큐잉 이론을 이용한 DBSW 알고리즘 성능 분석

DBSW Algorithm Performance Criteria Using Queueing Theory

조해성
건양대학교 전자정보공학과

Hae-Seong Cho(hscho@konyang.ac.kr)

요약

WRR 기법은 대역을 보장해야 하는 스케줄링 기법으로 개발되었다. 최근에는 무선 통신 네트워크에서 다양한 대역이나 지연과 같은 차등화 된 서비스들을 제공 할 수 있도록 각 큐에 다른 가중치를 지정함으로써 여러 큐들을 직접적으로 관리할 수 있도록 하였다. 또한, WRR 알고리즘은 DBSW 구조에 의해 효율적으로 구현할 수 있다.

본 논문은 WRR 구현의 대표적인 방법인 DBSW 구조의 수학적 분석을 수행하고 분석결과와 시뮬레이션 결과를 비교 분석한다. 분석 데이터와 시뮬레이션 데이터는 수학적 분석이 타당하게 전개되었음을 입증하는 성능 결과를 보여주고 있다.

■ 중심어 : □WRR□스케줄링□DBSW□수학적분석□

Abstract

WRR scheme was developed for scheduling scheme for guaranteeing bandwidth requirement. In these days, WRR scheme can handle multiple queues straightforwardly by putting a different weight on each queue in order to offer differentiated services such as different bandwidth or delay bound in wireless communication network. Also, WRR algorithm can be implemented efficiently by dynamic binary scheduling wheel(DBSW) architecture.

This paper performs the mathematical analysis of the DBSW architecture and compares the results with simulation results. The analysis data and simulation data show that the mathematical analysis developed properly.

■ keyword : □WRR□Scheduling□DBSW□Mathematical Analysis□

I. 서 론

차세대 인터넷에서 다양한 서비스를 보장하기 위하여 차등화 서비스 개념이 광범위하고 유연한 네트워크 설계에 적합한 구조로 받아들여지고 있다[1]. 차등화 서비스란 다양한 서비스 요구사항을 만족할 수 있도록 네트워크에서 전송 서비스를 차등으로 나누어 제공하는 서비스를 의미한다. 이러한 차등화 서비스가 등장하게 된 배경은, 인터넷에서 초고속 네트워크 실현으로 VoIP(Voice of IP), VPN(Virtual Private Network),

접수번호 : #080831-001
접수일자 : 2008년 08월 31일
심사완료일 : 2008년 10월 31일
교신저자 : 조해성, e-mail : hscho@konyang.ac.kr

VOD(Video On Demand), 화상회의 등과 같은 다양한 형태의 응용서비스를 요구하고 있기 때문이다. 이러한 서비스들은 다양한 수준의 서비스 품질을 요구하지만, 최선형 서비스(Best-Effort Service)를 근간으로 한 현재의 인터넷은 서비스 품질에 대한 고려가 없기 때문에 다양한 서비스 제공에 많은 문제점을 가지고 있다. 이러한 문제들을 해결하기 위해 새로운 IP QoS(Quality of Service)모델 등에 관한 연구가 진행되어 왔다[2].

QoS에 관련된 많은 연구 중에서 모든 패킷 흐름에 대해서 QoS를 제공해주기 위한 IntServ 모델은 현실적으로 구현이 복잡하고 오버헤드가 많기 때문에 확장성이 나쁜 단점이 있다[3]. 이러한 단점을 극복하기 위해 IETF(Internet Engineering Task Force)에서는 각기 다른 사용자 그룹에게 서로 다른 수준의 서비스를 제공할 수 있는 DiffServ 모델을 제안하게 되었다[4].

DiffServ는 복잡한 기능의 처리는 경계(edge) 라우터에서 수행하게 하고 코어(core) 라우터는 각 클래스의 PHPs(Per -Hop Behaviors)에 따라 패킷을 전달함으로써 확장성을 향상시켰다[5]. DiffServ에서 제공하는 서비스 중에 중요한 서비스 중에 하나가 Expedited Forwarding(EF) 서비스이다. EF 클래스는 가장 우선순위가 높은 클래스로서 인터넷 전화, 화상회의 등에 알맞은 작은 패킷 지연시간과 지터, 작은 패킷 폐기율을 제공받는다[6]. 현재 DiffServ에서 많이 거론되고 있는 대표적인 스케줄링 기법으로는 PQ(Prio- rity Queue), WRR(Weighted Round Robin)와 WFQ (Weighted Fair Queueing)등의 스케줄링 기법이 연구되어져 있다. WRR은 각 큐에 가중치를 할당하여 가중치에 따라 최소 대역을 보장한다[7]. 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[8]. BSW 구조는 최소 하드웨어 비용으로 광범위한 전송률을 제공하고 과부하에는 최소 전송률을 보장할 수 있는 효율적인 패킷 스케줄링 구현 알고리즘 이다.

본 논문에서는 기존에 제안된 DBSW 구조의 수학적 분석을 수행하고 수학적 분석의 결과와 시뮬레이션 결과를 비교하여 DBSW 구조의 수학적 성능분석을 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.

본 논문의 구성은 제 2장에서 DBSW 구조에 대하여 살펴보고, 제 3장에서는 WRR 알고리즘의 수학적 분석 방법을 설명한다. 제 4장에서는 WRR 분석 방법을 확장하여 DBSW 방식의 수학적 분석 방법을 서술하였다. 제 5장에서 시뮬레이션에 의한 성능측정과 수학적 분석의 성능 결과를 비교 분석하여 수학적 분석 방법의 타당성을 검증하고, 마지막으로 6장에서는 결론으로 논문 내용을 개괄적으로 정리하였다.

II. DBSW 구조

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 서비스 율이 떨어지는 단점을 DBSW 구조 알고리즘은 개선하고 있다[8].

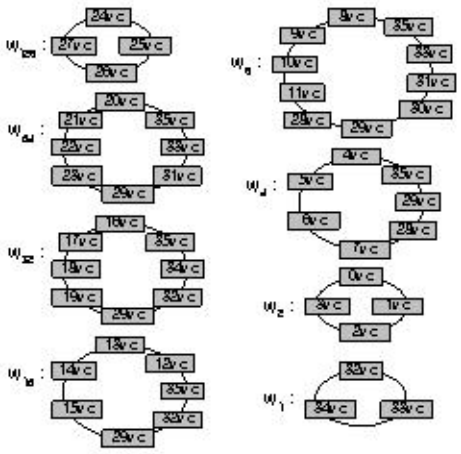

그림 1. 이진 스케줄링 바퀴

[그림 1]은 DBSW 구조의 이진 버퍼를 보여 주고 있다. DBSW 구조는 기존 BSW 구조에서 각 VC에 최소 대역의 2의 지수승 만을 제공하는 문제점을 개선하기 위하여 각 VC에 임의 대역을 할당 할 수 있도록 다음과 같은 가중치 값과 가중치 목록을 작성하였다. VC i의 가중치 계산은 식 (1)과 같다.

□ □

(1)

□

식 (1)에서 C는 출력 링크의 용량이고 W는 모든 가중치 값들의 합이고 ri는 VC i에 할당된 전송율이다. DBSW 구조는 각 VC는 스케쥴링 서버가 제공하는 가중치 값에 근거하여 가중치 목록을 식 (2)에 따라 구성한다.

□ (2)□

식 (2)에서 VCk(W)는 VC k의 가중치 전송목록이고 Wk는 VCk의 가중치 값을 표시한다. 식 (2)에서 스케쥴링 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.

□

Ⅲ. WRR 수학적 분석 □

차등서비스를 제공하는 네트워크에 사용되는 스케쥴러는 각 VC에 큐를 할당하고 이 큐에 대하여 스케쥴링을 수행한다. 그러므로 이를 해석하기 위해서는 먼저 큐에 대한 모델을 적용해야 한다. 아래 [그림 2]는 일반적인 큐의 모델을 보여주고 있다. [그림 2]에서 λ는 입력의 평균 부하이고 μ는 큐에 대한 서비스 율이다.

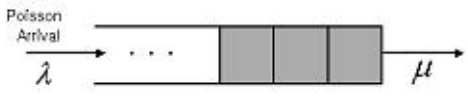

그림 2. 큐잉 모델

[그림 3]은 VC큐 모델에 대한 큐의 상태 천이를 보여주고 있다. [그림 3]에서 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.

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

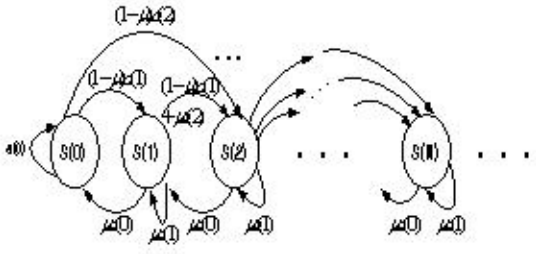

그림 3. VC 큐의 상태 천이도

큐의 상태 천이도를 근거로 한 각 상태에 대한 확률식은 식 (4)와 같다.

(4)

...

WRR 서비스 구조에서는 각 VC 큐의 서비스 율이 일정하지 않기 때문에 매 서비스가 발생할 경우에 대한 상태를 가정하고 이 상태에 따른 수식을 작성하고자 한다. 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)는 VC의 큐에 q개의 셀이 존재하고 가중치 계수기의 값이 w인 상태를 나타낸다. VCi에 대한 상태 천이 확률 행렬을 GTi라 정의하면 GTi(q,w)→(q′,w′)는 VCi의 상태가 (q,w)에서 (q′,w′)로 천이 될 확률을 나타낸다. 또한, 각 VC에 대한 서비스 수행을

나타내는 si(t)를 정의 하고자 한다. si(t)는 VC i가 서비스 시간 t에 서비스를 받는다면 1로 세트하고 그렇지 안으면 0으로 세트된다. WRR의 경우 각 VC에 할당된 가중치 Wi 값에 따라 서비스가 수행되므로 Si(t)를 다음과 같이 정의 할 수 있을 것이다.

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

수식 (5)에서 Wtot는 모든 VC의 가중치 값을 다 더한 값이고 $\square x \square$는 x 보다 큰 최소 정수를 나타낸다. 이를 기반으로 버퍼의 크기가 무한한 WRR 모델에 대한 상태 천이 확률 GTi(q,w)→(q',w')를 구하면 다음과 같다.

$$G^t_{(q,w)\to(q',w')}=\begin{cases}a_i(q'-q+1) & \text{if } x_i(0)=1, w'=w-1\ge 0, q'\ge q)\ 0\\ a_i(q'-q) & \text{if } x_i(0)=1, w'=w-1\ge 0, q'\ge q=0\\ a_i(q'-q) & \text{if } x_i(0)=0, w'=w, q'\ge q\ge 0\\ & \text{or } x_i(0)=1, w'=w=0, q'\ge q\ge 0\\ 0 & Otherwise\end{cases} \quad (6)$$

수식 (6)에서 ai(q)는 서비스 타임 슬롯 동안에 VC i의 큐로 q개의 셀이 들어올 확률을 나타낸다. 그리고 GTi(q,w)→(q',w')는 서비스 시간 t에 VC i 큐의 상태가 (q,w)에서 다음 서비스 시간 t+1에 (q',w')상태로 천이 될 확률을 나타낸다. 위 식에서 각 VC의 상태 천이는 서비스 수행 상태 즉, si(t)에 따라 다른 확률로 천이 됨을 보여주고 있다. 큐 상태 천이 확률 행렬을 구하기 위하여 큐 상태 천이 확률 행렬 Ait를 정의한다. 또한, VC i의 상태 확률 행렬 Sit를 정의한다. Sit 행렬은 VC i의 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 i에 대하여 서비스 시간 t에서 가중치 변환 행렬은 Kit로 정의한다.

$$K^t_i=\begin{bmatrix}1&0&0&\square&0&0\\1&0&0&\square&0&0\\0&1&0&\square&0&0\\0&0&1&\square&0&0\\\square&\square&\square&\square&\square&\square\\0&0&0&\square&0&1\\0&0&0&\square&0&0\end{bmatrix} \quad (7)$$

수식 (7)에서 Kit의 차원은 VC i에 할당된 가중치 값 Wi이다. 그러므로 Kit는 가중치 값이 0인 것을 포함하기 때문에 (Wi+1)×(Wi +1) 행렬이다.

Ait는 서비스 시간 t에 VC i큐의 상태가 천이 될 확률들을 나타낸다. 상태 천이 확률 행렬은 si(t)의 상태에 따른 큐의 상태 천이 확률 행렬 Ait를 식 (6)으로부터 구할 수 있다.

$$\begin{bmatrix}\square&&\square&&\square&\square&\square&\square\\&&&&&\square&\square&\square\\\square&&\square&&\square&\square&\square&\square\\&&&&&\square&\square&\square\\\square&&\square&&\square&\square&\square&\square\end{bmatrix} \quad (8)$$

$$\begin{bmatrix}\square&&\square&&\square&\square&\square&\square\\&&&&&\square&\square&\square\\\square&&\square&&\square&\square&\square&\square\\&&&&&\square&\square&\square\\\square&&\square&&\square&\square&\square&\square\end{bmatrix} \quad (9)$$

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

위에서 살펴본 바와 같이 서비스 시간 t에서의 VC i의 상태 확률 행렬 Sit는 이전의 상태 확률 행렬 Sit-1, 큐 상태 천이 확률 행렬 Ait, 가중치 변환 행렬 Kit의 곱에 의해서 구할 수 있고 Sit-1은 그 이전의 상태로 구할 수 있으므로 이를 반복적으로 수행하면 다음 수식 (10)과 같이 구할 수 있다.

$$S^t_i=A^t_iS^{t-1}K^t_i=A^t_iA^{t-1}_iS^{t-2}K^{t-1}K^t_i= \square=A^t_iA^{t-1}_i\square A^1_iS^0K^1_iK^2_i\square K^t_i \quad (10)$$

수식 (10)에서 서비스 시간 t에서 VC i의 상태 확률 행렬은 초기 상태 확률 행렬 Si0와 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.

$$K'_t = \begin{matrix} 0 & 0 & 0 & \square & 1 \\ 0 & 0 & 0 & \square & 1 \\ 0 & 0 & 0 & \square & 1 \\ 0 & 0 & 0 & \square & 1 \\ \square & \square & \square & \square & \square \\ 0 & 0 & 0 & \square & 1 \end{matrix} \qquad t = WeightCountReset \tag{11}$$

가중치 리셋은 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 평균지연을 구할 수 있다.

Ⅳ. DBSW 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 스케쥴링 알고리즘에 의해서 각 VC의 서비스 수행이 결정된다. 그러므로 서비스 수행을 알려주는 변수 si(t)를 식 (12)와 같이 정의 할 수 있다.

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

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

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

식 (13)은 DBSW 구조의 셀 스케쥴링 알고리즘에서 어느 한 셀이 서비스되기 전에 먼저 서비스 받는 VC i의 si(t)를 1로 세트하여 서비스 받게 됨을 알려준다. 그러나 DBSW 구조의 셀 스케쥴링 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 Si(q)로 정의한다. Si(q)는 VC i의 큐에 q개의 셀이 존재하는 상태를 나타낸다.

식 (13)을 이용하여 기존의 DBSW 구조의 스케쥴링 알고리즘에서의 상태 천이 확률을 이용하여 큐 상태 천이 확률 행렬을 계산할 수 있다. 이는 WRR 스케쥴링 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 N으로 제한하였다. 큐의 길이를 N으로 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 $G^T_{[q]\to[q]}$은 아래 수식과 같다.

$$G^i_{(q \to q')} = \begin{cases} a_i[q'-q+1] & \text{if } s_i(t)=1,\ N > q' \ge q-1 \ge 0 \\ \sum_{k=N}^{\infty} a_i[k-q+1] & \text{if } s_i(t)=1,\ q'=N,\ q>0 \\ a_i[q'-q] & \text{if } s_i(t)=1,\ N > q' \ge q = 0 \\ a_i[q'-q] & \text{if } s_i(t)=0,\ N > q' \ge q \ge 0 \\ \sum_{k=N}^{\infty} a_i[k-q] & \text{if } s_i(t)=0,\ N=q' \ge q \ge 0 \\ 0 & \text{Otherwise} \end{cases} \quad (14)$$

식 (14)에서 DBSW 구조의 알고리즘은 큐에 셀이 없는 경우에는 si(t)가 1인 경우에도 셀 전송 서비스를 수행하지 않는다. 식 (14)의 상태 천이 확률 식을 근거로 큐의 길이가 N인 경우의 상태 천이도를 작성하면 아래 [그림 4]와 같다. [그림 4]에서 큐가 비어 있는 경우 서비스 상태에 관계없이 서비스가 수행되지 못하기 때문에 입력되는 셀의 개수에 따라서 상태가 변화함을 확인할 수 있다.

[그림 4]를 참조하여 큐의 길이가 N으로 제한된 경우의 큐의 상태 천이 확률 행렬 Ait를 구하면 아래 식 (15), (16)과 같다.

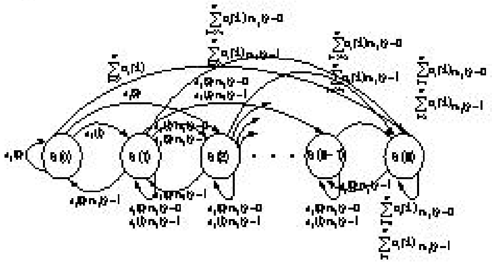

그림 4. DBSW 구조의 큐 상태 천이도

$s_i(t)=0$

$$A^t_i = \begin{bmatrix} a_i[0] & 0 & 0 & \cdots & 0 \\ a_i[1] & a_i[0] & 0 & \cdots & 0 \\ a_i[2] & a_i[1] & a_i[0] & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_i[N-1] & a_i[N-2] & a_i[N-3] & \cdots & 0 \\ \sum_{k=N}^{\infty} a_i[k] & \sum_{k=N-1}^{\infty} a_i[k] & \sum_{k=N-2}^{\infty} a_i[k] & \cdots & \sum_{k=1}^{\infty} a_i[k] \end{bmatrix} \quad (15)$$

$s_i(t)=1$

$$A^t_i = \begin{bmatrix} a_i[0] & a_i[0] & 0 & \cdots & 0 \\ a_i[1] & a_i[1] & a_i[0] & \cdots & 0 \\ a_i[2] & a_i[2] & a_i[1] & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_i[N-1] & a_i[N-1] & a_i[N-2] & \cdots & a_i[0] \\ \sum_{k=N}^{\infty} a_i[k] & \sum_{k=N}^{\infty} a_i[k] & \sum_{k=N-1}^{\infty} a_i[k] & \cdots & \sum_{k=1}^{\infty} a_i[k] \end{bmatrix} \quad (16)$$

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

$$S^t_i = A^t_i S^{t-1}_i = A^t_i A^{t-1}_i S^{t-2}_i = \cdots = A^t_i A^{t-1}_i \cdots A^1_i S^0_i \quad (17)$$

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 Sit, Ait 계산
step ⅴ) step ⅲ)와 step ⅳ) 반복

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

Load = 50%, Rate = 64Kbps (18)

Ai =

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	0.058	0.224

Load = 90%, Rate = 1Mbps (19)

Ai =

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

위의 상태 천이확률 행렬을 살펴보면 입력 부하가 적고 전송률이 적을 때는 큐에 셀이 적게 있을 확률이 크므로 상태 천이확률 행렬의 윗 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.

-

(20)

식 (20)에서 정상 상태의 큐 상태 천이 확률을 구할 수 있고 평균 버퍼에 있는 셀의 개수는 정상 상태의 상태 천이 행렬 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 는 의 원소들이고 N-1은 최대 버퍼의 크기이다.

(21)

V. 성능 측정 및 평가

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.

Pr(T시간동안에 k개 도착) = λ: 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.

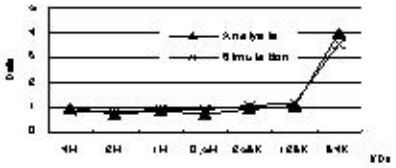

그림 5. 평균 큐 길이 (입력부하 50%)

[그림 5]는 입력 부하가 출력 용량의 50%일 때 수학적 분석에 의한 평균 큐 길이와 시뮬레이션 수행에 의한 평균 큐의 길이를 나타내고 있다. [그림 7]에서 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 회수가 다른 VC들에 비해 매우 작기 때문이다. 또한, 저속의 VC는 오랜 주기로 셀 서비스를 받기 때문에 평균 큐의 길이가 다른 VC에 비해 크다. [그림 4]의 수학적 분석에 의한 결과와 시뮬레이션에 의한 결과가 비슷한데, 이는 수학적 분석과 시뮬레이션 수행이 정확함을 입증한다.

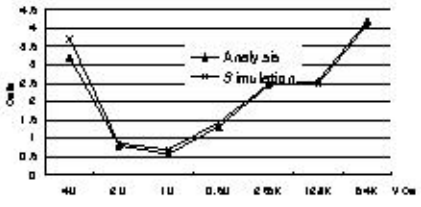

그림 6. 평균 큐 길이 (입력부하 70%)

[그림 6]은 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 서비스 율이 다른 VC들의 서비스 율보다 작게 구성되어 평균 큐의 길이가 다소 큰 결과를 보이고 있다.

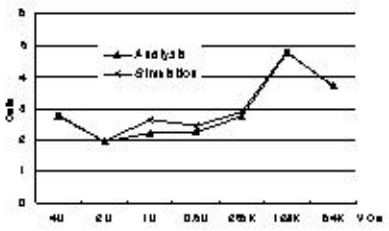

그림 7. 평균 큐 길이 (입력부하 90%)

[그림 7]은 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 바뀌가 같은 슬롯에 속해 128Kbps VC의 셀들이 할당된 서비스 시간보다 늦게 서비스를 받고 있기 때문이다.

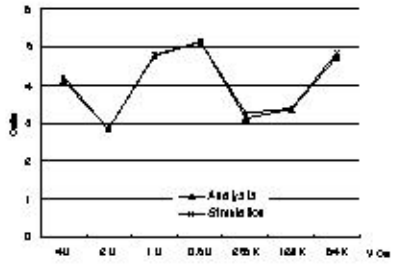

그림 8. 평균 큐 길이 (입력부하 95%)

[그림 8]은 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 8]에서 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 5][그림 6][그림 7][그림 8]에서 시뮬레이션 결과와 수학적 분석에 의한 결과가 위에서 언급한 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 값임을 위의 그림들로부터 확인 할 수 있는데, 이러한 사실 또한 수학적 분석과 시뮬레이션의 수행이 정확하게 수행되었음을 입증한다.

VI. 결 론

차세대 유무선 통합망에서 다양한 서비스를 제공하기 위한 차등화 서비스 방식에서 스케쥴링 알고리즘은 처리속도가 빨라야하므로 구현 복잡도가 간단해야 한다. 이러한 차세대 통신망에 적합한 스케쥴링 방식으로 WRR 알고리즘이 널리 활용되고 있다. 이러한 WRR 알고리즘을 효율적으로 하드웨어로 구현하기 위하여 DBSW 구조가 개발되었다. DBSW 구조는 기존의 BSW 구조 보다 입력 부하에 관계없이 할당된 가중치에 충실하게 서비스를 수행한다. 그러므로 DBSW 구조가 WRR 알고리즘의 하드웨어 구현을 가능하게 하며 패킷 스케쥴링 성능을 그대로 유지할 수 있는 구조로 사용될 수 있을 것이다.

본 논문에서는 WRR 스케쥴링 알고리즘을 효율적으로 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.

참 고 문 헌

[1] A. Charny et al., *Supplemental information for the new definition of the EF PHB*, IETF RFC 3247, Mar. 2002.

[2] X. Xiao and L. M. Ni, "Internet Qos: A Big Picture," IEEE Network, Vol.13, No.2, pp.8-18, Mar. 1999.

[3] S. Shenker, C. Partridge, and R. Guerin, *Specification if Guaranteed Quality of Service to Intergrated Services(IntServ)*, RFC 2212, Sep. 1997.

[4] S. Blake, D. Black, M. Carlson, Wang and Weiss, *An Architecture for Differentiated Services*, RFC 2475, Dec. 1998.

[5] K. Nichols, S. Blake, F. Baker, and D. Black, *Definition of the Differentiated Services Field(DS Field) in the IPv4 and IPv6 Headers*, RFC 2474, Dec. 1998.

[6] Y. Jiang, "Delay bound and packet scale rate guqrantee for some expedited forwarding networks," Comp. Networks 50, pp.15-28, 2006.

[7] Y. T. Wang, T. P. Lin, and 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.

[8] H. S. Cho, N. H. Kim, K. T. Chung, S. T. Lee, and B. S. Chun, "DBSW Packet Scheduling for Multimedia Service," Proc. of TENCON2001, Aug. 2001.

저 자 소 개

조 해 성(Hae-Seong Cho) 종신회원

- 1994년 2월 : 전북대학교 전자공학과 (공학사)
- 1996년 2월 : 전북대학교 전자공학과 (공학석사)
- 2001년 2월 : 전북대학교 전자공학과 (공학박사)
- 2006년 현재 : 건양대학교 전자정보공학과 교수

<관심분야> : QoS 스케줄링, 멀티미디어 통신, DiffServ, MPLS