

論文2000-37TE-3-12

# 실시간 전송시스템의 무손실 압축을 통한 전송효과 연구 (Transaction Effect Analysis through Compressing Realtime Transfer system)

朴仁淳\*, 南相曄\*\*, 朴仁政\*\*\*

(In-Soon Park, Sang-Yep Nam, and In-Jung Park)

## 요약

한정된 용량의 전송선로에서 데이터를 고속으로 전송하기 위해서는 데이터의 압축이 필요하다. 압축은 데이터의 양을 줄이면서 전송선로를 통한 데이터 전송은 빠르고 비용이 절감되도록 만들어 준다. 본 논문에서는 데이터 압축 소프트웨어나 하드웨어를 이용하여 데이터 통신의 응용에 어떻게 적용하는 과정을 전개하고자 한다. 본 논문에서는 이 적용 과정에서 문제 시 되는 시스템이 원하는 효율과 시스템의 지연 문제를 해결을 위한 방안을 마련하고자 무손실과 다중이력을 지원하는 LZS 압축시스템을 이용하여 실시간 데이터 압축 전송 시스템을 구현하여 효율을 분석하고자 한다.

## Abstract

In the communication technology, there is necessary data compression to transfer data more quickly and bigger amount in the same condition. And that should bring down the cost consequently. This paper shows that how to apply compression software and hardware to attain the issue that mentioned previous in data communication technology as well as, trying to find the solution for the system delay problem and to improve efficiency demanded to them. To achieve these purposes, I will study and analyze the efficiency of compression system to realize realtime data compression transfer system using LZS compression algorithm.

## I. 서론

최근 정보 압축 기술은 통신미디어, 방송미디어, 저장 미디어에 있어서 멀티미디어 시대를 창조하기 위한 핵심 기술입니다. 이에 의해 경제성, 격납 공간의 축소, 고품질 전송, 다채널화 등이 가능해진다.

멀티미디어 환경에 요구되는 컴퓨터의 성능은 영상 및 음성 등의 압축, 신장을 어느 정도 실현되고 있지만 품질에 있어서는 만족할 만한 수준이 못된다. 그러므로 멀티미디어 정보로서 가장 중요한 음성, 영상 등의 압축, 신장의 처리에 필요한 연산량과 그 실현 방법은 MIPS, MOPS이고 전송 데이터의 입출력 관계는 MIPS로 나타낸다. 이 비율은 특정 디바이스에 비례하여 얻어지는 압축비율에 상관없이 상수치로 표현된다.

압축에는 유손실 압축과 무손실 압축이 있는데 유손실 압축은 주로 JPEG나 MPEG와 같은 압축을 나타내고 본 논문에서는 정확한 데이터 정보를 유지하면서 압축하는 문서일 경우에는 무손실 압축이 필요하게 되므로 본 논문에서는 무손실 압축과 다중이력을 지원하는 LZS 압축 알고리즘을 이용하여 실시간 데이터 압축 시스템인 모토로라 MPC860-40Mhz 시스템을 구현하여

\* 正會員, (株) 하나기연

(Hanakiyeon)

\*\* 正會員, 京文大學 情報通信科

(Dept. of Information & Communication, KyungMoon College)

\*\*\* 正會員, 檀國大學校 電子工學科

(Dept. of Electroniucal Engineering DanKook University)

接受日字:2000年3月30日, 수정완료일:2000年9月16日

효율을 개선 하고자 한다.

## II. 압축동작 개요 및 알고리즘의 분석

압축방식에는 동기식과 비동기식이 있는데 동기식은 비동기의 압축 문제점인 부호화 되는 기호들 사이의 시간적 지연을 최소화하여 데이터 처리량을 최대화하고 기호들 사이의 중복성을 제거하여 효율적으로 데이터를 전송하는 방법이다.<sup>[1]</sup>

LZS알고리즘의 기본이 되는 LZC에 대하여 동기식 압축이 적당한지를 분석해 본다. LZC는 정보원의 엔트로피에 근사적으로 접근하고 구현의 용이성과 부호화 속도와 효율성 때문에 컴퓨터상의 표준 알고리즘으로 채택하고 있다. 이 기법의 특징은 입력기호 낱말의 확률로 압축하지 않고 적응 적으로 부호책을 형성하여 부호책의 문자열과 입력기호에 대해 문자열 맞추기를 수행하여 맞춤된 문자열을 하나의 부호어로 압축을 수행한다. 이러한 부호책을 기반으로 하는 압축기법은 두 가지로 구분되는데 첫 번째로 일정한 길이의 창을 계속 옮기면서 그 창안의 기호들로부터 시작되는 문자열을 부호책의 요소로 활용하는 기법이고 두 번째는 부호책을 입력데이터의 처음부터 마지막까지 모두 저장하며 사용하는 기법이다. 압축효율 성능 분석 면에서 보면 압축엔진에 의해 얻어지는 압축비율이 완전히 데이터 그 자체에 종속 되기 때문에 어느 압축된 데이터가 압축이나 신장 엔진으로 들어가고 나오는가 하는 것은 예측할 수도 없고 항상 일정한 값을 얻을 수 없다. 이 비율은 얻어진 지속적 압축비율에 의해 나누어진 열 데이터 율의 몫으로서 결정된다.

데이터 통신 응용에서 통신채널을 통하여 전송되는 데이터는 압축할 수 있다. 전송된 데이터는 LZS엔진을 통해 입, 출력되는 압축된 데이터의 대역폭을 결정될 수 없기 때문에, 만일 데이터 비율을 모른다면 전송되는 데이터를 파이프라인에 채워서 전송하는 설계 문제가 발생하게 된다.<sup>[2]</sup>

이 문제에 대한 해법은 아래와 같이 구현할 수 있다.

- 1) 데이터 통신의 데이터 양에서 데이터 세트의 평균 압축비율을 결정한다.
- 2) 샘플 데이터 세트에서 가장 좋은 압축을 결정하여 설계한다. 예로서 전체 데이터 세트의 평균 압축비율이 2:1이고 표본 데이터 압축비율에서 가장 높은 비율이 3:1 이하이라면 설계의 가장 알맞은

압축비율은 3:1로 정한다.

- 3) 가장 낮은 압축된 데이터 비율을 구하기 위해 가장 좋은 압축엔진에 명시된 데이터 비율로 나눈다.
- 4) 전형적인 데이터 통신의 데이터양은 LZS데이터 압축을 이용한 WAN의 평균 전송률이 어떤 곳에서는 2.5:1에서 3:1이 된다. WAN에 의해 송수신된 데이터는 대부분 4:1 이상으로 떨어진다.
- 5) 상기 2)에서 이것은 일반적인 데이터 세트에 기초로한 실제 압축비율을 결정하여야할 필요가 있다.

데이터 파이프는 단순히 전체 데이터 파이프 비율로 데이터를 송수신 할 것이다.

설계에서 명시된 비율보다 큰 패킷의 압축비율로 얻어지는 결과는 압축엔진의 내부나 외부에서의 압축된 데이터의 데이터 비율은 데이터 통신채널보다는 작을 것이고, 순간적으로 파이프 안의 거품을 만들 것이다. 그렇지만 데이터가 아주 높은 비율로 압축되어 있기 때문에 전체적인 통신 채널의 압축비율은 만일 아무 압축이 얻어지지 않았을 때보다도 더욱 크게될 것이다. 사실 64Kbps의 채널에 3:1로 설계를 하였고, 데이터를 4:1로 압축을 하였으면, 전체적인 데이터 전송량은 파이프 안 거품이 있더라도  $64\text{Kbps} \times 3:1 = 192\text{Kbps}$ 가 될 것이다. 사실, 데이터 채널의 최대 대역폭은 압축되지 않은 채널의 대역폭에 설계를 원하는 압축비율을 곱한 것이 된다. 예를 들어, 설계에서 3:1의 최대 압축비율을 설정하였고, 압축비율이 이 비율에 못 미치게 압축되면, 데이터 채널의 대역폭은 파이프 안 거품이 있더라도 아직 압축되지 않은 데이터의 3배가 된다.

LZS 알고리즘은 LZC알고리즘에 다중이력과 무손실을 추가한 것으로 입력 데이터 문장의 과도한 데이터 스트림을 찾아서 이 스트림을 출력 데이터 스트림에 작은 길이의 기호화된 토큰을 대체하고 입력 문장으로부터 데이터의 포인터로 구성된 것과 일치하는 이 스트림의 테이블을 만든다.

압축 프로그램의 동작 원리는 긴 중복되는 문장을 짧은 하나로 대체하고 토큰으로 대체된 문자의 열로 복사된다. LZS 알고리즘에서 반복되는 문자는 2,000문자(2048 byte) 정도 이내에 있어야 한다. 그렇지 않으면 그들은 압축되지 않은 데이터로 전달될 것이다. 2000 문자인 이유는 많은 양의 글들은 더 많은 압축을 생성

한다. 그래서 커다란 윈도우는 더 많은 압축비율을 갖는 반면 압축 속도는 매우 느리게 된다. 압축의 주요 요소는 전송속도를 증가시키는 것이다. 그리고 컴퓨터 네트워크 안에서 압축은 매우 빠르게 전달되어야 하기 때문에 수행속도는 매우 중요한 문제다.<sup>[3][4][5]</sup>

압축효과와 압축하는데 필요한 시간을 위해 두개의 이중화된 문자의 페이지는 잘 압축할 수 있다. 데이터의 중복되는 문장을 반복해서 찾는 것은 더욱 압축비율을 높일 수가 있으나 압축 속도를 떨어뜨린다.

표 1. LZS 와 WINZIP의 압축비율 비교  
Table 1. Comparison Compression rate "LZS Vs. Winzip"

압축 엔진	원 크기	압축된 크기	압축비율	압축시간
LZS	785KB	458KB	1.7:1	1초
WINZIP	785KB	322KB	2.4:1	5초

표 1에서 LZS와 PC에서 가장 많이 이용되는 WINZIP으로 텍스트 데이터를 압축했을 때 압축비율과 압축 속도의 차이를 비교해 보았다.

표 1에서 WINZIP은 압축크기를 작게 만들 수 있으나 압축속도는 5배가 더 걸렸다. 압축에 걸리는 시간은 문제가 되지 않을 것이나 인터넷에서 속도는 가장 중요한 요소이다.

데이터 스트림은 전화선의 최대 전송률로 이동하여야 한다. T1 전송선로를 이용할 때는 1.54Mbps의 속도로 데이터를 전송할 것이다. 데이터 거품이 데이터 스트림 안에서 빠르게 만들어 질 것이고 이 데이터 거품은 T1 수용능력을 소비할 것이며 전송비용의 증가를 가지고 온다.<sup>[6]</sup>

자주 반복되는 문서는 그렇지 않은 것 보다 더욱 많이 압축될 수가 있다. 전자우편과 HTML 언어에 의해 만들어진 웹 페이지는 가장 많이 압축될 수 있다. 다른 말로 암호화된 데이터 즉 난수로 이루어진 페이지와 유사한 것은 전혀 압축할 수가 없다. 또한 먼저 압축된 데이터는 때로는 다시 압축할 수가 있으나 그 이상의 압축효과를 얻을 수는 없다.<sup>[5]</sup>

### III. LZS 데이터 압축 및 적용

LZS 알고리즘은 입력 데이터 스트림의 장황한 데이

터 스트림을 찾아서 이 스트링을 출력 데이터 스트림에 작은 길이의 기호화된 토큰으로 대체한다. LZS 알고리즘은 입력 스트림 으로부터 데이터의 포인터로 구성된 것과 일치하는 이 스트링의 테이블을 만든다. 잡다한 스트링을 대체하기 위하여 사용되는 인코딩된 토큰은 이 테이블 안의 정보로부터 만들어진다. 이런 방법으로 앞으로의 데이터는 앞질의 데이터에 기반하여 만들어진다.<sup>[6][7]</sup>

이것은 입력 스트림에서 더 많이 반복되는 데이터는 더 높은 압축비율을 만들고, 반대로 스트림에서 많은 랜덤한 데이터는 더 낮은 압축비율을 이룬다.

압축 엔진은 입력 데이터의 마지막 2Kbyte를 가지고 있는 압축 이력의 압축동작을 가속하기 위해 다른 데이터 구조처럼 관리한다. 이 이력은 압축엔진에서 토큰을 만들기 위해 부합되는 스트링을 찾기 위해 사용한다. 유사하게 데이터 링크의 마지막의 신장 엔진은 출력데이터의 마지막 2Kbyte를 신장 이력으로 관리한다. 그러므로 압축이력에 의해 관리되는 입력데이터의 마지막 2Kbyte는 신장 이력에 의해 관리되는 출력데이터의 마지막 2Kbyte와 동등하여야 한다.<sup>[5]</sup>

신장 엔진은 이 이력을 압축엔진에 의해 제공되는 토큰으로부터 열 데이터를 다시 생성하기 위해 사용한다. 각 데이터 링크의 끝의 압축과 신장이력은 맞아야 하고, 그렇지 않으면 신장기는 토큰에 의해 가리키는 쓰레기를 출력할 것이다.

압축된 스트림 데이터는 문자상의 데이터와 압축된 토큰으로 구성된다. 문자상의 데이터는 압축되지 않은 입력 데이터 스트링이다. 압축된 토큰은 합치하는 스트링을 가지고 있는 이력의 위치에 대한 읍셋을 가지고 있고, 이 스트링이 합치하는 바이트의 개수의 길이를 가지고 있다.

압축비율은 2개의 핵심요소로부터 이루어진다 : 압축되는 실제 데이터와 스트링이 맞추기 위해 어떤 양의 검색에 의해서 이루어진다. 이것에서 압축은 강력한 스트링 검색과 일치되므로 구성된다. LZS는 압축비율이 실행되는 일정량의 검색을 조정함으로써 압축 속도를 위해 압축비율을 조정 가능한 형태를 가지고 있다.<sup>[5]</sup>

신장은 아주 단순하며 빠른 알고리즘이다. LZS신장 알고리즘은 압축된 데이터와 토큰을 위해 한번의 검색을 구현한다. 신장기는 시작을 읽고, 위치로 이동 한 후 그 읍셋에서 출발하는 바이트의 길이를 출력한다. 이것

은 압축보다 신장이 빠르고 조정 가능한 형태를 가지고 있지 않다.

각 압축 동작의 마지막에서, LZS 엔진은 데이터를 출력한다. 출력은 LZS 엔진에 구성된 데이터와 마지막 표시를 출력하는 것으로 구성된다. 데이터는 긴 스트링이 합치하는 중간에 위치하기 때문에 LZS 엔진 안에 위치한다. 마지막으로 이 스트링 합치 동작을 끝낸다. 마지막 표시는 하나의 토큰이며 신장기가 압축된 데이터의 마지막을 찾는 데 이용된다.<sup>[5]</sup>

#### IV. 압축전송 시스템의 구현

LZS 압축 알고리즘의 기능을 구현하기 위하여 다음 변수를 결정하였다.

1. 효율과 시스템에 요구되는 압축 신장율(일반적으로 전이중 WAN 속도).
2. 압축비율 분배와 평균/지정
3. CPU의 성능과 가용한 버스 대역폭

이 데이터로 라우팅을 위한 데이터 압축엔진을 가장 최적으로 구현 할 수가 있다. 본 논문에서는 이 모든 라우팅 과 포워딩 기능을 모토로라 MPC860-40MHz CPU PPC시스템에서 구현하였다. MPC860은 WAN이나 LAN을 연결할 수 있는 통신채널을 4개 가지고 있기 때문에 본 시스템에 가장 적절히 사용할 수 있었다. 이 실험을 위해 10Mbps 이더넷 포트와 단일 128K WAN포트로 구성되어 있다. WAN 연결이 설정된 상태에서 CPU는 LAN 데이터 패킷을 10Mbps로 송수신하고, WAN 데이터를 128Kbps로 송수신 한다.<sup>[1]</sup> 그리고 압축기로는 범용 PCI 버스에 연결할 수 있는 LZS압축 알고리즘을 이용한 9751압축기를 이용하였다.

압축프로세서의 PCI밴드 폭 점유를 유추하기 위해 POLLING FREQUENCY영역을 세팅하는 방법이다.

*Bandwidth used*

$$\begin{aligned}
 &= \frac{\text{average buest size(bytes)}}{\text{burst period (sec onds)}} \\
 &= \frac{\text{average buest size(bytes)}}{\text{PCI clock period(sec/clock)* clocks per burst period(clock)}} \\
 &= \frac{\text{average buest size(bytes)}}{\text{PCI clock period(sec/clock)* POLLING FREQUENCY Value* 16 PCI clock}} \\
 &= \frac{64(\text{bytes})}{30 * 10^{(-9)} (\text{sec/clock}) * 44 * 16 \text{clocks}} \\
 &= 3.03 \text{Mbytes/sec}
 \end{aligned}$$

다음은 2개의 E1연결에서 PCI 버스 밴드 폭을 구해

본다.

$$\begin{aligned}
 &[\text{\#of ports}] \times [\text{WAN speed(Mbps)}] \times [2 \text{ for full-duplex}] \\
 &= 2 \times 2.48 \text{ Mbps} \times 2 = 8 \text{Mbps}
 \end{aligned}$$

$$\begin{aligned}
 &2:1 \text{의 압축율은 다음과 같은 LAN 속도를 구현한다.} \\
 &[\text{WAN speed (Mbps)}] \times [\text{compression ratio}] \\
 &8 \text{ Mbps} \times 2 = 16 \text{Mbps}
 \end{aligned}$$

그러므로 PCI 버스에서는 33Mbps의 데이터 전송을 하기 때문에 위의 조건을 만족시킨다.

압축엔진은 LAN에서 WAN으로부터의 데이터를 압축한다. 그리고 신장엔진은 WAN에서 LAN으로부터의 데이터를 신장한다. 2:1의 압축 율에서 압축기의 소스 밴드 폭은 8Mbps 이고 목적 밴드 폭은 4Mbps이다. 유사하게 신장기의 소스 밴드 폭은 4Mbps이고 목적 밴드 폭은 8Mbps이다.

본 연구에서 데이터 압축은 B채널 둘 다 지원하고 2:1의 평균 데이터 압축비율을 가진다. 압축비율은 3:1로 하였다. 이 원칙은 파이프 안에 거품 없이 주어진 효율 레벨에서 압축과 신장을 할 수 있다. 이것은 3:1이나 4:1의 압축 비율에서도 동작을 할 수 있다.

라우터의 특성은 다음과 같다.

1. 단일 시스템의 패킷 버스 와 메모리
2. 각 B 채널은 64Kbps
3. PRI당 2개의 B채널

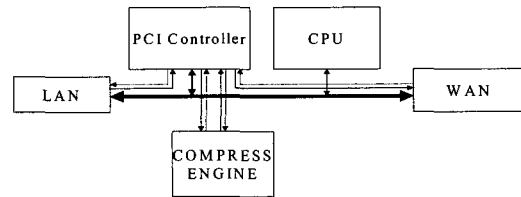


그림 1. 전송되는 데이터 경로  
Fig. 1. Data transferred Path

라우터 내의 데이터의 흐름을 그림 1에 나타낸다.

전송된 데이터는 그림 1의 실선과 같은 방향으로 동작한다. 데이터는 LAN 인터페이스에서 수신되어 패킷 메모리로 전달된다. CPU는 만일 압축 동작이 패킷에 실행되는 것이 필요하면 압축을 위해 압축엔진으로 전송한다.

데이터가 압축된 후에는 패킷은 메모리로 놓여지고 그때 WAN 포트 로 전달된다. 압축 엔진의 목적은

WAN인터페이스에 WAN 파이프가 차도록 충분히 압축된 데이터를 제공하는데 있다.

그리고 수신된 데이터는 그림 1의 점선과 같은 방향으로 동작한다.

데이터는 WAN 인터페이스에서 수신되어 패킷 메모리로 전달된다. CPU는 패킷이 신장이 필요한가를 관리하고 이 패킷을 압축엔진으로 신장을 위해 전달한다. 데이터가 신장된 후에는 패킷은 메모리로 다시 놓여지고 다음에는 LAN포트로 내보내 진다.

일반적인 데이터 통신 환경에서, 채널은 전이중 또는 반이중 모드가 될 수 있다. 예를 들어 T1 전송로에서 데이터는 동시에 1.54Mbps로 송수신을 할 수 있다.

모든 압축엔진은 반이중 모드 엔진이다. 이것은 이들이 동시에 압축과 신장을 할 수 없다는 것을 의미한다. 라우터는 데이터를 압축엔진이 시분할 다중형태로 전송과 수신 양쪽의 데이터를 처리할 수 있도록 버퍼링을 하여야 한다. 데이터 압축 엔진은 시간의 한 부분에서 압축과 신장을 하여야 한다. 결과적으로 의사적인 전이중 압축 신장율을 얻을 수가 있다.

전이중 압축비율의 시간평균과 주기를 통한 특정 데이터의 신장에 의한 데이터율은 수학적으로 얻을 수가 있다. 시간 평균의 공식은 다음과 같이 구해진다. 1Mbps, 1초, n바이트의 전이중 데이터 압축률은 압축엔진에 의해 압축될 것이고, b바이트가 신장될 것이다. 만일 반이중 신장율이 D-Mbps이고 반이중 압축비율이 C-Mbps이면, n바이트를 압축하고 n-바이트를 신장하는 시간은 다음과 같이 된다.

$$1\text{sec} = n\text{-Mbytes}/C\text{-Mbps} + n\text{-Mbytes}/D\text{-Mbps} \quad (4.1)$$

주기를 통한 전이중 압축과 신장 방정식  
따라서 전이중, 시간 평균된 압축엔진의 밴드폭은

$$n\text{-Mbps(FDX)} = 1/(1/C\text{-Mbps} + 1/D\text{-Mbps}) \quad (4.2)$$

### V. 압축 전송시스템의 성능분석

실제 원칙을 보면 엔진은 최소한 2×64Kbps=128Kbps 정도의 전 이중 압축과 신장을 지원하여야 한다는 것을 의미한다. 이것은 1초의 주기에 압축 엔진은 128Kbits의 압축된 데이터를 WAN으로부터 입력을 하여 256Kbps의 신장된 데이터를 출력하여야 하고 마찬가지로 256Kbits의 열 데이터를 LAN으로부터 입력하

여 128Kbits의 압축된 데이터를 출력하여야 한다.

MPC860 시스템에서 LZS 압축엔진은 데이터를 1.248Mbps로 압축하고, 2.352Mbps로 데이터를 신장한다. 공식(4.2)로부터 이것은 CPU 리소스를 100% 사용하여 전이중 데이터효율이 1/(1/1.25Mbps + 1/2.35Mbps) = 816Kbps를 산출한다. 따라서 전이중 압축 신장을 256Kbps에서 보이기 위해서는 256Kbps/816Kbps = 31.4%의 전체 CPU 대역폭이 필요하다. 이것은 69.6%의 CPU대역폭을 라우팅 작업을 위해 남겨둔다.

WAN 링크에 압축을 추가함은 파이프의 대역폭을 증가시키고, 따라서 패킷을 처리하거나 라우팅 하기 위해 CPU의 능력증대가 요구된다. 예를 들어, 만일 압축 비율이 2:1이고 128Kbps 데이터 경로이면 이제 효과적으로 256Kbps경로가 된다. 이것은 압축엔진 자체를 위해 디바이스가 사용되지 않을지라도 라우팅 CPU의 요구를 증대시킨다.

압축을 적용한 라우팅 응용에서 압축은 CPU MIPS의 60%를 점유했고 라우팅은 나머지 40%를 점유했다. 이 비율로부터 만일 라우터가 MPC860의 전체 사용할 수 있는 MIPS의 31.4%를 사용하면, 그리고 이것에 전체 리소스의 60%가 요구된다면 라우팅은 (40/60)×31.4% 즉 20.6%의 CPU MIPS를 쓰게 된다.

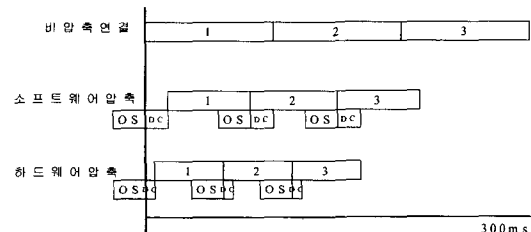


그림 2. 압축과 비압축의 데이터 전송률 비교  
Fig. 2. Comparison between Decompression and Compression data transferred rate.

이 공식에 기초해 설계는 전체의 52%(52.0% = 31.4%+20.6%)를 이용한다. 따라서 단일 MPC860프로세서가 라우팅과 압축을 구현할 수가 있다.

그림2에서 세 개의 일반 패킷을 WAN상에서 전송하는데 걸리는 시간을 비교하였다. 그래프는 비압축 전송하는데 걸리는 시간을 소프트웨어 압축과 하드웨어 압축을 함께 비교하였다. 비압축 연결에서는 원래 데이터 전송의 그대로의 시간이 걸렸다. 그리고 소프트웨어 압

축에서는 사이즈가 줄어든 만큼의 시간이 감소했다. 그리고 하드웨어 압축에서는 압축에 걸리는 시간이 아주 짧기 때문에 가장 적은 시간으로도 데이터를 전송할 수 있다. 이것은 라우팅에 의해 야기된 것이고, O/S에 의해 걸린 시간을 가리킨다. 그런데 처음 패킷 후에 지연은 다중패킷의 전송 안으로 감추어지고 있다. 또한 소프트웨어 압축은 이 지연을 증가시키고 있으나 이것은 전송시간에 감추어진다.<sup>[8]</sup>

다음과 같은 환경을 PPC시스템에서 구축하였다. 프로세서는 MPC860-40MHz CPU를 사용하였고, Line speed는 128Kbps, 압축비율=2:1, 데이터비율=256Kbps로 정했다.

운영체제 점유율은 전체 CPU시간의 20.6%를 점유하고 데이터 압축 점유율은 전체 CPU시간의 31.4%이다. 시간 평균의 공식은 공식(4.1) 주기를 통한 전이중 압축과 신장 방정식과 공식(4.2) 전이중 압축 신장률을 사용하여 그림 3의 결과를 나타냈다.

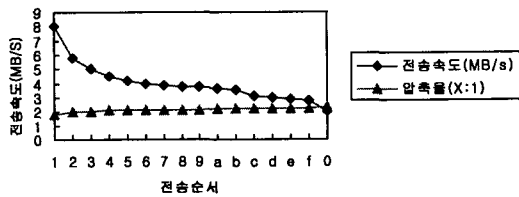


그림 3. 압축율과 전송속도 비교  
Fig. 3. Comparison between Compression rate & Transfer speed.

그림3은 텍스트 파일을 압축하였을 때 압축효율과 전송속도에 대한 비교를 보인다.

그림3에서 전송순서 1인 경우 전송속도 8Mbps이고 상대적인 압축율이 1.8 : 1, 전송순서가 2인 경우 전송속도는 5.8Mbps이고 상대적인 압축율이 2:1임을 알 수가 있다. 그러므로 전송속도가 빠르면서도 낮은 압축율을 보이고 전송속도가 느리면서도 높은 압축율을 가짐을 알 수 있다. 일반적으로 압축속도와 압축율 사이에는 대조적인 규칙이 있다. 빠른 셋팅에는 낮은 압축율을 가지며 대조적으로 낮은 세팅은 높은 압축율을 가진다.

## VI. 결 론

본 논문에서 보듯이 데이터 통신상의 데이터 압축을

완벽하게 구현하였을 때에는 대역폭을 충분히 개선시킬 수 있었다. 이것은 다른 응용을 추론할 수가 있다. 단순히 데이터 채널이 공급할 수 있는 최대 통신 대역폭을 결정함으로써, 일반적인 데이터 흐름을 위한 적절한 압축비율을 결정할 수 있으며 사용 가능한 CPU를 벤치마킹 할 수가 있었다.

데이터 압축은 그 비용에 비해 높은 효율을 제공한다. 이 기술을 어떻게 동작하고 그래서 어플리케이션이 가장 효과적으로 이용하는 것을 파악하는 것이 필요하다. 또한 몇 가지 기본적 원리와 데이터 압축에 대한 시스템의 응용을 분석하였다. 이러한 압축 시스템을 ADSL등에 적용할 때에는 더욱 효율적인 전송을 할 수 있다. 그리고 본 논문에서는 단일 채널을 이용한 압축 전송에 대한 실험만을 하였으나, 추후에 2채널 및 4채널에 대한 실험을 하여 단일 시스템에서 가장 높은 효율을 내는 것을 연구할 계획이다.

## 참 고 문 헌

- [1] Precise/MQX Realtime-Executive, Precise 1999.
- [2] J. Ziv, A.Lempel, "A universal algorithms for sequential data compression," IEEE Trans. on Information Theory. Vol. IT-23, pp.337-343, 1977.
- [3] J. Ziv, A.Lempel, "Compression of individual Sequence via variable-rate coding," IEEE Trans. on Information Theory. Vol. IT-24, pp. 530-536, 1978.
- [4] T. C. Bell, J. H. Cleary, I. H. Witten, Text Compression. New Jersey: Prentice Hall, 1990.
- [5] American National Standards Institute, Inc., "Data Compression Method for Information Systems," ANSI X3.241-1994, August 1994.
- [6] Shacham, A., "IP Payload Compression Protocol (IPComp)", RFC 2393, December 1998.
- [7] Rand, D., "The PPP Compression Control Protocol (CCP)", RFC 1962, June 1996.
- [8] Schneider, K., and R. Friend, "PPP LZS-DCP Compression Protocol (LZS-DCP)", RFC 1967, August 1996.

저 자 소 개

朴 仁 淳(正會員)

1967年11月 18日生. 1993年 2月 인천대학교 공과대학 전자공학과(학사). 2000年 2月 단국대학교 대학원 전자공학과(석사). 1995年 11月~1997年11月 (주)프로칩스. 1997年 11月~2000年 현재 (주)하나기연 대표. 주관심 분야 : 멀티미디어 신호처리, 컴퓨터통신, 인터넷라우터, 이동통신, 실시간처리시스템 등임



朴 仁 政(正會員) 電子工學會誌 第26卷 第7號

1948年1月 28日生. 1974年 2月 고려대학교 전자공학과(학사). 1980年 9月 고려대학교 대학원 전자공학과(석사). 1986年 2月 고려대학교 대학원 전자공학과(박사). 1981年 3月~2000

年 현재 단국대학교 교수. 1988年 10月~1989年 9월 Bowling Green주립대 객원교수. 1995年 1月~2000年 현재 대한전자공학회 교육이사. 1999年 1月~2000年 현재 대한전자공학회 멀티미디어 연구회 전문위원장. 1999年 3月~2000年 현재 한국XDSL포럼 의장. 주관심 분야 : 멀티미디어 신호처리, 컴퓨터통신, 이동통신, 인터넷방송. 실시간처리시스템, 인터랙션텍트, 정보보호기술 등임



南 相 曄(正會員)

1958年12月 2日生. 1982年 2月 단국대학교 공과대학 전자공학과(학사). 1984年 2月 단국대학교 대학원 전자공학과(석사). 2000年 2月 단국대학교 대학원 전자공학과(박사

수료). 1984年 1月~1992年 1月 삼성종합기술원 정보시스템연구소 주임연구원. 1992年 1月~1998年 3月 모토로라반도체통신(주) 기술연구소 차장. 1998年 3月~2000年 현재 경문대 정보통신과 교수. 주관심 분야 : 멀티미디어 신호처리, 컴퓨터통신, 마이크로프로세서, DVD, CD-R/W, D-TV, STB, 멀티미디어콘텐츠, 전자상거래, 3D 그래픽/애니메이션, DSP, 영상/음성인식, 인터넷방송/TV