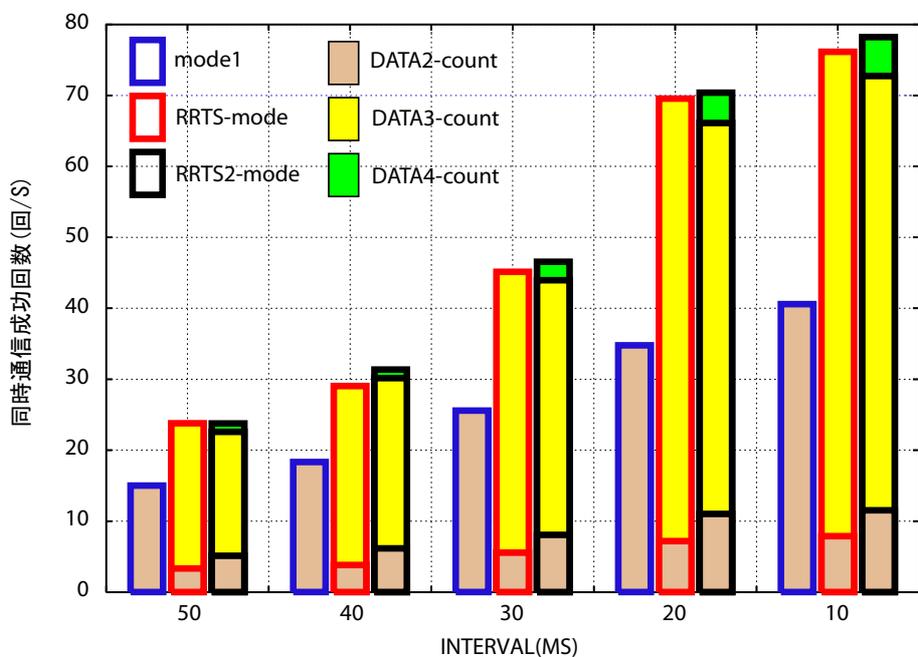


図 6.18: 400 × 400 6flow 送信電力制御を用いた場合の同時通信成功数

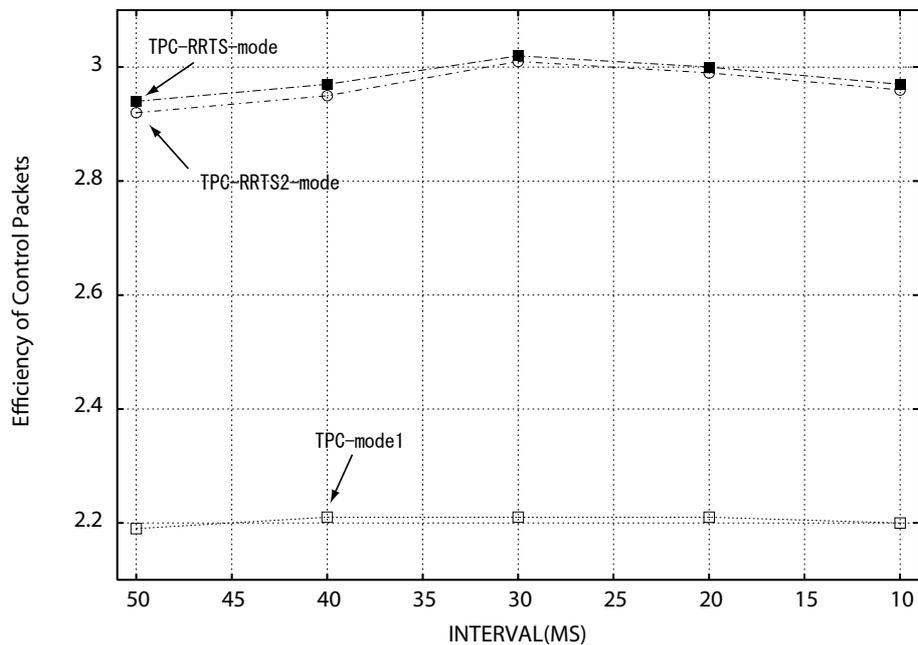


図 6.19: 400 × 400 送信電力制御を用いた場合のコントロールパケット効率

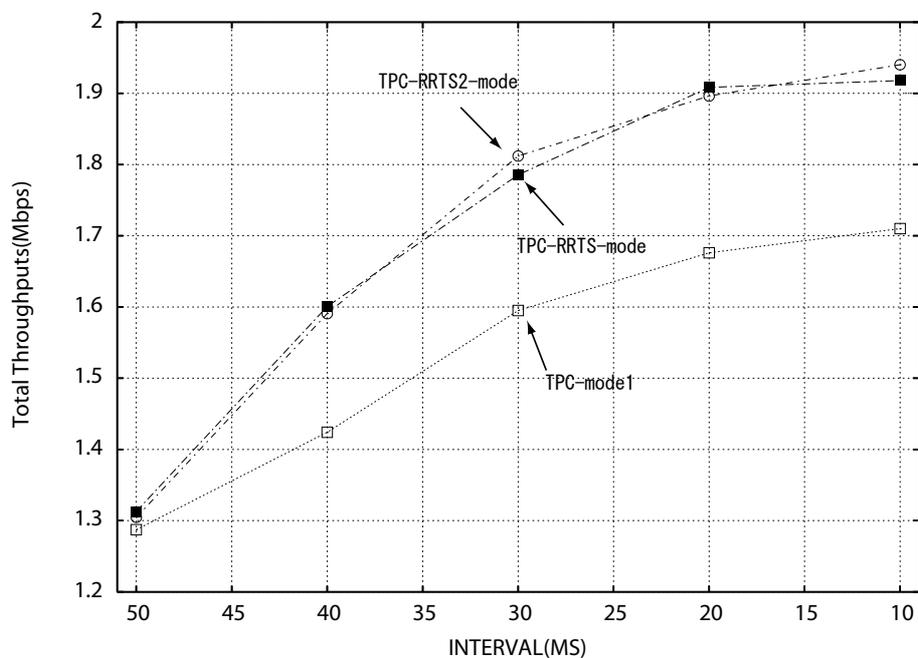


図 6.20: 400 × 400 8flow 送信電力制御を用いた場合のスループット

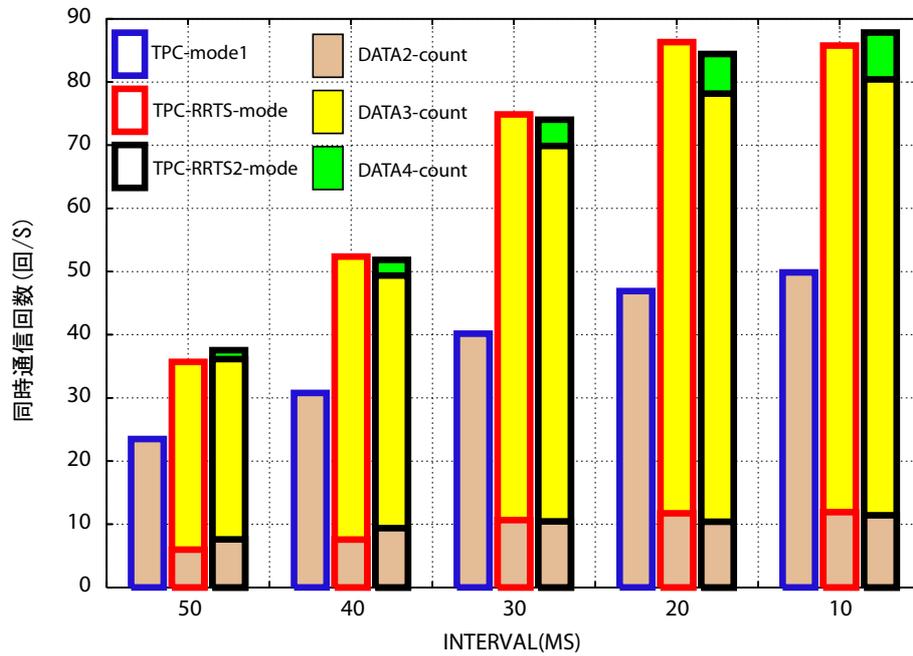


図 6.21: 400 × 400 8flow 送信電力制御を用いた場合の同時通信成功数

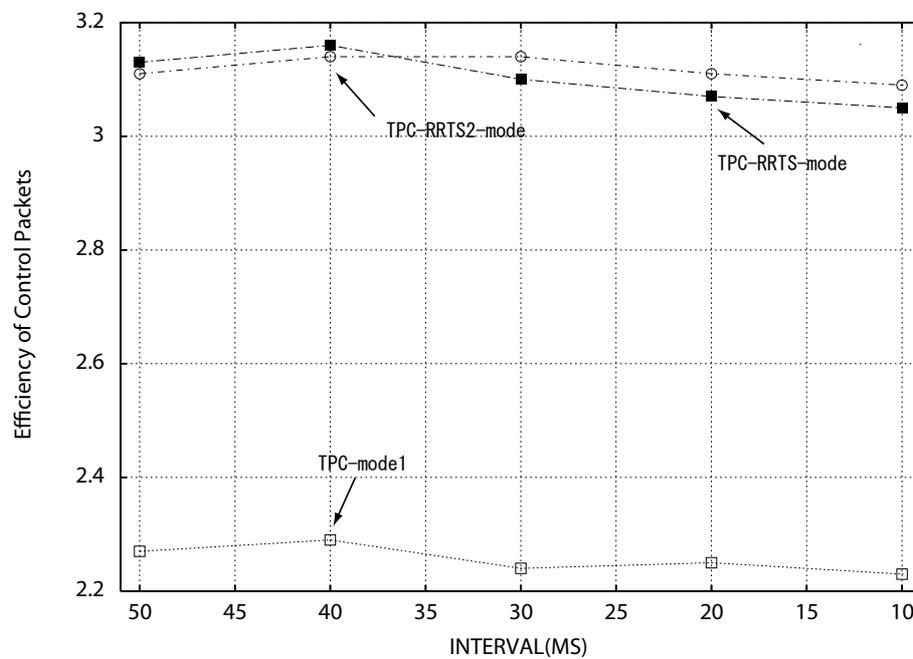


図 6.22: 400 × 400 送信電力制御を用いた場合のコントロールパケット効率

TPC-RRTS-mode では DATA3 に比べて DATA2 の増加が少ない。これは図 4.6 に示しているように、DATA3 の方が優先されているためである。固定電力の場合は DATA3 と DATA2 の値が同程度であったが、送信電力制御により、DATA3 が DATA2 の送信機会を奪う状況が生まれていると考えられる。DATA2 の同時通信ができる状況でも先に RRTS/RTS3/RTS4 により予約されてしまった場合、DATA2 の同時通信の代わりに DATA3 の同時通信が行われる。そのため、mode1 と RRTS-mode を比較して DATA2 の同時通信成功回数は減少しているが、DATA3 の同時通信が加わり、同時通信成功回数の合計値では常に大きく上回っている。

TPC-RRTS2-mode では送信電力制御を用いることにより DATA3 と DATA4 の競合がさらに発生すると考えられる。RTS3、RTS4 の送信判断は 4.3 章で述べたように電力計算と確率制御によって行われているが、成功した場合確率を増加し、失敗したら減少させているため、一度成功した端末が常に Access Window 中の競合に勝ち続ける可能性が高い。そのため、DATA3 の同時通信が DATA4 の同時通信機会を奪っていると考えられる。

## 6.4 600×600 36 端末におけるシミュレーション

600 × 600 のエリアに 36 端末をランダムに配置し、フローをランダムで選択する。先ほどのシミュレーションでは 1 ホップに限定したが、今回は 1 ホップ、2 ホップ、3 ホップのフローが混在する。マルチホップのフローが存在する場合、5.3 章で述べたように同時通信成功回数が減少することが考えられる。

6 フローの場合のシミュレーション結果を図 6.23 に示す。パケット発生間隔が長い場合は、DCF に比べて他の 3 手法のスループットが低い値になっている。マルチホップの影響と、低いトラフィックによりほとんど同時通信が成功していない

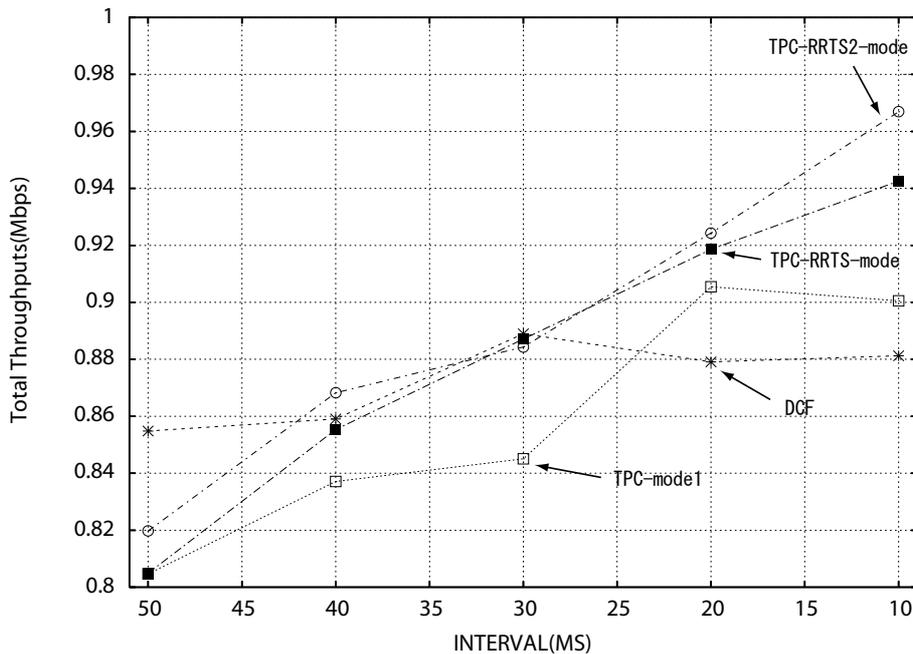


図 6.23: 600 × 600 6flow におけるスループット

からであると考えられる。トラフィックが高くなるにつれて、DCF は飽和状態に達しているが、他の 3 手法はスループット値が向上している。パケット発生間隔 10MS では、DCF に対して TPC-mode1、TPC-RRTS-mode、TPC-RRTS2-mode はそれぞれ 2.2%、7.0%、11.0% 向上している。次に同時通信成功回数を図 6.24 に示す。TPC-mode1 はトラフィックが増えるにつれて同時通信数が増加しており、パケット発生間隔が 10MS の場合では平均して 1 秒間に 20 回の同時通信が行われている。TPC-RRTS-mode では mode1 と比較して DATA2 の同時通信は減少しているが、DATA3 の同時通信が加わり、合計値で TPC-mode1 の 2 倍近くの同時通信が行われている。TPC-RRTS2-mode では TPC-RRTS-mode と比べて DATA2 の同時通信の回数はほぼ同じであり、DATA3 の回数は減少しているが DATA4 が追加されて合計値では上回っている。これは、エリアが拡大したことにより、DATA4 を送信する機会が増えたためであると考えられる。

エリアを拡大した影響でマルチホップのフローが増加した。そのため、中継端

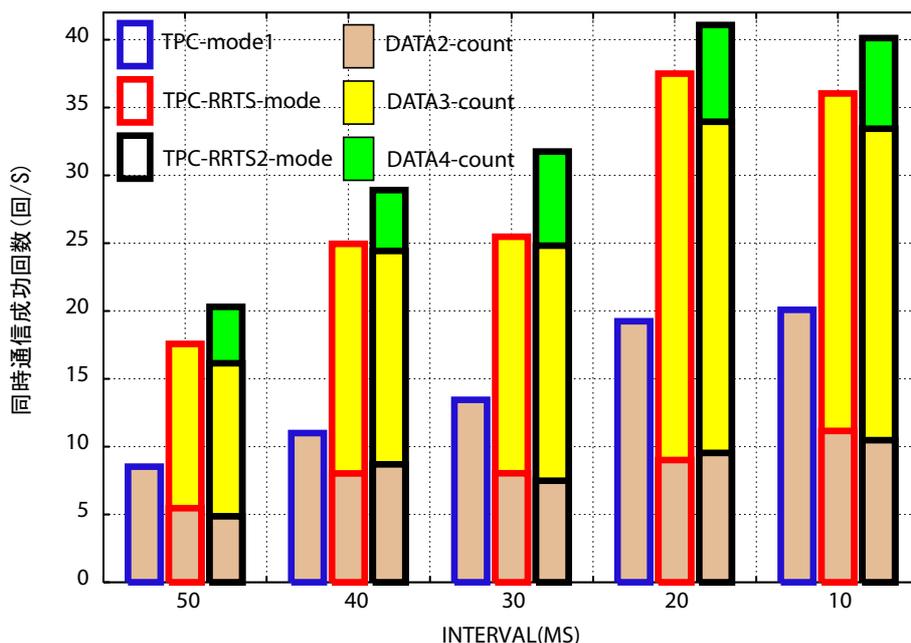


図 6.24: 600 × 600 6flow における同時通信成功回数

末がパケットを保持していない場合、同時通信を行うことができないため、同時通信成功回数は 400 × 400 と比較して減少した。一方、エリアを拡大した影響により、DATA4 の同時通信機会が増加し、TPC-RRTS2-mode が 3 手法の中で最も優れたスループットを達成した。

## 6.5 600×600 36 端末 移動環境におけるシミュレーション

最後に端末が移動する状況のシミュレーションを行う。端末の移動は Random Way Point モデルに従い、最低速度 = 1.0M/S，最高速度 = 5.0M/S，平均停止時間 = 30S とした。

スループット結果を図 6.25 に示す。DCF に関しては移動がない場合とほとんど変化ないが、他の 3 手法については多少の劣化が見られる。これは確率制御により同時通信が試みられているが、確率の値が収束するまでに多少の時間を要して

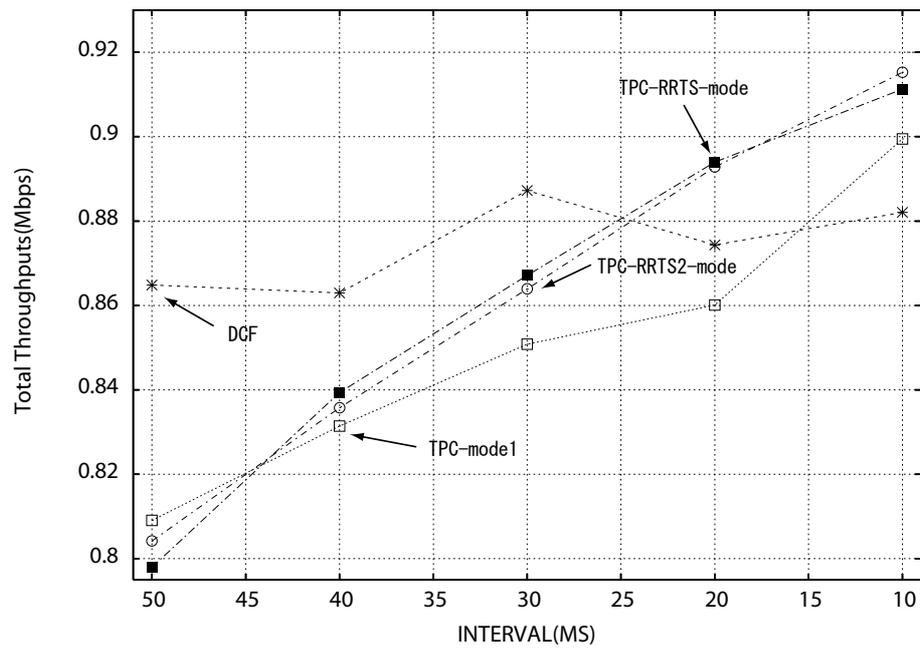


図 6.25: 600 × 600 移動環境, 6flow におけるスループット

いるからであると考えられる。

同時通信回数のシミュレーション結果を図 6.26 に示す。移動の影響により、3 手法とも同時通信成功回数が減少している。

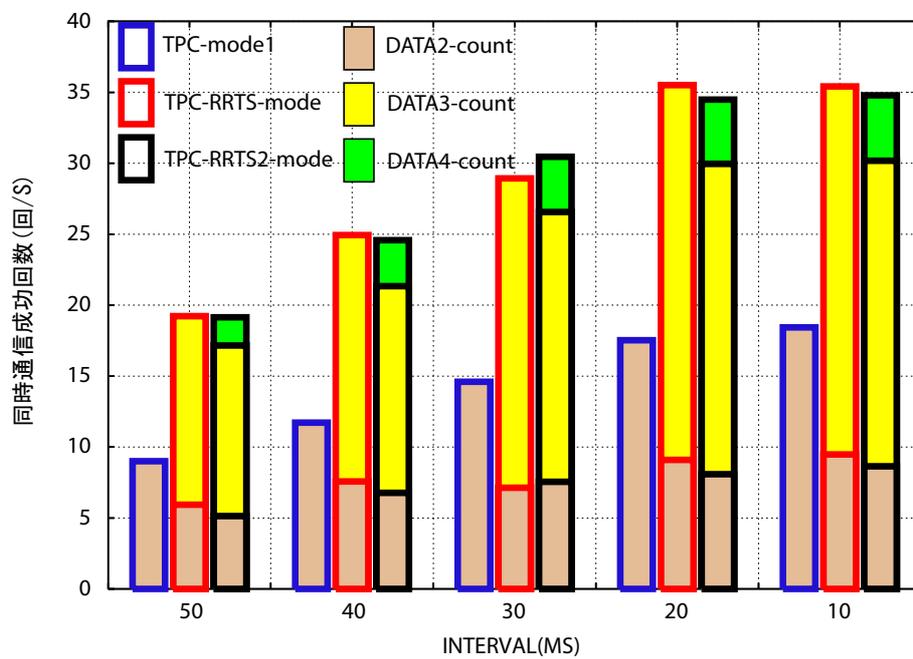


図 6.26: 600 × 600 移動環境, 6flow における同時通信成功回数

## 第7章 結論

### 7.1 まとめ

本論分では DCF の過剰な NAV を和わらげ、同時通信を行うことによりスループットの改善を図る MAC プロトコルの提案を行った。RRTS を導入することにより、コントロールパケットのオーバーヘッドは増大したが、Access Window の開始時刻を検知できる端末数が増加し、同時通信の可能性が大幅に向上し、スループット改善することをシミュレーションを通して実証した。送信電力制御を用いることにより、さらに同時通信の可能性が向上し、スループットが改善された。また、エリア拡大の影響、端末の移動の影響についてもシミュレーションを行い計測し、考察した。

### 7.2 今後の課題

#### 7.2.1 公平性

同時通信によりスループットは向上したが、端末の公平性という点では問題が残る。同時通信を行っている端末は常に同じペアである可能性が高いので、同時通信した端末のスループットが他の端末より高くなる可能性があり、公平性が保たれていないと考えられる。これを解消する 1 つの方法としてバックオフカウンタの値を増加させることが考えられる。DCF ではバックオフカウンタが 0 になる

と RTS を送信し、データパケットを送信していて、次のチャネルアクセスの際はバックオフカウンタの値を乱数により新たに与えられている。これにより理想的にはすべての端末が順々にチャネルアクセスすることができる。

本論文の model , RRTS-mode , RRTS2-mode では最初の通信の予約 (RTS1 送信時) をするためのバックオフカウンタと、Access Window 中に用いるバックオフカウンタは独立おり、RTS1 のバックオフカウンタの扱いに関しては DCF と同様である。そのため、Access Window 中に同時通信した端末が次のチャネルアクセスで連続してデータパケットを送信する機会を得てしまう可能性がある。そこで、Access Window の競合に勝ち、同時通信した端末は、RTS1 のバックオフカウンタの値を増加させることにより公平性の問題は多少解消されるのではないかと考えられる。

## 7.2.2 コントロールパケット効率の向上

Access Window 中にコントロールパケットを交換し、同時通信を試みることにより、スループットは向上したが、多くの無駄なコントロールパケットが送信された。これは、上述したように、Access Window 中にコントロールパケットを交換しても、SINR や送信電力の規制の問題で同時通信を断念する場合があるからである。さらに、Access Window 中の競合が激しくなると、コントロールパケット同士の衝突が増加してしまう。スループット値を保ちつつ、コントロールパケット効率を向上させる手法が必要である。

## 7.2.3 フェージング環境での評価

本論文では two way ground propagation モデルを仮定したが、1 連の通信は 6MS 程度で終了するため、端末が 10m/s の高速で移動していた場合でも通信中にわず

か6cmしか移動しないため、チャンネル利得の変化はほとんど無視できる。そのため、将来受ける干渉電力  $P_{post-interfere}$  や、式4.2の送信電力制限  $P_{T-allowed}$  等の推定が非常に有効であった。しかし、フェージング環境のようにチャンネル利得が時間とともに変化する場合、推定誤差が生じる。そのため、それぞれの推定値にマージンを与え、チャンネル利得が変化した場合でも対応できるようにする必要があると考えられる。マージンを取ることにより、同時通信の可能性は低くなり、さらにチャンネル利得がマージン以上に大きく変動した場合はパケットの衝突が発生し、伝送特性が劣化することが予想される。

## 第8章 謝辞

本研究を行うにあたり，指導教官の浅野正一郎教授には，親身のご指導と数多くの適切な助言を賜りました．ここに厚く御礼を申し上げます．また大学院生の生活を楽しく充実したものにしてくれた安達研究室の辻下氏，倉沢央氏，浅野研究室の Ahmad Suffian 氏，劉智星氏，木村英雄氏，NGUYEN PHONG VU 氏，三瓶氏に深く感謝いたします．最後に私の学生生活を支えて修士研究の機会を与えてくださった両親および兄弟に感謝いたします．

## 参考文献

- [1] S. Agarwal, S. Krishnamurthy, R. H. Katz, and S. K. Dao. Distributed Power Control in Ad-hoc Wireless Networks. In PIMRC01, 2001.
- [2] E.-S. Jung and N. H. Vaidya, “ A power control MAC protocol for ad-hoc networks, ” in ACM MOBICOM, 2002.
- [3] Alaa Muqattash and Marwan Krunz “A Single-Channel Solution for Transmission Power Control in Wireless Ad Hoc Networks,” Proc. ACM MobiHoc’04, pp.210-221, May 2004.
- [4] A. Muqattash and M. Krunz, “POWMAC: A single-channel power-control protocol for throughput enhancement in wireless ad hoc networks,” IEEE Journal on Selected Areas in Communications (JSAC)-Special Issue on Advances in Military Wireless Communications, 23(5):1067–1084, May 2005.
- [5] Kaaxin Xu, Mario Gerla, “Sang Bae How Effective is the IEEE 802.11 RTS/CTS Handshake in Ad Hoc Networks?,”IEEE Global Telecommunications Conference, 2002, Vol. 1, pp. 72 - 76
- [6] F.Talucci and M. Gerla, “MACA-BI: A Wireless MAC Protocol for High Speed Ad Hoc Networking,” Processings of IEEE ICUPC 1997, vol.2,pp.913-917,October 1997.

- 
- [7] V.Bharghavan, A.Demers, S.Shenker, L.Zhang, "MACAW: A media access protocol for wireless LAN's", ACM SIGCOMM Comput. Commun. Review, vol.24, no.4, pp.212-225, 1994/10.
- [8] 原田貴弘, 太田能, 森井昌克, "無線マルチホップ網における IEEE 802.11 DCF の TCP スループット特性の改善," IEICE Trans. on Commun., vol.J85-B, no.12, pp.2198-2208, Dec. 2002.
- [9] Fengji Ye, Su Yi and Biplab Sikdar "Improving Spatial Reuse of IEEE 802.11 Based Ad Hoc Networks,"IEEE GLOBECOM, 2003.
- [10] Vikas Kawadia and P. R. Kumar, " Principles and protocols for power control in ad hoc networks, " in IEEE Journal on Selected Areas in Communications, 2005.
- [11] T. A. ElBatt, S. V. Krishnamurthy, D. Connors, and S. Dao. "Power Management for Throughput Enhancement in Wireless Ad-Hoc Networks," In IEEE International Conference on Communications, pages 1506–1513, 2000.
- [12] J. Deng and Z. J. Haas. "Dual Busy Tone Multiple Access (DBTMA) : A New Medium Access Control for Packet Radio Networks," In IEEE International Conference on Universal Personal Communications (ICUPC 98) , volume 2, pages 973–977, 1998.
- [13] Z. J. Haas and J. Deng. "Dual Busy Tone Multiple Access (DBTMA) Performance Evaluation," In VTC 99 , volume 1, pages 314–319, 1999.
- [14] Qualnet, <http://www.scalable-networks.com/>