분산시스템에서 부하공유 알고리즘의 천이특성 해석

(Analysis of Transient Characteristics of Load Sharing Algorithms In Distributed Systems)

박세명*, 안광선**

Abstract

Load sharing in distributed systems improves systems performance. Research in the field has been focused on steady state performance for load sharing algorithms. However, transient charateristics of such algorithms may be important in a distributed system in which workload for some node is changing. Simulation is the only means to analyze such characteristics. This paper presents a simulation-based analysis of the transient characteristics of four load sharing algorithms: forward probing, reverse probing, symmetric probing, and multi-threshold symmetric probing algorithms. Discrete event system models for execution of the algorithms in a distributed system has been developed in a SIMSCRIPT II.5 environemnt. Simulation results indicate that the MSYM algorithm shows the shortest response time in the transient period.

1. 서 론

분산시스템에서 처리자원을 효율적으로 사용하여 사용자에게 더 나은 처리속도를 제공하기 위한 방법으로 부하공유(Load Sharing)에 관한 연구가 활발하다. 부하공유는 각 처리기에 입력되는 작업을 처리기의 부하와 작업을 전송하기 위한 통신비용등을 고려하여 유휴한 처리기에 적절히 할당하는 방식을 사용한다. 이를 위해서는 통신비용 및 작업의 성질에 따라서 입력되는 작업을 적절한 단위로 분류하고 분류된 각 단위 작업들을 유휴 처리능력이 있는 처리기에 할당하는 부하공유 알고리즘이 필요하게 된다.

워크스테이션들이 통신망을 통해서 연결되어 있는 환경에서 사용자에 의해서 제출된 작업은 일반적으로 다른 작업과 완전히 독립적이며 순차적으로 수행되는 제어구조를 가지고 있다. 이러한 경우에는 제출된 각 작업을 처리기에 할당하는 단위로 설정하는 것이 바람직하며 이와 관련된 연구로는 임계값을 이용하는 부하공유(Threshold Load Sharing)[2-6, 8, 9, 12]를 들 수 있다. 임계값을 이용한 부하공유 방식은 공통적으로 처리되는 작업이 CPU -bound 임을 가정하고 있으며, 제안된 알고리즘들은 크게 전진탐색(forward probing)과 후진탐색(reverse-probing) 그리고 이들을 혼합한 대칭탐색(symmetric-probing)으로 분류된

* 인제대학교 전산학과
** 경북대학교 컴퓨터공학과

다.

기존의 연구에서는 제안된 알고리즘의 성능을 평가하기 위해서 처리기의 부하와 전송비용의 변화에 따른 작업반환시간을 사용하며, 제안된 알고리즘이 시스템의 상황에 따라 최적의 성능을 보장하기 위해서 임계값의 적절한 조정(fine-tuning)이 필요함을 보이고 있다. 또한 알고리즘의 성능평가를 위한 환경으로 부하 안정상태(steady state)에서 동일부하(homogeneous workload) 환경이거나 이부하(heterogeneous workload) 환경을 고려하고 있다. 즉 동일부하 환경하에서는 시스템의 부하가 저부하이거나 고부하로 고정된 환경, 또는 이부하 환경하에서는 부하가 서로 다른 처리기들의 비율이 고정된 환경하에서 알고리즘의 성능을 평가하고 있다. 그러나 실 상황하에서는 예측할 수 없는 부하의 변화가 발생하고 있으며 이러한 상황에 대한 부하공유 알고리즘 동작특성의 분석은 전무한 실정이다.

본 논문에서는 처리기의 부하가 일정한 시간에 따라서 변화하는 입력(즉, 천이상태 입력)형태를 3가지로 분류하고, 각각의 부하상황하에서 기존의 전진탐색 알고리즘, 후진탐색 알고리즘, 대칭탐색 알고리즘 그리고 MSYM 알고리즘의 부하 변화상태(천이상태 : transient state) 동작특성을 실험을 통해서 분석한다. 알고리즘의 천이특성을 해석하기 위한 가장 효율적인 방법은 시뮬레이션을 사용하는 것이다. 따라서 본 논문에서는 분산시스템에서 부하공유 알고리즘의 동작을 이산사건 시스템으로 모델링하여 SIMSCRIPT II.5 환경에서 시뮬레이션을 수행하였다.

실험 결과는 천이상태에서의 동작특성은 안정상태에서의 동작특성과는 달리 부하가 변화하는 구간의 폭이 기존의 알고리즘의 동작특성을 결정하는 중요한 인자로 고려되어야 함을 보여주고 있는데 이러한 동작특성은 각 알고리즘이 부하의 변화에 적응하는 형태와 밀접한 연관을 가짐을 볼 수 있다. 처리기의 부하, 작업의 전송비용, 그리고 부하 변화구간의 폭에 무관하게 MSYM알고리즘이 우수한 동작특성을 보장함을 보이고 있다.

본 논문은 다음과 같이 구성되어 있다. 2절에서는 부하공유 알고리즘과 관련된 연구를 간략하게 소개한다. 3절에서 시뮬레이션 모델과 실험환경을 설명한다. 다음 절에서 실험으로 부터 얻은 결과를 분석하며 끝으로 5절에서 논문의 결론을 맺는다.

2. 관련 연구 및 비교 대상 알고리즘

[9]는 전진탐색 방식에서 지역작업(local job)과 외부작업(remote job)의 도착분포 변화를 추정하는 변화추정자(gradient estimator)를 제안하고 각 처리기들이 변화추정자를 이용하여 추정된 정보를 주기적으로 교환함으로써 임계값을 자동 조정하는 방식을 제안하였다.

[4,5,6]에서는 3가지 방식의 수학적 분석을 이용하여 대칭탐색방법이 후진탐색과 전진탐색에 비해서 처리기의 부하와 작업 전송시간의 변화에 따라서 개선된 작업반환시간을 가짐을 보이고 있다. 또한 각 처리기의 부하와 전송되는 작업의 전송시간의 변화에 따른 최적 임계값을 결정하였으며 전송시간이 일정한계 이상으로 증가하게 되면 최적 임계값을 결정하더라도 부하공유를 하지 않는것이 유리한 상황이 발생함을 보이고 있다.

따라서 작업전송비용의 변화에 따라서 임계값을 적절히 조절하는 기능, 즉 처리기의 부하의 변화에 따라서 작업의 전송율을 적절히 조정하는 기능을 가지는 알고리즘이 필요한데 이와 관련된 연구로는 실 시스템에서 제출된 작업중에서 CPU-bound작업 또는 다른 처리기에서 처리하는 것이 유리한 작업을 식별하는 방법에 대한 연구[13]와 Ethernet 을 기반으로 하는 분산처리시스템에서 전송비용이 처리시간의 1/10일때 부하의 변화에 따른 임계값의 조정문제를 해결한 연구[10], 그리고 전송비용의 변화에 따라서 작업의 전송율을 적절히 조정함으로써 개선된 작업반환 시간을 보장하는 MSYM (Multiple Threshold Symmetric Probing)알고리즘[11]을 들 수 있다.

기존의 연구는 처리기의 부하와 전송비용의 변화에 따라서 알고리즘의 임계값을 조절하는 기능을 부여하는 방법에 주안점을 두고 있으며 따라서 다양한 형태의 부하의 변화에 대한 알고리즘의 성능의 평가는 이루어지지 못한 것으로 판단된다. 따라서 본 논문에서는 위의 4가지 알고리즘이 부하 천이상태하에서 어떠한 동작특성을 보여주는가를 시뮬레이션을 통해 해석한다. 또한 시뮬레이션으로 부터의 결과가 기존의 연구결과와도 어떠한 관련이 있는지를 분석한다.

본 논문에서 고려하는 4개의 알고리즘의 대략적인 동작은 다음과 같이 기술된다.

FRD(전진탐색 알고리즘) : 각 처리기의 작업 전송시도

절차와 작업 전송시도에 대한 응답은 아래와 같다.

〈작업 전송시도 절차〉

1. 각 처리기는 지역작업이 도착할 때 대기중인 작업의 수(처리중인 작업 포함)가 임계값보다 크면 작업의 전송을 시도한다. 그렇지 않으면 전송을 시도하지 않는다.
2. 작업 전송이 타당하면 임의로 선택된 L_p개의 처리기를 탐색한다.
3. 탐색 패킷에 대한 응답을 기다려서 긍정적으로 응답한 처리기로 작업을 전송한다. 만약 긍정적으로 응답한 처리기가 둘 이상이면 임의로 하나를 선택한다. 긍정적인 응답이 하나도 없으며 탐색은 실패하고 새로운 작업이 도착할 때까지 기다린다.

〈작업 전송시도에 대한 응답〉

1. 탐색을 받은 처리기는 대기중인 작업의 수(처리중인 작업 포함)가 임계값보다 작으면 긍정적인 응답을 하고 그렇지 않으면 부정적인 응답을 한다.

REV(후진탐색 알고리즘) : 작업 전송을 요구하는 절차와 작업 전송요구에 대한 응답은 다음과 같다.

〈작업 전송요구 절차〉

1. 각 처리기는 하나의 작업이 수행완료되었을때 대기중인 작업의 수가 임계값보다 작고, 작업의 전송을 요구하지 않았다면 작업의 전송을 요구한다.
2. 작업의 전송 요구를 위해서 임의로 선택된 L_p개의 처리기를 탐색한다.
3. 탐색 패킷에 대한 응답을 기다려서 긍정적인 응답을 한 처리기에 작업의 전송을 요구한다. 만약 긍정적인 응답을 한 처리기가 둘 이상이면 임의로 하나를 선택하여 작업의 전송을 요구한다. 긍정적인 응답을 한 처리기가 없으면 탐색은 실패하고 새로운 작업의 수행이 완료될 때까지 기다린다.

〈작업 전송요구에 대한 응답〉

1. 탐색을 받은 처리기는 대기중인 작업의 수(처리중인 작업 포함)가 임계값보다 크면 긍정적인 응답을 하고 그렇지 않으면 부정적인 응답을 한다.

SYM(대칭탐색 알고리즘) : FRD와 REV의 경우를 결합한 형태로써 각 처리기는 새로운 작업이 도착하면 전송시도절차를 수행하고 작업의 수행이 완료되면 작업 전송요구 절차를 수행한다.

MSYM(다중 임계값을 갖는 대칭탐색 알고리즘) : 각 처리기에서 작업 전송시도, 작업 전송시도에 대한 응답, 작업 전송요구, 그리고 작업 전송요구에 대한 응답의 기준으로 정의된 4개의 임계값을 사용한다. 작업전송시도 절차와 그에 대한 응답, 작업 전송요구 절차와 그에 대한 응답은 아래와 같다.

〈작업 전송시도 절차〉

1. 각 처리기는 지역작업이 도착할 때 $N_{sp} \neq 0$이면, T_{sp}를 구하여 $T_{sp} > 0$면 도착한 작업의 전송을 시도한다. 그렇지 않으면 전송을 시도하지 않는다.
2. 작업 전송이 타당하면 임의로 선택된 L_p개의 처리기를 탐색한다.(탐색 패킷에 계산된 $T_{sr}(=T_{sp})$을 실어서 보낸다.)
3. 탐색 패킷에 대한 응답을 기다려서 긍정적으로 응답한 처리기로 작업을 전송한다. 만약 긍정적으로 응답한 처리기가 둘 이상이면 임의로 하나를 선택한다. 긍정적인 응답이 하나도 없으며 탐색은 실패하고 새로운 작업이 도착할 때까지 기다린다.

〈작업 전송시도에 대한 응답〉

1. 탐색을 받은 처리기는 $N_{sr} < T_{sr}$이면 긍정적인 응답을 하고 그렇지 않으면 부정적인 응답을 한다.

〈작업 전송요구 절차〉

1. 각 처리기는 하나의 작업이 수행완료되었을때 $N_{rp} < T_{rp}$이고, 이미 작업의 전송을 요구하지 않았다면 작업의 전송을 요구한다.
2. 작업의 전송 요구를 위해서 임의로 선택된 L_p개의 처리기를 탐색한다.(탐색 패킷에 N_{rp}를 실어서 보낸다.)
3. 탐색 패킷에 대한 응답을 기다려서 긍정적인 응답을 한 처리기에 작업의 전송을 요구한다. 만약 긍정적인 응답을 한 처리기가 둘 이상이면 임의로 하나를 선택하여 작업의 전송을 요구한다. 긍정적인 응답을 한 처리기가 없으면 탐색은 실패하고 새로운 작업의 수행이 완료될 때까지 기다린다.

〈작업 전송요구에 대한 응답〉

1. 탐색을 받은 처리기는 $T_{rr} > 0$ 이면 긍정적인 응답을 하고 그렇지 않으면 부정적인 응답을 한다.

다음절에서 분산시스템에서 위의 4가지 알고리즘의 동

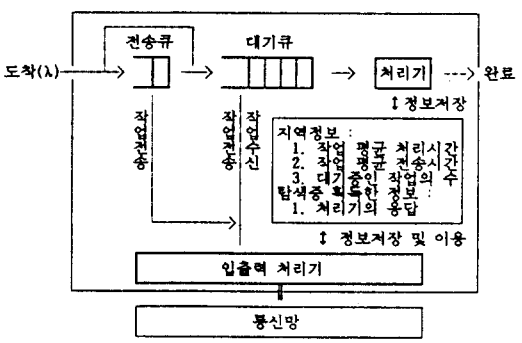
작을 이산사건 시스템으로 모델링한다.

3. 시뮬레이션 모델

본 연구에서 고려한 분산시스템의 시뮬레이션 모델은 여러개의 처리기와 이들을 연결하는 통신망으로 구성되며, 본 절에서는 처리기 모델과 통신망 모델, 그리고 실험환경에 대해서 언급한다. 본 연구를 위한 실험환경은 Simscript II.5를 이용하여 구축하였다.

3.1 처리기 모델

본 연구에서 고려하는 각 처리기의 구조는 〈그림 1〉과



〈그림 1〉 처리기 구조

같다. 처리기는 작업을 처리하는 처리기와 입출력을 전담하는 처리기로 구성된다. 처리기는 전송큐와 대기큐를 가지며, 작업을 입력순서대로 처리한다. 입출력 처리기는 DMA(Direct Memory Access) 기능과 프로그램 수행기능이 있어 입출력에 관련된 모든 사항, 즉 전송큐나 대기큐에 저장된 작업의 전송, 수신된 작업을 대기큐에 저장, 탐색패킷의 전송및 수신, 공유하는 정보의 갱신및 이용, 수신 패킷의 응답을 처리기의 간섭없이 독자적으로 처리한다. 따라서 각 처리기는 작업의 처리와 입출력의 수행을 병행할 수 있다. 이와 같은 상황하에서 각 처리기에서의 동작은 Simscript II.5하에서 4개의 프로세스, 즉 GEN, SPROB, RPROB, SEL.NTASK로 구성하였으며 각각의 동작은 다음과 같다.

프로세스 GEN은 설정된 부하에 따라서 지역작업을 생성시키며, 생성된 새로운 작업의 배치를 담당한다. 즉 새로운 지역작업이 생성되면 해당 처리기의 상태에 따라서 전송큐나 대기큐에 저장하고, 전송큐에 저장할 경우에는 저장후에 SPROB를 호출한다. 프로세스 SPROB는 전진탐색을 담당하는 프로세스이며, 호출되면 임의로 선택된 처리기를 탐색하여 작업을 전송할 처리기를 선정하고 선정된 처리기로 작업을 전송하며, 전송큐에 저장된 작업이 없으면 동작을 중지한다. 그리고 전송이 실패하면 전송큐의 작업을 대기큐로 이동시킨다. 프로세스 SEL.NTASK는 하나의 작업이 종료되면 처리할 새로운 작업을 선택하는 기능을 가지며 이를 위해서 우선 대기큐를 검사한다. 대기큐에서 대기중인 작업이 없으면 전송큐를 검사하여 처리할 작업을 선택한다. 전송큐에도 작업이 없으면 RPROB를 호출한다. 프로세스 RPROB는 후진탐색을 담당하는 프로세스이며, 호출되면 작업을 전송가능한 처리기를 선정하여 작업을 수신하는 기능을 담당한다.

이러한 기능을 통해서 각 처리기는 작업이 도착하면 우선 전송큐에 저장되며 전송이 타당하면 입출력 처리기는 전송절차를 수행하며, 작업의 전송이 타당하지않거나 탐색에 실패하면 전송큐의 작업을 대기큐로 이동한다. 전진탐색에 의해 수신된 작업의 저장과 후진탐색에 의한 전송작업의 선택은 대기큐의 마지막에서 일어난다.

3.2 통신망 모델

통신망은 각 처리기사이의 작업 및 탐색패킷의 전송을 담당하며 모든 처리기사이에 통신이 가능한 구조를 가정하였다. 단지 모든 처리기간에 통신비용은 작업처리시간의 일정한 비율을 평균으로 하는 지수분포를 따른다고 가정한다.

3.3 실험 환경

본 논문에서 전송비용의 변화에 따른 부하의 천이상태, 즉 외부작업의 도착 비율이 변화하는 상황하에서 기존의 부하공유 알고리즘, 즉 전진탐색 알고리즘(FRD: forward probing algorithm), 후진탐색 알고리즘(REV : reverse probing algorithm), 그리고 대칭탐색 알고리즘(SYM : symmetric probing algorithm)과 다중임계값을 갖는 대칭탐

색 알고리즘(MSYM : multiple threshold symmetric probing algorithm)의 성능을 비교하기 위해서 다음과 같은 실험환경을 설정하였다.

실험을 위해서 사용된 인자는 다음과 같다.

1) 작업 도착율(λ: job arrival rate)
2) 평균 처리시간(S: service time)
3) 탐색한계
4) 전송지연 시간(C: transfer delay time).

S와 C는 각각 평균 μ,γ인 지수분포를 따르고, μ=1 (UT: unit time)로 가정하였고, $\gamma = R * \mu$(R=0.1, 1.0, 그리고 10)인 경우를 고려하였다. 그리고 처리기의 부하(ρ)는 $\mu^*\lambda$로 나타낼수 있다. 탐색한계는 기존의 연구에서 '2' 또는 '3'이 적절하다는 결론에 따라 '3'으로 고정하였다. 그리고 천이 상태의 동작을 비교하기 위해서 다음과 같이 3개의 작업 도착율 모델을 사용하였다.

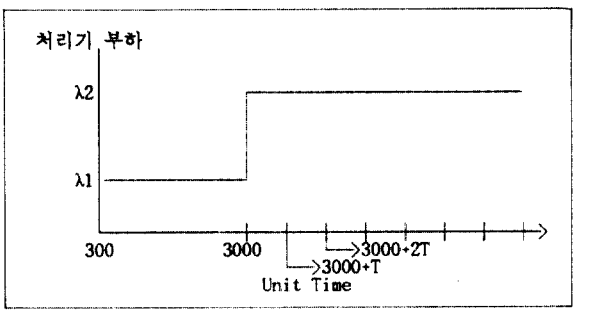
〈그림 2〉, 〈그림 3〉, 그리고 〈그림 4〉에서 가로축은 작업의 평균 처리시간과 같은 단위를 가지는 시간의 변화를 나타내며, 세로축은 작업 도착율을 나타낸다. 〈그림 2〉, 〈그림3〉에서는 처리기의 부하를 3000(Unit TIme)까지 λ1 (λ2)으로 지속한후 λ2(λ1)로 변경하였다. 〈그림 4〉에서는 〈그림 2〉와 〈그림 3〉이 혼합된 형태를 고려하고 있으며, 3000시간 이후에는 매 T(Unit TIme)마다 작업 도착율을 λ1에서 λ2, 또는 λ2에서 λ1으로 변화시켰다.

실험에서는 λ1은 0.2와 0.5를 사용하였고, λ2는 0.9를 사용하였으며, 그림2)와 그림3)에서는 부하의 변화에 따른 작업반환 시간의 변화를 상세히 고찰하기 위해서, 그림4)의 경우에 처리기의 부하가 변화되는 시간폭의 효과를 고찰하기 위해서 구간폭을 이용하였으며, 구간 T는 60, 30, 15인 경우를 고려하였다.

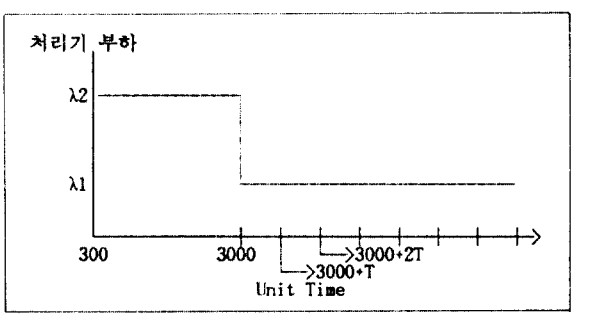
이와 같은 실험 환경하에서 알고리즘의 성능을 비교하기 위해서 각 구간 즉, (300,3000], (3000,3000+T], (3000+T,3000+2*T] ... (3000+5*T, 3000+6*T]에서 실행완료된 작업의 평균 작업반환 시간을 구하고, 서로 다른 10개의 seed를 이용하여 독립된 실험을 반복하여 각 구간별로 평균을 구하여 성능을 비교하였으며, warm-up 효과를 제거하기 위해서 300(Unit TIme)동안에 처리된 작업을 제거하였다. 본 논문에서 사용한 작업반환시간은 아래와 같이 정의한다.

작업 반환시간(JRT: Job Response Time)

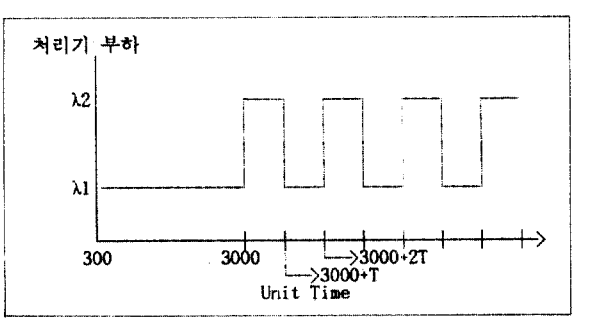
$$= 1/N^* \sum_{i=1}^{N} (J_i\text{의 완료시간} - J_i\text{의 제출시간})$$



〈그림 2〉 처리기의 부하 변화형태 1.



〈그림 3〉 처리기의 부하 변화형태 2.



〈그림 4〉 처리기의 부하 변화형태 3.

(J_i : i번째 처리된 작업, N : 각구간에서 처리된 작업의 수)

4. 시뮬레이션 결과 및 고찰

$\lambda1=0.2$, 0.5 그리고 $\lambda2=0.9$인 경우에 대해서 R을 0.1, 1.0 그리고 10.0으로 고정한 상태에서 T를 300, 60, 30, 15으로 변화시키면서 네가지 알고리즘의 부하천이특성을 고찰하였으며, FRD, REV, SYM의 임계값은 1인 경우를 고려하였다. $\lambda1=0.5$인 경우에도 $\lambda1=0.2$인 경우와 유사한 결과를 보이고 있으므로 본 절에서는 $\lambda1=0.2$인 경우에 대해서 언급한다.

4.1 처리기의 부하변화 형태가 〈그림 2〉인 경우

〈그림 5-a,b,c,d〉는 R=0.1이고 T=300, 60, 30, 그리고 15인 경우를 각각 나타내며, 저부하에서 고부하로 변화되는 경우 알고리즘의 동작특성을 나타낸다. 〈그림 5-a〉의 경우처럼 작업반환 시간을 측정하는 구간을 300으로 두는 경우에서는 처리기의 부하의 변화에 따른 알고리즘의 동작에는 별 영향을 주지 않는것 같아 보인다. 그러나 3000에서 3300구간을 확대해서 작업반환 시간을 측정하는 구간을 60, 30, 15로 변화시켜 보면 다음과 같은 동작 특성을 볼 수 있다. REV가 나머지 알고리즘에 비해서 부하의 변화에 민감하게 반응하여 부하가 변화하는 구간에서는 오히려 FRD보다 작업반환 시간이 나빠지는것을 보이나 FRD보다는 짧은 시간안에 작업반환 시간이 안정화되는 것을 볼 수 있다. 그리고 SYM과 MSYM은 부하의 변화에 따라서 적응하는 형태는 유사하나 FRD와 REV에 비해서 개선된 작업반환 시간을 보이면서 부하의 변화에 빨리 적응하여 안정화되는 것을 볼 수 있다. 그리고 MSYM은 SYM에 비해서 안정화이후에는 개선된 작업반환 시간을 보임을 알 수 있다.

4.2 처리기의 부하변화 형태가 〈그림 3〉인 경우

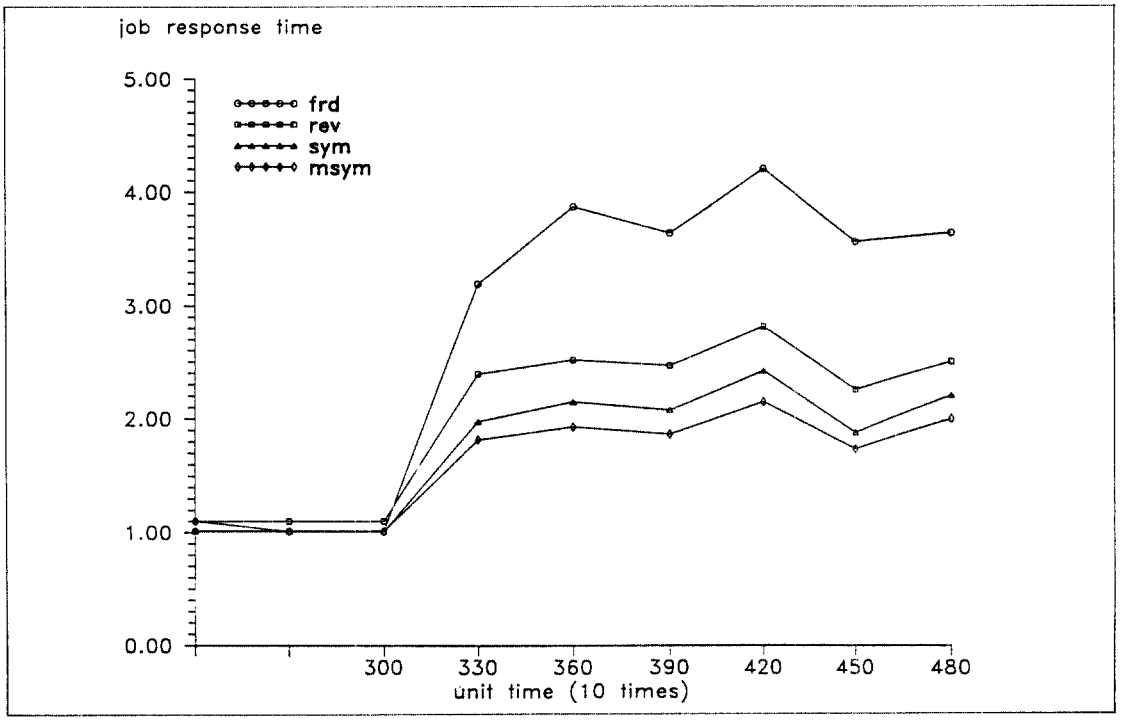
〈그림 6-a,b,c〉는 R=0.1이고 T=60, 30, 그리고 15인 경우를 각각 나타내며 고부하에서 저부하로 변화하는 경우 알고리즘의 동작특성을 나타낸다. 그리고 저부하에서 고부하로 변화하는 경우와는 달리 4개의 알고리즘이 거의 유사한 시간내에 안정된 작업반환 시간을 보이는 것을 볼 수 있다.

4.3 처리기의 부하변화 형태가 그림4)인 경우

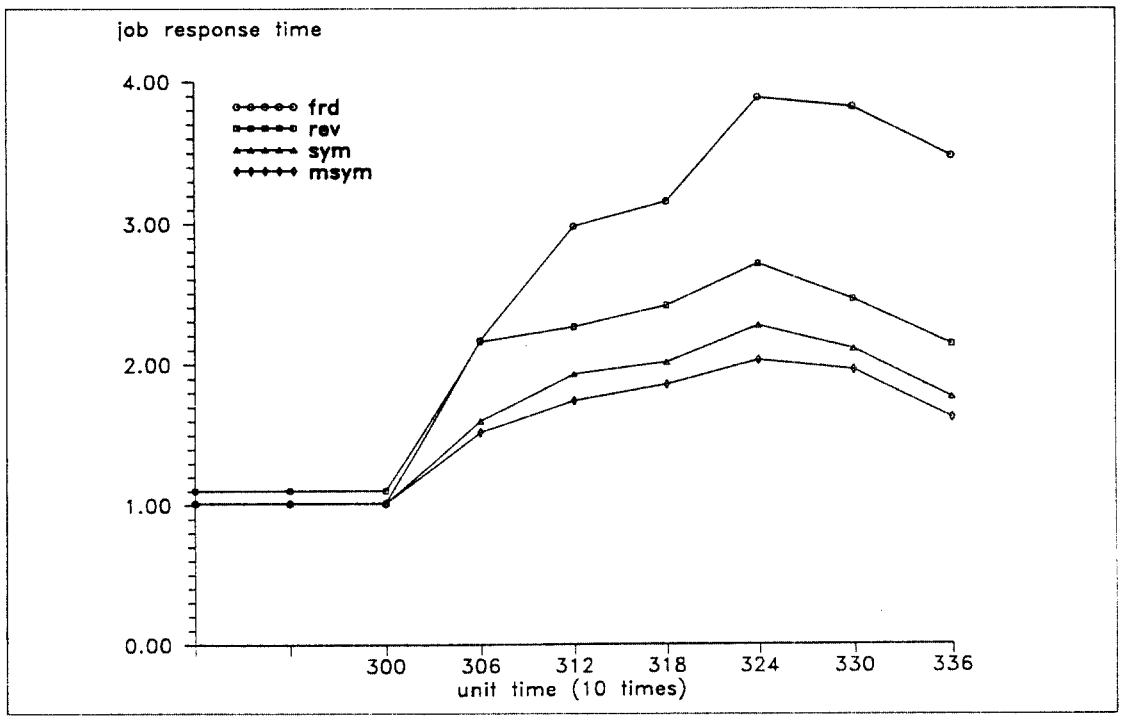
〈그림 7-a,b,c〉는 R=0.1일때 T=60인 경우, T=30인 경우, T=15인 경우를 각각 나타낸다. 우선 저부하 안정상태 구간인 [300, 3000]에서는 기존의 결과와 동일하게 SYM, FRD, 그리고 REV순으로 개선된 작업반환 시간을 보이고 있다. 그러나 부하천이 구간에서는 부하의 변화 폭(T)에 따라서 기존의 결과와는 상이한 결과를 보이는데 상세한 내용은 다음과 같다. T=60인 경우에는 부하안정상태와는 달리 구간의 부하와는 무관하게 REV가 FRD에 비해서 근소하나마 개선된 작업반환 시간을 보이고 있다. T=30인 경우에는 REV가 FRD에 비해서 부하의 변화에는 민감하게 반응하고 있으나 FRD에 비해서 반드시 우수한 작업반환 시간을 보장한다고는 할 수 없는 상황을 보여준다. 그러나 T=15인 경우에는 구간의 부하와 무관하게 오히려 FRD가 REV에 비해서 우수한 작업반환 시간을 보장함을 보여주고 있다. 이상의 결과를 통해서 부하천이 구간에서는 부하의 변화폭(T)이 작아질수록 FRD가 REV에 비해서 개선된 성능을 보인다는 사실을 알수 있고 또한 SYM와 MSYM은 유사한 특성을 보여준다.

〈그림 8-a,b,c〉는 그림7의 환경에서 전송비용이 10배로 증가한 경우를 나타내고 있는데 이처럼 전송비용이 증가하면 또다른 동작특성을 나타낸다. 이러한 경우에 나타나는 또다른 동작특성은 구간의 폭(T)에 무관하게 REV의 성능이 가장 나쁜것을 볼수있고, 구간의 폭(T)이 작아질수록 SYM가 FRD와 유사한 성능을 보이고 있음을 볼 수 있다. 이러한 특성은 전송에 따른 비용이 증가함에도 불구하고 SYM의 전송형태는 유지됨으로써 전송을 하는 것이 작업반환시간을 개선하는데 도움을 주지 않는다는 것을 보여주고 있으며, 이러한 효과는 부하의 변화폭이 작을수록 커짐을 알 수 있다. 그러나 MSYM의 경우는 SYM에 비해서 구간의 폭에 무관하게 개선된 성능을 보임을 알 수 있고 또한 전송비용의 증가에 따라서 전송을 제어하는 기능을 가짐으로써 개선된 성능을 유지함을 보여준다.

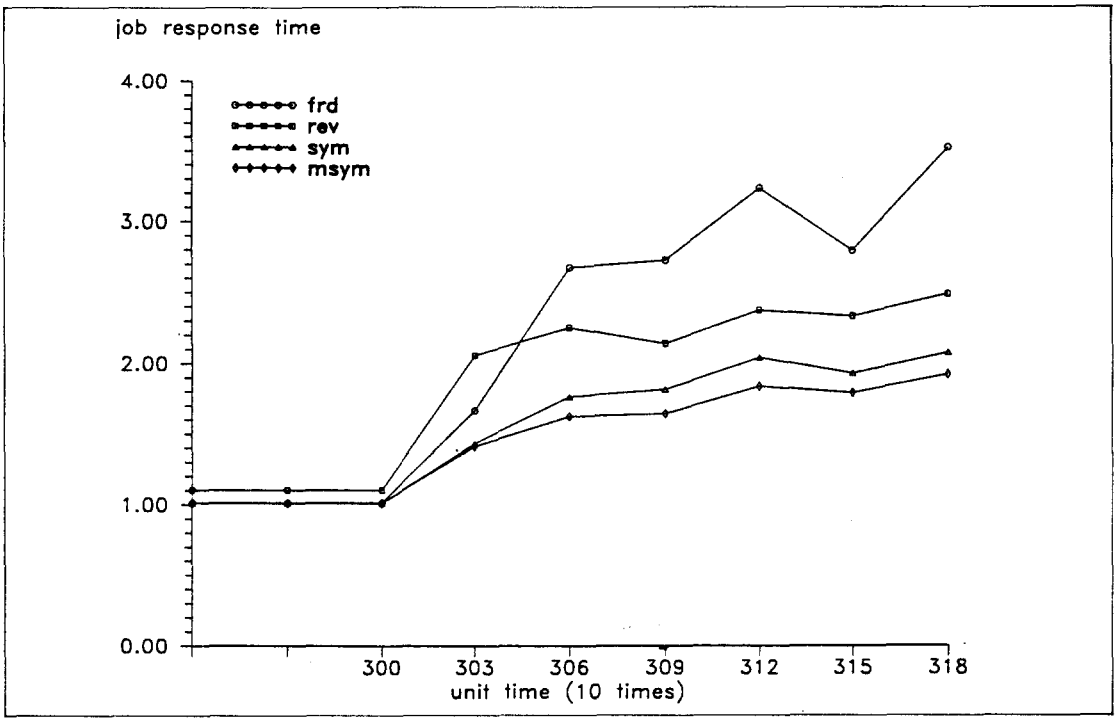
〈그림 9-a,b,c〉는 R=10.0일때, T=60인 경우, T=30인 경우, T=15인 경우를 각각 나타낸다. (300, 3000)구간에서는 MSYM, REV, FRD, 그리고 SYM순의 성능을 나타내고, 나머지 구간에서도 SYM의 성능이 가장 나쁘고 REV와 FRD는 성능의 비교가 어려우며, MSYM가 가장 우수한 성능을 보이고 있다. 이러한 특성은 주로 전송비용의



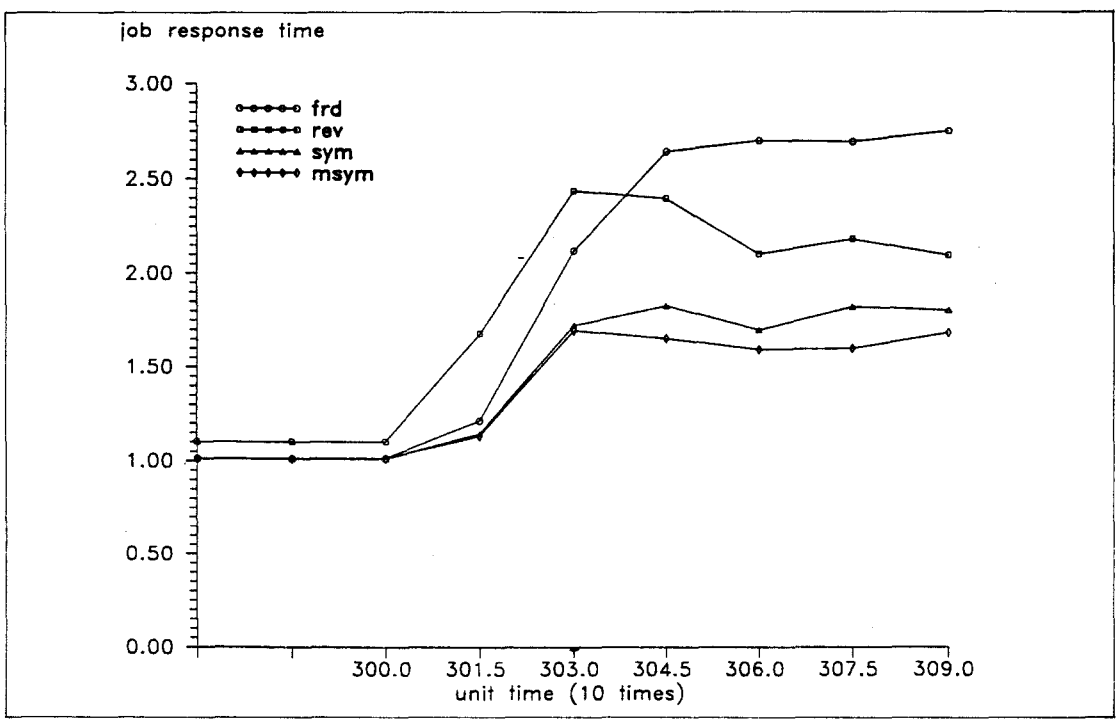
〈그림 5-a〉 처리기의 부하변화 형태 1.의 경우 작업반환 시간
(T=300, R=0.1)



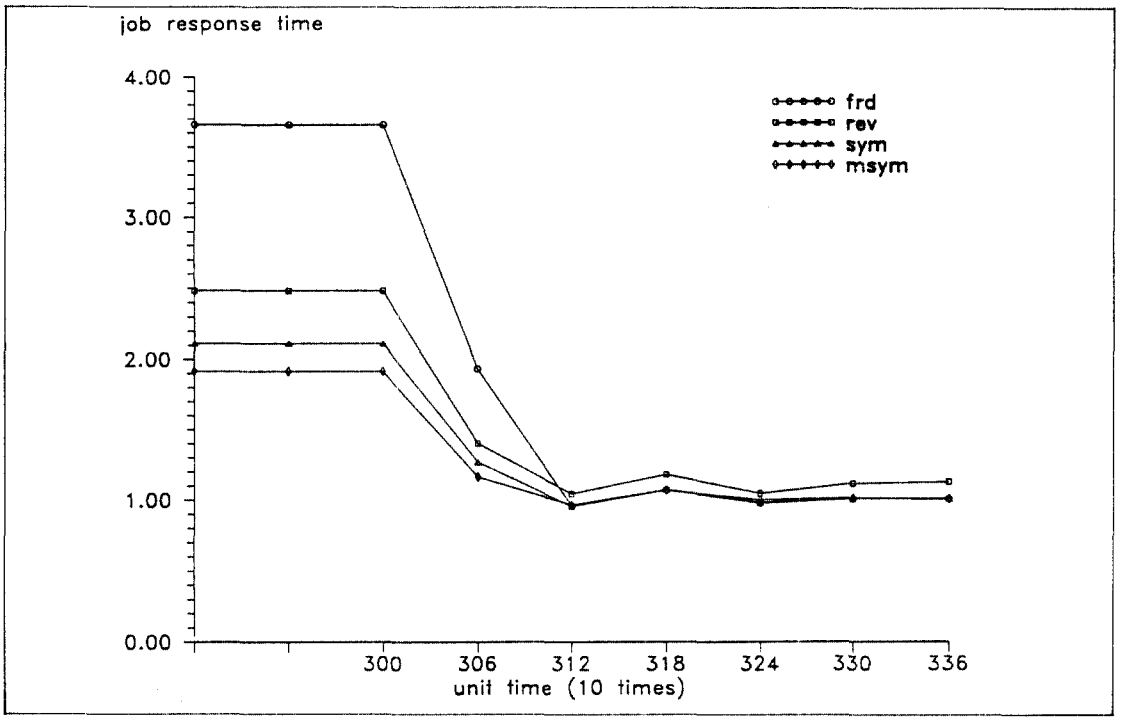
〈그림 5-b〉 처리기의 부하변화 형태 1.의 경우 작업반환 시간
(T=60, R=0.1)



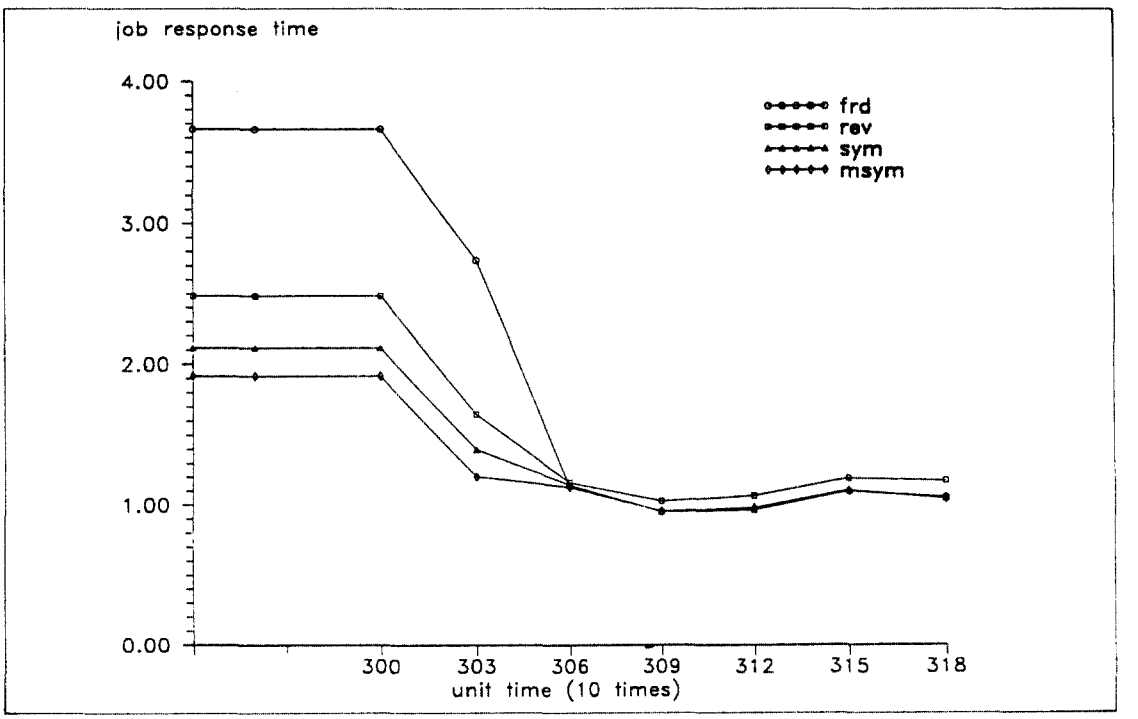
〈그림 5-c〉 처리기의 부하변화 형태 1.의 경우 작업반환 시간
(T=30, R=0.1)



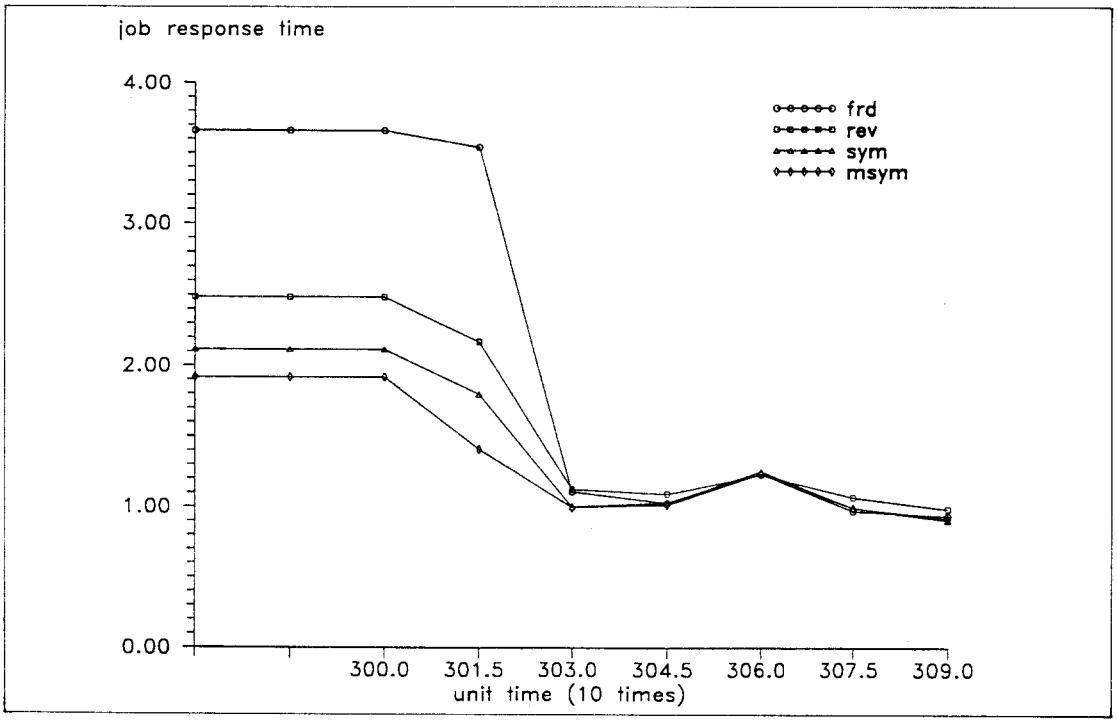
〈그림 5-d〉 처리기의 부하변화 형태 1.의 경우 작업반환 시간
(T=15, R=0.1)



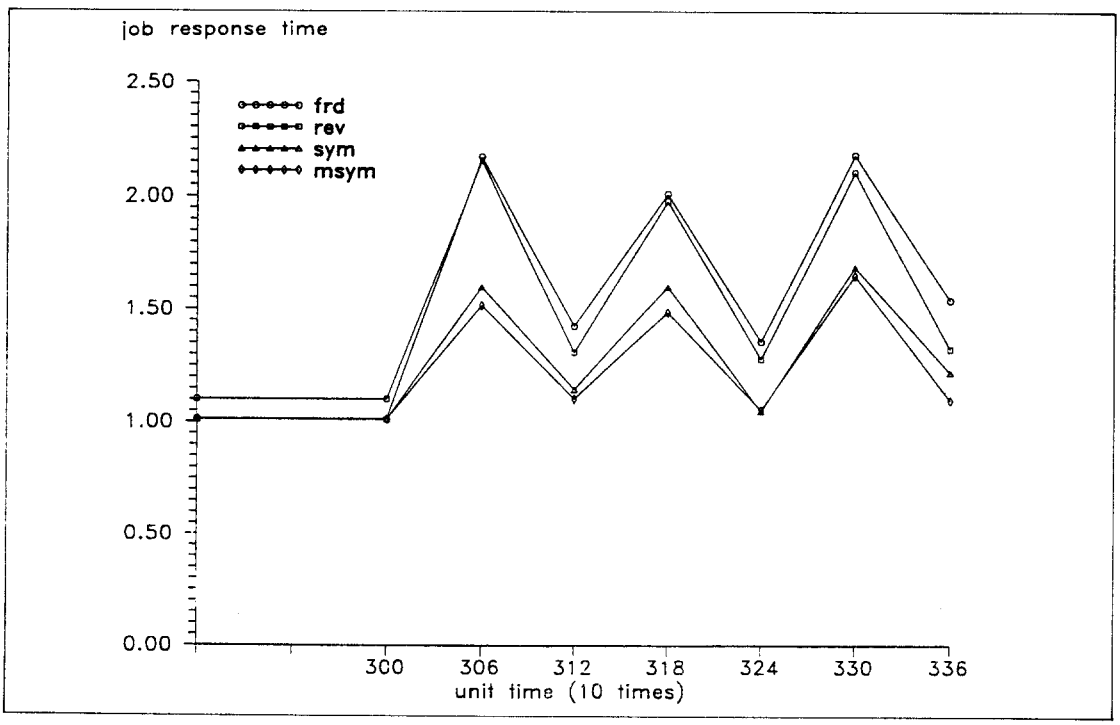
〈그림 6-a〉 처리기의 부하변화 형태 2.의 경우 작업반환 시간
(T=60, R=0.1)



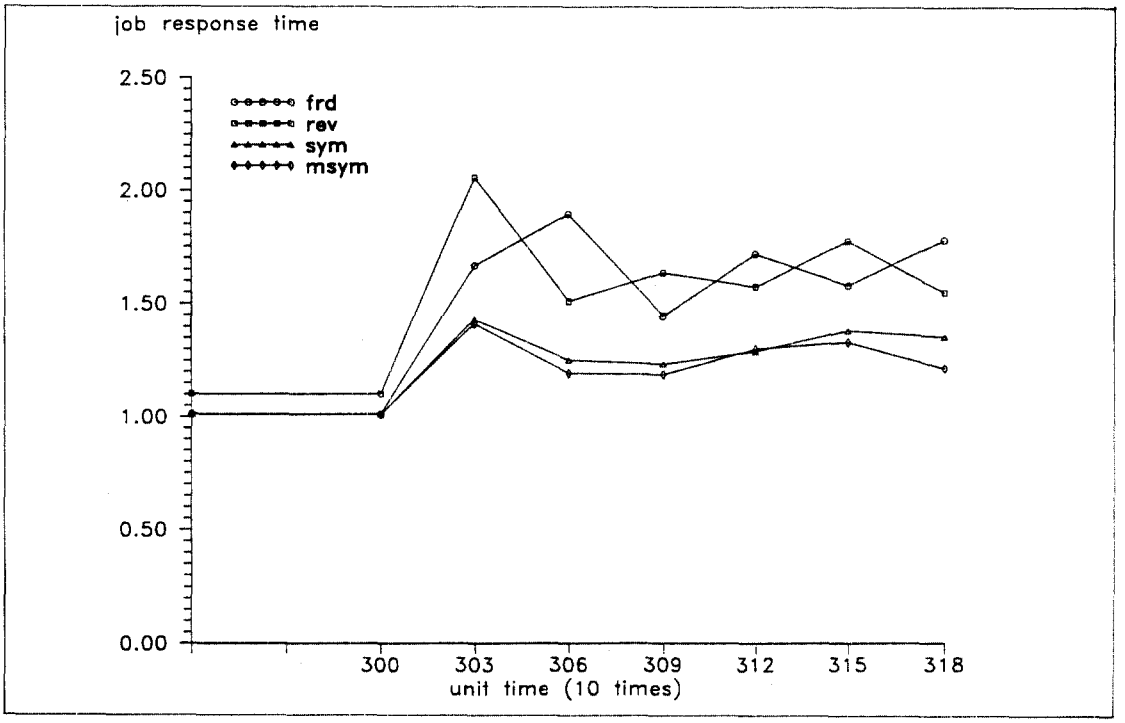
〈그림 6-b〉 처리기의 부하변화 형태 2.의 경우 작업반환 시간
(T=30, R=0.1)



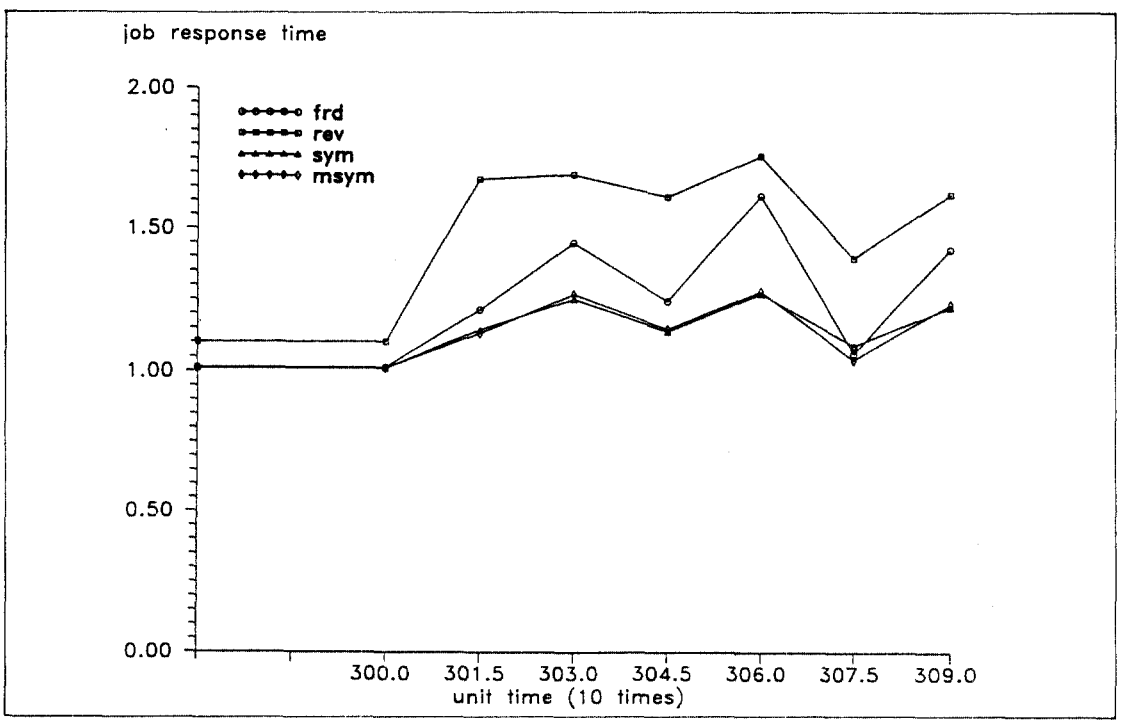
〈그림 6-c〉 처리기의 부하변화 형태 2. 의 경우 작업반환 시간
(T=15, R=0.1)



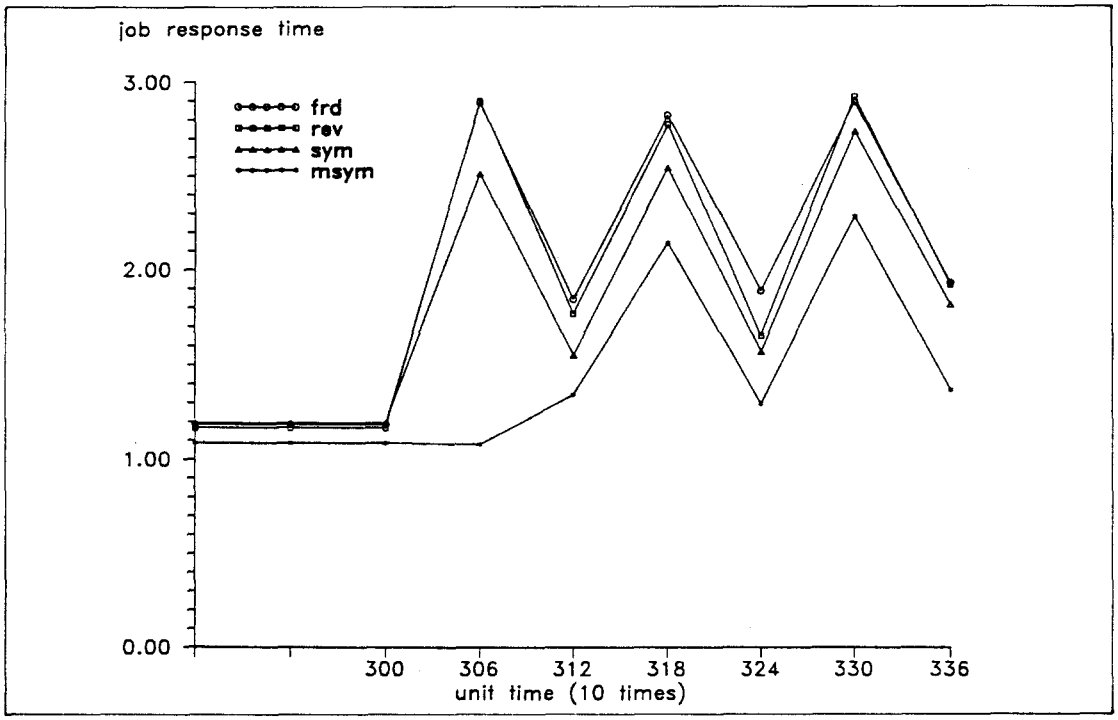
〈그림 7-a〉 처리기의 부하변화 형태 3. 의 경우 작업반환 시간
(T=60, R=0.1)



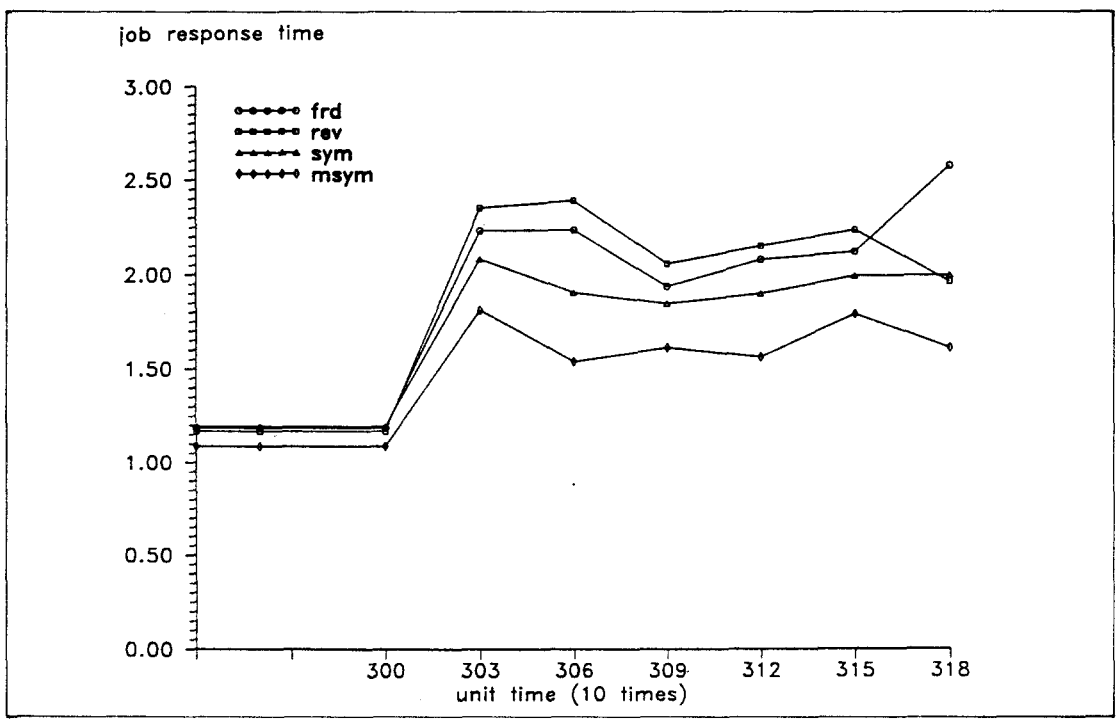
〈그림 7-b〉 처리기의 부하변화 형태 3.의 경우 작업반환 시간
(T=30, R=0.1)



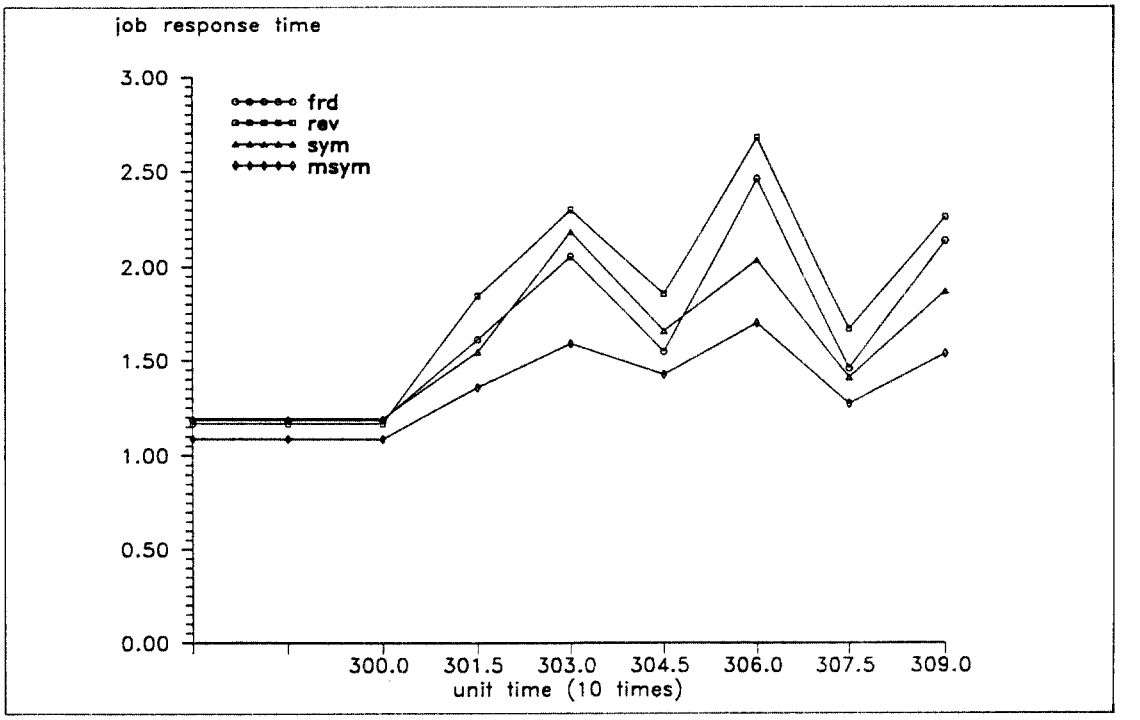
〈그림 7-c〉 처리기의 부하변화 형태 3.의 경우 작업반환 시간
(T=15, R=0.1)



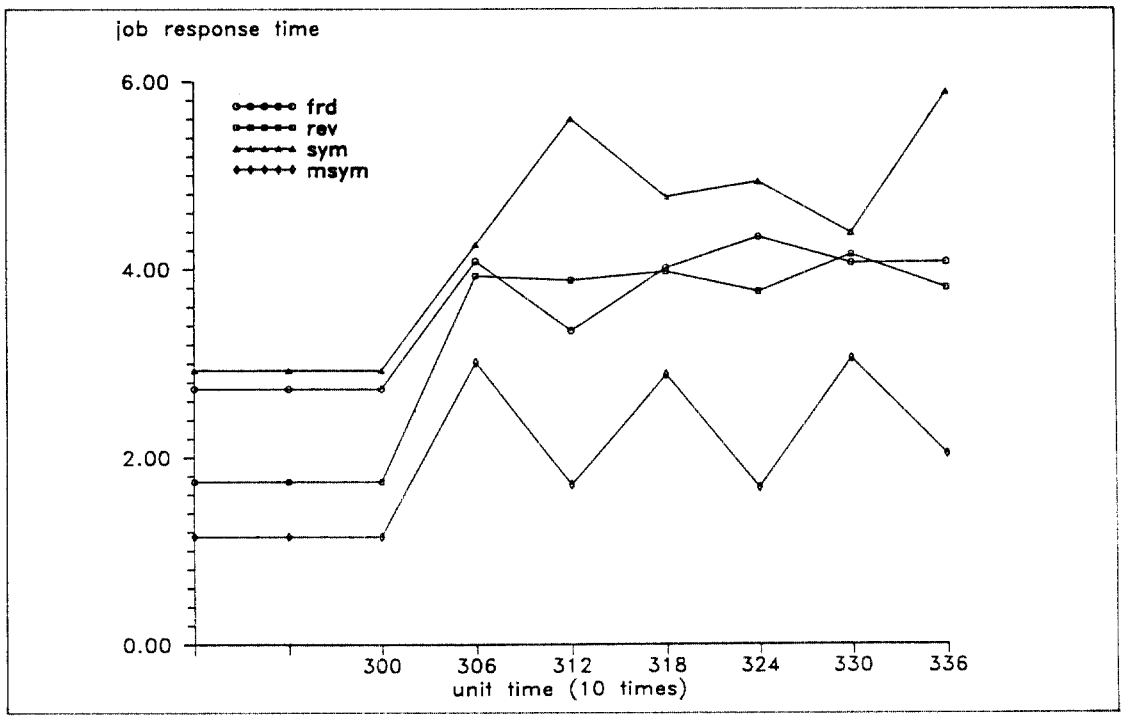
〈그림 8-a〉 처리기의 부하변화 형태 3.의 경우 작업반환 시간
(T=60, R=0.1)



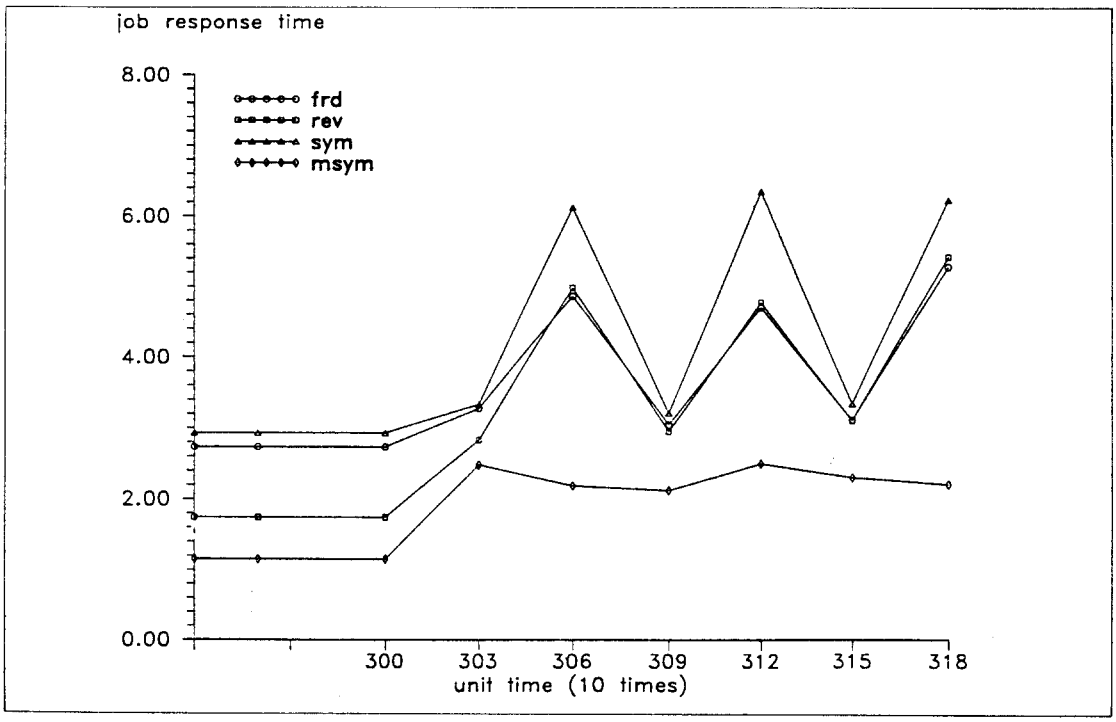
〈그림 8-b〉 처리기의 부하변화 형태 3.의 경우 작업반환 시간
(T=30, R=0.1)



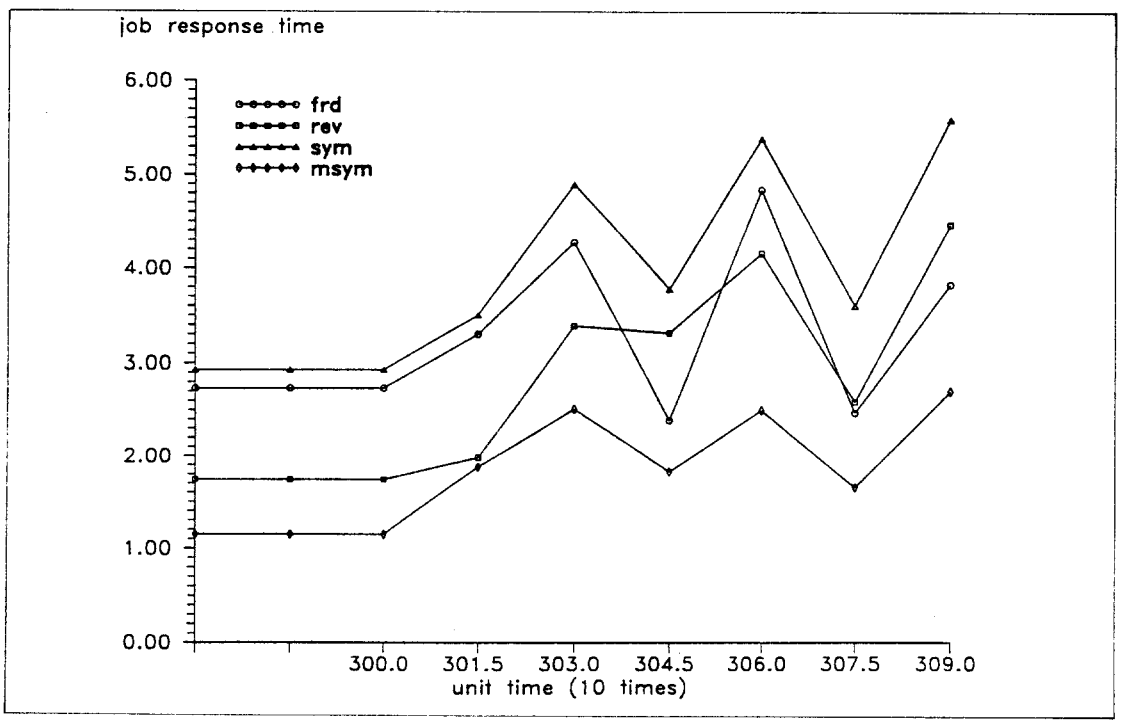
〈그림 8-c〉 처리기의 부하변화 형태 3.의 경우 작업반환 시간
(T=15, R=0.1)



〈그림 9-a〉 처리기의 부하변화 형태 3.의 경우 작업반환 시간
(T=60, R=10.0)



〈그림 9-b〉 처리기의 부하변화 형태 3.의 경우 작업반환 시간
(T=30, R=10.0)



〈그림 9-c〉 처리기의 부하변화 형태 3.의 경우 작업반환 시간
(T=15, R=10.0)

증가로 인한 전송의 효율성이 가장 중요한 요인인 것으로 판단된다. 그리고 부하의 변화에 적응하는 형태를 고려해 보면 작업의 전송비용에 무관하게 T가 작은 경우에는 각 구간의 부하와 작업반환 시간의 증감이 반대현상을 보이고 있는데 이와 같은 현상은 4.1과 4.2의 결과를 통해 볼 때, 각 알고리즘이 부하의 변화에 적응하는 시간이 필요하므로 구간에서 도착된 작업의 처리가 해당 구간에서 완료되지 못하고 다음 구간에서 처리 완료되기 때문으로 판단되며, 이러한 현상은 4개의 알고리즘에서 공통적으로 나타나고 있음을 볼 수 있다. 그럼에도 불구하고 작업반환 시간의 측면에서 보면 MSYM가 가장 우수한 성능을 보임을 볼 수 있다.

이상의 결과에서 SYM, FRD, 그리고 REV는 작업의 전송비용과 부하의 변화폭에 따라서 여러 형태의 동작을 보임을 알 수 있다. 따라서 기존의 3개의 알고리즘을 실상황에 적용하는 경우에는 처리기의 부하와 전송비용 그리고 처리기 부하의 변화폭도 함께 고려되어야 함을 알 수 있다. 그러나 MSYM의 경우는 이상의 세가지 요인에 무관하게 개선된 성능을 보임을 알 수 있다. 따라서 실시스템에서 발생 가능한 부하와 전송되는 작업의 전송비용의 여러 형태의 변화에 대해서 MSYM이 나머지 3개의 경우보다 우수한 성능을 보이리라 판단된다.

5. 결 론

본 논문에서는 부하천이 상태에서 전송비용의 변화에 따른 기존의 부하공유 알고리즘들의 특성을 시뮬레이션을 통해 고찰하였다. 이를 위해 분산시스템에서 각 알고리즘의 동작을 이산사건 시스템으로 모델링한 후에 이를 SIMSCRIPT II.5 환경에서 구현하여 시뮬레이션 환경을 구축하였다. 각 처리기에 입력되는 작업도착율을 3가지 형태의 부하변화 형태로 모델링하여 반복적인 실험을 수행하였다.

시뮬레이션 결과로부터 안정상태에서는 대체로 저부하인 경우에는 FRD의 성능이 REV보다 성능이 우수하고, 고부하의 경우에는 REV가 FRD보다 우수한 성능을 보이며, 부하에 무관하게 SYM가 FRD와 REV에 비해서 우수한 작업반환 시간을 보장함을 알 수 있었다. 그리고 전송비용이 증가하면 FRD, REV, 그리고 SYM의 임계값을 적절히 조정할 수 있어야 한다. 안정상태에서는 전송비용과 처리기의 부하만을 고려하면 되지만 천이상태에서는 천이구간의 폭도 중요한 인자로 고려해야 하는 것을 알 수 있다. 따라서 부하와 작업의 전송비용 그리고 부하의 변화폭을 알 수 없는 실 상황하에 기존의 세 알고리즘을 적용하려면 많은 어려움이 따르리라 예상된다. 그러나 MSYM 알고리즘은 이상에서 언급한 여러 상황에 무관하게 가장 개선된 작업반환 시간을 보장함을 실험을 통해서 알 수 있었다. 따라서 MSYM알고리즘은 부하의 천이구간의 폭이나 전송비용 그리고 처리기의 부하의 변화를 예측할 수 어려운 실상황하에서 기존의 세가지 방안보다 개선된 작업반환 시간을 보장하리라 판단된다.

[참고 문헌]

[1] Yung-Terng Wang and Robert J.T. Morris, "Load Sharing in Distributed System," IEEE Trans. on Comput., Vol.c-34, No.3, pp.204-217, MARCH 1985.

[2] D.Eagat, E.Lazowska, and J.Jahorjan, "Adaptive Load Sharing in Homogeneous Distributed Systems," IEEE Trans. on Software Engineering, SE-12, pp.662-675, May 1986.

[3] Niranjan G. Shivaratri, Phillip Krueger, and Mukesh Sunghal, "Load Distributing for Locally Distributed Systems," IEEE Computer, pp.33-44, Dec. 1992.

[4] Ravi Mirchandaney, Don Towsley, and John A. Stankivic, "Adaptive Load Sharing in Heterogeneous System, 9th International Conf. on Distributed Computing Systems," pp.298-306. 1989.

[5] Ravi Mirchandaney, Don Towsley, and Jhon A. STANKOVIC, "Analysis of the Effects of Delays on Load Sharing," IEEE TRANS. on Comput., Vol.38, No. 11, pp.1513-1525, Nov. 1989.

[6] Ravi Mirchandaney, Don Towsley, and John A. Stankivic, "Adaptive Load Sharing in Heterogeneous Distributed Systems," Journal of Parallel and Distributed Computing, pp.331-346, 1990.

[7] Thomas Kunz, "The Influence of Different Workload Descriptions on a Heuristic Load Balancing Scheme," IEEE Trans. on Software Engineering, Vol.17, No.7, pp.725-730, July. 1991.

[8] Niranjan G.Shivaratri and Phillip Krueger, "Two Adaptive Location Policies for Global Scheduling Algorithm, 10th International Conf. on Distributed Computing Systems," pp.502-509. 1990.

[9] Spiridon Pulidas, Don Towsley, and John A.Stankovic, "Imbedding Gradient Estimators in Load Balancing Algorithms, 8th International Conf. on Distributed Computing Systems," pp.2482-490,1988.

[10] 박세명, 안광선,"An Effective Load Sharing Algorithm for Heterogeneous Distributed System," 정보과학회 논문지, VOL.20, NO.8, Augeust 1983.

[11] 박세명, 안광선,"Symmetric-probing Load Sharing Algorithm using Multi-Threshold," to be published.

[12] Margaret Schaar, Kemal Efe, Lois Delcambre, and Laxmi N. Bhuyan, "Load Balancing with Network Cooperation, 11th International Conf. on Distributed Computing Systems," pp.328-335,1988.

[13] Anders Svensson, "History, an Intelligent Load Sharing Filters, 10th International Conf. on Distributed Computing Systems," pp.546-553, 1990.

저자소개

박세명

1983년 2월 경북대학교 전자공학과 전자계산전공 졸업
1985년 2월 경북대학교 전자공학과 전자계산전공 석사학위 취득
1990년 2월 경북대학교 전자공학과 전산기공학 전공 박사과정 수료
1990년 3월~현재 인제대학교 전산학과 교수
관심분야 : 분산 및 병렬처리, 분산 운영체제, 분산 데이타베이스, 이산사건 시뮬레이션

安光善

1949년 8월 13일생
1972년 연세대학교 전기공학과 졸업
1980년 연세대학교 대학원 전자공학과 졸업(공학박사)
1977년 9월부터 현재 경북대학교 컴퓨터공학과 교수
주관심분야 디지탈시스템설계 및 테스트,
디지탈 논리 시뮬레이션,
컴퓨터 성능평가, 결함허용 시스템 설계