論文 89 — 26 — 11 — 22

다단계 상호 연결망에서의 효율적인 분산 라우팅 알고리듬의 설계 및 분석

(Design and Analysis of an Efficient Distributed Routing Algorithm in Multistage Interconnection Networks)

孫 有 翼*, 安 光 善**

(Yoo Ek Son and Gwang Seon Ahn)

要 約

본 논문은 회선 교환식 다단계 상호연결 네트워크에서 효율적인 분산 제어 방식의 라우팅 알고리듬에 관하여 제안하고 평가한다. 제안된 방법은 각 스위칭따라 제어되는 분산 제어 방식을 사용하고 있는데 이는 하나의 시작지 주소로부터 임의의 여러 목적지 주소로 연결하고자 하는 방송형 연결에 효과적인 방법임을 보이고 있다. 제안된 라우팅 알고리듬의 성능은 컴퓨터 시뮬레이션을 통하여 분석평가되었다.

Abstract

This paper presents and evaluates a distributed routing algorithm for effective routing control in circuit-switched multistage interconnection networks. The proposed method uses the distributed control based on the individual-switching element control and it is very effective for allowing any broadcast connection from a source to arbitrary number of destinations. The performance of the proposed method is analyzed and evaluated by computer simulation in terms of the normalized average time delays.

I. 서 론

최근, 대규모 집적회로 기술수준의 발전과 컴퓨터 시스템의 고성능화 요구로 인하여 병렬/다중 처리 컴퓨터 시스템에 관한 연구가 활발히 진행되고 있다.[1,2] 이러한 종류의 컴퓨터 시스템내에서, 프로세서 및 메모리 모듈사들 사이를 상호연결시키는 상호 연결망은 전체 시스템의 성능을 결정하는 주요 요인들중의 하나로써 시스템의 골격을 이루는 부분이며, 특히 다단계 상호연결망은 알려져있는 바와 같이 가격/성능면에서 우수하므로 현재까지 많은 종류의 네트워크들이 연구 제안되어 왔다.[3-7] 그러나 대체로 이들 다단계 상호 연결망들이 그 구조적인 면에서 동등하다는 것은 주지하는 바와 같으며[3] 단지 이들 네트워크들의 차이점은 라우팅 제어방식, 방송형 전송 방식의 유무 그리고 입출력 이름의 배열상의 차이가 있을 뿐이다. 특히 이들중에서 라우팅 제어 문제는 시

*正會員, 啓明大學校 電子計算學科
(Dept. of Computer Science, Keimyung Univ.)
**正會員, 慶北大學校 컴퓨터工學科
(Dept. of Computer Eng., Kyungpook Nat'l Univ.)
接受日字 : 1989年 7月 18日

작지 주소와 목적지 주소들 사이의 효율적인 경로를 제공하기 위한 방법을 연구하는 한 분야로써 아주 중요하다. 그러나 이 분야에서의 현재까지의 연구는 주로 one-to-one 연결에 대해서는 많은 진행을 보여왔으나[8,9] one-to-many 즉, 방송형 연결에 대해서는 Siegel이 제안한[10] 단계별 제어 방식을 이용한 방법을 비롯하여 일부 진행되어 왔으나 이러한 단계별 제어 방식에서는 주어진 제한 조건 내에서 즉, 전송하고자 하는 목적지 주소들간의 Hamming distance 를 만족시켜야하는 제한 조건으로 인하여, 임의의 다수 목적지로 전송시키고자할 경우에는 이러한 조건을 만족시키는 여러개의 그룹으로 나누어 순차적으로 전송시킬 수 밖에 없기 때문에 이 방법에서는 여러 그룹으로 분리하는데 따른 오버헤드가 많이 생길수 있는 문제점이 제기될 수 있다.

본 논문에서는 회선 교환식 다단계 상호연결 네트워크에서 하나의 시작지 주소에서 여러개의 목적지 주소로 자료를 전송시킬 경우, 이러한 제한 조건에 구애받지 않는 새로운 라우팅 방법을 제안하고자 하였으며 이를 위한 라우팅 알고리듬을 스위칭 소자별 제어방식을 이용하여 구현하였다. 특히 제시된 방법에서는 네트워크 내에서의 라우팅을 목적지의 갯수와 무관하게 일반화시켰기 때문에 one-to-one방식을 따로 분리시켜 고려할 필요가 없이 one-to-many방식의 일부로 간주될 수 있게 하였다. 또한 제안된 시뮬레이션 알고리듬은 네트워크의 크기를 증가시켜가면서, 랜덤으로 발생시킨 임의의 목적지 주소를 대상으로, 제안된 라우팅 방식에 적용시켜 통계적으로 분석한 결과 시간 복잡도면에서 상당한 개선이 있음을 보였다. 2절에서는 베이스라인 네트워크를 이용한 상호 연결망의 시스템 구성과 Siegel이 제안한 단계별 제어 방식의 동작 상태를 설명하였으며, 3절에는 네트워크의 구조적 특성이 용이한 새로운 방식의 분산 알고리듬을 제시하였으며 이를 부분별로 나누어 설명하고 있다. 4,5절에서는 제안된 알고리듬을 분석하고 또한 타당성을 조사하기 위한 시뮬레이션을 수행하였으며 이를 토대로 제안된 방식을 검토하였다. 마지막으로 6절에 결론을 기술하였다.

Ⅱ. 관련 연구

1. 다단계 상호연결망 구조

본 논문에서는 라우팅 문제를 고려함에 있어서 다단계 상호연결 네트워크들 중의 한 모델로써 베이스라인 네트워크를 이용하였다. 베이스라인 네트워크는 N개의 시작지 주소(sources)와 N개의 목적지 주소(destinations)들로 구성되며 시작지 주소는 단계 m-1(네트워크의 크기가 8인 경우, 단계 2 가 된다.)에 있는 스위칭 소자에 연결되어있고 목적지 주소는 단계 0에 있는 소자에 연결되어 있다. 여기서 $N=2^m$, m은 네트워크를 구성하는 단계수이며, 각 단계는 $\frac{N}{2}$개의 2×2스위칭 소자들로 구성된다. (그림 1 참조) 스위칭 소자의 동작상태는 4개의 입출력 단자들의 결합에 따라 16종류가 있을수 있으나 여기서는 on-to-one 연결방식의 straight, exchange 연결과 one-to-many 연결방식의 lower broadcast, upper broadcast 연결등 4가지 동작상태를 네트워크를 구성하는 스위칭 소자들의 정상상태로 간주한다. 베이스라인 네트워크가 가지는 구조적 특성으로부터 아래와 같은 정의 및 정리를 추출해 낼 수 있다.

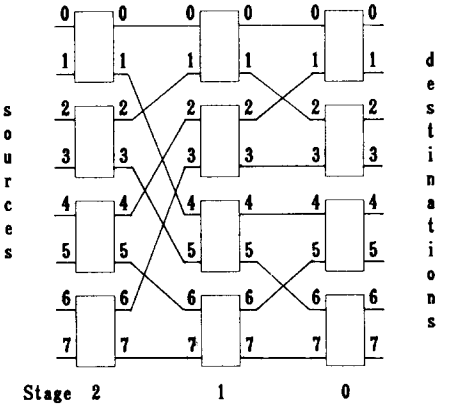

그림 1. N=8인 경우의 베이스라인 네트워크
Fig. 1. A baseline network with N=8.

〔정의 1〕 각 스위칭 소자는 단계 내에서의 위치를 $p_{m-1}\cdots p_2p_1$의 이진표현 코드워드로 나타낼 수 있으며, 각 링크 아래 방법에 따라 $p_{m-1}\cdots p_1p_0$의 코드워드로 표현될 수 있다. 여기서 p_0는 아래 방법에 따라 정해진다.

$p_0=0$, 입력 링크가 스위칭소자의 상위링크로 연결된 경우

$p_0=1$, 입력 링크가 스위칭소자의 하위링크로 연결되는 경우

〔정의 2〕 베이스라인 네트워크의 각 단계들 사이의 상호연결 패턴은 아래의 상호연결 함수에 의해 정의된다.

$$F_i[P_{m-1}\cdots P_i\ P_0]=p_{m-1}\cdots p_{i+1}\ p_0\ p_i\ p_{i-1}\cdots p_1,$$
$$0<i\le m-1.$$

〔정의3〕 단계 i에 있는 스위칭 소자는 아래의 사상 규칙에 의해 다음 단계(단계 i−1)에 있는소자로 연결된다.

$$M_{i,0}[(p_{m-1}\cdots p_2\ p_1)_i]=(_{m-1}\cdots p_{m-i}\ 0\ p_{m-i-1}\cdots p_2)_{i-1},$$
$$\text{for upper link}(p_{m-1}\cdots p_1\ 0)_{i-1},\quad 0<i\le m-1.$$

$$M_{i,1}[(p_{m-1}\cdots p_2p_1)_i]=(p_{m-1}\cdots p_{m-i}\ 1\ p_{m-i-1}\cdots p_2)_{i-1},$$
$$\text{for lower link}\ (p_{m-1}\cdots p_1)_{i-1},\quad 0<i\le m-1.$$

〔정리1〕 베이스라인 네트워크에서, 주어진 시작지 주소로 부터 목적지 주소 사이에는 단 하나의 경로만 존재한다.

증명 : 시작지 주소 $S=s_{m-1}\cdots s_1\ s_0$에서 목적지 주소 $D=d_{m-1}\cdots d_1\ d_0$로 연결한다고 가정한다. S는 한 단계를 통과할때 마다 〔정의2〕의 상호연결 함수를 적용시키므로서 다음 단계에 있는 스위칭 소자의 입력으로 연결될 수 있다. 단계i에 있는 스위칭 소자는 목적지 택 비트 d_i에 따라 소자의 출력으로 연결되며 이 과정을 단계 수 만큼 반복하면 하나의 주어진 목적지 주소로 연결된다. 즉, 지정된 하나의 시작지 주소로 부터 위에서 언급한 사상 규칙 및 상호연결 함수에 의해 지정되는 각 목적지 주소는 유일하다.

2. Siegel 방법에 대한 분석

다단계 상호 연결망에서의 방송형 연결을 위하여 Siegel은[10] 각 단계마다 하나의 제어선을 이용한 단계별 제어방식을 논리주소를 이용하는 다단계 큐브 네트워크를 이용하여 제시하고 있다. 이러한 방법을 이용하여 하나의 시작지 주소로부터 임의의 다수 목적지 주소로 연결시키고자 할 경우 네트워크의 라우팅은 크게 세가지의 과정으로 나누어 질 수 있는데, 먼저 제어 조건을 만족시키는 목적지 주소들로 그룹화시키는 과정과, 방송정보 및 라우팅정보{R, B}추출 과정, 그리고 그룹별 전송 과정으로 고려될 수 있는 목적지 주소들의 조건이란 그룹내의 목적지 주소의 갯수가 k개인 경우 임의의 두 목적지 주소의 Hamming distance가 $\log_2 k$이어야하며, 따라서 네트워크에서 주어진 목적지 주소들이 이러한 조건을 만족시키지 못할 경우 그룹화 과정에서 처럼 위 조건을 만족시키는 여러개의 그룹으로 분리해야 한다는 점이다. 다음은 각 그룹별 정보추출 과정으로써 방송형연결 제어를 위한 방송정보와 one-to-one연결로써 상위 및 하위연결을 결정해주는 라우팅정보를 결정하는 과정이다. 여기서, 라우팅 택은 방송정보 m bits와 라우팅정보 m bits로 구성되어 전체로는 2m bits의 {R, B}로 표시된다.[10,11]

예를들어, 주어진 목적지 주소들의 집합을 {2, 3, 6, 7}이라할 경우, 목적지주소의 갯수 k는 4이므로 Hamming distance는 $\log_2 k$ 즉, 2가 된다. 따라서 위 조건을 만족시키므로 하나의 그룹으로 구성될 수 있으며 여기서 이진표현 형태에서 Hamming distance 를 만족시키는 두개의 목적지 주소{2, 7}를 취한후, 이들 중 하나를 취하여 그 이진표현을 라우팅정보로 사용하고, '010', '111'(2와 7의 이진표현)을 exclusive-or시킨 '101'은 방송정보로 사용한다. 이들 정보는 연결망내의 각 단계에 있는 스위칭소자의 제어에 사용되며 그 결정 과정은 알고리듬 1 과 같다.

```
begin
  if (broadcast information=1)
    then(broadcast connection)
    else begin
           if(routing information=0)
             then(upper link connection)
             else(lower link connection)
         end ;
end ;
```

알고리듬 1. 스위칭소자의 제어신호 발생
Algorithm 1. SE control signal generation.

언급된 바와같이 Siegel이 제시한 단계별 제어방식에서의 문제점은 주어진 Hamming distance를 만족시켜야 한다는 제한조건을 가진 방송형 연결에 관한 알고리듬이기 때문에 기본적으로 임의의 방송형 연결에 적용시키고자할 경우 이들 제한조건에 따른 오버헤드가 상당히 커지게 된다.

Ⅲ. 라우팅 알고리듬

먼저 베이스라인 네트워크의 토폴로지로 부터 알고리듬의 기본 원리로 제공될 몇가지 구조적 특성을 추출해 낼 수 있다. 다단계 큐브 네트워크처럼 논리

주소를 사용하는 네트워크들도 있으나 베이스라인 네트워크에서는 물리주소를 사용한다. 네트워크에 있는 스위칭 소자와 링크들의 물리 주소는 각각 $p_{m-1}\cdots p_2 p_1$ 및 $p_{m-1}\cdots p_2 p_1 p_0$로 표현된다고 할때, 〔정리1〕에 의해 임의의 시작지 주소로 부터 목적지 주소 사이에 존재하는 스위칭 소자 및 링크의 주소들을 모두 계산해 낼 수 있다. 예를들어, N=8의 경우 시작지 주소가 2이고 목적지 주소가 6인 경우, 아래의 네가지 단계를 통해 네트워크의 각 단계별로 스위칭 소자들을 통과하는 입출력 링크들의 집합을 {2 3 5 5 6 6}과 같이 구할 수 있으며 경로 추적과정은 알고리듬 2 과 같다.

[Step 1] 여기서는 목적지주소 택 방법을 사용하기 때문에 목적지 주소 6 (이진표현으로 '110')을 네트워크의 라우팅 택으로 취한다. ($d_i = 110, 0 < i \leq m-1$)

[Step 2] 단계 m−1 (여기서는 N=8을 가정하므로, 단계 2 가 단계 m−1이 된다.)에 연결되는 입력 링크는 시작지 주소가 2 (이진표현으로 '110') 이므로 단계 2 에 있는 소자의 입력 주소는 2가 된다. 그리고 단계 m−1을 제어하는데 사용하는 택비트는 d_{m-1}이므로 이것에 해당하는 택 비트 1에 의해 출력 주소는 [정의1] 에 의해 6이 된다. (여기서, $m=\log_2 N$, N : network size)

[Step 3] 단계 2 에 있는 소자의 출력 링크주소 3을 [정의 2]의 상호연결 함수에 적용시키면 그 다음 단계인 단계 1 에 있는 스위칭 소자의 입력링크의 주소는 5가 된다.

[Step 4] 위의 Step 2, Step 3 과정을 단계 수 만큼 m회 반복한다.

알고리듬 2. 링크주소의 계산과정
Algorithm 2. Procedure of link address computation.

이상의 제어 방식에서 알 수 있듯이 제어대상을 네트워크의 각 단계를 하나의 신호로써 제어하는 것이 아니라 스위칭소자 각각에 대한 독립적 제어가 가능한 분산 제어 방법을 사용하므로써 앞서 Siegel의 제한조건에 따른 분리 과정과 이에 따른 순차적 정보 전송으로 인한 시간 지연을 개선할 수 있다. 이를 위한 라우팅 신호 발생과정은 알고리듬 3에서 보이고 있다.

[Step 1] 먼저 시작지 주소와 목적지 주소를 읽은후 이진 표현으로 바꾼다.

[Step 2] [정의 1, 2, 3], [정리 1]을 이용하여 연결 경로상에 있는 스위칭 소자들의 위치 및 링크 주소들을 구한다.

[Step 3] 시작지 주소로부터 시작하여 네트워크의 첫번째 단계에 있는 스위칭 소자의 제어 신호를 구한다. 이것은 선택된 목적지 주소들 중에서 d_i의 값을 조사한 후 각 비트의 값에 따라 [정리2 참조], 스위칭소자는 상위연결, 하위연결, 상위방송연결, 그리고 하위방송연결의 네가지 신호중 하나를 갖게 된다.

[Step 4] 마지막으로 Step 4 에서는 Step 3 과정을 네트워크를 구성하고 있는 단계 수 만큼의 회수로 반복한다. 다만 마지막 단계(단계 0)에서는 소자의 출력주소들이 목적지 주소가 될 것이므로 다음 단계의 입력 주소를 구할 필요는 없다.

알고리듬 3. 라우팅 신호 발생과정
Algorithm 3. Routing signals generation procedure.

〔정리 2〕 주어진 출력주소들의 집합을 $\{d^k \mid 0 < k =< N\}$이라할때, 이들 각각은 $d^k_{m-1}\, d^k_{m-2}\cdots d^k_1 d^k_0$와 같이 (출력주소의 이진표현) 나타낼 수 있으며 여기서 i 번째 단계에서 ($0 < i \leq m-1$) 사용될 제어신호는 모든 출력주소들중에서 m−1번째 비트로부터 i+1번째까지의 비트들을 i 번째 비트와 비교하므로써 구할 수 있다. 스위칭소자의 라우팅 제어신호는 목적지주소 d_i 값을 조사하여 결정한다. d_i가 0으로만 구성되어있을 경우는 상위연결, 1로만 구성되어 있다면 하위연결이 된다. 그러나 0, 1 모두 포함할 경우에는 다시 입력주소를 조사하여 S_{m-1}가 0이면 상위 방송연결이 되며, S_{m-1}가 1이면 하위방송 연결을 위한 라우팅 제어신호를 발생시킨다.

증명 : 여기서는 목적지주소 택 방식을 채택하고 있기때문에 다음과 같은 조건에 따라 출력 주소를 이용하여 제어 신호를 추출할 수 있다. 즉, 임의의 주소에 대한 코드워드를 $p_{m-1}p_{m-2}\cdots p_1 p_0$라 할때, i 단계에서의 임의의 입력단자 $d_{m-1}\, d_{m-2}\cdots d_{i+2} d_{i+1} p_{m-1}\, p_{m-2}\cdots p_{i+1} p_i$로 부터 출력단자 $d_{m-1} d_{m-2}\cdots d_{i+1}\, d_i\, p_{m-1} p_{m-2}\cdots p_{i+1} p_i$로의 연결 경로를 형성하게 된다. 여기에 상호연결 함수를 적용시키면 i−1단계의 입력주소 $d_{m-1} d_{m-2}\cdots d_{i+1}\, d_i\, p_{m-1} p_{m-2}\cdots p_{i+1}$에 연결되며 여기서 $d_{m-1} d_{m-2}\cdots d_{i+1}\, d_i$ 부분이 동일한 목적지주소가

존재할 수 있다. 이 경우 목적지주소는 d_i의 값을 조사하여 d_i가 0으로 구성되어 있으면 $d_{m-1}d_{m-2}\cdots d_{i+1}$ $d_i p_{m-1}p_{m-2}\cdots p_{i+1}0$의 형태로 되며 d_i가 1로 구성되어 있으면 $d_{m-1}d_{m-2}\cdots d_{i+1}d_i p_{m-1}p_{m-2}\cdots p_{i+1}1$ 로 된다. 따라서 이러한 조건을 만족하는 전체 출력주소들의 각 d_i번째 비트들이 모두 0이면 상위연결 제어신호를, 모두 1이면 하위연결 제어신호를, 0,1을 모두 포함할 경우에는 입력단자의 위치가 스위칭소자의 상위링크에 연결되어 있으면 상위방송연결 제어신호를, 입력단자의 위치가 스위칭소자의 하위링크에 연결되어 있으면 하위방송연결 제어신호를 발생시키게 된다.

본 논문에서는 스위칭 소자별로 제어 신호를 할당해야 하기때문에 제어 신호를 발생시키는 과정이 Siegel의방식에서 보다는 복잡하다. 스위칭 소자별 제어 신호를 발생시키기 위해서는, 연결 경로내에 포함되어있는 스위칭 소자들이 네트워크내의 어느 단계에 속하며 또한 그 단계 내에서도 몇번째 위치에 있는가를 찾아내는 위치 결정과정이 필요하다. 따라서 이를위해 어떤 단계내에 있는 스위칭 소자들이 다음 단계에 있는 어느 소자와 연결되는지를 나타내는 매핑규칙과 단계 사이의 상호연결 형태를 결정해주는 함수가 필요하며 그것은 [정의2], [정의3]에 있다. 시작지주소는 각 단계를 거치면서 목적지주소로 전송되어져 나가는 과정에서 두가지 변형과정을 거치게 된다. 하나는 스위칭소자를 제어하게되는 라우팅 택의 값에 따라 'Straight', 'Exchange', 'Upper broadcast' and 'Lower broadcast'가운데 하나로 세팅되면서 그 단계에 포함되어있는 스위칭 소자의 출력 주소를 발생시키게 되며, 둘째로는 이 출력주소를 받아 단계와 단계 사이를 연결시켜주는 상호연결 함수에 적용시키므로서 다음 단계의 입력 주소를 발생시키게 되는데 이러한 과정을 단계 수 만큼 반복해 나가면서 목적지 주소로 전송되게 된다.

SIG방식과 IEC라우팅 방식에서 살펴본 바와같이 라우팅 택을 결정하는데 필요한 정보량의 비교해보면, 먼저 SIG방식의 경우, 목적지 주소의 갯수 $k=2^j$이며, j는 그중의 임의의 두 목적지 주소 사이의 Hamming distance가 되는 조건하에서 추출된 것으로써 라우팅 택{R, B}중에서 R의 값은 목적지 주소중 임의의 것을 선택하며, B값에 대한 계산은, j만큼의 Hamming distance를 갖는 임의의 두 목적지 주소를 Exclusive-or시키므로서 얻을 수 있다. 따라서 SIG방식에서는 라우팅을 결정하는데 필요한 정보는 한개의 Group마다 m bits의 라우팅 정보와 m bits의 방송정보가 필요하게되어, 총 $(g\times 2m)$ bits의 라우팅 택이 사용된다. 여기서 $m=\log_2 N$(N : network size, m : no. of stages, g : no. of groups)이다. IEC방식에서의 경우, 하나의 연결 요구를 가정할때, 주어진 네트워크의 크기내에 존재하는 가능한 모든 목적지 주소로 전송한다 하더라도 (2^m-1) 갯수의 스위칭소자가 필요하게 되며 각 스위칭소자는 주어진 네가지의 유효상태중 하나로 세팅되기 위해서는 2bits씩 필요하게 되므로 결국 $2(2^m-1)$ bits가 되므로 전체 IEC방식에서 소요되는 시간은 5절에서 설명되는 T_{cont_g} (the control signal generation time in IEC)이 차지하는 부분이 커지게되는 요인이 된다.

Ⅳ. 시뮬레이션 시스템 구성

시뮬레이션을 위해 제안된 라우팅 방법과 네트워크의 상호연결 함수에 의거하여 구현하였다. 알고리듬 4는 시뮬레이션 전체 과정을 나타낸 알고리듬이다. 여기서 사용된 알고리듬 SE_cont_gen은 임의의 단계에 있는 스위치소자에서 입력단자에 대한 연결경로를 구할 수 있는 목적지주소를 선택한 후 특정한 제어신호를 발생시키는 함수로써 알고리듬 5에 나타나 있으며, 알고리듬 6은 소자별 제어방법 에서 임의의 단계에 속한 스위칭소자의 입력단자에 대하여 그 단자와 연결된 스위칭 소자의 위치, 즉 스위칭 소자의 물리주소를 계산하고 입력단자의 최소 비트를 조사하여 입력단자의 위치를 결정하기 위한 것으로써 select_SE 알고리듬에서 보이고 있다. 또한 알고리듬 connect는 SE_cont_gen로 부터 구해진 스위칭 소자 제어신호에 따라, 주어진 베이스라인 네트워크의 상호연결 함수를 적용시켜 다음단계에 있는 스위칭 소자에 연결되는 입력단자의 주소를 구하는 것으로써 알고리듬 7 에 나타내 보였다.

시뮬레이션 프로그램시 사용된 여러가지 형태의 자료구조가 있으나 논문에서 보여진것은 그중 알고리듬 설명에 필요한 일부만 나타나 있다. 여기서 max_m은 주어진 네트워크 크기내에서의 가장 번호가 큰 단계번호를 나타내며, max_size은 네트워크의 크기를 증가시켜가면서 시뮬레이션을 수행해 나갈때 네트워크의 최대크기를 어느 값으로 결정할 것인지를 나타내며, max_mimnusl 은 주어진 네트워크내에서 stage-level을 증가시켜갈수 있는 가장 큰 stage번호를 나타낸다. max_sizemminus1은 주어진 네트워크내에서 link_level을 증가시켜갈 수 있는 가장 큰 link번호를 나타내며, max_sizediv2minus 1은 주어진 네트워크내에서 SE_level을 증가시켜갈 수 있는 가장 큰

```
const
  max_m;            /* max. no. of stages                    */
  max_size;         /* max. network size.                    */
  max_mminus 1;     /* the largest stage number              */
type
  list 1=array[ 0.. max_mminusl ] of integr;      1
  list2=array[0.. max_size, 0.. max_m] off integer;
var
  n                 : integer; /* network size               */
  m                 : integer; /* number of stages (0.. mminus1)  */
  mminus1           : integer; /* m-1                        */
  destin_numd       : integer; /* number of destinations     */
  upper_or_lower: integer; /* upper or lower link            */
  SE_cont_sig       : integer; /* control signal of each element  */
  count             : integer; /* no. of new source of next stage */
  stage_level       : integer; /* stage level of source address   */
  first             : list1;
  second            : list2;

begin
  read source and destination addresses.
  sort the destination addresses.
  change the addresses to binary form.
  if (stage_level 〈 〉 mminusl) then
  begin
    for j:=0 to mminus 1 do
      first[j] := second [sum, j];
    select_SE (first, selected_SE, upper_or_lower);
    /* bcate the switching element */
    /* refer to algorithm 5. */
    upper-or-lower := first [mminus1];
    if (destin_numd = mminus1)
      decide network size;
      else SE_cont_gen (destin_binary, stage_level, SE_cont_sig);
      /* generate the SE control signals */
      /* refer to algorithm 6.          */
    se[selected_SE, stage_level] := SE_cont_sig;
    connect (first, SE_cont_sig, news);
    /* applying the interconnection function to stages */
    /* refer to algorithm 7.                           */
    for i := 0 to count do
      generate the new sources of current stage;
    end;
    increment the stage_level;
end;
```

Algorithm 4. Overall simulation process.
알고리듬 4. 시뮬레이션 전체과정

```
var
  upper_or_lower: integer; /* upper or lower link:          */
  SE_level      : integer; /* SE location in each stage     */

begin
  SE_level := 0;
  for SE_level: = m-1 to 1 do
  begin
    check the decimal values of souce addresses:  ;
  end;
  upper_or_lower := sel_SE [mminus1];
  check the last bit of source addrss;
end;
```

알고리듬 5. select-SE알고리듬.
Algorithm 5. select-SE Algorithm.

```
var
  destin_numd: integer; /* number of destinations          */
  stage_level : integer; /* stage level of source address  */

begin
  case stage-level of
      0 : begin/* when stage_level is equal to 0           */
            for i:0 to destin_numd do
            test bit patterns of destination addresses;
          end;
  1.. m: begin/* when stage_level is not equal to 0        */
            for i: 0 to destin_numd do
            test bit patterns from 0th to stage_level_1 of
            destination addresses;
            begin
              for j := to stage_level_1 do
              select the destinations including the same bit
              patterns;
            end;
          end;
  end;
  assign the control signals to switching elements;
  /* 0 : upper link connection */
  /* 1 : lower link connection */
  /* 2 : upper broadcast       */
  /* 3 : lower broadcast       */
end;
```

알고리듬 6. SE_cont_gen 알고리듬
Algorithm 6. SE_cont_gen Algorithm.

```
const
   max-mminus 1;        /* the largest stage number                    */
   max_sizediv 2 minus 1; /* the largest switching element number      */

type
   list 3 = array [ 0. . max_sizediv 2minus 1, 0. . max_mminus 1] of integer;

var
   destin_addr : integer; /* destination address                       */
   destin_numd : integer; /* number of destinations                    */
   stage. level : integer; /* stage level of source address            */
   se          : list 3  ; /* used in control signal generation        */

begin
   if (stage-level < > mminus 1) then
   begin
      generate the source of next stage by applying the
      interconnection function of the baseline network;
      case x of
        0, 1 : begin
        /* one_to_one connection, 0 : upper link connect ion */
        /*                        1 : lower link connection */
              conut : 0;
              for j : 0 to mminusl do
                  next_stage_S[0, j ] : = se [ j ];
              for j : stage_level to m-2 do
                  next_stage_S[ 0, j + 1] : se [ j ];
              if (x = 0)
                  then next_stage_S[ 0, stage-level ] : = 0;
                  else next_stage_S[ 0, stage-level ] : = 1;
           end;
        2, 3 : begin
        /* broadcast connections, 2 : upper broadcast   */
        /*                        3 : lower broadcast   */
                  cont : = 1;
                  for i : 0 to 1 do
                  begin
                     for j : = 0 to mminusl do  1
                        next_stage_S [ i, j] : = se [ j ];
                     next_stage_S[ i ,stage-level ] : = se [ mminus 1 ];
                     for j : = stage-level to m-2 do
                        next_stage_S[ i, j + 1] : = se[ j ];
                  end;
                     if (next_stage_S[ 1, stage_level] : = 0)
                        then next_stage-S[ 1, stage_level] : = 1;
                        else next_stage- S[ 1, stage_level] : = 0;
               end;
      end;
   end;
end;
```

알고리듬 7. Connect 알고리듬
Algorithm 7. Connect Algoriehm

스위칭 소자의 위치번호를 나타낸다. 그리고 max_m 2minus 1은 주어진 네트워크의 각 단계마다 입출력되는 링크들의 최대수를 나타낸다.

V. 성능분석 및 고찰

본 논문에서 제시한 분산 라우팅 알고리듬의 시간 복잡도를 평가하기 위하여 Siegel 방식과 IEC방식에 대한 각각의 전송 경로 설정과정시의 소요시간을 살펴보면 아래와 같다.

$$T_{SIG} = T_{group} + g \times (T_{cont_s} + T_{net})$$
$$= T_{group} + g \times (T_{cont_s} + T_{setup} + T_{data})$$
$$T_{IEC} = T_{cont_E} + T_{net}$$
$$= T_{cont_E} + (T_{setup} + T_{data})$$

where T_{SIG} : the total transmission time in Siegel's method.
T_{IEC} : the total transmission time in IEC control method.
T_{net} : the network transmission delay time ($T_{setup} + T_{data}$).
T_{cont-S} : the control signal generation time in ISC.
T_{cont-E} : the control signal generation time in IEC.
T_{setup} : the routing path setup time.
T_{data} : the actual data transfer time.
T_{group} : the grouping time.
g : the number of groups.

위에서 보인 소요시간들중에서 T_{net}이 시작 되기전 수행되어야할 SIG방식에서의 T_{group} 및 T_{cont-S} 과정과 IEC제어 방식에서의 T_{cont-E}이 전체시간의 대부분을 차지하는 시간으로써 이에대한 효율적인 알고리듬의 설계가 중요하게 된다. 이를 위해 앞서 제시된 알고리듬의 타당성을 검토하기 위하여 컴퓨터 시뮬레이션을 수행하였다. 시뮬레이션에 사용된 데이타는 컴퓨터에서 발생시킨 랜덤 수를 통해 얻어진 수의 배열로 부터 목적지 주소의 개수를 1개부터 N개까지 순서적으로 추출하여 사용하였으며 표1는 네트워크의 크기 N값을 증가시켜 가면서 Siegel이 제안한 방식과 본 논문에서 제시한 분산제어 방식에 적용한 결과를 보이고 있다. 나타나 있는 결과 데이타는 지연 시간을 정규화한 것이다. 표 1 에서 보면 ISEC 알고리듬을 이용한 방법이 SIG방법에서 보다 전체적으로 개선되어 있음을 알 수 있다. 특히, 목적지 주소의 갯수 d_n이 네트워크의 크기와 같은 경우 즉, N이 ($N = 2^m$, $m = 0, 1, \cdots \log_2 N-1$) 어떤 값이던

주어진 네트워크의 목적지주소 전체로 전송시키고자 할 경우 이들은 하나의 그룹으로 묶여질 수 있기 때문에 시뮬레이션시 이를 미리 조사하므로써 소요시간의 감소를 기할 수 있다. 시뮬레이션 결과, 본 논문에서 제안된 IEC방법을 기존의 SIG방법과 비교해보면 표 2 에서 볼수 있는 바와같이 N의 값을 확장시켜 나갈때 두가지 방법에 대한 평균 지연 시간은 N=8인 네트워크의 경우 약 18.3%, N= 16일때 37.3%, N=32일때 50.4%, 그리고 N=64의 경우 약 58.39%의 속도 개선이 있음을 알 수 있었으며 더우기 N의 값이 커짐에 따라 더욱 큰 효율성을 보이고 있음을 알 수 있다. 그림 2 에서는 SIG 방식과 IEC방식 각각에 대한 분석결과와 시뮬레이션 결과를 보이고 있다. 여기서 시뮬레이션의 결과가 낮게 나타난것은 분석결과에서의 시간복잡도는 upper limit에 둔것에 반해 시뮬레이션에서는 여러회 수행한 결과의 평균값이기 때문인 것으로 볼 수 있다.

Ⅵ. 결 론

본 연구에서는 다단계 상호 연결망에서 중요시되는 라우팅 문제를 다루고 있다. 상호 연결망내의 각 스위칭소자가 개별 제어를 갖는 분산 제어 방식을 이용하여 하나의 시작지 주소에서 임의의 여러개의 목적지 주소로 연결시키기 위한 방송형 연결 방법에 관하여 연구하였다. 기존의 Siegel의 방식에 따르면 한번에 전송이 가능한 목적지 주소들의 집합은 이들의 이진 표현 주소들 상호간의 Hamming distance가 j인 (이때 임의의 목적지주소의 수 d_n은 2^j)조건을 만족해야 하는데, 이때 소요되는 그룹화 과정에서 발생되는 오버헤드 문제와, 또한 그룹들의 순차적 전송에 따른 시간 지연이 문제점으로 제기된다. 이에대한 개선책으로서 본 논문에서는 스위칭 소자별로 제어가 가능한 분산 제어 방식의 알고리듬을 제안하였으며, 시뮬레이션 결과 본 논문에서 제안된 방식에서의 성능은 전송 속도면에서 Siegel이 제시한 단계별 제어방식과 비교하여 상당한 개선이 있음을 보였으며 특히 네트워크의 크기 N이 커짐에 따라 더욱 큰 효율성이 있음을 알 수 있었다.

앞으로의 연구계획은, 여러 Source로 부터의 라우팅 요구가 있을 경우에 발생될 수 있는 Conflict에 대한 해결 방안에 대한것으로 Dual network 및 Network partitioning문제를 포함시킬것을 고려하고 있다. 그러나 본 논문에서 언급하고 있는 SIG방식 및 IEC방식에서 Conflict발생을 가정할 경우 SIG 방식에서의 Conflict에 대한 고려는 기본적으로 이러한 방

표 1. N값의 증가에 따른 평균 시간지연 비교
Table 1. Comparison of average time delays.

N	2		4		8		16		32	
d_n	T_{SIG}	T_{IEC}	T_{SIG}	T_{IEC}	T_{SIG}	T_{IEC}	T_{SIG}	T_{IEC}	T_{SIG}	T_{IEC}
1	3.0	2.0	3.0	5.0	7.0	8.0	9.0	11.0	11.0	14.0
2	3.0	1.0	8.6	7.8	14.0	12.6	18.4	17.8	23.4	25.6
3			14.0	10.0	13.2	17.8	25.2	25.6	40.4	34.4
4			5.0	4.0	14.4	21.2	32.6	30.0	67.6	42.8
5					36.6	24.8	54.0	38.0	76.0	51.2
6					30.6	27.6	62.8	43.0	81.6	55.2
7					41.0	31.0	69.8	46.6	105.8	67.6
8					8.0	10.0	92.4	50.0	138.0	68.6
9							98.2	57.0	134.6	78.2
10							123.0	60.8	112.6	85.6
11							137.2	65.8	196.6	91.8
12							134.6	69.8	187.0	100.6
13							127.2	74.0	236.0	106.0
14							122.6	78.0	207.0	109.0
15							110.0	82.0	252.4	118.4
16							13.0	22.0	369.2	121.8
17									299.6	127.0
18									309.4	132.2
19									314.6	139.0
20									329.2	143.6
21									321.4	149.0
22									369.2	154.4
23									366.6	160.6
24									354.6	165.2
25									371.0	170.4
26									372.6	175.6
27									387.6	181.0
28									359.6	185.8
29									387.6	191.0
30									327.2	196.0
31									309.0	201.0
32									22.0	46.0
Ave.	3.0	1.5	8.2	6.7	23.5	19.2	76.9	48.2	232.7	115.3

where d_n : number of destinations
N : network size
T_{SIG} : time delay by using the Siegel's method
T_{IEC} : time delay by using the distributed method
$A_{ve.}$: average time delay.

표 2. N값의 증가에 따른 평균 시간지연 비교
Table 2. Comparison of the two methods in average time delay by increasing the network size, N.

네트워크 크기 (N)	SIG 방식 적용결과	IEC 방식 적용결과	개 선 율 (%)
2	3.00	1.5	-
4	8.15	6.7	-
8	23.50	19.2	18.30
16	76.87	48.2	37.30
32	232.27	115.3	50.36
64	616.20	256.4	58.39

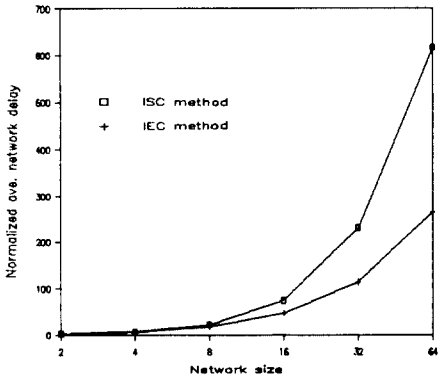

그림 2. 두 가지 방법에 대한 비교 그래프.
Fig. 2. Comparison graph of the two methods.

식이 Stage별로 제어신호를 발생시킨다는 점에서 Single connection이던 Multiple connections이던 순차적 전송이 될 수 밖에 없으므로 Stage내에있는 단하나의 소자에서 Conflict가 발생된다 하더라도 Resolution될때까지는 그 Stage내에 있는 모든 소자들이 Wait해야하므로 효율성에 있어 상당한 의문이 제기된다고 볼 수 있다. 반면에 IEC제어 방식의 경우는 스위칭 소자별로 제어가 가능하므로 네트워크 내에서의 Conflict 발생빈도는 훨씬 감소될 것으로 보인다.

參 考 文 獻

[1] *IEEE Trans. on Computers,* "Special issues on supercomputers, vol. C-31, May 1982.

[2] K. Hwang and F.A. Briggs, *Computer Architecture and Parallel Processing,* McGraw-Hill, 1984.

[3] C.L. Wu and T.Y. Feng, "On a class of multistage interconnection networks," *IEEE Trans. on Computers,* vol. C-29, Aug. 1980.

[4] M.C. Peace III, "The indirect binary n-cube microprocessor array," *IEEE Trans. on Computers,* vol. C-26, May 1977.

[5] H.J. Siegel and R.J. McMillen, "The multistage cube:a versatile interconnection network," *IEEE Computer,* vol. 14, Dec. 1981.

[6] D.H. Lawrie, "Access and alignment of data in an array processor," *IEEE Trans. on Computers,* vol. C-24, Dec. 1975.

[7] K.E. Batcher, "The flip network in STARAN," *Int. Conf. on Parallel Proc.,* Aug. 1976.

[8] N.J. Davis and H.J. Siegel, "Performance analysis of partitioned circuit switched multistage interconnection networks, *The 12th Annual Int. Symp. on Computer Arch. Conf. Proc.,* pp. 387-394. 1985.

[9] N.J. Davis and H.J. Siegel, "Performance studies of multiple packet multistage cube networks and comparison to circuit switching," *Int. Conf. on Parallel Proc.,* pp. 108 114, 1986.

[10] H.J. Siegel, *Interconnection Networks for Large-scale Parallel Processing: Theory and Case Studies,* Lexington Books, pp. 120-125, 1985.

[11] 손유익, 안광선, "다단계 베이스라인 네트워크에서의 분산 라우팅 제어방법에 관한 연구", 대한전자공학회 논문지, 제26권 제 6 호, 1989년 6 월.

著 者 紹 介

孫 有 翼(正會員)

1976년 경북대학교 전자공학과 졸업. 1979년 경북대학교 대학원 전자공학과 석사학위 취득. 현재 경북대학교 대학원 전자공학과 박사과정 수료. 1979년부터 1984년까지 한국전자통신연구소 근무, 선임연구원. 1984년 부터 계명대학교 공과대학 전자계산학과 근무, 현재 조교수. 주관심분야는 컴퓨터구조, 병렬처리구조, Interconnection Network등임.

安 光 善 (正會員) 第26卷 第 6 號 參照

현재 경북대학교 공과대학 컴퓨터공학과 교수