Petri-net을 이용한 전송 부하 분산 알고리즘에 관한 실험적 비교 연구

이태경, 김재민
동국대학교 컴퓨터 공학부
e-mail:{ tklee, ,its1304 }@dongguk.ac.kr

A Study on Experimental Comparison of Load Balancing Algorithms using Petri-net

Tae Kyung Lee, Jae Min Kim
Dept of Computer Engineering, Dongguk University

요 약

본 논문에서는 유한한 자원인 네트워크를 보다 많은 이용자가 원활하게 사용하기 위해 고안된 전송 부하 분산 알고리즘인 Round Robin, Weighted Round Robin과 전송 부하 분산 알고리즘이 존재 하지 않는 네트워크를 Petri-net 이론을 기반으로 한 시뮬레이터(CPN-Tools)를 통해 실험 결과를 비교 분석하여 알고리즘의 필요성과 전송 부하 분산 알고리즘의 특징과 장단점을 실험을 통하여 보였다.

1. 서론

최근, 다양한 네트워크를 이용한 디바이스의 공급으로, 네트워크에 대한 수요가 폭발적으로 늘어났지만, 네트워크의 공급이 수요를 충족시키지 못하면서, 이용자가 많은 지역에서 원활하게 네트워크를 이용하지 못하는 현상이 대두되고 있다.

이러한 현상을 해결하기 위해서는 네트워크의 효율성을 극대화를 시켜야 하는데, 그 방법으로는 공평한 분배를 통해서 이용자에게 분배해야 한다. 그러나 단순 분배를 한다면 이용자가 한 곳으로 집중이 되면서 이용이 원활하게 이루어 질 수 없는 경우가 발생한다.

결국, 유한한 자원을 이용자에게 공평하게 분배를 하면서, 분산을 통한 효율성 극대화가 필요하다.

이러한 효율성 극대화를 위한 부하 분산은 하드웨어에 의해 적용하는 방법, 소프트웨어에 의해 적용하는 방법이 있는데, 본 논문에서는 부하 분산 알고리즘을 이용한 소프트웨어에 의해 적용되는 방법을 이용하고자 한다.

현재까지 개발된 대표적인 부하분산 알고리즘은 Hashing, Round-robin, Weighted round-robin, Least connection, Weighted least-connection 이 있다.[4]

이중에서 Round-robin, Weighted round-robin 알고리즘을 구현하여 데이터의 부하분산도와 Queue에서의 Overflow 비율을 분석으로 각 알고리즘의 특성과 장·단점에 대하여 분석 하였다.

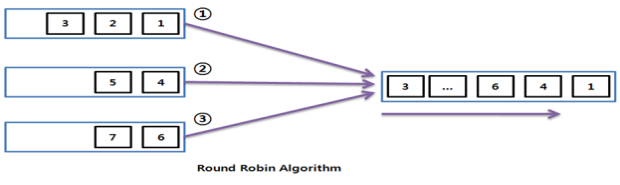
본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 설명하고, 3장에서는 전체 및 각 알고리즘의 설계 및 구현에 대해 설명한다. 4장에서는 실측 자료들을 바탕으로 IP네트워크에서의 알고리즘의 특징과 결과를 분석하며, 5장에서는 결론 및 향후 과제에 대해 설명한다.

2. 관련 연구

2.1 전송부하 알고리즘

2.1.1 Round robin 방식

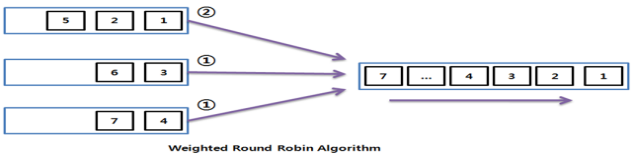
Queue내의 데이터를 순차적으로 선택하여, 처리 능력이나 부하와 상관없이 한 개의 데이터를 순차적으로 선택한다. 처리 능력이 비슷하거나 같을 경우에 효율적이며, 그렇지 않을 경우에는 오히려 효율이 떨어지는 알고리즘이다.[3]



(그림 2) Round robin의 실행

2.1.2 Weighted Round robin 방식

Round robin의 발전된 모습이며 서로 다른 처리능력의 가중치를 추가한 알고리즘이며, 추가된 가중치의 개수 만큼 데이터를 선택하며, 처리 능력이 서로 상이한 경우에 보다 효율적이나, 부하가 매우 다양하면 부하 불균형을 초래 한다.[4]



(그림 3) Weighted round robin의 실행

2.2 Petri-net

Petri-net은 1962년에 처음 소개되면서 시스템의 성능 테스트와 통신 규약의 일관성 및 타당성 테스트등의 다양한 분야에 성공적으로 사용되면서 분산소프트웨어시스템, 분산데이터베이스시스템, 이산적 사건시스템 등의 모델과 분석에 알맞은 도구로서 각광을 받고 있다. Petri-net 모형을 구축하기가 용이하고 구축된 모형을 분석하는 수학적 방법이 널리 연구 되었기 때문에, 관련된 연구가 활발히 진행되고 있다.[1]

Petri-net의 정의는 표 1과 같으며[Murata, 1989], 역동적 성질에 대해서 모의 실험을 가능케 해주는 변천 격발의 정의는 다음과 같다.[1][2][5]

(1) 하나의 변천 t에 대하여, 입력 장소 p가 최소한 W(p, t)만큼의 토큰을 갖고 있으면, t는 장전되었다고 한다.
(2) 장전된 t는 격발 될 수도, 아니 될 수도 있다.
(3) t의 격발은 W(p, t)만큼의 토큰을 각 입력 장소 p에서 제거 하고, W(t, p) 만큼의 토큰을 각 출력 장소에 더해 준다.

<표 1> 페트리 넷의 정의

페트리 넷 N은 5가지 요소로 구성 된다.

$N=(P, T, F, W, M_0)$, 단,

$P=\{p_1, p_2, \cdots, p_n\}$은 장소(place)라는 유한집합,

$T=\{t_1, t_2, \cdots, t_n\}$은 변천(transaction)이라는 유한집합,

F :유향간선이라는 $(P\times T)$와 $(T\times P)$의 합집합의 부분집합

$W: F\rightarrow\{1, 2, 3, \cdots\}$는 유향간선 가중치 결정 함수

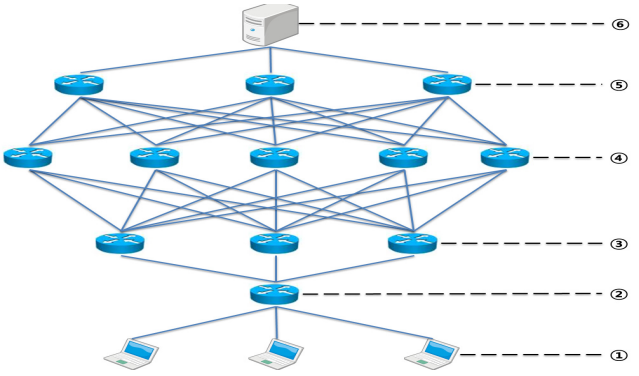
$M_0: P\rightarrow\{0, 1, 2, 3, \cdots\}$은 초기에 각 장소에 놓인 토큰의 수를 표현하며, 초기 마킹(marking)이라고 함.

단, P와 T의 교집합은 공집합이고, P와 T의 합집합은 공집합이 아님.

3. 가상 네트워크와 알고리즘 설계

3.1 가상 네트워크 구조

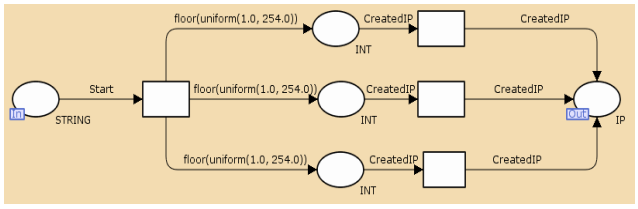
그림 4는 실험에 사용될 가상 네트워크 구조도를 나타내고 있다. ①출발지, ②출발지 게이트웨이, ③중간 라우터, ④중간 라우터 ⑤목적지 게이트웨이, ⑥목적지를 나타낸다.



(그림 4) 전체 가상 네트워크 구조도

3.2 이용자

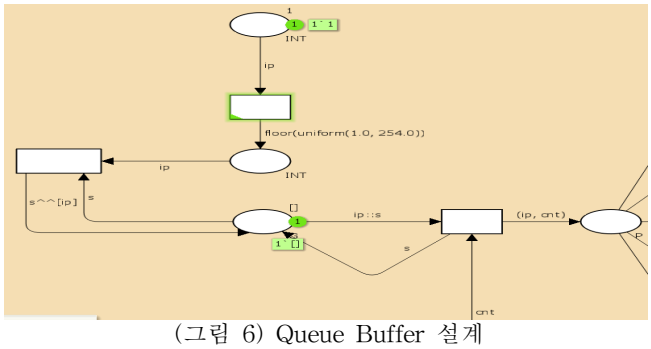
그림 5는 본 실험에서 적용될 가상의 이용자 생성에 대한 설계 모습이다. IP네트워크에서 IP데이터를 1부터 254까지 난수로 생성하며, 각 프로세스는 쓰레드로 실행 되며, 쓰레드 9개가 실험이 진행되는 동안 IP데이터를 지속적으로 생성한다.



(그림 5) 가상의 이용자 생성 설계

3.2 Queue Buffer

그림 6은 Queue Buffer의 설계한 모습이며, 한쪽 끝에서는 새로운 자료들이 삽입되고, 다른 반대편 끝에서는 가장 오래 기다렸던 자료들이 나오는 자료구조를 아래와 같이 설계하였다.

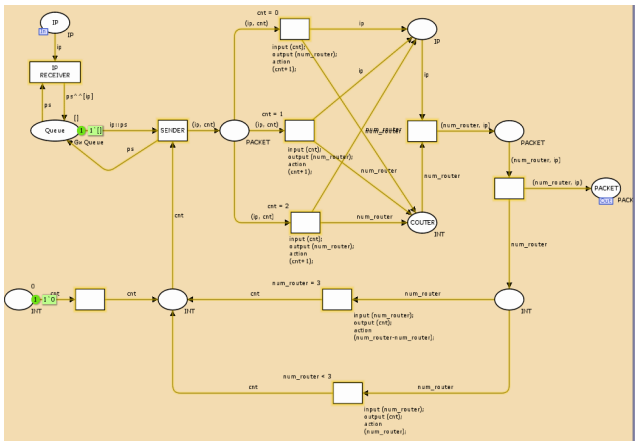


(그림 6) Queue Buffer 설계

3.4 알고리즘 설계와 구현

3.4.1 Round robin 방식

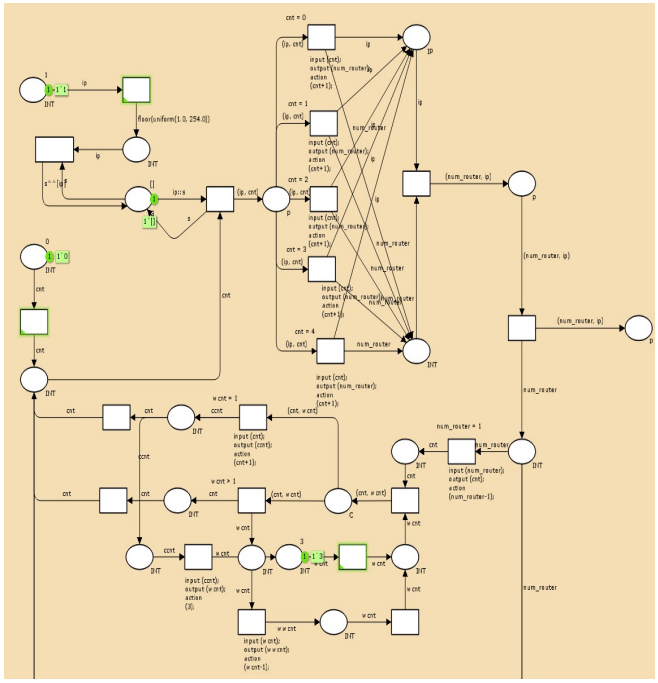
그림 7은 Round robin에 대한 설계 모습이며, IP데이터를 매개변수로 입력을 받아 Queue Buffer에 저장한 후, 순차적으로 한 개의 IP데이터를 각 라우터 장비에 전송 한다.



(그림 7) Round robin 알고리즘 설계

3.4.2 Weighted round robin 방식

그림 8은 본 실험에서 적용될 Weighted round robin에 대한 설계 모습이며, IP데이터를 매개변수로 입력을 받아 Queue Buffer에 저장한 후, 각 장비의 가중치의 개수 만큼 IP데이터를 선택하여 순차적으로 각 라우터 장비에 전송 한다.



(그림 8) Weighted round robin 알고리즘 설계

4. 성능 측정 및 분석

본 논문에서는 성능 측정을 하기 위해서 Petri-net 기반의 시뮬레이터(CPN-Tools)를 사용하였다. 부하분산 알고리즘이 존재하지 않는 IP네트워크와 부하분산이 존재하는 IP네트워크의 성능(IP데이터 분산, Overflow 비율)을 서로 비교 하였다.

4.1 Queue Buffer 크기

표 2는 원활한 실험을 위한 적절한 Queue Buffer의 크기를 알아 내기 위해 2000Step 동안 Queue Buffer크기에 따른 Overflow의 비율을 산출한 결과이다. 크기를 2부터 1씩 늘려가면서 Overflow의 횟수를 측정해 보았다.

표 2에서 볼 수 있듯이 Queue Buffer의 크기를 늘려준다면 Overflow의 횟수가 현저히 줄어드는 것을 알 수 있다. 이는 Queue Buffer의 사용가능한 공간이 늘어나면서 수용 가능한 IP데이터의 양이 늘어나기 때문인 것으로 분석이 된다.

그러나 Queue Buffer의 크기를 원하는 만큼 늘릴 수 없는 제한성이 존재하기 때문에, 실험을 통해서 최적의 효율성을 나타내는 크기를 알맞게 선정해야 한다.

표 2의 결과를 비교·분석해 보았을 때, Queue Buffer 크기를 늘려 가면 Overflow의 비율이 현저히 줄어들지만, Queue Buffer 크기에 비해서 Overflow의 비율이 비효율적인 구간이 존재하였다.

Queue Buffer 크기가 4개 일 때가 가장 효율적임을 알 수 있어, 크기는 4로 선정하였다.

<표 2> Queue Buffer 크기에 따른 Overflow 비교

Queue Buffer 크기	Overflow 비율	효율 비교
2	2.29%	·
3	1.27%	1.83
4	0.49%	2.59
5	0.47%	1.04
6	0%	·

4.2 IP데이터 분산

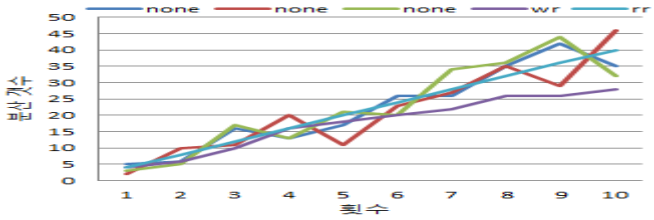
그림 4에서 ⑤에 해당하는 3개의 장비의 IP데이터 부하분산을 실험결과로서 확인해 보았다.

그림 9, 10, 11은 3개의 장비별로 이용자의 수를 늘려가면서 부하분산 알고리즘이 존재하지 않는 IP네트워크와 부하분산이 존재하는 IP네트워크의 IP데이터 분산을 서로 비교한 것을 그래프로 나타낸 결과이다.

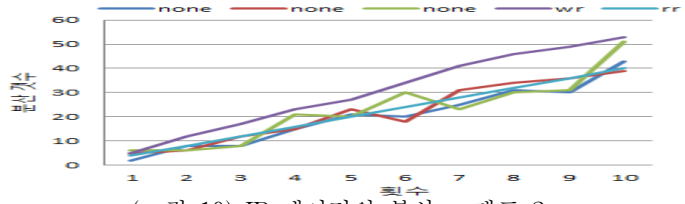
Round robin을 적용한 IP네트워크에서는 항상 같은 수의 IP데이터가 일정하게 증가(4, 8, 12, …)하면서, 부하 분

산 된다는 것을 그래프를 통해서 알 수 있었고, Weighted round robin을 적용한 IP네트워크에서는 가중치에 따라서 IP데이터가 분산이 되는데, 가중치의 값은 그래프1 < 그래프3 < 그래프2의 순으로 부여했으며, 그 값에 따라서 부하 분산이 된다는 것을 그래프를 통해서 알 수 있었다.

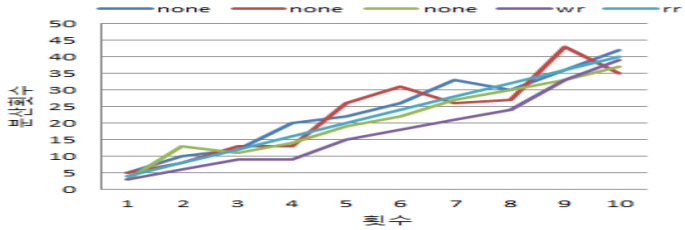
이에 반해서, 부하 분산 알고리즘이 존재 하지 않는 IP 네트워크에서는 그래프에서 볼 수 있듯이 IP데이터가 실험을 반복할 때 마다 다른 분산을 보였으며, 일정하지 못함을 알 수 있었다.



(그림 9) IP 데이터의 분산 그래프 1



(그림 10) IP 데이터의 분산 그래프 2



(그림 11) IP 데이터의 분산 그래프 3

4.3 Overflow 횟수

표 4는 2000Step 동안 부하분산 알고리즘이 존재하지 않는 IP네트워크와 부하분산 알고리즘이 존재하는 IP네트워크에서 발생한 Overflow 비율을 서로 비교한 결과이다.

표 4에서 Overflow 비율을 통해 부하분산 알고리즘이 존재하는 IP네트워크가 부하분산 알고리즘이 존재하지 않는 IP네트워크에 비해 효율적이라고 알 수 있다.

<표 4> 전송 부하 분산 알고리즘의 Overflow 비율

전송 부하 분산 알고리즘	Overflow 비율	비교
None	2.8%	·
Round robin	1.65%	1.69
Weighted round robin	0.95%	1.73

5. 결론 및 향후 과제

본 실험에서는 Petri-net 기반의 IP네트워크 환경에서의 전송 부하 분산 알고리즘의 필요성을 알아보기 위해서 Round robin, Weighted round robin의 알고리즘을 적용 후, IP 데이터의 분산, Overflow의 비율을 통해서 알고리즘의 특성과 장단점을 분석하였다.

그 결과 Queue Buffer의 크기가 커질수록, 보다 효율적인 전송부하 분산 알고리즘이 적용 될수록 낮은 Overflow 비율, 예측이 가능한 분산을 보인다는 것을 알 수 있었다.

이로 인해 유추할 수 있는 것은 IP데이터의 분산이 예상 가능한 범위 내에서 발생해, 향후 발생 할 수 있는 Overflow에 대해 보다 효율적인 대응이 가능해 질것으로 보인다.

Round robin 알고리즘은 모든 장비들이 동일한 성능을 보일 때, Weighted round robin 알고리즘은 서로 다른 성능을 보일 때, 효율적임을 실험 결과를 통해 알 수 있었다.

향후 장비들이 네트워크에 대한 정보를 정적으로 가지는 것이 아니라, 동적으로 장비 간 Agent들이 정보를 교환하면서 부하 분산을 하는 알고리즘에 대한 실험을 통해 네트워크 환경에서의 성능 분석을 통해 알고리즘의 특징과 장단점을 분석할 예정이다.

참고문헌

[1] 임재걸, 주재훈, 남윤석, 페트리넷을 적용한 위치기반 시스템의 타당성 분석, 정보기술과 데이터베이TM 저널 제12권 제1호, 2005

[2] 임재걸, 이계영, 김경징, 김규식, 페트리넷을 이용한 표준 발음법 분석 시스템 설계, 정보과학회 제26권 제1호 (B), 1999

[3] 문종배, 김명호, 분산 웹 서버 시스템에서의 DNS 기반 동적 부하 분산 기법, 한국정보과학회, 시스템 및 이론 제33권 제3호, 2006

[4] 최동준, 정광식, 손진곤, 웹 클러스터 시스템에서 개선된 WLC 스케줄링 알고리즘, 한국컴퓨터종합학술대회 제33권 제1호, 2006

[5] Documentation, http://cpntools.org/

[6] http://en.wikipedia.org/wiki/Weighted_round_robin