저전력 통신을 위한 에너지 효율적인 한글 압축 알고리즘

임근수 이세환° 고 건
서울대학교 컴퓨터공학부
{ksyim, trinite, kernkoh}@oslab.snu.ac.kr

An Energy-Efficient Compression Algorithm of Korean Language for Low-Power Communications

Keun Soo Yim Sehwan Lee° Kern Koh
School of Computer Science and Engineering, Seoul National University

요 약

모바일 컴퓨팅 장비에서 전송 데이터를 압축해 송수신하는 데이터의 양을 줄임으로써 궁극적으로는 통신에 사용되는 전력 소모를 줄일 수 있다. 본 논문에서는 이 기법을 활용하여 한글 데이터를 에너지 효율적으로 전송하는 기법을 제안한다. 제안하는 알고리즘은 한글의 표기 단위인 2 바이트 단위로 데이터를 압축하며 한글의 표기상의 특성을 활용하는 장점이 있다. 실험 결과 제안하는 알고리즘은 다양한 한글 데이터에 대해서 평균적으로 압축 효율을 약 5% 가량 증가시킨다. 이와 함께 제안하는 알고리즘은 실행 시에 사용하는 에너지가 비교적 적어 기존 알고리즘에 비해 한글을 보다 에너지 효율적인 방식으로 압축해 전송함으로써 모바일 장비의 소모 전력 측면의 효율을 증가시킬 수 있다.

1. 서론

최근 노트북, 휴대폰, 개인휴대단말기 등과 같은 모바일 컴퓨팅 장비들의 사용이 급증하고 있다. 모바일 장비들은 제한된 크기의 배터리를 사용해 전력을 공급 받기 때문에 사용시간을 연장하기 위해서 장비 내부의 구성 요소들을 에너지 효율적인 형태로 관리해야 한다. 이 중에서도 무선 통신 모듈의 경우 1 비트를 전송하는 데 드는 에너지가 CPU를 사용해 약 1000개의 명령어를 수행하는데 드는 에너지와 비슷한 특성이 있다 [1].

따라서 1000개 이하의 명령어를 사용해 송신하려는 데이터의 크기를 1 비트만 줄일 수 있다면 전체적으로 해당 모바일 컴퓨터의 총 전력소모량은 감소할 수 있음을 쉽게 인지할 수 있다. 하지만 실제로는 압축 알고리즘이 동작하는 동안에 CPU 내장캐시의 접근 실패가 발생해 주 메모리를 접근해야 하는데 한번 주 메모리 접근하는데 드는 에너지가 CPU에서 약 200개의 명령어를 수행하는데 드는 에너지와 비슷하기 때문에 압축 알고리즘을 내장캐시 접근 실패를 최소화하는 형태로 최적화할 필요가 있다. 그림 1에 제시된 것과 같이 역으로 기지국에서 모바일 장비로 데이터를 송신하는 경우에는 복원 알고리즘의 내장캐시 특성을 최적화할 필요가 있다.

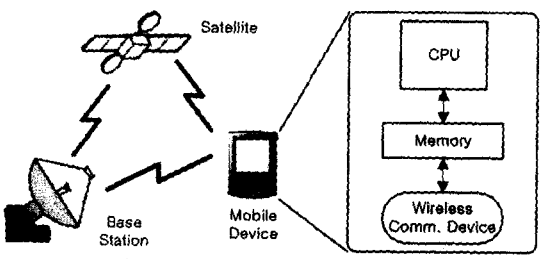


그림 1. 모바일 컴퓨팅 장비들의 통신 환경.

국내에서 사용하는 핸드폰과 개인휴대단말기와 같은 모바일 장비의 경우 주로 한글로 된 문자메시지나 문서파일 등을 전송하는 경우가 많기 때문에 데이터 압축 기법을 활용한 저전력 통신을 위해서는 압축 및 복원 알고리즘을 한글 데이터를 위해 최적화해야 할 필요가 있다.

본 논문에서는 자소 단위의 압축에 기반한 에너지 효율적인 한글 압축 및 복원 알고리즘을 설계하고 성능을 최적화 한다. 그리고 성능 평가를 통해 기존의 Ziv-Lempel (LZ) [2] 알고리즘을 사용하는 경우에 비하여 한글 데이터에 대해 압축 효율을 5% 가량 개선함을 보인다. 뿐만 아니라 다양한 CPU 및 무선 통신 에너지 효율을 가진 시스템에 제안하는 알고리즘을 적용했을 때 해당 시스템 전체 통신 에너지상의 효율을 분석 및 평가한다.

본 논문의 구성은 다음과 같다. 2장에서는 대표적인 무 손실 데이터 압축 알고리즘과 한글의 디지털 표기법에 대해 소개한다. 3장에서는 제안하는 에너지 효율적인 한글 압축 알고리즘을 설명한다. 4장에서는 실험 환경에 대해 설명한 후 이를 바탕으로 제안하는 알고리즘의 성능을 기존 알고리즘과 비교 분석해 평가한다. 그리고 5장에서는 결론을 맺는다.

2. 관련 연구

이 장에서는 대표적인 무 손실 데이터 압축 알고리즘들에 대해 살펴본 후 한글의 디지털 표기법의 하나인 조합형 한글의 장점과 세부 표기법에 대해서 살펴본다.

2.1. 압축 알고리즘

데이터 압축기법은 전송하려는 데이터의 크기를 줄여 통신 대역폭을 확장하거나 저장하려는 데이터의 크기를 줄여 기억공간을 실질적으로 확장하는데 널리 사용하고 있다. 데이터 압축기법은 압축하려는 데이터상에 자주 나타나는 기호를 이와 일대일로 대칭관계에 있는 다른 짧은 기호로 대치함으로써 압축된 데이터의

크기를 줄인다. 이때 원 데이터와 압축 데이터 사이에 관계를 표기하는 매핑 테이블의 고정 유무에 따라 크게 정적 압축 알고리즘과 동적 압축 알고리즘으로 분류한다.

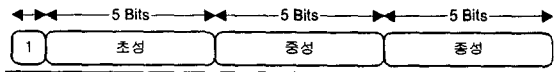
정적 압축 알고리즘의 대표적인 예로는 허프만 인코딩이 있다 [3]. 허프만 인코딩은 원 데이터를 한번 읽어 한 바이트 크기의 기호들의 발생 빈도수를 수집한 후 이를 역순으로 정렬하여 가장 자주 발생하는 기호를 가장 짧은 기호로 매핑하는 테이블을 구성한다. 그리고 이 테이블을 바탕으로 다시 한번 원 데이터를 읽으며 데이터를 압축된 기호를 사용해 표기함으로써 압축을 수행한다. 따라서 허프만 인코딩은 최적의 정적 압축 알고리즘이다.

하지만 원 데이터의 크기가 큰 경우에 각 부분에 따라 기호들의 발생 빈도수에 차이가 있을 수 있다. 따라서 적절하게 매핑 테이블을 동적으로 변환하는 동적 압축 알고리즘을 사용하면 일반적으로 정적인 압축 알고리즘에 비해서 더 좋은 압축 효율을 보일 수 있다. 뿐만 아니라 매핑 테이블이 압축하는 과정에서 동적으로 변화하는 형태로 관리되기 때문에 허프만 인코딩과 같이 부과적으로 매핑 테이블 생성을 위한 단계를 거치지 않아도 된다. 대표적인 동적 압축 알고리즘에는 Ziv-Lempel (LZ) 알고리즘이 있다 [2].

또한 주 메모리의 데이터를 효율적으로 압축하기 위해 비교적 간단한 방식의 X-Match 알고리즘이 고안되었다 [4]. X-Match 알고리즘은 메모리의 접근 단위인 4 바이트 단위로 데이터를 압축하며, 이를 하드웨어로 구현하면 매우 빠르게 동작하는 특성이 있다.

2.2. 한글의 디지털 표기법

한글을 디지털 코드로 표기하는 기법에는 크게 완성형 한글과 조합형 한글이 있다. 완성형은 음절 단위로 표기하는 기법이며 조합형은 음절 내부의 초성 중성 종성 단위 표기법이다. 완성형 한글은 총 1만 1천 1백 72자의 한글 가운데 8천여 자를 제외한 2천 3백 50자만 표기하고 있으며 그 구성이 한글의 입력 체계와 맞지 않다는 단점이 있다. 반면에 조합형 한글은 그림 2와 같이 각각 5 비트를 사용해 초성과 중성 그리고 종성을 표기하는 방식으로 완성형 한글의 두 가지 문제점을 갖고 있지 않다. 반면에 조합형 한글의 경우 한 글자에 해당하는 2 바이트 중에서 두 번째 바이트의 가장 상위 비트가 항상 '1'로 설정되지 않고 종성이 종류에 따라 '0' 또는 '1'로 설정되기 때문에 통신시에 제어 코드와 충돌을 일으킬 여지가 있다. 그럼에도 불구하고 본 논문에서는 보다 효율적인 체계를 가지고 있어 입력과 처리가 수월해 모바일 장비와 같이 작은 계산 능력을 가진 시스템에 보다 적합한 조합형 한글을 사용한다.



코드	초성	중성	종성	코드	초성	중성	종성	코드	초성	중성	종성
00000	-	-	-	01011	ㅅ	ㅋ	ㄻ	10110	-	ㅖ	ㅆ
00001	-	-	-	01100	ㅆ	ㅟ	ㄼ	10111	-	ㅗ	ㅇ
00010	ㄱ	-	ㄱ	01101	ㅇ	ㅗ	ㄽ	11000	-	-	ㅈ
00011	ㄲ	ㅏ	ㄲ	01110	ㅈ	ㅘ	ㄾ	11001	-	-	ㅊ
...	...	...	...	...	...	...	...	...	...	...	...

그림 2. 조합형 한글 표기법.

3. 제안하는 한글 압축 알고리즘

데이터를 압축해 전송하면 전송하는 데이터의 양을 줄임으로써 통신 에너지 소모를 줄일 수 있다. 에너지 통신 에너지의 감소 비율은 압축률과 밀접한 관련이 있다. 동시에 압축과 복원하는 과정에서 부과적으로 에너지를 소모하는데, 이때 소모되는 에너지의 양은 압축 알고리즘의 수행 시간과 내장캐시 접근 실패율과 관련이 있다. 따라서 저전력 통신을 위한 효율적인 알고리즘은 압축 효율은 높되 수행 시간이 짧고 내장캐시 접근 실패율이 낮아야 한다.

허프만 인코딩과 같은 정적 압축 알고리즘들은 부과적으로 매핑 테이블을 만드는 단계가 있어 동적 압축 알고리즘들에 비해서 수행시간이 오래 걸릴 뿐만 아니라 이렇게 생성한 매핑 테이블을 압축된 데이터와 함께 전송해야 한다. 이러한 특성은 위에서 제시한 저전력 통신을 위한 압축 알고리즘의 특성에 상반되는 것이기 때문에 정적 압축 알고리즘은 이에 적합하지 않다. 따라서 본 논문에서는 동적 압축 알고리즘을 바탕으로 효율적인 압축 알고리즘을 설계한다.

제안하는 알고리즘은 한글의 표기 단위인 2 바이트 단위로 데이터를 압축한다. 그림 3에 제시된 것과 같이 2 바이트 단위로 데이터를 읽어서 만약 해당 데이터가 사전에 존재하면 완전 일치로 고려하고 <'1'> + <"사전상의 위치"> + <"일치 유형"> 형태로 표기한다. 이때 사전상의 위치는 사전에 총 D 개의 데이터가 저장되어 있다고 하면 $\lg D$ 비트를 사용해 표기된다. 만약 사전에 128 개의 데이터를 저장할 수 있다면 이는 7 비트이다. 그리고 일치한 데이터를 사전상에서 최 상단으로 옮기고 그 사이에 위치한 데이터들의 위치를 하나씩 아래로 내린다. 이렇게 함으로써 최근에 나타난 데이터가 사전상의 가장 위에 위치하게 되어, 데이터의 공간적 지역성을 반영해 사전을 관리할 수 있다. 그리고 만약에 해당 데이터가 사전에 존재하지 않은 경우에는 <'0'> + <"원 데이터"> 형태로 표기해 1 비트의 손실이 발생한다. 이때 사전의 모든 항목은 한 단계씩 아래로 이동하고 최 상단에 해당 데이터를 새롭게 추가한다.

그리고 추가적으로 초성과 중성, 초성과 종성, 또는 중성과 종성만 일치하는 경우에는 부분 일치로 고려하여 <'1'> + <"사전상의 위치"> + <"일치 유형"> + <"일치하지 않은 부분"> 형태로 표기한다. 이때 일치 유형별 발생 빈도에도 큰 차이가 있기 때문에 이 자체를 정적 허프만 인코딩을 사용해 표기한다. 완전 일치의 경우 '1'로 표기하고 중성과 종성 부분 일치의 경우 '01'로 표기한다. 그리고 초성과 중성 부분 일치와 초성과 종성 부분 일치는 각각 '000'과 '001'로 표기한다. 이렇게 함으로써 가장 자주 발생하는 완전 일치에 맞춰 압축 알고리즘의 성능을 최적화할 수 있다.

즉, 제안하는 한글 압축 알고리즘의 가장 큰 특징은 한글의 표기 단위를 고려한 2 바이트 단위의 압축과 조합형 한글의 세부 표기법을 활용한 부분 일치 기법에 있다.

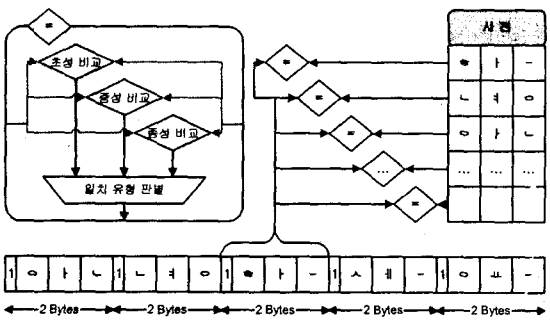


그림 3. 제안하는 한글 압축 알고리즘의 구조.

4. 성능 평가

이 장에서는 제안하는 압축 알고리즘의 다양한 성능을 널리 사용하는 무 손실 압축 알고리즘인 허프만 인코딩, Ziv-Lempel (LZRW), 그리고 X-Match 알고리즘과 비교해 평가한다. 그리고 이를 바탕으로 모바일 장치에서 제안하는 알고리즘을 사용해 한글을 압축해 무선 통신망을 사용해 전송하는 경우 해당 모바일 장치 전체 통신 전력 소모량을 분석한다.

압축 알고리즘의 효율을 평가하기 위해서 무 손실 압축 알고리즘의 성능평가에 널리 사용하는 Canterbury Corpus [5] 벤치마크를 영어 데이터로 표본으로 사용하였다. 그리고 한글 데이터 표본으로는 일반적으로 사용하는 신문, 논문, 소설, 그리고 노래 가사를 조합해 사용하였다.

그림 4(상)은 널리 사용하는 무 손실 압축 알고리즘의 압축 효율이다. 여기서 압축 효율은 압축된 데이터의 크기 분에 원 데이터의 크기를 의미한다. 실험 결과 기존의 알고리즘들은 일반 영어 데이터에 대해서 50-60%의 높은 효율을 보이는데 반하여 한글 데이터에 대해서는 70% 이상의 낮은 효율을 보임을 알 수 있다. 따라서 한글을 위한 고효율 압축 알고리즘이 반드시 필요함을 알 수 있다.

그림 4(하)는 제안하는 알고리즘의 압축 효율을 분석한 결과이다. 사전의 크기를 조정함에 따라 압축 효율에 영향이 있음을 알 수 있다. 세부적으로 사전의 크기가 128 개의 2 바이트 크기의 데이터를 저장할 수 있을 경우 압축률이 66.5%로 최적화된다. 기존의 알고리즘에 비해서 한글 데이터에 대한 압축 효율을 약 5% 가량 개선한 것이다. 역으로 제안하는 알고리즘은 일반 영어 데이터에는 기존 알고리즘에 비해서 낮은 압축 효율을 가짐을 알 수 있다.

즉, 제안하는 알고리즘은 한글 데이터에 대해 가장 좋은 성능을 보이며, 영어 데이터에 대해서는 LZRW1 알고리즘의 최적의 압축 효율을 보인다. 따라서 이 두 알고리즘을 각각 한글과 영어에 대해 상보적으로 사용할 필요가 있다. 이때 한글과 영어의 구분을 위해 한글의 첫 바이트의 최상위 비트가 항상 '1'로 설정되는 특성을 활용할 수 있다.

그림 5는 이상의 특성을 바탕으로 다양한 CPU 및 무선 통신 에

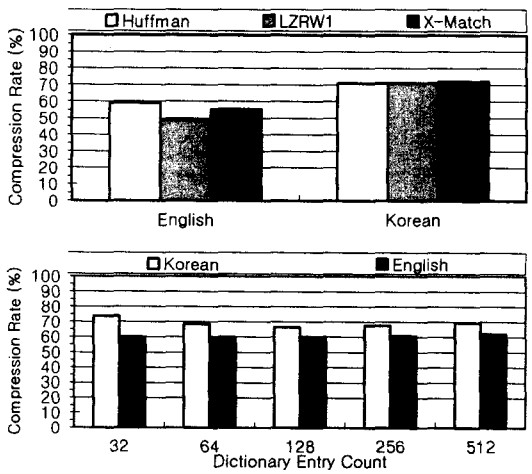


그림 4. 기존 알고리즘(상)과 제안하는 알고리즘 (하)의 압축 효율.

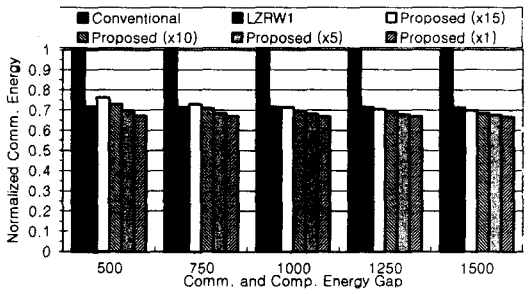


그림 5. 전체 통신 에너지.

너지 효율을 가진 시스템에 제안하는 압축 알고리즘과 LZRW1 알고리즘을 적용해 한글을 압축해 전송하는 경우 전체 통신 에너지 상의 효율을 제시한다. 그림에서 X축은 1 비트를 전송하는데 드는 에너지 분에 1 명령어를 CPU에서 수행하는데 드는 에너지를 의미하며, 모바일 장비에서 이 값은 일반적으로 485-1267 사이이다 [1]. 실험 결과 제안하는 알고리즘과 LZRW1을 사용해 데이터를 압축해 전송하는 경우 이를 사용하지 않은 경우에 비하여 통신 에너지를 29-34% 가량 감소시킨다.

이때 사용한 LZRW1은 수행 시간 및 내장캐시 접근 실패율 면에서 충분히 최적화된 모듈을 사용하였기 때문에 LZRW1과 비교한 제안하는 알고리즘의 상대적인 수행속도는 약 15-20배 가량이었다. 하지만 LZRW1과 기본 동작 원리가 비슷하기 때문에 구현 시에 동일한 최적화 과정을 거친다면 이와 대등한 수행 속도를 발휘할 수 있을 것으로 기대한다. 제안하는 알고리즘의 수행 속도가 LZRW1의 그것에 비하여 5배 가량 느린 경우에도 통신과 계산 사이에 에너지 차이가 1000인 경우에 제안하는 알고리즘을 사용하면 약 5% 가량의 통신 에너지 상의 효율을 높일 수 있다.

5. 결론

본 논문에서는 모바일 장비에서 저전력 통신을 위해 데이터 압축 기법을 활용하였다. 세부적으로 한글을 에너지 효율적인 형태로 압축하고 전송하기 위해서 새로운 한글 압축 알고리즘을 설계하고 성능을 평가하였다. 실험 결과 제안하는 알고리즘은 한글에 대한 압축 효율은 평균적으로 5% 가량 개선하며 수행 시에 요구되는 에너지가 상대적으로 적음을 보였다. 이러한 특성으로 인해 제안하는 알고리즘을 사용해 한글을 압축해 전송하는 경우 모바일 장비의 통신에 사용되는 에너지를 29-34%가량 개선함을 보였다.

참고 문헌

[1] K. Barr and K. Asanovic, "Energy Aware Lossless Data Communication," *In Proceedings of the 1st USENIX International Conference on Mobile Systems, Applications, and Services*, pp. 231-244, 2003.

[2] D. A. Lelewer and D. S. Hirschberg, "Data Compression," *ACM Computing Surveys*, Vol. 19, No. 3, 1987.

[3] D. A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," *In Proceedings of IRE*, Vol. 40, pp. 1098-1101, 1952.

[4] M. Kjelso, M. Gooch, and S. Jones, "Design and Performance of a Main Memory Hardware Data Compressor," *In Proceedings of the EuroMicro Conference*, IEEE Computer Society Press, pp. 422-430, 1996.

[5] R. Arnold and T. Bell, "A Corpus for the Evaluation of Lossless Compression Algorithms," *In Proceedings of the 7th IEEE Data Compression Conference*, pp. 201-210, 1997.