

# 가중치 원형 분배 기반 이진 스케줄링 바퀴구조의 성능 분석

(Performance Analysis of Binary Scheduling Wheel Structure  
based on Weighted Round Robin)

조해성<sup>†</sup> 이상태<sup>\*\*</sup> 전병실<sup>\*\*\*</sup>

(Hae-Seong Cho) (Sang-Tae Lee) (Byoung-Sil Chon)

**요약** 프레임 기반 스케줄링의 일종인 라운드 로빈 스케줄링은 네트워크에서 다양한 대역이나 지연과 같은 차별화된 서비스들을 제공할 수 있도록 각 큐에 다른 가중치를 지정함으로써 여러 큐들을 직접적으로 관리할 수 있는 매우 간단한 원리이다. 이러한 원리들 중 가장 대표적인 알고리즘이 WRR이다. 또한, WRR 알고리즘은 DBSW 구조에 의해 효율적으로 구현될 수 있다.

본 논문은 DBSW 구조의 수학적 분석을 수행하고 분석결과와 시뮬레이션 결과를 비교한다. 분석 데이터와 시뮬레이션 데이터는 DBSW 구조가 각 VC에 할당된 가중치를 정확하게 유지할 수 있기 때문에 DBSW 구조의 평균 퍼버 길이가 감소됨을 보여준다.

**Abstract** Round robin scheduling discipline, which is a sort of frame-based scheduling, is quite simple and straightforward for handling multiple queues, and by putting a different weight on each queue, a network can offer differentiated services such as different bandwidth, or delay bound. The most typical algorithm among this disciplines is the weighted round robin(WRR). Also, WRR algorithm can be implemented efficiently by dynamic binary scheduling wheel(DBSW) architecture.

This paper performs the analysis of the DBSW architecture and compares the results with simulation results. The analysis data and simulation data show that the DBSW structure decreases average buffer length because it is capable of maintaining the allocated weight of each VC correctly.

## 1. 서론

트래픽 흐름 제어를 위해서는 각 스위치의 출력 단에서 각 VC(Virtual Connection)에 공평한 대역을 할당할 수 있어야 한다. 이를 수행하기 위해서는 패킷 스케줄링 알고리즘이 필수적이다. 패킷 스케줄링의 성능을 결정하는 요소들은 트래픽의 통계적 특성, QoS 파라미터와 스케줄링 알고리즘이다. 이중 스케줄링 알고리즘은 패킷 스케줄링의 성능을 결정하는 가장 중요한 요소이다. 패킷 스케줄링 알고리즘이 고려해야 할 사항은 트래

픽 흐름의 분리, 패킷 지연, 자원 이용률, 공평성, 알고리즘의 간편성, 알고리즘의 확장용이성 등이다.

패킷 스케줄링은 패킷 교환기에 도착하여 전송을 기다리는 패킷의 서비스 순서를 결정하는 방법이다. 가장 보편적으로 사용되는 FCFS(First-Come-First-Service) 방법은 도착 순서대로 패킷을 공통 버퍼에 저장하였다가 도착 순서대로 패킷을 전송한다. 이 방법은 가장 단순한 방법으로 대부분의 인터넷 라우터는 FCFS 스케줄링 방법을 사용한다. 그러나, FCFS 방법은 각 사용자 플로우의 독립성을 유지하지 못하고 공평성을 보장하지 못한다. 스케줄링 방법을 이용하여 QoS를 지원하는 문제 외에 패킷 교환망에서 플로우 간의 간섭을 없애고 독립성을 유지하는 것도 매우 중요한 문제이다. 인터넷 전송 프로토콜인 TCP(Transmission Control Protocol)는 망에서 체증이 발생하면 각 송신자가 자발적으로 패킷의 생성량을 감소시켜 체증 현상을 제어한다. 만약 어느 사용자가 망 체증

<sup>†</sup> 비회원 : 건양대학교 IT센터 교수  
hscho@konyang.ac.kr

<sup>\*\*</sup> 비회원 : 한국표준과학연구원 인간정보그룹 그룹장  
stlee@kriss.re.kr

<sup>\*\*\*</sup> 종신회원 : 전북대학교 전자정보공학부 교수  
bschon@cslab.chonbuk.ac.kr

논문접수 : 2001년 3월 21일

심사완료 : 2001년 8월 27일

이 발생하여도 패킷을 많이 생성한다면 다른 사용자가 피해를 입게될 수 있다. 공평성을 보장하는 스케줄링 방식은 각 플로우가 다른 플로우에게 피해를 주지 않도록 제어하며 각 플로우에게 공평하게 서비스를 배분한다. 미래의 인터넷 및 패킷 교환망에서 사용될 수 있는 공평성과 QoS 지원 능력을 보유한 다양한 스케줄링 방식이 개발되었다[1]. 이들 중 WFQ(Weighted Fair Queueing)와 WRR(Weighted Round Robin)은 각 큐에 대해 대역을 보장할 수 있기 때문에 보편적인 스케줄링 알고리즘으로 발전 되어왔다[2]. WFQ는 다양한 길이의 패킷을 처리할 수 있고 비교적 적은 지연 제한을 가진다[3]. 그러나, WFQ는 각 연결에 대하여 VC의 개수에 비례하는  $O(V)$ 의 계산량을 필요하기 때문에 하드웨어 복잡도가 매우 크다. 이러한 계산 복잡도를 해결하기 위하여 제안된 알고리즘이 SCFQ(Self-Clocked Fair Queueing)이다[4]. 반면, WRR 기법은 ATM과 같은 고속 패킷 스위칭 네트워크에서 계산의 단순성과 저 비용 구현의 장점을 가지고 있어 셀 스케줄링 기법으로 널리 쓰인다. 이 기법은 특정 큐의 서버에 대한 접근 시간의 양을 제어함에 의해서 대역을 할당한다. 또한, WRR은 각 큐에 대역을 보장해 주기 때문에 스케줄링 기법으로 많이 사용되었다. WRR은 각 큐에 가중치를 할당하여 가중치에 따라 최소 대역을 보장한다[5,6]. WRR은 계산의 단순성과 적은 구현 비용 때문에 ATM 스위치에서 더 많이 사용된다.

WRR 스케줄러를 구현하기 위해 계산 복잡도와 하드웨어 요구사항을 현저히 경감시키는 BSW(Binary Scheduling Wheels) 기법이 제안되었다[7]. BSW 구조는 최소 하드웨어 비용으로 광범위한 전송률을 제공하고 과부하에는 최소 전송률을 보장할 수 있는 효율적인 패킷 스케줄링 구현 알고리즘이다. BSW 구조의 문제점을 개선하기 위해서 WRR 스케줄링 알고리즘을 충실히 수행할 수 있고 기능이 향상된 DBSW(Dynamain Binary Scheduling Wheel) 구조가 제안되었다[8]. DBSW 구조는 BSW 구조에서 발생하는 공평성과 큐의 효율적 사용 문제들을 개선하였다.

본 논문에서는 기존에 제안된 DBSW 구조의 수학적 분석을 수행하고 수학적 분석의 결과와 시뮬레이션 결과를 비교하여 DBSW 구조의 성능을 평가하고자 한다. DBSW 구조의 수학적 분석은 먼저 각 VC의 큐를 모델링 한 다음 각 큐의 모델 수식에 근거하여 DBSW 구조의 셀 스케줄링 알고리즘에 따른 큐 상태 변화를 관찰하는 과정으로 수행하였다. DBSW 구조의 성능은 수학적 분석에 의해 큐 상태 확률을 구하고 이에 근거하여 평균 큐 길이를 구하여 성능을 평가한다. 본 논문

의 구성은 2장에서는 DBSW 구조에 대하여 살펴보고 3장에서는 DBSW 구조의 수학적 분석 내용을 서술하였다. 4장에서는 DBSW 구조의 수학적 분석의 결과와 시뮬레이션 결과를 비교 평가하였고 5장 결론에서는 본 논문의 내용과 결과의 내용을 분석하고 향후 연구방향을 제시하였다.

## 2. DBSW 구조

BSW는 최소 하드웨어 비용으로 광범위한 전송률을 제공하고 과부하에는 최소 전송률을 보장할 수 있다. 이 알고리즘에서는 링크 대역을 2의 지수승으로 나눈 값을 각 VC의 전송율로 규정하고 최소 전송율을  $1/2^N$ 으로 제한하였다. 따라서, VC  $i$ 의 전송율은  $w_i / 2^N$ 으로 계산되며  $w_i$ 는 가중치로서 2의 지수승이다. 그러므로 모든 VC의 전송율은 출력 링크의 2의 지수승으로 나눈 값들 중의 하나이고 최소 전송율의 2의 지수승이다. 그러나, 기존의 BSW 스케줄링 알고리즘은 각 VC의 전송율을 링크 용량의 2의 음의 지수승 형태로 변환하기 때문에 실제의 가중치보다 더 많은 가중치를 얻게 된다. 또한, 어느 한 VC의 큐가 비어있게 되면 셀 서비스를 수행하지 못하므로 각 VC는 가중치, 셀 지연, 버퍼 사용 등에서 심각한 성능저하를 유발한다.

BSW 스케줄링 알고리즘의 문제점을 개선하기 위해 DBSW 구조가 제안되었다. 기존의 BSW 구조의 스케줄링 알고리즘은 이진으로 서비스를 수행하기 때문에 가중치 보다 많은 서비스를 받는 경우가 발생하여 공평한 서비스가 이루어지지 않고 각 VC의 큐 상황에 관계없이 서비스를 수행하기 때문에 서비스율이 떨어지는 단점을 DBSW 구조 알고리즘은 개선하고 있다[8].

그림 5는 DBSW 구조의 이진 바퀴를 보여 주고 있다. DBSW 구조는 기존 BSW 구조에서 각 VC에 최소 대역의 2의 지수승만을 제공하는 문제점을 개선하기 위하여 각 VC에 임의 대역을 할당할 수 있도록 다음과 같은 가중치 값과 가중치 목록을 작성하였다. VC  $i$ 의 가중치 계산은 식 (1)과 같다.

$$W_i = \lfloor r_i * W/C \rfloor$$

$$W = W_1 + W_2 + \dots + W_N \quad (1)$$

식 (1)에서  $C$ 는 출력 링크의 용량이고  $W$ 는 모든 가중치 값들의 합이고  $r_i$ 는 VC  $i$ 에 할당된 전송율이다. DBSW 구조는 각 VC는 스케줄링 서버가 제공하는 가중치 값에 근거하여 가중치 목록을 식 (2)에 따라 구성한다.

$$VC_k(W) = \{1, 4, \dots, 2^k\}, \text{ if } W_k = 1 + 4 + \dots + 2^k \quad (2)$$

식 (2)에서  $VC_k(W)$ 는 VC  $k$ 의 가중치 전송목록이고

Wk는 VCk의 가중치 값을 표시한다. 식 (2)에서 스케줄링 서버가 2의 지수승의 가중치만을 제공할 때 제공되는 가중치 값을 조합하여 임의 크기의 가중치 값을 구성할 수 있기 때문에 임의 크기의 가중치를 가진 VC에 서비스가 가능하다. 아래 그림 1은 위에서 구성된 각 VC의 가중치 목록을 근거로 하여 각 가중치 값에 해당하는 제안된 이진 스케줄링 바퀴를 보여주고 있다. 여기서 BSW 구조의 이진 바퀴에서는 한 VC는 하나의 바퀴에만 포함될 수 있는 반면 DBSW 구조에서는 한 VC가 여러 바퀴에 포함될 수 있음을 그림 1에서 보여주고 있다.

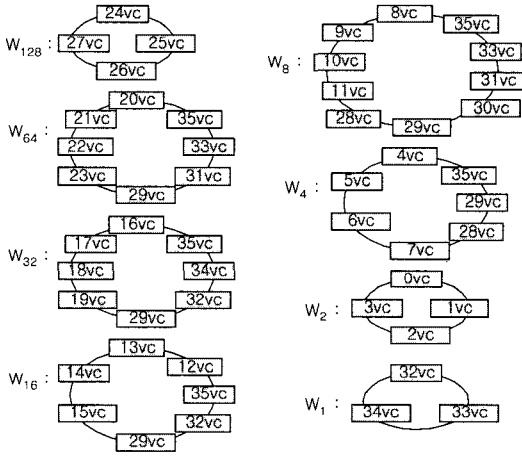


그림 1 제안된 이진 스케줄링 바퀴

그림 2는 DBSW 구조의 이진 전송을 목록을 보여주고 있다. DBSW 구조의 이진 전송을 목록은 그림 1에서 보여준 이진 바퀴에 속한 VC들의 가중치 합에 의거하여 이진 바퀴가 연결 될 슬롯을 결정하여 이진 전송을 목록을 구성한다.

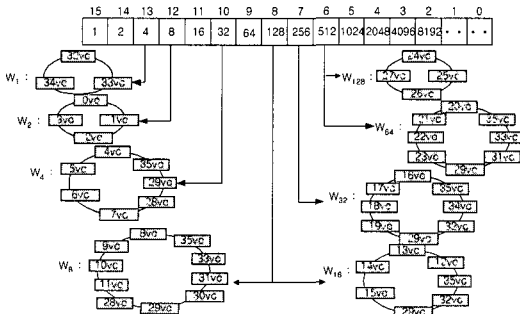


그림 2 제안된 구조의 이진 전송을 목록

DBSW 구조의 셀 스케줄링 알고리즘에서는 각 VC에 W, lend, borrow의 추가적인 변수를 사용하여 가중치, 서비스를 빌려준 횟수, 서비스를 빌린 횟수를 기록하여 서비스를 공평하게 수행할 수 있도록 한다. 서비스는 이진 전송을 목록의 슬롯에 있는 각 바퀴의 큐를 순차적으로 선택하여 서비스를 수행하는데, 선택된 VC의 borrow값이 0보다 크면 borrow값을 1 감소시키고 현재 슬롯에 있는 VC중에서 lend 값이 제일 큰 VC를 찾고 이 VC에 셀이 있으면 한 셀을 서비스하고 lend 값을 1 감소시킨다. 만약 lend 값이 가장 큰 VC의 큐가 비어 있으면 다음으로 lend 값이 큰 VC로 이동하여 서비스를 수행한다. 그러나 슬롯에 있는 모든 VC의 lend값이 0이면 선택된 VC의 한 셀을 서비스하고 W값을 1 감소시킨다. 이는 이전에 서비스를 빌려 받거나 빌려준 VC가 존재할 경우 서비스를 공평하게 수행하기 위함이다. 선택된 VC의 borrow값이 0인 경우 선택된 VC에 셀이 존재할 경우 서비스를 수행하고 가중치를 1 감소시킨다. 그런데 선택된 VC의 가중치가 0일 경우에는 슬롯 안에 있는 VC들 중 현재 가중치가 가장 큰 VC의 한 셀을 서비스하여 할당된 가중치 값만을 서비스하고 그 이외에는 다른 VC를 서비스 할 수 있도록 함으로서 각 VC를 가중치 값에 비례하여 공평하게 서비스를 수행한다. 또한, 선택된 VC가 서비스를 못 받을 경우에 다른 VC를 서비스하여 전체적인 서비스율을 개선할 수 있다. 각 서비스를 받은 VC의 W값을 1씩 감소하여 서비스 받을 횟수를 기록한다. 만약, 선택된 VC에 셀이 없을 경우 lend값을 1증가시키고 그 슬롯에 있는 VC들 중에서 큐의 길이가 가장 긴 VC를 찾아 서비스를 수행하고 borrow값을 1 만큼 증가시킨다. 이는 큐의 길이가 긴 VC를 먼저 서비스 해 줌으로서 전체적인 큐의 길이를 감소시킬 수 있다.

그림 3은 DBSW 구조의 셀 스케줄링 흐름도를 보여주고 있다. DBSW 구조의 셀 스케줄링의 특징은 서비스 순서가 된 VC의 가중치 계수가 값, 큐의 상태, 이전에 빌린 서비스의 개수 등을 고려하여 서비스를 수행하는 것이다. 만약 서비스 순서가 돌아와 선택된 VC의 큐에 셀이 존재하지 않으면 서비스를 수행 할 수 없기 때문에 선택된 VC가 속한 VC들 중에 이전에 서비스를 제일 많이 빌려준 VC에 우선권을 주어 서비스를 수행할 수 있도록 한다. 또한, 서비스 받을 VC의 가중치 값이 0일 경우에는 서비스 받을 VC가 속한 이진 바퀴에서 가중치 값이 제일 큰 VC에게 서비스를 받을 수 있도록 한다.

DBSW 구조의 셀 스케줄링 알고리즘은 기존의 셀 스

케줄링 알고리즘보다 최대 큐 길이 및 큐잉 지연 측면에서 향상된 성능을 보일 것이다. 이는 기존의 셀 스케줄링 알고리즘은 큐의 상태와 관계없이 서비스를 수행하므로 최대 큐잉 길이가 커지고 또한, 적절하게 서비스를 받지 못하므로 큰 큐잉 지연이 발생하는데 비해 제안된 알고리즘은 이러한 문제점을 개선하였기 때문이다.

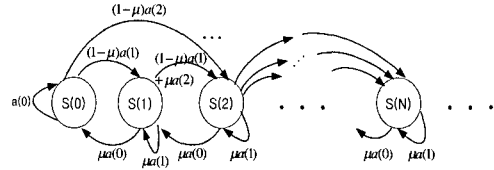


그림 5 VC 큐의 상태 천이도

그림 5는 그림 4의 VC큐 모델에 대한 큐의 상태 천이를 보여 주고 있다. 그림 5에서 S(i)는 큐에 셀이 i개 있는 상태이다. 그리고 a(i)는 한 셀 서비스 시간 동안에 i개의 셀이 큐로 입력될 확률을 나타내고 있다. 셀 입력이 Poisson 분포로 입력된다고 가정하면 a(i)는 식 (3)과 같다.

$$a(i) = \frac{(\lambda)^i}{i!} e^{-\lambda} \quad \lambda : \text{input load} \quad (3)$$

그림 5의 상태 천이도를 근거로 한 각 상태에 대한 확률식은 식 (4)와 같다.

$$\begin{aligned} s(0) &= 1 - \mu \\ s(0) &= a(0)s(0) + \mu a(0)s(1) \\ s(1) &= \mu a(1)s(1) + (1 - \mu)a(1)s(0) \\ s(2) &= \mu a(1)s(2) + (1 - \mu)a(1)s(1) + (1 - \mu)a(2)s(0) \\ &\vdots \\ s(N) &= \mu a(1)s(N) + (1 - \mu)a(1)s(N-1) + \dots + (1 - \mu)a(N)s(0) \end{aligned} \quad (4)$$

각 VC의 서비스 확률은 주어지므로 수식 (4)에서 S(0)를 구할 수 있고 s(0)를 수식 (4)의 두 번째 식에 대입하여 s(1)을 구할 수 있다. 이러한 과정을 반복 수행하면 VC큐의 모든 상태에 대한 확률 s(1)부터 s(∞)을 구할 수 있다. 그러나 WRR 알고리즘의 경우 각 VC에 대한 서비스율이 일정한 확률을 가지지 않고 각 VC의 상태를 고려하여 서비스가 수행되기 때문에 이를 고려한 모델이 필요하다. 이를 위해서는 WRR 알고리즘의 서비스 스케줄링을 파악하여 각 서비스 시간동안에 서비스되는 가상 연결을 결정하고 이에 따른 WRR 알고리즘의 수학적 분석을 수행해야 할 것이다.

DBSW 구조 알고리즘의 수학적 분석을 수행하기 위하여 우선 WRR 서비스 구조의 각 VC의 상태 천이 행렬을 구하여 이를 토대로 수학적 분석을 수행하고자 한다. WRR 서비스 구조에서는 각 VC 큐의 서비스율이 일정하지 않기 때문에 매 서비스가 발생할 경우에 대한 상태를 가정하고 이 상태에 따른 수식을 작성하여 이를 분석함으로써 성능을 예측하고자 한다[70]. WRR의 서비스 수행은 각 VC의 가중치 상태에 따라 이루어지기 때문에, 이를 위해 먼저 각 VC의 상태를 큐의 상태와 가중치 상태에 따른 상태, S(q,w)를 정의한다. S(q,w)

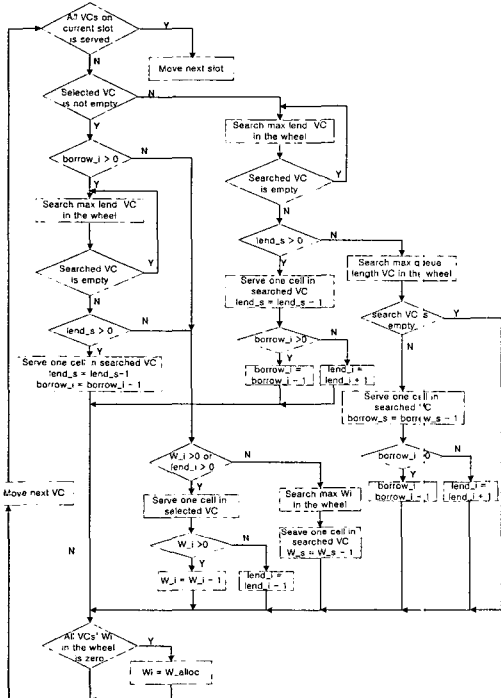


그림 3 제안된 BSW 구조의 셀 스케줄링 알고리즘 흐름도

### 3. DBSW 구조의 수학적 분석

#### 3.1 WRR 구조의 수학적 분석

ATM 스위치에서 다중화기에 사용되는 스케줄러는 각 VC에 큐를 할당하고 이 큐에 대하여 스케줄링을 수행한다. 그러므로 이를 해석하기 위해서는 먼저 큐에 대한 모델을 적용해야 한다. 아래 그림 4는 일반적인 큐의 모델을 보여주고 있다. 그림 4에서 λ는 입력의 평균 부하이고 μ는 큐에 대한 서비스율이다.

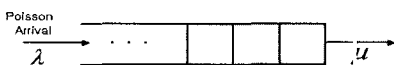


그림 4 큐 모델

는 VC의 큐에  $q$ 개의 셀이 존재하고 가중치 계수기의 값이  $w$ 인 상태를 나타낸다. VC <sub>$i$</sub> 에 대한 상태 천이 확률 행렬을  $G_i^T$ 라 정의하면  $G_{i(q,w) \rightarrow (q',w')}^T$ 는 VC <sub>$i$</sub> 의 상태가  $(q,w)$ 에서  $(q',w')$ 로 천이 될 확률을 나타낸다. 또한, 각 VC에 대한 서비스 수행을 나타내는  $s_i(t)$ 를 정의하고자 한다.  $s_i(t)$ 는 VC  $i$ 가 서비스 시간  $t$ 에 서비스를 받는다면 1로 세트하고 그렇지 않으면 0으로 세트된다. WRR의 경우 각 VC에 할당된 가중치  $W_i$  값에 따라 서비스가 수행되므로  $S_i(t)$ 를 다음과 같이 정의할 수 있을 것이다.

$$s_i(t) = \begin{cases} 1, & t \bmod (W_{tot} + \lfloor W_{tot}/W_i \rfloor) = 0 \\ 0, & \text{Otherwise} \end{cases} \quad (5)$$

수식 (5)에서  $W_{tot}$ 는 모든 VC의 가중치 값을 다 더한 값이고  $\lfloor x \rfloor$ 는  $x$ 보다 큰 최소 정수를 나타낸다. 이를 기반으로 버퍼의 크기가 무한한 WRR 모델에 대한 상태 천이 확률  $G_{i(q,w) \rightarrow (q',w')}^T$ 를 구하면 다음과 같다.

$$G_{i(q,w) \rightarrow (q',w')}^T = \begin{cases} a_i(q' - q + 1) & \text{if } s_i(t) = 1, w' = w - 1 \geq 0, q' \geq q > 0 \\ a_i(q' - q) & \text{if } s_i(t) = 1, w' = w - 1 \geq 0, q' \geq q = 0 \\ a_i(q' - q) & \text{if } s_i(t) = 0, w' = w, q' \geq q \geq 0 \\ & \text{or } s_i(t) = 1, w' = w = 0, q' \geq q \geq 0 \\ 0 & \text{Otherwise} \end{cases} \quad (6)$$

수식 (6)에서  $a_i(q)$ 는 서비스 타임 슬롯 동안에 VC  $i$ 의 큐로  $q$ 개의 셀이 들어올 확률을 나타낸다. 그리고  $G_{i(q,w) \rightarrow (q',w')}^T$ 는 서비스 시간  $t$ 에 VC  $i$  큐의 상태가  $(q,w)$ 에서 다음 서비스 시간  $t+1$ 에  $(q',w')$ 상태로 천이 될 확률을 나타낸다. 위 식에서 각 VC의 상태 천이는 서비스 수행 상태 즉,  $s_i(t)$ 에 따라 다른 확률로 천이 됨을 보여주고 있다. 큐 상태 천이 확률 행렬을 구하기 위하여 큐 상태 천이 확률 행렬  $A_i^t$ 를 정의한다. 또한, VC  $i$ 의 상태 확률 행렬  $S_i^t$ 를 정의한다.  $S_i^t$  행렬은 VC  $i$ 의 큐 상태와 가중치 상태에 따른 각 확률 값을 원소로 한다. 또한, 서비스가 수행됨에 따라 각 VC의 가중치 값이 변하므로 가중치 변환 행렬을 필요로 한다. VC  $i$ 에 대하여 서비스 시간  $t$ 에서 가중치 변환 행렬은  $K_i^t$ 로 정의한다.

$$K_i^t = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix} \quad (7)$$

수식 (7)에서  $K_i^t$ 의 차원은 VC  $i$ 에 할당된 가중치 값  $W_i$ 이다. 그러므로  $K_i^t$ 는 가중치 값이 0인 것을 포함하기 때문에  $(W_i+1) \times (W_i + 1)$  행렬이다.

$A_i^t$ 는 서비스 시간  $t$ 에 VC  $i$ 큐의 상태가 천이 될 확률들을 나타낸다. 상태 천이 확률 행렬은  $s_i(t)$ 의 상태에

다른 큐의 상태 천이 확률 행렬  $A_i^t$ 를 식 (6)으로부터 구할 수 있다.

$$s_i(t) = 0$$

$$A_i^t = \begin{bmatrix} a_i(0) & 0 & 0 & \dots & 0 & \dots \\ a_i(1) & a_i(0) & 0 & \dots & 0 & \dots \\ a_i(2) & a_i(1) & a_i(0) & \dots & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_i(N) & a_i(N-1) & a_i(N-2) & \dots & a_i(0) & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (8)$$

$$s_i(t) = 1$$

$$A_i^t = \begin{bmatrix} a_i(1) + a_i(0) & a_i(0) & 0 & \dots & 0 & \dots \\ a_i(1) & a_i(1) & a_i(0) & \dots & 0 & \dots \\ a_i(2) & a_i(2) & a_i(1) & \dots & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_i(N) & a_i(N) & a_i(N-1) & \dots & a_i(1) & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (9)$$

수식 (8)과 (9)에서 각 행렬의 원소는 큐의 상태가 행 값의 상태에서 열 값의 상태로 천이 할 확률을 각 서비스 값  $s_i(t)$ 에 따라 보여주고 있다. 예로 수식 (8)은 VC  $i$ 큐의 서비스 상태가 0으로서 서비스를 받지 못할 경우 큐의 상태 0에서 다음 상태 0로 천이 할 확률은 셀 입력이 0인 확률 즉,  $a_i(0)$ 임을 알 수 있다. 그리고  $A_i^t$ 의 차원은 큐의 길이가 되는데 일반적으로 큐의 길이를 무한히 크게 하기 때문에  $\infty$ 로 가정한다. 그러나, 행렬 계산을 위해서는 행렬의 차원을 유한히 해야 하고 이를 위해서는 큐의 길이를 제한해야만 한다.

위에서 살펴본 바와 같이 서비스 시간  $t$ 에서의 VC  $i$ 의 상태 확률 행렬  $S_i^t$ 는 이전의 상태 확률 행렬  $S_i^{t-1}$ , 큐 상태 천이 확률 행렬  $A_i^t$ , 가중치 변환 행렬  $K_i^t$ 의 곱에 의해서 구할 수 있고  $S_i^{t-1}$ 은 그 이전의 상태로 구할 수 있으므로 이를 반복적으로 수행하면 다음 수식 (10)과 같이 구할 수 있다.

$$S_i^t = A_i^t S_i^{t-1} K_i^t = A_i^t A_i^{t-1} S_i^{t-2} K_i^{t-1} K_i^t = \dots = A_i^t A_i^{t-1} \dots A_i^0 S_i^0 K_i^t K_i^{t-1} \dots K_i^1 \quad (10)$$

수식 (10)에서 서비스 시간  $t$ 에서 VC  $i$ 의 상태 확률 행렬은 초기 상태 확률 행렬  $S_i^0$ 와 각 서비스 시간별 큐 상태 천이 확률 행렬, 가중치 천이 행렬의 곱에 의해서 구할 수 있음을 알 수 있다.

그러나 WRR 알고리즘의 경우 모든 VC들의 가중치 만큼 서비스를 수행하면 가중치 상태에 관계없이 할당된 가중치로 리셋을 수행하는데 이를 고려해야 한다.

$$K_i^t = \begin{bmatrix} 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \quad (11)$$

$t = \text{Weight Count Reset}$

가중치 리세트는 큐의 상태와는 별개로 모든 가중치 상태가 할당된 가중치로 치환되므로 다음과 같이 가중치 변환 행렬을 구할 수 있다. 식 (11)에서 최대 가중치 상태의 확률은 모든 가중치 상태의 합이 됨을 보여 주고 있다. 이는 가중치를 리세트 하면 반드시 가중치 상태는 할당된 가중치 값 즉, 최대의 가중치 값이 되므로 최대의 가중치 상태 확률은 모든 가중치 상태 확률을 합해서 얻을 수 있다.

위 수식 (7), (8), (9), (10), (11)을 이용하여 일정한 서비스 시간 후 안정한 상태에서 각 VC의 상태 확률을 구할 수 있고 이를 이용하여 평균 큐의 길이 및 평균 지연을 구할 수 있다.

3.2 DBSW 구조 알고리즘의 수학적 분석

DBSW 구조는 각 VC의 전송률에 따라 전송 바퀴를 구성하고 이 바퀴의 총 전송률에 따라 이진 전송률 목록이 구성된다. 각 VC에 대한 서비스 수행은 구성된 전송률 목록에 따라 이루어지기 때문에 항상 일정한 순서대로 서비스가 이루어지지 않고, DBSW 구조의 스케줄링 알고리즘에 의해서 각 VC의 서비스 수행이 결정된다. 그러므로 서비스 수행을 알려주는 변수  $s_i(t)$ 를 식 (12)와 같이 정의 할 수 있다.

$$s_i(t) = \begin{cases} 1, & \text{VC } i \text{ is served at } t \text{ time slot} \\ 0, & \text{Otherwise} \end{cases} \quad (12)$$

DBSW 구조의 서비스 알고리즘에서  $s_i(t)$ 는 VC가 바뀌지 않는 한 일정한 순서로 반복한다. 그러나, DBSW 구조의 알고리즘의  $s_i(t)$ 는 다음과 같이 좀더 구체적으로 정의 할 수 있다.

$$s_i(t) = \begin{cases} 1, & \text{VC } i \text{ is selected by the scheduling algorithm} \\ & \text{at } t \text{ time slot} \\ 0, & \text{Otherwise} \end{cases} \quad (13)$$

식 (13)은 DBSW 구조의 셀 스케줄링 알고리즘에서 어느 한 셀이 서비스되기 전에 먼저 서비스 받는 VC  $i$ 의  $s_i(t)$ 를 1로 세트하여 서비스 받게 됨을 알려준다. 그러나 DBSW 구조의 셀 스케줄링 알고리즘은 가중치 계수기 값에 관계없이 항상 일정한 순서에 의해서 각 VC들의 셀을 서비스한다. 그러므로 DBSW 구조에서는 상태변수가 가중치 값과는 관계가 없으므로 상태변수를 큐의 상태만으로 정의할 수 있다. 그래서 DBSW 구조에서는 상태 변수를 큐의 상태에 따라  $S_i(q)$ 로 정의한다.  $S_i(q)$ 는 VC  $i$ 의 큐에  $q$ 개의 셀이 존재하는 상태를 나타낸다.

식 (13)을 이용하여 기존의 DBSW 구조의 스케줄링 알고리즘에서의 상태 천이 확률을 이용하여 큐 상태 천이 확률 행렬을 계산할 수 있다. 이는 WRR 스케줄링 알고리즘과 비슷한 형태로 구성되나 상태 확률 및 상태

천이 확률을 구하여 성능을 평가한다.

상태 확률 행렬과 상태 천이 확률 행렬을 구하고자 할 경우 행렬 곱을 수행해야 하는데 행렬의 차원이 무한대일 경우 행렬 곱셈을 수행하기가 불가능하므로 행렬의 차원을 줄여야만 한다. 이를 위하여 상태천이 행렬의 차원을 제한해야 하는데, 이는 각 VC 큐의 길이를 제한하는 것과 같다. 본 장에서는 큐의 길이 즉, 행렬의 차원을  $N$ 으로 제한하였다. 큐의 길이를  $N$ 으로 제한하였을 경우 상태 확률 값에 변동이 발생한다. 이는 큐의 길이를 제한하였기 때문에 큐 길이 이상의 큐 상태는 발생하지 않기 때문이다. 그러므로 큐의 길이를 제한하였을 경우의 상태 천이 확률  $G_{i(q) \rightarrow (q')}^T$ 은 아래 수식과 같다.

$$G_{i(q) \rightarrow (q')}^T = \begin{cases} a_i(q' - q + 1) & \text{if } s_i(t) = 1, N > q' \geq q - 1 \geq 0 \\ \sum_{k=N}^{q'} a_i(k - q + 1) & \text{if } s_i(t) = 1, q' = N, q > 0 \\ a_i(q' - q) & \text{if } s_i(t) = 1, N > q' \geq q = 0 \\ a_i(q' - q) & \text{if } s_i(t) = 0, N > q' \geq q \geq 0 \\ \sum_{k=N}^{q'} a_i(k - q) & \text{if } s_i(t) = 0, N = q' \geq q \geq 0 \\ 0 & \text{Otherwise} \end{cases} \quad (14)$$

식 (14)에서 DBSW 구조의 알고리즘은 큐에 셀이 없는 경우에는  $s_i(t)$ 가 1인 경우에도 셀 전송 서비스를 수행하지 않는다. 식 (14)의 상태 천이 확률 식을 근거로 큐의 길이가  $N$ 인 경우의 상태 천이도를 작성하면 아래 그림 6과 같다. 그림 6에서 큐가 비어 있는 경우 서비스 상태에 관계없이 서비스가 수행되지 못하기 때문에 입력되는 셀의 개수에 따라서 상태가 변화함을 확인할 수 있다.

그림 6을 참조하여 큐의 길이가  $N$ 으로 제한된 경우의 큐의 상태 천이 확률 행렬  $A_i^T$ 를 구하면 아래 식 (15), (16)과 같다.

$$A_i^T = \begin{matrix} s_i(t) = 0 \\ \begin{bmatrix} a_i(0) & 0 & 0 & \dots & 0 \\ a_i(1) & a_i(0) & 0 & \dots & 0 \\ a_i(2) & a_i(1) & a_i(0) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_i(N-1) & a_i(N-2) & a_i(N-3) & \dots & 0 \\ \sum_{k=N}^{q'} a_i(k) & \sum_{k=N-1}^{q'} a_i(k) & \sum_{k=N-2}^{q'} a_i(k) & \dots & \sum_{k=0}^{q'} a_i(k) \end{bmatrix} \end{matrix} \quad (15)$$

$$A_i^T = \begin{matrix} s_i(t) = 1 \\ \begin{bmatrix} a_i(0) & 0 & 0 & \dots & 0 \\ a_i(1) & a_i(1) & a_i(0) & \dots & 0 \\ a_i(2) & a_i(2) & a_i(1) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_i(N-1) & a_i(N-1) & a_i(N-2) & \dots & a_i(0) \\ \sum_{k=N}^{q'} a_i(k) & \sum_{k=N}^{q'} a_i(k) & \sum_{k=N-1}^{q'} a_i(k) & \dots & \sum_{k=0}^{q'} a_i(k) \end{bmatrix} \end{matrix} \quad (16)$$

위식 (15), (16)의 큐 상태 천이 확률 행렬을 이용하여 각 VC의 상태 확률 행렬을 구하는 과정은 식 (17)과 같이 표현할 수 있다.

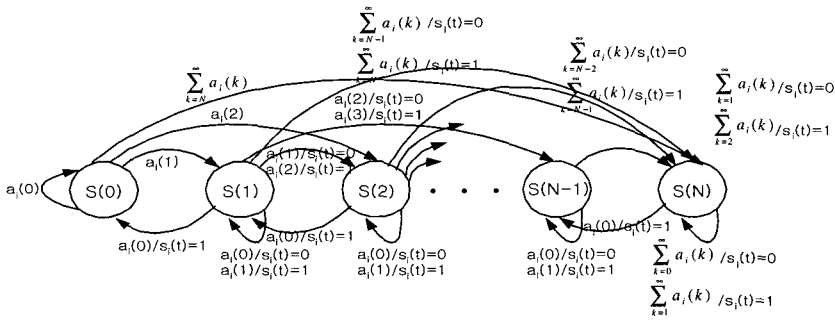


그림 6 DBSW 구조의 큐 상태 천이도

$$S_i^t = A_i^t S_i^{t-1} = A_i^t A_i^{t-1} S_i^{t-2} = \dots = A_i^t A_i^{t-1} \dots A_i^1 S_i^0 \quad (17)$$

위에서 정리한 수식을 이용하여 DBSW 구조의 서비스 알고리즘의 상태를 정의하고 이에 따른 상태 확률 행렬과 큐 상태 천이 확률 행렬을 구하는 과정을 정리하면 다음과 같다.

**BSW 구조의 상태 확률 행렬 계산 과정**

- step i) 이진 바퀴 구성
- step ii) 이진 전송 목록 구성
- step iii) 이진 전송 목록에 따른 서비스 VC 결정
- step iv) 수식 (15), (16), (17)에 따라  $S_i^t, A_i^t$  계산
- step v) step iii)와 step iv) 반복

위의 상태 확률 행렬 계산 과정을 이용하여 큐의 길이  $N = 9$ 이고  $t = 1000$ 일 때의 각 VC의 상태 천이 확률 행렬을 입력 부하에 따라 식 (18), (19) 행렬식에서 보여주고 있다.

Load = 50%, Rate = 64Kbps

$$A_i^t = \begin{bmatrix} 0.758 & 0.641 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.209 & 0.300 & 0.641 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.030 & 0.073 & 0.300 & 0.641 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.003 & 0.012 & 0.073 & 0.300 & 0.641 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.001 & 0.012 & 0.073 & 0.300 & 0.641 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.001 & 0.012 & 0.073 & 0.300 & 0.641 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.001 & 0.012 & 0.073 & 0.300 & 0.641 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.001 & 0.012 & 0.073 & 0.300 & 0.641 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.001 & 0.012 & 0.075 & 0.300 & 0.776 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.002 & 0.012 & 0.068 & 0.224 \end{bmatrix} \quad (18)$$

Load = 90%, Rate = 1Mbps

$$A_i^t = \begin{bmatrix} 0.271 & 0.262 & 0.234 & 0.188 & 0.133 & 0.079 & 0.038 & 0.014 & 0.003 & 0.001 \\ 0.280 & 0.273 & 0.252 & 0.216 & 0.169 & 0.116 & 0.069 & 0.034 & 0.013 & 0.005 \\ 0.196 & 0.194 & 0.191 & 0.181 & 0.164 & 0.137 & 0.101 & 0.064 & 0.035 & 0.021 \\ 0.119 & 0.122 & 0.130 & 0.140 & 0.148 & 0.146 & 0.129 & 0.101 & 0.070 & 0.052 \\ 0.067 & 0.071 & 0.083 & 0.103 & 0.125 & 0.141 & 0.145 & 0.134 & 0.114 & 0.099 \\ 0.035 & 0.039 & 0.051 & 0.070 & 0.096 & 0.123 & 0.145 & 0.155 & 0.154 & 0.150 \\ 0.017 & 0.020 & 0.029 & 0.045 & 0.069 & 0.099 & 0.130 & 0.159 & 0.179 & 0.188 \\ 0.008 & 0.010 & 0.016 & 0.028 & 0.047 & 0.073 & 0.107 & 0.144 & 0.176 & 0.194 \\ 0.004 & 0.005 & 0.010 & 0.018 & 0.033 & 0.056 & 0.088 & 0.127 & 0.165 & 0.187 \\ 0.002 & 0.002 & 0.004 & 0.009 & 0.017 & 0.029 & 0.047 & 0.069 & 0.091 & 0.104 \end{bmatrix} \quad (19)$$

위의 상태 천이확률 행렬을 살펴보면 입력 부하가 적고 전송률이 적을 때는 큐에 셀이 적게 있을 확률이 크므로 상태 천이확률 행렬의 위 부분의 숫자는 크나 아래 부분에는 거의 0값을 가진다. 이와 반대로 입력 부하가 많고 전송률이 클 때는 큐에 셀이 많이 있을 확률이 크므로 상태 천이확률 행렬의 아래 부분의 숫자가 크고 입력 셀의 변화가 다양하게 일어날 가능성이 많으므로 상태 천이확률 행렬의 원소들의 값이 골고루 분포되어 있음을 확인 할 수 있다. 위의 큐 상태 천이 행렬의 평균값 행렬을 식 (20)과 같이 정의 할 수 있다.

$$\bar{A} = \sum_{i=0}^{\infty} A_i^t \quad (20)$$

식 (20)에서 정상 상태의 큐 상태 천이 확률을 구할 수 있고 평균 버퍼에 있는 셀의 개수는 정상 상태의 상태 천이 행렬  $\bar{A}$ 의 원소들의 값으로부터 구할 수 있다. 즉, 버퍼에 저장되어 있는 평균 셀의 개수는 초기상태 확률 행렬에서 안정한 상태의 확률 행렬로 천이 되므로 안정한 상태에서의 상태 확률 행렬의 원소들의 평균 버퍼 길이가 될 것이다. 그러므로 초기 상태를 모든 가능한 상태가 일정한 확률을 가진다고 하면, 평균 버퍼 길이를 다음 식 (21)과 같이 구할 수 있다. 식 (21)에서  $\bar{a}_{i,j}$ 는  $\bar{A}$ 의 원소들이고 N-1은 최대 버퍼의 크기이다.

$$\bar{Q} = \frac{1}{N} \sum_{i=0}^{i=N-1} \sum_{j=0}^{j=N-1} \bar{a}_{i,j} * j \quad (21)$$

**4. 성능측정 및 분석**

본 장에서는 DBSW 구조의 시뮬레이션을 수행하기 위한 환경을 설정하고 설정된 환경에서의 시뮬레이션을 수행한 결과와 수학적 분석의 결과를 비교 분석하여 수학적 분석의 정확성에 대해 논하고자 한다. 시뮬레이션 환경에서 출력 링크 용량은 128Mbps로 하였고 입력 부하를 조정하기 위해서 일정한 전송률을 가진 VC들의

개수를 조정하였다. 그리고 입력되는 셀 발생은 식 (22)와 같이 Poisson 분포를 따르도록 하였다.

$$P_r(T\text{시간동안에 } k\text{개 도착}) = \frac{(\lambda T)^k}{k!} e^{-\lambda T}$$

$\lambda$ : 입력 부하 (22)

수학적 결과는 3장의 상태 확률 계산 과정에 따라 상태 천이 확률 행렬을 반복적으로 계산하여 서비스 시간 t에서의 행렬의 원소 값을 구하고 식 (20)과 (21)에 의하여 평균 큐의 길이를 구한다. 입력부하에 따른 수학적 분석에 의한 평균 큐의 길이와 시뮬레이션을 수행하여 얻어진 평균 큐의 길이 비교 분석한다.

그림 7은 입력 부하가 출력 용량의 50%일 때 수학적 분석에 의한 평균 큐 길이와 시뮬레이션 수행에 의한 평균 큐의 길이를 나타내고 있다. 그림 7에서 제안된 알고리즘의 64Kbps VC들의 평균 큐의 길이가 갑자기 증가하여 평균 큐의 길이가 약 4셀 정도임을 알 수 있다. 이는 제안된 알고리즘이 BSW 구조상 입력 부하가 작을 경우 저속의 VC에 대해서는 서비스 호수가 다른 VC들에 비해 매우 작기 때문이다. 또한, 저속의 VC는 오랜 주기로 셀 서비스를 받기 때문에 평균 큐의 길이가 다른 VC에 비해 크다. 그림 7의 수학적 분석에 의한 결과와 시뮬레이션에 의한 결과가 비슷한데, 이는 수학적 분석과 시뮬레이션 수행이 정확함을 입증한다.

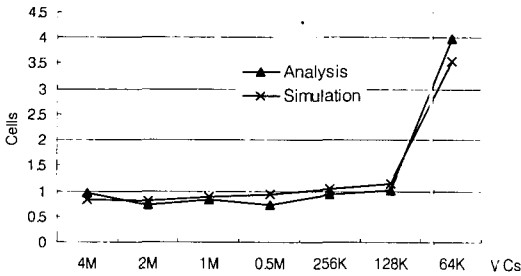


그림 7 평균 큐 길이 (입력부하 50%)

그림 8은 입력 부하가 출력 용량의 70%일 때 수학적 분석에 의한 평균 큐 길이와 시뮬레이션 수행에 의한 평균 큐의 길이를 나타내고 있다. 입력 부하가 70%일 때 시뮬레이션 결과 값과 수학적 분석에 의한 값이 전체적으로 비슷한 결과를 보이고 있다. 평균 큐의 길이가 4M VC들의 경우 크게 나오는 경우는 이진 전송 목록의 구성이 4M VC들의 서비스율이 다른 VC들의 서비스율보다 작게 구성되어 평균 큐의 길이가 다소 큰 결과를 보이고 있다.

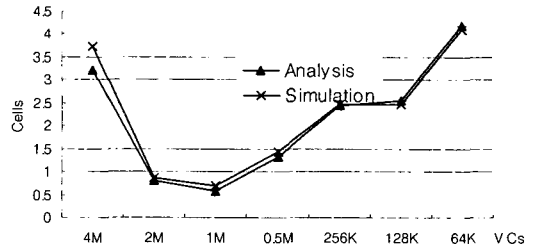


그림 8 평균 큐 길이 (입력부하 70%)

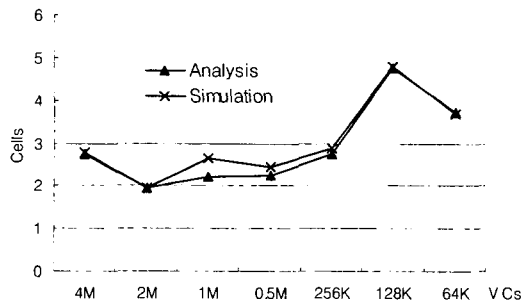


그림 9 평균 큐 길이 (입력부하 90%)

그림 9는 입력 부하가 출력 용량의 90%일 때 수학적 분석에 의한 평균 큐 길이와 시뮬레이션 수행에 의한 평균 큐의 길이를 나타내고 있다. 입력 용량이 90%에서는 128Kbps의 VC들의 평균 큐 길이가 약간 큰 결과를 보이고 있다. 이러한 결과는 128Kbps의 VC들의 서비스를 제 서비스 시간에 적절히 서비스를 수행하지 못한 결과이다. 이러한 이유는 이진 전송 목록의 구성이 128Kbps VC와 64Kbps VC등의 바쿠가 같은 슬롯에 속해 128Kbps VC의 셀들이 할당된 서비스 시간보다 늦게 서비스를 받고 있기 때문이다.

그림 10은 입력 부하가 출력 용량의 95%일 때 수학적 분석에 의한 평균 큐 길이와 시뮬레이션 수행에 의한 평균 큐의 길이를 나타내고 있다. 위의 그림과 비슷하게 시뮬레이션 결과와 수학적 분석에 의한 결과가 거의 비슷함을 보이고 있다. 그림 10에서 평균 큐의 길이가 각 VC에 대해 불규칙한 모양을 하고 있는데, 이는 제안된 알고리즘이 각 VC들을 서비스 할 때 버퍼 상태나 가중치 값의 상태에 따라 서비스를 수행하므로 서비스가 각 VC들에 대해 불규칙하게 수행되기 때문이다.

평균 큐의 길이는 각 VC들의 입력 부하에 비례하고 서비스율에 반비례한다. 그림 7, 8, 9, 10에서 시뮬레이션 결과와 수학적 분석에 의한 결과가 위에서 언급한



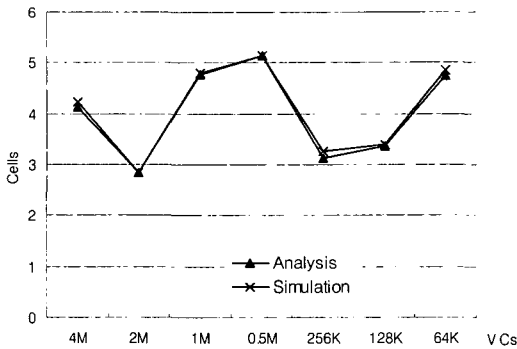


그림 10 평균 큐 길이 (입력부하 95%)

바와 같이 거의 비슷한 평균 큐 길이를 보이고 있다. 이는 수학적 분석과 시뮬레이션이 정확하게 수행되었음을 입증해 주고 있다. 시뮬레이션 결과와 수학적 분석의 결과 값이 입력 부하에 관계없이 거의 비슷한 값을 위의 그림들로부터 확인 할 수 있는데, 이러한 사실 또한 수학적 분석과 시뮬레이션의 수행이 정확하게 수행되었음을 입증한다. 또한, 위 그림에서와 같이 DBSW의 평균 큐 길이는 5셀 크기 이하인 반면 기존의 BSW 구조에서는 그 크기가 8정도로서[7] DBSW 구조에서 평균 큐 길이를 50%정도 줄일 수 있음을 확인할 수 있다.

## 5. 결론

패킷 스케줄링 알고리즘은 짧은 시간에 모든 계산을 수행해야 하므로 계산 복잡도가 작아야 한다. 이에 적합한 알고리즘이 WRR 알고리즘이다. 이러한 WRR 알고리즘을 효율적으로 하드웨어를 구현하기 위하여 DBSW 구조가 개발되었다. DBSW 구조가 기존의 BSW 구조보다 평균 큐잉 지연, 최대 버퍼의 크기, 각 VC를 공평하게 서비스 할 수 있는 공평성 등에서 성능이 우수함을 확인할 수 있다[8]. DBSW 구조는 기존의 BSW 구조보다 입력 부하에 관계없이 할당된 가중치에 충실하게 서비스를 수행한다. 이는 DBSW 구조가 BSW 구조보다 공평하게 서비스를 수행하고 있다는 증거이다. 그러므로 DBSW 구조가 WRR 알고리즘의 하드웨어 구현을 가능하게 하며 패킷 스케줄링 성능을 그대로 유지할 수 있는 구조로 사용될 수 있을 것이다. 또한, 멀티미디어 통신을 위해서는 다양한 서비스 요구사항을 만족시킬 수 있는 스케줄링 구조가 필수적인데 DBSW 구조는 이러한 멀티미디어 통신을 충실히 수행할 수 있을 것이다.

본 논문에서는 WRR 스케줄링 알고리즘을 효율적으

로 구현 할 수 있는 DBSW 구조에 대하여 살펴보고 이의 성능을 측정하기 위하여 수학적 성능 분석을 수행하였다. 수학적 분석은 기존의 큐잉 이론을 이용하여 DBSW 구조에서 서비스가 수행될 때마다 각 VC 큐의 상태 변화를 확률로 표현하여 그 변화 추이를 살펴 각 VC의 평균 큐 길이를 산출하였다. 그리고 수학적 분석의 타당성을 입증하기 위하여 시뮬레이션 결과와 수학적 분석 결과를 비교 분석하였다. 성능 분석결과 시뮬레이션 결과와 수학적 분석에 의한 결과가 비슷하였고 기존의 BSW 구조 보다 평균 큐 크기가 50%정도 감소함을 확인할 수 있었다. 이는 DBSW 구조가 기존의 BSW 구조보다 성능이 우수하며 수학적 분석이 정확함을 입증한다. 그러나 DBSW 구조가 BSW 구조보다는 하드웨어 복잡도가 약간 증가할 것이다. 이는 패킷 스케줄링 알고리즘의 복잡성이 약간증가하기 때문이다. DBSW 구조의 하드웨어 복잡도 증가정도를 파악하기 위해서 DBSW 구조를 하드웨어로 설계하여 그 복잡성을 확인해야 할 것이다.

## 참고 문헌

- [1] 김종권, "QoS 지원을 위한 패킷 스케줄링 알고리즘", TELECOMMUNICATIONS REVIEW, 제10권 3호, pp.481-491, May 2000.
- [2] J. M. Hyman, A. A. Lazar, G. Pacifici, "Real Time Scheduling with Quality of Service Constraints," IEEE Journal on Selected Areas in Communications, vol. SAC-9, no. 7, pp. 1052-1063, Sep. 1991.
- [3] Yuhua Chen, Jonathan S. Turner, "Designing of a Weighted Fair Queueing Cell Scheduling for ATM Network," *Proceedings of the Globecom '98 vol. 1*, pp. 405-410, Nov. 1998.
- [4] J. Golestani, "A Self-Clocked Fair Queueing Scheme for Broadband Applications," in *Proceedings of Infocom '94*, pp. 636-646, Apr. 1994.
- [5] M. Katevinis, S. sidiropoulos, C. Courcoubetis, "Weighted round-robin cell multiplexing in a general-purpose ATM switch chip," *IEEE J. on Sel. Area in Commun.*, Sep. 1991.
- [6] Y. T. Wang, T. P. Lin, K. C. Gan, "An improved scheduling algorithm for weighted round-robin cell multiplexing in an ATM switch," *Proc. of IEEE ICC'94*, pp. 1032-1037, May 1994.
- [7] 조해성, 정경택, 전병설, "셀 지연과 버퍼 이용률을 개선한 WRR 셀 스케줄링 알고리즘", 통신학회 논문지 제 25권 11호, pp. 272-275, Nov. 2000.
- [8] H. S. Cho, N. H. Kim, K. T. Chung, S. T. Lee, B. S. Chun, "DBSW Packet Scheduling for Multimedia

Service," Proc. of TENCON2001, Aug. :2001.



#### 조해성

1971년 10월 18일생. 1994년 2월 전북대학교 전자공학과 졸업. 1996년 2월 전북대학교 전자공학과 석사. 2001년 2월 전북대학교 전자공학과 박사. 2001년 3월 ~ 2001년 8월 한국표준과학연구원 Post-Doc. 2001년 9월 ~ 현재 건양대학교 IT 센터 교수. 관심분야는 멀티미디어 통신, 패킷 스케줄링, ATM 트래픽제어



#### 이상태

1954년 12월 6일생. 1977년 2월 아주대학교 공학사(공학사). 1992년 2월 전북대학교 전자공학과(공학석사). 1998년 2월 전북대학교 전자공학과(공학박사). 1985년 ~ 현재 한국표준과학연구원 연구원. 1999년 10월 ~ 현재 한국표준과학연구원 인간정보그룹 그룹장. 관심분야는 트래픽제어, 광대역통신망, 지능망



#### 전병실

1967년 2월 전북대학교 전자공학과(공학사). 1969년 2월 전북대학교 전자공학과(공학석사). 1971년 3월 전북대학교 전자공학과 교수 부임. 1974년 2월 전북대학교 전자공학과(공학박사). 1979년 ~ 1980년 미국 University Ncrtre Dame 객원교수. 1998년 9월 ~ 2000년 8월 전북대학교 학생처장. 1998년 3월 ~ 현재 전북대학교 정보통신연구소 연구위원. 1971년 ~ 현재 전북대학교 전자공학과 교수. 관심분야는 ATM 교환기술, ATM트래픽 제어, 무선 ATM망에서 멀티미디어 서비스