

3. E. Hildreth The detection of intensity changes by Computer and biolocal vision system, CGIP, vol. 2 2, ppl-27, 1983.
4. E. Oran Brigham The Fast Fouries Transform and its Application, Prentice-Hall, 1979.
5. Pratt, Digital Image Processing, John Willey & Sons, 1978.
6. Kenneth R. Castleman, Digital Signal Processing, Prentice-Hall INC., 1979.
7. N. Ahmed K. R. Rao, Orthogonal Transforms for Digital Signal Processing, Springer-Verlag, 1975.
8. Dan E. Dudgeo, Multidimensional Digital Signal Processing, Prentice-hall, 1984.
9. Friedrich M. Wahl, Block segmentation and Text Extraction in Mixed Text / Image Documents, CGIP, vol 20, p375-390, 1982.
10. 옥 철호 외, 문서영상의 분절과 구분에 관한 연구, 경희대학교 석사 학위 논문, 1988. 8. AA30

玉哲昊(Cheol Ho OHK) 準會員
 1964年11月3日生
 1987年2月：慶熙大學校 電子工學科 卒業(工學士)
 1989年8月：慶熙大學校大學院電子工學科 卒業(工學碩士)
 現在：軍服務中。

許道根(Do Geun HUH) 正會員
 1951年10月11日生
 1975年2月：蔚山大學校電氣工學科卒業(工學士)
 1980年2月：慶熙大學校 大學院 電子工學科 卒業(工學碩士)
 1980年2月～現在：圓光大學校 電子工學科 副教授



陳廣玉(Yong Ohk CHIN) 正會員
 1943年3月21日生
 1968年：延世大學校 工科學 電氣工學科 卒業
 1975年：延世大學校大學院 電子工學科 工學碩士
 1981年：延世大學校大學院 電子工學科 工學博士
 1980年：通信技術士
 現在：慶熙大學校 工科學 電子工學科 教授

분할된 다단상호접속망의 성능 분석에 관한 연구

正會員 金 煥 善* 正會員 崔 眞 圭**

A Study on the Performance Analysis of Partitoned Multistage Interconnection Network

Young Sun KIM*, Jin Kyu CHOE *Regular Members*

要 約 상호접속망은 병렬처리 시스템에서 매우 중요한 부분이다. 최근 다단상호 접속망에 대한 연구가 많이 행해지고 있다.

본 논문에서는 회선스위칭 다단상호접속망의 시뮬레이션 방법을 확장하여 분할된 ADM/IADM 접속망의 성능 평가를 하였다. 시뮬레이션 데이터에 의하여, 적용된 분할 방법에 의한 접속망 성능과 접속망에서 사용된 저지처리방식에 의한 접속망 성능 간의 관계를 보였다. 분석 결과, hold 방식을 사용하는 IADM 접속망이 RST면에서 가장 나은 접속망 동작을 보이는 것으로 나타났다.

ABSTRACT The interconnection network is an integral part of parallel processing system. The multistage interconnection networks(MINs) have been the objects of intense research in recent years.

In this paper, simulation techniques for circuit switching MIN are extended to allow the performance evaluation of partitioned ADM/IADM network. Based on simulation data, the relationship between the network performance, the partitioning scheme employed, and the conflict resolution strategies used within the network is enumerated. It is shown that IADM network coupled with the use of the hold strategy produces the best network operation in terms of RST (Request Service Time).

I. 서 론

처리 속도의증대 및 시스템 신뢰도의 개선을

위하여 다수의 프로세서(processor)를 구성되는 병렬처리 시스템(Parallel Processing System)이 미래의 컴퓨터 구조로 주목을 받고 있다.⁽¹⁾ 경제적으로도 최근 VLSI 기술의 발전은 많은 수의 프로세서를 이용하는 병렬처리 시스템의 개발을 가능하게 하고 있다. 그러나 프로세서의

*韓國電子通信研究所
Electronics and Telecommunications Research Institute.
**大田工業大學 電子計算學科
Dept. of Computer Science Taejon National Univ. of
Technology
論文番號 : 89-65(接受1989. 9. 12)

수가 많아질수록 프로세서와 프로세서, 프로세서와 메모리 모듈을 효과적으로 연결 구성하는 것은 복잡하고 어려운 문제이며, 점점 더 중요한 문제가 되고 있다. 따라서 상호 접속망(Interc-connection Network)의 설계는 병렬처리 시스템 개발의 관건이 되고있다.⁽²⁾

상호접속망 중 다단 상호 접속망(Multistage Interconnection Network)⁽³⁾은 스위칭 소자로 구성되며, 연결 능력과 가격 면에서 우수하여 많은 연구가 행해지고 있다. 다단 상호 접속망은 토폴로지(topology)에 따라 크게 큐브(cube)형과 PM2I(Plus-Minus 2)형으로 나눌 수 있다.⁽⁴⁾ 큐브 형은 단경로 접속망으로 일반화된 큐브(Generalized Cube)⁽⁵⁾, Omega, Baseline 접속망등이 있고, PM2I 형은 다중경로 접속망으로 Data Manipulator, ADM(Augmented Data Manipulator)⁽⁶⁾, IADM(Inverse ADM), Gamma 접속망 등이 있다.

일반적으로 병렬처리 시스템은 SIMD(Single Instruction stream, Multiple Data stream)나 MIMD(Multiple Instruction stream, Multiple Data stream) 중 한가지로 동작한다. 그러나 상호 접속망에 따라 분할 가능한 시스템이 연구 개발되고 있다. 분할 가능 시스템(partitionable system)이란 하나의 시스템에서 하나 이상의 SIMD 또는 MIMD 시스템이 여러가지 크기로 구성될 수 있으며, 각기 독립적으로 동작될 수 있는 시스템을 말한다.⁽⁷⁾분할 가능 시스템의 예로는 Ultracomputer(multiple MIMD), TRAC(multiple SIMD 및 MIMD)등이 있다.

분할 가능 시스템의 장점은 다음과 같다.

1. Fault tolerance가 높아진다. (fault된 부분을 실행에서 제외함으로써 fault를 없앤다.)
2. 시스템을 동시에 이용할 수 있는 사용자의 수가 늘어난다.
3. 프로그램 개발이 용이해진다. (대규모 시스템을 위한 프로그램을 실제보다 작은 '가상'시스템에서 개발할 수 있다.)
4. 효율적인 동작을 위하여 시스템의 크기를 task의 요구에 맞출 수 있다.

5. 하나의 job의 독립적인 task들은 병렬로 처리할 수 있다.

이상과 같이 분할 가능 시스템은 많은 장점을 가지고 있다. 그러나 이러한 장점을 최대한 이용하자면 분할된 상호 접속망에 대한 성능 평가가 필요하다. 다단 상호 접속망의 성능 평가에 대하여는 많은 연구가 행해졌지만 대부분 분할되지 않은 큐브형 접속망에 관한 것이고 PM2I형에 대한 연구는 거의 없다.⁽⁸⁾⁽⁹⁾ PM2I 형은 큐브 형과 비교하여 토폴로지와 스위칭 소자가 다소 복잡하지만, permutation의 수나 fault tolerance 면에서 우수하여 최근 주목을 받고 있다. 이 중 ADM과 IADM 접속망은 PM2I 형의 대표적인 접속망으로 PM2I형 접속망 연구에 주로 사용되고 있다.⁽¹⁰⁾ADM과 IADM 접속망은 분할 가능하므로, 분할 가능 시스템의 상호 접속망으로 설계될 수 있다. 또한 ADM과 IADM 접속망은 토폴로지만 역으로 배열되었을 뿐 나머지는 동일한 접속망이어서 분할 방법과 저지처리방식에 따른 비교 분석은 분할 가능 시스템의 성능 예측에 매우 중요한 연구가 된다. 다단 상호접속망의 성능 평가에는 해석적 모델에 의한 수학적 해석 방법과 시뮬레이션에 의한 방법이 있다. 그러나 접속망이 커지고 복잡해짐에 따라 그 복잡성과 해석적 모델의 해를 구하기 어려우므로, 접속망의 해석에는 수학적 해석의 증명보다 시뮬레이션이 자주 사용된다.⁽⁶⁾따라서 본 논문에서는 ADM과 IADM 접속망의 동작을 모델링하여 ADM-IADM 접속망 시뮬레이터(simulator)를 개발하고, 이 시뮬레이터를 이용하여 분할 방법과 저지처리방식을 달리하면서 분할 시 ADM 접속망과 IADM 접속망의 성능을 비교 분석하였다.

II. 상호접속망 모델과 시뮬레이션

ADM 접속망은 1978년에 data manipulator 구현시의 일부 조건을 수정하여 제안된 다중경로 접속망이다.⁽¹¹⁾ 그림 1은 입력이 8, 출력이 8인 ADM 접속망이다. 그림 2는 IADM(Inverse ADM) 접속망이다. IADM 접속망은 ADM 접속

망과 단의 순서만 역으로 배열되었을 뿐 같은 구조이다.

크기가 N인 ADM 접속망에서 하나의 단(stage)은 N개의 스위칭소자(또는 node)와 다음 단의 입력과 연결되는 3×N개의 데이터 경로로 구성된다(그림 1에서는 N=8). 이들 접속망은 PM2I(Plus-Minus 2ⁱ) 상호접속기능을 기초로 한다. 이 기능은 다음과 같이 정의되는 2n 개의 기능으로 이루어진다.

$$PM2_{+i}(j) = j + 2^i \pmod N \quad (1-a)$$

$$PM2_{-i}(j) = j - 2^i \pmod N \quad (1-b)$$

여기서 $0 \leq j < N, 0 \leq i < N$.

ADM과 IADM 접속망은 다중 경로 접속망이므로 입력 노드 번호와 출력 노드 번호가 같지 않다면, 입력과 출력 사이에 하나 이상의 경로가 존재한다.

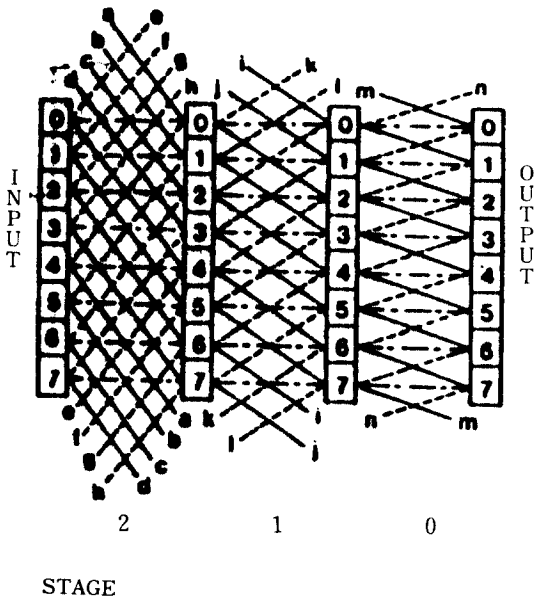


그림 1. 8×8 ADM 접속망. 8×8 ADM network.

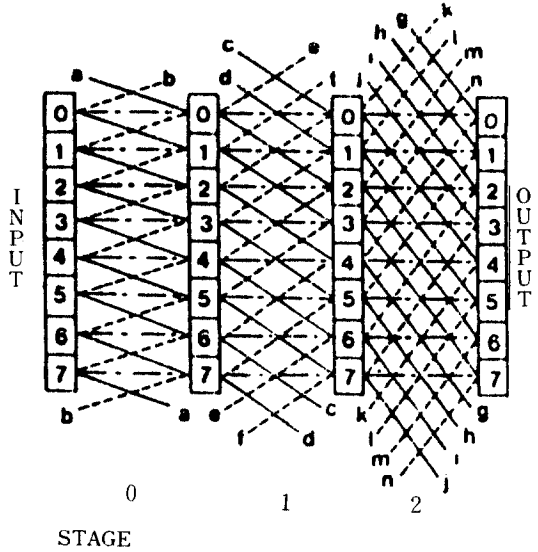


그림 2. 8×8 IADM 접속망. 8×8 IADM network.

접속망을 갖는 시스템 구조에는 P-to-M (Processor-to-Memory) 방식과 PE-to-PE(Processing Element-to-Processing Element) 방식이 있다.¹⁷⁾PE는 프로세서와 메모리로 구성되는 처리소자(Processing Element)이다. 본 논문에서는 시스템 구조로 PE-to-PE 방식을 가정한다. 분석 결과는 P-to-M 방식에도 적용된다.

성능 평가의 요소로 RST(Request Service Time)과 유용도 (Utilization Factor)를 구하였다. RST는 요구 발생부터 그 요구에 의한 데이터 전송이 끝날 때까지의 시간으로 정의된다.

Request Service Time(RST)

$$= \text{경로 설정 시간} + \text{데이터 전송 시간} \quad (2)$$

유용도는 일정 시간동안의 network cycle수와 그동안 데이터 전송에 사용된 cycle수의 비율로 정의한다. 즉,

유용도(Utilization factor)=

$$\frac{\text{데이터 전송에 사용된 cycle의 수}}{\text{총 network cycle수}} \quad (3)$$

이 두가지는 접속망의 성능을 평가, 비교하는 요소로 널리 쓰이고 있으며, RST는 정보 교환의 시간에 관한 것이고 유용도는 접속망 이용도의 척도가 된다.⁽⁹⁾

성능 평가를 위한 접속망의 동작에는 다음과 같은 가정을 사용한다. 입력측에서는 요구를 내는 프로세서를 sender라 하고, 출력측에서 요구를 받는곳을 receiver라 하면,

가정 1): Sender와 receiver사이에 경로가 설정 되면, 이 경로는 평균 d cycle 동안 데이터 전송에 사용된다. 하나의 요구나 끝나면 sender는 다음 cycle에 확률 r에 따라 새로운 요구를 발생한다. 이 정의에 의해 r=1 이면 sender는 항상 요구를 낼 수 있다.

가정 2): 하나의 요구의 목적지가 될 확률은 N개의 receiver가 모두 같다. 즉, 임의의 receiver가 선택될 확률은 모두 같다.

가정 3): 하나의 스위칭소자에 동시에 하나 이상의 요구가 입력되면, 그중 하나만이 임의로 선택되고 나머지는 저지(blocking)된다.

가정 1)은 데이터 전송 시간과 요구 발생 분포를 나타낸다. 가정 2)는 프로세서-메모리 연결 방식에 적용된다. 접속망의 적용 형태에 따라 적절하지 않을 수도 있으나, 일반적으로 대부분의 성능 평가 모델에 적용된다. 가정 3)의 임의의 요구선택은 실제적으로 구현하기 어려우나 해석적 모델에 일반적으로 적용된다. 본 연구에서는 구현 상 가정 1)과 2)는 그대로 사용하고, 가정 3)은 프로그램 구현상 스위치 소자의 입력에 하나 이상의 요구가 동시에 입력되면, plus, straight, minus 링크 순으로 우선권을 갖도록 하였다.

이상과 같은 가정을 전제로 하고, 회선 스위칭 방식은 데이터 전송이 길어짐에 따라 저지(blocking) 시간이 길어지므로 패킷 스위칭 방식과 달리 저지에 의한 영향이 크다. 그러므로 저지당한 요구를 어떻게 처리하는가에 따라 그 성능이 달라진다. 저지 처리 방식으로는 drop과 hold 모델이 제안되고 있다.⁽⁹⁾

Drop 모델이란 요구가 저지되어 저지 신호가 sender에 전달되면, 현재까지 설정된 경로를 지우고 다시 첫번째 단부터 같은 과정을 경로가 완전히 설정될때까지 반복하는 방식이다. Hold 모델은 요구가 저지되더라도 이미 설정된 경로를 유지하면서 경로가 완전히 설정될 때까지 계속 요구를 하는 방식이다.

그림 3은 ADM 상호 접속망의 시뮬레이션 알고리즘을 간단히 나타내었다.

Program ADM-network- Simulation:

```

Begin
Initialization:
For i:=1 to Simulation- Cycles do
  Begin
  For j:=i to Number-of Processors do
    Begin
    if Processor-Status(j)=idle
    then Request
    else if Processor-Status(j)=requesting
    then Proceed
    else if Processor-Status(j)=data-transfer
    then Transfer:
    End:
  End:
  For k:=1 to Number-of-Processors do
  Begin
  if Processor-Status(j)=End-of-Data-Transfer
  then Path-Release:
  End:
  End:
Print-Result:
End.
    
```

그림 3. ADM 접속망의 시뮬레이션 알고리즘.
Simulation algorithm for ADM network.

그림 3에서 Procedure Request에서는 요구율에 따라 요구를 발생시키고, 발생된 요구의 목적지를 발생시킨다. Procedure Proceed에서는 진행 중인 요구를 클럭마다 1단씩 진행시킨다. 저지된 요구는 각 모델마다 제안된 방식에 맞게 처리한다. Procedure Transfer에서는 경로를 이용하여 'd' cycle 동안 데이터를 전송한다. Procedure Path-Release에서는 데이터 전송이 끝난 뒤 설정된 경로를 지우고 sender를 idle상태로 만든다.

이상과 같은 과정을 일정한 cycle동안 반복하여 결과를 구한다.

본 연구에서는 프로세서의 수, 요구율, 데이터 전송 시간을 달리하며 각 경우 80000 node-cycle씩 실행하였다. 80000 node-cycle은 정상적인 동작의 결과를 얻기위한 충분한 실행시간을 제공하는 것으로, 8개의 sender에 대하여 10000 cycle씩, 16개의 sender에 대해서는 5000 cycle씩 실행한 것을 의미한다. 또한 초기 접속망 동작의 과도적인 효과를 없애기 위하여 각 경우 최초 100cycle의 실행 결과는 최종 결과에서 제외하였다. 결과치의 통계적 타당성을 위하여, 80000 node-cycle은 10번으로 나누어 실행하고 평균을 계산하였다.

III. 분할된 상호 접속망에 대한 분석

크기 $N(=2^n)$ 의 상호 접속망이 크기 $M(=2^m)$ 인 독립적인 부접속망으로 나누어 질 수 있다면 분할 가능하다고 한다. $(M \lt N)^n$ 각 부접속망은 원래의 접속망이 할 수 있는 모든 접속함수를 행할 수 있어야 한다. 접속망의 분할 능력은 분할가능 시스템에 사용될 상호 접속망의 설계에 중요한 고려 사항이 된다. ADM 접속망과 IADM 접속망의 분할 능력을 요약하면 다음과 같다. 예를 들어 그림 1에 있는 크기 $N=8$ 인 접속망을 크기 $N=4$ 인 두개의 부접속망으로 분할하려면, 유일한 방법은 그림 4과 같이 입력 포트와 출력 포트가 짝수인 부접속망(A)과 홀수인 부접속망(B)으로 나누는 것이다.

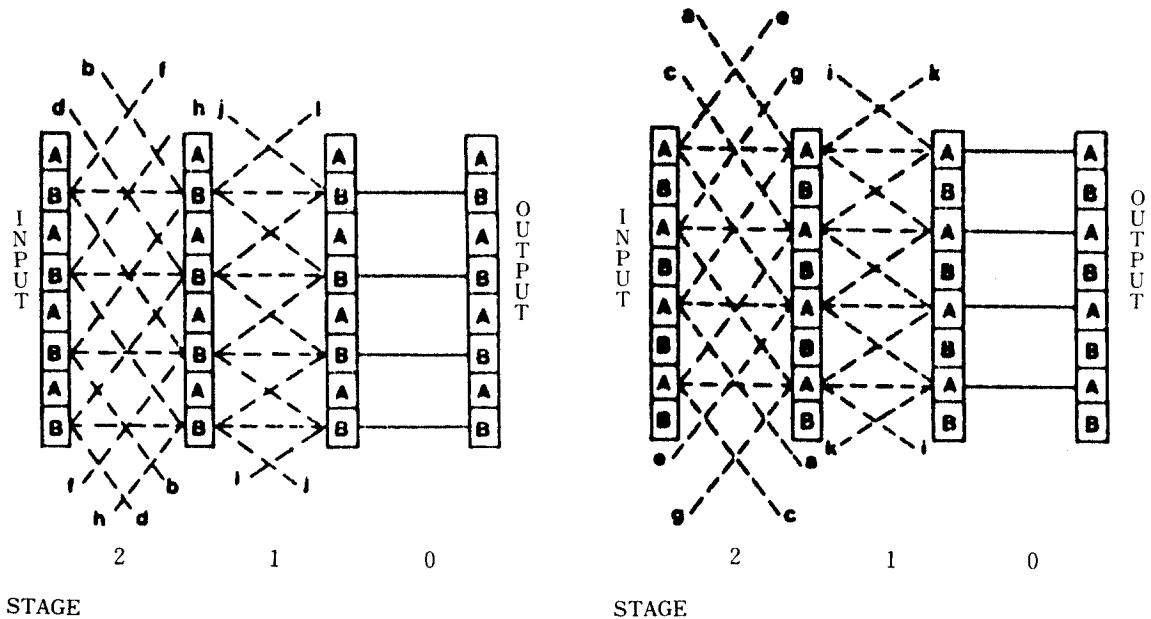


그림 4. ADM 접속망 분할의 예.
Examples of ADM network partitioning.

이 경우 짝수의 PE가 홀수의 PE와 연결되려면 단 0의 $+2^0$ 또는 -2^0 링크를 사용해야 하는데

단 0의 링크를 모두 straight로 설정함으로써 두개의 독립적인 분할이 이루어져 부접속망(A)

는 모두 짝수 PE를 포함하고 부접속망(B)는 모두 홀수 PE를 포함하게 된다. 분할된 접속망의 각 부접속망은 부접속망 크기의 ADM 접속망의 모든 속성을 지니므로, 각 부접속망은 더 나누어질 수 있다. 이 때 제한 사항으로

- 1) 각 부접속망의 크기는 2의 지수 제공이 되어야 하며
- 2) 크기가 2^m 인 부접속망의 입출력 포트의 번호는 그들의 하위 $n-m$ 개의 bit가 일치해야

한다.

그러므로 $N/2$ 크기의 부접속망은 추가로운 bit를 더 선택함으로써 $N/4$ 크기의 두개의 부접속망으로 나누어질 수 있다. 예를 들면 그림 4에서 크기 $N=4$ 인 부접속망 B를 크기 $N=2$ 인 부접속망 C와 D로 나누고자한다면 그림 5와 같이 단1의 링크를 straight로 설정하면 된다.

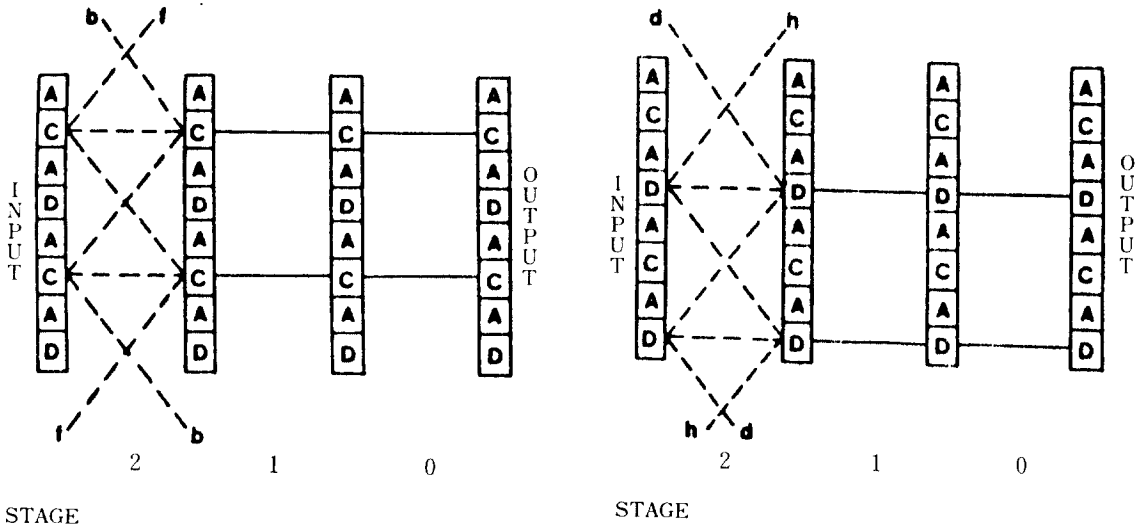


그림 5. 부접속망 분할의 예.
Examples of subnetwork partitioning.

IADM 접속망의 분할은 ADM 접속망과 비슷하다. 즉, 예로 들은 그림 4과 5를 역으로 배열하면 IADM 접속망 분할의 예가 된다. 이러한 접속망의 분할은 routing tag를 masking하여 구현할 수 있다. 일반적으로 입출력 포트 j 가 크기 2^m 인 부접속망에 속한다면, 단 $0, 1, \dots, (n-m)-1$ 에 있는 node j 는 straight로 설정되어야 한다.

다단 상호접속망에서 drop이나 hold방식에 따른 성능 평가는 발표된 바가 있다. 본 연구에서는 이 성능 평가 모델을 확장하여 분할된 접속

망에서의 drop방식과 hold방식에 따른 성능 평가를 하였다. 또한 분할된 ADM 접속망과 분할된 IADM 접속망의 성능을 비교 분석하였다. 이상의 성능 분석은 ADM / IADM 접속망 시뮬레이션을 이용하였다.

III-1. Hold 방식

ADM 접속망과 IADM 접속망은 단 만을 역으로 배열되었을 뿐 topology가 같으므로 분할하지 않았을 때는 거의 같은 성능을 보인다. 접속망을 분할하게 되면 ADM 접속망은 분할 단이 출력

측에 위치하게 되고, IADM 접속망은 분할 단이 입력 측에 위치하게 된다(그림 4). 즉, ADM 접속망의 분할을 출력 분할, IADM 접속망의 분할을 입력 분할이라 할 수 있다.

이러한 분할 단의 위치에 따른 성능의 변화를 부접속망의 크기와 데이터 전송 시간을 달리하면서 구하였다. 분할에 따른 RST의 결과가 표 1에 있다. RST면에서 IADM 접속망은 ADM 접속망에 비하여 분할 크기와 데이터 전송 시간에 따라 최소 3%에서 최대 23% 정도 적게 나타난다. 그림 6에 분할된 접속망에서 요구율 변화에 따른 RST의 변화가 나타나 있다.

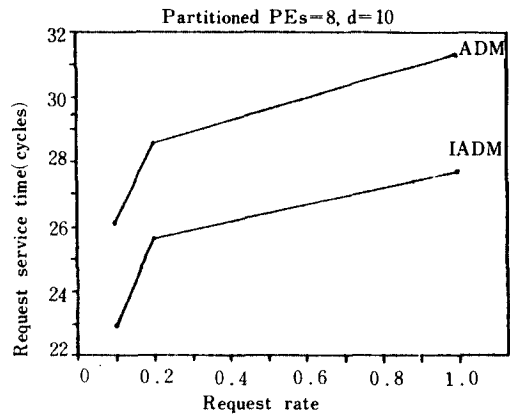


그림 6. 8 PE group으로 분할된 접속망들의 RST(hold 방식, N=32, d=10). RST for networks partitioned to 8 PE groups (hold strategy, N=32, d=10).

표 1. Hold 방식을 사용한 RST (Request Service Time). (N=32)
RST using hold strategy. (N=32)

Data trans. cycles	Request rate	4PE group		8PE group		16PE group	
		IADM	ADM	IADM	ADM	IADM	ADM
5	1.0	13.89	17.19	16.96	20.10	20.07	22.14
	0.2	12.86	15.27	15.01	18.03	18.27	19.36
	0.1	12.33	14.27	13.87	16.15	16.14	17.33
10	1.0	22.45	26.57	27.63	31.30	33.07	35.09
	0.2	21.00	24.31	25.65	28.57	31.33	32.49
	0.1	19.50	22.82	22.87	26.11	27.09	29.92
15	1.0	39.43	43.11	47.48	52.94	58.44	62.30
	0.2	38.25	41.62	46.37	49.63	55.45	56.89
	0.1	35.76	37.81	44.41	48.26	51.02	54.27

이와 같은 결과는 다음과 같이 분석될 수 있다. ADM의 경우에서 분할에 의하여 만들어지는 부접속망은 분할된 단의 앞에 있게 된다. 따라서 요구가 분할된 단에 도달하면, 더 이상 저지 받음이 없이 목적지로 진행하게 된다. m개의 단이 분할되어 있다면, 분할된 단을 진행하는데에는 m network cycle time이 걸린다. 이 요구가 분할된 단을 통과하는데 걸리는 지연 시간은 부접속망 내의 다른 요구들의 저지 확률을 높인다. 이에 비하여 IADM 접속망 경우는 분할된 단이 부접속망보다 앞에 있게 된다. 분할된 단내

에서는 저지가 없으므로 분할된 단을 통과하는데 발생하는 지연 시간은 ADM 접속망과는 달리 부접속망내 저지확률을 높이지 않는다. 따라서 RST가 ADM 접속망보다 적게 된다.

III-2. Drop 방식

표 2에 drop 방식에 의한 분할된 ADM 접속망과 분할된 IADM 접속망의 RST 결과가 있다. RST면에서 IADM 접속망은 ADM 접속망에 비하여 분할 크기와 데이터 전송 시간에 따라 최소 3%에서 최대 11% 정도 적게 나타난다.

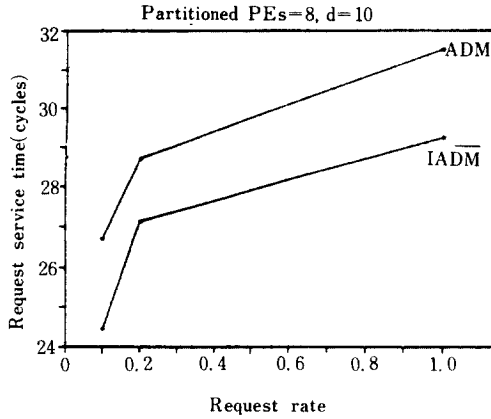


그림 7. 8 PE group으로 분할된 접속망들의 RST(drop 방식, N=32, d=10), RST for networks partitioned to 8 PE groups(drop strategy, N=32, d=10).

그림 7에 drop 방식에서의 요구를 변화에 따른 RST의 변화를 나타내었다.

ADM 접속망에서의 현상은 hold 방식의 경우와 비슷하다. 즉, 분할된 단을 통과하는데 걸리는 지연으로 인하여 부접속망내의 전체적인 저지가 증가된다. IADM 접속망에서 분할된 단에서의 요구 선파지연은 부접속망내 다른 요구의 저지 확률과 관계없다. 분할된 단을 통과한 요구는 부접속망을 진행해나가면서 저지당하면 경로를 해제하고 다시 분할 단을 진행하게 된다. 이러한 반복되는 분할 단의 영향은 부접속망 내의 저지 확률을 낮게한다. 이러한 영향으로 IADM 접속망의 성능은 ADM 접속망보다 좋게 나타난다.

표 2. Drop 방식을 사용한 RST(Request Service Time). (N=32) RST using drop strategy. (N=32)

Data trans. cycles	Request rate	4PE group		8PE group		16PE group	
		IADM	ADM	IADM	ADM	IADM	ADM
5	1.0	16.11	17.87	18.78	20.72	22.33	23.25
	0.2	14.42	16.03	16.72	18.59	19.95	20.91
	0.1	13.43	14.56	15.53	16.67	17.79	18.18
10	1.0	24.44	27.07	29.27	31.52	33.50	35.57
	0.2	22.58	24.73	27.14	28.74	30.48	32.66
	0.1	21.00	22.67	24.44	26.71	28.02	30.29
15	1.0	41.43	44.01	49.43	52.09	56.44	58.22
	0.2	39.51	39.71	44.90	47.38	52.27	52.99
	0.1	37.44	38.36	44.32	46.13	49.58	51.67

III-3. 성능 비교

분할되지 않은 접속망에서의 성능 평가 결과는 drop 방식이 hold 방식보다 낮은 RST를 보였다. 그러나 분할 가능한 접속망은 앞에서 분석한 바와 같이 저지 처리방식과 분할 방법에 따라 비교해 보아야 한다. 표 1과 표2를 보면 저지 처리 방법에 관계없이 IADM 접속망이 항상 낮은 RST를 보인다. 저지 처리 방법으로 비교해 보면 일반적으로 hold 방식이 drop 방식에 비하여 작은 RST값을 보인다. 그림 8은 ADM 접속망에서의 저지 처리 방법의 비교를 나타내고, 그림 9는 IADM 접속망에서의 비교를 나타내었

다. 저지 처리 방법에 따른 비교는 데이터 전송 시간이 짧을때 가장 두드러지게 나타난다. 이것은 데이터 전송 시간이 짧을 경우에는 전송 시간에 의한 지연보다 저지 처리 방식 자체의 효과가 큰 영향을 미치기 때문이다. 즉 drop 방식에서 저지는 분할된 단을 다시 통과하는 추가적인 지연이 생기지만 hold 방식에는 없다. 이러한 점에서 분할이 사용된다면 대부분의 경우 drop 보다 hold가 좋을 것이라는 것을 예상할 수 있다.

데이터 전송시간이 길어진다면, RST의 증가는 부접속망 내의 저지 확률에 의해 지배되므로

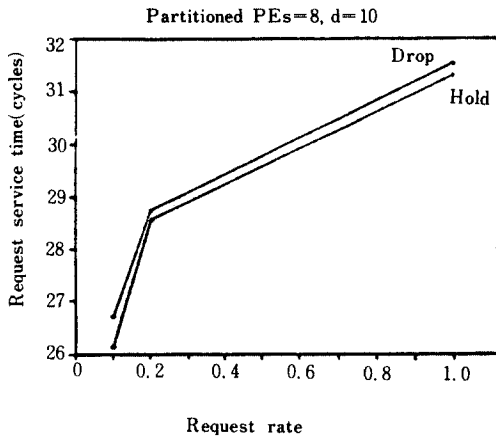


그림 8. 8 PE group으로 분할된 ADM 접속망의 RST (N=32, d=10).
RST for ADM network partitioned to 8 PE group (N=32, d=10).

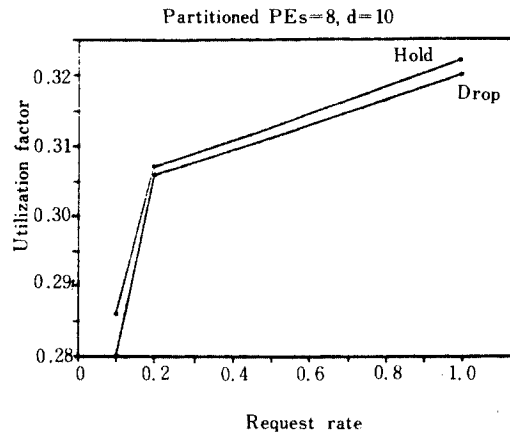


그림 10. 8 PE group으로 분할된 ADM 접속망의 유용도 (N=32, d=10).
Utilization factor for ADM network partitioned to 8 PE group (N=32, d=10)

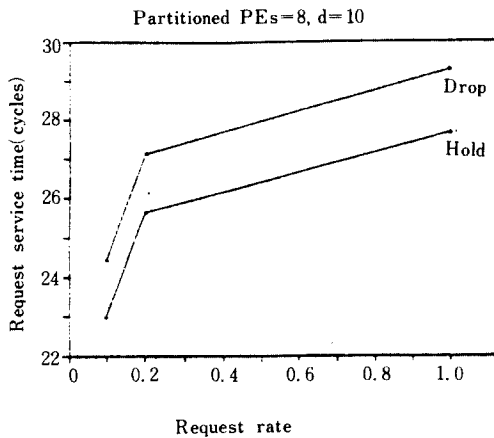


그림 9. 8 PE group으로 분할된 IADM 접속망의 RST (N=32, d=10).
RST for IADM network partitioned to 8 PE group (N=32, d=10).

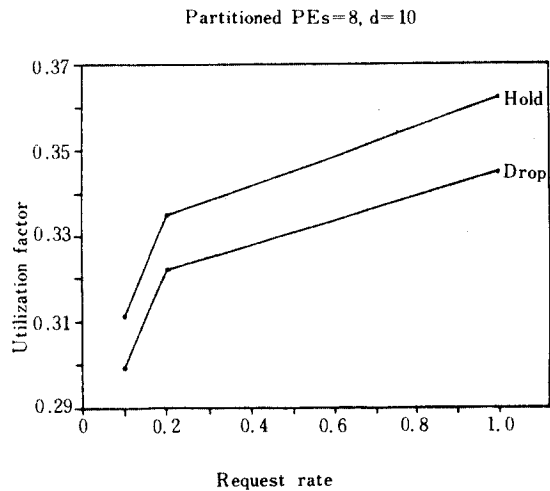


그림 11. 8 PE group으로 분할된 IADM 접속망의 유용도 (N=32, d=10). Utilization factor for IADM network partitioned to 8 PE group (N=32, d=10).

저지 처리 방식 자체의 영향은 적어진다. 접속망의 이용도를 나타내는 유용도의 경우도 hold 방식이 drop 방식보다 나은 것으로 나타난다. 그림 10와 그림 11은 ADM 접속망과 IADM 접속망에 대한 유용도를 나타낸다.

위의 분석 결과 접속망 기능면에서 IADM 접속망이 ADM 접속망보다 나음을 알 수 있지만, ADM 접속망의 출력 측에서 입력시킨다면 IADM 접속망과 같은 결과를 얻을 수 있을 것이다.

IV. 결 론

분할된 다단상호접속망의 성능 평가를 위하여 회선스위칭 ADM과 IADM 접속망을 모델링하여 pascal 언어로 시뮬레이터를 구현하였다. 이 시뮬레이터를 이용하여 ADM과 IADM 접속망의 분할시 동작을 시뮬레이션하고, RST와 유용도로 그 결과를 비교 분석하였다. 시뮬레이션 결과, 저지처리방식에 관계없이 IADM 접속망이 ADM 접속망보다 RST면에서 최대 23% 만큼 낮은 값을 나타내었다. 이 이유는 분할된 단이 부접속망 뒤에 위치하는 ADM 접속망에서 분할된 단을 통과하는 지연 시간이 부접속망 내의 저지확률을 높이는데 비하여, 부접속망 앞에 분할된 단이 위치하는 IADM 접속망은 지연 시간이 부접속망 내의 저지확률을 높이지 않는 것으로 분석된다. 따라서 분할된 단이 입력 측에 위치하는 IADM 접속망이, 분할된 단이 출력 측에 위치하는 ADM 접속망보다 정보 교환 면에서 효율적인 것을 알 수 있다. 또한 저지처리방식의 비교에서는 hold 방식이 나은 것으로 나타나는데, 이 이유는 drop 방식의 경우 분할된 단을 다시 통과하는 추가적인 지연 시간 때문에 전체적인 RST 값이 커지는 것으로 분석 된다. 그러므로 분할 가능 시스템에서 hold 방식을 사용하는 IADM 접속망이 가장 나은 접속망 동작을 보인다는 것을 알 수 있다. 이 분석 결과로 접속망의 분할이 접속망 동작에 새로운 결과를 나타내는 것을 알 수 있다. 본 논문에서는 제시한 상호 접속망 동작에 관한 분석은 앞으로 분할가능 시스템의 상호접속망 설계에 활용할 수 있을 것이다.

1. K. Hwang and F. A. Briggs, "Computer Architecture and parallel processing," McGraw-Hill, 1984.
2. C. Wu, "Interconnection networks in multiple-processor systems," Wayne state Univ. PH.D., 1978.
3. C. L. Wu and T. Y. Feng, "Tutorial: Interconnection networks for parallel and distributed processing," IEEE com. soc. press, 1984.
4. R. J. McMillen, "A study of Multistage interconnection networks," Purdue Univ. PH.D., 1982.
5. H. J. Siegel and S. D. Smith, "Study of multistage SIMD interconnection networks," 5th Annual Int'l Symp. Computer Architecture, pp.223-229, Apr. 1978.
6. S. D. Smith, "Use of Augmented Data Manipulator Multistage Network for SIMD Machines," Proceedings 1980 Int'l conf. on Parallel Processing, pp.75-890, 1980
7. H. J. Siegel, "Interconnection Networks for Large-Scale parallel Processing System," Lexington Books, 1985.
8. A. L. Overvig, "The simulation of the generalized cube interconnection network," Purdue Univ. M.S., 1982.
9. M. Lee, "Design and analysis of an interconnection network for multiprocessing," Texas Univ. PH.D., 1986.
10. G. B. Adams, "On the Number of Permutations performable by the Augmented Data Manipulator Network," IEEE Transactions on Computers, Vol.C-31, pp.75-80, Apr.1982.
11. R. J. McMillen "Routing Schemes for the Augmented Data Manipulator Network in an MIMD System," IEEE Transactions on Computers, vol.C-31, pp.184-196, Dec. 1982.

參 考 文 獻



金 煥 善 (Young Sun KIM) 正會員
1956年11月29日生
1980年 2月：高麗大 電子工學科 卒業
(工學士)
1982年 2月：高麗大學院 電子工學科 卒
業 (碩士)
1987年 9月～現在：高麗大學院 電子 工
學科 博士課程
1982年 3月～現在：韓國電子通信研究所
시스템 2室長



崔 眞 圭 (Jin Kyu CHOE) 正會員
1958年 9月20日生
1980年 2月：高麗大學校 電子工學科 卒
業
1982年 2月：高麗大學校 大學院 電子工
學科 卒業 (工學碩士)
1987年 8月：高麗大學校 大學院 電子工
學科 博士課程 卒業 (工學
博士)
1987年 9月～現在：國立 大田工業 大學
電子計算學科 助教授