

# CRC64해시를 이용한 에너지 효율적인 파일 전송 기법 설계 및 구현

정호민\*, 강성운\*, 이정근\*, 고영웅\*

\*한림대학교 컴퓨터공학과

e-mail:{chorogyi, upersbird, Jeonggun.lee, yuko}@hallym.ac.kr

## Design and Implementation of Energy Efficient File Transfer Mechanism Using CRC64

Ho Min Jung\*, Sung Woon Kang\*, Jeong Gun Lee, Young Woong Ko\*

\*Dept of Computer Science, Hallym University

### 요 약

모바일 단말에서 소프트웨어의 기능을 구현할 때, 배터리 소모를 최소화 시키는 것이 매우 중요하다. 본 연구에서는 모바일 단말에서 파일 전송을 수행함에 있어서 에너지 효율적으로 처리하는 기법을 제안한다. 주요 아이디어는 클라이언트 부분에서 전송하려는 파일을 128Kbyte 크기의 블록으로 나누고 CRC64를 사용하여 블록의 해시를 서버로 보내 블록을 전송할 지 결정한다. 보내고자 하는 블록이 서버에 있는 크기만큼 대역폭을 줄이면서 모바일 단말의 에너지 소모를 줄이는 것이다. 실험 결과 5% 이상의 중복율을 가지는 파일에 대해서 제안하는 기법이 배터리 소모를 크게 줄일 수 있음을 보였다.

### 1. 서론

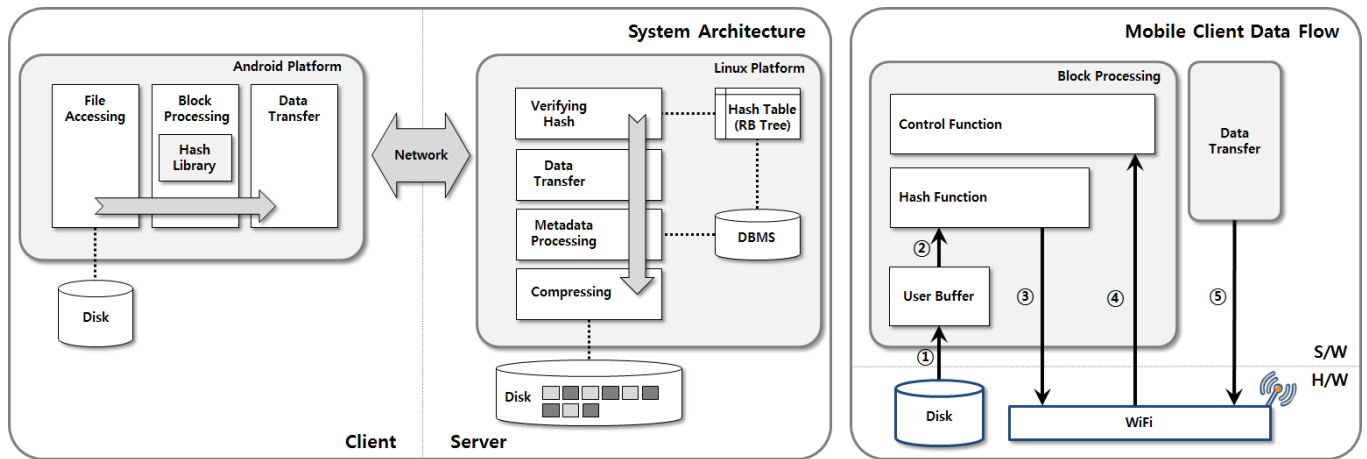
WiFi 기술의 발전과 스마트 폰의 저장 공간의 확대로 인해 컴퓨터로 데이터를 전송하는 사용자들이 늘고 있다. 무선 통신망을 이용해 데이터를 전송하는 방법은 편리성을 제공하지만 데이터 전송을 위해서 사용되는 무선네트워크 장치에서 배터리를 과도하게 소모하는 것이 큰 약점으로 작용한다. 또한 스마트폰에서 데이터를 전송할 때 동일한 데이터나 중복된 블록을 가지는 데이터가 존재하는 경우 불필요한 데이터 전송으로 에너지를 소모하게 된다. 본 연구에서는 에너지 효율적으로 파일 전송을 하는 방법을 제공하기 위해서 스마트 폰에서 데이터 중복을 제거하여 데이터를 전송하는 기법을 제안한다. 스마트 폰에서 WiFi가 사용하는 에너지 소모량이 매우 높기 때문에 CPU 사용률은 높지만 네트워크 대역폭을 줄일 수 있는 방식을 사용해야 한다.[1] 본 논문에서 제안하는 방식은 보내고자 하는 파일을 128Kbyte로 나누고 CRC64[2]를 사용하여 블록의 해시를 만들고 이를 서버로 보내 중복인지 확인하는 것이다. 만약 서버에 중복된 데이터가 존재한다면 해당 데이터에 대한 전송을 피할 수 있기 때문에 네트워크 대역폭을 줄여 에너지 소모를 줄일 수 있다. 실험 결과 본 연구에서 제안하는 기법이 기존의 파일 전송 방식보다 불필요한 에너지 소모를 크게 감소시킬 수 있음을 보였다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 중복제거와 관련된 최근 동향을 살펴보고, 3장에서는 CRC64를 사용하는 이유와 모바일 단말에서 에너지를 측정하는 방법에 대해서 알아보고, 4장에서는 전체 시스템의 설계와 구현내용을 설명하고 5장에서는 성능평가에 대해 기술하고 6장에서는 결론 및 향후 연구방향을 제시한다.

### 2. 관련 연구

중복 제거에 관한 연구는 대규모 백업 시스템이나 P2P 시스템 그리고 CDN(Contents Delivery Network)를 중심으로 이루어져 왔으며, 대표적인 연구 결과는 Rsync[3], Venti[4], LBFS[5], HydraFS[6], DEDE[7] 등이 있다. Rsync는 네트워크로 연결된 디렉토리의 데이터를 동기화 시켜주는 프로그램으로 롤링 체크섬(Rolling Checksum)이라는 중복 데이터를 검색하는 알고리즘을 사용해 새로운 데이터의 복사만 일어나게 한다. Plan9[8]의 Venti는 네트워크 스토리지 시스템에서 중복 데이터를 제거하여 저장하는 스토리지 시스템이다. Venti는 데이터를 저장할 경우 파일을 고정된 크기(8Kbyte)의 블록으로 나누고 각 블록에 SHA1 해시를 적용하여 160bit 크기의 해시를 만들고 전송한다. LBFS는 자주 끊기거나 품질이 좋지 않은 네트워크 환경을 위해 설계된 네트워크 파일 시스템이다. LBFS에서는 파일 전송 효율을 높이기 위해 CDC(Content-define Chunks) 방식을 사용한다. CDC는 Rabin Fingerprint[9]로 해시하여 특별한 값을 반환하는 앵커(Anchor) 블록을 파일에 삽입하고 데이터 전송 전에 앵커 사이의 블록을 S

본 연구는 교육과학기술부와 한국연구재단의 지역혁신인력양성사업과 기초연구사업(No.2010-0016143)의 지원을 받은 결과물임을 밝힙니다.



[그림 1] 시스템 구성도 및 모바일 클라이언트 데이터 흐름도

HA1, MD5같은 해시 함수를 사용해 해시를 만들고 그 해시와 비교하여 중복을 검색하는 방법이다.

Rsync는 프로젝트 파일들을 배포할 때 유용하게 쓰이지만 사용자가 동시에 접속하여 같은 파일을 접근할 때 문제가 발생하며 Venti와 LBFS는 파일 시스템이기 때문에 모바일 단말에 적용하기 어렵다. 본 연구에서는 스마트 폰에 애플리케이션 형태로 클라이언트를 구현하였다.

### 3. 에너지 측정방법 및 해시 계산

모바일 단말의 파일 전송에서 에너지를 소모하는 컴포넌트로 CPU와 WiFi가 있으며 CPU 에너지 소모는 점유율(Utilization)과 주기(Frequency)값으로 구할 수 있다. 모바일 단말 모델에 따라서 계수 값이 달라지며 실시간으로 CPU 에너지 소모량을 구하기 위해 /proc 파일시스템에서 jiffy 값과 /sys 파일시스템에서 cpuinfo\_cur\_freq를 구해 점유율과 주기를 구한다. WiFi의 에너지 소모량을 구하기 위해 WiFi 인터페이스의 low 상태와 high상태를 정의하고 초당 전송, 수신 개수를 구해야 한다.

CRC(Cyclic Redundancy Check)는 통신 링크로 전송되어온 데이터 내에 에러가 있는지 확인하기 위한 방법 중의 하나이다. CRC는 16비트에서 256비트까지 표현 할 수 있는 다양한 종류들이 있으며 본 연구에서는 CRC64를 사용하였다. CRC64는 64비트 다항식( $x^{64} + x^4 + x^3 + x + 1$ )를 적용하여 해시를 구할 수 있다.

$$1 - \prod_{i=1}^{N-1} e^{-\frac{i}{M}} = 1 - e^{-\sum_{i=1}^{N-1} \frac{i}{M}} = 1 - e^{-\frac{1}{M} \frac{N(N-1)}{2}} \quad \dots \dots (1)$$

널리 사용되고 있는 CRC16, CRC32를 사용하지 못하는 이유는 해시 충돌 확률이 높기 때문이다. (1)식은 블록의 개수를 N으로 하고 해시 데이터의 범위로 M으로 하여 해시 충돌을 계산하는 식이다. 테라바이트 단위의 데이터를 사용한다고 하였을 때 CRC16과 CRC32의 경우  $10^3$  이상이 충돌할 수 있지만 CRC64는  $10^{-6}$  이하의 해시만 충돌할 수 있다.[10]

### 4. 시스템 설계 및 구현

그림1은 본 논문에서 제안하는 CRC64 기반 파일 전송 시스템의 개념도와 클라이언트 데이터의 흐름을 보이고 있다. 시스템은 모바일 클라이언트와 서버로 구성되며 클라이언트는 디렉토리를 선택하고 단일 또는 복수 개의 파일들을 선택할 수 있는 파일 접근 모듈(File Accessing), 파일을 블록으로 나누고 해싱 하여 해시 목록을 서버로 전송하는 블록 처리 모듈(Block Processing), 최종적으로 데이터를 서버로 전달하는 데이터 전송 모듈(Data Transfer)로 구성된다. 서버는 클라이언트에서 수신한 해시들을 기존의 저장되어 있는 해시목록들과 비교해 클라이언트가 전송하려는 해시가 중복된 해시인지 검사하는 해시 검색모듈(Hash Lookup), 디스크 공간을 절약하기 위해 수신된 블록 데이터들을 압축하여 디스크에 저장하는 압축 모듈(Compressing), 저장된 블록들을 파일 단위로 복원하기 위해 필요한 메타데이터를 생성하고 저장하는 메타데이터 처리 모듈(Metadata Processing)로 이루어져 있다. 효율적으로 해시를 검사하기 위해서 디스크 액세스를 피해야 하고 모든 해시를 RB 트리 자료구조로 배열하여 검색한다. 서버의 디스크에는 유일한 해시 값을 갖는 블록들만 저장하여 저장 공간을 줄인다.

그림1에서 데이터의 흐름은 (1)사용자가 전송할 파일들을 선택하고 전송 명령을 보내면 스마트 폰의 SD카드에 저장되어 있는 파일들을 파일 전송 프로그램의 버퍼에 복사한다. (2)파일을 블록크기 만큼 읽고 블록의 해시를 계산한다. 그러나 본 연구에서는 해시 계산 구간과 데이터 전송 구간을 나누어 에너지 소모를 구해야 하기 때문에 정확하게 모니터링 할 수 있도록 (2)에서 전체 파일의 블록 해시들을 한꺼번에 계산한 후 (3)해시 목록을 서버로 전송하였다. (4)서버는 클라이언트에서 전송한 해시 값들을 가지고 기존에 저장한 파일 중에 중복 된 블록이 있는지 검사하고 만약 해시가 가리키는 블록이 중복이라면 '1', 중복이 아니라면 '0'인 값으로 블록의 중복 여부를 클라이언트로 전송한다. (5)클라이언트는 (4)에서 받은 비트 조합을 가지고 블록을 전송할지 결정한다. 이 때 중복 제거 파일 전송 시스템의 실제 데이터 전송이 일어난다.

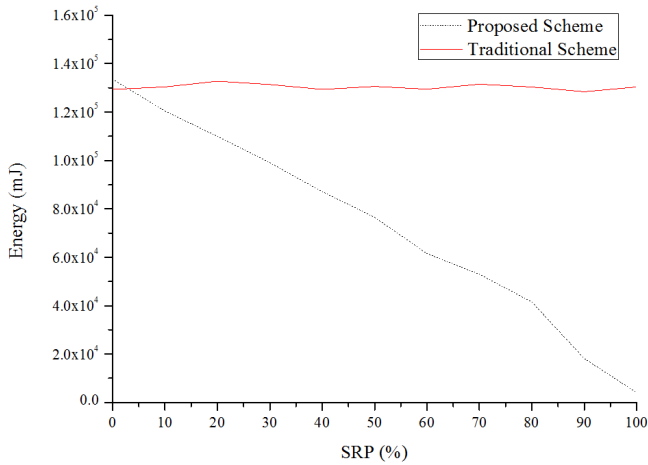
5. 성능평가

본 연구에서는 표1과 같은 환경에서 개발 및 실험하였다. 안드로이드에서 지원되는 언어는 Java이며 이를 위해 개발 도구로 이클립스를 사용하였다. 서버 환경도 Java 플랫폼으로 개발하였다.

<표 1> 개발 및 실험 플랫폼

S/W 플랫폼	Client	OS	Andorid 1.6
		Tool	eclipse
	De-Duple. Server	OS	Fedora Core 9
		Kernel	2.6.18
	Tool	eclipse, cubrid dbms	
H/W 플랫폼	CPU	Pentium 4 3.0 GHz	
	Memory	512 MB	
	Hard Disk	WD-1600JS(7200/8MB)	
	Network	myLGnet(WiFi)	
	Mobile	HTC Magic	

실험 데이터는 서버에 100Mbyte의 데이터를 저장해 놓고 모바일 단말에서 서버의 데이터와 10%단위로 0~100%의 중복을 보이는 100Mbyte파일을 전송하는 것이다. 총 5차례 실험하여 측정된 에너지 소모 값의 평균을 그림 2와 같은 그래프로 나타내었다.



[그림 2] 중복 데이터 비율 대비 에너지 소모

그림 2의 그래프에서 기존의 파일 전송의 총 에너지 소모와 CRC64를 적용한 에너지 총 에너지 소모를 중복율에 따라 나타내었다. CRC64를 적용하여 파일 전송을 한 경우 총 에너지 소모가 중복 율에 비례하여 감소하는 것을 확인할 수 있다. 중복 데이터가 5%이하 일 때부터 제안한 방법이 기존의 파일 전송 보다 에너지 소모가 줄어들고 있다. 이것은 해시를 구할 때 사용하는 에너지가 주로 CPU를 사용하고 데이터를 전송하는 에너지가 WiFi를 주로 사용하기 때문이다. 특히 해시를 변경하여 해시 계산에 소모되는 에너지를 줄이거나 듀얼 코어 같은 물리 장치를 활용하여 계산 속도를 향상 시킨다면 효율이 높아질 것으로 보인다.

6. 결론 및 향후연구

본 논문에서는 스마트폰에서 무선 네트워크를 통한 파일 전송을 수행함에 있어서 에너지 효율적인 파일 전송 모델을 제안하고 있다. 제안하는 기법은 CRC64를 이용한 중복 제거 작업을 수행하여 WiFi로 전송하는 데이터 용량을 줄여 에너지 소모를 줄이는 것이다. 구현된 플랫폼에서 5% 이상의 중복율을 가지는 데이터에 대해서 제안하는 방법이 기존의 파일 전송 기법보다 에너지 소모량을 급격히 줄일 수 있음을 보였다. 향후 연구로는 중복 데이터가 얼마나 포함되고 있는지를 예측하여 동적으로 데이터 전송 방식을 선택 할 수 있도록 연구할 계획이다.

참고문헌

[1] Lide Zhang, et al. "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," In Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis, CODE/ISSS '10, 2010, pp. 105-114.

[2] Bairoch A., Apweiler R. The SWISS-PROT protein sequence data bank and its new supplement TrEMBL. Nucl. Acids Res.24, 1996, pp. 21-25.

[3] A. Tridgell. Efficient algorithms for sorting and synchronization. PhD thesis, The Australian National University, 1999.

[4] QUINLAN, S., AND DORWARD, S. "Venti: a new approach to archival storage," In Proceedings of the 1st USENIX Conference on File and Storage Technologies (FAST), 2002.

[5] Athicha Muthitacharoen, Benjie Chen, and David Mazieres. A Low-Bandwidth Network File System. In Proceedings of the Symposium on Operating Systems Principles (SOSP'01), pages 174 - 187, 2001.

[6] C. Ungureanu, B. Atkin, A. Aranya, S. R. et al. HydraFS: a High-Throughput File System for the HYDRAsstor Content-Addressable Storage System. In FAST, 2010.

[7] C. Ungureanu, et al. HydraFS: a High-Throughput File System for the HYDRAsstor Content-Addressable Storage System. In FAST, 2010.

[8] plan9 home page, <http://plan9.bell-labs.com/plan9/>

[9] M. O. Rabin. Fingerprinting by random polynomials. Technical Report TR-15-81, Center for Research in Computing Technology, Harvard University, 1981.

[10] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of the cipher block chaining message authentication code. Journal of Computer and System Sciences, 61(3), 2000, pp. 362 - 399.