

이동통신망에서 WRR 기법의 스케줄링 알고리즘 연구

A Study on Scheduling Algorithm of WRR Method in Wireless Network

조해성

건양대학교

Cho Hae-Seong

Konyang University

요약

본 논문에서는 이동통신 망에 적합한 WRR 기법의 스케줄링 알고리즘을 제안하였고 그 성능을 분석하였다. 이동통신 네트워크에서 실시간 응용 서비스를 제공하기 위한 서비스품질(QoS)에 대한 요구는 상당기간 많은 연구와 표준화 활동을 독려해왔다. 특히, 무선 환경에서의 스케줄링 알고리즘에 대한 많은 연구가 진행되었다. 이러한 연구 결과로, WRR 기법의 BSW 스케줄링 알고리즘이 무선 환경에 적합한 알고리즘으로 개발되었다. 그러나 BSW 알고리즘은 구현의 복잡성으로 인해 빠른 스케줄링을 요구하는 무선 환경에서는 그 성능이 저하된다. 이러한 문제를 해결하기 위해서 본 논문에서는 BSW 알고리즘의 구현 복잡성을 감소시키며 성능을 향상시키는 알고리즘을 제안하고 그 성능을 분석하였다.

Abstract

In this paper, I proposed the algorithm that realize scheduler of WRR method in mobile network and analyze the performance of the proposed algorithm. The need for providing quality of service(QoS) for real-time applications in wireless networks has been driving research activities and standardization efforts for some time. In particular, there have been considerable research of scheduling algorithm for wireless environments. The BSW algorithm of WRR method that is suited to wireless environment is developed to the results of these effort. But BSW algorithm is deteriorated the performance by realization complexity in wireless environment is necessary to fast scheduling. To solve of these problem, this paper proposed the scheduling algorithm which degrades of implementation complexity and which improves the performance and analysis of the performance of the proposed algorithm.

I. 서론

향후 몇 년간 가입자망은 경쟁의 가속화로 서비스 제공자들은 가입자에게 다양한 솔루션 및 서비스를 제공하기 위해 치열한 기술개발 경쟁을 하고 있다. 특히, 무선 영상 서비스의 약진에 의한 고속 액세스 장비의 개발이 필요하게 되었다. B-WLL은 기존의 트윈스터 페어 전화선, 동축 및 광케이블을 대체할 수 있는 준 밀리미터파(20GHz대) 및 밀리미터파 대역(40GHz)의 무선 가입자 망으로 구성되며, 10 BaseT(10Mbps), ATM-25(25Mbps), T1/E1 등의 가입자 회선을 제공하여 고속 인터넷, 음성전화, VOD, 화상회의 등의 서비스 제공이 가능하도록 개발되고 있다. B-WLL 시스템은 광케이블을 통해 분배 서비스를 가입자 근처의 무선 전송장치까지 전송한 후, 무선으로 분배하는 고주파 단거리 시스템이다[1]. 가입자 이용측면에서 볼 때 B-WLL은 속도증가에 요구되는 비용이 상대적으로 적으며, 유선에 비해 품질이 우수하고 제공 속도 예측이 용이하여 여러 가지 요소에 의해 경제적으로 서비스가 가능하다. 이 같은 장점 때문에 국내외적으로 인

터넷사업을 준비하고 있는 경쟁사업자 뿐만 아니라 기존 사업 자까지도 B-WLL 주파수 획득에 주력하고 있다[2].

위와 같이 향후의 가입자 망은 무선으로 진화해 가고 있으며 무선망에서의 음성, 인터넷, 화상전화, CATV 등의 다양한 서비스가 제공되어야 한다[3]. 이러한 서비스들은 다른 트래픽 특성과 성능 요구사항을 가진 다양한 트래픽 형태를 가지고 있기 때문에 무선 네트워크에서의 QoS 보장은 중요한 문제의 하나가 되었다. 최고전송률, 평균전송률, 버스트니스 등의 트래픽 특성과 지연, 셀 손실률, 지터 등의 서비스 요구사항에 근거하여 무선 네트워크는 각 응용에 필요한 자원을 할당한다.

WRR(Weighted Round Robin)[4,5]은 초기 FSA(Fair Scheduling Algorithm)의 한 형태로 제안되었다. 그러므로 WRR 셀 스케줄링 알고리즘은 CBR, VBR 트래픽 뿐만 아니라 ABR, UBR 트래픽의 데이터에 적합하도록 설계되었다. WRR 알고리즘은 스위치에서 각 VC에 QoS 요구사항을 만족시키며 공평하게 대역을 할당하기 위해서 각 VC별로 대역을 할당하도록 설계되었다. 위와 같은 특성들로 인해 WRR은 무

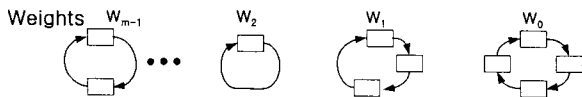
선통신 네트워크에서 스케줄링 알고리즘으로 각광받고 있다. 그러나 WRR 스케줄러를 구현하기 위해 계산 복잡도와 하드웨어 요구사항이 스케줄링을 빠르고 간편하게 수행해야 하는 무선 네트워크 환경에서는 지나친 부담으로 작용한다. 이러한 문제를 해결하기 위해 WRR 스케줄러를 구현하기 위해 계산 복잡도와 하드웨어 요구사항을 현저히 경감시키는 BSW (Binary Scheduling Wheel) 기법이 제안되었다[6]. 그러나, BSW 방식은 어느 한 VC의 큐가 비어있게 되면 셀 서비스를 수행하지 못하므로 무선 환경에서는 각 VC의 가중치와 스케줄러의 처리율 등의 심각한 성능저하를 유발한다.

본 논문에서는 위에서 언급한 BSW 기법의 문제를 해결하기 위한 새로운 BSW 방법을 이용한 WRR 스케줄링 알고리즘을 제안하였다. 제안된 알고리즘은 BSW 기법의 문제인 각 VC큐가 비어있을 경우 다른 VC 셀을 서비스하게 함으로써 WRR 스케줄러의 성능을 유지할 수 있도록 하였다.

II. BSW 알고리즘

1. Scheduling wheels

이 알고리즘에서는 링크 대역을 2의 지수승으로 나눈 것을 rate로 규정하고 최소 rate를 $\frac{1}{2^n}$ 으로 제한하였다. 따라서 rate i 는 $\frac{w_i}{2^n}$ 으로 계산되며 w_i 는 가중치로서 정수형이다. BSW 알고리즘의 동작원리는 그림 1에서와 같이 각 가중치에 대해서 한 개씩, m 개의 Scheduling Wheels를 구성하고 rate와 목적지에 따라 각 VC들을 Scheduling Wheels에 배치해서 같은 wheel에 놓인 VC들은 원형 리스트로 구성되게 된다. 그래서 서로 다른 rate의 수가 m 개를 초과하지 않는 범위에서 각 VC에 할당된 rate가 동적으로 변화될 수 있다. 그림 1은 m 개의 Scheduling Wheels에 대한 예로서 작은 상자들은 비어있지 않는 큐를 나타내고 있다.



▶▶ 그림 1. Scheduling Wheels

Notations

m = number of distinct rate

$$\text{MinimumRate} = \frac{1}{2^n}$$

w_i = weight i

$$r_i = \frac{w_i}{2^n}, \text{ rate } i$$

Count_i = Number of queues on scheduling wheel i

$$\text{WeightCountProduct}_i = w_i \times \text{Count}_i$$

$$\text{BinaryProduct}_i = 2^{\log_2 \lceil \text{WeightCountProduct}_i \rceil}$$

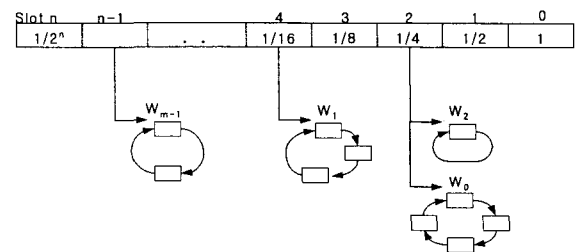
$$\text{Diff}_i = \text{BinaryProduct}_i - \text{WeightCountProduct}_i$$

$$\text{BinaryRate}_i = \frac{\text{BinaryProduct}_i}{2^n}$$

Next Q_i = Pointer to the next queue to be served on wheel i

각 Wheel들에 대해 먼저 WeightCounterProduct, binary product 그리고 WeightCounterProduct와 binary product의 차를 계산한다. 그리고 할당된 대역폭이 1을 넘지 않는 범위에서 m 개의 서로 다른 binary rate를 그림3과 같이 구성할 수 있다. 그런데 BinaryProduct는 WeightCountProduct를 2의 지수승으로 맞추는 값이기 때문에 서로 다른 WeightCountProduct 값을 가진 scheduling wheel들이 BinaryProduct와 Binary rate는 서로 같게 될 수 있다. 이러한 이유로 해서 몇몇 VC들은 공평하게 할당된 대역보다 더 많은 대역을 얻게 되는 결과를 초래하게 된다.

그림 2는 $n+1$ 개의 슬롯으로 구성된 Binary rate list를 보여주고 있는데 각 슬롯은 각각의 Binary rate와 관련이 되어있으며 같은 Binary rate를 갖고있는 scheduling wheel들은 하나의 슬롯에 놓이게 된다. 일반적으로 Binary rate list의 슬롯에 있는 scheduling wheel이 일단 선택되어지면 그 슬롯에 있는 모든 scheduling wheel들이 한번씩 서비스되며 이때 현재의 포인터의 위치가 다음 wheel의 큐로 이동하게 된다. 그래서 이러한 선택 과정으로 인해 각각 다른 가중치를 가진 VC의 셀들이 서로 끼워지게 되어 출력의 Burstness가 감소하게 된다.



▶▶ 그림 2. 이진바퀴 목록

첫 번째 슬롯의 scheduling wheel은 두 번째 슬롯의 scheduling wheel보다 두 배로 서비스되고 세 번째 슬롯의 scheduling wheel보다는 네 배로 서비스를 받게 된다. 그래서

슬롯의 scheduling wheel들을 서비스하기 위해서 $n+1$ bit의 binary counter를 사용하여 binary rate list의 슬롯들을 선택하는데 binary counter에서 처음으로 1이 나오는 비트는 다음 비트보다 두 배만큼 변하게 되는데 j 번째 비트의 변화로 j 슬롯의 scheduling wheel들에 대한 서비스가 시작된다.

2. Fast forward counter

Binary counter를 1씩 증가시켜 서비스할 슬롯을 찾아야 하는데 Binary counter에서 변화된 비트에 해당하는 위치를 서비스 할 슬롯으로 결정하는데 이 슬롯을 eligible slots으로 부른다. 그런데 카운터 값을 증가시키면서 변화된 비트에 해당하는 슬롯에 붙어 있는 scheduling wheel들을 찾아야 하는데 이 eligible slots에 scheduling wheel이 없는 경우에 계속해서 비어있지 않은 slot을 검색하는데 최악의 경우 2ⁿ 정도의 단계가 요구된다. 이를 해결하기 위해서 빈 슬롯은 검색하지 않고 통과하는 fast forward counter 알고리즘을 아래와 같이 소개한다.

Initially,

PreviousCounter = 0;

CurrentCounter = 0;

Mask : Bit i is set to '1' slot i is non-empty,
'0' if slot i is empty;

Loop:

CarryIn = Position of the least significant '1' bit of

Mask;

PreviousCounter = *CurrentCounter*,

CurrentCounter = *CurrentCounter* + *CarryIn*;

ChangingBits = Changing bits of Counter;

CurrentMask = *Mask*;

While((*CurrentMask* & *ChangingBits*) != 0)

CurrentSlot = Position of the least significant '1'
bit of (*CurrentMask* & *ChangingBits*);

Serve all scheduling wheels on slot *CurrentSlot*;

CurrentMask [*CurrentSlot*] = 0;

If(slot *CurrentSlot* becomes empty)

Mask [*CurrentSlot*] = 0;

end

If(New scheduling wheel is added to empty slot j)

Mask [j] = 1;

end

end

위 알고리즘에서 CarryIn을 가지고 CurrentCounter를 갱신함으로써 빈 슬롯은 통과하고 비어있지 않은 슬롯만을 선택하여 서비스를 수행하는데 이때 Selection Time은 슬롯의 전체 수에 무관하게 된다.

III. 제안된 BSW 알고리즘

기존의 BSW 스케줄링 알고리즘은 각 VC의 전송률을 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. 본 논문에서는 기존의 BSW 스케줄링 알고리즘의 문제점을 개선하기 위한 셀 스케줄링 알고리즘을 제안하였다. 제안된 셀 스케줄링 알고리즘에서는 각 VC에 W , lend, loan의 추가적인 변수를 사용하여 binary rate list의 슬롯에 있는 각 wheel의 큐를 순차적으로 방문하는 과정에서 그 큐에 전송할 셀이 있고 가중치가 0이 아니면 한 셀을 전송한 후 가중치를 감소시키는데, 각 VC에 할당된 실제의 가중치가 0이 되면 서비스가 중지된다. 이는 기존의 BSW 스케줄링 알고리즘이 실제의 가중치 값이 아닌 계산된 BinaryProduct 값까지 서비스하는 것과 다르다. 그런데 실제의 가중치가 0이 되었는에도 다음 서비스 시간에서 보낼 셀이 있는 경우 한 개의 셀을 서비스 받고 loan 값을 증가시키게 된다. 그리고, 방문된 큐에 셀이 없는 경우에는 가장 큰 가중치를 가진 VC를 검색하여 검색된 VC의 한 개의 셀을 전송하고 그 VC의 loan 값을 증가시킨다. 이처럼 가중치가 0이 되었을 때 한 VC가 전송할 셀이 있다고 해서 가중치를 빌려서 셀을 전송하고 어떤 VC는 가중치가 0이 아님에도 불구하고 큐에 셀이 없다고 해서 다른 VC로 서비스 기회가 넘어 간다면 공정성에 대한 문제가 생기게 된다. 이를 해결하기 위해서 변수 load를 사용했는데 방문된 wheel의 VC의 lend 값이 0을 초과했을 경우 그 VC에 대해서는 서비스를 수행하지 않고 대신에 loan 값을 감소시키고 그 wheel에서 lend 값이 가장 큰 VC를 찾아 서비스를 하게 되는데 가장 큰 lend 값을 가진 VC일지라도 큐가 비어 있다면 그 다음으로 큰 lend 값을 가진 VC를 찾아 서비스가 이루어지게 되고 해당 VC의 lend 값을 감소시킨다. 실제로 binary rate list에 할당된 VC들은 자신의 가중치보다 더 많은 서비스를 받게 되는데 이렇게 더 할당된 가중치를 각 VC의 큐의 상태를 파악하지 않고 할당하는 기존의 BSW 스케줄링 알고리즘은 각 큐의 셀 지연 시간이 커지게 되고 버퍼에 차있는 셀의 수도 많아지게 되는데 제안된 WRR 스케줄링 알고리즘을 사용하여 셀 지연과 버퍼 관리의 측면에서 보다 우수하다는 것을 다음

장에서 시뮬레이션 결과를 통해서 확인하게 된다.

입력 부하에 따른 VC 개수와 할당 대역을 보여주고 있다.

Proposed Cell Scheduling Algorithm

While(all scheduling wheels on slot CurrentSlot is served)

```

If(loan in selected VC i queue > 0)
    loan = loan - 1;
    Search max lend VC queue;
    While(searched VC queue is empty)
        Search next max lend VC queue;
    end;
    if(lend in searched VC queue > 0)
        Serve one cell in searched VC queue;
        lend = lend - 1;
    else
        Serve one cell in selected VC i queue;
         $W_i = W_i - 1$ ;
    end;
end;
else
    If(selected VC i queue is not empty)
        if(  $W_i$  is not zero)
            Serve one cell in VC i queue;
             $W_i = W_i - 1$ ;
        else
            Search max  $W$  VC queue;
            Serve one cell in searched VC queue;
             $W = W - 1$ ;
        end;
    else
        lend  $i = lend i + 1$ ;
        Search max length VC queue;
        Serve one cell in VC i queue;
        loan  $i = loan i + 1$ ;
    end;
end;

```

If(W in all VC is zero)

```

Initialize W;
end;

```

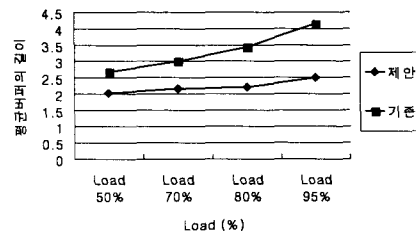
IV. 성능분석

시뮬레이션 환경에서 출력 링크 용량은 130M, 각 VC의 셀 도착 시간은 Poisson 분포를 따르고 각 VC에 할당된 대역과 VC 개수는 입력 부하에 따라 변화를 주었다. 아래의 표 1은

[표 1] 각 부하와 대역에 따른 VC의 개수

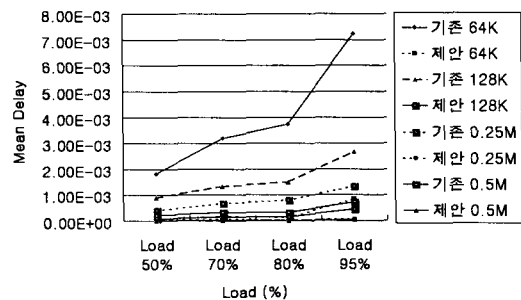
대역 \ 부하	50%	70%	80%	95%
8M	0	4	4	4
4M	4	4	4	8
2M	8	8	8	8
1M	16	8	16	16
0.5M	16	16	16	16
0.25M	16	16	16	16
128K	16	16	16	16
64K	32	32	32	32

그림 3은 입력 부하에 따른 평균 버퍼의 길이를 보여주고 있다. 그림에서 보는 바와 같이 제안된 알고리즘은 기존의 BSW 알고리즘보다 평균 버퍼의 길이가 작음을 알 수 있다. 제안된 알고리즘에서는 버퍼가 비어있을 경우 셀이 가장 많이 저장된 버퍼를 서비스하기 때문에 전체적인 평균 버퍼의 길이가 감소함을 확인할 수 있다.



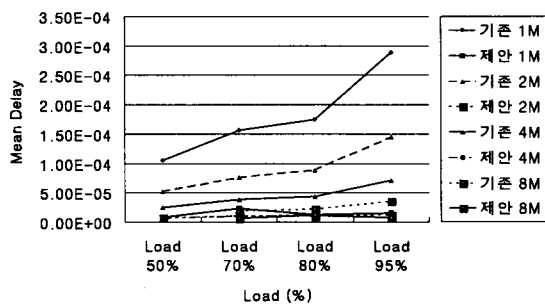
▶▶ 그림 3. 평균 버퍼의 길이

그림 4는 전송률이 낮은 VC에 대해서 입력 부하에 따른 평균 셀 지연을 보여주고 있다. 제안된 알고리즘이 기존 BSW 알고리즘보다 평균 셀 지연이 현저히 감소함을 알 수 있다. 이는 다른 VC의 버퍼가 비어 있거나 VC가 할당된 서비스를 다 받은 경우에 셀이 많이 쌓인 버퍼는 다른 VC에 비해 서비스를 받을 확률이 높아짐으로 전체적인 셀 지연이 감소하는 결과를 가져오기 때문이다.



▶▶ 그림 4. 낮은 전송률의 평균 셀 지연

그림 5는 전송률이 높은 VC에 대해서 입력 부하에 따른 평균 셀 지연을 보여 주고 있다. Binary rate list 구조의 특성으로 인해 전송률이 높은 VC가 상대적으로 전송률이 낮은 VC에 비해서 먼저 서비스가 되고, 가중치를 초과한 경우 전송률이 낮은 VC보다 더 많은 서비스 기회로 인해서 가중치를 빌려올 수 있는 확률이 커지게 되어 전송률이 낮은 VC에 비해서 셀 평균 지연이 더 작음을 알 수 있다.



▶▶ 그림 5. 높은 전송률의 평균 셀 지연

IV. 결론

기존의 BSW 알고리즘은 2의 지수승에 맞춘 서비스 율로 인해서 각 VC는 실제로 할당된 가중치보다 더 큰 가중치를 얻게 되고, VC의 큐가 비어 있을 경우 서비스를 수행하지 못하여 대역을 낭비하게 되므로 셀 지연 및 전송률에서 성능을 저하시키는데 비하여, 본 논문에서는 이를 개선한 셀 스케줄링 알고리즘을 제안하였다. 제안된 알고리즘은 전체적으로 입력 VC의 부하에 관계없이 기존의 BSW 알고리즘보다 평균 버퍼의 길이가 작아짐을 보여주고 평균 셀 지연에 대해서도 향상된 결과를 나타내고 있다. 특히 평균 버퍼의 길이는 과부하 상태에서 제안된 알고리즘이 좀 더 탁월한 성능 향상을 나타내고, 기존의 BSW 알고리즘은 부하가 적은 경우에 전송률이 작은 VC일수록 더욱 더 큰 지연을 갖게 되는데 제안된 알고리즘에서는 전송률이 작은 VC에 대해서도 탁월한 성능 향상을 보여주고 있다. 제안된 알고리즘은 기존의 BSW 스케줄링 구조에서 셀 지연 및 버퍼의 크기를 개선함으로써 각 VC의 QoS를 만족시킬 수가 있다.

■ 참고 문헌 ■

- [1] G. Anastasi, M. Conti, E. Gregori, A. Passarella, "A performance study of power-saving policies for Wi-Fi hotspots", *Computer Networks* 45 (2004) 295-318.
- [2] V. Baiamonte, C.-F. Chiasserini, An energy-efficient MAC layer scheme for 802.11-based WLANs, in: Proc. of

the 2004 IEEE Int. Conf. on Performance, Computing and Communications, 2004, pp.689-694.

- [3] Q. Ni, L. Romdhani, T. Turetli, "A survey of QoS enhancements for IEEE 802.11 wireless LAN", *Wireless Communications and Mobile Computing* 4 (2004) 547-566.
- [4] S. Archambault and J. Yan, "Performance Analysis of Per-VC Queueing," Proc. IEEE GLOBECOM'96, Vol.3, pp.1721-1725, November 1996.
- [5] Shimonishi H, Suzuki H, "Performance Analysis of Weighted round robin Cell Scheduling and its Improvement in ATM Networks," *IEICE Transactions on Communications*, V.E81-B No.5, May, 1998.
- [6] 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.