

저전력 환경에서 멀티미디어 자료 재생을 위한 디스크 스케줄링 기법

(Power Conscious Disk Scheduling for Multimedia Data Retrieval)

최정완[†] 원유집^{**} 정원민[†]
(Jungwan Choi) (Youjip Won) (Wonmin Jung)

요약 최근에, 휴대용 컴퓨터와 모바일 장치(Smart Phone, PDA, MP3 Player 등)의 보급 증가로 인하여 모바일 환경에서의 전력 관리 기술의 필요성은 매우 증가하고 있다. 왜냐하면, 전력 관리 기술은 모바일 장치에서 가장 중요한 요소 중 하나이기 때문이다. 한편, 하드 디스크는 가격에 비하여 용량이 크고 속도가 빠르며 최근에는 모바일 장치에서 사용하기에 충분히 작게 만들 수도 있으므로 모바일 기기에 사용하기 적당하나, 모바일 장치에 사용하기에는 소모 전력이 너무 크다는 단점이 있다. 이러한 이유들로 인하여 본 논문에서는 디스크에 저장된 연속적인 멀티미디어 자료를 실시간으로 재생하는데 소모되는 전력을 최소화하기 위한 방법을 제안하고 평가하였다. 모바일 장치에서 소모되는 전력을 엄격히 제한하는 것은 모바일 장치들의 하드웨어와 소프트웨어 디자인에 지대한 영향을 준다. 실시간 멀티미디어 재생이 기존의 텍스트 서비스와 다른 점은 디스크가 저장된 데이터 블록을 연속적으로 공급한다는 것이다. 이것은 디스크 드라이브가 멀티미디어 데이터 전체를 재생할 동안 활성화 상태를 유지해야 하게 하므로, 전력 관리 측면에서는 매우 큰 부담으로 작용한다. 전형적인 모바일 디스크 드라이브의 전력 관리 기능은 디스크가 대기 상태에 있을 때의 과도한 입출력 요청으로 인하여 멀티미디어 재생의 품질 측면에서는 부정적인 영향을 준다. 따라서, 본 논문에서는 디스크 드라이브의 전력 소모 형태를 세밀히 분석하고, 저장된 멀티미디어 데이터를 보다 효율적으로 전력을 덜 사용해 재생할 수 있는 알고리즘을 개발하였다. 이 알고리즘은 읽어야 할 데이터 블록의 수와 활성화/대기 상태의 길이를 계산한다. 이로써 본 논문에서 제안한 알고리즘은 모바일 디스크 드라이브에 저장된 데이터 블록의 연속적인 재생을 보장하는 최적의 스케줄링을 한다. 기존에 공개된 MPEG 재생기에 본 알고리즘을 적용한 멀티미디어 재생기를 만들었다. 이 재생기는 전체 재생동안 활성화 상태를 유지하는 경우와 비교하여 60%의 전력 소모량을 절약하였고, 전력 관리 모듈이 있는 디스크 드라이브에서 재생할 경우에 기존의 재생기와 비교하여 38%의 전력 소모량을 절약하였다.

키워드 : 디스크 스케줄링, 전력 관리, 멀티미디어, 재생, 모바일 장치

Abstract In the recent years, popularization of mobile devices such as Smart Phones, PDAs and MP3 Players causes rapid increasing necessity of power management technology because it is most essential factor of mobile devices. On the other hand, despite low price, hard disk has large capacity and high speed. Even it can be made small enough today, too. So it appropriates mobile devices, but it consumes too much power to embed in mobile devices. Due to these motivations, in this paper we had suggested methods of minimizing power consumption while playing multimedia data in the disk media for real-time and we evaluated what we had suggested. Strict limitation of power consumption of mobile devices has a big impact on designing both hardware and software. One difference between real-time multimedia streaming data and legacy text based data is requirement about continuity of data supply. This fact is why disk drive must persist in active state for the entire playback duration, from power management point of view; it may be a great burden. A legacy power management function of mobile disk drive affects quality of multimedia playback negatively because of excessive

[†] 학생회원 : 한양대학교 전자전기컴퓨터 공학부
chrys@ece.hanyang.ac.kr
wonmin82@ece.hanyang.ac.kr

^{**} 종신회원 : 한양대학교 전자전기컴퓨터 공학부 교수

yjwon@ece.hanyang.ac.kr

논문접수 : 2003년 6월 14일
심사완료 : 2005년 12월 7일

I/O requests when the disk is in standby state. Therefore, in this paper, we analyze power consumption profile of disk drive in detail, and we develop the algorithm which can play multimedia data effectively using less power. This algorithm calculates number of data block to be read and time duration of active/standby state. From this, the algorithm suggested in this paper does optimal scheduling that is ensuring continual playback of data blocks stored in mobile disk drive. And we implement our algorithms in publicly available MPEG player software. This MPEG player software saves up to 60% of power consumption as compared with full-time active stated disk drive, and 38% of power consumption by comparison with disk drive controlled by native power management method.

Key words : Disk Scheduling, Power Management, Multimedia, Playback, and Mobile Device

1. 서론

1.1 연구 배경

휴대용 모바일 장치가 대중화되고, 급속한 기술력 발전으로 다양한 기능을 지원하면서, 모바일 장치에서 사용되는 전력을 감소시키는 연구는 가장 중요한 이슈가 되었다. 게다가, 유한한 전력을 가진 모바일 장치는 휴대용 장치의 기능을 제한하는 중대한 원인 중의 하나이다[1]. 특히, 모바일 장치에서는 자료의 전송에 쓰이는 전력 소모가 기기 전체의 그것에서 많은 부분을 차지한다[2]. 용량대비 가격이 저렴한 디스크(하드디스크, 광디스크)들의 크기가 축소되어 모바일 장치에 사용되면서, 소모전력에 대한 효율적이고 엄격한 관리가 더욱 절실히 지면서 그에 관련한 연구가 많이 진행되고 있다. 그러나 아직도 모바일 장치에서는 소모전력을 더욱 효율적으로 관리할 수 있는 중요한 부분들이 남아있다. 소모전력을 감소하기 위한 전통적인 방법은 I/O 요청 큐가 비어 있을 때, 디스크 드라이브의 동작을 정지 시키는 것이다. 즉, I/O 요청이 발생하였을 때, 디스크를 구동시키며, I/O 요청이 없을 때, 디스크를 정지 시킨다[3]. 디스크 드라이브 동작정지 알고리즘은 크게 예측 방법[4]과 통계 방법[5]으로 분류할 수 있다. 플래시 메모리 같은 반도체와 달리 디스크 드라이브의 상태를 대기 상태에서 동작 상태로 전환하는 것은 많은 전력이 소모된다. 따라서, 디스크 드라이브를 스케줄링하기 위해서는 전력 소모의 특성을 정확히 분석하여 디스크 드라이브 동작을 제어하는 것이 반드시 필요하다. 일반적인 컴퓨터와 다르게, 다양한 모양의 휴대용 모바일 장치는 비교적 간단한 기능과 특징으로 최적화할 수 있는 시스템이다. 여기서 멀티미디어 자료의 재생장치 또한 예외가 아니다. 그러므로, 우리는 멀티미디어 자료의 특징과 I/O 요청을 정확히 분석하면 더욱 더 효율적인 소모전력 감소 방법을 찾을 수 있다고 확신한다. 본 논문에서는 지난 10년간 산업현장과 학계에서 꾸준히 연구해 온, 실시간 멀티미디어 자료 재생에 필요한 소모전력 최소화에 대한 문제에 대해 연구하였다. 우리는 저전력 디스크 드라이브의 전력소모형태를 정교하게 수식화하였고, 물리적 단위

와 논리적 단위를 기반으로 한 알고리즘들은 개발하였다. 그 알고리즘들은 최소의 전력으로 멀티미디어 자료를 실시간으로 재생할 수 있다.

1.2 관련 연구

모바일 장치에서의 전력 관리는 산업체와 학계에서 매우 흥미로운 관심사이다. 현재에도 하드디스크의 전력 관리에 대한 많은 연구들이 계속 되고 있다. 지난 몇 년간의 선행연구에서는, 디스크 드라이브 스케줄링방법을 통하여 10%에서 25%정도의 소모전력을 감소시키는데 성공하였다[6]. Simunic[7,8]은 semi-Markov 결정 진행 모델을 사용하였다. 이 방법은 하드디스크에 대한 요청을 구분할 수 있는 세션(session)으로 분류한 후에, 디스크 드라이브를 세션단위로 동작정지 시키는 것이다. Ramanathan[9]은 내장형 시스템을 위한 전력 관리 알고리즘 평가를 위한 해석 방법을 소개하였고, 마지막 요청과 다음 요청의 도착간의 간격을 예측할 수 있는 간단한 알고리즘을 제안하였다. Lu[10]는 감지된 세션의 종료를 위한 임계값을 적절히 변화시켰다. IBM[6]에서는 디스크 드라이브의 상태를 여러 단계로 나누어서 디스크 드라이브에 소모되는 전력을 감소시켰다. Greenawalt[1]는 다양한 전력 상태와 시스템 성능과의 관계를 고안하였다.

또한 운영체제 관점에서 바라본, 전력관리에 대한 다양한 연구가 존재한다[8,10,12,13]. Helmbold[14]는 모바일 컴퓨터에서 소모전력을 감소하기 위한 디스크 회전정지(spin down) 결정 시점에 대한 알고리즘을 소개하였다. Share라 불리는 이 알고리즘은 최근의 디스크 활용상태에 따라 일시중단(time-out) 값을 동적으로 조절한다. 이 전력관리 기법은 모바일 컴퓨터가 일반적인 작업에서 동작한다고 가정하는데, 모바일 장치가 특별한 기능(MP-3 재생기, 디지털 카메라, 캠코더 등)을 위해 디자인 되었을 때, 저장장치의 소모전력을 더 효율적으로 감소할 수 있다. Makoto[15]는 디스크 제이기가 스스로 버퍼의 빈공간을 감지하여 읽기/쓰기 회로에 전력을 재공급하는 방법에 의한 소모전력 감소 방법을 발견하였다. 이 방법은 커널의 미리읽기 버퍼(read-ahead

buffer)가 디스크에 저장된 자료를 하드디스크내의 버퍼로 읽어들이게 한다. 시간이 지난 후 버퍼가 가득 차면 미리읽기 버퍼를 정지하고 읽기/쓰기 회로에 전력 공급을 중단시킨다. 캐시가 히트되거나 다음 읽기 명령에 의해 버퍼가 비워질 때, 버퍼 제어기로부터 인터럽트 신호가 CPU에 전달된다. 이 신호는 CPU가 읽기/쓰기 회로에 전력공급 재개하도록 하고, 디스크 제어기는 디스크에 저장된 다음 섹터(sector)를 읽어들이게 하여 전력공급을 억제하는 방법이다. Koji[16]는 CD-ROM에서의 전력 감소 방법을 소개하였다. 이 방법은 CD-ROM의 버퍼 상태로 서보를 제어하는 방법이다.

이러한 방법들은 항상 성능의 저하를 일으키게 되는데, 디스크에 저장된 멀티미디어 자료를 실시간으로 재생하는데 있어서 가장 중요한 과제는 멀티미디어 자료를 손실(loss), 지연(delay) 그리고 지터(jitter) 없이 재생하는 것이다. 순환주기(round) 기반의 디스크 드라이브 스케줄링 알고리즘은 자료의 연속성을 보장하는데 널리 사용되고 있다[17,18]. 이러한 알고리즘은 디스크 드라이브가 항상 정상상태에서 동작한다고 가정한다. 그러나 동적 전력 관리(dynamic power management)에 의해 제어된 디스크 드라이브는 작업 상태에 따라 디스크 드라이브의 상태(동작 또는 대기)를 변화시키기 때문에[6,7], 모바일 장치의 배터리 수명은 충분히 연장시켰으나 멀티미디어 자료의 정상적인 재생에 어려움을 야기한다.

2. 멀티미디어 데이터 재생

우리는 멀티미디어 자료의 재생에 적합한 버퍼크기를 결정하고자 VBR 특성과 재생 방법에 대해 분석하였다. 본 논문에서는 여러 멀티미디어 파일 중 MPEG 비디오 파일에 대해 분석하였다. MPEG 파일은 I, P, 그리고 B 프레임으로 구성되어 있고, 또한 각 프레임의 패턴으로 구성된 GOP(Group Of Picture)를 가지고 있다. 멀티미디어 파일의 재생이 이루어질 때에는 먼저, 암호화된 자료를 디스크로부터 버퍼로 옮긴 후, 파일에 정의된 GOP에 따라 버퍼에서 제거한다. 만약 기대한 시간 동안에 버퍼가 채워지지 않으면, 해당 그림은 복호화 되지 않고, 버퍼에 복호화 할 자료를 채운다. 반면, 버퍼가 범람(overflow)하면, 멀티미디어 파일 재생시에 지터 현상이 나타난다. 본 논문에서, 이러한 문제를 해결하기 위한 버퍼 크기를 논할 것이다.

다음으로, 우리는 적합한 버퍼의 크기를 결정하기 위해서 각 프레임의 크기와 초당 필요한 프레임의 크기를 비교하였다. 그림 1에서 보는 것과 같이, 프레임은 종류별로 다른 크기를 가진다. 15KB 이상의 크기를 가지는 것은 I프레임, 5KB에서 15KB는 P 프레임, 5KB 미만은

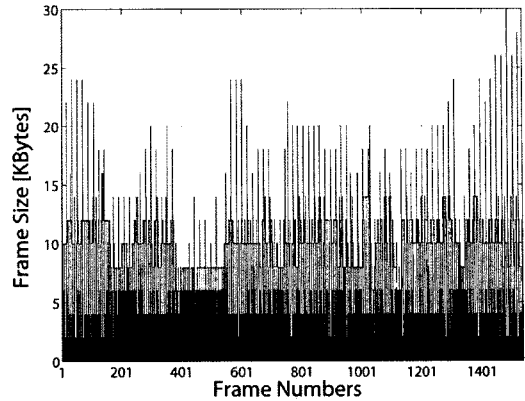


그림 1 MPEG2 파일의 각 프레임 크기

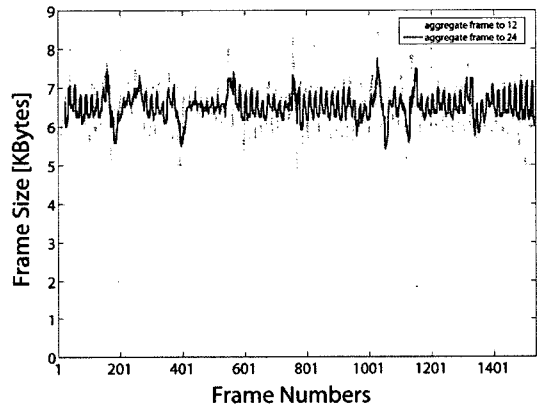


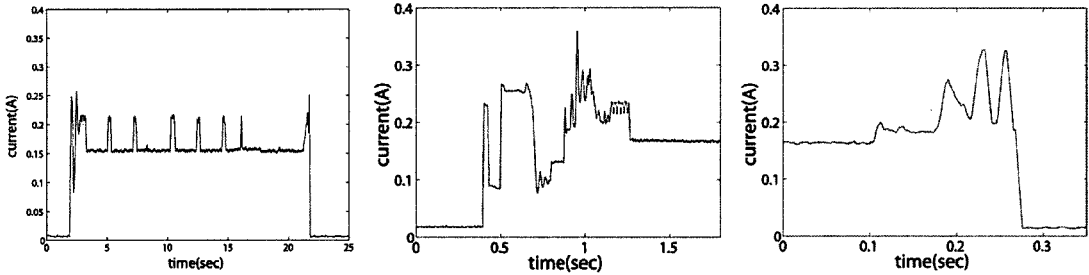
그림 2 결합된 프레임들의 자취

B 프레임이다. 이렇게 변화량이 심한 점들을 토대로 적합한 버퍼크기를 결정하는 것은 매우 어렵다. 하지만, 그림 2에서 보는 것과 같이 일정한 주기로 결합된 프레임의 변화는 매우 부드럽으므로, 우리는 결합된 프레임의 가장 큰 크기로 멀티미디어 재생에 적합한 버퍼의 크기를 구할 수 있다. 이로써 버퍼의 범람 현상을 방지할 수 있다.

3. 저전력 데이터 재생

3.1 하드 디스크 드라이브의 전력 소모 모델

그림 3에서 보는 것과 같이, 우리는 저전력 디스크 드라이브의 전력소모 형태를 고찰하였다. 이 그림은, 재생율이 1.5 Mbps인 20초 분량의 MPEG 파일을 Compaq사의 Windows XP가 설치된 ARMADA M700 노트북에서 재생하였고, 이것을 LeCroy사의 WavePro 950 오실로스코프로 파형을 측정하였다. 그림 3(a)는 MP3 파일의 전체 재생의 전력소모를, 그림 3(b)는 파일 재생의 시동 부분(startup)을, 마지막으로 그림 3(c)는 정상상태



(a) MPEG 파일 전체 재생동안의 전력 소모 형태 (b) 시동 구간의 전력 소모 형태 (c) 마무리 구간의 전력 소모 형태

그림 3 IBM Micro Drive(DMDM-10340)의 전력 소모 형태

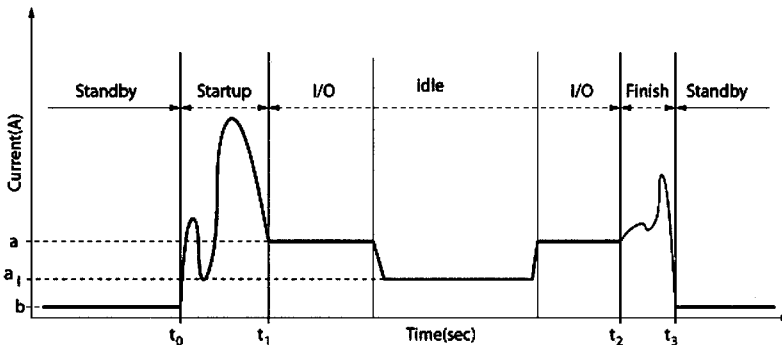


그림 4 자료 전송에서의 디스크 드라이브의 동작의 도식화

의 디스크 드라이브가 마무리(finish phase)되는 부분을 나타낸다.

그림 4에서 보는 것과 같이, 우리는 디스크 드라이브가 동작할 때의 전력 소모 상태를 분석하였다. 디스크 드라이브로부터 자료를 전송하는 상태는 시동(Startup), 읽기(Read), 공전(Idle), 마무리(Finish) 그리고 대기(Standby) 상태로 분류된다. 디스크 드라이브는 초기에 대기 상태에 있다. 이 대기 상태에서, 디스크 드라이브가 I/O 요청을 받으면, 디스크 드라이브 상태는 시동상태로 전환된다. 시동상태는 디스크 드라이브의 스피업(Spin up), 포커스(Focus) 그리고 트래킹(Tracking) 작업을 포함한다. 읽기 상태에서는 디스크에 저장된 자료를 버퍼로 전송한다. 모든 디스크 드라이브 I/O 요청이 서비스 되면, 디스크 드라이브는 읽기 상태에서 공전 상태로 전환된다. 디스크 드라이브가 공전 상태일 때, 디스크의 헤드는 초기위치(parking position)로 이동하지만, 디스크 플래터는 최고 속도로 계속 회전한다. 이 상태에서 새로운 I/O 요청이 디스크 드라이브에 전달되면, 디스크 드라이브는 읽기 상태로 전환된다. 전력 절약 알고리즘은 공전상태를 위한 일시중단 시간(time-out value)이 있다. 만약 디스크 드라이브가 공전 상태로 전환된 후 일시중단 시간동안에 새로운 I/O 요청이 없으면, 디스크

드라이브는 마무리 상태로 전환되어, 디스크 회전을 정지시키고, 대기 상태로 전환된다. 또한 대기 상태에서 디스크의 헤드는 초기 위치로 이동한다. 디스크의 회전이 정지하고, 호스트 인터페이스(Host Interface)를 제외한 모든 회로는 전력 절약 모드로 전환된다. 이것은 다섯 가지 상태를 기반으로 한 소형 디스크에서 자료를 회복시키는 데 있어서 매우 중요한 정보이다.

실시간 멀티미디어 재생을 위해서, 응용프로그램에서는 주기적인 대량 읽기 요청과 디스크 드라이브는 재생율에 의한 주기로 자료를 공급한다. 대량 읽기의 크기와 대량 읽기 간의 간격은 유용한 버퍼 크기, 적절한 시동 대기 시간, 탐색과 같은 오버헤드 그리고 회전 대기 시간에 따라 매우 신중히 결정하여야 한다[19]. 저전력 디스크 드라이브에서의 전력 절약 방법은 대기 상태 때문에 위에서 언급한 여러 값들은 결정해야 하는 복잡한 부분을 요구한다. 멀티미디어 재생에 필요한 전력 소모를 최소화 하기 위한 자료 재생 스케줄에서 대기 상태의 길이는 매우 중요한 역할을 한다.

위에서 분석한 사실을 기반으로 하여 알고리즘 도출을 위해 하드 디스크의 특성에 따라 몇 가지 파라미터를 추출해 내었는데 그에 대하여 표 1에 기재하였다. 여기에는 각 구간에 대한 시간 파라미터를 정의하였다. 다

표 1 디스크의 동작에 대한 파라미터

파라미터	단위	설명
T_s	Second	시동 구간의 시간
T_p	Second	읽기 구간의 시간
T_f	Second	마무리 구간의 시간
T_o	Second	대기 구간의 시간
P_s	Joule	시동 구간의 소모 에너지
F_f	Joule	마무리 구간의 소모 에너지
a	Watt	읽기 구간의 소모 전력
β	Watt	대기 구간의 소모 전력
R	MBytes/sec	디스크 전송률
B^*	Mbytes/sec	버퍼의 크기

음으로 각 구간에 대한 소모 자원을 표현하였다. 시동 구간과 마무리 구간의 소요 시간은 상수이며, 그러므로 에너지 단위를 사용하여 나타내었다. 따라서 단위를 Joule로 나타내었으며, 읽기 구간과 대기 구간의 소요 시간은 사용자의 요청에 따라 변화할 수 있으므로 전력 단위를 사용하여 나타내었다. 따라서 단위는 Watt로 나타내게 되었다.

3.2 문제 정의

그림 5는 데스크탑과 모바일 환경에서의 멀티미디어 자료 재생에 대한 동작을 각각 표현한 것이다. 좌측의 “데스크탑 환경”은 동적 전력 관리 기능이 없는 디스크 드라이브의 동작을 나타낸다. 따라서, 멀티미디어 자료 전체가 재생이 완료될 때까지 디스크 드라이브는 활성화(Active) 상태를 유지한다. 그림 5의 좌측그림과 같이 전력 관리 기능이 없는 알고리즘에서는 디스크 드라이브는 멀티미디어 재생이 끝날 때까지 디스크가 회전한다. 우리는 이것을 “Normal Playback”이라 정의한다. 반면 그림 5의 우측 그림과 같이, 동적 전력 관리 기능이 있는 저전력 디스크 드라이브는 디스크로부터 한번에 많은 양의 자료를 전송하고 대기상태로 전환한다. 이것은 멀티미디어 자료를 재생하는 동안에 디스크 드라이브가 활성화 상태에 있는 시간을 감소시킬 수 있다는

것이다. 본 논문에서 제시한 3개의 알고리즘은 이러한 점에 착안해 디스크로부터 자료를 전송하는데 필요한 전력소모를 최소화하기 위해 고안되었다.

3.3 Normal Playback 정책에서의 전력 소모량

우리는 일반적인 Normal Playback 정책에 의한 멀티미디어 데이터 재생에서 소모되는 전력량을 살펴보았다. B^* 는 버퍼의 크기이다. R 은 디스크에 저장된 자료를 버퍼로 전송하는 최대 전송 속도이고, γ 은 멀티미디어 파일의 평균 재생율을 나타낸다. L 은 멀티미디어 파일의 총 재생시간이다. Normal Playback 정책 하에서 디스크의 자료를 버퍼로 전송하는 읽기 상태의 길이 T_p 는 다음 식 (1)과 같이 구할 수 있다.

$$T_p = \frac{B^*}{R-r} + \frac{L \cdot r - B}{r} \tag{1}$$

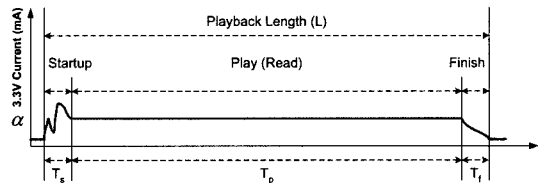


그림 6 Normal Playback 정책에 의한 전력 소모 형태

T_p 를 식 (1)과 같이 계산할 수 있는 이유는 디스크에서 최대속도로 자료가 전송되어 버퍼가 한번 가득 채워지게 되고, 그러면 그 다음부터는 버퍼에게 자료가 소모되는 속도에 맞추어 디스크에서 버퍼로 자료를 전송하기 때문이다. P_s 는 시동시의 전력 소모량이고, F_f 는 마무리상태에서 필요한 전력 소모량이다. α 는 읽기 상태에서의 전력 소모율이다. 그림 6은 Normal Playback 방법을 이용한 멀티미디어 파일의 재생에 대한 전력 소모 형태를 보여준다. 우리는 식 (2)와 같이 Normal Playback에 필요한 총 전력 소모량 P_N 을 공식화했다.

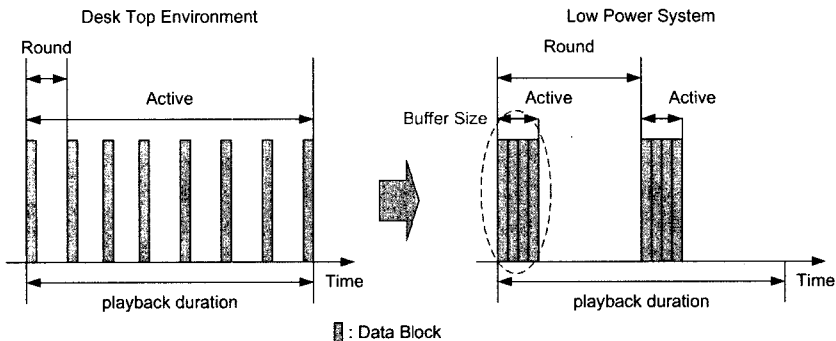


그림 5 멀티미디어 자료 회복: 데스크탑 대 모바일 환경

$$P_N = P_s + P_f + \alpha T_p$$

$$= P_s + P_f + \alpha \left(\frac{B^*}{R-r} + \frac{L \cdot r - B}{r} \right) \quad (2)$$

3.4 물리적 단위 기반의 재생(Physical Unit Based Retrieval) : Full Buffering

본 절에서는 물리적 단위, 즉 블록을 대상으로 멀티미디어 데이터를 재생하는 새로운 알고리즘을 소개한다. 우리는 멀티미디어 파일을 재생하는 또 다른 방법인 Full Buffering을 고안하였다. Full Buffering 방식은 버퍼가 가득 채워질 때까지 디스크에서 자료를 읽어 들인 후, 버퍼가 가득 채워지면 즉시 디스크 드라이브의 상태를 활성화 상태에서 대기 상태로 전환한다. 이러한 방식은 멀티미디어 전체 재생에 필요한 디스크 드라이브 동작을 여러 개의 순환주기로 분할한다. 각 순환주기마다 시동, 읽기 그리고 마무리 상태로 구성된 활성화 상태를 가지고 있다. Full Buffering 방식을 사용한 멀티미디어 재생에 대한 전력 소모 형태를 그림 7에 표현하였다. Full Buffering 방식에서 각 순환주기의 읽기 구간의 길이 T_p 는 식 (3)과 같이 계산 가능하다.

$$T_p = \frac{B^*}{R-r} \quad (3)$$

또한 각 순환주기에서, T_s 는 시동 상태의 시간, T_f 는 마무리 상태의 시간, 그리고 T_o 는 대기 상태의 시간을 지칭한다. 각 순환주기의 길이는 T 이다. 만약 읽기 상태의 시간이 T_p^* 이면, 우리는 각 순환주기의 길이 T 를 식 (4)와 같이 계산할 수 있다.

$$T = T_s + T_p + T_f + T_o$$

$$= \frac{T_p^* \cdot R}{r} \quad (4)$$

마지막 순환주기를 제외한 모든 순환주기는 동일한 길이이고, 멀티미디어 재생의 마지막 순환주기는 앞의 동일한 길이를 갖는 순환주기를 보다 길거나 작다. 한 순환주기에서의 전력 소모를 P_p 라 정의하고 마지막 순환주기에서의 전력 소모를 P'_p 라 할 때, P_p 는 식 (5)로

계산할 수 있고, P'_p 는 식 (6)으로 구할 수 있다.

$$P_p = P_s + P_f + \alpha T_p \quad (5)$$

$$P'_p = P_s + P_f + \alpha T'_p \quad (6)$$

동일한 길이를 갖는 순환주기들의 수를 N 이라 하면, N 은 다음과 같이 구할 수 있다.

$$N = \left\lfloor \frac{L \cdot r(R-r)}{B^* \cdot R} \right\rfloor \quad (7)$$

이제 우리는 위의 식들에서 Full Buffering 방식을 사용한 총 전 소모량 P_F 을 식 (8)과 같이 계산할 수 있다.

$$P_F = N \cdot P_p + P'_p \cdot I$$

$$= N(P_s + P_f + \alpha T_p) + (P_s + P_f + \alpha T'_p)I$$

$$= \left\lfloor \frac{L \cdot r(R-r)}{B^* \cdot R} \right\rfloor (P_s + P_f + \alpha \frac{B^*}{R-r}) + (P_s + P_f + \alpha T'_p)I \quad (8)$$

여기에서 T'_p 은 마지막 순환주기에서 읽기 상태의 시간이다. 만약 $\left\lfloor \frac{L}{T} \right\rfloor = 0$ 이면, 색인 함수 I 는 0이고, $\left\lfloor \frac{L}{T} \right\rfloor \neq 0$ 이면, I 는 1이다. 또한 I 가 0일 때, Full Buffering 방식은 최적의 전력 소모 형태이다.

3.5 논리적 단위 기반의 재생(Logical Unit Based Retrieval)

본 장에서는 프레임과 같은 논리적 단위를 대상으로 한 멀티미디어 재생 방법을 제안한다. 먼저, 그림 8에서 보는 것과 같이, MPEG 파일은 서로 다른 크기를 갖은 여러 개의 프레임들로 구성되어 있다. 이렇게 구성된 파일은 매초마다 일정한 수의 프레임(fps)을 디코딩하여 화면에 표시함으로써 재생된다. 이러한 점을 바탕으로 고안된 LUB알고리즘은 멀티미디어 파일 재생시에 지터와 지연 현상이 발생하지 않는 임계치의 프레임 수를 산출한 후, 디스크 드라이브는 산출된 최소의 프레임 수만큼 버퍼에 전송한 후에 바로 대기 상태로 전환하고, 전송된 프레임들이 재생되어지는 동안에 대기 상태를

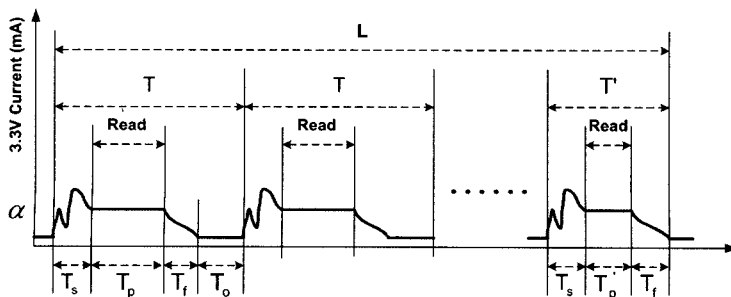


그림 7 Full Buffering 정책에 의한 전력 소모 형태

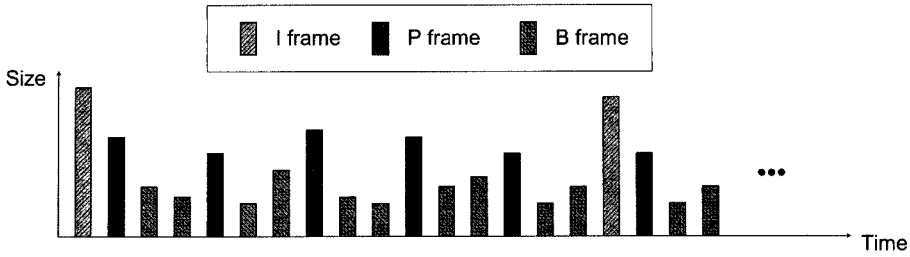


그림 8 MPEG 파일의 각 프레임 배열도

유지한다. 이러한 방식은 Full Buffering 방식과 마찬가지로 멀티미디어 재생에 필요한 디스크 드라이브의 동작을 여러 개의 순환주기로 분할한다. 각 순환주기마다 시동, 읽기 그리고 마무리 상태로 구성된 활성화 상태를 가지고 있다. LUB 알고리즘으로 멀티미디어 재생시에 나타나는 디스크 드라이브의 전력 소모 형태를 그림 9에 표현하였다.

LUB 알고리즘을 공식화하기 위해 필요한 새로운 표기들은 살펴보면, 다음과 같다. f 는 프레임 비율, 즉 초당 프레임 수이고, mf 은 각 순환주기의 읽기 상태에서 전송 받아야 할 최소 프레임의 수이다. $B(f)$ 은 fps를 기준으로 결합된 프레임의 크기 중 가장 큰 크기를 나타낸다. $T(f)$ 는 fps기준의 프레임 수를 디스크에서 버퍼로 전송하는 시간이고, $T_{max}(f)$ 은 각 순환주기의 $T(f)$ 중 가장 긴 시간이다. 먼저, mf 을 fps로 초기화한 후, 각 프레임을 fps단위로 결합한 크기 중 가장 큰 크기 $B(f)$ 는 식 (9)이다.

$$B(f) = \{ \max(x) \mid x = \sum_{i=m}^{fps+m-1} \text{SizeofFrame}_i, 1 \leq m \leq (\text{total_frame} - \text{fps}) \} \quad (9)$$

이렇게 구해진 $B(f)$ 는 멀티미디어 재생시의 여러 순환주기 중 읽기 상태의 시간이 가장 긴 시간을 요한다. 따라서, $B(f)$ 의 프레임 mf 개를 디스크에서 버퍼로 전송하는데 걸리는 최대시간 $T_{max}(f)$ 은 다음 식 (10)과 같이

구할 수 있다.

$$T_{max}(f) = \frac{B(f)}{R-r} \quad (10)$$

대기시간 T_o 를 과도한 입출력을 피하기 위해 시동 시간과 마무리 시간의 합보다 크다고 가정한다. 순환주기 T 는 디스크 드라이브의 시동 시간, 마무리 시간, 그리고 디스크의 자료를 버퍼로 전송하는 시간의 합보다 큰 시간을 가져야 하고, 한 순환주기에서 읽어 들인 프레임을 재생하는 시간보다 순환주기는 같거나 작으므로, 한 순환주기 동안에 디스크에서 버퍼로 전송할 최소 프레임 수는 식 (11)과 같다.

$$mf > f(2(T_s + T_f) + T_{max}(f)) \quad (11)$$

우리는 LUB 알고리즘에 필요한 최소 버퍼 크기를 구하기 위해, 멀티미디어 파일의 각 프레임의 크기를 최소 프레임 mf 를 기준으로 결합하였다. 이렇게 결합된 프레임들의 크기 중에 가장 큰 크기 $B(mf)$ 은 식 (12)와 같다.

$$B(mf) = \{ \max(x) \mid x = \sum_{i=m}^{mf+m-1} \text{SizeofFrame}_i, 1 \leq m \leq (\text{total_frame} - mf) \} \quad (12)$$

만약, 멀티미디어 재생기에 할당되는 버퍼의 크기가 $B(mf)$ 보다 작으면, 어떤 순환주기에서는 디스크에서 버퍼로 전송되어지는 프레임 손실이 발생할 수 있다. 왜냐하면, 그 순환주기에서 전송될 프레임의 수를 결합한 크

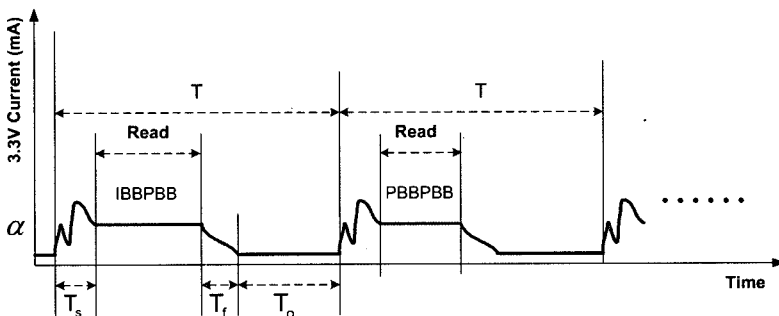


그림 9 MPEG 파일의 각 프레임 배열도

기보다 멀티미디어 재생기에 할당된 버퍼의 크기가 작기 때문이다. 따라서, 멀티미디어 재생기에 할당되는 버퍼의 크기는 반드시 $B(mf)$ 보다 커야 한다. 우리는 또한 LUB 알고리즘을 이용하여, 멀티미디어 파일을 재생할 때 소모되는 총 전력량은 공식화하였다. 먼저, 전체 순환주기의 수 N 은 멀티미디어 파일의 전체 프레임 수를 한 순환주기에서 읽어 들여야 할 프레임 수 mf 로 나누어 계산할 수 있다. 각각의 순환주기마다 디스크에서 버퍼로 전송되는 프레임 수는 같으나, 그 프레임들을 결합한 크기는 각기 다르다. 그러므로 각 순환주기의 읽기 상태 시간은 모든 순환주기마다 상이하다. 또한, 마지막 순환주기에서 디스크로부터 버퍼로 전송되는 프레임수의 크기 D_r 은 멀티미디어 파일의 총 크기에서 마지막 순환주기 전까지 읽어 들인 프레임의 총 크기를 뺀 값과 같다. 새로운 알고리즘을 이용한 멀티미디어 재생시에 디스크 드라이브의 동작으로 소모되는 총 전력량 P_R 은 식 (13)과 같다.

$$P_R = (n+1)(P_s + P_f) + \alpha \left(\sum_{i=1}^n \frac{\sum_{j=[1+(i-1)mf]}^{imf} \text{Frame}_j}{R-r} + \frac{D_r}{R} \right) \quad (13)$$

4. 최종 주기 병합 알고리즘(Adaptive Round Merge Algorithm)

Full Buffering 방식에서, 버퍼가 가득 채워지면, 디스크 드라이브는 자료 전송을 중지하고, 대기 상태로 전환된다. 이러한 방식은 전력 소모를 감소시키는데 매우 효율적이다. 그러나, 대기 상태에서 다른 순환주기로 변환되는 시점에 시동과 마무리 구간을 동반해야 되는 오버헤드가 존재한다. 만약, 마지막 순환주기에서 읽어 들여야 할 남아있는 자료가 작으면, 시동과 마무리 구간을 동반한 새로운 순환주기에서 자료 전송을 하는 것은 전력 소모 관점에서는 가치 있는 일이 아니다. 버퍼가 자료로 가득 채워지면, 우리는 남아있는 자료를 현재의 순환주기에서 계속 전송할 것인가, 아니면, 디스크 드라이브를 마무리 상태로 전환할 것인가를 결정할 필요가 있다.

우리는 이러한 결정에 대한 알고리즘을 개발 하였다. 먼저, 대기 상태의 디스크 드라이브에서 소모되는 전력은 무시한다. 대기 상태의 길이 T_o 는 식 (14)와 같이 계산할 수 있다.

$$T_o = \frac{T_p \cdot R}{r} - (T_s + T_f + T_p) \quad (14)$$

여기서 빈 버퍼를 채우는 시간은 $\frac{B^*}{R-r}$ 이다. 버퍼가 자료로 한번 채워지면, 디스크 드라이브는 버퍼에서 자료가 소모되는 비율로 디스크에서 버퍼로 자료를 전송

한다. 그러므로, T_p 동안 읽어들이는 자료량은 식 (15)과 같이 표현된다.

$$B = \begin{cases} T_p \cdot R, & \text{if } T_p \leq \frac{B^*}{R-r} \\ \frac{B^*}{R-r} R + \left(T_p - \frac{B^*}{R-r} \right) r, & \text{otherwise} \end{cases} \quad (15)$$

B 는 하나의 순환주기 동안에 디스크로부터 버퍼에 전송되는 자료량이다. 마지막 순환주기에서 읽어드려야 할 남은 자료량 B_1 은 $L \cdot r - N \cdot B$ 와 같다. 또한 마지막 순환주기에서 소모되는 전력량은 $P_s + P_f + \alpha \frac{B_1}{r}$ 이다. 만약, 우리가 마지막 순환주기를 바로 앞의 순환주기와 병합할 수 있다면, 우리는 마지막 순환주기의 시동과 마무리 구간에서 소모되는 전력도 절약할 수 있다. 그러나, 두 개의 순환주기를 병합하게 되면, 앞 순환주기의 읽기 구간은 $\frac{B_1}{r}$ 만큼 증가하고, 또한 전력 소모는 $\alpha \frac{B_1}{r}$ 만큼 증가한다. 따라서, 우리는 마지막 순환주기를 바로 전 순환주기에 병합할 것인가에 대한 판단을 할 수 있는 함수 P^* 를 식 (16)과 같이 만들었다.

$$P^* = \underbrace{(P_s + P_f + \alpha \frac{B_1}{R})}_{\text{No merge term}} - \underbrace{\left(\alpha \frac{B_1}{r} \right)}_{\text{Merge term}} \quad (16)$$

만약 $P^* > 0$ 이면, 마지막 순환주기는 바로 전의 순환주기에 병합되고, 그렇지 않으면, 남아있는 자료 B_1 은 새로운 순환주기에서 버퍼로 전송된다. 이 알고리즘은 주어진 멀티미디어 파일을 최소의 전력 소모로 재생하기 위한 것이다. 본 알고리즘의 의사코드는 표 2에 자세히 표현하였다. 이 알고리즘에서, 우리는 버퍼에 자료를 채우고 비우는 과정에서 동기화 문제를 접하게 된다. 따라서, 이 문제를 해결하기 위한 최소의 버퍼 크기를 식 (17)과 같이 제안한다.

$$B^* > \frac{(R-r)r}{R-2r} (T_s + T_f), \quad R > 2r \quad (17)$$

T_s 는 시동 상태의 시간이고, T_f 는 마무리 상태의 시간을 나타낸다. 멀티미디어 파일 재생에 최소의 버퍼 크기를 적용함으로써, 효율적인 전력 소모와 지터 없는 실시간 멀티미디어 재생을 할 수 있다.

5. 시뮬레이션

본 논문에서 제안한 알고리즘들의 효율성을 시뮬레이션에 의해 평가하였다. 시뮬레이션에 사용한 IBM DMDM-10340[20] 디스크 드라이브의 각 변수 값들은 다음과 같다. 시동 시간은 1초이고, 소모전력은 0.858 Joule이다. 마무리 시간은 0.7초이고, 소모전력은 0.7 Joule이다. 또한, 정상상태에서의 읽기 동작에 필요한

표 2 Adaptive Round Merge 알고리즘

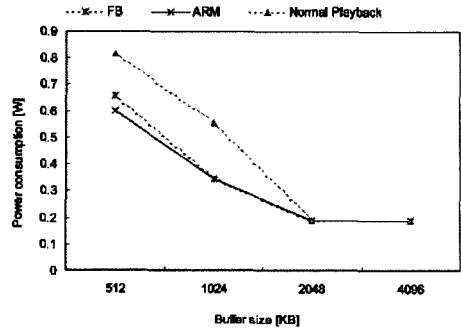
```

Algorithm: Adaptive Round Merge Algorithm() {
    i = 0;
    compute N;
    compute PN, PF;
    compute P*;
    IF ( PN ≤ PF ) {
        startup();
        continue reading;
        finish();
    }
    ELSE {
        WHILE ( N ≠ 0 ) {
            Dr = L · r - i · B;
            startup();
            read();
            IF ( Dr < B AND P* < 0 ) {
                continue reading;
                finish();
                N--;
            }
            finish();
            N++;
            i++;
        }
    }
}
    
```

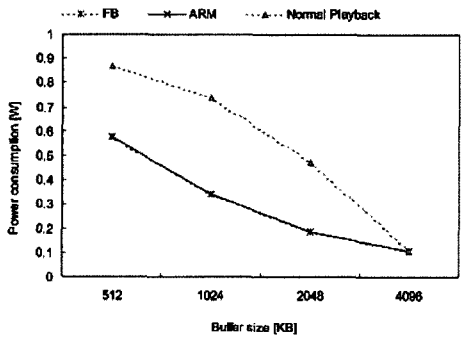
전력 소모율은 0.924 Watt이다. 디스크로부터 버퍼로 전송하는 최대 전송률은 41.6 Mbps이다.

5.1 버퍼 크기 대 전력 소모량

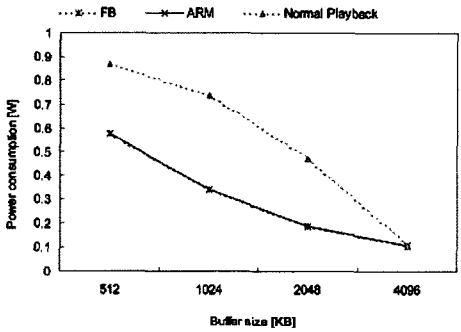
우리는 먼저 서로 다른 버퍼 크기 B^* 를 기반으로 소모되는 전력량을 비교하였다. 그림 10(a), 10(b), 10(c), 10(d)에서, X축은 버퍼의 크기를 나타내고, Y축은 전력 소모량을 나타낸다. 그림 10(a), 10(b), 10(c), 10(d)에서 보는 것과 같이, 전력 소모량과 버퍼의 크기는 반비례한다. 버퍼 크기가 클수록, 한 순환주기에 많은 양의 자료가 디스크 드라이브로부터 버퍼에 전송되고, 전력 효율은 개선된다. 재생시간이 10초인 그림 10(a)에서 보는 것과 같이, 버퍼크기가 512 KB인 경우에, 마지막 순환주기와 바로 전 순환주기를 병합하는 ARM 알고리즘은 Full Buffering 방식과 비교하여 약 9%의 전력을 절약하였다. 일반적으로 멀티미디어 파일의 재생 시간이 60초보다 짧은 경우에 순환주기를 병합하는 효과가 더욱 두드러지게 나타난다. Normal Playback 방식은 멀티미디어 파일 크기보다 더 큰 버퍼크기를 사용할 경우에 효율적이다. 그러나 일반적으로 멀티미디어 파일은 수



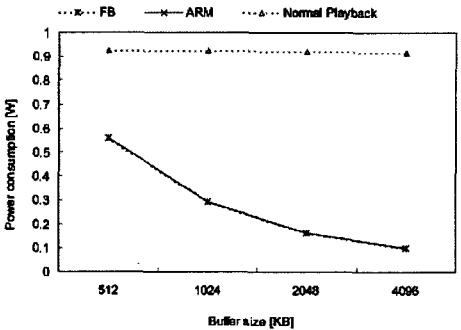
(a) 재생 시간이 10초일 때, 버퍼 크기 대 전력 소모량



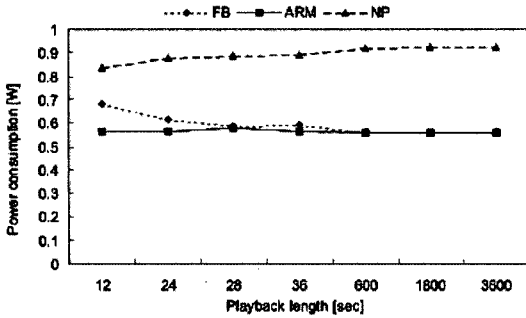
(b) 재생 시간이 20초일 때, 버퍼 크기 대 전력 소모량



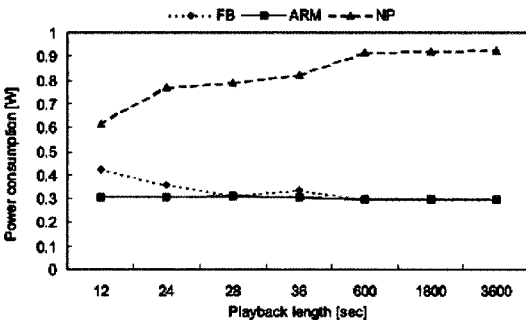
(c) 재생 시간이 1분일 때, 버퍼 크기 대 전력 소모량



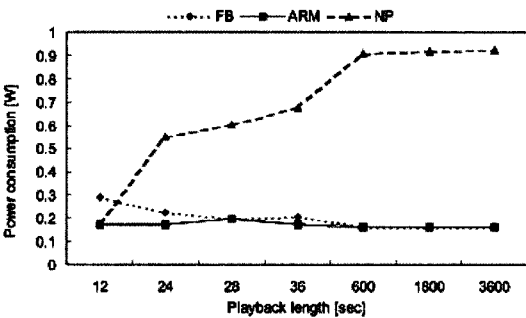
(d) 재생 시간이 60분일 때, 버퍼 크기 대 전력 소모량
 그림 10 재생 시간에 따른 버퍼 크기 대 전력 소모량



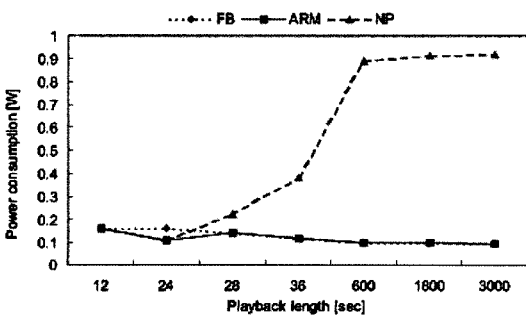
(a) 버퍼 크기가 0.5 Mbytes일 때, 재생 시간 대 전력 소모량



(b) 버퍼 크기가 1 Mbytes일 때, 재생 시간 대 전력 소모량



(c) 버퍼 크기가 2 Mbytes일 때, 재생 시간 대 전력 소모량



(d) 버퍼 크기가 4 Mbytes일 때, 재생 시간 대 전력 소모량
그림 11 버퍼 크기에 따른 재생 시간 대 전력 소모량

MB에서 많게는 수백 MB의 크기를 갖는다. 따라서, 이와 같이 큰 버퍼의 크기를 응용프로그램에 할당하는 것은 비현실적이다. 반면에, 그림 10(a)과 10(b)에서 보는 것과 같이, 멀티미디어 파일의 크기가 작을 경우에는 이보다 큰 크기의 버퍼를 사용하는 것은 전력 소모량을 개선시킨다. 왜냐하면, 멀티미디어 파일의 크기보다 할당된 버퍼 크기가 크므로, 디스크의 자료를 버퍼로 한번에 전송할 수 있기 때문이다. 그러나, 그림 10(c)와 10(d)에서 보는 것과 같이, 멀티미디어 파일의 재생 시간이 1분 이상이면, 4MB의 큰 버퍼크기를 사용하여도 전력 소모량은 개선되지 않는다.

그림 11(a), 11(b), 11(c), 11(d)은 재생 시간과 전력 소모량의 관계를 도시하였다. X축은 재생 시간을 나타내고, Y축은 전력 소모량을 나타낸다. 우리는 4가지 경우의 서로 다른 버퍼크기(0.5, 1, 2 그리고 4MB)를 고려하였다. 또한 멀티미디어 자료를 재생하는데 ARM 알고리즘을 이용하여 전력을 절약할 수 있는 이점을 관찰하였다. Full Buffering과 ARM 알고리즘에서, 버퍼가 가득 채워지면, 디스크 드라이브는 대기 상태로 전환된다. 채워진 버퍼가 비워질 때까지, 디스크 드라이브는 대기 상태를 유지한다. 그러나 Normal Playback 방식에서는 멀티미디어 재생 중에 디스크 드라이브가 대기 상태로 전환되지 않는다. 따라서, 멀티미디어 파일의 재생 시간이 길어질수록 전력 소모량은 읽기 상태의 전력 소모율에 점점 수렴한다. 앞서서도 언급했듯이, 마지막 순환주기와 바로 전 순환주기를 병합하는 장점은 재생 시간이 짧은 멀티미디어 파일에서 더욱 뚜렷하게 나타난다. 그림 11(a)에서 보는 것과 같이, 재생 시간이 12, 24 그리고 36 초의 경우를 보면, ARM 알고리즘은 Full Buffering 방식과 비교하여, 각각에 대해 약 18%, 28%, 그리고 40%의 전력 소모량을 감소시켰다. 이와 마찬가지로, 버퍼 크기가 1 MB일 경우에는 각각에 대해 9%, 15%, 그리고 23%의 전력 소모량을 감소시킨 것을 그림 11(b)에서 보여주고 있다.

5.3 LUB, Normal Playback, Full Buffering을 이용한 재생시 소모 전력 비교

이번 시뮬레이션에서는 4개의 MPEG2 파일을 사용하였다. 그 파일들에 대한 정보는 표 3에 나열하였다. 그림 12은 LUB, Normal Playback, Full Buffering 방식에 의해 멀티미디어 파일 전체를 재생할 때 소모되는 총 전력량을 보여준다. Y축은 총 전력 소모량을 나타낸다. 여기서 우리는 LUB와 Full Buffering 정책이 Normal Playback 정책에 비하여 소모 전력 절약 측면에서 월등한 능력이 있다는 것을 관찰할 수 있다. 수치적으로 살펴보면, LUB는 Normal Playback과 비교하여 44~49%의 전력을 절약하였고, Full Buffering은 Normal

표 3 시뮬레이션에 사용된 멀티미디어 파일의 특성들

이름	크기	평균 Bit Rate	프레임 수	재생 시간
santafe.mpeg	9.77(MB)	1.22(Mbps)	1535	64.0(sec)
jun.mpeg	6.45(MB)	1.30(Mbps)	357	39.7(sec)
benz.mpeg	8.61(MB)	1.23(Mbps)	1349	56.2(sec)
dunkman.mpeg	15.44(MB)	0.87(Mbps)	3389	141.2(sec)

Buffering과 비교하여 48~53%의 전력을 절약하였다. Full Buffering이 LUB보다 좀 더 많은 전력을 절약하였다. 이것은 Full Buffering은 매 순환주기마다 할당된 버퍼에 데이터를 가득 채우지만, LUB는 그렇지 않기 때문이다. 우리가 기대했던 것과 같이, LUB와 Full Buffering은 디스크에 있는 멀티미디어 데이터를 재생하는데 소모되는 전력을 충분히 감소할 수 있었다. 더욱 흥미로운 것은 멀티미디어 파일의 크기가 증가할수록 이 알고리즘들의 효과도 증가한다는 것이다.

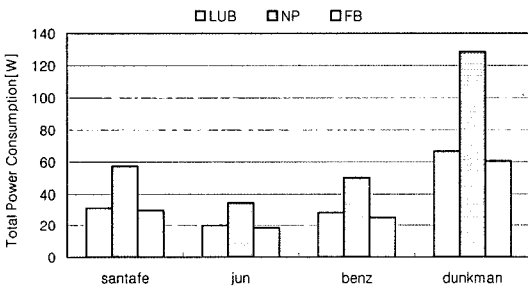


그림 12 LUB, Normal Playback, Full Buffering 각 방식에 의한 총 소모 전력량

6. 구현

본 논문에서 제안한 알고리즘을 Linux 커널 2.4.18을 사용한 Redhat 7.3 플랫폼에서 구현하였다. 구현된 재생기의 이름은 PoMP(Power conscious Multimedia

Player) v0.1라 명하였다. PoMP를 Compaq 노트북(ARMADA M700)의 CF/PCMCIA소켓을 사용한 IBM DMDM-10340에서 실험 하였다. CF/PCMCIA 카드의 핀 배열도는 그림 13에 나타나 있다. 17과 51번 핀을 통하여 +5V의 전원이 CF/PCMCIA 카드에 공급된다. 따라서 이 핀들에 흐르는 전류를 측정함으로써 디스크 드라이브에서 소모되는 전력을 알 수 있다. 그림 14에서 보는 것과 같이, 디스크 드라이브에서 소모하는 전력을 측정하기 위하여 CF/PCMCIA 카드의 각 핀들을 연장하였다. PoMP v0.1은 디스크에 있는 멀티미디어 파일을 최소의 전력으로 재생하기 위해 디스크 드라이브의 회전을 제어하는 알고리즘으로 구현된 MPEG 재생기이다. 현재, PoMP v0.1은 Linux기반의 mpeg2dec 재생기를 기초로 변형하였다. Mpeg2dec에 멀티쓰레드(multi-threads)를 구현하였고, 전력 감소 모듈을 추가하였다. 이것은 CPU자원 사용량을 최소화 할 수 있다. 우리는 두개의 쓰레드를 만들었다. 하나는 디스크로부터 데이터를 버퍼로 전송하는 역할을 하고, 다른 하나는 버퍼에 있는 데이터를 디코딩(decoding)하고 렌더링(rendering)하는 작업을 한다. 이 두개의 쓰레드는 고전적인 생산과 소비의 관계를 기초하여 만들어졌다. 두 쓰레드의 임계 영역은 세마포어를 이용하여 보호하였다. 또한 동시에 작업을 위하여 더블 버퍼링 기법을 사용하였다. 본 논문에서 제시된 알고리즘은 디스크에서 읽어야 할 데이터의 최소량을 계산한다. 이것은 멀티미디어 재생기에게 최소의 메모리를 할당하기 위한 것이다. 디스크의 특성을 분

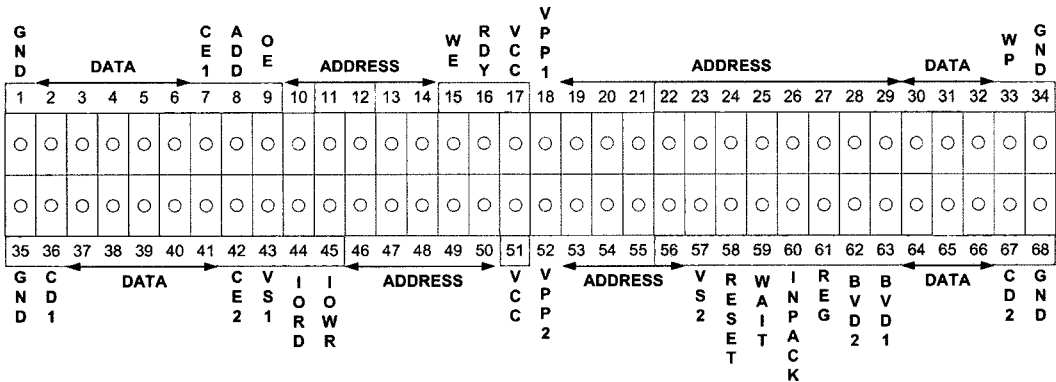


그림 13 CP/PCMCIA 카드의 69핀 정의

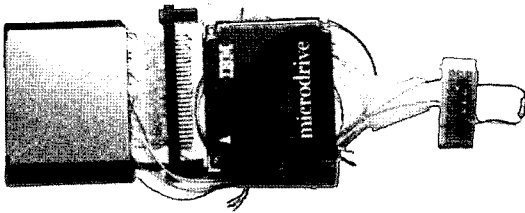


그림 14 IBM micro-drive DMDM10340

석하여 최소의 버퍼를 계산하는 이러한 방법은 제한된 자원을 사용하는 모바일 환경에서 매우 유용하게 사용할 수 있다

7. 실험

본 논문에서 제안한 알고리즘과 기존의 멀티미디어 재생 방식들을 비교한 실험을 통해, 제안한 알고리즘의 효율성을 보여준다. 이 실험에는 IBM사의 DMDM-10340 마이크로 드라이브를 사용하였다. 다음 표 4에 실험에 사용된 마이크로 드라이브의 몇 가지 파라미터들을 기술하였다.

표 4 마이크로 드라이브의 파라미터

파라미터	설명	값
T_s (sec)	시동 시간	0.8
T_f (sec)	마무리 시간	0.17
P_s (J)	시동 시간의 에너지 소모	0.78
P_f (J)	마무리 시간의 에너지 소모	0.19
α (Watt)	활성화 상태의 전력 소모	0.22
β (Watt)	대기 시간의 전력 소모	0.02
R (MBytes/sec)	디스크 전송률	5.0

본 실험에서 사용한 멀티미디어 재생기는 다음에 나열한 3개이다. 기존에 존재하는 멀티미디어 재생기를 변형한 PoMP v0.1, Windows Media Player v7.10, 그리고 mpeg2dec-0.2.1이다. Mpeg2dec는 한번에 많은 데이터를 전송하기 위하여 버퍼의 크기를 증가시켰다. 디스크 드라이브의 전력 관리 모듈은 디스크의 동작을 모니터링한다. 만약 디스크 드라이브가 일정한 시간동안 공전상태에 있으면, 디스크 드라이브를 대기 상태로 만든다. 이 실험에 사용된 MPEG2 파일은 표 3에 나열하였다. 그림 15, 16, 17은 3개의 재생기에서 멀티미디어 파일을 재생할 때의 전력 소모 형태를 보여준다. 그림 15에서 보는 것과 같이, 멀티미디어 파일의 전체 재생동안 디스크 드라이브는 활성화 상태를 유지한다. 그 후, 모든 재생이 완료되면, 디스크 드라이브는 활성화 상태에서 대기 상태로 전환되어 진다. 그림 16은, 변형된 mpeg2dec-0.2.1에 의해 재생할 때의 전력 소모 형태를

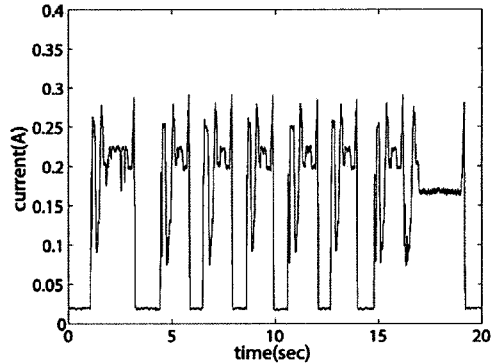


그림 15 Window Media Player에서의 전력 소모 형태

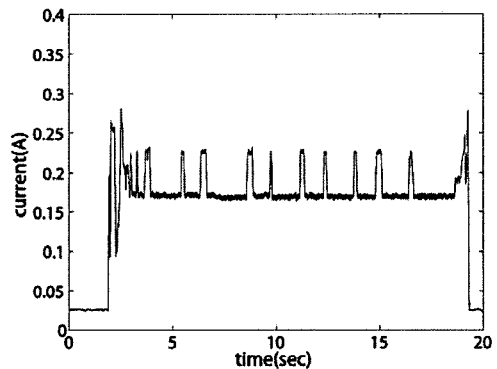


그림 16 mpeg2dec에서의 전력 소모 형태

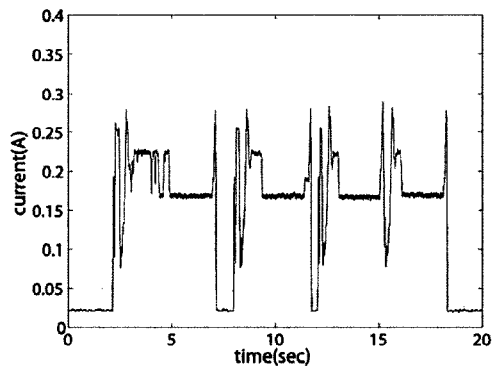


그림 17 PoMP v0.1에서의 전력 소모 형태

보여준다. 이것은 window media player보다는 전력 소모 측면에서 효율적이다. 그 이유는 한번에 많은 데이터를 디스크에서 버퍼로 전송하고 디스크 드라이브의 회전을 제어하기 때문이다. 그러나 이러한 접근에는 중요한 문제가 있다. mpeg2dec-0.2.1는 시동 구간의 길이를 고려하지 않았기 때문에 데이터 블록을 연속적인 전송을 보증하지 못한다. 게다가, 우리는 디스크에서 데이터

를 읽어오는 동안 매번 지터를 보게 된다. 그림 17에서 보여주는 것과 같이, PoMP 재생기는 전력 소모가 매우 효율적이고, 데이터를 지터없는 재생을 보장한다. 이것은 디스크 드라이브의 전력 소모 형태와 데이터 블록의 특성을 분석하여 저전력 드라이브의 활성화/대기 상태를 정확히 제어할 수 있는 위대한 성과라 사료된다. 각 재생기의 활성화 상태, 즉 전력을 소모하는 시간을 보면, windows media player는 17.1초, mpeg2dec-0.2.1 재생기는 14.3초, 그리고 PoMP는 10.5초이다. 따라서, PoMP는 windows media player와 비교하면, 약 38.5%의 소모 전력이 감소되고, mpeg2dec-0.2.1과 비교하면, 약 26.3%의 소모 전력을 감소시켰다.

8. 결론

멀티미디어 파일의 실시간 재생은 디스크 드라이브 스케줄링에 많은 부분을 의존한다. 디스크의 자료를 회복하는 것은 각각의 자료들이 이상 없이 재생되고, 시동 대기 시간 그리고 버퍼 요청과 같은 여러가지의 오버헤드를 최소화하기 위한 적당한 스케줄링이 필요하다. 저전력 디스크 드라이브의 전력 관리 기능은 멀티미디어 재생에 있어서 효율적인 전력 소모를 할 수 있는 새로운 기틀을 만들었다. 그러나, 이러한 전력 관리 기능은 디스크의 자료 회복을 스케줄링해야 하는 또 다른 복잡한 문제를 야기시켰다. 본 논문에서는, 저전력 디스크 드라이브의 전력 소모 형태를 분석하였고, 디스크에 저장된 블록의 연속성을 보장하면서, 최소의 전력으로 멀티미디어 파일을 재생할 수 있는 Full Buffering, LUB, Adaptive Round Merge 알고리즘을 제안하였다. 디스크 드라이브의 전력 소모 특성, 디스크의 자료 전송 속도, 그리고 멀티미디어 파일의 재생율을 알고 있으면, 이 알고리즘들은 각 순환주기에서 읽어 들어야 할 자료의 크기와 대기 상태의 길이를 계산할 수 있다. 이러한 값들을 유용하게 이용한 본 알고리즘들은 디스크의 회전을 시작할지 정지할지를 결정하는 디스크 드라이브 스케줄링이다. 시뮬레이션과 실험에서, 우리는 본 알고리즘들이 멀티미디어 파일을 재생하는데 필요한 전력 소모량을 충분히 개선시킨 것을 관찰하였다. 전력 관리 기능이 없는 Normal Playback 방식과 비교하면, 우리의 알고리즘들은 1Mbyte의 버퍼를 사용할 경우, 60%까지 소모되는 전력을 절약 할 수 있었다. 우리는 본 논문에서 제안한 알고리즘을 구현하였고, 기존의 재생기에 내장하였다. 현대의 디스크 드라이브 대부분은 적응 전력 관리 기능을 가지고 있다. 우리가 실험에서 사용한 IBM 마이크로 드라이브 또한 요청의 상태에 따라서 디스크의 활성화/대기 상태를 동적으로 제어한다. 그러나, 적응 전력 관리 기능은 멀티미디어 프로그램이 가지고

있는 과도한 입출력 요청에서는 제 기능을 발휘하지 못한다. 우리가 제안한 알고리즘은 기존의 적응 전력 관리 기능을 가진 디스크 드라이브와 비교하여 38%의 소모 전력을 감소 시켰다. 이러한 효율적인 재생기인 PoMP의 여러 정보를 공개하였다[21].

참고 문헌

- [1] H. Woesner, J. P. Ebert, M. Schlager, and A. Wolisz, "Power-saving mechanisms in emerging standards for wireless lans: The mac level perspective," *IEEE Personal Communications*, pp. 40 - 48, 1998.
- [2] N. Bambos, "Toward power-sensitive network architectures in wireless communications: Concepts, issues, and design aspects," *IEEE Personal Communications*, pp. 50 - 59, 1998.
- [3] Y.-H. Lu and G. D. Micheli, "Adaptive hard disk power management on personal computers," in *In Proceedings of the IEEE Great Lakes Symposium on VLSI*, 1999, IEEE, 1999. 26.
- [4] F. Douglis, P. Krishnan, and B. Bershad, "Adaptive disk spin-down policies for mobile computers," *Computer Systems*, vol. 8, pp. 381 - 413, 1995.
- [5] E. Y. Chung, L. Benini, A. Bogliolo, and G. D. Micheli, "Dynamic power management for non-stationary service requests," in *Proceedings of Design Automation and Test in Europe*, 1999.
- [6] IBM, *Adaptive power management for Mobile Hard Drives*. IBM Co. Storage Systems Division, San Jose, CA, USA, 1999.
- [7] T. Simunic, L. Benini, P. Glynn, and G. D. Micheli, "Dynamic power management of laptop hard disk," in *Proceedings of Design, Automation and Test in Europe Conference and Exhibition 2000*, p. 736, 2000.
- [8] T. Simunic, L. Benini, P. Glynn, and G. D. Micheli, "Event driven power management," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 7, pp. 840 - 857, 2001.
- [9] P. Ramanathan, K. M. Sivalingam, P. Agrawal, and S. Kishore, "Resource allocation during handoff through dynamic schemes for monile multimedia wireless networks," in *presented at Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, 1999.
- [10] Y.-H. Lu, L. Benini, and G. D. Micheli, "Operating-system directed power reduction," in *Proceedings of the International Symposium on Low Power Design, 2000* (ACM, ed.), (New York, USA), 2000.
- [11] P. M. Greenawalt, "Modeling power management for hard disks," in *Proceedings of the Second International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication*

Systems(MASCOTS'94), 1994.27.

- [12] L. Benini, A. Bogliolo, S. Cavallucci, and B. Ricco, "Monitoring system activity for os-directed dynamic power management," in *Proceedings of IEEE International Symposium on Low Power Electronics and Design*, pp. 185 - 190, 1998.
- [13] D. Ramanathan, S. Irani, and R. K. Gupta, "An analysis of system level power management algorithms and their effects on latency," *IEEE Trans. on Computer-Aided Design of Intergrated Circuits and Systems*, vol. 21, no. 3, pp. 291 - 305, 2002.
- [14] D. P. Helmbold, D. D.E.Long, T. L. Sconyers, , and B. Sherrod, "Adaptive disk spin-down for mobile computers," *Mobile Networks and Applications*, vol. 5, no. 4, pp. 285 - 297, 2000.
- [15] K. Makoto, I. Yasuhiko, and I. Shuichi, "Disk storage device and power supply control method for same device," (Japan), 1999.
- [16] M. Koji, "Device for reading data out of disk recoding medium," (Japan), 1999.
- [17] D. R. Kenchammana-Hosekote and J. Srivastava, "Scheduling continuous media on a video-on-demand server," in *Proceedings of International Conference on Multi-media Computing and Systems*, IEEE, 1994.
- [18] A. L. N. Reddy and J. Wyllie, "Disk scheduling in a multimedia i/o system," in *Proceeding ACM Multimedia conference*, (New York), 1992.
- [19] Y. Won and J. Srivastava, "Smdp: Minimizing buffer requirements for continuous media servers," *ACM/Springer Multimedia Systems Journal*, vol. 8, pp. 105 - 117, 2000.
- [20] IBM, *Oem hard disk drive specifications for DMDM-10340/10170 IBM microdrive revision 1.0*. IBM Co. Storage Systems Division, USA, 1999.
- [21] J. Choi and Y.Won, "Pomp" "<http://www.dmclab.hanyang.ac.kr/research/project/pods/pods/main.htm>," 2003.



원 유 집

1990년 서울대학교 자연과학대학 계산통계학과 학사. 1992년 서울대학교 자연과학대학 계산통계학과 전산학 석사. 1997년 University of Minnesota 전산학 박사. 1997년~1999년 Server Performance Analyst, Intel Corp. 1999년 3월~현재 한양대학교 공과대학 전자전기컴퓨터 공학부 교수. 관심분야는 멀티미디어 시스템, 멀티미디어 네트워크, 인터넷 프로토콜, 성능평가이론, 데이터베이스, 운영체제



정 원 민

2005년 한양대학교 공과대학 전자전기컴퓨터공학부 학사. 2005년~현재 한양대학교 공과대학 전자통신컴퓨터공학부 석사 과정. 관심분야는 멀티미디어 시스템, 실시간 시스템, 내장형 시스템, 운영체제



최 정 완

2000년 순천향대학교 제어계측공학과 학사. 2003년 한양대학교 전자통신전파공학과 석사. 현재 LG전자기술원. 관심분야는 모바일, 멀티미디어