

無線ネットワークにおける完了確認付 ブロードキャストアルゴリズムについて

大桑 貴哉[†], 今井 良樹[‡], 伊藤 暢浩[†], 陳 慰[‡], 和田 幸一[†]

[†] 名古屋工業大学 電気情報工学科

[‡] 南山大学数理情報学部

[‡] 日本テクシード

あらまし

データの送受信機能を持つデバイスを節点と呼び、節点の集合により構成されたネットワークを無線ネットワークと呼ぶ。ARB (acknowledged radio broadcasting) とはソースが持っているメッセージを無線ネットワーク中の全節点に伝達し、その完了をソースが確認できることである。本稿で扱う無線ネットワークのモデルは、各節点はデータの送信範囲が等しく、ラウンド単位で各節点は同期をとり、送信もしくは受信を1ラウンドで行う。また、各節点は衝突検出機能があり、自身のIDのみを知っている。文献 [1] では n を無線ネットワークの節点数とすると、 $O(n)$ ラウンドで ARB を解くアルゴリズムが示されている。本稿では r をソースメッセージのビット長、 ecc をソースからの最大距離とすると、 $O(r \cdot ecc)$ ラウンドで ARB を解くアルゴリズムを示す。

An acknowledged radio broadcasting algorithm in radio networks

Takaya Ohkuwa[†], Yosiki Imai[‡], Nobuhiro Itoh[†], Wei Chen[‡] and Koichi Wada[†]

[†]Nagoya Institute of Technology

[‡]Nanzan University

[‡]Nippon Tecseed

abstract

A radio network is a collection of transmitter-receiver devices referred to as nodes. In radio broadcasting(RB) the goal is simply to communicate the source message to all nodes. In acknowledged radio broadcasting(ARB) the goal is to achieve RB and inform the source about it. This paper, considers the following model of radio networks. Each node has the equal transmitting range, all the nodes are synchronized every round in which each node acts either as transmitter or as a receiver in every round, each node has a collision detection function, and its priori knowledge is limited to its own ID. It is shown in [1] that ARB is accomplished in $O(n)$ rounds, where n is the number of nodes of the network. In this paper, we show that ARB is accomplished in $O(r \cdot ecc)$ rounds, where r is the bit length of a source message, and ecc is the maximum distance from the source.

1 まえがき

無線ネットワークは送受信機能を持つデバイス(節点と呼ぶ)の集合として定義される。各節点は送信能力の及ぶ範囲内にある節点にデータを送信できる。無線ネットワークは到達グラフと呼ばれる有向グラフ $G = (V, E)$ でモデル化される。ここで V は節点の集合であり、節点 u から節点 v へ送信可能であるとき $(u, v) \in E$ と定義する。 $(u, v) \in E$ のとき u から v へ隣接しているという。

無線ネットワークにおける各節点は地域時計に同期してラウンド単位で動作する。各ラウンドで節点はデータを送信するか、もしくは受信するかのいずれかの動作をする。各節点が無線ネットワークに関して知っている情報は自身の ID のみと仮定する。節点 u が受信機として動作しており、節点 u へのちょうど 1 つの隣接節点 v がデータを送信しているとき節点 u は節点 v の送信したデータを受信することができる。しかし、節点 u への 2 つ以上の隣接節点が同時にデータを送信するとき、節点 u はデータを正しく受信することができない。節点 u への隣接節点のいずれもがデータを送信しない場合と節点 u への 2 つ以上の隣接節点がデータを送信した場合とが区別できるとき、衝突検出機能を持つという。

無線ネットワークにおける基本的な問題には、特定の節点(ソース)からネットワーク内の全ての節点へデータを伝達するブロードキャスト (radio broadcasting:RB) 問題がある。また、RB を行い RB が完了したことをソースが検知する問題を完了確認付ブロードキャスト (Acknowledged radio broadcasting:ARB) 問題という。ソースが繰り返し複数のデータをブロードキャストする場合、ソースは各ブロードキャストの終了を知る必要があるため、ARB は重要である。

文献 [1] では上記の無線ネットワークに対して衝突検出機能を持たないモデルと持つモデルの上で RB と ARB に対する決定性アルゴリズムを示した。主な結果を図 1 に示す。ここで、 n は節点の総数、 ecc はソースからの最大距離、 r はソースメッセージのビット長である。また、対称有向グラフ $G = (V, E)$ は、任意の 2 点 $u, v (u, v) \in V$ に対して $(u, v) \in E \iff (v, u) \in E$ となるグラフである。また、衝突検出機能を持たない場合対称有向グラフでさえ ARB に対するアルゴリズムは存在しないことが示されている。RB に対しては [1] の結果はさらに改良されていることが報告されている ([2],[3],[4])。

本稿では、衝突検出機能を持つ無線ネットワークのモデル上で、 r を RB データのビット長、 ecc をソースからの最大距離とすると、対称有向グラフに対して $O(r \cdot ecc)$ ラウンドで ARB を解く決定性アルゴリズムを示す。 r や ecc が小さい ($r \cdot ecc = o(n)$) 場合、本稿のアルゴリズムは従来の結果を改善する。

問題	衝突検出	グラフ	実行時間
RB	無し	対称有向	$O(n)$ [1]
		任意	$O(n^{\frac{11}{6}})$ [1]
		任意	$O(n^{\frac{5}{3}}(\log n)^{\frac{1}{3}})$ [4]
		任意	$O(n^{\frac{3}{2}})$ [2]
	任意	$O(n(\log n)^2)$ [3]	
有り	強連結	$O(n \cdot ecc)$ [1]	
	任意	$O(r \cdot ecc)$ [1]	
ARB	無し	対称有向	アルゴリズムが存在しない [1]
		対称有向	$O(n)$ [1]
	有り	強連結	$O(n \cdot ecc)$ [1]

図 1 : モデルと実行時間 (n : 節点の総数、 ecc : ソースからの最大距離、 r : RB データのビット長)

2 モデル

データの送受信機能を持つデバイスを節点と呼ぶ。節点の集合で構成されたネットワークのことを無線ネットワークと呼ぶ。全ての節点は、地域時計でラウンド単位で同期して動作するものとする。1 ラウンドで節点はデータの送信、受信をする。また、送受信の前処理である送信機か受信機どちらで起動するかの判断や受信したデータによる処理はそのラウンド内で行われるものと仮定する。

本稿では、ソースのみがアルゴリズム開始ラウンドを知ってるものとし、ソース以外の節点は、アルゴリズムの開始ラウンドを知らないと仮定する。また、各節点は自身がソースであるか否かを知っているものとし、ID は必ずしも全節点異なる必要はない。文献 [1] では全ての節点がアルゴリズムの開始ラウンドを知っている必要がある。また各節点の ID は全て異なり、 $ID = O(n)$ であることが必要である。この仮定を本稿のモデルに拡張した場合、文献 [1] のアルゴリズムは本質的に実行できない。

各節点は送信能力の及ぶ範囲内にある節点にデータを送信できる。無線ネットワークは到達グラフと呼ばれる有向グラフ $G = (V, E)$ でモデル化される。ここで V は節点の集合であり、節点 u から節点 v へ送信可能であるとき $(u, v) \in E$ と定義する。本稿では、全ての節点の送信範囲が等しい場合を考える。この場合、 G は対称有向グラフとなる。特に対称有向グラフ $G = (V, E)$ において、 $(u, v) \in E$ のとき u と v は隣接するという。

また、節点は衝突検出機能を持つものと仮定する。節点が隣接節点からメッセージを正しく受信した場合もしくは衝突を検知した場合、その節点はシグナル μ を聞くという。節点が受信機として働き、その隣接節点が発信しないときすなわち、衝突が発生せず、メッセージを受信しなかった場合、その節点はシグナル λ を聞くという。本稿では、メッセージの受信にはシグナル μ とシグナル λ を区別するだけでよいので、送信データは 1 ビットで十分である。1

ビットのメッセージをコンタクトメッセージと呼ぶ。

本稿では、上記の無線ネットワークのモデルに対して完了確認付無線ブロードキャスト (ARB) 問題を取り扱う。また、 $G = (V, E)$ における 2 つの節点 $u, v (v \in V)$ の距離は、その 2 つの節点を結ぶ経路の中で最短の経路における辺の数で定義される。 $G = (V, E)$ において s から各節点の距離の中で最大の距離を ecc で表す。

3 RB アルゴリズム

ソースメッセージのビット列を (a_1, a_2, \dots, a_r) と表す。ソースメッセージの送受信には、以下のようなエンコード法を用いる。[1]

1 ビットを送信するのに 2 ラウンド使用する。 $a_i = 0$ を送信する場合は、2 ラウンドとも送信しない。 $a_i = 1$ を送信する場合は 1 ラウンド目は送信をしないで、2 ラウンド目にコンタクトメッセージを送信する (図 2 参照)。よって、受信側の節点は 2 ラウンドともシグナル λ を聞いたとき、 $a_i = 0$ であり、1 ラウンド目にシグナル λ を聞いた後に次のラウンドでシグナル μ を聞いたとき $a_i = 1$ である (図 3 参照)。ソースメッセージの送受信には $2r$ ラウンド必要となる。ソースメッセージ送信の開始と終了合図としてそれぞれ 2 ラウンドずつ送信する。

したがって、受信節点は、ソースメッセージ受信開始の合図としてシグナル μ を 2 回聞き、次の $2r$ ラウンドでソースメッセージを受信して、ソースメッセージ受信終了の合図としてシグナル λ を 2 回聞き、受信は完了する。すなわち、ソースメッセージの送受信には $2r + 4$ ラウンド必要となる。

次に、エンコード法を用いた RB アルゴリズム (RBEM と呼ぶ) の概略を示す。ソースがエンコード法によりソースメッセージの送信を行い、ソースからの距離が 1 である節点がソースメッセージを受信する。次に、ソースからの距離が 1 である節点がエンコード法によりソースメッセージの送信を行い、ソースからの距離が 2 である節点がソースメッセージを受信する。以後、初めてソースメッセージを受信した節点が、ソースメッセージを送信することを繰り返す。これにより、ソースを中心とした同心円状に $2r + 4$ ラウンドごとにソースメッセージが伝達される。

RBEM において、送信はコンタクトメッセージを送信、未送信は送信機のものにも送信しないことを表す。受信 (R) とは、受信機として働きシグナル $R (\in \{\lambda, \mu\})$ を聞くことを表す。各節点は 1 ラウンドで必ず送信、未送信、受信 (R) のどれか 1 つを行う。RBEM で用いている変数として、 r はソースメッセージのビット長を表し、 $a = (a_1, a_2, \dots, a_r)$ はソースメッセージを表す。以下の RBEM は RB を $O(r \cdot ecc)$ ラウンドで実現している。[1]

送信ビット・合図	1 ラウンド	2 ラウンド
0	未送信	未送信
1	未送信	送信
ソースメッセージ 送信開始・終了合図	送信	送信

図 2 : エンコード法による送信

1 ラウンド	2 ラウンド	受信ビット・合図
シグナル λ	シグナル λ	0
シグナル λ	シグナル μ	1
シグナル μ	シグナル μ	ソースメッセージ 受信開始・終了合図

図 3 : エンコード法による受信

アルゴリズム RBEM

```

ソース {
  SendMessage;
}

ソース以外の節点 {
  ReceiveMessage(a, r);
  SendMessage;
}

SendMessage {
  送信 ; 送信 ;
  for (i = 1; i ≤ r; i++)
    if (a_i == 0)
      { 未送信 ; 未送信 ; }
    else if (a_i == 1)
      { 未送信 ; 送信 ; }
  送信 ; 送信 ;
}

ReceiveMessage(a, r) {
  受信 (R) ;
  while (R == λ) {
    受信 (R) ;
    if (R == μ)
      受信 (R) ;
  }
  i = 0;
  受信 (R_1) ; 受信 (R_2) ;
  while not (R_1 == μ & R_2 == μ) {
    i = i + 1;
    if (R_1 == λ & R_2 == λ)
      a_i = 0;
    else if (R_1 == λ & R_2 == μ)
      a_i = 1;
    受信 (R_1) ; 受信 (R_2) ;
  }
  r = i;
}

```

4 ARB アルゴリズム

4.1 ARB アルゴリズムの概略

今回考案した ARB アルゴリズム (ARBEM と呼ぶ) は 2 つのステージ A、B から構成される。ステージ A では節点を 3 種類に分類することを目的とする。実際には、各節点がソースから自身への距離を 3 で割った余りを認識する。この節点の分類によりステージ B で「終了状態」であることを各節点が衝突なく隣接節点かつソースからの距離が自身の値より 1 小さい節点に伝えることができる。

ステージ B は複数のフェイズで構成され、各フェイズは $2r+10$ ラウンドから成る。ステージ B はソースがソースメッセージをエンコード法により送信し、ソースからの距離が 1 の節点がソースメッセージを受信することで開始される。以後ソースメッセージを初めて受信した節点がフェーズの前半部 ($2r+4$ ラウンド) でエンコード法により、ソースメッセージを送信する。したがって RB の場合と同様に同心円状に沿ってソースメッセージが伝達される。ソースからの距離が ecc の節点はソースメッセージを受信するとソースメッセージの受信を終了したことを認識し、その情報を逆順にソースまで伝達することにより、ソースがすべての節点がソースメッセージを受信したことを知る。この伝達はソースからの距離を 3 で割った余りの情報を用いてフェーズの後半部 (6 ラウンド) で行われる。

4.2 各節点の動作

ARBEM において、送信、未送信、受信 (R) は先程と同様であり、各節点は 1 ラウンドで必ず送信、未送信、受信 (R) のどれか 1 つを行う。変数として、 mod はソースからの距離を 3 で割った余りを表し、 $state$ は節点の状態を示す。以下に具体的なアルゴリズムを示す。SendPhase, NoopPhase はそれぞれ $2r+10$ ラウンドで構成される。ReceivePhase はソースメッセージ送信開始合図を受信したラウンドからは $2r+10$ ラウンドを要する。ソース以外の節点はソースメッセージを受信するまでは ReceivePhase の STEP1 の *ReceiveMessage* を実行している。ソースメッセージを受信した節点は ReceivePhase の STEP2 以降になると、他の節点が行っている SendPhase と NoopPhase の各 STEP で同期を取る。

アルゴリズム ARBEM

```
ソース {
// ステージ A
  mod = 0;
  送信; 未送信; 未送信;
  for(i = 4; i ≤ 9; i++)
    未送信;
//ステージ B
  SendPhase;
  while(state == 未終了)
    NoopPhase;
}
```

```
ソース以外の節点 {
//ステージ A
  受信 (R);
  while(R == λ)
    受信 (R);
  受信 (R1); 受信 (R2);
  if(R1 == λ & R2 == λ)
    mod = 1;
    送信; 送信; 未送信;
    未送信; 未送信; 未送信;
  else if(R1 == μ & R2 == λ)
    mod = 2;
    送信; 送信; 送信;
    未送信; 未送信; 未送信;
  else if(R1 == μ & R2 == μ)
    mod = 0;
    送信; 未送信; 未送信;
    未送信; 未送信; 未送信;
// ステージ B
  RECEIVEPHASE;
  SendPhase;
  while(1)
    NoopPhase;
}
SendPhase {
//STEP1
  SendMessage;
//STEP2
  受信 (R);
  if(R == μ)
    state = 未終了;
  else if(R == λ)
    state = 終了;
//STEP3
  未送信;
//STEP4A
  if(mod == 0 & state == 未終了)
    送信;
  else 未送信;
//STEP4B
  if(mod == 1 & state == 未終了)
    送信;
  else 未送信;
//STEP4C
  if(mod == 2 & state == 未終了)
    送信;
  else 未送信;
//STEP5
  未送信;
}
ReceivePhase {
//STEP1
  ReceiveMessage;
//STEP2
  state = 未終了;
  送信;
//STEP3
  未送信;
```

```

//STEP4A
if(mod == 0 & state == 未終了)
    送信;
else 未送信;
//STEP4B
if(mod == 1 & state == 未終了)
    送信;
else 未送信;
//STEP4C
if(mod == 2 & state == 未終了)
    送信;
else 未送信;
//STEP5
未送信;
}
NoopPhase {
//STEP1
for(i = 1; i ≤ 2r + 4; i++)
    未送信;
//STEP2
未送信;
//STEP3
未送信;
//STEP4A
if(mod == 0 & state == 未終了)
    送信;
else if(mod == 2 & state == 未終了)
    受信 (R);
    if(R == λ)
        state = 終了;
    else 未送信;
//STEP4B
if(mod == 1 & state == 未終了)
    送信;
else if(mod == 0 & state == 未終了)
    受信 (R);
    if(R == λ)
        state = 終了;
    else 未送信;
//STEP4C
if(mod == 2 & state == 未終了)
    送信;
else if(mod == 1 & state == 未終了)
    受信 (R);
    if(R == λ)
        state = 終了;
    else 未送信;
//STEP5
未送信;
}

```

4.3 実行例

図4の無線ネットワークにおいて、ARBEMの実行例を図5に示す。図5は、各節点の各ラウンドにおける動作を示している。送は送信で、未は未送

信、 λ, μ は受信 (R) を行いシグナル λ, μ を聞いたことを表す。mは mod であり、終は state=終了を表す。ソースメッセージは1ビットで0とする。

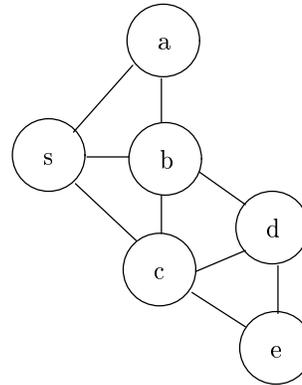


図4：無線ネットワーク (sをソースとする)

4.4 各節点の実行時間

親と子グループの定義を述べる。

- 節点 i の親：ソースから節点 i までの距離を d_i とし、節点 i の親 $A(i)$ を次のように定義する。

$$A(i) = \{ j \mid j \text{ は } i \text{ の隣接節点 かつ } d_j = d_i - 1 \}$$

- 節点 i の子グループ：ソースから節点 i までの距離を d_i とし、節点 i の子グループ $C(i)$ を次のように定義する。

$$C(i) = \{ j \mid (j \text{ は } i \text{ の隣接節点 かつ } d_j = d_i + 1) \text{ または } (j' \in C(i) \text{ かつ } j \in C(j')) \}$$

例えば、図4の無線ネットワークで $A(d) = \{b, c\}$ であり、 $C(a) = \phi$ 、 $C(c) = \{d, e\}$ である。

ソースがステージ A の最初の送信を行うラウンドを1ラウンド目とする。ソースがステージ B の最初の送信を行うラウンド (10ラウンド目) から $2r + 10$ ラウンドを1フェイズ目とする。節点 k のソースからの距離を d_k とする。 $C(k)$ の節点でソースからの距離が最大の値を $d_{max}^k (> d_k)$ とする。

節点がステージ A を終了する時刻

ステージ A では節点は、初めてシグナル μ を聞いたラウンドから3ラウンド受信 (R) を行い、次の3ラウンドは送信 (未送信) を行い、さらに次の3ラウンドは未送信を行う。したがって、3ラウンドごとにソースを中心とした同心円状に沿って送信 (未送信) を行うことになる。ゆえに、ステージ A でソース以外の節点 k が初めてシグナル μ を聞くのは $3(d_k - 1) + 1 = 3d_k - 2$ ラウンド目であり、初めて送信するのは、 $3d_k + 1$ ラウンド目である。また、ステージ A が終了するのは、節点 k が初めてシグナル μ を聞いたラウンドから9ラウンド目であるので $3d_k + 6$ ラウンド目でステージ B が開始されるのは $3d_k + 7$ ラウンド目である。

ラウンド 節点	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
s	$m=0$ 送	未	未	未	未	未	未	未	未	送	送	未	未	送	送	μ
	ステージ A									SendPhaseSTEP1						STEP2
a	μ	λ	λ $m=1$	送	送	未	未	未	未	μ	μ	λ	λ	μ	μ	送
	ステージ A									RecievePhaseSTEP1						STEP2
b,c	μ	λ	λ $m=1$	送	送	未	未	未	未	μ	μ	λ	λ	μ	μ	送
	ステージ A									RecievePhaseSTEP1						STEP2
d,e	λ	λ	λ	μ	μ	λ $m=2$	送	送	送	未	未	未	λ	λ	λ	μ
	ステージ A															

ラウンド 節点	17	18	19	20	21	22	23	24	25	26	27	28	29
s	未	送	未	未	未	未	未	未	送	送	未	未	未
	STEP3	STEP4			STEP5	NoopPhaseSTEP1						STEP2	STEP3
a	未	未	送	未	未	送	送	未	未	送	送	λ 終	未
	STEP3	STEP4			STEP5	SendPhaseSTEP1						STEP2	STEP3
b,c	未	未	送	未	未	送	送	未	未	送	送	μ	未
	STEP3	STEP4			STEP5	SendPhaseSTEP1						STEP2	STEP3
d,e	λ	λ	μ	λ	λ	μ	μ	λ	λ	μ	μ	送	未
	ReceivePhaseSTEP1											STEP2	STEP3

ラウンド 節点	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
s	送	μ	未	未	未	未	未	未	未	未	未	未	送	μ	未
	STEP4			STEP5	NoopPhaseSTEP1						STEP2	STEP3	STEP4		
a	未	未	未	未	未	未	未	未	未	未	未	未	未	未	未
	STEP4			STEP5	NoopPhaseSTEP1						STEP2	STEP3	STEP4		
b,c	未	送	未	未	未	未	未	未	未	未	未	未	未	送	λ 終
	STEP4			STEP5	NoopPhaseSTEP1						STEP2	STEP3	STEP4		
d,e	未	未	送	未	送	送	未	未	送	送	λ 終	未	未	未	未
	STEP4			STEP5	SendPhaseSTEP1						STEP2	STEP3	STEP4		

図5：図4の無線ネットワークの各ラウンドにおける各節点の動作

節点kがstate=終了となる時刻

ステージBはソースが10ラウンド目にSendPhaseのSTEP1のSendMessageを行う。ステージBはRBEMを利用しているのでフェイズごとにソースを中心とした同心円状に沿ってソースメッセージが伝達される。したがって、ソースメッセージを受信(ReceivePhaseのSTEP1-ReceiveMessage終了)するのは、 d_k フェイズ目であり、受信したソースメッセージを送信(SendPhaseのSTEP1-SendMessage終了)するのは($d_k + 1$) フェイズ目である。

アルゴリズムより、state = 終了となるのはSendPhaseのSTEP2もしくはNoopPhaseのSTEP4(STEP4A,4B,4C)である。STEP2とは、ソースメッセージを送信した節点が受信(R)を行い、ソースメッセージを受信した節点が送信を行うので、このSTEPはソースメッセージを送信した節点の子グループが空ならば、そのノードはstate=終了となる。

NoopPhaseのSTEP4Aでは、mod=0の節点かつstate=未終了の節点が送信を行い、mod=2かつstate=未終了の節点が受信(R)を行う。すなわち、mod=0の節点の親は、mod=2の節点であるので、

mod=0の節点かつstate=未終了の節点が送信を行えば、その節点の親はシグナル μ を聞き、state=未終了である。mod=0の節点が全て未送信ならば、その節点の親はシグナル λ を聞き、state=終了となる。NoopPhaseのSTEP4B,4Cも同様である。

- $C(k)$ が空の場合
SendPhaseのSTEP2でstate=終了となる。すなわち、ソースメッセージを送信するのは、 $d_k + 1$ フェイズ目であるので、 $9 + (2r + 10)d_k + 2r + 5$ ラウンド目で節点kはstate=終了となる。
- $C(k)$ が空でない場合
 $C(k)$ の節点の中でソースからの距離が d_{max}^k の節点がソースからの距離が最大であるのでこの節点は子グループが空である。よって $d_{max}^k + 1$ フェイズ目のSTEP2でソースからの距離が d_{max}^k の節点がstate=終了となる。以降、1フェイズ進むにつれ、NoopPhaseのSTEP4でmod=0の節点がstate=終了になった場合その節点の親と親の親が順にstate=終了となる。mod=1, mod=2の節点がstate=終

了になった場合親が state=終了となる。したがって

- $d_{max}^k = d_k + 1$ の場合
 $d_{max}^k + 1$ フェイズ目の STEP 4 (節
 点 k が $\text{mod}=0$ なら STEP4A、 $\text{mod}=1$ なら
 STEP4B、 $\text{mod}=2$ なら STEP4C) で
 節点 k は state=終了となる。この場合
 節点 k が state=終了となるラウンドは
 $9 + (2r + 10)d_{max}^k + 2r + 7 + \text{mod}$ ラウ
 ンドである。
- $d_{max}^k > d_k + 1$ の場合
 $Y = \{x | d_k + 2 \leq x \leq d_{max}^k \text{ かつ}$
 $x = 3i (i = 0, 1, 2 \dots)\}$
 とすると、ソースからの距離が d_{max}^k
 の節点 k が state=終了になったフェイズ
 から節点 k が state=終了となるのは
 $(d_{max}^k - d_k - |Y| - 1)$ フェイズ後で
 ある。よって、ステージ B は $d_{max}^k +$
 $1 + (d_{max}^k - d_k - |Y| - 1) = 2d_{max}^k -$
 $d_k - |Y|$ フェイズ目の STEP4 (d_k の節
 点 k が $\text{mod}=0$ なら STEP4A、 $\text{mod}=1$ なら
 STEP4B、 $\text{mod}=2$ なら STEP4C) で
 節点 k が state=終了となる。この場合
 節点 k が state=終了となるラウンドは
 $9 + (2r + 10)(2d_{max}^k - d_k - |Y|) + 2r +$
 $7 + \text{mod}$ ラウンドである。

5 ARBEMの正しさの証明

5.1 ソースメッセージ伝達の正当性

ソースメッセージをエンコード法で正確に伝達するための1つ目の条件は、節点がステージ A の動作をしている間に、隣接節点がステージ B での送信を行うのを防ぐことである。もし、ある節点がステージ A での送信を行い、その節点の隣接節点がステージ B での送信を行うとソースメッセージを間違えて受信してしまう。

ソースメッセージをエンコード法で正確に伝達するための2つ目の条件は、各節点が ReceiveMessage を実行中にソースメッセージ送信開始、終了合図以外でシグナル μ を2回連続して聞かないことである。もし、ソースメッセージ送信開始終了合図以外でシグナル μ を2回連続して聞いてしまうと、それをソースメッセージ送信開始終了合図と勘違いして、ソースメッセージが間違えて受信してしまうことになる。

ソースメッセージをエンコード法で正確に伝達するための3つ目の条件は、節点がソースメッセージを送信する1ラウンド前にその隣接節点がシグナル μ を聞かないことである。もしそのとき、その隣接節点がシグナル μ を聞いてしまうとシグナル μ を3ラウンド連続して聞いてしまうのでソースメッセージ送信開始合図が最初の2ラウンドと勘違いしてしまい、ソースメッセージを正しく受信できないからである。以下ではこの3つの条件が成り立つことを証明する。

捕題1 ソース以外の任意の節点 k は必ずステージ A 終了後のステージ B での STEP1 の ReceivePhase でソースメッセージを受信する。すなわち、ステージ A の途中で、節点 k の隣接節点がソースメッセージを送信することはない。

証明 節点 k のソースからの距離を $d_k > 0$ とする。節点 k がステージ A を終了するのは $3d_k + 6$ ラウンド目である。節点 k のステージ B の開始ラウンドは $3d_k + 7$ ラウンド目からになる。節点 k がソースメッセージ送信開始合図でシグナル μ を聞くのは、 $9 + (d_k - 1)(2r + 10) + 1$ ラウンド目である。したがって、 $9 + (d_k - 1)(2r + 10) + 1 - (3d_k + 7) = (2r + 7)(d_k - 1) \geq 0$ であるので節点 k がステージ A の動作を行っている間にその隣接節点がステージ B でのソースメッセージを送信してくることはない。

捕題2 ソース以外の節点は ReceivePhase の ReceiveMessage を実行している間、ソースメッセージ送信開始と送信終了合図以外で2ラウンド連続してシグナル μ を聞くことは無い。

証明 ステージ B では、ソース以外の節点は ReceivePhase を行い、ソースメッセージを受信するまでは ReceiveMessage を実行することになる。基本的にステージ B はエンコード法による RB アルゴリズムを利用している。ReceiveMessage が終了したら SendMessage を行うのがエンコード法による RB アルゴリズムであるが、ARB アルゴリズムでは ReceivePhase の前半部で ReceiveMessage を実行し、後半部では、送信を行うラウンドも存在する。もし、後半部で2ラウンド連続して送信を行ってしまうと次にソースメッセージを受信する節点は、それをソースメッセージ送信開始合図として受信してしまう。しかし、ReceivePhase の後半部では、2ラウンド連続して送信は行わないので RecieveMessage を実行している節点はソースメッセージ送信開始終了合図以外で2回連続してシグナル μ を聞くことは無い。

捕題3 節点がソースメッセージを送信する1ラウンド前にその隣接節点はシグナル μ を聞かない

証明 ソース以外の節点はソースメッセージを受信する前は ReceivePhase を実行し、ソースメッセージを受信した後 STEP2 から STEP5 を行い次に SendPhase の SendMessage を実行(ソースメッセージ送信)する。すなわちソースメッセージを送信する1ラウンド前とは ReceivePhase での最後のラウンドのことである。ReceivePhase の最後のラウンドは STEP5 の未送信である。また、ソースメッセージを受信していない節点は ReceivePhase の ReceiveMessage をソースメッセージを受信するまで実行するので送信することはない。したがって、節点がソースメッセージを送信する1ラウンド前にその隣接節点はシグナル μ を聞くことは無い。

よって、捕題1, 2, 3より次の定理が成立する。
 定理1 ARBEMはRBを実現している。

5.2 ブロードキャストの完了確認の正しさ

節点 i が終了状態とは、節点 i のソースメッセージ受信済みで子グループ $C(i)$ が空であるか、もしくは子グループ $C(i)$ の全ての節点が終了状態であることを認識している状態である。

捕題 4 任意の節点 k において節点 k の $state = 終了$ ならば、節点 k は終了状態である

証明 節点 k のソースからの距離を d_k とする。 $C(k)$ の節点でソースからの距離が最大の値を $d_{max}^k (\geq d_k)$ とする。

アルゴリズムより、 $state = 終了$ となるのは ReceivePhase 実行終了後の SendPhase の STEP2 もしくは NoopPhase の STEP4 (STEP4A, 4B, 4C) であるので、 $state = 終了$ ならば、ソースメッセージは受信済みである。

SendPhase の STEP2 とは、子グループが空であるかどうかの判定である。よって d_{max}^k の節点は子グループは空であるので STEP2 でシグナル λ を聞き、 $state = 終了$ となる。

NoopPhase の STEP4 では、親に自身の $state$ が終了か非終了であるかを知らせる STEP である。 $C(k)$ の節点でソースからの距離が d_{max}^k の節点が $state = 終了$ であれば $C(k)$ の節点でソースからの距離が $d_{max}^k - 1$ の節点は STEP4 でシグナル λ を聞くので $state = 終了$ となる。以後、親が $state = 終了$ となるのを繰り返すことになる。したがって、節点 k は子グループの全ての節点が $state = 終了$ となった後に $state = 終了$ となる。節点 k が $state = 終了$ となるのは、子グループの全ての節点が $state = 終了$ であると判定したと考えられる。すなわち、子グループ全ての節点がソースメッセージを受信していることを節点 k が知ることである。これは、定義より節点 k は終了状態である。

捕題 5 任意の節点 k において節点 k は終了状態ならば、節点 k の $state = 終了$ である

証明 証明する命題の対偶である節点 k の $state \neq 終了$ ならば、節点 k は終了状態でないを示す。

節点 k が $state \neq 終了$ ということは、ソースメッセージ未受信であるか、もしくは子グループの全ての節点が $state = 終了$ であると判別していない状況である。すなわち、子グループの全ての節点がソースメッセージを受信しているかどうかは節点 k にはまだわからない。したがって、これは定義より終了状態ではない。対偶が証明されたので、任意の節点 k において節点 k は終了状態ならば、 $state = 終了$ も成立する。

捕題 4, 5 により次の定理が成立する。

定理 2 節点 k は終了状態になるの必要十分条件は節点 k の $state = 終了$ である。

アルゴリズムから任意の節点は $state = 終了$ となるのでソース s も $state = 終了$ となる。ソース s の子グループ $C(s)$ とは、定義より $C(s) = V - \{s\}$ であるのでソース s が $state = 終了$ すなわち、終了状態になるということは $C(s)$ の全節点が終了状態であるとソースが判定したことになる。したがって、 $C(s)$ の全節点が終了状態であるということは定義より $C(s)$ の全節点はソースメッセージ受信済みである。ゆえに、ソース s が $state = 終了$ になった時点で ARB が達成されたことになる。

定理 3 ARBEM は ARB を実現している。

5.3 ARB アルゴリズムの時間評価

ARB が達成されるのはソースが $state = 終了$ となったラウンドである。したがって、4.4 節における $d_k = 0, d_{max}^k = ecc$ の場合である。

$d_{max}^k = ecc > 1$ の場合ソースが $state = 終了$ となるのは $9 + (2r + 10)(2d_{max}^k - d_k - |Y|) + 2r + 7 = (2r + 10)(2ecc - |y|) + 2r + 17$ ラウンド目である。

したがって、ARB が達成されるラウンド数は $O(r \cdot ecc)$ である。

6 あとがき

文献 [1] では、全節点がアルゴリズム開始時刻を知っている仮定で $O(n)$ ラウンドで ARB が達成されるアルゴリズムが示されているが、本稿では、ソース以外の節点がアルゴリズム開始時刻を知らないという拡張したモデルで、 $O(r \cdot ecc)$ ラウンドで ARB を達成するアルゴリズムを示した。エンコード法を用いているのでアルゴリズムの実行時間にソースメッセージ長が依存するが、ソースメッセージ長が短いとき文献 [1] のアルゴリズムより有効である。

参考文献

- [1] B.S. Chlebus, L. Gąsieniec, A. Gibbons, A. Pelc, W. Rytter, Deterministic broadcasting in unknown radio networks, In Proc. 11th Ann. ACM-SIAM SODA'2000(861–870)
- [2] B.S. Chlebus, L. Gąsieniec, A. Östlin, J.M. Robson, Deterministic radio broadcasting, ICALP2000 LNCS 1853(717–728)
- [3] M. Chrobak, L. Gąsieniec, W. Rytter, Fast Broadcasting and Gossiping in Radio Networks, Proc. 41st Symp. on FOCS'2000(575–581)
- [4] G.D. Marco, A. Pelc, Faster broadcasting in unknown radio networks, IPL 79,2001(53–56)