

무선 임베디드 환경에서의 시간 동기화

(Clock Synchronization in Wireless Embedded Applications)

노진홍[†] 홍영식^{**}

(Jin Hong No) (Young Sik Hong)

요약 최근 무선 통신의 발달과 함께 임베디드 시스템의 성능 향상 및 보급률 증가로 기존의 분산 시스템 환경에 무선 임베디드 시스템들이 포함되기 시작하였다. 분산 시스템을 구성하는 요소들 간의 동기화, 순서화, 그리고 일관성 유지를 위하여 시간 동기화는 반드시 필요하고, 지난 20여 년간 분산 시스템에서의 시간 동기화에 관한 많은 연구가 이루어져 왔다. 하지만 무선 임베디드 시스템에서의 시간 동기화는 메시지 지연과 손실이 많다는 점과 풍부하지 않은 시스템 자원을 고려해야 하므로, 기존 유선 환경에서 사용되었던 시간 동기화 알고리즘을 그대로 적용하기에는 어려운 점이 많다. 이에 본 논문에서는 IEEE 802.11 표준을 확장하여 무선 임베디드 환경에 적합한 시간 동기화 방법을 제안한다. 제안된 방법은 브로드캐스트 통신의 특성을 활용하여 무선 임베디드 환경에서의 제약 조건을 완화함으로써 높은 정확성을 제공하면서 메시지 손실을 감내하여 연속적인 시간 동기화를 제공할 수 있다. 이를 위해 마스터/슬레이브 방식의 구조에서 마스터는 시간 동기화를 위한 시간 정보를 브로드캐스트하고, 슬레이브는 편차와 편차율을 계산하여 마스터의 시간을 추정하고 동기화된 시간인 가상 시간을 계산하였다. 실험을 통해 제안된 시간 동기화 알고리즘을 사용하는 경우 200 μ s 정도의 표준 편차 범위로 동기화할 수 있음을 보였다.

키워드 : 무선 임베디드 시스템, 시간 동기화, 무선 통신, IEEE 802.11 표준

Abstract With the proliferation of wireless network and the advances of the embedded systems, the traditional distributed systems begin to include the wireless embedded systems. Clock synchronization in the distributed systems is one of the major issues that should be considered for diverse purposes including synchronization, ordering, and consistency. Many clock synchronization algorithms have been proposed over the years. Since clock synchronization in wireless embedded systems should consider the low bandwidth of a network and the poor resources of a system, most traditional algorithms cannot be applied directly. We propose a clock synchronization algorithm in wireless embedded systems, extending IEEE 802.11 standard. The proposed algorithm can not only achieve high precision by loosening constraints and utilizing the characteristics of wireless broadcast but also provide continuous time synchronization by tolerating the message losses. In master/slave structure the master broadcasts the time information and the slave computes the clock skew and the drift to estimate the synchronized time of the master. The experiment results show that the achieved standard deviation by the proposed scheme is within the bound of about 200 microseconds.

Key words : Wireless Embedded Systems, Clock Synchronization, Wireless Communication, IEEE 802.11 standard

1. 서론

분산 시스템은 네트워크로 연결된 여러 컴퓨터들이 메시지 교환을 통해 상호 통신하고 협력하는 집합체를 말한다. 분산 시스템의 구성 요소 간에 동기화, 순서화,

그리고 일관성 유지를 위해서 시간 동기화는 반드시 필요하므로 지난 20여 년간 시간 동기화에 관한 많은 연구가 이루어져왔고 다양한 형태의 시간 동기화 기법이 소개되어 왔다[1-5].

최근 무선 통신의 발달과 함께 임베디드 시스템의 확산으로 기존의 분산 시스템 환경에 무선 임베디드 시스템들이 포함되기 시작하였다. 무선 임베디드 시스템에서의 시간 동기화는 여러 가지 제약 조건을 가지고 있다. 첫 번째로 무선 네트워크의 특성상 메시지의 손실과 충돌이 많아 메시지 전송 시간의 변화가 심하므로 메시지

· 본 연구는 대학 IT연구센터 육성·지원사업의 연구결과로 수행되었음

[†] 학생회원 : 동국대학교 컴퓨터공학과
jhno@dongguk.edu

^{**} 종신회원 : 동국대학교 컴퓨터, 멀티미디어공학과 교수
hongys@dongguk.edu

논문접수 : 2005년 6월 24일

심사완료 : 2005년 8월 10일

지연 시간과 손실을 감내할 수 있어야 한다. 두 번째로 무선 네트워크는 제한된 대역폭과 에너지를 가지고 있기 때문에 경량화(light-weight)된 방법을 사용하여 과도한 메시지 전송에 따른 부하와 대역폭 사용을 감소시켜야 한다. 세 번째로 임베디드 시스템은 기존의 자원이 풍부한 데스크톱과 서버에 비해 시간을 측정하는 작업이 부정확할 수 있으므로 적절한 에러 범위를 가질 수 있음을 고려해야 한다. 이와 같은 제한 사항을 만족하기 위해서 기존 유선 환경에서 사용되었던 시간 동기화 알고리즘을 그대로 무선 임베디드 환경에 사용하기에는 부적합하다[6,7].

무선 임베디드 환경에서 이와 같은 문제점을 해결하기 위한 한 가지 방법으로 GPS(Global Positioning System)를 사용하여 외부 표준 시간과 동기화를 수행할 수 있는 방법이 있지만 임베디드 시스템의 확장성과 가격상의 문제로 인해 모든 임베디드 시스템에 대한 해결책이 될 수 없다. 또한 GPS와 같은 시간 동기화를 수행하는 하드웨어를 장착하지 않고 유선 환경에서의 시간 동기화 방법을 간소화하여 소프트웨어적인 시간 동기화를 하는 Sntp[8]과 같은 방법은 현재 시간과 동기화할 시간의 차가 너무 크게 되는 경우에 시간 동기화를 거절하기 때문에 무선 네트워크 환경을 사용하는 시스템에서는 시간 정확성이 떨어질 수 있다는 단점을 가지고 있다.

이에 본 논문에서는 임베디드 시스템을 대상으로 무선 랜 환경에서 사용가능한 소프트웨어적인 시간 동기화 기법을 제안한다. 제안된 기법은 IEEE 802.11 표준[9]에서 권고하는 마스터/슬레이브 형태를 기반으로 브로드캐스트 통신의 특성을 활용하여 메시지 지연 시간에 관계없이 상대방의 시간을 측정하여 높은 정확성을 제공할 수 있다. 마스터로부터 매 주기마다 브로드캐스트되는 메시지 하나만으로 슬레이브는 편차값과 편차율을 계산하여 마스터의 시간을 추정하고 가상 시간을 계산하므로 시간 동기화에 따른 작업 부하를 감소하였다. 또한 계산된 편차와 편차율을 통하여 가상 시간을 계산하여 연속적인 시간 동기화를 제공할 수 있으므로 메시지 손실을 감내할 수 있다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 본 논문과 관련된 기존 연구를 기술한다. 3장에서는 본 논문에서 제안하는 시간 동기화 방법을 설명하고 시간 동기화 오차를 수식을 통하여 증명한다. 4장에서는 제안된 방법을 구현하여 실험한 결과를 통하여 성능을 평가하고 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구

2.1 동기화 에러를 발생시키는 요인들

모든 시간 동기화 방법들은 노드들 간에 시간 정보를

교환하는 방법에 의존한다. 하지만 네트워크를 통한 메시지 송수신시 발생하는 여러 가지 비결정적(non-deterministic) 요소들은 시간 정보를 부정확하게 하므로 시간 동기화의 정확성을 떨어뜨린다. 즉, 어떤 노드가 동기화를 위하여 시간을 측정하고 다른 노드에게 전송하거나 전송된 시간을 수신자의 수신 시간과 비교할 때 사용되는 시간 정보들은 네트워크와 운영체제로 인한 여러 종류의 지연 시간을 포함하게 된다. 이 지연 시간으로 인해 수신자는 정확한 자기의 시간을 비교하여 송신자와의 시간을 동기화하는데 문제를 야기한다. [3]에서는 시간 동기화 방법에 에러를 발생 시키는 네 가지 기본 구성요소를 다음과 같이 정리하였다.

- 송신 시간(send time) : 송신 시간은 송신자에서 메시지를 생성하는데 소요되는 시간이다. 콘텍스트 스위치(context switch)나 시스템 콜(system call)과 같은 운영체제에서 발생하는 지연과 호스트에서 네트워크 인터페이스(network interface)에 메시지를 전송하는 시간이 포함된다.
- 접근 시간(access time) : 접근 시간은 생성된 메시지가 전송을 위해 전송 채널로 접근하기 위해 기다리는 시간이다. 각 패킷은 실제 전송이 발생하기 전에 MAC(Medium Access Control) 계층에서 일정 지연 시간을 가지게 되는데 이더넷과 같은 회선 쟁탈 기반 MAC(contention-based MAC)은 채널이 유희상태가 될 때까지 기다려야하고, TDMA(Time-Division Multiple Access)는 해당 타임 슬롯을 사용할 때까지 기다려야 한다. 802.11 네트워크와 같은 RTS/CTS 기법에서는 데이터 전송 전에 제어 패킷을 교환하는데 소요되는 시간이다. 그러므로 접근 시간은 어떤 MAC 기법을 사용하느냐에 따라 다르다.
- 전파 시간(propagation time) : 전파 시간은 패킷이 송신자를 떠나 수신자의 네트워크 인터페이스까지 전송되는데 필요한 시간이다. 랜이나 무선 랜과 같이 송신자와 수신자가 동일한 물리적 매체로의 접근을 공유할 때는 단순히 매체를 통한 메시지의 물리적 전파 시간을 의미하기 때문에 아주 적은 시간이다. 반대로 WAN(Wired Area Network)과 같이 메시지가 네트워크를 통과할 때마다 각 라우터에서 큐잉과 스위칭 지연이 발생할 때는 전파 시간이 대부분의 지연 시간을 소요한다.
- 수신 시간(receive time) : 수신 시간은 수신자의 네트워크 인터페이스가 메시지를 수신한 후 호스트에게 메시지 도착을 통지하는데 소요되는 시간이다. 이는 일반적으로 네트워크 인터페이스가 메시지 수신 신호를 생성하는데 필요한 시간이다. 만약 도착 시간이 네트워크 드라이버의 인터럽트 처리기와 같은 운영체제

커널의 저수준에서 타임스탬프 되었다면 수신 시간은 시스템 콜과 콘텍스트 스위치, 심지어 네트워크 인터페이스에서 호스트까지의 전송으로 인한 오버헤드를 포함하지 않을 수 있다.

기존의 시간 동기화 알고리즘들은 이러한 네 가지 종류의 동기화 에러 원인을 추정하고 보정하여 시간 정보를 계산하는 방법에서 차이가 있다고 볼 수 있다.

2.2 편차를 기반 알고리즘

시스템의 물리적 시간은 정기적으로 이벤트를 발생시키는 진동기(oscillator)의 진동수(frequency)로 측정하며, 진동기는 주변 온도와 진동기의 노후 정도와 같은 환경적인 영향에 따라 변화가 발생할 수 있다[10]. 하지만 이러한 변화는 무시할 정도로 미약하고, 진동기가 시간에 따라 일정한 진동수를 가진다는 가정 하에 시스템의 물리적 시간은 일정한 비율로 실제 시간에 대한 편차(skew)를 보이며 이를 편차율(drift)이라고 한다. 실제 시간이 t일 때 시스템 i의 시간을 T_i 라고 한다면 다음과 같이 실제 시간과 시스템 시간과의 관계를 구할 수 있다.

$$T_i = a_i t + b_i \tag{1}$$

식 (1)에서 a_i 는 t의 변화량에 대한 T_i 의 변화량의 비율로서 편차율을 나타내고, b_i 는 실제 시간 t에서 두 시간의 차를 나타내는 오프셋(offset)이다.

식 (1)을 사용하여 마스터 m과 슬레이브 s의 시간을 비교할 수 있다.

$$T_m = a_{ms} \cdot T_s + b_{ms} \tag{2}$$

식 (2)에서 a_{ms} 는 마스터와 슬레이브 사이의 상대적인 편차율이고, b_{ms} 는 마스터와 슬레이브 사이의 상대적인 오프셋이라고 한다. 편차율 기반 알고리즘(rate_based algorithm)은 슬레이브에서 마스터와의 상대적인 편차율과 상대적인 오프셋을 계산하며 시간을 동기화하는 방법이고 많은 방법들이 제안되었다[1-3].

무선 환경에서 편차율 기반 알고리즘을 사용하는 [7]은 무선 통신 매체의 브로드캐스트 특성에 의해 수신자들이 거의 동일한 순간에 메시지를 받는다는 특징을 이용하여 편차율을 계산하는 방법을 제시하였다.

그림 1과 같이 일반적인 일대일 메시지를 통하여 시간 정보를 교환하는 경우 전송 시간, 접근 시간, 전파 시간, 수신 시간의 차이에 의해 동기화 에러가 발생한다. 하지만 그림 2와 같이 무선 브로드캐스트를 사용하면 전송 시간과 접근 시간에 영향을 받지 않고, 전파 시간 역시 전자기파가 공기 중으로 이동하는 점을 감안하면 0에 가까운 값이므로 고려하지 않아도 된다. 그러므로 수신 시간이 크게 변하지 않는다는 가정 하에 편차율을 계산할 수 있다. 시간 동기화를 위해 AP (Access Point)는 정기적으로 높은 우선순위의 메시지를 보내는 마스터의 역할을 수행하고, 이 메시지는 마스터를 포함

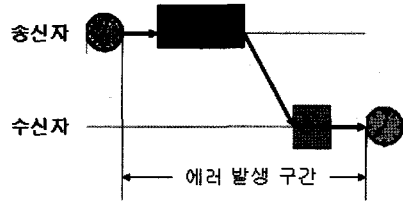


그림 1 일대일 메시지 전송

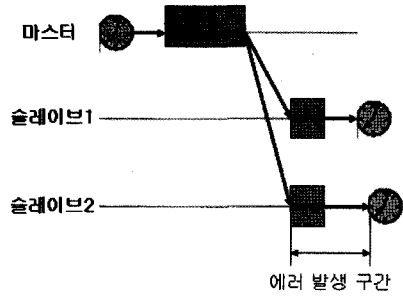


그림 2 무선 브로드캐스트

한 모든 슬레이브에게 브로드캐스트로 전송된다. 전송되는 메시지에는 마스터 자신이 바로 이전 메시지를 수신하였을 때 측정된 시간 정보가 포함되며, 이 값과 슬레이브에서 이전에 측정했던 값은 브로드캐스트의 특성으로 인해 거의 동일한 순간에 측정된 값들이다. 따라서 슬레이브는 이 값들로부터 편차율을 계산해 나갈 수 있다.

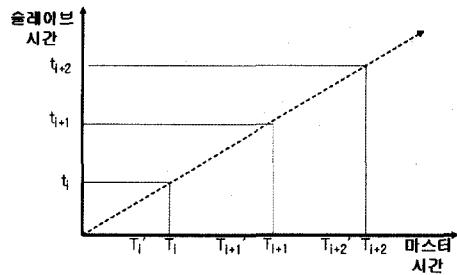


그림 3 두 시스템간 편차율

그림 327은 시간이 흐름에 따라 변화하는 마스터와 슬레이브의 물리적 시간 관계를 보여준다. 마스터가 이전 메시지 수신 시간을 포함한 동기화 메시지를 T_1' , T_{i+1}' , T_{i+2}' 에 브로드캐스트 한다면 마스터는 T_1 , T_{i+1} , T_{i+2} 에 메시지를 수신하고, 슬레이브는 t_1 , t_{i+1} , t_{i+2} 에 메시지를 수신하게 된다. 슬레이브는 t_{i+2} 에 메시지를 받은 순간 T_1 , T_{i+1} , t_1 , t_{i+1} 의 시간 정보를 가질 수 있고, 이때 T_1 에서 T_{i+1} 로 지나간 마스터의 시간 변화는 t_1 에서 t_{i+1} 로 변화한 슬레이브 측의 시간 변화와 대응한다. 그러므로 편차율과 오프셋을 통하여 슬레이브는 마스터의 시

간을 다음과 같이 표현할 수 있다.

$$T_i = \alpha \cdot t_i + b \quad (3)$$

$$T_{i+1} = \alpha \cdot t_{i+1} + b \quad (4)$$

두 개의 식을 통해 편차율과 오프셋을 다음과 같이 계산할 수 있다.

$$\alpha = \frac{T_i - T_{i+1}}{t_i - t_{i+1}} \quad (5)$$

$$b = T_i - \alpha \cdot t_i \quad (6)$$

편차율 기반 알고리즘을 사용한 방법은 메시지의 손실이 발생하였을 경우 이전에 계산되어진 편차율을 이용하여 연속적으로 동기화해 나갈 수 있다는 장점이 있다.

3. 무선 랜에서의 시간 동기화 기법

3.1 가상 시간 계산

본 논문에서는 다음과 같은 가정에 기반을 두어 가상 시간을 계산한다.

- 모든 시스템에서 시간을 측정하는데 사용되는 진동기는 일정한 진동수를 가지며 동작한다. 즉, 편차값은 일정한 변화값을 가지고 편차율은 일정하다.
- 마스터는 데스크톱이나 서버와 같은 일반 컴퓨터로서 임베디드 시스템이 가지는 자원 제약사항이 적고 시간을 측정하는 작업이 안정적이다.
- 수신 시간은 결정적 범위(deterministic bound)를 가지고, 마스터는 슬레이브보다 수신 시간이 일정한 값을 가진다.

제한하는 동기화 기법은 [7]과 유사하게 무선 브로드캐스트의 특성을 사용하여 시간을 동기화한다. 하지만 임베디드 환경에서는 브로드캐스트 메시지를 수신하고 자신의 시간을 측정하는 작업이 다른 중요한 작업에 의해 지연될 수 있다. 즉, 마스터와 슬레이브가 동시에 메시지를 받는 경우에 슬레이브가 측정하는 시간은 마스터보다 매우 큰 지연 시간이 포함될 수 있다. 이와 같이 슬레이브에서 메시지를 받는 시간을 측정하는 작업이 오차를 가질 경우 기존 방법으로 계산된 편차율이 정확하지 않게 되고 오프셋마저 부정확하게 된다. 그러므로 편차율의 작은 오차로도 식 (2)로 계산된 마스터의 시간은 부정확해지게 된다.

그림 4와 같이 마스터가 브로드캐스트를 하고 마스터 자신이 수신한 시간을 측정하여 다시 브로드캐스트를 수행하는 상황을 고려해 보자. 편차율을 계산하는 식 (5)에서의 분모인 슬레이브의 시간차는 임베디드 환경에서 오차를 가질 수 있으므로 슬레이브의 시간차를 마스터의 시간으로 변환하여 계산할 수 있다. 마스터가 브로드캐스트 메시지를 받는 시간의 차는 슬레이브가 브로드캐스트 메시지를 받는 시간의 차로 다음과 같이 표현할 수 있다.

$$T_{i+1} - T_i = t_{i+1} - t_i + C \quad (7)$$

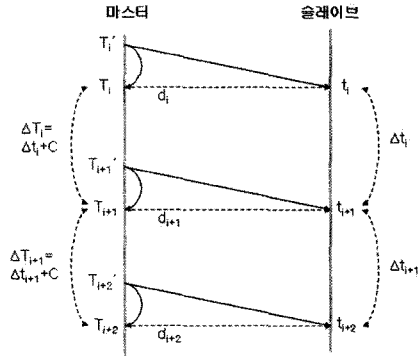


그림 4 마스터의 브로드캐스트

식 (7)에서 C는 브로드캐스트 메시지 수신 시간 동안 두 개의 메시지를 동시에 받은 마스터와 슬레이브 간에 발생한 시간의 편차를 의미한다. 브로드캐스트 수신 시간이 일정하다면 두 시스템간 시간차의 변화는 일정한 C만큼의 차이를 보이게 된다. 만약 슬레이브에서 브로드캐스트를 수신하여 측정하는 시간이 증가하면 두 시스템간 시간차의 변화는 지연값의 증가분만큼을 더 가지게 된다. i번째 브로드캐스트 메시지를 수신했을 때의 마스터와 슬레이브의 시간차를 d_i 라 하고 이때의 편차값을 C_i 라고 하면 식 (7)은 다음과 같이 표현될 수 있다.

$$C_i = (T_{i+1} - t_{i+1}) - (T_i - t_i) = d_{i+1} - d_i \quad (8)$$

정확한 C의 값을 측정할 수는 없기 때문에 브로드캐스트 메시지를 수신할 때마다 두 시스템간 시간차의 변화량을 측정하여 초기 변화량과 비교함으로써 C를 다음과 같이 유추할 수 있다.

$$\begin{aligned} d_2 - d_1 &= C_1 \approx C \\ d_3 - d_1 &= (d_3 - d_2) + (d_2 - d_1) = C_2 + C_1 \approx 2C \\ &\vdots \\ d_{n+1} - d_1 &= C_n + C_{n-1} + \dots + C_2 + C_1 \approx nC \end{aligned}$$

위의 식을 모두 더하면 다음과 같이 유추된 C의 값인 C_n 을 계산할 수 있다.

$$\sum_{i=2}^{n+1} (d_i - d_1) = C + 2C + \dots + nC = \frac{n(n+1)}{2} C \approx \frac{n(n+1)}{2} C_n$$

$$\text{따라서, } C_n = \frac{2 \sum_{i=2}^{n+1} (d_i - d_1)}{n(n+1)} \quad (9)$$

계산된 C_n 으로부터 편차율을 계산하기 위해서 식 (7)을 다음과 같이 변환한다.

$$\begin{aligned} T_{n+1} - T_n &= t_{n+1} - t_n + C_n \\ T_{n+1} - T_n &= \frac{T_{n+1} - b}{\alpha} - \frac{T_n - b}{\alpha} + C_n \\ (1 - \frac{1}{\alpha}) &= \frac{C_n}{T_{n+1} - T_n} \\ \alpha &= \frac{T_{n+1} - T_n}{T_{n+1} - T_n - C_n} \\ \alpha &= \frac{\Delta T_n}{\Delta T_n - C_n} \quad [T_{n+1} - T_n = \Delta T_n \text{로 대체}] \quad (10) \end{aligned}$$

이렇게 구해진 편차율을 식 (2)에 적용하면 가상 시간을 구할 수 있으나 식 (2)는 동기화가 진행됨에 따라 슬레이브의 시간인 t_s 가 증가하므로 편차율의 작은 오차에도 가상 시간의 정확성이 떨어진다는 단점을 가진다. 그러므로 식 (2)에서 사용된 t_s 를 줄이기 위해 식 (2)를 만족하는 임의의 점 (t_{bs} , t_{bm})을 원점으로 하도록 그래프를 평행 이동하면 될 것이다. 그러므로 식 (2)를 수정하여 가상 시간은 다음과 같이 계산된다.

$$T = \frac{\Delta T_n}{\Delta T_n - C_n} (t_s - t_{bs}) + T_{bm} \quad (11)$$

식 (11)에서 T_{bm} 을 마스터 기준 시간이라고 하며, t_{bs} 를 슬레이브 기준 시간이라고 한다. 마스터 기준 시간과 슬레이브 기준 시간은 임의의 순간의 두 시스템의 시간을 나타내고 이는 브로드캐스트의 특성을 통해 쉽게 구할 수 있는 값이다.

식 (10)으로 계산된 편차율은 식(5)와 달리 슬레이브의 시간의 수신 시간 에러를 감소하기 위하여 마스터의 시간과 누적된 편차값 C_n 만을 이용하여 계산하였다. 이때 C_n 은 시간차의 변화를 누적하여 평균을 내었으므로 수신 시간 에러로 인한 영향을 크게 줄일 수 있다. 그리고 식 (11)을 사용하여 계산된 가상 시간은 기준 시간을 사용함으로써 식 (2)를 사용한 방법보다 편차율의 오차에 의한 영향이 작다.

3.2 동기화 방법

제안하는 방법은 IEEE 802.11 표준에서 권고하는 마스터/슬레이브 방식을 사용하며, 초기화 단계와 재 동기화 단계로 구분되어 동작한다. 각 단계는 [7]에서 제시된 브로드캐스트 통신의 특성을 활용하여 수신 시간의 에러에 관계없이 마스터의 시간을 측정하여 높은 정확성을 제공한다. 또한 시간 동기화의 작업 부하를 줄이기 위해 마스터로부터 매 주기마다 브로드캐스트되는 메시지 하나만으로 슬레이브는 3.1절에서 설명된 방법으로 편차값을 통해 편차율을 계산하여 마스터의 시간을 추정하여 가상 시간을 계산한다.

초기화 단계에서는 초기 기준 시간을 구하기 위하여 마스터가 두 번의 브로드캐스트를 수행한다. 마스터는 $T_{i'}$ 에 초기화 메시지를 브로드캐스트하고 마스터와 슬레이브는 각각 거의 동일한 시간인 T_i 과 t_i 에 메시지를 수신한다. 마스터는 자신이 메시지를 수신한 시간인 T_i 을 포함한 메시지를 다시 브로드캐스트한다. 그러면 슬레이브는 T_i 시간을 알 수 있고, T_i 과 t_i 이 물리적으로 같은 시간이므로 각각 마스터 기준 시간과 슬레이브 기준 시간으로 설정한다. 또한 d_i 를 계산하여 유지한다.

재 동기화 단계에서는 편차율과 최신 기준 시간을 구하기 위하여 마스터가 브로드캐스트 메시지를 수신했던 최근 시간을 브로드캐스트한다. 마스터가 $T_{i+1'}$ 에 T_i 시

간을 브로드캐스트하면 슬레이브는 T_i , t_i 를 가지게 되어 d_i 를 계산할 수 있다. 그러면 식 (9)를 통해 C_n 을 계산할 수 있다. 이를 위해 슬레이브는 이전 단계에서 계산되었던 $\sum(d_i - d_1)$ 와 재 동기화 횟수인 n 을 유지한다. 계산된 C_n 값을 통해 편차율을 계산하고, T_i 와 t_i 를 새로운 기준 시간으로 하여 식 (11)을 통해 마스터의 시간을 유추할 수 있다.

제안된 시간 동기화 방법은 초기화 단계에서만 두 번의 브로드캐스트가 필요하고, 그 이후부터는 재 동기화 단계에서 한 번의 브로드캐스트만을 통하여 가상 시간을 계산할 수 있으므로 메시지 전송에 따른 오버헤드가 적고 대역폭 사용을 최대한 감소할 수 있다. 또한 재 동기화를 위해서는 세 가지 정보인 초기 편차값 d_i , 이전 편차값들의 합 $\sum(d_i - d_1)$, 그리고 재 동기화 횟수 n 만을 기억하면 되므로 임베디드 시스템의 자원을 절약할 수 있다는 장점을 가지고 있다.

3.3 동기화 에러

본 절에서는 제안하는 동기화 방법의 오차 구간을 구한다. 식 (11)을 통해 계산된 가상 시간이 마스터의 시간과 정확히 일치하는 경우는 슬레이브의 수신 시간 에러가 없이 C_n 이 C 가 되어 편차율을 계산할 때이다. 하지만 슬레이브는 임베디드 시스템의 제약 사항으로 인해 수신 시간 에러를 가지게 되어 C_n 은 오차를 포함하게 된다.

그림 2에서 슬레이브가 메시지를 수신하고 시간을 측정하기 전의 에러 발생구간인 수신 시간 에러가 최대인 경우의 시간을 ϵ_{max} 라고 하자. 식 (9)에서 구해지는 C_n 이 최대인 경우는 $d_i - d_1$ 의 합이 최대가 되는 경우이다. 즉, $(T_i - t_i) - (T_1 - t_1) = T_i - T_1 - t_i + t_1$ 의 값이 최대가 되는 경우이고 가정에 따라 T_i , T_1 은 에러가 없는 값이므로 t_i 의 값이 작고 t_1 의 값이 크면 된다. 이는 처음 브로드캐스트 메시지 수신 시간이 ϵ_{max} 만큼 늦고 이후의 브로드캐스트 메시지 수신 시간은 에러가 없는 경우이다. 그러므로 유추된 C 가 최대값인 C_{max} 는 다음과 같이 구할 수 있다.

$$\begin{aligned} C_{max} &= \frac{2 \sum_{i=2}^{n+1} (d_i - (d_1 - \epsilon_{max}))}{n(n+1)} \\ &= \frac{2 \sum_{i=2}^{n+1} (d_i - d_1) + 2n\epsilon_{max}}{n(n+1)} \\ &= \frac{2 \sum_{i=2}^{n+1} (d_i + d_1)}{n(n+1)} + \frac{2\epsilon_{max}}{(n+1)} \\ &= C_n + \frac{2\epsilon_{max}}{(n+1)} \end{aligned}$$

반대로 C_n 이 최소인 경우는 $d_i - d_1$ 의 합이 최소가 되기 위해 t_i 의 값이 크고 t_1 의 값이 작을 때이다. 이는 처음 브로드캐스트 메시지 수신 시간은 에러가 없고 이후의 브로드캐스트 메시지 수신 시간이 ϵ_{max} 만큼 늦게 오는 경우이다. 그러므로 유추된 C 가 최소값인 C_{min} 은 다음

과 같이 구할 수 있다.

$$C_{\min} = \frac{2 \sum_{i=2}^{n+1} ((d_i - \epsilon_{\max}) - d_1)}{n(n+1)}$$

$$= \frac{2 \sum_{i=2}^{n+1} (d_i - d_1) - 2n\epsilon_{\max}}{n(n+1)}$$

$$= \frac{2 \sum_{i=2}^{n+1} (d_i - d_1)}{n(n+1)} - \frac{2\epsilon_{\max}}{(n+1)}$$

$$= C_n - \frac{2\epsilon_{\max}}{(n+1)}$$

그러므로 C는 다음과 같은 범위를 가지고 있다.

$$C_n - \frac{2\epsilon_{\max}}{n+1} \leq C \leq C_n + \frac{2\epsilon_{\max}}{n+1}$$

식 (10)으로 계산되는 편차율이 최대가 되는 경우는 C가 최대인 경우이고, 가상 시간이 최대가 되는 경우는 편차율이 최대이며 t_s 가 ϵ_{\max} 만큼 지연된 측정 시간을 가지는 경우이다. 이를 식 (11)에 적용하면 가상 시간의 최대값을 구할 수 있다.

$$T_{\max} = t_{bm} + \frac{\Delta T_n}{\Delta T_n - (C_n + \frac{2\epsilon_{\max}}{n+1})} \cdot ((t_s + \epsilon_{\max}) - t_{bs})$$

[$\Delta t_n = \Delta T_n - C_n$, $E = \frac{2\epsilon_{\max}}{n+1}$ 로 대체]

$$= t_{bm} + \frac{\Delta t_n \cdot \Delta T_n}{\Delta t_n (\Delta t_n - E)} \cdot (t_s - t_{bs} + \epsilon_{\max})$$

$$= t_{bm} + \frac{\Delta t_n \cdot \Delta T_n + (E \cdot \Delta T_n - E \cdot \Delta T_n)}{\Delta t_n (\Delta t_n - E)} \cdot (t_s - t_{bs} + \epsilon_{\max})$$

$$= t_{bm} + \frac{(\Delta t_n \cdot \Delta T_n - E \cdot \Delta T_n) + E \cdot \Delta T_n}{\Delta t_n (\Delta t_n - E)} \cdot (t_s - t_{bs} + \epsilon_{\max})$$

$$= t_{bm} + \frac{\Delta T_n}{\Delta t_n} \cdot (t_s - t_{bs})$$

$$+ \frac{\Delta T_n (\Delta t_n - E) \cdot \epsilon_{\max} + E \cdot \Delta T_n \cdot (t_s - t_{bs} + \epsilon_{\max})}{\Delta t_n (\Delta t_n - E)}$$

$$= T + \frac{\Delta T_n (\Delta t_n - E + E) \cdot \epsilon_{\max} + E \cdot \Delta T_n (t_s - t_{bs})}{\Delta t_n (\Delta t_n - E)}$$

$$= T + \frac{\Delta t_n \cdot \epsilon_{\max} + E(t_s - t_{bs})}{\Delta t_n (\Delta t_n - E)} \cdot \Delta T_n$$

또한 편차율이 최소가 되는 경우는 C가 최소인 경우이고 식 (11)에 적용하면 가상 시간의 최소값을 계산할 수 있다. C가 최소인 경우는 슬레이브의 메시지 수신 시간이 지연되는 경우이므로 슬레이브 기준 시간은 ϵ_{\max} 만큼 지연된다.

$$T_{\min} = t_{bm} + \frac{\Delta T_n}{\Delta T_n - C_n + \frac{2\epsilon_{\max}}{n+1}} \cdot (t_s - (t_{bs} + \epsilon_{\max}))$$

[$\Delta t_n = \Delta T_n - C_n$, $E = \frac{2\epsilon_{\max}}{n+1}$ 로 대체]

$$= t_{bm} + \frac{\Delta t_n \cdot \Delta T_n}{\Delta t_n (\Delta t_n + E)} \cdot (t_s - t_{bs} - \epsilon_{\max})$$

$$= t_{bm} + \frac{\Delta t_n \cdot \Delta T_n + (E \cdot \Delta T_n - E \cdot \Delta T_n)}{\Delta t_n (\Delta t_n + E)} \cdot (t_s - t_{bs} - \epsilon_{\max})$$

$$= t_{bm} + \frac{(\Delta t_n \cdot \Delta T_n + E \cdot \Delta T_n) - E \cdot \Delta T_n}{\Delta t_n (\Delta t_n + E)} \cdot (t_s - t_{bs} - \epsilon_{\max})$$

$$= t_{bm} + \frac{\Delta T_n (\Delta t_n + E) - E \cdot \Delta T_n}{\Delta t_n (\Delta t_n + E)} \cdot (t_s - t_{bs} - \epsilon_{\max})$$

$$= t_{bm} + \frac{\Delta T_n}{\Delta t_n} \cdot (t_s - t_{bs})$$

$$- \frac{\Delta T_n (\Delta t_n + E) \cdot \epsilon_{\max} + E \cdot \Delta T_n \cdot (t_s - t_{bs} - \epsilon_{\max})}{\Delta t_n (\Delta t_n + E)}$$

$$= T - \frac{\Delta T_n (\Delta t_n + E - E) \epsilon_{\max} + E \cdot \Delta T_n (t_s - t_{bs})}{\Delta t_n (\Delta t_n + E)}$$

$$= T - \frac{\Delta t_n \cdot \epsilon_{\max} + E(t_s - t_{bs})}{\Delta t_n (\Delta t_n + E)} \cdot \Delta T_n$$

그러므로 마스터의 계산된 시간인 가상 시간 T는 다음과 같은 범위를 가진다.

$$T - \frac{\Delta t_n \cdot \epsilon_{\max} + E(t_s - t_{bs})}{\Delta t_n (\Delta t_n + E)} \cdot \Delta T_n \leq T \quad (12)$$

$$\leq T + \frac{\Delta t_n \cdot \epsilon_{\max} + E(t_s - t_{bs})}{\Delta t_n (\Delta t_n - E)} \cdot \Delta T_n$$

동기화가 진행됨에 따라 n의 값이 커지므로 E는 0에 가까워지고, 제안된 동기화 방법으로 계산되는 가상 시간의 에러 범위는 점차 ϵ_{\max} 에 비례하는 일정한 에러 범위를 가지게 된다.

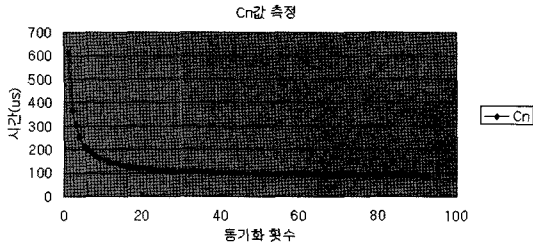
4. 실험

제안된 시간 동기화 방법 [7]에서 기술된 방법과 비교하여 실험하였다. 실험 환경은 Windows XP를 운영체제로 하는 센트리노 1.6GHz CPU의 노트북 PC가 마스터 역할을 수행하고, WinCE 3.0을 운영체제로 하는 ARM 200MHz CPU의 PDA가 슬레이브 역할을 수행하였다. AP는 IEEE 802.11b 표준에 기반을 둔 Linksys사의 BEFW11sS4 기종을 사용하였다.

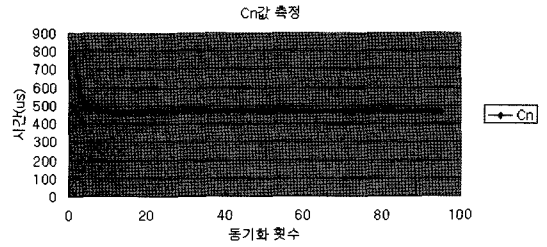
마스터는 시간 동기화 메시지를 동기화 주기마다 총 100회를 전송하였고, 슬레이브는 동기화 메시지를 받으면 편차율과 시간 차이를 계산하였다. 이때 동기화 메시지는 메시지 종류와 동기화 메시지 횟수, 타임스탬프를 포함하는 16 바이트로 구성하였다. 동기화 주기는 1초와 2초로 설정하여 각각 5회씩 실험하며 동기화 횟수에 따른 편차율과 시간차의 평균을 구하였다. 또한 마스터는 자신의 브로드캐스트 메시지를 수신하기 위해 전송되는 동기화 메시지의 MAC 주소와 IP 주소를 변경하여 전송하였다.

편차율을 정확하게 계산하기 위해서는 메시지를 받은 시간과 측정된 시간이 최대한 같아야 한다. 이를 위해 시간을 측정하는 작업이 하드웨어나 드라이버 레벨에서 이루어져야 하고, 본 실험에서는 winpcap 라이브러리 [11]을 사용하여 마스터와 슬레이브의 메시지 수신 시간을 측정하였다.

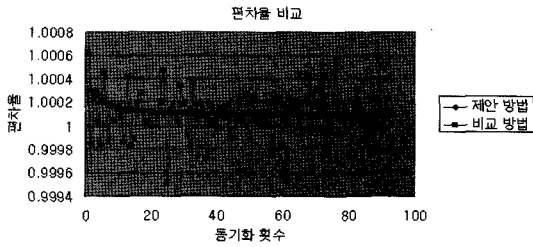
그림 5와 그림 6의 (a)는 동기화 횟수에 따른 C_n의



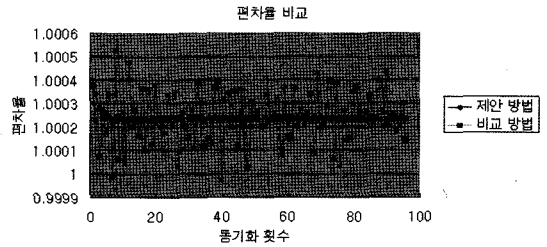
(a) Cn 값의 변화



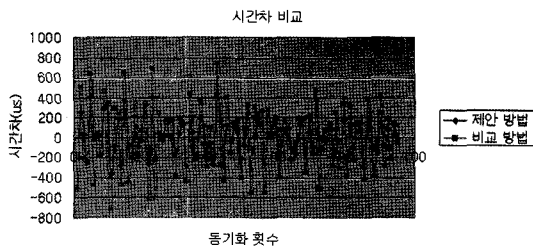
(a) Cn 값의 변화



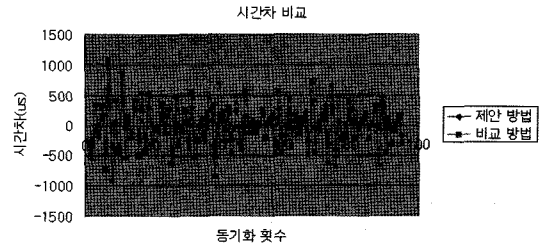
(b) 편차율의 변화



(b) 편차율의 변화



(c) 시간차 비교



(c) 시간차 비교

그림 5 동기화 주기 1초일 때의 성능 분석 결과

그림 6 동기화 주기 2초일 때의 성능 분석 결과

변화값을 나타낸다. 동기화 초기에는 무선 환경에서 발생하는 쿨 스타트(cool start) 현상으로 불안정한 모습을 보이지만 동기화가 20회 정도에 이르면 거의 일정한 값을 가지며 안정되는 것을 알 수 있다.

그림 5와 6의 (b)는 측정된 C_n 으로부터 식 (10)을 통해 계산된 편차율과 기존 방법인 식 (5)를 통해 계산된 편차율을 보여준다. 제안된 방법으로 계산된 편차율은 쿨 스타트 기간을 지나면서 수신 시간 에러에 상관없이 일정하고 안정된 편차율을 가지지만, 기존 방법은 편차율의 변화가 심하다는 것을 알 수 있다. 이는 제안된 기법은 누적된 편차값을 통해 편차율을 계산하므로 동기

화가 진행됨에 따라 수신시간 에러에 의한 영향이 작지만, 기존 방법은 현재의 시간 정보만을 사용하여 편차율을 계산하기 때문에 동기화 진행과 상관없이 메시지 수신시 발생하는 수신 시간 에러에 크게 좌우되기 때문이다.

그림 5와 6의 (c)는 계산된 편차율을 통해 시간 동기화 에러를 측정된 그래프이다. 시간 동기화 에러를 측정하기 위해서 동기화 메시지를 받았을 때 현재 슬레이브의 시간과 이전 단계에서 계산된 편차율을 가지고 가상 시간을 계산하여 마스터의 시간과 비교하였다. 제안된 방법의 시간 동기화 에러가 기존의 방법보다 동기화 에러가 작다는 것을 알 수 있다. 특히 쿨 스타트 기간을

표 1 시간 동기화 에러

비교 \ 측정값(us)	동기화 주기 1초			동기화 주기 2초		
	평균	최대값	최소값	평균	최대값	최소값
제안 방법	-9.083±196.672	438.209	-505.166	-0.002±235.238	604.022	-554.575
비교 방법	-2.846±329.537	739.475	-724.531	-12.228±404.199	1077.038	-926.459

지난 이후의 동기화 에러는 최대·최소 에러의 차이가 일정한 모습을 보이며 안정화되고, 이는 동기화가 진행됨에 따라 식 (12)에서의 E값이 0에 가까워지기 때문이다. 하지만 기존 방법은 일시적인 시간 정보만을 사용하여 동기화하기 때문에 전 단계에 걸쳐 일정하지 않은 동기화 에러를 보이고 있다.

표 1은 시간 동기화 에러 통계를 나타낸다. 제안된 방법의 표준 편차가 기존 방법의 표준 편차보다 100 μ s 이상 작으므로 성능이 향상되었다는 것을 알 수 있다. 또한 최대·최소값도 기존 방법에 비해 2배정도 작으므로 수신 동기화 에러가 적고 안정적임을 알 수 있다.

5. 결론

본 논문에서는 임베디드 시스템을 대상으로 무선 랜 환경에서 사용가능한 소프트웨어적인 시간 동기화 기법을 제안하였다. 제안된 기법은 IEEE 802.11 표준에서 권고하는 마스터/슬레이브 형태를 기반으로 브로드캐스트 통신의 특성을 활용하여 메시지 지연 시간에 관계없이 상대방의 시간을 측정하여 높은 정확성을 제공할 수 있다. 본 논문에서 제안된 편차율 계산 방법은 마스터의 시간과 편차값만을 이용하여 계산하였다. 이때 편차값은 시간차의 변화를 누적하여 평균을 내었으므로 수신 시간 에러로 인한 영향을 크게 줄일 수 있었다. 또한 가상 시간을 추정할 때 기준 시간을 사용함으로써 편차율의 오차에 의한 영향을 감소하였다. 제안된 시간 동기화 방법은 초기화 단계에서만 두 번의 브로드캐스트가 필요하고, 그 이후부터는 재 동기화 단계에서 한 번의 브로드캐스트만을 통하여 가상 시간을 계산할 수 있으므로 메시지 전송에 따른 오버헤드가 적고 대역폭 사용을 최대한 감소할 수 있다. 또한 재 동기화를 위해서는 세 가지 정보만을 기억하면 되므로 임베디드 시스템의 자원을 절약할 수 있다는 장점을 가지고 있다. 실험을 통해 제안된 시간 동기화 알고리즘을 사용하는 경우 기존의 무선 임베디드 환경에서 사용되는 시간 동기화보다 성능이 우수함을 입증하였다.

참 고 문 헌

[1] K. Arvind, "Probabilistic Clock Synchronization in Distributed Systems," *IEEE Trans. On Parallel and Distributed Systems*, Vol. 5, No. 5, pp. 474-487, May 1994.
 [2] F. Cristian, "Probabilistic Clock Synchronization," *Distributed Computing* 3, pp. 146-158, 1989.
 [3] H. Kopetz and W. Schwabl. "Global time in distributed real-time systems," Technical Report 15/89, Technishe Univesitt Wien, 1989.

[4] 김남희, 이상태, 김대찬, 노기용, 송양섭, 전병실, "세계 표준시간 분배시스템의 개발", 한국정보과학회 논문지 : 컴퓨팅의 실제, 제5권 제3호, pp. 366-374, 1999년 6월.
 [5] 유민수, 박정근, "시간 동기화 문제의 재 고찰 : 실시간 시스템을 위한 정적/동적 제약 변환 기법", 한국정보과학회 논문지 : 시스템 및 이론, 제26권 제10호, pp. 1264-1274, 1999년 10월.
 [6] Y. S. Hong, J. H. No, "Clock Synchronization in Wireless Distributed Embedded Applications," IEEE Workshop on Software Technologies for Future Embedded Systems, pp. 101-104, May 2003.
 [7] M. Mock, R. Frings, E. Nett, and S. Trikiotis, "Clock Synchronization for Wireless Local Area Networks," 12th Euromicro Conference on Real-Time Systems (ECRTS), pp. 183-189, June 2000.
 [8] David L. Mills, "Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI," IETF RFC 2030, Network Working Group, October 1996.
 [9] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," IEEE Std. 802.11-1997, November 1997.
 [10] John R. Vig, "Introduction to Quartz Frequency Standards," Technical Report SLCET-TR-92-1, Army Research Laboratory, Electronics and Power Sources Directorate, October 1992.
 [11] Loris Degioanni, Mario Baldi, Fulvio Rizzo and Gianluca Varenni, "Profiling and Optimization of Software-Based Network-Analysis Applications," Proceedings of the 15th IEEE Symposium on Computer Architecture and High Performance Computing (SBAC-PAD 2003), pp. 226-234, November 2003.



노진홍
 1999년 동국대학교 컴퓨터공학과(공학사). 2001년 동국대학교 대학원 컴퓨터공학과(공학석사). 2001년~현재 동국대학교 박사과정. 관심 분야는 객체 지향 분산 실시간 처리, 고장 감내 시스템, 임베디드 시스템, 센서 네트워크



홍영식
 1973년 서울대학교 응용수학과(공학사) 1975년 한국과학원 수학 및 물리학과(이학석사). 1976년~현재 동국대학교 컴퓨터,멀티미디어공학과 교수. 1983년~1984년 미국 Rensselaer Polytechnic Institute 방문교수/강사. 1989년~1990년 미국 University of North Carolina-Charlotte 방문교수. 관심분야는 분산 처리, 객체 지향 분산 실시간 처리, 고장 감내 시스템, 암호 알고리즘