

スロットの並行割当てを行う新しい自律分散型 TDMA 放送計画アルゴリズム

小原 誠[†] 松野 浩嗣^{†a)}

A New Distributed TDMA Broadcast Scheduling Algorithm Assigning Slots Transmitting in Parallel

Makoto KOBARA[†] and Hiroshi MATSUNO^{†a)}

あらまし Ephremides と Truong (1990) は、対称型パケット無線ネットワークにおいて、パケットの衝突回避条件に基づきながら、各ノードになるべく多くの送信スロットを割り当てることのできる自律分散型 TDMA 放送計画アルゴリズムを提案している。しかし同時に、彼らのアルゴリズムでは各ノードに割り当てられる送信スロットに大きな偏りが生じることを今後解決すべき問題として残している。本論文では、Ephremides と Truong (1990) のアルゴリズムよりも少ないスロットで 1 フレームを構成することができる新しい自律分散型 TDMA 放送計画アルゴリズムを提案する。更にこれが送信スロットの偏りを改善できることを計算機実験によって示す。

キーワード パケット無線ネットワーク、時分割多重 (TDMA) 方式、送信スロット割当て、分散アルゴリズム、並行送信

1. ま え が き

パケット無線ネットワーク (PRN) を構築する場合、CSMA [1] や MACA [2] に代表される競合型のプロトコルでは、パケットの衝突がネットワーク全体のスループットを著しく低下させることが知られている。一方、無線ネットワークのみならず有線ネットワークにおいても、予約型媒体アクセス方式として時分割多重 (TDMA) 方式が使われているが [3]、この方式は各ノード間で同期をとり、あらかじめ決定されたスケジュールで送受信を行うため、原則としてパケットの衝突が起こることはない。

Truong [4] は集中型アルゴリズムで PRN の TDMA 放送計画問題を考え、このうち Sidi ら [5]、Chlamtach ら [6] により、パケットの衝突を許す放送計画問題の自律分散型 TDMA 放送計画アルゴリズムが提案された。

そして、Ephremides と Truong は文献 [7] において、パケットの衝突が起こらない TDMA スケジュールを生成できる自律分散型 TDMA 放送計画アルゴリズムを提案している。このアルゴリズムは、各ノードに一意的な識別番号が割り当てられている対称型 PRN において、PRN を構成するノード数と同じ数のスロットで送受信スケジュールの 1 フレームを構成することで、ノードごとに一つの送信スロット割当てを保証したうえで、更にパケットの衝突回避条件に基づきながらより多くの送信スロットを割り当てるものである。このアルゴリズムの特徴は、各端末には一意的な識別番号と PRN を構成するノード数の二つの情報のみがある。あらかじめ割り当てられていればよく、各々のノードが PRN 全体の接続状況を知っておく必要がないことである。

更に、文献 [8], [9] ではこのアルゴリズムをもとに、スケジューリングの高速化や周波数資源の効率的な割当てなどの考察を行っている。

この Ephremides と Truong のアルゴリズムは、自律分散的にノード間の通信スケジュールが決定できることから、筆者らはこのアルゴリズムに注目した。

本論文では、Ephremides と Truong のアルゴリズム

[†] 山口大学大学院理工学研究科, 山口市
Graduate School of Science and Engineering, Yamaguchi University, 1677-1 Yoshida, Yamaguchi-shi, 753-8512 Japan

^{††} 山口大学理学部, 山口市
Faculty of Science, Yamaguchi University, 1677-1 Yoshida, Yamaguchi-shi, 753-8512 Japan

a) E-mail: matsuno@sci.yamaguchi-u.ac.jp

ムよりも更に効率の良い TDMA スケジュールを生成できる自律分散型 TDMA 放送計画アルゴリズムを提案する。具体的には、文献 [7] 中で今後検討すべき問題として残されている、ノードごとの送信スロットの割当数に大きな偏りが生じることが、提案アルゴリズムによって改善できることを示す。

2. 時分割多重アクセス (TDMA)

無線局に対して一定の速度と遅延時間をもつ回線を提供する媒体アクセス制御方式として、時分割多重アクセス (TDMA: Time Division Multiple Access) 方式がある [3]。

時分割多重方式とは、時間軸を一定間隔に分割し、分割された時間のなかでノードの送信権を切り換えながら与えることで、パケットの衝突なしに伝送を行う同期方法である。一定間隔に分割された時間軸はスロットと呼ばれ、スロット群の 1 周期をフレームと呼ぶ。各ノードは送信すべきスロットと受信すべきスロットが記された送受信スケジュールに基づき送信を行う。

[TDMA 方式における衝突回避条件と並行送信]

TDMA 方式によるスケジューリングを考えると、パケットの衝突発生とその回避条件は次のように表される。なお、本論文で扱う無線ネットワークは対称経路型パケット無線ネットワークであり、 n ノードからなるノード集合を V 、ノード間リンク集合を E とするグラフ $G = (V, E)$ で考える。またこのグラフでは、一般性を失うことなく $V = \{1, 2, \dots, n\}$ とする。

スロット i にて放送を行うノードの集合を A_i とし、ノード $p \in A_i$ について p より 2 ホップ先までのノード集合を放送ゾーンと呼び C_p^i とする。送受信スケジュールの任意のスロット i において、ノード p と任意のノード $q (\neq p)$ との間でパケットが衝突しないためには、以下に示す衝突回避条件を満たしている必要がある。

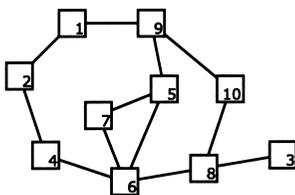


図 1 PRN の例
Fig. 1 A graph representing a PRN.

$$q \notin C_p^i$$

このことより、ノード q がスロット i で送信を行うノード p の放送ゾーン C_p^i に含まれないとき、ノード q はスロット i にてノード p との間でパケットの衝突なしに同時に送信を行うことができる。これをノード p とノード q が並行送信が可能であるという。

例えば、図 1 に示される無線ネットワークで、スロット 1 にてノード 1 が送信を行うとすると、その放送ゾーン C_1^1 は、 $C_1^1 = \{2, 4, 5, 9, 10\}$ となる。

すなわち、スロット 1 でノード 1 が送信している間では、ノード $\{2, 4, 5, 9, 10\}$ は送信を差し控えなければ、パケットが衝突してしまうことになる。なお、グラフが密であるほど (ノード間のリンク数がより多くなるほど)、放送ゾーンに含まれるノード数は多くなるが、隠れ端末となるノードの数は少なくなることに注意されたい。

3. Ephremides と Truong のアルゴリズム

Ephremides と Truong (以下、E&T と略記) は、先に示した衝突回避条件より、パケットの衝突回避のために必要となるのは 2 ホップ先までのノードのスロット割当情報であることに着目し、並行送信を行うことにより各ノードがもつ送受信スケジュールの 1 フレーム当りの送信スロット数をなるべく大きくするような、自律分散型の TDMA 放送計画アルゴリズムを提案している [4]。

このアルゴリズムでは、同期がとられた各々のノードが自律分散的に情報交換を行うことで、送受信スケジュールを構成していく。この送受信スケジュールを構成するためのフレームを、SchedulingTDMA と呼び、データを送信するフレームと区別する (図 2)。E&T の場合は、SchedulingTDMA はスロット p でノード $p (p \in V)$ が送信を行う 1 フレーム n スロットからなり、 $2n$ フレームの時点で、送受信スケジュー

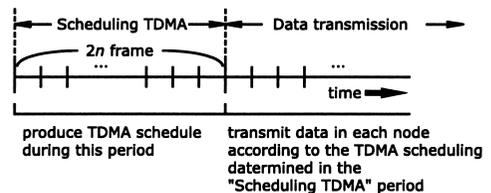


図 2 Scheduling TDMA
Fig. 2 Scheduling TDMA.

ルの構成が完了する．なお各ノードには一意な識別番号と，PRN を構成するノード数の二つの情報のみがあらかじめ割り当てられているが，PRN 全体の接続状況をあらかじめ知っておく必要はない．

図 1 の 10 ノードの無線ネットワークを用いて，その動作を簡単に説明する．E&T のアルゴリズムでは，まず SchedulingTDMA が n フレームまで周回した時点で（この例の場合第 10 フレーム第 10 スロットまで周回した時点で），図 3 に示すような送受信スケジュールを構成する．これを特に，最終的に出力する送受信スケジュールの骨組みとなることから，スケルトンスケジュールと呼ぶ．

図 3 に示した，E&T のアルゴリズムでのスケルトンスケジュールは，ノード q ($q \in V$) がスロット q を送信スロットとして予約し（図中の “R”），ノード q の放送ゾーン内のノードはスロット q を受信スロットとして予約（図中の “—”）した，1 フレームが n スロットからなる作りかけの送受信スケジュールである．このスケルトンスケジュールの段階では，まだ送信にも受信にも予約されていない未割当スロットを多数含んでいる．例えば，ノード 1 はスロット 1 を送信スロットとして予約しているが，放送ゾーン内のすべてのノード $C_1^1 = \{2, 4, 5, 9, 10\}$ では，ノード 1 のスロット 1 での送信に対して衝突をおさえるために，スロット 1 を受信スロットとして予約している．しかし，その他のノードではまだこの時点ではスロット 1 を送信スロットとも受信スロットとも確定しておらず，並行送信が可能であれば送信スロットとして予約することができる．この処理を，ノード間で情報交換を行いながら，次の SchedulingTDMA が $n+1 \sim 2n$ フレームまで周回する間で行う．

未割当てとなっている送信スロットは，番号の小さいノードから優先的に予約していく．例えば図 3 では，ノード 1 が最も優先度が高く，ノード 1 はスケルトンスケジュールが完成した時点で未割当てとなっているスロット 3, 6, 7, 8 をすべて送信スロットとして予約する（図 4）．そしてこのことを放送ゾーン内のすべてのノード $C_1^1 = \{2, 4, 5, 9, 10\}$ に伝え，ノード C_1^1 はスロット 3, 6, 7, 8 を受信スロットとする．同様の処理を，ノード 2, 3, …, 10 まで繰り返し，結果として図 4 のような送受信スケジュールを構成する．ここで，図 4 より明らかなように，スケルトンスケジュール（図 3）を完成させたときに，未割当スロットを送信スロットとして予約できる優先度が最も高いノード

		node #									
		1	2	3	4	5	6	7	8	9	10
slot #	1	R	—	—	—	—	—	—	—	—	—
	2	—	R	—	—	—	—	—	—	—	—
	3	—	—	R	—	—	—	—	—	—	—
	4	—	—	—	R	—	—	—	—	—	—
	5	—	—	—	—	R	—	—	—	—	—
	6	—	—	—	—	—	R	—	—	—	—
	7	—	—	—	—	—	—	R	—	—	—
	8	—	—	—	—	—	—	—	R	—	—
	9	—	—	—	—	—	—	—	—	R	—
	10	—	—	—	—	—	—	—	—	—	R

R: reserved (TX)
 -: blocked (RX)
 blank: unknown

図 3 E&T のアルゴリズムでのスケルトンスケジュールの例
 Fig. 3 Skeleton schedule matrix of E&T's algorithm.

		node #									
		1	2	3	4	5	6	7	8	9	10
slot #	1	R	—	R	—	—	—	R	—	—	—
	2	—	R	R	—	R	—	—	—	—	—
	3	R	—	R	—	—	—	—	R	—	—
	4	—	—	R	R	—	—	—	—	R	—
	5	—	R	R	—	R	—	—	—	—	—
	6	R	—	—	—	—	R	—	—	—	—
	7	R	—	—	—	—	—	R	—	—	—
	8	R	—	—	—	—	—	—	R	—	—
	9	—	—	R	R	—	—	—	—	R	—
	10	—	R	—	—	—	—	R	—	—	R

図 4 E&T のアルゴリズムにより得られる送受信スケジュールの例
 Fig. 4 The final form of schedule matrix of E&T's algorithm.

1 は，送信スロットを多数獲得しているものの，ノード 1 の放送ゾーンに含まれたノードには，少ない送信スロット数しか与えられていない．そして次に送信スロット数が多いのは，ノード 1 の放送ゾーンに含まれず，ノード 1 の次に優先度が高いノード 3 である．

このように，E&T のアルゴリズムで求められる送受信スケジュールでは，送信スロット割当数に大きな偏りができることが指摘されている [7] ．

4. 提案アルゴリズムとその効果

4.1 概要

E&T の自律分散型 TDMA 放送計画アルゴリズムは，それぞれのノード間の双方向通信と同期が保証されている環境において，ネットワークを構成するノード数と各々のノードの識別番号を入力とし，1 フレーム n スロット（ただし， n はノード数）からなる TDMA (SchedulingTDMA と呼ぶ) に基づき自律

分散的に各々のノードが情報交換を行うことで、衝突問題を解消しながらも並行送信が可能で、1 フレーム n スロットからなる TDMA の送受信スケジュールを求めるものである。このアルゴリズムでは、各ノードが自身の送受信スケジュールを求めるにあたり、まずネットワークを構成するノード数と同じ数のスロットでその送受信スケジュールの 1 フレームを構成すると決めたとうえで、ノードごとに一つの送信スロットの割当てを保証し、更にパケットの衝突回避条件と並行送信の性質を利用することで、より多くの送信スロットを割り当てる [7]。

これに対して筆者らが提案するアルゴリズムでは、出力される送受信スケジュールが、E&T のアルゴリズムでは 1 フレーム n スロット「ちょうど」で構成されることを、各ノードに最低 1 回の送信スロット割当てを保証しながら、1 フレームを n スロット「以下で」構成できる。更に、筆者らのアルゴリズムで求められる送受信スケジュールは、E&T のアルゴリズムで求められる送受信スケジュールにおけるノードごとの送信スロット割当ての偏りを大きく軽減しながらも、E&T のアルゴリズムと同程度の数の送信スロットを割り当てることができる。

なお、提案するアルゴリズムは同期のとれた対称型 PRN を対象とし、その伝送方式などについては規定しない。また E&T のアルゴリズムと同様に、各ノードはスケジューリング前に PRN に含まれるノード数 n を知っている必要があるが、PRN 全体の接続状況をあらかじめ把握しておく必要はなく、任意のノードが送信したパケットを隣接ノードが確実に受信可能であることが保証されていればよい。

4.2 動作

本論文で提案する、各ノードが実行する送受信スケジュールを求めるための自律分散型 TDMA アルゴリズムとそこで用いられる変数を、図 15 に示す。

提案アルゴリズムは、E&T のアルゴリズムと同様に、同期をとりながら自律分散的にそれぞれのノードが情報を交換し合うことで送受信スケジュールを構成するものである。各ノードは 1 フレームが n スロットよりなる SchedulingTDMA の間で送信手続き schedulerTX、受信手続き schedulerRX を実行し、この SchedulingTDMA が n フレーム周回しきった時点でスケルトンスケジュールを、 $2n$ フレーム周回しきった時点で送受信スケジュールを出力する (図 15, 19~42 行目)。ただし、E&T の SchedulingTDMA が

ノード p がスロット p にて送信を行うものであったのに対して、筆者らのアルゴリズムでは、SchedulingTDMA の $1 \sim n$ フレームではスロット p でノード p が送信手続き schedulerTX を、それ以外のノードは受信手続き schedulerRX を実行し (25~32 行目)、 $(n+1) \sim 2n$ フレームでは、スロット p でノード $(n+1) - p$ が送信手続き schedulerTX を、それ以外のノードは受信手続き schedulerRX を行う (34~42 行目)。

以下は提案アルゴリズムの動作の説明である。説明のため、任意のノード p が実行している手続きに使われている変数には添字 p を付けるが (ここで、 $self_p = p$)、どのノードにおいても同時刻の SchedulingTDMA のフレーム番号変数 $scTDMAframe$ 、スロット番号変数 $scTDMAslot$ は同じ値をとるため、 $scTDMAframe$ 、 $scTDMAslot$ については添字を付けないものとする。

(A) 手続き schedulerTX : 送信

(A-1) $1 \leq scTDMAframe \leq n$ かつ $scTDMAframe = self_p$ のとき、ノード p は送受信スケジュール sc_p 中に自身の送信スロットを一つ確保する (49~56 行目)。更に必要であれば送受信スケジュール sc のスロット数 $slots$ を更新する (57~60 行目)。そして送受信スケジュール sc_p とそのスロット数 $slots_p$ を送信バッファ $outsc_p$ に収め (61~63 行目)、手続き broadcast により隣接するノードに送信する (75 行目)。ここで送信スロットの確保とスロット数の計算は次のように行う。

$$sc_p[k] = 'R'; \text{ただし } k = \min\{i | sc_p[i] = '\cdot'\}$$

$$slots_p = \max\{slots_p, \max\{i | sc_p[i] \neq '\cdot'\}\}$$

(A-2) $n+1 \leq scTDMAframe \leq 2n$ かつ $(scTDMAframe - (2 \times scTDMAslot) + 1) = self_p$ のとき^(注1)、ノード p はその送受信スケジュール sc_p 中でまだ空白のスロットがあれば、そこを送信スロットとし (67~70 行目)、送受信スケジュール sc_p とそのスロット数 $slots_p$ を送信バッファ $outsc_p$ に収め (71~73 行目)、手続き broadcast により隣接するノードに送信する (75 行目)。

(A-3) それ以外のとき、ノード p は送信バッファ $outsc_p$ のデータを隣接するノードに対して送信する

(注1): 例えば、5 ノードからなるグラフのノード 5 の SchedulingTDMA では、フレーム 6 スロット 1 のとき、となる。

(75 行目).

(B) 手続き schedulerRX : 受信

隣接ノードより 1 ホップまたは 2 ホップ先のノードのもつ送受信スケジュールを受信バッファ in_{sc_p} に受信したとき (84 行目の条件を満たすとき), 次の条件に基づき, ノード $in_{sc_p}.snode$ が送信を行うスロットでは自身は送信を控えるように, 自身の送受信スケジュール sc_p を更新する (86~90 行目).

$$sc_p[k] = '-', \text{ただし } k = \{i | sc_p[i] = 'R'\}$$

更に, 受信した送受信スケジュール in_{sc_p} が隣接するノードのものであるときは特に ($in_{sc_p}.tnode = in_{sc_p}.snode$), これを送信バッファ out_{sc_p} に収め, 中継処理を行うようにする (91~95 行目).

また受信した送受信スケジュール in_{sc_p} のスロット数 $in_{sc_p}.slots$ が自身の送受信スケジュール sc_p のスロット数 $slots_p$ より大きい場合, 自身の送受信スケジュール sc_p のスロット数 $slots_p$ を受信した送受信スケジュールのスロット数 $in_{sc_p}.slots$ で更新し, これを送信バッファ out_{sc_p} に収める (97~99 行目).

4.3 動作の例

先ほど示した図 1 の PRN の例において, 提案アルゴリズムの動きを図 15 に示すアルゴリズムの行番号を利用しながら簡単に説明する.

まず, SchedulingTDMA のフレーム 1, スロット 1 にて, ノード 1 は自身の送受信スケジュール中のうち, 送信にも受信にも予約されていないスロットのうち最もスロット番号の小さいものである, スロット 1 を送信スロットとして予約し (48~56 行目), 自身の送受信スケジュールの 1 フレームが 1 スロットより構成される (57~60 行目: $slots = 0 \rightarrow 1$), 隣接ノード {2, 9} に送信する (61~63 行目, 75 行目).

これを受信した隣接ノード {2, 9} は, ノード 1 がその送受信スケジュールのスロット 1 を送信スロットとして予約したことを知り, 自身の送受信スケジュールのスロット 1 を非送信 (受信) スロットとする (86~91 行目). また受信した送受信スケジュールが隣接ノードのものであることから ($in_{sc}.snode = in_{sc}.tnode$), 中継処理によって 2 ホップ目に届けるためこれを送信バッファに収め待機する (91~95 行目).

更に, ノード {2, 9} は, ノード 1 から受信した送受信スケジュールの 1 フレーム当りのスロット数 ($in_{sc}.slots = 1$) が, 自身の知っている 1 フレーム当りのスロット数 ($slots = 0$) と比較して大きいので, 受信

		node #									
		1	2	3	4	5	6	7	8	9	10
slot #	1	R	-								-
	2										
	3										
	4										
	5										
	6										
	7										
	8										
	9										
	10										

図 5 図 1 での送受信スケジュール構成の例 (SchedulingTDMA: frame=1, slot=1)

Fig. 5 Schedule example of Fig. 1 in proposed algorithm (SchedulingTDMA: frame=1, slot=1).

したもので更新を行う (97~99 行目: $slots = 0 \rightarrow 1$).

このようにして図 5 のような送受信スケジュールを得る.

次に, SchedulingTDMA のフレーム 1, スロット 2 にて, ノード 2 は送信バッファに収められたノード 1 の送受信スケジュールを隣接ノード {1, 4} に送信し (75 行目), これを受信したノード 4 は, 先ほどと同様にノード 1 がスロット 1 を非送信 (受信) スロットとするとともに (86~91 行目), 1 フレームが 1 スロットより構成されるとする (97~99 行目: $slots = 0 \rightarrow 1$) (注 2).

このときは, ノード 4 が受信した送受信スケジュールが 2 ホップ先のものであることから ($in_{sc}.snode \neq in_{sc}.tnode$), ノード 1 の送受信スケジュールの中継処理は行わず, 1 フレームを構成するスロット数のみを中継する (図 6). このようにして, SchedulingTDMA のフレーム 1 においては, ノード 1 の送信スロット予約とそれに対する放送ゾーン内の全ノードの送信抑制, 送受信スケジュールの 1 フレーム当りのスロット数のフルフラディングが行われる.

SchedulingTDMA のフレーム 2, スロット 2 では, ノード 2 が自身の送受信スケジュールのなかで, スロット 2 を送信スロットとして予約するとともに, 自身の求める送受信スケジュールの 1 フレームが 2 スロットよりなるとして, 隣接ノード {1, 4} に送信する.

これを受信した隣接ノード {1, 4} は, ノード 2 がスロット 2 を送信スロットとして予約したことを知り, 自身の送受信スケジュールのスロット 2 を受信スロ

(注 2): このときノード 1 は受信した送受信スケジュールが自身のものであるため, この動作は行わない。

		node #									
		1	2	3	4	5	6	7	8	9	10
slot #	1	R	-	-						-	
	2										
	3										
	4										
	5										
	6										
	7										
	8										
	9										
	10										

図 6 図 1 での送受信スケジュール構成の例
(SchedulingTDMA: frame=1, slot=2)

Fig. 6 Schedule example of Fig. 1 in proposed algorithm (SchedulingTDMA: frame=1, slot=2).

		node #									
		1	2	3	4	5	6	7	8	9	10
slot #	1	R	-	R	-	-				-	-
	2	-	R		-						
	3										
	4										
	5										
	6										
	7										
	8										
	9										
	10										

図 8 図 1 での送受信スケジュール構成の例
(SchedulingTDMA: frame=3, slot=3)

Fig. 8 Schedule example of Fig. 1 in proposed algorithm (SchedulingTDMA: frame=3, slot=3).

		node #									
		1	2	3	4	5	6	7	8	9	10
slot #	1	R	-	-	-					-	-
	2	-	R	-	-						
	3										
	4										
	5										
	6										
	7										
	8										
	9										
	10										

図 7 図 1 での送受信スケジュール構成の例
(SchedulingTDMA: frame=2, slot=2)

Fig. 7 Schedule example of Fig. 1 in proposed algorithm (SchedulingTDMA: frame=2, slot=2).

		node #									
		1	2	3	4	5	6	7	8	9	10
slot #	1	R	-	R	-	-				-	-
	2	-	R		-						
	3		-		R	-					
	4										
	5										
	6										
	7										
	8										
	9										
	10										

図 9 図 1 での送受信スケジュール構成の例
(SchedulingTDMA: frame=4, slot=4)

Fig. 9 Schedule example of Fig. 1 in proposed algorithm (SchedulingTDMA: frame=4, slot=4).

トとして予約するとともに、現在知っている送受信スケジュールの構成スロット数 ($slots = 1$) と比較して受信した送受信スケジュールの 1 フレーム当りのスロット数 ($insec.slots = 2$) が大きいため、受信したもので更新する ($slots = 1 \rightarrow 2$)。そして、更に隣接ノードへと中継するためにこれを送信バッファに収め待機する (図 7)。このようにして、SchedulingTDMA のフレーム 2 においては、ノード 2 の送信スロット予約とその通知 (放送ゾーン内の全ノードの送信抑制)、送受信スケジュールの 1 フレーム当りのスロット数のフルフラッシングが行われる。

SchedulingTDMA のフレーム 3, スロット 3 では、ノード 3 が自身の送受信スケジュールのなかで、送信にも受信にも予約されていないスロットのうち最もスロット番号の小さいものである、スロット 1 を送信スロットとして予約し、自身の送受信スケジュールを隣接ノード 8 に送信する。これを受信したノード 8 で

は、そのときまでにフルフラッシングにより自身の 1 フレーム当りのスロット数が 2 スロットよりなることを知っているため ($slots = 2$)、スロット 1 を送信スロットとして予約したもののスロット数は、1 とはしない。更に隣接ノードへと中継するために送信バッファに収めて待機する (図 8)。

このようにして、SchedulingTDMA のフレーム 3 においては、ノード 3 の送信スロット予約とその通知 (放送ゾーン内の全ノードの送信抑制) が行われる。

同様に、SchedulingTDMA のフレーム 4, スロット 4 では、ノード 4 がスロット 3 を送信スロットとして予約し、これを隣接ノードに伝える。このとき 1 フレームを構成するスロット数は 3 スロットとして更新される (図 9)。

以上のようにして、SchedulingTDMA のフレーム 10, スロット 10 では、各々のノードがその送受信スケジュールのなかに一つの送信スロットをもち、また送

信スロットにも受信スロットにも予約されていない未割当スロットを含んだ、スケルトンスケジュールを完成させる (図 10)。

このとき、最後に送信スロットを予約したノード 10 とその隣接ノードであるノード {8,9} は、送受信スケジュールの 1 フレームが 6 スロットより構成されることを知り、これが最終的に決定された、すべてのノードで共有される 1 フレーム当りのスロット数として、すべてのノードに伝えられる。

ここからは SchedulingTDMA のフレーム 11 スロット 1 でノード 10 が、フレーム 12 スロット 2 でノード 9 が、という順で未割当スロットを獲得していく。すべてのノードは自身の送受信スケジュールの 1 フレームが 6 スロットより構成されることを知るため、最終的に SchedulingTDMA のフレーム 20, スロット 10 を完了した時点で、図 11 に示されるような、1 フレーム 6 スロットからなる送受信スケジュールを出力する。

4.4 提案アルゴリズムの正当性

本アルゴリズムが出力するのは衝突の発生しない送

		node #									
		1	2	3	4	5	6	7	8	9	10
slot #	1	R	-	R	-	-	-	R	-	-	-
	2	-	R	-	-	R	-	-	-	-	-
	3	-	-	-	R	-	-	-	-	-	R
	4	-	-	-	-	-	R	-	-	-	-
	5	-	-	-	-	-	-	-	R	-	-
	6	-	-	-	-	-	-	-	-	-	R
	7										
	8										
	9										
	10										

図 10 図 1 での送受信スケジュール構成の例

(SchedulingTDMA: frame=10, slot=10)

Fig. 10 Schedule example of Fig.1 in proposed algorithm (SchedulingTDMA: frame=10, slot=10).

		node #									
		1	2	3	4	5	6	7	8	9	10
slot #	1	R	-	R	-	-	-	R	-	-	-
	2	-	R	R	-	R	-	-	-	-	-
	3	-	-	R	R	-	-	-	-	-	R
	4	R	-	-	-	-	R	-	-	-	-
	5	-	R	-	-	-	-	-	R	-	-
	6	-	-	-	R	-	-	-	-	-	R

図 11 図 1 での送受信スケジュール構成の例

(SchedulingTDMA: frame=20, slot=10)

Fig. 11 Schedule example of Fig.1 in proposed algorithm (SchedulingTDMA: frame=20, slot=10).

受信スケジュールであることを示す。

[定理 1] 提案アルゴリズムが構成する送受信スケジュールでは、パケットの衝突は起こらない。

(証明) ノード p より 2 ホップ先までの任意のノードにノード p の送受信スケジュールを伝送するには、ノード p が自身の送受信スケジュール sc_p を任意の隣接ノード $q(q \in V, q \neq p)$ に送信し、これをノード q が中継送信することとなるが、ノード p が送信を行ってからノード q が中継を行うまでに必要となる SchedulingTDMA のスロット数 d は、次のように求められる。

図 12 のように、ノード p と q の番号の大きさによって、SchedulingTDMA のフレーム p スロット p にてノード p が自身の送信スロットを送受信スケジュール中に予約し、これを送信したあとで同じフレーム p のスロット q にてノード q が中継可能な場合と ($p < q$)、ノード p が自身の送受信スケジュールを送信した次のフレーム $p+1$ のスロット q でノード q が中継可能な場合 ($p > q$) の二つに分けて考える。すると中継に必要な SchedulingTDMA のスロット数は次のように求めることができる。

$$d = \begin{cases} q - p & \text{if } q > p \\ n + q - p & \text{if } q < p \end{cases}$$

d の値はただか $n - 1$ であるから、ノード p が 2 ホップ先までの全ノードにその送受信スケジュールを送信するためには、 $n - 1$ スロットあれば十分である。ここで、ノード i は

$$scTDMAslot = scTDMAframe = self_i$$

のとき送受信スケジュール sc_i 中に送信スロットを決定するため、ノード $i + 1$ が自身の送受信スケジュー

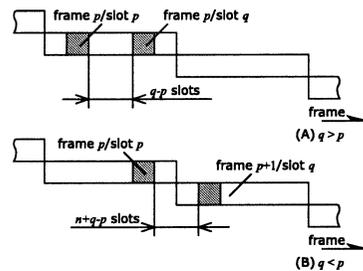


図 12 2 ホップ先のノードまでに送受信スケジュールを伝送するまでに必要なスロット数

Fig. 12 The number of slots for informing the schedule of a node to the nodes 2 hops apart.

ル sc_{i+1} 中に送信スロットを決定するのは、ノード i が送信スロットを決定してから n スロット後であることに注意されたい。このことから、送受信スケジュール中の送信スロットを決定するノードは、そのノードより 2 ホップ先までの全ノードがどのスロットで送信を行うかを知っているため、どのスロットで送信を行えば衝突が起こらないかを判断できる。

よって、提案アルゴリズムはパケット衝突の発生しない送受信スケジュールを構成することができる。□

4.5 提案アルゴリズムの効果

E&T のアルゴリズムで求められる送受信スケジュールは、必ずノード数 n 個のスロットで 1 フレームを構成するが、筆者らの提案アルゴリズムで求められる送受信スケジュールでは、 n スロット以下の送受信スケジュールを求めることができる。本節ではこのことを証明する。

まず、次の命題 1 が成り立つことは提案アルゴリズムの作り方により容易にわかる。

[命題 1] グラフ G の scheduled node set とは、送受信スケジュール上にてスロット i で送信を行うノードの集合をいい、これを S_i で表す。各 i ($1 \leq i \leq n$) に対して、グラフ G の scheduled node set S_i に新しく加えられるノードは、その scheduled node set S_i に属するどのノードからも 2 ホップ以上離れているノードである。

[定理 2] 提案アルゴリズムは、任意のノード $v \in V$ に対して、 v の放送ゾーンが v 以外の全ノードであるようなグラフ ($C_v = V - \{v\}$) を満たすようなグラフを与えたとき、かつそのときに限り、 n スロットの送受信スケジュールを求める。

(証明) 任意のノード $v \in V$ に対して、 $C_v = V - \{v\}$ を満たすようなグラフ G を提案アルゴリズムに与える。命題 1 より、スケジュール決定時においてすべての i ($1 \leq i \leq n$) に対して、 $S_i = \{i\}$ となる。よってこの場合、提案アルゴリズムは必ずスロット n の送受信スケジュールを出力する。

一方、提案アルゴリズムが n スロットの送受信スケジュールを求めるのは、スケジュール決定時において、すべての i ($1 \leq i \leq n$) について $S_i = \{i\}$ となるときかつそのときに限る。ここで、命題 1 より、 $S_i = \{i\}$ となるのは、 $V - \{i\}$ 中のどのノードもノード i から 2 ホップ以内にあるときであるから、提案アルゴリズムに与えられるのは任意の $v \in V$ に対して $C_v = V - \{v\}$ を満たすようなグラフである。□

定理 2 より次の系が成り立つ。

[系 1] 提案アルゴリズムは n スロット以下からなるフレームの送受信スケジュールを求めることができる。

5. E&T のアルゴリズムとの比較

E&T のアルゴリズムと筆者らの提案するアルゴリズムの違いをよりめいりょうにするため、計算機を用いてノードが 5, 10, 20, 50, 100 個からなるグラフを、グラフの疎密さ別にそれぞれ 10,000 通りランダムに生成し、これに E&T のアルゴリズムと筆者らの提案するアルゴリズムを適用した。

PRN の形状を表す自然な尺度として、ここではノード数と疎密さを採用する。ここでのグラフの疎密さとは、任意のノードとノードの間のリンクの切断率が 0~1 までの値で表したものであり、切断率が 0 のときすべてのノードとノードの間にリンクが生成されるため、PRN は完全グラフとなり、切断率が 1 のとき、すべてのリンクが切断された PRN となる (すべてのノードは孤立ノードとなる)。

5.1 PRN の疎密さと送受信スケジュールを構成するスロット数の関係

図 13 は、縦軸に各アルゴリズムが出力する送受信スケジュールの 1 フレームを構成するスロット数を取り、PRN のノード数 (5, 10, 20, 50, 100 ノード) ごとに E&T のアルゴリズムによるもの (破線) と提案アルゴリズムによるもの (実線) を描いたものである。このことより、PRN が密であると、提案するアルゴリズムと E&T のものとの比較して送受信スケジュールの 1 フレームを構成するスロット数はほぼ同じであるが、PRN がより疎になると、これがより少

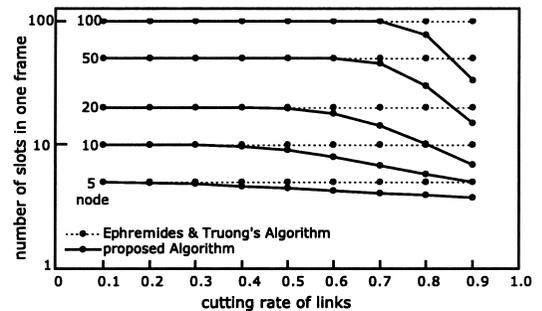


図 13 PRN の疎密さと送受信スケジュールを構成するスロット数の関係

Fig. 13 The relation between the density of PRNs and slots per frame of composing the schedule.

なくなることがわかる。

これは、PRN が密であるほど放送ゾーン（自身を除いた 2 ホップ先までの全ノードの集合）に含まれるノード数が多くなるため並行送信が難しくなり、そのため送受信スケジュールの 1 フレームを構成するスロット数を少なくすることが難しくなるためである。結果として、提案アルゴリズムが求めるスケルトンスケジュールと、E&T のアルゴリズムによって求められるものとの差がなくなってくる。逆に PRN が疎であるほど、並行送信の機会が多くなることによって、提案アルゴリズムは求める送受信スケジュールの 1 フレーム当りのスロット数をより少なくすることができる。

また、どのノード数の場合においても、切断率がある値を超えた時点から、提案アルゴリズムのほうが E&T のアルゴリズムよりも 1 フレームのスロット数が少なくなっていくことが観察できる。このときの切断率を a とすると、PRN のノード数が多くなるにつれて、 a も大きくなっている。これは、ノード数が少ないほうが、あるノードに対してその放送ゾーンに含まれるノードの個数が少なくなるため、並行送信が起こりやすくなるからである。

5.2 PRN の疎密さと送受信スケジュールの送信スロット割当ての偏りの関係

次に、表 1、表 2 に E&T のアルゴリズムと提案アルゴリズムとの間で、ノードごとの送信スロットの割当てにどれだけ偏りがあるかを比較したものを示す。送信スロットの割当ての偏りは、それぞれのアルゴリズムで求められた送受信スケジュールに含まれる、各ノードの送信スロット割当ての分散で表現したものである。

例えば、図 1 に示す PRN の場合、E&T のアルゴリズムで与えられる送受信スケジュールは図 4 となるが、このときの各ノードが得る送信スロット数は、ノード 1 が 5 個、ノード 2 が 3 個、ノード 3 が 7 個、... となり、送信スロット数の割当ての分散は約 3.95 となる^(注3)。これに対して提案アルゴリズムでは、図 11 の送受信スケジュールが与えられ、この分散は 0.5 となる。よって、提案アルゴリズムは E&T のものと比較して分散は $0.5/3.95 = 0.13$ 倍と改善されていることがわかる。

表 1 に示すのは、この方法により生成した 10 ノードのグラフでの、各アルゴリズムでの分散を比較したものである。これより、E&T のアルゴリズムでは

表 1 10 ノードからなる PRN での送信スロット割当ての分散

Table 1 The deviations of reserved slots in PRNs of 10 nodes.

cutting rate of links	deviations of E&T's algorithm (A)	deviations of proposed algorithm (B)	ratio of deviations (B/A)
0.1	0.000018	0.000000	0.000000
0.2	0.000383	0.000001	0.026110
0.3	0.013719	0.001998	0.145637
0.4	0.116088	0.030701	0.264463
0.5	0.495152	0.143966	0.290751
0.6	1.369229	0.362798	0.264965
0.7	2.669936	0.522796	0.195808
0.8	4.148788	0.548278	0.132154
0.9	5.439538	0.473708	0.087086

表 2 ノード数と送受信スケジュールの送信スロット割当ての偏りの関係

Table 2 The relation between the nodes and deviations of reserved slots.

nodes	max ratio of deviations
5	0.1733080
10	0.2907511
20	0.3431011
50	0.1523668
100	0.2656071

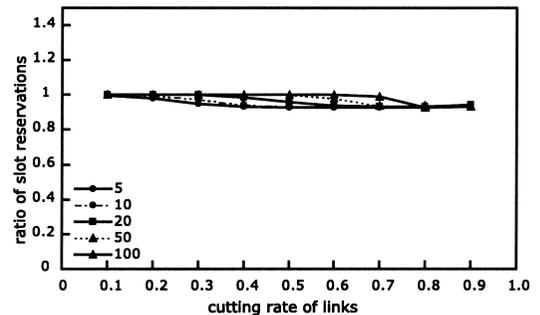


図 14 グラフの疎密さと送受信スケジュールの送信スロット割当て比の関係

Fig. 14 The relation between the density of graphs and ratio of slot reservations.

PRN が疎になるほど各ノードごとの送信スロット割当ての分散が極めて大きくなるのと比較して、提案アルゴリズムでは分散の増加がおさえられており、その値は最大でも E&T のアルゴリズムの分散の値の約 30%（疎密さが 0.5 のとき）である。

このようにしてノード数を 5, 10, 20, 50, 100 として PRN の疎密さを変えながら、送信スロット割当ての偏りを求めた結果を表 2 に示す。表には疎密さを

(注3): $\{(5^2 + 3^2 + 7^2 + 2^2 + 2^2 + 1^2 + 4^2 + 1^2 + 2^2 + 1^2) \times 10 + (5 + 3 + 7 + 2 + 2 + 1 + 4 + 1 + 2 + 1)^2\} / \{10 * (10 - 1)\} = 3.95$.

```

1:  type
2:      dec: {'R', '-', null };
3:      schedule: array[1,...,n] of dec;
4:      BUFFER = record
5:          sc : schedule;
6:          tnode : integer;
7:          node : integer;
8:          slots : integer;
9:      end;
10:
11:  var
12:      sc : schedule;
13:      self : integer;
14:      slots : integer;
15:      scTDMAslot, scTDMAframe : integer;
16:      insc, outsc : BUFFER;
17:
18:  { main scheduling loop }
19:  begin
20:      slots:=0;
21:      sc:=null;
22:      insc.sc:=null; insc.tnode:=insc.snode:=insc.slots:=0;
23:      outsc.sc:=null; outsc.tnode:=outsc.snode:=outsc.slots:=0;
24:      { make "the skeleton schedule" }
25:      for scTDMAframe:=1 to n do
26:          begin
27:              for scTDMAslot:=1 to n do
28:                  begin
29:                      if scTDMAslot=self then schedulerTX();
30:                      else schedulerRX();
31:                  end;
32:              end
33:              { reserve the non-reserved slot(s) }
34:              for scTDMAframe:=(n+1) to 2n do
35:                  begin
36:                      for scTDMAslot:=1 to n do
37:                          begin
38:                              if (n+1-scTDMAslot)=self then schedulerTX();
39:                              else schedulerRX();
40:                          end;
41:                      end
42:                  end;
43:              { broadcast own schedule, or hookup the neighbor's schedule }
44:              procedure schedulerTX
45:              begin
46:                  if scTDMAframe=self then
47:                      begin
48:                          for i:=1 to n do
49:                              begin
50:                                  if sc[i]=null then
51:                                      begin
52:                                          sc[i]:='R';
53:                                          break;
54:                                      end;
55:                                  end;
56:                              for i:=1 to n do
57:                                  begin
58:                                      if sc[i]!=null and i>slots then slots:=i;
59:                                  end;
60:                                  outsc.sc:=sc;
61:                                  outsc.snode:=outsc.tnode:=self;
62:                                  outsc.slots:=slots;
63:                              end;
64:                              else if (scTDMAframe-(2 scTDMAslot)+1)=self then
65:                                  begin
66:                                      for i:=1 to slots do
67:                                          begin
68:                                              if sc[i]=null then sc[i]:='R';
69:                                          end;
70:                                          outsc.sc:=sc;
71:                                          outsc.snode:=outsc.tnode:=self;
72:                                          outsc.slots:=slots;
73:                                      end;
74:                                      broadcast(outsc);
75:                                      outsc.sc:=null; outsc.snode:=outsc.tnode:=outsc.slots:=0;
76:                                  end;
77:                              { listen the schedule which is 1-hop or 2-hops neighboring node's }
78:                              procedure schedulerRX
79:                              begin
80:                                  insc.sc:=null; insc.snode:=insc.tnode:=insc.slots:=0;
81:                                  listen(insc);
82:                                  if insc.tnode!=0 then
83:                                      begin
84:                                          if insc.snode!=self then
85:                                              begin
86:                                                  for i:=1 to n do
87:                                                      begin
88:                                                          if insc.sc[i]='R' then sc[i]:='-';
89:                                                      end;
90:                                                      if insc.tnode=insc.snode then
91:                                                          begin
92:                                                              outsc:=insc;
93:                                                              outsc.tnode:=self;
94:                                                          end
95:                                                      end;
96:                                                  if slots<insc.slots then
97:                                                      slots:=insc.slots;
98:                                                  outsc.slots:=slots;
99:                                                  end;
100:                                     end;

```

<i>n</i>	the number of nodes.
<i>scTDMAframe</i> & <i>scTDMAslot</i>	value that specifies the current frame and slot.
<i>self</i>	the identification number of each node.
<i>sc</i>	an array that specifies the schedule in which slot the node is able to the transmit
<i>slots</i>	contains the number of slots of current frame.
<i>insc</i> & <i>outsc</i>	buffer of array which specifies schedule: 1), <i>sc</i> , that contains current schedule, 2), <i>slots</i> , that contains the number of slots of current frame, 3), <i>snode</i> , that represents which node's schedule it is, and that has been received from a node of 4), <i>tnode</i> .

図 15 提案アルゴリズム
Fig.15 Proposed algorithm.

変えたときの分散値の最大値を示している．例えば 10 ノードのときの値 0.290751 は，表 1 の分散値 (B/A) の最大値に一致している．このことから，提案アルゴリズムにより求められる送信スロット割当数の偏りは，E&T のものと比較して 35% 以下におさえられていることがわかる．

アルゴリズムのつくりから，E&T のアルゴリズムで求められるスケルトンスケジュールでは，筆者らの提案アルゴリズムと比較して送信にも受信にも予約されていない未割当スロットがより多く含まれている．このことが後に予約される送信スロット数に大きなばらつきを生じさせる原因となる．これに対して，筆者らの提案するアルゴリズムでは，1 フレーム当りのスロット数を少なくすることで，未割当スロットを多く含まないスケルトンスケジュールを求めることに成功している．これが送信スロット割当ての偏りを軽減できている理由である．

5.3 PRN の疎密さと送受信スケジュールの送信スロット割当数の関係

最後に，図 14 に E&T のアルゴリズムと提案アルゴリズムでの，送信スロットの割当数を比較したものを示す．ただしここでは，提案アルゴリズムで与えられる送受信スケジュールの 1 フレームを構成するスロット数が，E&T のものと比較してより少なくなることから，両者を公平に比較するために，次のような計算を行った．

例として，図 1 の PRN を用いる．このとき E&T のアルゴリズムで与えられる送受信スケジュールは図 4 となり，各ノードに割り当てられた送信スロット数の総和 (すなわち “R” の個数) は 28 個である．これに対して提案アルゴリズムで得られた図 11 の送受信スケジュールでは，各ノードに割り当てられた送信スロット数の総和は 15 個である．ここで E&T の送受信スケジュールは 1 フレームが 10 スロットで，提案アルゴリズムは 1 フレームが 6 スロットより構成されるが，この最小公倍数である 30 スロットの TDMA を行った場合に換算すると，E&T のアルゴリズムでは $28 \times (30/10) = 84$ 個の送信スロット，提案アルゴリズムでは $15 \times (30/6) = 75$ 個の送信スロットとなる．これは E&T のアルゴリズムと比較して，約 0.89 倍 (= $75/84$) の送信スロット割当数が得られるということである．

ノード数が 5, 10, 20, 50, 100 の PRN を各々 10,000 個発生させ，それらの PRN について上記の送

信スロット割当ての比較を行った．これをまとめたものを図 14 に示す．

この結果，PRN の疎密さが変化しても送信スロット数の割当ての比はほぼ 1 となっており，筆者らのアルゴリズムと E&T のものでは，割り当てることができる送信スロット数にほとんど差がないことが確認できる．

6. む す び

本論文では特定の基地局を設けることなく，各ノードの 1 フレーム当り最低 1 回の送信スロット割当てを保証し，かつこれを，E&T のアルゴリズムではノードと同じ個数のスロットで 1 フレームを構成していたのに対して，ノード数以下のスロット数で 1 フレームを構成する送受信スケジュールを求める自律分散型 TDMA 放送計画アルゴリズムを示した．また，これにより E&T により問題点として指摘されていた各端末の送信スロット割当数の偏りについて大きく改善しながらも，そのスロット数は同程度得られることを示した．

今後は本論文で提案したアルゴリズムをもとにして，ノードの参加，離脱を考慮して送信スロットの動的割当機能を盛り込んだ自律分散型動的 TDMA 放送計画アルゴリズムについて検討していく予定である．

文 献

- [1] F.A. Tobagi and L. Kleinrock, “Packet Switching in Radio Channels: Part II-The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution,” IEEE Trans. Commun., vol.COM-23, no.12, 1975.
- [2] P. Karn, “MACA - A new channel access protocol for packet radio,” ARRL/CRRL Amateur Radio Ninth Computer Networking Conf., pp.134-140, 1990.
- [3] A.S. Tanenbaum, Computer networks (International Third Edition), Prentice-Hall International, 1996.
- [4] T. Truong, “TDMA in mobile radio networks : graph partitioning and time slot assignment,” Proc IEEE MILCOM Conf., Oct. 1985.
- [5] I. Cidon and M. Sidi, “Distributed assignment algorithms for multi-hop packet-radio networks,” IEEE Trans. Comput., vol.38, no.10, pp.1353-1361, 1989.
- [6] I. Chlamtach and A. Lerner, “Distributed assignment algorithm for multi-hop radio networks,” Dep. Comput. Sci., Technion, Israel, Tech. Rep. 363, Jan. 1985.
- [7] A. Ephremides and T.V. Truong, “Scheduling Broadcasts in multihop Radio networks,” IEEE Trans. Commun., pp.456-460, vol.38, no.4, April 1990.
- [8] 染矢克寿, 福羅充明, 松野浩嗣, “無線網における放送計画問題の複雑さとアルゴリズム” マルチメディア, 分散, 強

調とモバイル(DiCoMo)ワークショップ , pp.203-208, 1997.

- [9] K. Sayrafian-Pour and A. Ephremides, "Interference-free Time-frequency Broadcast Scheduling in Multihop Packet Radio Networks," IEEE Wireless Communications and Networking Conference, in CD-ROM, 2000.

(平成 14 年 9 月 27 日受付, 12 月 9 日再受付)



小原 誠 (学生員)

平 11 大島商船高専卒 . 平 13 山口大・理・自然情報卒 . 平 13 より同大学院修士課程在籍 . 計算機ネットワーク構築技術の研究に従事 . 現在 (株) 東芝に勤務 . 平 12 本会中国支部奨励賞受賞 .



松野 浩嗣 (正員)

昭 57 山口大・工・電子卒 . 昭 59 同大学院修士課程了 . 昭 59 ~ 62 山口短期大 , 昭 62 ~ 平 6 大島商船高専勤務 . 平 7 より山口大・理・助教授 . 計算機ネットワーク構築技術の研究と計算機によるゲノム解析支援の研究に従事 . 理博 . 情報処理学会 ,

IEEE 各会員 .