数理解析研究所講究録
第 790 巻 1992 年 201-207

A General Description of an Information Disseminating Scheme and Its Automorphism

情報散布方式の一般的記述と自己同型写像

Walter Unger*, Yoshihide Igarashi (五十嵐 善英)**,
and Shingo Osawa (大澤 新吾)**

* *Mathematik-Informatik,*
Universität GH-Paderborn,
Paderborn, Germany

** *Department of Computer*
Science,
Gunma University,
Kiryu, 376 Japan

Abstract

We describe an information disseminating scheme on a processor network in a general form. An automorphism with respect to the information disseminating process on the network is introduced. Conditions for the existence of such an automorphism and the effect of the start round to the fault tolerance of the scheme are studied.

あらまし

プロセッサネットワーク上の情報散布方式を一般的に記述する。ネットワーク上の情報散布過程に対応した自己同型写像を紹介する。自己同型写像が存在する条件とスキームの耐故障性へのスタートラウンドの影響について議論する。

1 Introduction

Data broadcasting is a very fundamental operation in parallel and distributed systems. It can be accomplished by the data disseminating process in the network in a way that each processor repeatedly receives and forwards messages without physical broadcast. A scheme introduced by Alon et al[1] specifies such an information disseminating process by a simple procedure. Han and Finkel [3] generalized the scheme given by Alon et al, and discussed its fault tolerance. A number of variations of binary jumping scheme have been shown and discussed by Kanai et al[5][6]. However, relations among the fault tolerance of these schemes are far from being well understood.

In this paper we describe an information disseminating scheme in a general form. Our scheme includes binary jumping scheme and its variations as special cases of the scheme.

In general, the fault tolerance of an information disseminating procedure depends on not only the configuration of faulty processors but also the start round of broadcasting. However, we do not know precisely at present how the fault tolerance is affected by the change of the start round. To understand the effect of the start round we introduce an automorphism on an information disseminating scheme. If such an automorphism on a scheme exists then the fault tolerance of the scheme is independent of the start round. We study what conditions are required so that such an automorphism on an information disseminating scheme exists.

2 An Information Disseminating Scheme

Throughout this paper we consider a network with N processors. These processors are synchronized with a global clock and are linked according to the disseminating scheme. We assume that all links are faultless, but some processors in the network may be faulty. We consider only the case where a faulty processor cannot forward messages, but can receive messages. We do not consider cases where a faulty processor alters information and forwards wrong messages. The time interval for forwarding a message from a processor to one of its neighbors is called a round. Each processor can receive a message and can forward a message in the same round. We also assume that the source processor of broadcasting is always faultless.

The processors in the network are addressed by integers from 0 to $N-1$. We denote k *modulo* m by $[k]_m$. We only consider broadcasting schemes that can be described by the following procedure.

```
procedure broadcast($N$, $R$)
{$N$ is the number of processors in the network, and $R$ is a sequence of positive integers}
let $R = (a_0, \cdots, a_t)$
repeat
    for $round$ := 0 to $t$ do
        each processor $u$ sends a message to processor $[u + a_{round}]_N$ concurrently
forever
```

The length of R is denoted by $|R|$, and the period of the above procedure is $|R|+1$. If R is an infinite sequence, then $|R|$ is denoted by ∞. When $R = (1, 2, \cdots, 2^{\lceil \log_2 N \rceil - 1})$, the above procedure is exactly the same as binary jumping scheme. The initial message may occur at any time in any processor in the network. For any broadcasting scheme that can be specified by the above procedure, the realized network is symmetric with respect to the processors. When we discuss the fault tolerance of the network, without loss of generality we may therefore assume processor 0 to be the source of broadcasting. The broadcast starting at round i ($0 \leq i \leq |R|$) by the above procedure is denoted by $S_i(N, R)$.

Example 1. The direction of each processor at each round by procedure *broadcast*(9, (1, 2, 4, 8)) is shown in Table 1. A disseminating process of $S_2(9, (1, 2, 4, 8))$ is depicted in Figure. 1.

	processor								
round	0	1	2	3	4	5	6	7	8
0	1	2	3	4	5	6	7	8	0
1	2	3	4	5	6	7	8	0	1
2	4	5	6	7	8	0	1	2	3
3	8	0	1	2	3	4	5	6	7

Table 1: Dissemination at each round by *broadcast*(9, (1, 2, 4, 8)).

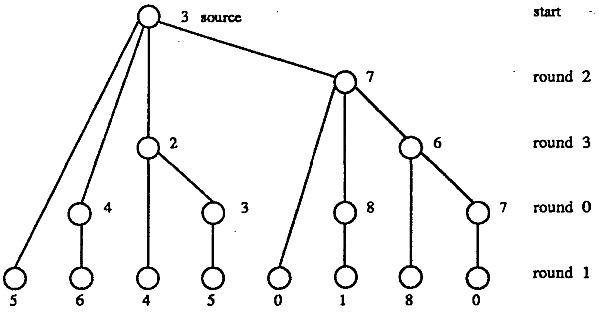

Figure 1: A disseminating process of $S_2(9, (1, 2, 4, 8))$.

3 An Automorphism

In order to understand the effect of the start round of broadcasting we introduce an automorphism on the network with respect to the information disseminating process. Throughout this section we assume that processor 0 is the source of broadcasting in the network, but we do not lose any generality by this assumption.

The next proposition is immediate.

Proposition 1 *For any $N \geq 1$, any finite sequence R of positive integers and any start round i, $S_i(N, R)$ can complete broadcasting within a certain number of rounds if R contains 1 and no processors have failed.*

Let $R_h^{(t)}$ be a sequence $(a_0, a_1, \cdots, a_t)$, where a_i $(0 \leq i \leq t)$ is h^i and $2 \leq h$, and let $R_h^{(\infty)}$ be the infinite sequence $(h^0, h^1, h^2, h^3, \cdots)$. In this section we only consider procedure broadcast(N, $R_h^{(t)}$), where t is a nonnegative integer or ∞. Note that $S_i(N, R_h^{(\infty)})$ may not complete broadcasting forever. For example, if $N = h^i$, $S_i(N, R_h^{(\infty)})$ cannot send the message to any processor from the source processor forever.

Definition 1 Two broadcasts $S_i(N, R_h^{(t)})$ and $S_j(N, R_h^{(t)})$ are equivalent if and only if there exists a one-to-one function f on $\{0, \cdots, N-1\}$ satisfying the following conditions:

1. $f(0) = 0$

2. For any $k \geq 0$ and $0 \leq p, q \leq N-1$, processor p sends the message to processor q by $S_i(N, R_h^{(t)})$ at round $[i+k]_{t+1}$ if and only if processor $f(p)$ sends the message to processor $f(q)$ by $S_j(N, R_h^{(t)})$ at round $[j+k]_{t+1}$.

A function f satisfying the conditions in Definition 1 is called an automorphism from $S_i(N, R_h^{(t)})$ to $S_j(N, R_h^{(t)})$.

Definition 2 If. for any pair of i and j ($0 \leq i, j \leq t$) $S_i(N, R_h^{(t)})$ and $S_j(N, R_h^{(t)})$ are equivalent, then procedure *broadcast*(N, $R_h^{(t)}$) is said to be automorphic.

Lemma 1 *If there exists an automorphism from $S_i(N, R_h^{(t)})$ to $S_j(N, R_h^{(t)})$, then it is unique.*

Proof. It is sufficient to show that the assertion holds true for the automorphism f from $S_0(N, R_h^{(t)})$ to $S_i(N, R_h^{(t)})$. Suppose that there exists an automorphism f from $S_0(N, R_h^{(t)})$ to $S_i(N, R_h^{(t)})$. At the start round the source processor sends a message to processor 1 by $S_0(N, R_h^{(t)})$ and to processor $[h^{[i]_{t+1}}]_N$ by $S_i(N, R_h^{(t)})$. At the $x(t+1)$-th round after the start round ($x = 1, 2, \cdots$) the source processor sends a message to processor $x+1$ by

$S_0(N, R_h^{(t)})$ whereas the source processor sends a message to processor $[(x+1)h^{[i]_{t+1}}]_N$ by $S_i(N, R_h^{(t)})$. Therefore, from the definition of an automorphism, $f(x) = [h^{[i]_{t+1}} x]_N$ for all $0 \leq x \leq N-1$ and the automorphism is uniquely determined. □

Theorem 1 *For any $0 \leq t \leq \lceil \log_{h+1} N \rceil - 1$ and $N \neq (h+1)^n - 1$, procedure broadcast(N, $R_h^{(t)}$) is not automorphic, where $n = \lceil \log_{h+1} N \rceil$.*

Proof. Suppose that there exists an automorphism f from $S_0(N, R_h^{(t)})$ to $S_1(N, R_h^{(t)})$, where $t \leq \lceil \log_{h+1} N \rceil - 1$. Processor 0 sends the message to processor h^t at round t by $S_0(N, R_h^{(t)})$ and to processor 1 at round $[t+1]_{t+1}$ (= round 0) by $S_1(N, R_h^{(t)})$. Hence, $f(h^t)$ should be 1. However, from the proof of Lemma 1, $f(x) = [hx]_N$ for any x ($0 \leq x \leq N-1$), but $[h^{t+1}]_N$ cannot be 1. This is a contradiction. Therefore, there is no automorphism from $S_0(N, R_h^{(t)})$ to $S_1(N, R_h^{(t)})$ if $0 \leq t \leq \lceil \log_{h+1} N \rceil - 1$. □

Lemma 2 *Let N be relatively prime to h and i be a nonnegative integer. Let f be a function on $\{0, \cdots, N-1\}$ defined as $f(x) = [h^i x]_N$ for all $0 \leq x \leq N-1$. Then f is a one-to-one function.*

Proof. Let $0 \leq x < y \leq N-1$. $h^i y - h^i x = h^i(y-x)$. Since N is relatively prime to h and $1 \leq y - x \leq N-1$, $h^i(y-x)$ cannot be a multiple of N. Hence, $[h^i x]_N \neq [h^i y]_N$. Hence, f is a one-to-one function. □

Theorem 2 *For any pair of nonnegative integers i and j, $S_i(N, R_h^{(\infty)})$ and $S_j(N, R_h^{(\infty)})$ are equivalent if and only if N is relatively prime to h.*

Proof. Let $j > i$. Suppose that N is not relatively prime to h and that there exists an automorphism f from $S_i(N, R_h^{(\infty)})$ to $S_j(N, R_h^{(\infty)})$. From the proof of Lemma 1, $f(x) = [h^{j-i}x]_N$ for all $0 \leq x \leq N-1$. Hence, $f(N/h) = [h^{j-i-1}N]_N = 0$, and then $f(0) = f(N/h)$. Therefore, f is not one-to-one and cannot be an automorphism from $S_i(N, R_h^{(\infty)})$ to $S_j(N, R_h^{(\infty)})$.

Let N be relatively prime to h and f be a function defined as $f(x) = [h^{j-i}x]_N$. From Lemma 2, f is a one-to-one function on $\{0, \cdots, N-1\}$. Let x be an arbitrary processor and r be a number of rounds after the start round. By $S_i(N, R_h^{(\infty)})$ processor x sends its message to processor $[x+h^{r+i}]_N$ at round $r+i$. We should prove that $f(x)$ will send the message to processor $f([x+h^{r+i}]_N)$ by $S_j(N, R_h^{(\infty)})$ at round $j+r$.

$$\begin{aligned} f(x) &= [h^{j-i}x]_N, \\ f([x+h^{r+i}]_N) &= [h^{j-i}[x+h^{r+i}]_N]_N \\ &= [h^{j-i}x+h^{r+j}]_N. \end{aligned}$$

Hence, $S_i(N, R_h^{(\infty)})$ and $S_j(N, R_h^{(\infty)})$ are equivalent. □

Theorem 3 *If N is relatively prime to h, then there exists an integer k such that $k \leq N-2$ and that for any i and j ($0 \leq i,j \leq k$) $S_i(N, R_h^{(k)})$ and $S_j(N, R_h^{(k)})$ are equivalent.*

Proof. There exists a pair of a and b ($0 \leq a < b \leq N-1$) such that $[h^a]_N = [h^b]_N$. For such a pair of a and b, $[h^a(h^{b-a}-1)]_N = 0$. Since N is relatively prime to h, $[h^{b-a}-1]_N = 0$. Let $k = b-a-1$. Then $S_i(N, R_h^{(\infty)})$ and $S_i(N, R_h^{(k)})$ are the same broadcast, and $S_j(N, R_h^{(\infty)})$ and $S_j(N, R_h^{(k)})$ are the same broadcast. Hence, from Theorem 2, $S_i(N, R_h^{(k)})$ and $S_j(N, R_h^{(k)})$ are equivalent. □

The next corollary is immediate from Theorem 3.

Corollary 1 *If N is relatively prime to h, then for any $i \geq 0$, $S_i(N, R_h^{(\infty)})$ can complete broadcasting within a certain number of rounds.*

Corollary 2 *If N is a prime, then for any i and j ($0 \leq i,j \leq N-1$) $S_i(N, R_h^{(N-2)})$ and $S_j(N, R_h^{(N-2)})$ are equivalent.*

Proof. It is immediate from Theorem 3 and the fact that $[h^{N-1}]_N = 1$ for any prime. □

Corollary 3 *If N is not relatively prime to h, then for any $t \geq 1$ procedure broadcast(N, $R_h^{(t)}$) is not automorphic.*

Proof. From Lemma 1 and its proof, any automorphism f from $S_0(N, R_h^{(t)})$ to $S_i(N, R_h^{(t)})$ should be $f(x) = [h^i x]_N$ for all $0 \leq x \leq N-1$. If i is not 0, this function is not one-to-one since there exists y ($0 < y \leq N-1$) such that $[h^i y]_N = 0$ (e.g., if $y = N/h$ then $[h^i y]_N = 0$). Hence, in this case f cannot be an automorphism. □

If procedure *broadcast*(N, $R_h^{(t)}$) is automorphic, then for any pair of i and j ($0 \leq i,j \leq t$) $S_i(N, R_h^{(t)})$ and $S_j(N, R_h^{(t)})$ have the same fault tolerance against faulty processors. This can be stated in the next theorem.

Theorem 4 *Suppose that the network with N processors contains some faulty processors and that procedure broadcast(N, $R_h^{(t)}$) is automorphic. Then for any pair of i and j ($0 \leq i, j \leq t$), the broadcast by $S_i(N, R_h^{(t)})$ completes within r rounds for any configuration of f faulty processors if and only if the broadcast by $S_i(N, R_h^{(t)})$ completes with in r rounds for any configuration of f faulty processors.*

4 Concluding Remarks

We have shown that if the period of procedure *broadcast*($N$, $R_h^{(t)}$) is less than $\lceil \log_{h+1} N \rceil$ and $N \neq (h+1)^n - 1$ then the scheme is not automorphic. We are interested in the relation between the fault tolerance of *broadcast*($N$, $R$) and the choice of $R$. This problem would be worthy for further investigation.

References

[1] N. Alon, A. Barak, and U. Manber, "On disseminating information reliably without broadcasting", Proceidings of the 7th International Conference on Distributed Computing Systems, pp.74-81 (1987).

[2] J-C. Bermond and C. Peyrat, "Broadcasting in de Bruijn networks", Congressus Numerantium, 66, pp.283-292 (1988).

[3] Y. Han and R. Finkel, "An optimal scheme for disseminating information", Proceedings of 1988 International Conference on Parallel Processing, Chicago, Illinois, pp.198-203 (1988).

[4] M. Imase, T. Soneoka and K. Okada, "Connectivity of regular directed graphs with small diameters", IEEE Trans. on Computer, 34, pp.267-273 (1985).

[5] K. Kanai, K. Miura and Y. Igarashi, "Information disseminating schemes for fault tolerance in computer networks", IEICE Technical Report, Computation, COMP 90-35, pp.45-52 (1990).

[6] K. Kanai, Y. Igarashi and K. Miura, "Fault tolerance of Han-Finkel's scheme for disseminating information", Technical Report, Algorithms 91-AL-24-2, Information Processing Society of Japan, Vol. 91 No. 102 (1991).