Vol. 35 No. 5 情報処理学会論文誌 May 1994

有限射影平面を利用した効率のよい分散合意プロトコル

中島 周†

本論文では，2ラウンドで計算を実行する分散合意プロトコルの通信構造について検討する．ネットワーク内の各ノードが持つ値を使用して計算を分散的に行う種々の分散アルゴリズムを総称して分散合意プロトコルと呼ぶ．分散合意プロトコルには分散コミットプロトコルや分散チェックポインティングなどが含まれる．分散合意プロトコルの処理は，各ノードの計算と通信の2つに分けられる．そのうち，通信の部分は異なる分散合意プロトコルで共通である．よって効率のよい通信方法を使うことにより分散合意プロトコルに必要なメッセージ数を減少させることができる．通信処理での各ノードの各ラウンドの宛先の集合を通信構造と呼ぶ．Lakshman と Agrawala は有限射影平面を利用した通信構造を用いて $4n\lfloor\sqrt{n}\rfloor$ 個のメッセージで動作する2ラウンドの分散コミットプロトコルを作成した．本論文では，有限射影平面の点と線の位置関係の対称性を利用した効率の良い通信構造の構成方法を示す．この通信構造を用いると Lakshman と Agrawala の方法に比べて半分のメッセージ数，$2n\lfloor\sqrt{n}\rfloor$ 個，で動作する分散合意プロトコルが作成できる．この通信構造を使用して分散合意プロトコルの例として2ラウンドの分散コミットプロトコルを構成し，その正当性を証明する．また，ノンブロッキングの分散コミットプロトコルへの応用についても述べる．

Efficient Decentralized Consensus Protocols Using a Finite Projective Plane

AMANE NAKAJIMA†

This paper discusses two-round communication structures of decentralized consensus protocols and creates an efficient communication structure using symmetric relation of points and lines in a finite projective plane. Decentralized consensus protocols are distributed algorithms that compute a function whose arguments are distributed in nodes of a nework. Processing of decentralized consensus protocols is devided into two parts : computation in each node and communication. The communication part is common in all decentralized consensus protocols. Thus, a good communication structure reduces the number of messages required for decentralized consensus protocols. The paper proposes an efficient two-round communication structure based on a finite projective plane. With the proposed communication structure, decentralized consensus protocols that require $2n\lfloor\sqrt{n}\rfloor$ messages are obtained. The paper constructs a two-round decentralized commit protol as an example of decentralized consensus protocols using the proposed communication structure, and proves the correctness of the protocol.

1. はじめに

近年，高性能のワークステーションを高速のネットワークで接続した計算環境が一般的となってきた．このように物理的に離れた複数の計算機と通信ネットワークから構成される分散システムでは，従来は1台の計算機による集中処理で実現されてきた機能を分散的に実現しなければならない．しかし，データの分散や非同期に動作する複数の計算主体のために，計算機間での通信が必要となる．分散システムでは，通信に要する時間は，計算機内の処理時間に比べるとかなり大きく，そのために処理に要するメッセージ数を減らすことが重要である．本論文では，2ラウンドの分散合意プロトコルの通信方法について論じ，従来の方法に比較して半分のメッセージ数で動作する通信方法を記述する．

分散合意プロトコルとは，ネットワーク内の各ノードが持つ値をデータとし，各ノードが通信を行いながら同じ処理を実行してある関数を計算したり，ある述語が真であるかを判定したりする分散アルゴリズムの総称である．分散合意プロトコルの例としては，最大値の計算，合計の計算，分散コミットプロトコル，分散チェックポインティングなどがある[1)~11)]．分散合意プロトコルは，複数のラウンドから構成され，最終ラウ

† 日本アイ・ビー・エム(株)東京基礎研究所メディア・システムズ・インスティチュート
Media Systems Institute, Tokyo Research Laboratory, IBM Japan, Ltd.

ンドの実行後，つまりプロトコルの終了時には，すべてのノードが同じ計算結果を有する．1つのラウンドは，各ノードでのメッセージの送受信と受信したメッセージ中のデータを使用した計算から構成される．各ノードは1ラウンド中に複数のノードにメッセージを送信し，また，複数のノードからメッセージを受信する．各ノードが各ラウンドでメッセージを送る宛先ノードの集合を通信構造と呼ぶ．種々の分散合意プロトコルにおいて異なっているのはノードでの計算であり，通信構造はすべての分散合意プロトコルで共通である．よって，効率のよい通信構造を構築することは，すべての分散合意プロトコルの効率を向上させることを意味する．

分散合意プロトコルの効率を判定する主要な尺度はメッセージ数とラウンド数である．メッセージ数は通信の回数を表すために最も重要である．分散合意プロトコルでは，各ノードは各ラウンドで自分が受信する予定のメッセージをすべて受信するまで待ち，その後計算を行い，次のラウンドへと進む．このように，各ラウンドで受信待ちを行うため，ラウンド数が少ないほどプロトコルの実行時間が短くなる．このような理由から，メッセージ数とラウンド数の積を評価基準にすることもある[8]．

分散合意プロトコルは，分散コミットプロトコルとして，Lakshman と Agrawala によって最初に提案された[1),2)]．Lakshman と Agrawala は，有限射影平面を利用した通信構造を採用し，ノード数が n のとき，2ラウンドで $4n\lfloor\sqrt{n}\rfloor$ 個のメッセージを要するプロトコルを作成した．その後，ラウンド数を任意の整数 k に拡張し，メッセージ数を減少させるために種々の通信構造の研究が行われた[4)~11)]．一般に，分散合意プロトコルの通信構造は有限射影平面，generalized hypercube, de Bruijn network などの数学的な構造に基づいている．2ラウンドの通信構造に関しては，これまでの研究で，generalized hypercube を利用すると $2n(\sqrt{n}-1)$ 個のメッセージで，de Bruijn network を利用する場合は $2n\sqrt{n}$ 個のメッセージで分散合意プロトコルが実行できることが示されている[6),7),9),10)]．しかし，有限射影平面を利用したときに，これらに近いメッセージ数で分散合意プロトコルを実行できる通信構造は発見されていない．

また，通信構造を構成するもととなる数学的構造にはそれぞれ制限がある．2ラウンドの場合，generalized hypercube と de Bruijn network ではノード数 n は整数の2乗でなければならない．また，有限射影平面の場合は，その位数が素数の累乗でなければならない．ノード数 n がこれらの制限を満たさない場合は，論理的なノード数がこれらの制限を満たす数になるようにし，論理的なノードを物理的なノードに対応させる．場合によっては，1つの物理的なノードが複数の論理的なノードとして振る舞うことになる．このような処理を行うと，ノードの対称性がくずれてしまうので，論理的なノード数と物理的なノード数が一致することが望ましい．有限射影平面と，generalized hypercube と de Bruijn network では，ノード数 n に対する制約条件が異なるので，ほぼ同じメッセージ数の通信構造でもそれぞれ異なるノード数に対して対称的な通信方法を提供できる．よって異なる数学的構造をもととするほぼ同じメッセージ数の通信構造を構成することは，分散合意プロトコルを現実のシステムに適用するときに有用となる．

本論文では，有限射影平面を利用する場合も，$2n\times\lfloor\sqrt{n}\rfloor$ 個のメッセージで分散合意プロトコルが実行できる通信構造が構成できることを示す．2章では分散合意プロトコルの例として2ラウンドの分散コミットプロトコルの概要を述べ，有限射影平面とそれを利用した Lakshman と Agrawala の2ラウンドの通信構造を説明する．3章では，有限射影平面を用い，Lakshman と Agrawala のものに比べてメッセージ数が半分になる通信構造の構成法を示し，それが分散コミットプロトコルを正しく実行できることを証明する．また，ノンブロッキングの分散コミットプロトコルへの応用についても記す．最後に4章で本論文のまとめを行う．

2. 2ラウンドの通信構造

2.1 2ラウンドの分散コミットプロトコル

分散合意プロトコルでは，以下のことを仮定する．

- ネットワーク内にはグローバルクロックなどの集中化されたコントロールは存在せず，処理はすべて分散的に実行され，すべてのノードは対等である．
- ネットワーク内には共有メモリは存在せず，ノード間のすべての通信はメッセージの送受信によって行われる．
- 通信は信頼性を有する．ノードからのメッセージは内容を変更されず，紛失や重複なしで有限時間内に宛先に到達する．

● ノードや通信リンクは故障しない．通信はすべて1対1で行われ，マルチキャストやブロードキャストは使用しない．

このような仮定のもとで効率良い通信構造を考えることが問題となる．通信時間の上限を仮定しない場合には，ノードに故障があれば分散的に合意に達することが不可能であることが証明されている[12),13)]．また，通信時間の上限を仮定し，故障や悪意のあるノードを許して分散的に合意する問題はビザンチン将軍問題となる[14)]．

本節では分散合意プロトコルの例として2ラウンドの分散コミットプロトコルを説明する．複数のデータベースにデータを複製する場合には，これらの複製間のデータの一貫性を保たなければならない．このため，トランザクションの原子性を保証することが必要になる．集中制御を用いる方法では，2相コミットプロトコルによってこの問題を解決できる[15)]．分散コミットプロトコルでは，トランザクションの原子性の保証を分散的に実現することを目標とする．これを実直に1ラウンドで実現すると，各ノードにデータが複製されている場合，各ノードが他のすべてのノードと通信することが必要になり，全体で $n(n-1)$ 個のメッセージが必要となる．

2ラウンドの分散コミットプロトコルの有限状態オートマトンを**図1**に示す[1)]．各ラウンドで各ノードは複数のノードに対してメッセージを送信する．ノードが第 i ラウンドでメッセージを送る宛先のノードの集合を第 i 送信集合と呼ぶ．各状態の処理を以下に記す．

s：各ノードは，トランザクションを受信したら，そのトランザクションをコミットするかアボートするかを自分自身で決める．アボートするなら，“*no*” メッセージを自分のノードの第1送信集合に送り，トランザクションをアボートし，状態 a に移る．コミットするなら，“yes^1” メッセージを自分のノードの第1送信集合に送り，状態 w_1 に移る．

w_1：自分のノードが第1ラウンドで受信することになっている “yes^1” メッセージをすべて受信したら，“yes^2” メッセージを自分のノードの第2送信集合に送り，状態 w_2 に移る．もし，1つでも “*no*” メッセージを受信したら，“*no*” メッセージを自分のノードの第2送信集合に送り，トランザクションをアボートし，状態 a に移る．

w_2：自分のノードが第2ラウンドで受信することになっている “yes^2” メッセージをすべて受信したら，トランザクションをコミットし状態 c に移る．もし，1つでも “*no*” メッセージを受信したら，トランザクションをアボートし，状態 a に移る．

c：コミット状態

a：アボート状態

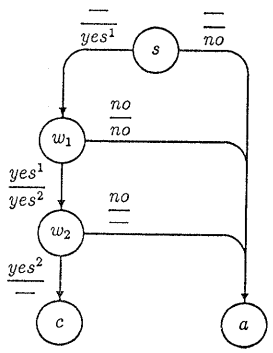

図1 2ラウンドの分散コミットプロトコルの有限状態オートマトン（1）
Fig. 1 Finite state automaton of a two-round decentralized consensus protocol (1).

このプロトコル実行後は，すべてのノードが同じ状態，すなわちコミット状態かアボート状態，に達する．この有限状態オートマトンを実行したときに，トランザクションの原子性を保持できる効率よい通信構造，すなわち各ノードの第1，第2送信集合，を有限射影平面を利用して作成することが本論文の目的である．

2.2 有限射影平面

Lakshman と Agrawala の通信構造を記述し，さらに効率の良い通信構造を議論する前に，有限射影平面について簡単に説明する．

定義1 射影平面とは，以下の公理[16)]を満たす点の集合と線（線とは点の集合である）の集合である．

公理1 異なる2点を通る線はただ1本存在する．

公理2 異なる2線はただ1点で交わる．

公理3 4点で，そのどの3点も1線上にないようなものが存在する．

公理1と公理2は互いに双対的であり，これらの3つの公理から公理3の双対命題

公理4 4線で，そのどの3線も1点で交わらないものが存在する．

が導かれる．よって射影平面は点と線に関して自己双

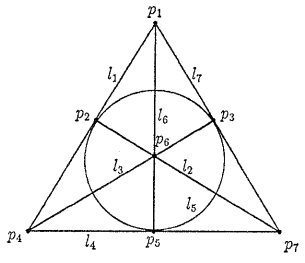

図 2 位数2の有限射影平面（1）
Fig. 2 Finite projective plane of order two (1).

対的である．公理3は1線と2点のような縮退した場合を除外するために存在する．

定義 2 射影平面の中で，点の数が有限なものを有限射影平面と呼ぶ．

有限射影平面に関して以下の定理が存在する[17),18)]．

定理 1 m は $m\geq 2$ なる整数とする．有限射影平面 π において次の各性質はすべて互いに同値である．

1. 1線はちょうど $(m+1)$ 個の点を含む．
2. 1点はちょうど $(m+1)$ 個の線上にある．
3. すべての線はちょうど $(m+1)$ 個の点を含む．
4. すべての点はちょうど $(m+1)$ 個の線上にある．
5. π にはちょうど (m^2+m+1) 個の点がある．
6. π にはちょうど (m^2+m+1) 個の線がある．

m を有限射影平面 π の位数という．

定理 2 q が素数，α が正整数のとき，位数 $m=q^\alpha$ の有限射影平面が存在する．

位数2，つまり7点と7線から構成される有限射影平面の例を**図 2** に示す．

2.3 Lakshman と Agrawala の通信構造

Lakshman と Agrawala は有限射影平面を利用して2ラウンドの通信構造を構成した．このようにある特定の組合せ論的性質を持つ有限集合族をアルゴリズムに利用することは他にも行われている．有限射影平面は分散合意プロトコルでの利用以前には，分散排他制御に用いられていた[19)]．また，属性からレコードを検索するためのファイル構成法では有限幾何学が応用された[20),21)]．組合せ論でデザインと呼ばれる集合族を分散アルゴリズムに利用した例もある[22)]．

有限射影平面 π の点 i を p_i，線 i を l_i と表すことにする $(1\leq i\leq m^2+m+1)$．Lakshman と Agrawala は有限射影平面の点と線を

$$l_i \ni p_i \qquad (1\leq i\leq m^2+m+1) \qquad (1)$$

となるように配置できることを証明し，この性質を持った有限射影平面をもとにして通信構造を作成した[1)]．また，同じ通信構造を使用した分散チェックポインティングのアルゴリズムも発表されている[3)]．図2に示した有限射影平面も式(1)を満たしている．

ノード i の第 j 送信集合を $S_j(i)$ と書くことにすると，Lakshman と Agrawala の通信構造は，

$$S_1(i)=S_2(i)=\{a : l_i\ni p_a, i\neq a\} \cup \{b : l_b\ni p_i, b\neq i\}$$
$$(1\leq a, b, i\leq m^2+m+1), \qquad (2)$$

となる．ノード i を p_i と l_i に対応させると，ノード i は各ラウンドで l_i 上の点に対応するノードと，p_i を通る線に対応するノードのうち自分以外のノードにメッセージを送ることになる．このとき，

$$|S_1(i)|=|S_2(i)|=2m,$$
$$n=m^2+m+1,$$

なので，プロトコルに必要となる総メッセージ数は，

$$\sum_{i=1}^{n}(|S_1(i)|+|S_2(i)|)=4mn=4n\lfloor\sqrt{n}\rfloor, \qquad (3)$$

となる．

図2に示した有限射影平面を使用したときの Lakshman と Agrawala の通信構造を**表 1** に示す．

表 1 Lakshman と Agrawala の通信構造の例
Table 1 Example of Lakshman and Agrawala's communication structure.

i	$S_1(i)$	$S_2(i)$
1	2, 4, 6, 7	2, 4, 6, 7
2	1, 5, 6, 7	1, 5, 6, 7
3	4, 5, 6, 7	4, 5, 6, 7
4	1, 3, 5, 7	1, 3, 5, 7
5	2, 3, 4, 6	2, 3, 4, 6
6	1, 2, 3, 5	1, 2, 3, 5
7	1, 2, 3, 4	1, 2, 3, 4

3. 効率のよい通信構造

本章では，Lakshman と Agrawala の通信構造に比べて，メッセージ数を半減する通信構造を構成する．まず通信構造とそれに伴う分散コミットプロトコルの一部の変更を記述し，その後でそれらの正当性を証明する．

3.1 通信構造の構成法

本論文の通信構造は，式(1)を満たす有限射影平面を使用し，以下のように定義される．

$$S_1(i)=\{a : l_i\ni p_a\} \quad (1\leq a, i\leq m^2+m+1), \ (4)$$
$$S_2(i)=\{a : l_a\ni p_i\} \quad (1\leq a, i\leq m^2+m+1). \ (5)$$

ノード i を p_i と l_i に対応させると，ノード i は第1ラウンドで l_i 上の点に対応するノードに，第2ラ

表 2 本論文の通信構造の例
Table 2 Example of our communication structure.

i	$S_1(i)$	$S_2(i)$
1	1, 2, 4	1, 6, 7
2	2, 6, 7	1, 2, 5
3	3, 4, 6	3, 5, 7
4	4, 5, 7	1, 3, 4
5	2, 3, 5	4, 5, 6
6	1, 5, 6	2, 3, 6
7	1, 3, 7	2, 4, 7

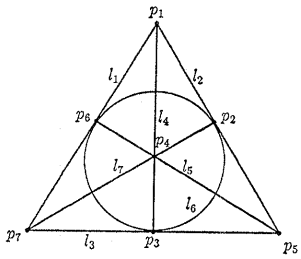

図 3 位数 2 の有限射影平面 (2)
Fig. 3 Finite projective plane of order two (2).

ウンドで p_i を通る線に対応するノードにメッセージを送ることになる．このとき，

$$|S_1(i)|=|S_2(i)|=m+1,$$

であるが，自分自身に対するメッセージは処理が自ノード内で行われるため，メッセージ数の計算時には数えないことになっている．式(1)が成立するので，プロトコルに必要となる総メッセージ数は，

$$\sum_{i=1}^{n}(|S_1(i)|-1+|S_2(i)|-1)=2mn=2n\lfloor\sqrt{n}\rfloor, \quad (6)$$

となる．

図 2 に示した有限射影平面を使用したときの，本論文の通信構造を**表 2** に示す．

有限射影平面は公理 1，2，3 からわかるように点と線は双対である．よって，有限射影平面の点と線を入れ換えたものも有限射影平面になる．つまり，ある点がある線上にある，ある線はある点を通過する，といった位置関係を逆にして，p_i を l_i に，l_i を p_i にしたものも有限射影平面となる．図 2 に示した有限射影平面をこのようにして変換した有限射影平面を**図 3** に示す．図 2 では l_1 上に p_1, p_2, p_4 が存在していたのが，図 3 では p_1 を l_1, l_2, l_4 が通過するように変換されている．よって，

$$S_1(i)=\{a : l_a \ni p_i\} \quad (1\le a, i\le m^2+m+1), \ (7)$$
$$S_2(i)=\{a : l_i \ni p_a\} \quad (1\le a, i\le m^2+m+1), \ (8)$$

のように式(4)，(5)の $S_1(i)$ と $S_2(i)$ を入れ換えたものも本論文で記述したのが通信構造の 1 つとなる．表 2 の $S_1(i)$ と $S_2(i)$ を入れ換えて作った通信構造は，図 3 の有限射影平面をもとにして式(4)，(5)を使って作成したものと同じになる．

3.2 分散コミットプロトコルへの適用

本論文の通信構造を分散コミットプロトコルに使用する場合は，図 1 で示した処理を**図 4** のように変更する必要がある．この変更は，あるノードがアボートしたときに "*no*" メッセージが他のすべてのノードに到達するために必要となる．同様の変更は，複数ラウンドの分散コミットプロトコルにおいても文献 9) で行われている．図 1 の処理との変更，追加の部分のみを以下に記す．

s：各ノードは，トランザクションを受信したら，そのトランザクションをコミットするかアボートするかを自分自身で決める．アボートするなら，"*no*" メッセージを自分のノードの第 1 送信集合に送り，状態 t_1 に移る．コミットするなる，"yes^1" メッセージを自分のノードの第 1 送信集合に送り，状態 w_1 に移る．

t_1："*no*" メッセージを自分のノードの第 2 送信集合に送り，トランザクションをアボートし，状態 a に移る．

次に，本章で記述した通信構造と分散コミットプロトコルの正当性を証明する．

補題 1 各ラウンドで，ノード i は $(m+1)$ 個のメッセージを，第 1 ラウンドでは p_i を通過するすべての線に対応するノードから，第 2 ラウンドでは l_i 上のすべての点に対応するノードから受信する．

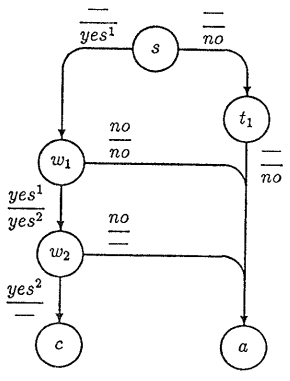

図 4 2 ラウンドの分散コミットプロトコルの有限状態オートマトン (2)
Fig. 4 Finite state automaton of a two-round decentralized consensus protocol (2).

証明 ノード i は第1ラウンドで l_i 上の点に対応するノードに，第2ラウンドで p_i を通る線に対応するノードにメッセージを送ることになる．これをすべてのノードが実行する．よってノード i は第1ラウンドでは p_i を通過するすべての線に対応するノードからメッセージを，第2ラウンドでは l_i 上のすべての点に対応するノードからメッセージを受信する．公理1，2により1つの点を通る線は $(m+1)$ 個，1つの線の上には $(m+1)$ 個の点が存在する．よって各ノードが各ラウンドで受信するメッセージの個数は $m+1$ である． □

補題2 あるノードがコミット状態にあれば，すべてのノードは "yes^1" メッセージを送信している．

証明 コミット状態にあるノードをノード i とする．このとき，"yes^1" メッセージを送信していないノードが存在すると仮定し，このノードをノード j とする．ノード i は補題1により，l_i 上の $(m+1)$ 個の点に対応するノードから，"yes^2" メッセージを受信している．これらの点を $p_{i_1}, p_{i_2}, \cdots, p_{i_{m+1}}$ とする．このとき，ノード $i_1, i_2, \cdots, i_{m+1}$ は "yes^2" メッセージを送信しているので，状態 w_1 にいるか状態 w_1 を通過したことになる．状態 w_1 に遷移するためには第1ラウンドで受信予定のすべての "yes^1" メッセージを受信していることが必要となる．よってこれらのノードは補題1により，自分に対応する点を通るすべての線に対応するノードから "yes^1" メッセージを受信していることになる．

ノード j は "yes^1" メッセージを送信していないので，l_j は $p_{i_1}, p_{i_2}, \cdots, p_{i_{m+1}}$ のどれをも通過しないことになる．すると，$p_{i_1}, p_{i_2}, \cdots, p_{i_{m+1}}$ は l_i 上のすべての点であるので，l_j は l_i と交わらないことになる．これは公理2に矛盾する．よって仮定は誤りである． □

定理3 あるノードがアボートを決意すれば，すべてのノードが最終的にはアボートする．

証明 補題2により，あるノードがアボートを決意したならば，どのノードもコミット状態にはいない．よってすべてのノードが最終的に "no" メッセージを受信することを証明すればよい．

アボートを決意したノードをノード i とする．するとノード i は式(4)により，第1ラウンドで自分に対応する線 l_i を通るすべての点に対応するノードに "no" メッセージを送信する．これらの点を $p_{i_1}, p_{i_2}, \cdots, p_{i_{m+1}}$ とする．第1ラウンドで "no" メッセージを受信したノード $i_1, i_2, \cdots, i_{m+1}$ は式(5)により，第2ラウンドで自分に対応する点を通る線に対応するノードに "no" メッセージを送信する．このとき，式(1)により $p_{i_1}, p_{i_2}, \cdots, p_{i_{m+1}}$ のうちの1点は p_i であるが，状態 t_1 が存在するので，この点に対応するノード i も第2ラウンドで送信を行う．よって第2ラウンドでは，$p_{i_1}, p_{i_2}, \cdots, p_{i_{m+1}}$ を通過するすべての線に対応するノードに "no" メッセージが送られる．

このとき，"no" メッセージを受信しなかったノードがあると仮定し，そのノードを j としよう．すると l_j は $p_{i_1}, p_{i_2}, \cdots, p_{i_{m+1}}$ のどれをも通過しないことになる．$p_{i_1}, p_{i_2}, \cdots, p_{i_{m+1}}$ は l_i 上のすべての点であるので，l_j は l_i と交わらないことになる．これは公理2に矛盾する．よって，"no" メッセージを受信しなかったノードがあるという仮定は誤りである．つまり，すべてのノードが "no" メッセージを受信する． □

定理4 どのノードもアボートを決意しなければ，すべてのノードは最終的にはコミットする．

証明 どのノードもアボートを決意しなければ，すべてのノードは "yes^1" メッセージを各ノードの第1送信集合に送信し，状態 w_1 に遷移する．すべてのメッセージは有限時間内に宛先に到達するので，すべてのノードはそれぞれ第1ラウンドで受信する予定の "yes^1" メッセージをすべて受信し，"yes^2" メッセージを各ノードの第2送信集合に送信して，状態 w_2 に遷移する．第1ラウンドと同様に各ノードは受信予定の "yes^2" メッセージをすべて受信し，コミット状態 c に達する． □

3.3 ノンブロッキング分散コミットプロトコルへの応用

本論文で構成した分散コミットプロトコルは，ノードが故障した場合には，他のノードでのプロトコルの実行が停止してしまうブロッキングプロトコルである．しかし，2章での仮定に追加をし，メッセージの通信時間の上限が知られており，タイムアウトによってノードの障害が検出できるとすれば，プロトコルをノンブロッキングにすることができる．ノンブロッキングプロトコルでは，ノードに故障が発生しても，他のノードは実行を継続でき，最終的にコミットまたはアボートの同じ状態に到達できる．また，故障したノードも回復後リカバリプロトコルを実行することにより同じ状態に達することが可能となる．この場合，3相コミットプロトコルとも呼ばれる Skeen のノンブロッキングプロトコル[23]を分散化させ，メッセージ数を減

少させたプロトコルを本論文の通信構造を使って構成することになる．ただしこの場合は2相コミットプロトコルにさらにバッファ状態を設けるので，ラウンド数，メッセージ数ともブロッキングの場合の2倍になる．この手法は Skeen によって開発され，Lakshman と Agrawala によって2ラウンドの分散コミットプロトコルへも応用された[1]．本論文の通信構造を用いても，文献1)と同様の方法によってノンブロッキングの分散コミットプロトコルが構成できる．

4. おわりに

本論文では，有限射影平面を利用して，メッセージ数 $2n\lfloor\sqrt{n}\rfloor$ の分散合意プロトコルの通信構造を構成した．その通信構造を分散コミットプロトコルに適用した場合のプロトコルを記述し，その正当性を証明した．

有限射影平面を利用した通信構造では，メッセージ数 $4n\lfloor\sqrt{n}\rfloor$ のものが Lakshman と Agrawala によって示されていたが，本論文では同じく有限射影平面を利用しながらメッセージ数を半分に減らせることを示した．ラウンド数を一般の整数 k に拡張した複数ラウンド用の通信構造をラウンド数2に対して適用すると，generalized hypercube を利用する場合には $2n\times(\sqrt{n}-1)$ 個のメッセージを要する通信構造が，de Bruijn network を利用する場合には $2n\sqrt{n}$ 個のメッセージを要する通信構造が構成できることが知られていたが，本論文により有限射影平面をもとにした場合もほとんど同じメッセージ数で分散合意プロトコルが実現できることが明らかとなった．

完全分散ですべてのノードの処理が等しい場合の各ノードの各ラウンドでの送信メッセージ数，受信メッセージ数の下限は，ラウンド数が2のときには $\sqrt{n}$ になる．このとき，分散合意プロトコルに必要となるメッセージ数は $2n\sqrt{n}$ となる．これを実現する通信構造で，

$$S_j(i)\ni i \qquad (1\leq i\leq n, 1\leq j\leq 2)$$

を満たすもののメッセージ数が $2n(\sqrt{n}-1)$ となる．

それぞれの通信構造のもととなる数学的構造には1章で述べたような構成できるための条件があり，そのため対称性と効率を保ったまま分散合意プロトコルを実行できるようなノード数 n には制限がある．よってほぼ同じメッセージ数の通信構造でも，それぞれ異なる構成条件を持つ通信構造は，現実への適用を考えた場合に有用となる．

参考文献

1) Lakshman, T. V. and Agrawala, A. K.: $O(n\sqrt{n})$ Decentralized Commit Protocols, *Proc. 5th Symp. Reliability Distrib. Softw. Database Syst.*, pp. 104-110 (1986).
2) Lakshman, T. V. and Agrawala, A. K.: Efficient Decentralized Consensus Protocols, *IEEE Trans. Softw. Eng.*, Vol. SE-12, No. 5, pp. 600-607 (1986).
3) Son, S. H.: An Algorithm for Efficient Decentralized Checkpointing, *Comput. Syst. Sci. Eng.*, Vol. 4, No. 1, pp. 27-34 (1989).
4) Lakshman, T. V. and Agrawala, A. K.: Communication Structure of Decentralized Commit Protocols, *Proc. 6th Int. Conf. Distrib. Comput. Syst.*, pp. 100-107 (1986).
5) Farrag, A. A. and Dawson, R. J.: On Designing Efficient Consensus Protocols, *Proc. IFIP WG 10.3 Working Conf. Distrib. Processing*, pp. 413-427 (1987).
6) Bermond, J-C., König, J-C. and Raynal, M.: General and Efficient Decentralized Consensus Protocols, *Proc. 2nd Int. Workshop Distrib. Algo.*, pp. 41-56 (1987).
7) Bermond, J-C. and König, J-C.: General and Efficient Decentralized Consensus Protocols II, *Proc. Int. Workshop Paral. Distrib. Algo.*, pp. 199-210 (1988).
8) Ghafoor, A. and Berra, P. B.: An Efficient Communication Structure for Distributed Commit Protocols, *IEEE J. Select. Areas Commun.*, Vol. 7, No. 3, pp. 375-389 (1989).
9) Yuan, S. and Agrawala, A. K.: A Class of Optimal Decentralized Commit Protocols, *Proc. 8th Int. Conf. Distrib. Comput. Syst.*, pp. 234-241 (1988).
10) Hsieh, C. S.: Decentralized Evaluation of Associative and Communicative Functions, *Proc. 9th Int. Conf. Distrib. Comput. Syst.*, pp. 9-11 (1989).
11) Yuan, S.: The Communication Complexity for Decentralized Evaluation of Functions, *Info. Process. Lett.*, Vol. 35, No. 4, pp. 177-182 (1990).
12) Fischer, M. J., Lynch, N. A. and Paterson, M. S.: Impossibility of Distributed Consensus with One Faulty Process, *J. ACM*, Vol. 32, No. 2, pp. 374-382 (1985).
13) Taubenfeld, G.: One the Nonexistence of Resilient Consensus Protocols, *Inf. Process. Lett.*, Vol. 37, No. 5, pp. 285-289 (1991).
14) Lamport, L., Shostak, R. and Pease, M.: Byzantine Generals Problem, *ACM Trans. Prog. Lang. Syst.*, Vol. 4, No. 3, pp. 382-401 (1982).

15) Gray, J. N.: Notes on Data Base Operating Systems, *Operating Systems: An Advanced Course*, Bayer, R., Graham, R. M. and Seegmüller, G. (eds.), pp. 393-481, Springer-Verlag, New York (1978).
16) 長尾 汎:群とデザイン, 岩波書店, 東京(1974).
17) Hall, M. Jr.: *Combinatorial Theory*, Biaisdell Publishing Company (1967) (岩堀信子(訳):組合せ理論, 吉岡書店, 京都 (1971)).
18) Anderson, S. S.: *Graph Theory and Finite Combinatorics*, Markham Publishing Company, Chicago (1970).
19) Maekawa, M.: A $\sqrt{n}$ Algorithm for Mutual Exclusion in Decentralized Systems, *ACM Trans. Comput. Syst.*, Vol. 3, No. 2, pp. 145-159 (1985).
20) Abraham, C. T., Ghosh, S. P. and Ray-Chaudhuri, D. K.: File Organization Schemes Based on Finite Geometries, *Inf. Control*, Vol. 12, No. 2, pp. 143-163 (1968).
21) Ghosh, S. P. and Abraham, C. T.: Application of Finite Geometry in File Organization for Records with Multiple-Valued Attributes, *IBM J. Res. Dev.*, Vol. 12, No. 2, pp. 180-187 (1968).
22) Nakajima, A.: Fault-Tolerant Distributed Match-Making with Any Resiliency, *IEICE Trans.*, Vol. E 74, No. 2, pp. 427-434 (1991).
23) Skeen, D.: Nonblocking Commit Protocols, *Proc. ACM SIGMOD*, pp. 133-142 (1981).

(平成4年3月 6 日受付)
(平成6年1月13日採録)

中島 周 (正会員)

1961年生. 1983年東京大学工学部電子工学科卒業. 1985年同大学院修士課程修了. 同年日本アイ・ビー・エム(株)入社. 現在, 東京基礎研究所主任研究部員. 分散 OS, 分散アルゴリズム, グループウェア, マルチメディア等の研究に従事. 1987年電子情報通信学会論文賞受賞. IEEE Communications Magazine Feature Editor. 電子情報通信学会, IEEE, ACM 各会員.