인터넷 시각동기 클라이언트를 통한 안정적 시각오류 보정 방법

강봉호*, 민충식*, 이영종**, 김영호*
*부산대학교 전자계산학과, **네비콤(주)
Email : bhkang@juno.cs.pusan.ac.kr

A Stable Time Error Correction Using Time Synchronization Client over Internet

Bongho Kang*, Choongsik Min*, Youngjong Yi**, Youngho Kim*
*Department of Computer Science, Pusan National University, **NAVICOM Co.,Ltd.

요 약

컴퓨터 시스템은 기본 주파수를 제공하는 수정발진자의 환경적 요인에 의한 오차 등으로 인해 정확한 시각을 유지하지 못한다. 정확한 시각유지를 위한 여러 방법 중 본 논문에서는 인터넷을 통해 외부 시각원으로부터 시각신호를 받아 로컬 시스템의 시각오류를 보정하는 시각동기 클라이언트를 이용하는 방법을 연구하였다. 정밀도 향상을 위해 다중서버 선택기능을 보강하였고, 지연시간에 따른 시각원의 가중치를 적용하였다. 특히 시각 변화에 민감한 프로세스가 많은 시스템에서 시각 오류 보정시 시각의 급변은 시스템에 무리를 일으키므로 점근적 보정을 통한 안정적 방법이 필요하다. 이를 위해 축적결과기반 시각오류 보정방법을 적용하여 네트워크 지연의 급작스런 변화에도 분산을 통한 충격을 완화하여 안정적인 시각보정을 할 수 있도록 하였다. 시각오류 요인과 시각동기방법, 클라이언트 구현과 실험 결과를 제시한다.

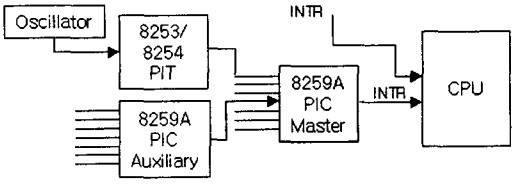
1. 서론

모든 컴퓨터 시스템은 현재시각 유지목적 이외에도 중앙처리장치에서의 프로세스 점유시간, 디스크 입출력 등을 위해 시각장치를 가지게 된다. 보통의 시스템은 시각장치의 구성으로 근원적인 주파수 발생 장치인 수정발진자(Crystal Oscillator)와 주파수 관련 정보를 저장하는 레지스터들로 구성이 된다. 가장 중요한 수정 발진자는 온도, 습도, 기압, 진동과 같은 외부의 환경적 요인으로 인해 오차를 가지기 때문에 시스템은 항상 부정확한 시각을 가지고 있다. 네트워크로 연결되어 있는 분산 시스템의 경우 어플리케이션의 작업이 각 노드에 나누어져 있어 정확한 실행결과를 보장하기 위해서 각 노드간의 시각동기가 필수적인 요소이다.[PRA96] 또한 인터넷을 통한 전자상거래의 발전으로 인해 수많은 트랜젝션의 정확한 시각 소인이 필요하기 때문에 시각소인서버의 정확한 시각유지가 필요하다. 시스템 시각의 부정확함으로 인해 시각을 보정하는 다양한 방법이 소개되었다. 본 논문에서는 인터넷을 통해 외부 시각원으로부터 시각신호를 받아 로컬 시스템의 시각오류를 보정하는 시각동기 클라이언트를 이용하는 방법을 제시한다. 정밀도 향상을 위해 다중서버 선택기능을 보강하였고, 지연시간에 따른 시각원의 가중치를 적용하였다. 시각 변화에 민감한 프로세스가 많은 시스템에서는 시각 오류 보정시 시각의 급작스런 변화는 시스템에 무리를 일으키므로 점근적 방법을 통한 안정적인 시각 보정방법이 필요하다. 이러한 문제 해결을 위해 축적결과기반 시각오류 보정방법을 적용하여 네트워크 지연의 급작스런 변화 등에도 분산을 통한 충격을 완화하여 안정적인 시각보정을 할 수 있도록 하였다.
논문의 전반부에서는 시각오류요인과 시각동기방법을 설명하고, 후반부에는 시각동기 클라이언트 구현과 실험결과를 제시한다.

2. 컴퓨터 시스템 내에서의 시각오류와 보정방법

2.1 시각 오류 원인

컴퓨터의 시각장치는 크게 소프트웨어 클럭과 하드웨어 클럭으로 나눌 수 있다. 소프트웨어 클럭은 부팅하면서 동작을 시작하고, 표준 PC의 경우 [그림 1]과 같이 수정 발진자로부터 일정한 클럭 신호(14.318Mhz)를 받아 8254 (PIT, Programmable Interval Timer)칩이 일정 펄스에 한번의 클럭(100hz)을 발생시키고, 8259A (PIC; Programmable Interrupt Controller)칩이 신호를 받아 CPU에 시각 인터럽트를 전달한다. 하드웨어 클럭은 컴퓨터의 전원이 꺼져있는 동안 사용되며, 배터리 전원을 이용 CMOS에서 시각을 유지하게 된다. 주로 모토로라의 MC146818 real time clock이 사용된다. [PAT96] 소프트웨어 클럭은 수정발진자의 문제, 다른 프로그램에 의한 인터럽트 변화와 클럭의 해상도에 따라 컴퓨터의 시각오차를 발생시키게 된다. 하드웨어 클럭의 경우 수정발진자의 정확도에 의해 결정되므로 환경적 요인에 의한 수정발진자의 상태에 따라 오차를 발생시키며, 통상 1초 단위의 변경으로 인해 초 이하의 값을 표현할 수 없는 문제가 있다.[MIC98]

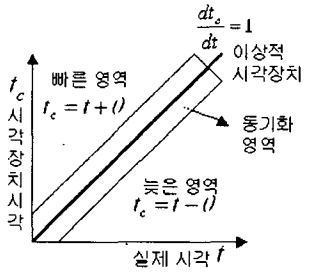


[그림 1] 표준 PC 시각장치의 구조

2.2 시각 보정 방법

시각보정 방법은 자체적인 방법과 외부의 도움에 의한 방법으로 나눌 수 있다. 자체적인 방법은 보통 $1미만의 가격을 가지는 낮은 품질의 수정발진자를 좋은 특성을 가진 것으로 교체하는 방법이다. 이 방법을 통해 외부 환경적인 요인으로 인한 오차와 1초 이하의 해상도를 가지지 못하는 문제를 해결할 수 있다.[MIC98] 외부 도움에 의한 방법은 인터넷을 통한 방법과 다른 망을 통한 방법으로 구분할 수 있다. 컴퓨터가 인터넷과 연결이 되어 있는 경우 인터넷을 이용하는 방법이 사용가능하며, 인터넷상의 시각 서버들에게서 시각정보를 받아 동기를 시키는 방법이다. 여기에는 크게 4가지 프로토콜이 사용되며 Time Protocol(RFC-868), Daytime Protocol (RFC-867), Network Time Protocol(NTP, RFC-1305), Simple Network Time Protocol(SNTP, RFC-2030)이 있다. [POS83a][POS83b] [MIL92] [MIL96] 컴퓨터 네트워크 망에 직접 연결이 되어 있지 않은 경우 다른 망을 통해 서비스를 받을 수 있으며, GPS 또는 GLONASS 같은 Navigation 시스템을 이용하는 방법이 있고, 전화망을 이용하는 방법, 방송망을 이용하는 방법등이 있다. 각 방법들의 차이에 의해 1마이크로초에서 수십밀리초까지의 시각동기가 가능하다. 본 논문에서는 인터넷을 이용하는 방법중 SNTP를 이용하는 시각보정방법을 사용한다. 실제시각과 시각장치시각은 이상적인 경우 [그림 2]와 같이 기울기 1의 직선을 가진다. 그러나 수정발진자의 오차로 인해 시각은 표류(Drift)하게 되며 최대 표류율이 ρ,

시각옵셋이 θ 인 경우 표류율 ρ에 따라 $\pm\theta$만큼의 오차를 가지게 된다. 옵셋이 허용오차(β)내에 들어가는 경우 동기화 되었다고 한다. 관계식은 <식 1> 과 같다. [PRA96]

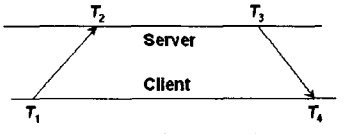


[그림 2] 시각장치의 정확도

$$1-\rho \le \frac{dt_c}{dt} \le 1+\rho \text{ (시각표류)}$$
$$|t-t_c| \le \beta \text{ (시각동기)} \qquad \text{<식 1>}$$

3. 시각동기 클라이언트를 이용한 시각보정

네트워크로 연결된 시스템의 경우, 전세계에 1700개 이상의 시각서버가 존재하고, 공용으로 사용할 수 있는 Stratum 1 서버도 100개 이상이다. 시각서버와 연결을 통해 정확한 시각을 유지할 수 있으며, 이런 목적으로 널리 사용되는 프로토콜에는 NTP와 SNTP가 있다. 네트워크를 통한 시각보정을 하는 경우 시각서버와 클라이언트간의 전송지연을 고려해야 한다. 서버와 클라이언트간 시각차를 offset(θ)이라고 하고, 네트워크를 통한 전송지연을 delay(δ)라고 하는 경우 시각차이와 전송지연의 관계는 [그림 3], <식 2>와 같다. [MIL92]

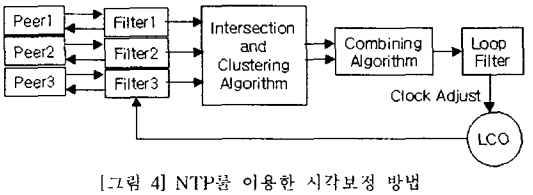


[그림 3] 서버와 클라이언트간의 시각소인

$$Offset\ \theta = \frac{1}{2}[(T_2-T_1)+(T_3-T_4)]$$
$$Delay\ \delta = (T_4-T_1)-(T_3-T_2) \qquad \text{<식 2>}$$

3.1 NTP를 이용한 시각동기

미국 델라웨어(Delaware)대학에서 주로 개발된 인터넷을 통한 컴퓨터 클럭보정을 위한 프로토콜이며, 분산된 시각서버와 클라이언트간의 시각을 동기화 하는데 사용된다. PLL(Phase Lock Loop) 방법을 사용하여 서버들간 또는 서버넷 동료(Peer)들 간에 시각 소인을 교환하여 신뢰성있는 범위내로 클럭을 동기화한다. NTP 버전 1은 RFC-958[MIL85]에서 처음 기술되었고, 빠르게 발전하여 RFC-1119에서 NTP버전 2의 기술이 이루어졌다. 이것은 비연결형 수송메카니즘을 제공하는 인터넷 규약(IP)과 사용자 프로그램 규약(UDP)의 한 부분으로 만들어졌다. 현재 RFC-1305에서 NTP 버전3 [MIL92]에 대한 정의가 이루어 졌으며, NTP버전 4를 위한 다양한 논의가 진행되고 있고 많은 기술백서들이 존재한다. NTP를 이용한 시각 보정 방법은 [그림 4]와 같다.



[그림 4] NTP를 이용한 시각보정 방법

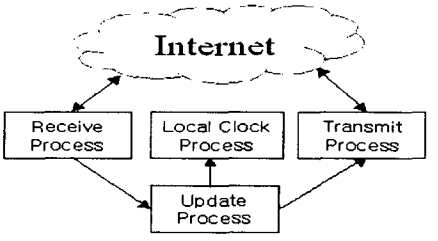
3.1 SNTP를 이용한 시각동기

SNTP는 NTP와 같이 64비트의 시각소인(Time Stamp)를 사용하는 특성은 같지만, 시각동기를 하는 방법을 단순화시켜 WAN에서 수십 밀리초 정도의 정도를 요구하는 곳에 사용하는 프로토콜이다. UDP/TIME 프로토콜에서 얻어지는 정확성과 신뢰성 정도로 동작을 한다.[POS83a] RFC-1769에서 SNTP 버전3에 대한 정의가 이루어 졌으며 현재 IPv6를 지원할 수 있는 SNTP 버전4 (RFC-2030)[MIL96]까지 정의되어 있다. SNTP의 경우 NTP의 경우처럼 클라이언트의 상태 정보를 유지하지 않기 때문에 더 많은 수의 클라이언트에게 서비스 해 줄 수 있고, 간단한 동작을 정의하므로 구현하기 쉬운 장점을 지니고 있다. 정확성과 신뢰성에서는 NTP에 비해 떨어지고 치명적으로 한개 서버의 시각 신호만 받는 단점이 있다.

3.3 시각동기 클라이언트 구성과 알고리즘

3.3.1 전체 구성

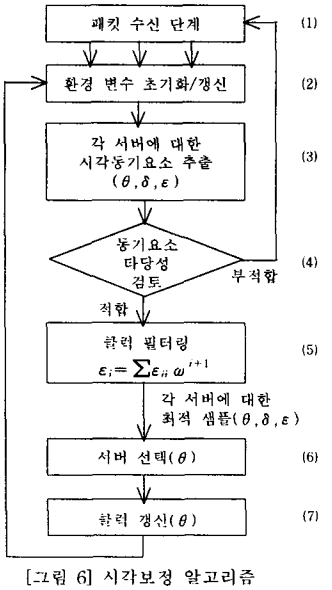
인터넷을 통한 시각동기를 하는 클라이언트는 [그림 5]에서와 같이 4개의 프로세스로 나눌 수 있다. 각각은 인터넷을 통해 시각정보를 읽어오는 과정, 현재 가지고 있는 정보를 처리하는 과정, 로컬 클럭을 보정하는 과정과 서버에 현재의 시각소인을 보내는 과정이다.



[그림 5] 시각보정 클라이언트 구성

3.3.2 보정 알고리즘

시각보정을 하는 알고리즘은 [그림 6]과 같다.



[그림 6] 시각보정 알고리즘

(1) 패킷수신단계
- 인터넷을 통한 시각패킷을 수신하는 단계.
- 소켓을 사용하여 구현.

(2) 환경변수 초기화 및 갱신
- 내부저장변수와 시각관련 변수를 초기화.
- 기존 데이타를 읽어 시각보정을 위한 기본정보생성.

(3) 각 서버에 대한 시각 동기 요소 추출
- 패킷이 유효한지 검사.
- 옵셋(θ)과 지연시간(δ), 분산(ε)등 시각동기 요소 추출.

(4) 시각 동기 요소 타당성 검토
- 시각동기요소가 시각보정에 도움이 되는지 검토.

(5) 클럭 필터링
- 시각정보에 가중치를 곱하여 클럭에 대한 값을 필터링.
- $\varepsilon_i = \sum \varepsilon_{ij} \omega^{i+1}$ (ε:분산, ω:가중치)

(6) 서버 선택
- 각 서버에 대한 최적의 샘플을 선택.

(7) 클럭 갱신
- 로컬 시각 보정.
- 현재 시각과 보정시각을 파라미터로 하여 점근적인 보정을 할 수 있도록 한다.

3.3.3 시각보정방법

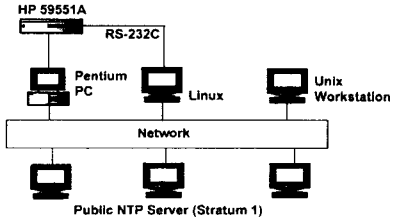
패킷수신단계에서 SNTP에 없는 다중 서버의 패킷을 받아 처리할 수 있도록 하여 여러 개의 서버에서 시각정보를 받아 계산함으로써 시각보정의 정확도를 높였다. 클럭 필터링과 서버선택의 단계에서 정밀한 시각보정을 위해 거리(λ), 분산(ε), 지연시각(δ)따른 가중치(ω)를 계산하여 옵셋(θ)을 처리함으로써 보다 정밀한 시각 유지를 할 수 있도록 하였다. 클럭갱신을 하는 경우 안정적인 시각 보정을 위해 *settimeofday()*와 같은 time count에 시각의 값을 직접 변경하는 시스템 함수를 지양하고, *adjtime()*을 사용하여 시스템이 PIT의 입력값을 조정하여 점근적인 보정을 할 수 있도록 보장하였다. 최초 구동시 또는 네트워크 전송지연의 급변에 의해 계산된 옵셋이 상당한 오차가 나는 경우가 있으며, 이 경우 급작스런 시각의 변화 또는 데몬 프로그램의 종료를 초래할 가능성이 있다. 이런 문제를 해결하기 위해 서버에 대한 로깅을 하여 이전의 옵셋을 참고하여 보정하도록 하였다. <식 3>에서 보이듯이 서버 j에 대한 옵셋 θ_j는 전체 로깅되어 있는 옵셋의 합과 현재의 옵셋을 더한 다음 전체 자료의 수로 나누어 처리한다.

$$\theta_j = \frac{\sum_{i=0}^{n} \theta_{ji}}{n} \qquad \text{<식 3>}$$

4. 실험 및 결과분석

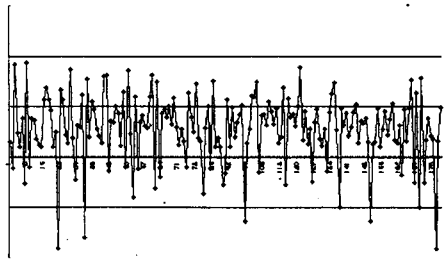
4.1 실험환경

실험환경의 네트워크 대역폭은 10Mbps이며, 공용 NTP서버를 이용하였다. 유닉스 워크스테이션과 리눅스 서버에서 동기화 응용 프로그램을 수행하였고, 시각 보정 결과를 모니터링하기 위하여 고정밀도 시각 소인용 장치인 HP 59551A와 리눅스 서버를 RS232C로 연결하였고, HP59551A의 데이터를 추출하는 Windows용 어플리케이션을 실행하기 위하여 Pentium PC와 RS-232C로 연결을 하였다. 연결도는 [그림 7]과 같다.



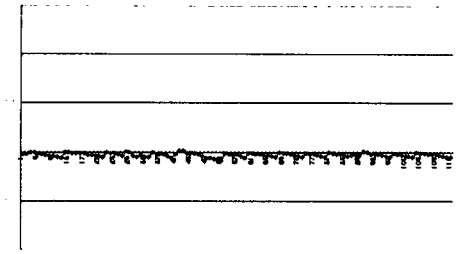
[그림 7] 실험 시스템 구성

4.2 실험결과

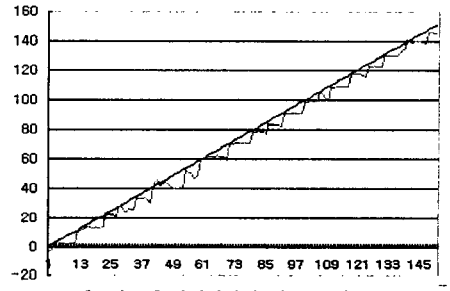


[그림 8] 일반 방법을 이용한 결과

[그림 8]은 일반적인 시각보정 방법을 사용하였을 경우의 결과이다. 옵셋의 변위가 ±10밀리초에서 많은 진동을 하고 있음을 보이고 있다. 반면 [그림 9]는 제안한 보정방법을 사용한 경우의 결과이며, 옵셋 0 부근에서 안정적인 시각보정을 이루고 있는 모습을 확인할 수 있다. [그림 10]은 옵셋의 변위을 150배 확대하여 그린 그림으로 x축이 실제시각, y축이 시스템 시각인 경우를 보이고 있다. 기울기 1의 직선을 따라가면서 시스템의 시각이 보정되고 있음을 보이고 있다.



[그림 9] 제안한 보정방법을 이용한 결과



[그림 10] 실제시각과 시스템 시각

5. 결론 및 향후과제

본 논문은 인터넷을 통해 외부 시각원으로부터 시각신호를 받아 로컬 시스템의 시각오류를 보정하는 시각동기 클라이언트를 이용하는 방법을 제시하였다. 정밀도 향상을 위해 다중서버 선택기능을 보강하였고, 지연시간에 따른 시각원의 가중치를 적용하였다. 컴퓨터 시스템의 안정적인 시각보정을 위해 축적결과기반 시각오류보정방법을 적용하였다. 유닉스와 리눅스상에서 실험을 하였고, 실험결과를 제시한다. 향후 정확한 시각보정을 위한 서버 선택 알고리즘과 클럭보정을 위한 알고리즘을 개선할 수 있다. 유닉스 기반의 시스템 뿐 아니라 윈도우즈 운영체제에서 안정적인 시각보정 방법을 연구할 수 있을 것이다.

[참고문헌]

[HP96] HP, "HP59551A and HP58503A GPS Receivers Operating and Programming Guide", HP, May, 1996.

[KYI98] 김영호, "Timing Distribution with Network:NTP", 98GPS-WS Proc. pp397-443, November 1998.

[LSJ99] 이성진, GPS시각원과 FLL을 사용한 시각정밀도 향상기법, 이학석사학위논문, February 1999.

[LSY98] 이상엽, 시각동기를 위한 컴퓨터 통신망상의 동적시각 오류요소 분석, 이학석사학위논문, Feburary 1998.

[MIC98] Micael Lombadari, "Computer Time Synchronization", Time and Frequency Division NIST, 1998.

[MIL90] Mills, "Measured performance of the Network Time Protocol in the Internet System", ACM Computer Communication Review 20, 1, January 1990.

[MIL85] D. Mills, "Experiments in network clock synchronization", DARPA Netwok Working Group Report RFC-958, September 1985.

[MIL92] D. Mills, "Network Time Protocol (v3)", DARPA Netwok Working Group Report RFC-1305, April 1992.

[MIL94] Mills, "Improved Algorithm for Synchronizing Computer Network Clocks", Proc. ACM SIGCOMM 94 Symposium, September 1994.

[MIL96] D. Mills, "Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI", DARPA Netwok Working Group Report RFC-2030, October 1996.

[OJS99] 오지석, NTP 시각 동기 특성 분석, 이학석사학위논문, February 1999.

[PAT96] Steve D Pate, "Unix Internals A Practical Approach", Addison-Wesley, 1996.

[POS83a] Postel, J, "Time Protocol", DARPA Network Working Group Report RFC-868, USC Information Sciences Institute, May 1983

[POS83b] Postel, J, "Day Time Protocol", DARPA Network Working Group Report RFC-867, USC Information Sciences Institute, May 1983

[PRA96] Pradepp K. Sinha, "Distributed Operating Systems Concepts and Design", IEEE Computer Society Press, 1996.

[SEV98] W. Richard Stevens, "Unix Network Programming Vol1, 2", Prentice Hall, 1998