

論文2002-39TC-12-4

멀티미디어 트래픽에 대한 무선 환경에서의 순방향 패킷 스케줄링 알고리즘

(A Wireless Downlink Packet Scheduling Algorithm for Multimedia Traffic)

金 東 會 * , 柳 炳 漢 *

(Dong Hoi Kim and Byung Han Ryu)

요 약

본 논문에서의 패킷 스케줄러는 무선 채널 환경에서 실시간 비디오 트래픽과 비실시간 인터넷 트래픽을 동시에 서비스한다. 그리고 실시간 트래픽에 대한 지연을 감소 시키기 위하여 누적카운터와 SIR값을 동시에 고려하는 새로운 패킷 스케줄링 알고리즘을 제안한다. 본 알고리즘에서는 실시간 트래픽을 비실시간 트래픽에 비하여 우선적으로 처리한다. 시뮬레이션 결과를 고찰할 때, 실시간 비디오 트래픽과 같이 지연성능에 민감한 서비스의 경우에는 AC(Accumulation Counter) 방식이 기존의 M-LWDF방식보다 더 적합한 알고리즘으로 관찰되었다. 본 논문에서는 HSDPA 시스템과 유사한 구조를 채택하여 시간 축으로는 프레임 주기를 가지며 동시에 코드 축으로는 OVSF 코드를 사용할 수 있는 구조를 채택하여 모의 실험을 수행하였다.

Abstract

In this paper, we consider a wireless multimedia environment to service both real-time video traffic and non-real-time WWW-application traffic. In our suggested new packet scheduling algorithm, we consider both the accumulation counter and SIR to reduce delay in real-time traffic. In addition, our packet scheduling algorithm gives priority first to real-time video traffic service and then to non-real-time internet packet service when real-time traffic service is absent. From the simulation results, we find that the AC (Accumulation Counter) scheme has much smaller delay than the conversional M-LWDF scheme for real-time video data users, which has a special quality sensitive to its own delay. We also consider the transmission structure of using both the frame period in the time-axis and the OVSF codes in the code-axis at the same time, which is similar to the structure of HSDPA system.

Keywords : Packet scheduling algorithm, Multimedia traffic, Buffering window, Accumulation counter, Signal to Interference Ratio

* 正會員, 韓國電子通信研究院 移動通信研究所

(Mobile Traffic Engineering Team, Mobile Telecommunication Research Laboratory, Electronics and Telecommunications Research Institute, Korea)

接受日字:2002年10月22日, 수정완료일:2002年12月3日

I. 서 론

최근에 초고속 패킷 전송을 위하여 3GPP에서는 HSDPA(High Speed Downlink Packet Access) 그리고 3GPP2에서는 1xEV-DV라는 시스템을 제안하였고 특히

HSDPA 시스템은 최고 2 Mbps의 서비스 속도를 목표로 하였던 IMT-2000의 3세대 시스템과 비교하여 최고 10 Mbps의 전송속도를 목표로 하고 있다. HSDPA 시스템은 IMT-2000 3세대의 WCDMA 서비스와 동일한 반송파를 사용하여 공존하는 것을 가정하므로 사용자 전용 물리채널 (Dedicated Physical CHannel, DPCH) 사용자와 공존한다. 또한 IMT-2000 3G 시스템의 순방향 공통채널 (Downlink Shared CHannel, DSCH)를 강화한 고속 순방향 공통채널 (High Speed Downlink Shared CHannel, HS-DSCH)을 사용한다^[1, 2].

하향링크에서의 패킷 스케줄러는 Node-B에 위치하며 기존의 ARQ 기법에 전진 오류 정정(forward error correction)기법을 복합하여 combining의 효과를 가지는 복합 자동 재전송 요구 (Hybrid Automatic Repeat reQuest, H-ARQ) 기능과 연동하고 링크 상황에 따라서 변복조 방식과 코딩 방식을 달리하는 기술인 적응변조 및 코딩 (Adaptive Modulation and Coding, AMC)기능을 제어하면서 동작한다^[1~5].

새로운 하향링크 패킷 스케줄링 알고리즘은 다른 QoS를 가지는 멀티미디어 트래픽을 지원해야 한다. 또한 제한된 자원을 사용하면서 실시간 패킷 트래픽과 비실시간 패킷 트래픽을 동시에 서비스하면서도 실시간 트래픽에 대해서는 지연을 감소시키고 비실시간 트래픽 사용자들에게는 처리량을 증가시키는 효율적인 알고리즘이 필요하다. 본 논문에서는 무선 채널 환경에서 실시간 비디오 트래픽 서비스와 비실시간 인터넷 트래픽 서비스를 동시에 지원하면서 지연을 극소화하는 방법을 제시 하였다. 제안한 패킷 스케줄링에서는 HSDPA 시스템과 유사하게 시간 축으로는 2 ms의 프레임울 가지고 동시에 코드 축으로는 16개의 확산 코드 (Spreading Factor, SF)중에서 12개의 코드를 사용할 수 있는 구조에서 사용자수는 12명까지 코드 다중화할 수 있다고 가정 한 후 모의 실험을 수행하였다^[4, 5].

본 논문의 나머지는 다음과 같이 구성한다. 2장에서는 실시간과 비실시간 트래픽 발생 모델을 살펴본다. 3장에서는 기존의 패킷 스케줄링 알고리즘과 제안하는 새로운 알고리즘에 대한 시스템 모델링을 한다. 4장과 5장에서는 제안하는 알고리즘에 대한 자세한 설명에 그에 따른 시뮬레이션 결과를 기술한다. 마지막으로 6장에서 본 논문의 결론을 기술한다.

II. 트래픽 모델

1. 실시간 비디오 트래픽

<그림 1>은 실시간 비디오 트래픽 모델을 보여주고 있다. 일정한 간격 T는 고정된 프레임의 주기로 정해진다. 본 논문에서는 일정한 간격 T를 버퍼링 윈도우 주기로 정의하였고 시간 제한을 가지는 실시간 트래픽에 대한 지연 성능의 기준으로 사용하였다. <그림 1>에서 보는 바와 같이 각각의 프레임은 고정된 개수의 패킷들로 구성되고 매번 비디오 화면이 변화하므로 각각의 패킷들의 크기는 버스트(burst) 특성을 가지게 된다. 이러한 패킷 크기와 하나의 프레임내에 있는 패킷들의 상호 도착 시간은 절단된 파레토(truncated pareto) 분포로 정의 하였다. 32 kbps의 속도를 가지는 소스 비디오 트래픽에 대한 상세한 모델 파라메타는 <표 1>에 기술되었다^[6].

2. 비실시간 인터넷 트래픽

비실시간 인터넷 트래픽 모델은 인터넷 사용의 시간 간격에 의해서 정해지는 세션(session)들로 구성되었고

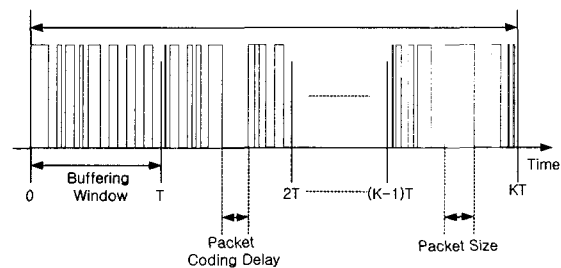


그림 1. 실시간 비디오 트래픽을 위한 트래픽 모델
Fig. 1. Traffic model for real-time streaming video.

표 1. 비디오 스트리밍 트래픽 모델 파라메타
Table 1. Video Streaming Traffic Model Parameters.

Inter-arrival time between the beginning of each frame	Deterministic (Based on 10fps)	100 ms
Number of packets in a frame	Deterministic	8
Packet size	Truncated Pareto (Mean=50bytes, Max=125bytes)	k=20bytes, (=1.2)
Inter-arrival time between packets in a frame	Truncated Pareto (Mean=6ms, Max=12.5ms)	k=2.5ms, (=1.2)

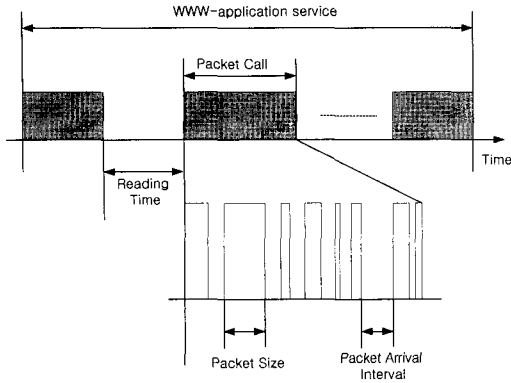


그림 2. 비실시간 패킷 서비스 세션을 위한 트래픽 모델
Fig. 2. Traffic model for non-real-time packet service session.

표 2. WWW 응용 트래픽 모델에 대한 파라메타
Table 2. WWW-application traffic Model Parameters

Information type	Distribution	Distribution Parameters
Number of packet calls per Session	Exponential	5
Reading time between packets calls	Exponential	412
Number of datagrams within a packet call	Exponential	25
Inter-arrival time between datagrams (within a packet call)	Exponential	0.0277
Size of a datagram	Truncated Pareto (Max=125bytes)	k=81.5 byte, (=1.1)

각각의 세션은 여러 개의 패킷 호(packet call)로 이루어진다. 또한 각각의 패킷 호는 버스트(burst) 특성을 가지는 여러 개의 패킷으로 구성되어 있다. 위와 같이 많은 버스트한 패킷이 모여 한 개의 패킷 호를 구성하는 것이 비실시간 트래픽의 중요한 현상이다.

WWW 응용 서비스에서 패킷 크기는 파레토 분포로 모델링이 될 수 있다. 각각의 패킷과 패킷 호의 개수는 지수 분포를 가진다. 패킷들의 상호 도착시간과 패킷 호 사이의 독서 시간(reading time)은 지수 분포를 가진다고 정의하였다. WWW 응용 트래픽 모델에 대한 파라메타는 <표 2>에 기술되었다^[7].

III. 시스템 모델링

1. 기존 알고리즘 분석

현재 적용되고 있는 패킷 스케줄링 알고리즘은 여러

가지가 있지만 본 논문에서는 M-LWDF(Modified Largest Weighted Delay First)와 제안하는 새로운 스케줄링 알고리즘에 대해 알아 보기로 한다.

M-LWDF 알고리즘에서는 사용자 i 의 지연 QoS(Quality of Service) 요구사항을 다음과 같이 정의하고 있다^[8].

$$\Pr\{W_i > T_i\} \leq \delta_i \tag{1}$$

사용자가 실시간 패킷 사용자일때, 데이터 패킷의 대부분의 지연은 특정 기준값 이하로 유지되어야 할 필요가 있다. 여기에서 W_i 는 사용자 i 에 대한 패킷 지연이고, 파라메타 T_i 와 δ_i 는 지연 기준값과 QoS를 만족하기 위한 허용 가능한 최대 확률값이다. M-LWDF 알고리즘은 다음 식의 결과값이 가장 큰 이용자에게 우선적으로 서비스를 수행하게 된다.

$$s = \arg \max_i \left(a_i W_i \frac{r_i(t)}{\bar{r}_i} \right) \tag{2}$$

여기에서, 파라메타 a_i 는 $-(\log \delta_i)/T_i$ 이고 $r_i(t)$ 와 \bar{r}_i 는 각각 사용자 i 에 관한 현재 시간 t 에서의 채널 전송률과 평균 채널 전송률이다. 모든 사용자들이 동일한 지연값 T_i 를 가지고 초과 확률 δ_i 가 동일하다면 식 (2)는 다음과 같이 재정의된다.

$$s = \arg \max_i \left(W_i \frac{r_i(t)}{\bar{r}_i} \right) \tag{3}$$

이는 비례 공정 알고리즘에 패킷 전송 버퍼에서의 지연 W_i 가 곱해진 형태와 유사하다.

본 논문에서는 제안하는 패킷 스케줄링과 M-LWDF와의 성능비교를 하기 위하여 지연 QoS 요구사항을 다음과 같이 재정의하고 있다

$$\Pr\{D_k^i > D_i^i\} \leq \gamma_i \tag{4}$$

여기에서, D_k^i 는 사용자 i 에 대한 k 번째 버퍼링 윈도우에 도착한 패킷중에서 마지막으로 도착한 패킷이 k 번째 버퍼링 윈도우를 벗어나서 처리될 때의 지연값이고 파라메타 D_i^i 와 γ_i 는 각각 버퍼링 윈도우의 지연 기준값과 QoS를 만족하기 위한 허용 가능한 최대 확률

값이다.

본 논문에서 제안하는 패킷 스케줄링 알고리즘에서는 실시간 비디오 트래픽의 경우에는 누적 카운터($AC(t)$)와 $r_i(t)$ 그리고 \bar{r}_i 를 기준으로 식 (5)의 결과값이 가장 큰 이용자를 우선적으로 서비스하게 된다.

$$s = \arg \max_i \left(AC_i(t) \frac{r_i(t)}{\bar{r}_i} \right) \quad (5)$$

또한 비실시간 인터넷 트래픽의 경우에는 $r_i(t)$ 와 \bar{r}_i 를 기준으로 식 (6)의 결과값에 따라서 값이 큰 이용자의 순서대로 서비스를 수행하게 된다.

$$s = \arg \max_i \left(\frac{r_i(t)}{\bar{r}_i} \right) \quad (6)$$

여기에서 $AC_i(t)$ 는 사용자 i 의 $AC(t)$ 를 표기하고 $r_i(t)$ 는 각각 사용자 i 에 대하여 갱신 시간 t 에서의 채널 전송률이고 \bar{r}_i 는 평균 채널 전송률이다.

2. 누적카운터의 정의

<그림 3>은 스케줄링 갱신 시간 t 에서 $AC_i(t)$ 를 계산하는 방법을 보여주고 있다. <그림 3>에서 T_i^k 는 사용자 i 의 k 번째 버퍼링 윈도우 간격을 나타내고 β_j^i 는 사용자 i 에 대한 j 번째 패킷의 크기이다. $AC_i(t)$ 는 사용자 i 에 대하여 T_i^k 의 시작 시간부터 스케줄링 갱신 시간 t 까지의 패킷 크기의 합으로 다음과 같이 표시된다.

$$AC_i(t) = \sum_{j=1}^M \beta_j^i \quad (7)$$

T_i^k 시간내에서 마지막으로 도착한 패킷이 서비스가 완료되었을 때 $AC_i(t)$ 는 리셋이되고 그리고 T_i^k 후에 도착된 패킷들의 크기의 합으로 다시 정의된다. 만약 식 (5)에서 $\frac{r_i(t)}{\bar{r}_i}$ 의 값이 같다면 <그림 3>에서 보는 것과 같이 사용자 i 에 대한 $AC_i(t)$ 는 그밖에 다른 사용자들의 $AC_j(t)$ ($j=1,2,\dots,n, j \neq i$)보다 크다면 제안한 패킷 스케줄링 알고리즘은 사용자 i 에 대한 HOL(Head Of Line) 패킷을 먼저 서비스하게 된다.

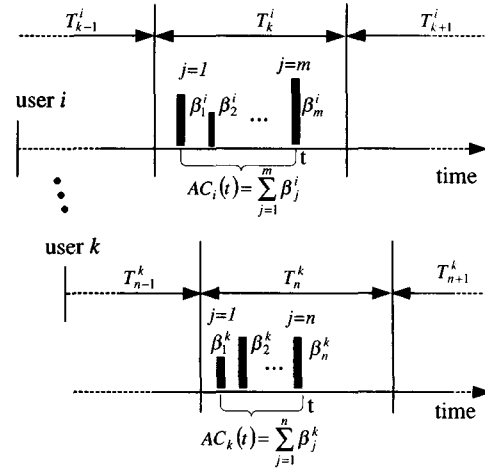


그림 3. 실시간 비디오 트래픽에 대한 누적 카운터와 스케줄링 갱신 시간

Fig. 3. Accumulation counter and scheduling instance for streaming video traffic.

3. 버퍼링 윈도우 지연의 정의

<그림 4>에서 보는 바와 같이 D_k^i 는 사용자 i 에 대한 k 번째 버퍼링 윈도우의 지연을 나타낸다. D_k^i 는 T_i^k 주기내에서 마지막 도착한 패킷이 T_i^k 주기가 끝난 후에 서비스되는 시간차로 정의가 하였다. 만약 T_i^k 주기내에서 마지막 도착한 패킷이 T_i^k 주기내에 서비스된다면 그때의 D_k^i 값은 0가 된다.

D_k^i 값을 계산하기 위하여는 T_i^k 주기내의 마지막 도착한 패킷들과 T_i^k 주기후에 새롭게 도착한 패킷들과의 구별이 필요하게 된다. D_{tot}^i 는 사용자 i 의 세션동안에 시작하여 종료될 때까지의 모든 버퍼링 윈도우들의 지연 값들을 합한 값으로 표시되고 D_{av}^i 는 D_{tot}^i 값을 사용자 i 의 세션동안에 발생한 모든 버퍼링 윈도우들의 총 개수로 나눈 값으로 정의한다. 따라서 전체 실시간 비디오 사용자의 버퍼링 윈도우 평균 지연인 D_{av} 값은 $\sum_{i=1}^C D_{av}^i$ 를 전체 실시간 비디오 사용자 수인 C 로 나눈 값이 된다.

제안하는 패킷 스케줄링에서는 여러 사용자중에서 어느 사용자를 먼저 서비스할지를 결정할 때 $AC_i(t)$ 의 값이 중요한 기준이 된다. 따라서 여러 사용자중에서 식 (5)의 $AC_i(t) \times \frac{r_i(t)}{\bar{r}_i}$ 가 가장 큰 사용자를 먼저 서비스함

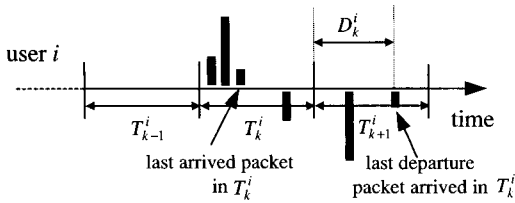


그림 4. 사용자 i 의 비디오 스트리밍 세션에 대한 k 번째 버퍼링 윈도우의 지연
 Fig. 4. Delay of k -th buffering window for video streaming session of user i .

으로 D_{av}^i 값이 기존의 다른 알고리즘보다 작아지는 효과를 가져올 수가 있다.

IV. 제안하는 패킷 스케줄링 알고리즘

본 논문에서 사용한 순방향 패킷 스케줄링 알고리즘에서는 실시간 비디오 트래픽 서비스의 경우에는 식 (5)를 기준으로 스케줄링 순서를 결정하였으며 비실시간 트래픽 서비스의 경우에는 식 (6)을 기준으로 스케줄링을 하였다.

소스 트래픽중에서 실시간 비디오 트래픽은 2.1절에서 설명한 실시간 비디오 트래픽 모델로 사용자 수만큼 병렬로 발생되어 각각의 사용자별로 구분된 데이터 버퍼에 입력되어 쌓이게 된다. 또한 비실시간 인터넷 트래픽도 2.2절에서 기술한 비실시간 인터넷 트래픽 모델링으로 발생하면서 사용자 수만큼 병렬로 데이터 버퍼에 저장되어 증가하게 된다.

제안된 패킷 스케줄링 알고리즘은 일차적으로 실시간 비디오 패킷 트래픽 서비스에 우선권을 주어 먼저 서비스하고 서비스할 실시간 비디오 패킷 트래픽이 없을 경우에만 비실시간 WWW 트래픽을 서비스하게 된다.

<그림 5>는 제안하는 패킷 스케줄러의 단순한 구성도를 설명하는 그림이다. m 명의 사용자가 실시간 데이터 버퍼에 쌓이게 되고 그와 동시에 n 명의 사용자가 비실시간 데이터 버퍼에 저장되는 것을 가정할 수가 있다. <그림 5>와 같이 데이터 버퍼에 누적된 패킷들은 <그림 6>의 순서도에서 설명된 것처럼 동작하게 된다.

다음은 본 논문에서 사용한 표기들에 대한 설명이다. OVSF 코드의 수는 N_{OVSF} 으로 표기가 된다. N_{OVSF} 는 HSDPA의 경우처럼 최대 12까지 설정되었다. 또한 $S_i(t)$ 는 스케줄링 갱신 시간 t 이전에 처리되지 않고

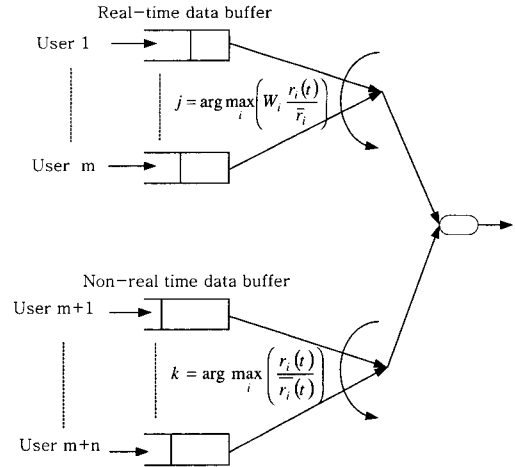


그림 5. 제안된 패킷 스케줄러의 단순한 구성도
 Fig. 5. The simple structural diagram for the proposed packet scheduler.

남은 패킷의 크기로 정의하고 $C_i(t)$ 는 단말로부터 수신된 채널 정보인 SIR를 수신하여 그 값에 따라서 결정되는 전송 방법에 의한 전송 데이터 크기를 나타낸다. 따라서 기지국에서 단말로 전송할 데이터 크기는 단말마다 그리고 매 프레임마다 달라질 수가 있게 된다.

<그림 6>에서 실시간 트래픽의 경우에 사용자 i 의 HOL (Head Of Line) 패킷들에 대해 식 (5)를 계산하여 가장 큰 값을 가지는 사용자 j 에 대해 서비스를 수행하게 되는데 스케줄링 작업중에 N_{OVSF} 가 12보다 큰 경우에는 무조건 스케줄링 작업은 종료하게 된다.

N_{OVSF} 가 12보다 작고 $S_i(t)$ 가 $C_i(t)$ 보다 크다면 사용자 j 에 대해서 $C_i(t)$ 만큼 서비스를 수행하고 N_{OVSF} 의 값도 1씩 증가시킨다. 그러다가 $S_i(t)$ 가 $C_i(t)$ 보다 작고 N_{OVSF} 도 12보다 작은 경우에도 사용자 j 에 대해서 $S_i(t)$ 만큼 서비스를 수행하고 N_{OVSF} 의 값도 1씩 증가시킨 후에 이미 서비스된 사용자 i 를 제외한 다른 사용자들의 패킷들에 대해 식 (5)를 다시 계산하여 계산된 값 중에서 가장 큰 값을 가지는 사용자에게 서비스를 재개한다. 위와 같은 방법으로 패킷 스케줄링을 수행하다가 더 이상 서비스할 실시간 비디오 데이터가 없을 경우에 비실시간 트래픽 서비스들에 대한 스케줄링을 시작하게 된다. 비실시간 서비스의 경우에는 여러 개의 데이터 버퍼의 HOL 패킷들에 대해 식 (6)을 기준으로 계산된 값 중에서 가장 큰 값을 가지는 사용자부터 서비스를 수행하게 된다. 그 밖의 자세한 수행동작은 실시간 패킷 서비스의 수행동작과 유사한다.

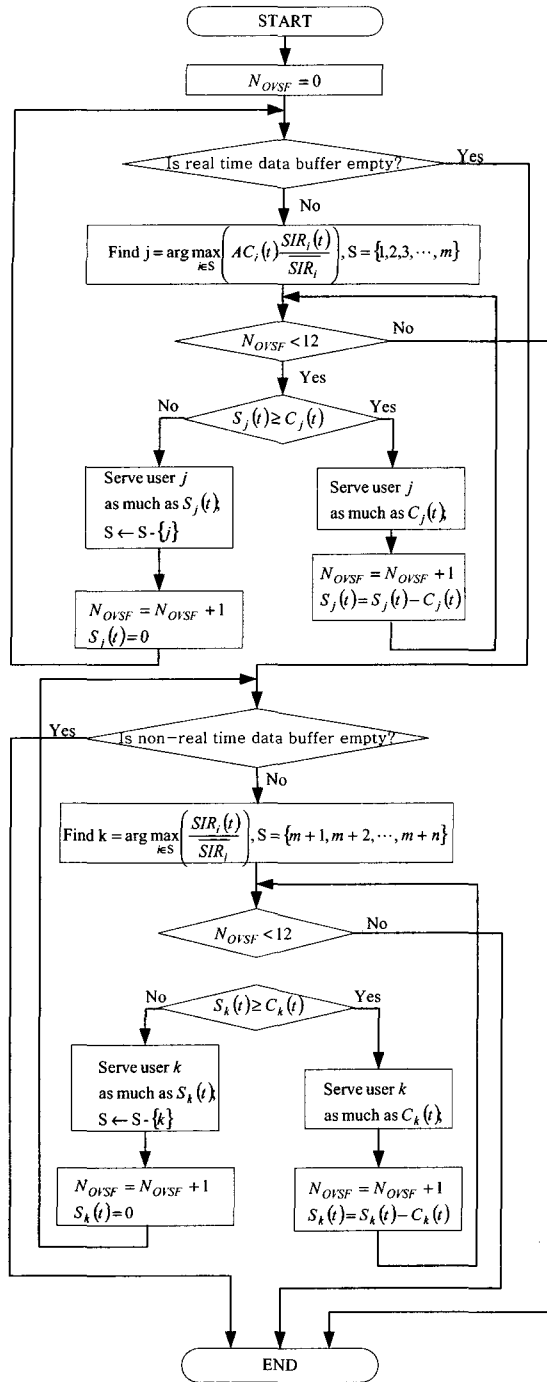


그림 6. 제안된 패킷 스케줄링 알고리즘의 순서도
Fig. 6. Flowchart of the proposed packet scheduling algorithm.

본 논문에서는 단말에서 기지국으로 수신된 SIR값은 패킷 스케줄링을 수행하게 하는 기준자료로도 사용되기도 하지만 물리채널에서의 변조 방법과 인코딩 방법을 결정짓는 잣대로도 사용된다.

제안된 패킷 스케줄링 알고리즘은 모든 사용자들에 대한 패킷들의 지연을 계산하기 위한 시간 스텐트와 누적 카운터만을 필요로 하기 때문에 구현하기가 쉬운 구조로 되어 있다.

V. 시뮬레이션 결과

SIR 값은 시뮬레이션을 위하여 실제 무선 채널 조건을 통해 발생되지 않고 임의로 발생시켰다. 그리고 각각의 호 발생시간은 포아송으로 발생하였다.

셀 내부에 존재하는 모든 단말은 Node-B에 SIR값을 보고한다. 보고된 SIR 값은 패킷 스케줄러에서 이용될 뿐만 아니라 <표 4>에서와 같이 해당되는 특정한 전송 방식(예로써, 변조 방식과 코딩율)을 선택하게 된다.

본 시뮬레이션 환경에서는 지연특성에 민감한 특성을 가지는 실시간 비디오 트래픽의 경우를 관찰하기 위하여 AC 방식과 기존에 제안된 M-LWDF 방식과의 지연 성능을 비교 분석하였다.

<그림 7>의 관찰된 결과는 비실시간 WWW 사용자의 수는 10명으로 고정하였고 시뮬레이션 시간은 100초까지의 결과를 살펴보았다.

표 3. 시뮬레이션 파라메타
Table 3. Simulation parameter.

파라메타	값
칩 속도	3.84 Mcps
확산코드	16
N_{OVSF}	12
T_k^i	100 ms
스케줄링 갱신 시간	2 ms

표 4. 시뮬레이션을 위해 사용된 물리 채널 전송 모드

Table 4. Physical transmission mode used for simulation.

SIR (dB)	Modulation Scheme	Coding Rate	Transmission bits per one frame
9~7	64 QAM	3/4	2160
7~3	16 QAM	3/4	1440
3~1	16 QAM	2/4	960
1~3	QPSK	3/4	720
-3~-7	QPSK	2/4	480
-7~-9	QPSK	1/4	240

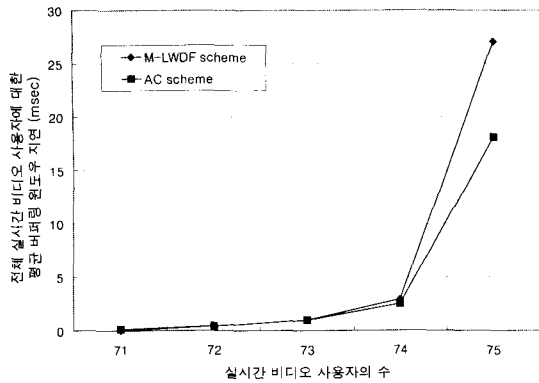


그림 7. AC 방식과 M-LWDF 방식에 대한 전체 실시간 비디오 사용자의 평균 버퍼링 윈도우 지연
Fig. 7. D_{av} for AC scheme and M-LWDF scheme.

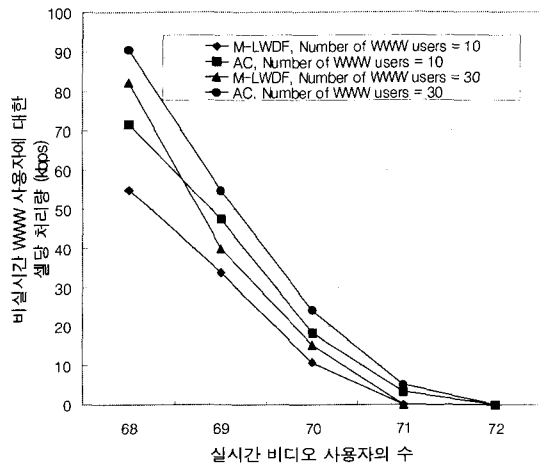


그림 8. 실시간 비디오 사용자와 비 실시간 WWW 사용자에게 대한 셀당 처리량
Fig. 8. The throughput per cell for non-real-time WWW data user and the number of video data users.

<그림 7>에서 볼 때, 실시간 비디오 사용자가 증가함에 따라 실시간 비디오 사용자의 평균 버퍼링 윈도우 지연이 증가함을 알 수 있다. 위 결과를 고려해 볼 때, AC 방식은 M-LWDF 방식보다는 실시간 사용자의 수가 과부하(overload)되는 상황에서는 지연성능이 훨씬 더 우수한 것으로 관찰되었다. 따라서 AC 방식은 실시간 비디오 트래픽과 같이 지연성능에 민감한 서비스의 경우에는 본 논문에서 제안한 방식이 더 적합한 알고리즘이라고 할 수가 있겠다. 또한 실시간 비디오의 사용자가 75명 이상인 경우의 평균 버퍼링 윈도우 지연값은 100 msec를 가지는 프레임의 주기보다 훨씬 큰 값을 가

짐을 알 수가 있었다.

<그림 8>은 실시간 비디오 사용자의 증가와 셀 당 처리량의 관계를 보여주고 있다. 이 경우에도 앞의 결과와 같이 시뮬레이션 시간을 100초 수행하였다. 전체적으로 실시간 비디오 사용자의 수가 증가함에 따라 비실시간 WWW 사용자의 수에 대한 셀 당 처리량은 감소함을 알 수가 있다.

<그림 8>에서 WWW 사용자에게 대한 셀당 처리량이 0이 되는 경우에 실시간 사용자의 수를 살펴보면 M-LWDF 방식에서는 71명이고 AC 방식에서는 72명이 된다. 또한 실시간 비디오 사용자당 서비스되는 비실시간 WWW 사용자에게 대한 셀당 처리량도 AC 방식이 M-LWDF방식일 때보다 증가함을 보여주고 있다.

VI. 결 론

본 논문에서는 무선 채널 환경에서 실시간 비디오 트래픽과 비실시간 인터넷 트래픽을 동시에 서비스하면서 실시간 트래픽에 대해서는 지연을 감소시키면서도 처리량이 작아지지 않게 하기 위하여 누적카운터와 SIR값을 동시에 고려하는 새로운 패킷 스케줄링 알고리즘을 제안하였다. 또한 실시간 트래픽에 대해 패킷 스케줄링을 수행하다가 더 이상 서비스할 실시간 비디오 데이터가 없을 경우에 비실시간 트래픽 서비스들에 대한 스케줄링을 시작하도록 하였다.

시뮬레이션 결과로 볼 때, 실시간 비디오 트래픽과 같이 지연성능에 민감한 서비스의 경우에는 제안하는 AC 방식이 기존의 M-LWDF방식보다 더 적합하고 또한 실시간 비디오 사용자당 서비스되는 비실시간 WWW 사용자에게 대한 셀당 처리량도 AC 방식이 M-LWDF방식일 때보다 증가함을 알 수가 있다.

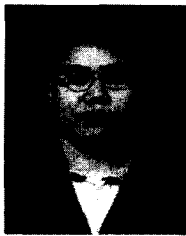
향후에는, 호 수락 제어(Call Admission Control, CAC)와 HARQ와의 관계를 고려한 전체 시스템 성능 분석이 필요하다.

참 고 문 헌

[1] 3GPP technical Specification 25.338 V5.1.0, High Speed Downlink Packet Access(HSDPA), Overall Description, Stage 2 (2001).
[2] 3GPP "Technical Specification Group Radio Access Network: High Speed Downlink Packet

- Access: Overall Description (Release 5),” TR 25.855 V5.0.0, 2001.
- [3] 3GPP “Technical Specification Group Radio Access Network: UTRA High Speed Downlink Packet Access (HSDPA): Overall description; Stage 2, (Release 5),” TR 25.308 V5.2.0, 2002.
- [4] 3GPP, “Technical Specification Group Radio Access Network: Physical Layer Aspects of UTRA High Speed Downlink Packet Access (Release 4),” TR 25.848, V4.0.0, Mar. 2001.
- [5] 3GPP, “Technical Specification Group Radio Access Network: High Speed Downlink Packet Access: Physical Layer Aspects (Release 5),” TR 25.848, V5.0.0, Mar. 2002.
- [6] 1xEV-DV Evaluation Methodology - Addendum (V6), WG5 Evaluation AHG, July 25, 2001.
- [7] J. P. Castro, The UMTS Network and Radio Access Technology, John Wiley & Sons, Inc., New York, NY, (2001).
- [8] Matthew Andrew et al., “Providing Quality of Service over a Shared Wireless Link”, IEEE Comm. Mag., Feb 20

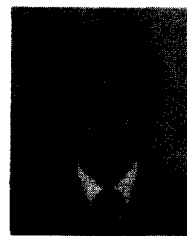
 저 자 소 개



金東會(正會員)

1989년 2월 : 한양대학교 전자통신 공학과 졸업(공학사). 1999년 2월 : 한양대학교 대학원 전자통신공학과 졸업(공학석사). 1989년 1월~1997년 2월 : 삼성전자 전임연구원. 2000년 8월~현재 : 한국전자통신연구원

이동통신연구소 선임연구원. <주관심분야 : 매체접근 제어 프로토콜, 광대역 무선 접속 망 및 4세대 이동통신 기술, 무선 자원 관리>



柳炳漢(正會員)

1985년 2월 : 한양대학교 산업공학과 졸업(공학사). 1988년 2월 : 서울대학교 대학원 산업공학과 졸업(공학석사). 1997년 3월 : 오사카대학교 대학원 기초공학연구과 정보공학과 졸업(공학박사). 1985년 1월~1986

년 1월 : (주)퍼시픽 콘트롤즈, 1988년 3월~현재 : 한국전자통신연구원 이동통신연구소 책임연구원/팀장. <주관심분야 : 트래픽 접속 프로토콜, 통신망, 트래픽 제어, 시스템 성능평가>