

アドホックネットワークの性能を向上させる ストロングビジートーン導入の検討と評価

伊藤 智洋¹ 旭 健作¹ 鈴木 秀和¹ 渡邊 晃¹

概要：アドホックネットワークの隠れ端末問題を解決するために，IEEE802.11 では RTS/CTS による方式が採用されている．しかし，この方式だけでは，トラフィックが増加するにつれ RTS/CTS そのものの衝突が発生しやすく，スループットが大きく低下する．本稿では，ビジートーンの電波到達範囲を拡大させたストロングビジートーン (Strong Busy Tone) と呼ぶ特殊な制御信号を用い，さらに CSMA/CA のバックオフアルゴリズムを修正することによりアドホックネットワークのスループットを向上させる方法について提案する．

Researches and Evaluation of Strong Busy Tone Introduced to Improve the Performance of Ad hoc Networks

TOMOHIRO ITO¹ KENSAKU ASAH¹ HIDEKAZU SUZUKI¹ AKIRA WATANABE¹

1. はじめに

ユビキタス社会に向け無線 LAN 技術の普及が急速に進んでいる．無線 LAN は配線工事が不要であり，端末の移動が可能であることから容易に LAN の構築が可能である．中でも，端末同士で直接通信を行うことができるアドホックネットワークが注目されている．しかし，アドホックネットワークは隠れ端末問題 [1] による影響が大きく，トラフィックの増加により大幅にスループットが低下することが知られている．

IEEE802.11[2] では，隠れ端末問題の対策として RTS (Request to Send) /CTS (Clear to Send) による方式が採用されている．RTS/CTS は，周辺端末を仮想的なキャリア検出状態に移行させ，一定時間送信を停止させることにより衝突を防止する．しかし，この方式だけではトラフィックが増加するにつれ RTS 同士の衝突や CTS とデータの衝突が発生することが避けられない．その理由として，RTS/CTS 自体がパケットであるため，一連のシーケンスの実行に所定の時間が必要となるためである．また，

RTS/CTS で衝突が発生すると周辺端末が送信停止状態のまま残ってしまう，いわゆるさらし端末問題が発生する．

ビジートーンを用いることにより，周辺の端末を制御しスループットを改善させる手法が提案されている．ビジートーンとは，単一の周波数の電波で，送信端末が通信中であることを周囲に伝える制御信号である．ビジートーンは，帯域が非常に狭いため電力消費が小さいという特徴がある．また，複数の装置が同時にビジートーンを発生させたとしても，周辺の装置はこれを検知することができる．

通信時に発生するノイズの影響を防止する方式 [3]~[6] は，ノイズの発生する範囲に対してビジートーンを送信することにより，ノイズによる影響を防止することが可能となる．また，RTS/CTS にビジートーンを適用することで隠れ端末問題を解決する方式 [7]~[9] は，RTS/CTS の送信と同時にビジートーンを送信することにより，隠れ端末による衝突を防止することが可能となる．

本稿ではビジートーンの有用性を改めて見直し，新たなスループット向上策として提案する．すなわち，ストロングビジートーン (以下 SBT: Strong Busy Tone) [10][11] と呼ぶビジートーンの電波到達範囲を拡大した特殊な制御信号を用い，周辺端末を広範囲にわたって制御することにより，隠れ端末問題とさらし端末問題を同時に防止

¹ 名城大学大学院理工学研究科
Graduate School of Science and Technology Meijo University, Nagoya, Aichi 468-8502, Japan

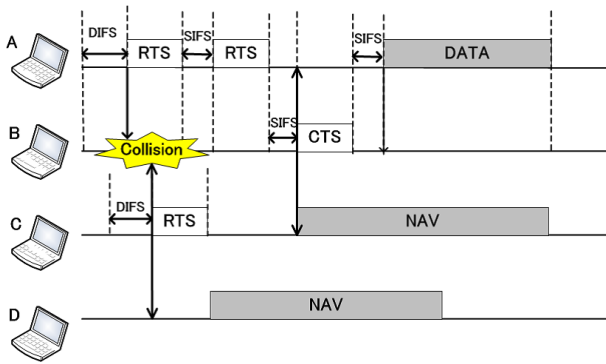


図 1 RTS/CTS の課題 1

Fig. 1 One example of the issue in RTS/CTS.

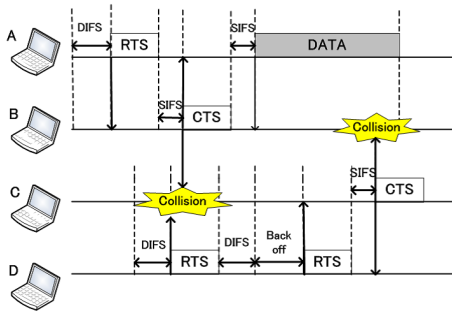


図 2 RTS/CTS の課題 2

Fig. 2 The second example of the issue in RTS/CTS2.

する方法を提案する。SBT を用いて制御することにより CSMA/CA の待機時間のスロットタイム Δt の値を短縮することが可能となり、スループットを更に向上させることができる。

SBT は広範囲に渡って周辺端末の送信を抑えることによるため、システムとしてのスループットを下げる要因にもなりえる。そこで ns-2(Network Simulator2) を用いてシミュレーション評価を行い、RTS/CTS による方式と比較した。その結果、あらゆるケースにおいて SBT を利用した方が通信性能が向上することがわかった。

以下、2 章では既存方式とその課題について、3 章では提案方式についてそれぞれ説明する。4 章ではシミュレーションとその結果を考察し、最後に 5 章でまとめを行う。

2. 既存技術とその課題

2.1 RTS/CTS による方式の課題

RTS/CTS による方式の課題の例を図 1、図 2 に示す。端末 A と端末 C は隠れ端末の関係にあり、端末 A から端末 B に送信が行われる例を示す。図 1 では、端末 A と端末 C がほぼ同時に端末 B に対して送信を開始しており、RTS の衝突が発生する様子を示している。RTS/CTS のやりとりには所定の時間を要するためこのような現象は避けられない。RTS 同士の衝突が発生すると CTS が返信されないため、端末 A、C とともに再度 RTS の送信から始める必要がある。ただし、RTS/CTS により、長データを再度

送信してしまうことは避けることができる。問題となるのは、端末 D が RTS を受信するため NAV 状態に入りさらに端末状態となることである。また、2.2 でも述べるように RTS/CTS のやりとりにかかる時間は無視できない程大きく、これもスループット低下の要因となっている。

図 2 では端末 A が送信した RTS に対して、端末 B は CTS を返信して送信を許可している。ここで、RTS/CTS のやりとりの間にさらに遠隔にある端末 D が RTS を送信すると、端末 B の CTS と端末 D の RTS が端末 C の部分で衝突する。端末 D は CTS の応答がないため、RTS を再送信する。一方、端末 A は端末 B からの CTS を受信しているので端末 B に対してデータ送信を始める。端末 C は端末 D の RTS に CTS を返信するため、端末 A のデータと端末 C の CTS が衝突する。これにより、端末 A は再送信が必要となる。この現象は、長データを無駄に送信してしまうと言う点で、スループット低下の原因となる。

これらの課題は RTS/CTS がパケットによる交換であるために所定の時間を必要とすることが原因である。

2.2 PLCP に起因する課題

RTS/CTS のやりとりにかかる時間は無視できない程大きい。その要因として PLCP (Physical Layer Convergence Protocol) のオーバーヘッドが挙げられる。PLCP は、無線でパケットを送信する際に必須となる物理ヘッダで、図 3 に示すように IEEE802.11MAC ヘッダの前に付加され、PLCP プリアンブルと PLCP ヘッダから構成されている。PLCP プリアンブル部分には受信装置が同期を確立するために必要な情報が記載されており、PLCP ヘッダ部分には MAC フレームの速度に係る情報が定義されている。

IEEE802.11g を例にとると、MAC フレーム部分の通信最大速度は 54Mbps であるが、PLCP 部分は全ての端末が受信できるよう 2Mbps と定義されている。このため、MAC フレームより PLCP 部分の方がはるかに長い時間を要する場合がある。PLCP はデータだけでなく RTS、CTS、ACK などにも付加される。表 1 に IEEE802.11g における一連のシーケンスの所要時間を示す。表 1 からわかるように RTS、CTS、ACK はいずれも MAC フレーム本体部分が $3\mu\text{s}$ 程度であるのに対し、PLCP 部分に $26\mu\text{s}$ もの時間を要する。RTS/CTS の MAC フレーム構造は短く定義されているもののパケット全体の送信時間は大きいことがわかる。そのため RTS/CTS のやりとりがオーバーヘッドになるとともに、隠れ端末同士が同時に送信し衝突する可能性が高くなっている。

2.3 ビジートーン

ビジートーンを用いて周辺端末を制御することにより、スループットを改善する技術が提案されている。[3]~[6] の方式は、通信時に発生するノイズの範囲に合わせてビジー

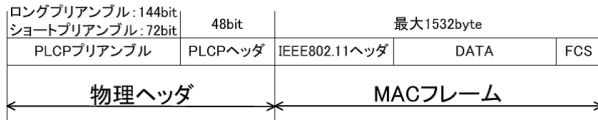


図 3 PLCP のフォーマット
Fig. 3 The format of the PLCP.

表 1 各シーケンスに要する時間
Table 1 Time of each sequence.

IEEE802.11g	時間 (μs)	
	PLCP	本体
DIFS		34
Backoff		135~9207
RTS	26	3
SIFS		10
CTS	26	3
DATA (MAX 長)	26	227
ACK	26	3

トーンの送信範囲を調節することでノイズの影響を防止することが可能となる。無線通信は、通信時にノイズが発生し、このノイズは通信距離に比例して拡大する。端末は周辺にキャリアが確認されないため通信を開始するが、ノイズが発生していると干渉してしまい通信にエラーが発生する。そこで、通信開始時にノイズの発生する最大の範囲に対してビジートーンを送信することにより、周辺の端末を抑制する。その後、単位時間ごとにエラーが発生しなければ範囲を狭め、発生すれば範囲を拡大することにより、ノイズによる影響を防止することができる。

近年では、RTS/CTS方式にビジートーンを付加することで隠れ端末問題を解決する方式 [7]~[9] が提案されている。RTS, CTS パケットの送信時に同時にビジートーンを送信することにより、制御パケット同士が衝突することを防止することが可能となっている。図 4 にその動作を示す。端末 A が RTS を送信した際に同時にビジートーンが送信することにより、周辺の端末を制御する。端末 B は CTS を返信する際に同時にビジートーンを送信し、DATA の受信が終了するまでビジートーンを送信し続ける。その後、端末 A は DATA の送信と同時にビジートーンを送信することにより周辺の端末を制御する。この方式を用いることにより、図 2 の端末 A の DATA と端末 C の CTS が端末 B において衝突することを防止することができる。しかし、この方式では図 1 に示す RTS 同士の衝突や図 2 に示す端末 B の CTS と端末 D の RTS が端末 C において衝突することを防止することはできず、端末 D が再送を繰り返すことを防ぐことはできない。そのため、ネットワーク全体のスループットが低下することに繋がると考えられる。

上記に示すように、既存のビジートーン技術では隠れ端末問題を完全に解決することができていない。

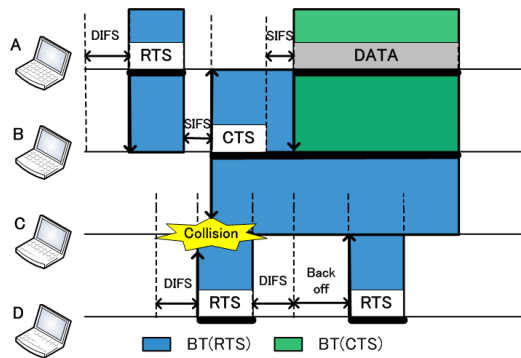


図 4 既存ビジートーンによる課題の解決
Fig. 4 To solve these issues with existing technology of busy tone.

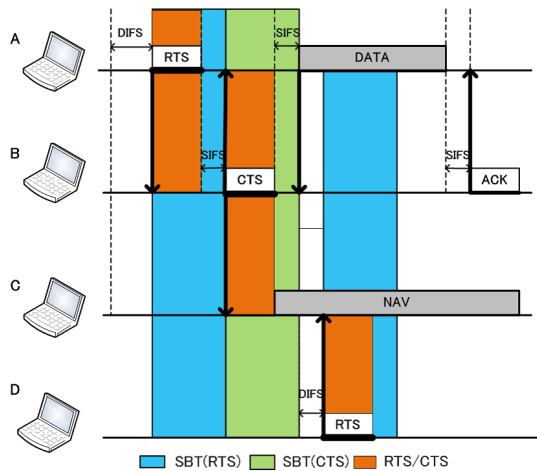


図 5 SBT の動作
Fig. 5 The operation of SBT.

3. 提案方式

本稿では、既存の技術の課題を解決するために SBT の導入とこれにより可能となるスロットタイムの短縮について提案する。

3.1 SBT の導入

SBT は、ビジートーンの電波到達範囲を拡大することにより遠隔の端末を制御することを可能とした制御信号である。各端末は、RTS および CTS を送信する際に、SBT を同時に送信することにより、遠隔端末が送信開始することを防止する。各端末は SBT を検出した場合、新たな通信を開始することができない。ただし、通信を既に行っている場合は、SBT を検出してもそれを無視する。

図 5 に SBT を導入した場合の動作を示す。端末 A は RTS 送信と同時に、端末 D まで到達するよう SBT の電波到達範囲を 3 倍に拡大し送信する。次に端末 B は CTS 送信と同時に、端末 D に到達するよう SBT の電波到達範囲を 2 倍に拡大し送信する。端末 D は SBT を受信している間、RTS を送信できない。この方法により図 1, 図 2 に示すよ

うなパケットの衝突を防ぐことができる。図5では、SBTによる送信抑制が終了した直後に、端末DがRTS送信を行った例を示している。端末Cは既にNAV状態に入っているためCTSを返信せず、端末Aの送信は正常に終了することができる。このとき端末Dの送信するSBTが端末Aにも到達するが、端末Aは送信中であるためSBTを無視し、データ送信に影響を与えることはない。

3.2 スロットタイムの短縮

SBTを導入することによりスロットタイム(Δt)の短縮が可能になる。 Δt を短縮することによりCSMA/CAの待機時間を減らし、スループットを向上することが可能である。CSMA/CAにおける再送時のバックオフ時間は以下の式によって決定される。

$$\text{Back off Time} = \Delta t \times r(CW)$$

ここで Δt はスロットタイム、 r は0~CWの乱数、CWはコンテンションウィンドウサイズである。CWは衝突回数に応じて、15, 31, 63, 127, のように変化し、最大値は1023である。 Δt の値はパケット情報や通信の制御を行う際に必要な時間の合計で802.11gでは $9\mu s$ と定義されている。802.11gの場合、 Δt の値 $9\mu s$ の内訳は以下のように設定されている。

$$\begin{aligned} \Delta t = & \text{CCATime} + \text{AirPropagationTime} \\ & + \text{RxTxTurnaroundTime} + \text{MACProcessingDelay} \end{aligned}$$

- CCATime: 端末の状態判定時間 ($4\mu s$)
- AirPropagationTime: 伝搬時間 ($1\mu s$)
- RxTxTurnaroundTime: 送受信状態切り替え時間 ($2\mu s$)
- MACProcessingDelay: MACの処理時間 ($2\mu s$)

これらの値は、送信される情報がパケットであることが前提で決定されている。ここで、SBTを用いた制御を行うことを前提にすると、不要な項目を省くことが可能である。CCATimeは、端末が送信状態か受信状態かを判定する時間である。SBTを用いることにより送信を行う端末以外は送信が抑えられることから、送信端末以外は受信状態と判断できるためこの値は省略することができる。AirPropagationTimeは、送信されるデータの伝搬時間である。通信を行う上で必須であり、省略することはできない。RxTxTurnaroundTimeは、送受信状態をハードウェア的に切り替えるために必要となる時間である。そのため、送信を行う際に状態を切り替えることは必須であるため省略することはできない。MACProcessingDelayは、MACの処理時間である。SBTを用いた場合、SBTは情報を一切含まない電波であることから、MAC処理時間は非常に少ないものであり、省略することができる。以上のことから、SBTを用いた制御方式においてはSBTの伝搬時

表2 全体のパラメータ

Table 2 Parameters for the entire.

アクセス方式	IEEE802.11g
SBT (RTS) 電波到達範囲 (m)	300
SBT (CTS) 電波到達範囲 (m)	200
フィールド (m)	300×300
伝搬方式	Two Ray Ground
アンテナタイプ	Omni Antenna
ルーティングプロトコル	AODV
計測時間 (s)	330
無線帯域 (Mbps)	54

間 (AirPropagationTime) と端末の送受信状態を切り替えるための時間 (RxTxTurnaroundTime) のみ考慮すればよい。

伝搬時間は端末間距離を100mとすると約 $0.3\mu s$ である。SBTによる制御は最大で3ホップ先まで制御する必要があることから、3ホップ先(300m)へSBTが到達するまでの時間を Δt として定義することができる。提案方式では、AirPropagationTimeの値を余裕をみて $1\mu s$ とする。従って、提案方式 Δt の値をAirPropagationTimeとRxTxTurnaroundTimeを合わせた $3\mu s$ まで短縮することができる。

4. シミュレーション

SBTを適用すると、衝突を減少させることはできるが広範囲に渡って周辺端末の送信を抑制するため、スループットを低下させる要因にもなりうる。そこで、ns-2により衝突を減少させたことの効果がどの程度あるかを検証した。さらにスロットタイムの短縮効果がどの程度あるかを検証した。

4.1 シミュレーションパラメータ

表2に計測環境のパラメータ、表3にTCP通信とUDP通信のパラメータを示す。パケット到達範囲は100mとし、SBTの到達範囲はRTS送信時は300m、CTS送信時は200mとした。TCPの通信タイプはFTP通信とし、パケットサイズは1000Byteとした。UDPはVoIP (Voice over Internet Protocol)を想定し、パケットサイズ200ByteのCBR (Constant Bit Rate)、パケット発生率は64Kbpsとした。

以下の3通りのCaseにおいて比較を行った。それぞれのCaseにおける通信方法を表4に示す

- Case1: RTS/CTSによる既存技術の通信
- Case2: SBTのみを適用した通信
- Case3: SBTを適用した上で Δt を短縮した通信

表 3 端末のパラメータ

Table 3 Parameters of the terminal.

TCP 通信	通信タイプ	FTP
	トランスポートプロトコル	TCP
	パケットサイズ (Byte)	1000
UDP 通信	通信タイプ	CBR
	トランスポートプロトコル	UDP
	パケットサイズ (Byte)	200
	パケット発生率 (Kbps)	64

表 4 各 Case における通信方法

Table 4 Communication method in each Case.

	SBT の有無	$\Delta t(\mu s)$
Case1	無	9.0
Case2	有	9.0
Case3	有	3.0

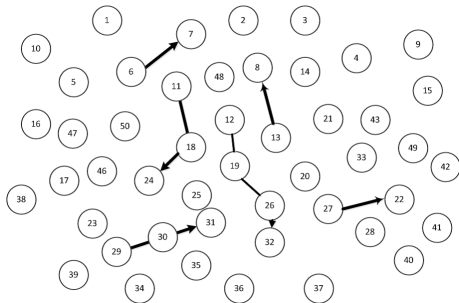


図 6 ランダムシミュレーション環境

Fig. 6 Random simulation environment.

4.2 通信プロトコルごとの計測

通信セッションが TCP の場合と UDP の場合を別々に調査した。

4.2.1 シミュレーション環境

図 6 にシミュレーション環境を示す。通信時に SBT が制御可能な縦 300m、横 300m の範囲にフィールドを設定する。シミュレーション開始時に 50 個の端末全てを 300m×300m の範囲内からランダムに座標を選択し配置を行う。端末の配置が重なった場合は再度選択し配置を行う。ランダムな配置にする理由は、端末が集中した場合や逆に端末が離れた状態においても SBT の効果があることを確認するためである。各シミュレーションは、開始から 20 秒後に任意の 2 台の端末を選択し、最初の通信セッションを発生させる。その後 5 秒ごとに任意の 2 台の端末を選択し通信セッションを発生させてゆき、最大で 61 対の通信セッションが発生させる。この時の総スループットの变化を測定する。

4.2.2 TCP 通信の測定結果

図 7 に TCP 通信のスループット測定結果を示す。横軸は TCP 通信ペア数、縦軸は TCP 通信のスループットの合計値である。図 8 に TCP 通信における単位時間毎の衝突数の推移を示す。横軸は TCP 通信ペア数、縦軸は TCP 通信の衝突数の推移である。表 5 にこの時の衝突数と衝突

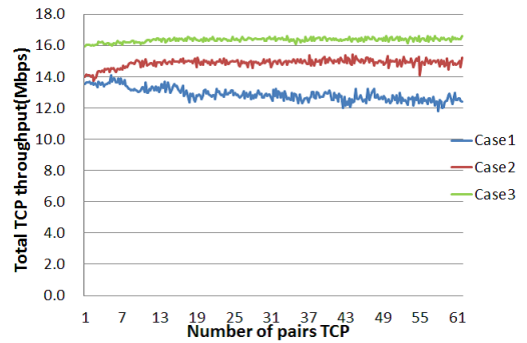


図 7 TCP 通信におけるスループット測定結果

Fig. 7 Throughput in TCP communication.

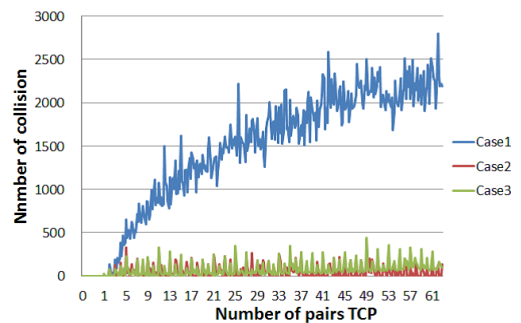


図 8 TCP 通信における衝突数の推移

Fig. 8 Changes in the number of collisions in TCP communication.

表 5 TCP 通信の測定結果

Table 5 Measurements of TCP communication.

	衝突数	衝突率 (%)
Case1	376,880	6.321
Case2	13,229	0.177
Case3	16,715	0.168

率を示す。いずれも 20 回試行した結果の平均値である。

図 7 から、既存方式 (Case1) のスループットは、TCP 通信の数が増加するとともに若干低下していることがわかる。それに対し、提案方式 (Case2, 3) のスループットは、常に Case1 より高く、通信ペア数が増えても低下することなく常に高い値を維持している。さらに、Case3 は Δt の値を短縮することにより、待機時間が短くなっている。そのため、SBT により衝突が大幅に防止された状態で通信回数が増加し、Case2 より常に高い値を示している。この結果から、 Δt の値を変更することによる効果が出ていることが分かる。図 8 より、既存方式と比較し提案方式では衝突数が確かに大幅に削減されていることがわかる。

図 7、図 8 に示されるスループットと衝突数の変化の関係から、既存方式において総スループットが低下しているのは、衝突により再送が増加し、輻輳制御によりウィンドウサイズが縮小されることが要因と考えられる。SBT を用いた場合、トラフィックが増加した状態においても衝突

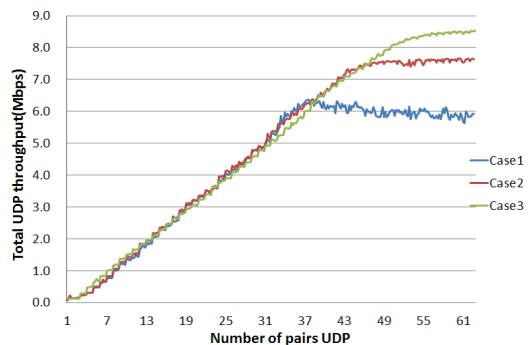


図 9 UDP 通信におけるスループット
Fig. 9 Throughput in UDP communication.

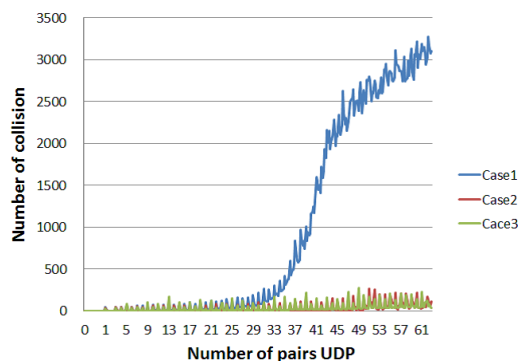


図 10 UDP 通信における衝突数の推移
Fig. 10 Changes in the number of collisions in UDP communication.

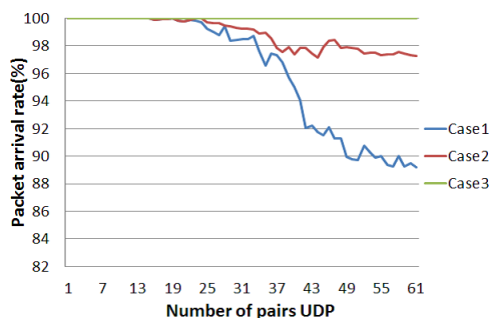


図 11 UDP 通信におけるパケット到達率
Fig. 11 Packet arrival rate in UDP communication.

を軽減することにより、再送を防ぐことができる。以上の結果より、SBTによる送信の抑制よりも、衝突が大幅に軽減されることによる効果の方が大きいことがわかる。

4.2.3 UDP 通信の測定結果

図 9 に UDP 通信のスループット測定結果を示す。横軸は UDP 通信ペア数、縦軸は UDP 通信のスループットの合計値である。図 10 に UDP 通信における単位時間毎の衝突数の推移を示す。横軸は UDP 通信ペア数、縦軸は UDP 通信の衝突数の推移である。図 11 に UDP 通信における通信数の増加に伴うパケット到達率の推移を示す。横軸は UDP 通信ペア数、縦軸は UDP 通信のパケット到達率の推

表 6 UDP 通信の測定結果

Table 6 Measurements of UDP communication.

	衝突数	衝突率 (%)	パケットロス率 (%)
Case1	318,149	7.804	9.904
Case2	9,745	0.169	5.451
Case3	6,281	0.101	3.215

移である。表 6 にこの時の衝突数と衝突率とパケットロス率を示す。いずれも 20 回試行した結果の平均値である。

図 9 から、既存方式 (Case1) では、37 対目の UDP 通信から通信が飽和し、その後は UDP 通信の増加と共にスループットの値が低下していることがわかる。提案方式 (Case2, 3) では、通信が飽和しにくくなっており、スループットの値も増加し続けていることが分かる。また Δt を短縮することよりスループットの飽和を遅らせることが可能であることがわかる。提案方式を用いることにより、衝突は大幅に防止されていることから通信の飽和が防止されていることがわかる。図 10、図 11 に示すように、図 9 に示すスループットの値が低下する時点において、衝突数が急激に増加しておりパケット到達率も比例して低下していることがわかる。Case2, 3 は SBT を用いて衝突を大幅に防止しており、パケット到達率が高い値を維持していることがわかる。特に Case3 においては、ほぼ全てのパケットが到達している。そのため、最も高いスループットを記録している。また、表 6 より、衝突数は既存方式と比較し提案方式では大幅に削減されており、パケットロス率も改善されていることがわかる。SBT を用いることにより、UDP 通信が増加したとしても通信が飽和することを防止することにより、スループットの低下やパケットロスの発生を防止することができる。

以上の結果から、提案方式を用いることで、UDP 通信においても SBT による送信の抑制よりも衝突を防止することによる効果が大きいことがわかる。

4.3 混在環境における計測

4.3.1 シミュレーション環境

図 12 にシミュレーション環境を示す。各端末は 1 ホップ先の端末までの電波が届くように 90m 間隔でメッシュ状に 37 台の端末を配置しており、通信の行いやすい環境を想定している。図 12 に示すように測定用端末として、送信端末を 12、宛先端末を 32 として TCP 通信を行わせる。背景負荷として、端末 12 と端末 32 を除く 35 台の端末からランダムに送信端末と宛先端末を選択し UDP 通信を行わせる。シミュレーション開始から 20 秒後に TCP 通信を開始する。この時は TCP セッションが 1 本確立されているだけである。その後 5 秒毎にランダムに選択された 2 台の端末間で UDP セッションを確立し、背景負荷を徐々に増やしていく。このときに測定対象の TCP スループット

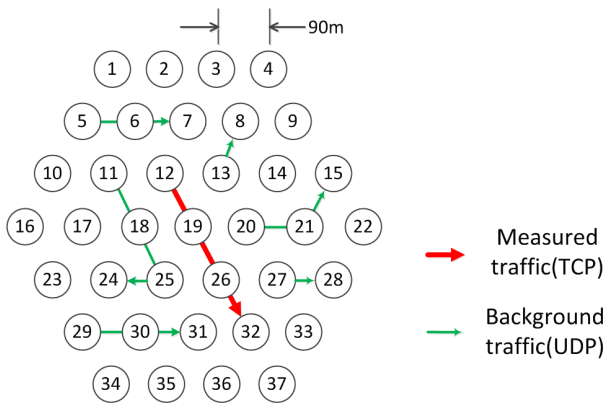


図 12 シミュレーション環境
Fig. 12 Simulation environment.

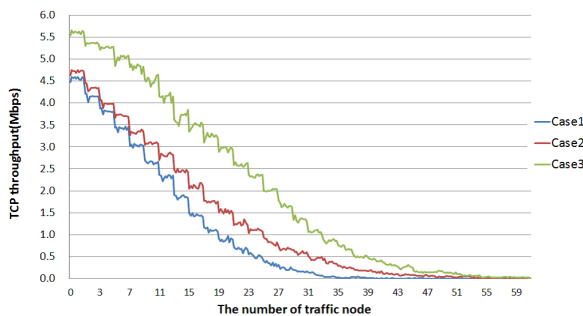


図 13 背景負荷量に対する TCP 通信のスループット測定結果
Fig. 13 Measurement for the amount of background traffic on TCP throughput.

がどのように変化するかを測定した。背景負荷として発生させる UDP 通信は最大で 60 対の通信ペアが発生するものとした。

4.3.2 混在環境の測定結果

図 13 に TCP 通信のスループット測定結果を示す。横軸は背景負荷端末数、縦軸は測定対象となる TCP スループットである。今回の結果は、20 回試行した結果の平均値である。

図 13 から、背景負荷数が増えるごとに段階的に TCP スループットが低下していくことが分かる。これは UDP の背景負荷が一部のトラフィックを占有するためである。背景負荷の端末数が増加するとともに、既存方式と比較し SBT を用いることによりスループットの低下が防止されていることがわかる。さらに、 Δt を短縮することによりスループットの値が大幅に向上していることが分かる。UDP 背景負荷が 31 対発生した状態で比較すると TCP スループットが 10 倍程度に改善されていることがわかる。また、Case3 は Case2 に対して常に高い値を記録しており、通信プロトコルが混在した状態においても Δt を短縮することによる効果が出ていることがわかる。以上の結果から、TCP、UDP が混在した環境においても提案方式の有用性が確認できる。

5. まとめ

RTS/CTS 方式における課題を解決するために、SBT を導入することにより大幅に衝突回数を軽減し、さらに Δt を短縮することにより、送信の待ち時間を低減する方式を提案した。シミュレーションにより TCP のみ、UDP のみ、TCP/UDP 混在環境においてパケット衝突の大幅な軽減やスループットの向上を確認した。SBT は周辺端末の送信を抑制する性質を持つが、衝突を防止することによる効果が大きいことを確認した。特に、 Δt の短縮効果が非常に大きいことも確認できた。

参考文献

- [1] Athanasia Tsertou, David I. Laurenson: Revisiting the Hidden Terminal Problem in a CSMA/CA Wireless Network, IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 7, NO. 7, JULY 2008
- [2] IEEE Std 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (2007).
- [3] Zygmunt J. Haas, Jing Deng.: Dual Busy Tone Multiple Access (DBTMA), A New Medium Access Control for Packet Radio Networks, IEEE ICUPC '98, Vol.2, pp.973-977 (1998)
- [4] Zygmunt J. Haas, Jing Deng.: Dual Busy Tone Multiple Access (DBTMA), A Multiple Access Control Scheme for Ad Hoc Networks, IEEE Trans. Communications, Vol.50, No.6, pp.975-985 (2002)
- [5] Supeng leng, Liren Zhang, Yifan Chen: IEEE 802.11 MAC Protocol Enhanced by Busy Tones, Communications, 2005. ICC 2005. 2005 IEEE International Conference on
- [6] Ke Liu, Supeng Leng, Huirong Fu, Longjiang Li: A Novel Dual Busy Tone Aided MAC Protocol for Multi-hop Wireless Networks, Dependable, Autonomic and Secure Computing, 2009. DASC '09. Eighth IEEE International Conference on
- [7] 萬代雅希, 笹瀬巖: 無線アドホックネットワークにおけるビジートーン信号を用いたメディアアクセス制御プロトコルの特性解析, 電子情報通信学会技術研究報告, CS, 通信方式 101(54), 7-12, 2001-05-11
- [8] 藤原敏秀, 関谷大雄, 萬代雅希, 呂建明, 谷萩隆嗣: 送信範囲の異なる端末で構成される無線アドホックネットワークにおけるビジートーンを使用した MAC プロトコル, 情報処理学会論文誌 47(9), 2815-2829, 2006-09-15
- [9] Abdullah, A.A.: Enhanced Busy-Tone-Assisted MAC Protocol for Wireless Ad Hoc Networks, Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd
- [10] 後藤秀暢, 渡邊晃: アドホックネットワークのスループットを向上させるストロングビジートーンの提案, IPSJ SIG Technical Report, 情報処理学会研究報告, 2011-MBL-57, Vol.2011, No.26, pp.1-8, Mar.2011.
- [11] 森一養, 渡邊晃, 後藤秀暢: ストロングビジートーンを用いたアドホックネットワークにおけるメディアアクセス方式の提案, 全国大会講演論文集, 2011(1), 151-153, 2011-03-02