WAN 실시간 전송시스템의 압축을 통한 전송효과 분석

박 인 순(朴 仁 淳). 박 인 정(朴 仁 政)
단 국 대학교 전자 공학과
전화 : 0417-550-3544

Transaction effect analysis through Compressing WAN Realtime Transfer system

In Soon Park. In Jung Park
Department of Electronic Engineering Dan Kook University
email : timemate@anseo.dankook.ac.kr

Abstract

In storage technology it is desirable to have greater storage capacity at lower costs. Data compression addresses these demands by reducing the amount of data that must be stored to a given size of media, thus lowering the cost of that storage device.
In data compressions it is desirable to have faster transfer rates at lower costs. Data compression addresses these demands by reducing the amounts of data that must be transferred over a media with a fixed bandwidth, thus reducing the connection time. Data compression also reduces the media bandwidth required to transfer a fixed amount of data with a fixed quality of service, thus reducing the costs on this service.

I.서론

고정된 용량의 전송선로에서 데이터를 고속으로 전송하기 위해서는 데이터의 압축이 필요하다. 압축은 데이터의 양을 줄여 인터넷을 통한 데이터 전송을 빠르고 적은 비용을 들일 수 있도록 만들어 준다.[1]
본 논문에서 어떻게 데이터 압축 소프트웨어와 하드웨어 중 하나를 선택해서 데이터 통신 응용에 이용하는가 하는 문제에 대한 답을 보인다. 본 논문에서는 원하는 효율과 시스템 지연의 두 문제를 가지고 결정한다.
본 논문을 통한 실험에서 주제를 명확히 하기 위해 무손실과 다중이력을 지원하는 LZS압축시스템을 이용 실시간 데이터 압축 전송 시스템을 구현하여 효율을 분석하고자 한다.

II. 압축 전송

데이터 통신에서의 아주 중요하고 기본적인 고려 사항은 압축되지 않은 데이터가 압축기와 신장기를 통하여 입출력 되는 것을 Mbps로 보이는 것이다. 이 비율은 특정 디바이스에 비례하여 얻어지는 압축비율에 상관없이 상수치가 된다.
압축엔진에 의해 얻어지는 압축비율이 완전히 데이터 그 자체에 종속되기 대문에 어느 압축된 데이터가 압축이나 신장엔진으로 들어가고 나오는가 하는 것은 예측할 수 없고 항상 일정한 값이 아니다. 이 비율은 얻어진 지속적 압축비율에 의해 나누어진 열 데이터 율의 몫으로서 결정된다.
데이터 통신 응용에서 통신채널을 통하여 전송되는 데이터는 압축될 수 있다. LZS엔진을 통해 들어오거나 나오는 압축된 데이터의 대역폭이 결정될 수 없기 때문에, 만일 데이터 율이 알려지지 않았다면 이것은 어떻게 데이터 파이프라인을 채워서 전송하는가 하는 설계 문제를 유발한다.
이것에 대한 답은 아래와 같이 하여 구한다.

1) 일반적인 데이터 통신량의 데이터 세트의 평균의 압축비율을 결정한다.
2) 샘플 데이터 세트를 통한 가장 좋은 예의 압축을 결정하여 설계한다. 만일 전체 데이터 세트의 평균 압축비율이 2:1 이면, 그리고 표본 데이터 압축비율의 가장 높은 비율이 3:1 이하이라면 설계의 가장 좋은 예로서 3:1압축을 정한다.
3) 가장 낮은 압축된 데이터 율을 구하기 위해 가장 좋은 예에 의한 압축엔진에 명시된 데이터 율을 나눈다.
4) 전형적인 데이터 통신량은 LZS데이터 압축을 이용한 WAN의 평균 전송률이 어느 곳에서는 2.5:1에서 3:1이 된다. WAN을 통한 송신되거나 수신된 데이터의 대부분은 4:1 이상으로 떨어질 것이다.
5) 상기 2)에서 이것은 일반적인 데이터 세트에 기초한 실제 압축비율을 결정하여야할 필요가 있다.

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

설계에서 명시된 율보다 큰 패킷의 압축비율로 얻어지는 결과는 압축엔진의 내부나 외부에서의 압축된 데이터의 데이터율은 데이터 통신채널보다는 작을 것이고, 순간적으로 파이프 안의 거품을 만들 것이다. 그렇지만 데이터가 아주 높은율로 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 전송양은 파이프 안 거품이 있더라도 64Kbps x 3:1 = 192Kbps가 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 3배가된다.

III. 압축의 동작

압축 프로그램은 다음과 같은 원리에서 동작한다 긴 중복되는 문장을 짧은 하나로 대체한다. 토큰으로 대체된 문자의 열로 복사한다. 예를 들어 다음 메시지는 같은 단어를 다른 세 곳에서 사용하고 있다. 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 같들은 더 많은 압축을 생성한다. 그래서 커다란 윈도우는 더욱 많은 압축비율을 가질 것이나 압축 속도는 매우 느리게 된다. 압축의 중요 요소는 인터넷의 속도를 올리는 것이다. 그리고 압축은 매우 빠르게 전달되는 컴퓨터 네트워크 안에서 이루어 져야 하므로 수행속도는 매우 중요한 문제이다.[1][2][4]

두개의 이중화된 문자의 페이지는 압축효과와 압축에 요구되는 시간을 위해 가장 잘 압축할 수 있다. 데이터의 중복되는 문장을 반복해서 찾는 것은 더욱 압축비율을 높일 수가 있으나 압축속도의 저하를 유발한다.

다음의 테이블에서 LZS 와 WINZIP으로 텍스트 데이터를 압축했을 때 압축비율과 압축 속도의 차이를 보인다.

이 테이블은 WINZIP은 파일을 더 작게 만드나 압축 속도는 5배가 걸렸다. 일반적인 목적의 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.

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

표1) LZS 와 WINZIP의 압축비율 비교

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 페이지 와 유사한 것은 전혀 압축할 수가 없다. 또한 먼저 압축된 데이터는 때로는 다시 압축할 수가 있으나 그 이상의 압축효과를 얻을 수는 없다.[4]

IV. LZS 데이터 압축

LZS 알고리즘은 입력 데이터 스트림의 장황한 데이터 스트림을 찾아서 이 스트링을 출력 데이터 스트림에 작은 길이의 기호화된 토큰으로 대체한다. LZS 알고리즘은 입력 스트림 으로부터 앞절의 데이터의 포인터로 구성된 것과 일치하는 이 스트링의 테이블을 만든다. 잡다한 스트링을 대체하기 위하여 사용되는 인코드된 토큰은 이 테이블 안의 정보로부터 만들어진다. 이런 방법으로 앞으로의 데이터는 앞절의 데이터에 기반하여 만들어진다.[4][6]

그것은 입력 스트림의 더 많은 반복되는 데이터는 더 높은 압축비율을 만들 것이고 , 대조적으로 스트림의 많은 랜덤한 데이터는 더 낮은 압축비율을 이룬다.

압축 엔진은 입력데이터의 마지막 2K byte를 가지고 있는 압축 이력의 압축동작을 가속 하기위해 다른 데이터 구조처럼 관리한다. 압축 엔진은 이 이력을 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 2K byte와 동등하여야 한다.[4]

신장 엔진은 이 이력을 압축엔진에 의해 제공되는 토큰으로부터 열 데이터를 다시 생성하기 위해 사용한다. 각 데이터 링크의 끝의 압축 과 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 스

트링을 가지고 있는 이력의 위치에 대한 옵셋을 가지고 있고 , 이 스트링이 합치하는 바이트의 개수의 길이를 가지고 있다.
압축비율은 2 개의 핵심요소로부터 이루어진다 : 압축되는 실제 데이터 와 스트링이 맞기 위해 어느 양의 검색이 이루어지는가에 따른다. 이것에서 압축은 강력한 스트링 검색과 합치로부터 구성된다. LZS는 압축비율이 실행되는 일정량의 검색을 조정함으로서 압축속도를 위해 압축비율을 조정 가능한 형태를 가지고 있다.[4]
신장은 아주 단순하며 빠른 알고리즘이다. LZS신장 알고리즘은 압축된 데이터와 각 압축된 토큰을 위한 한 번의 검색을 구현한다. 신장기는 시작을 읽고 , 위치로 이동 한 후 그 옵셋에서 출발하는 바이트의 길이를 출력한다. 이것은 압축보다 신장이 빠르고 조정 가능한 형태를 가지고 있지 않다.
각 압축동작의 마지막에서, LZS 엔진은 데이터를 출력한다. 출력은 LZS 엔진에 구성된 데이터와 마지막 표시를 출력하는 것으로 구성된다. 데이터는 긴 스트링이 합치하는 것의 중간에 위치할 것이기 때문에 LZS 엔진의 안에 위치한다. 마지막으로 이 스트링 합치 동작을 끝낸다. 마지막 표시는 하나의 토큰이며 신장기가 압축된 데이터의 마지막을 찾는데 이용된다.[4]
기능을 어떻게 구현하는가를 정하기 위해 다음 변수를 결정한다.

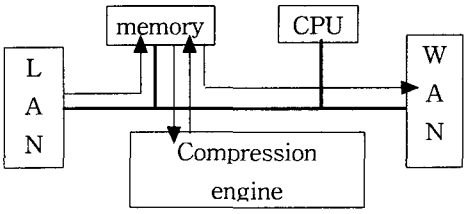
1. 효율과 시스템에 요구되는 압축 신장율(일반적으로 전이중 WAN 속도).
2. 압축비율 분배와 평균/지정
3. CPU의 성능과 가용한 버스 대역폭

이 데이터로 라우팅을 위한 데이터 압축엔진을 가장 최적으로 구현 할 수가 있다. 이 모든 라우팅 과 포워딩 기능을 모토롤러 MPC860-40MHz CPU 시스템에서 구현하였다. 그리고 실험을 한 시스템은 10Mbps 이더넷 포트와 단일 128K WAN포트를 가지고 있다. WAN 연결이 설정된 상태에서 CPU는 LAN 데이터 패킷을 10Mbps로 송수신하며, WAN 데이터를 128Kbps로 송수신 한다.
본 실험에서 데이터 압축 은 B채널 둘 다 지원할 것이고 2:1의 평균 데이터 압축비율을 가진다. 압축비율은 3:1로 하였다. 이 원칙은 파이프 안의 거품이 없이 주어진 효율 레벨에서 압축과 신장을 할 수 있는 것이다. 이것은 3:1 이나 4:1 의 아래와 같은 동작을 할 수 있다.
라우터는 다음과 같은 특성을 가진다.

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

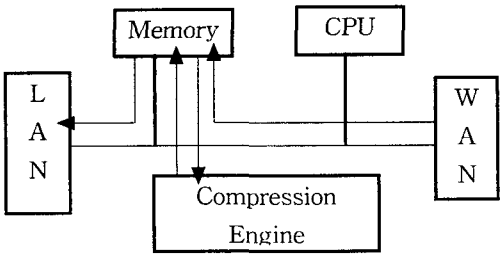
라우터를 통한 데이터의 흐름을 다음에 보인다.

[그림1] 전송된 데이터 (반이중 모드)



전송된 데이터는 위와 같은 경로를 따른다.
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 그 때 WAN 포트로 전달된다. 압축 엔진의 목적은 WAN 인터페이스에 WAN 파이프가 차도록 충분한 압축된 데이터를 제공하는데 있다.

[그림2] 수신된 데이터(반이중 방식)



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

V.성능요구 분석

설계 원칙을 보면 엔진은 최소한 2 x 64Kbps = 128Kbps정도의 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 입력하여 128Kbits의 압축된 데이터를 출력하여야 한다.
MPC860의 시스템에서 압축엔진은 데이터를 1.248Mbps로 압축하고, 2.352Mbps로 데이터를 신장한다. 공식 2로부터 이것은 CPU 리소스를 100% 사용하

여 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.

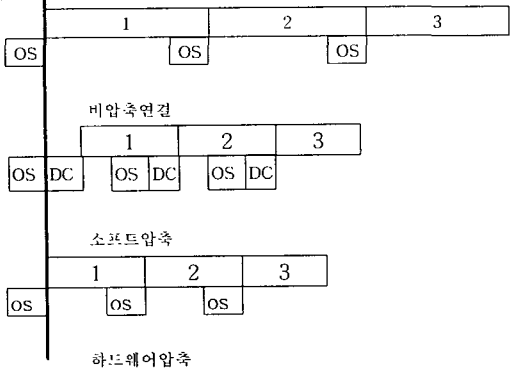
WAN 링크에 압축을 추가함은 파이프의 대역폭을 증가시키고, 따라서 패킷을 처리하거나 라우팅 하기 위해 CPU의 능력증대가 요구된다. 예를 들어 , 만일 압축비율이 2:1 이고 128Kbps 데이터 경로이면 이제 효과적으로 256Kbps경로가 된다. 이것은 압축엔진 자체를 위해 디바이스가 사용되지 않을 지라도 라우팅 CPU의 요구를 증대시킨다.

압축을 적용한 라우팅 응용에서 압축은 CPU MIPS의 60%을 점유했고 라우팅은 나머지 40%를 점유한다. 이 비율로부터 만일 라우터가 MPC860의 전체 사용할 수 있는 MIPS의 31.4%를 사용하면, 그리고 이것에 전체 리소스의 60%가 요구된다면 라우팅은 (40/60)x31.4%즉 20.6%의 CPU MIPS를 쓰게된다.

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

다음의 그림은 세 개의 일반 패킷을 WAN상에서 전송하는데 걸리는 시간을 보인다. 그래프는 압축 없이 전송하는데 걸리는 시간을 소프트웨어 압축과 하드웨어 압축을 함께 보인다. 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.[7]

MPC860-40MHz CPU
Line speed = 128Kbps ; 압축비율 = 2:1 ; 데이터율 = 256Kbps
운영체계 점유율 = 전체 CPU 시간의 20.6%.
데이터 압축 점유율 : 전체 CPU 시간의 31.4%.



[그림 3] 압축과 비압축의 데이터 전송 비교

시간 평균의 공식은 1Mbps, 1초, n바이트의 전이중 데이터 압축 율은 압축엔진에 의해 압축될 것이고, b바이트가 신장될 것이다. 만일 반이중 신장율이 D-Mbps이고 반이중 압축율이 C-Mbps이면, n바이트를 압축하고 n-바이트를 신장하는 시간은 다음과 같다.[4]

[공식1] 주기를 통한 전이중 압축과 신장 방정식
1Sec= n-Mbytes/C-Mbps + n-Mbytes/D-Mbps.
따라서 전이중, 시간 평균된 압축엔진의 밴드폭
[공식2] 전이중 압축 신장률.
n-Mbps(FDX) = 1/(1/C-Mbps + 1/D-Mbps).

VI. 결론 및 추후 연구

본 논문에서 보듯이 데이터 통신상의 데이터 압축을 완벽하게 구현하였을 때에는 대역폭을 충분히 개선시킬 수 있었으며 부 적절히 구현되었을 때에는 실제 대역폭을 감소시킨다. 이것은 다른 응용을 추론할 수가 있다. 단순히 데이터 채널이 공급할 수 있는 최대 통신 대역폭을 결정함으로서, 일반적인 데이터 흐름을 위한 적절한 압축비율을 결정 할 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. 압축 이력을 관리하는 것과 가상 연결을 압축 이력과 관련하여 비 확장을 관리하는것은 라우터에 최적의 압축을 성공적으로 구현하는 가장 중요한 요소가 됨을 알 수 있었다. 그리고 본 논문에서는 단일 채널을 이용한 압축전송에 대한 실험만을 하였으나, 추후에 2채널 및 4채널에 대한 실험을 하여 단일 시스템에서 가장 높은 효율을 내는 것을 연구할 계획이다.

참고문헌

[1] American National Standards Institute, Inc., "Data Compression Method for Information Systems," ANSI X3.241- 1994, August 1994.

[2] Shacham, A., "IP Payload Compression Protocol (IPComp)", RFC 2393, December 1998.

[3] Rand, D., "The PPP Compression Control Protocol (CCP)", RFC 1962, June 1996.

[4] Schneider, K., and R. Friend, "PPP LZS-DCP Compression Protocol (LZS-DCP)", RFC 1967, August 1996.

[5] Applied Cryptography, Second Edition, John Wiley & Sons,1996.

[6] MPC860 User's Manual, Motorola Press, 1999

[7] Precise/MQX Realtime-Executive, Precise, 1999