

A Revised Timing-sync Protocol for Sensor Networks by a Polling Method

Shi-Kyu Bae *

Abstract

TPSN(Timing-sync Protocol for Sensor Networks), the representative of time synchronization protocol for WSN(wireless sensor networks), was developed to provide higher synchronization accuracy and energy efficiency. So, TPSN's approach has been referenced by so many other WSN synchronization schemes till now. However, TPSN has a collision problem due to simultaneous transmission among competing nodes, which causes more network convergence delay for a network-wide synchronization. A Polling-based scheme for TPSN is proposed in this paper. The proposed scheme not only shortens network-wide synchronization time of TPSN, but also reduce collision traffic which lead to needless power consumption.

The proposed scheme's performance has been evaluated and compared with an original scheme by simulation. The results are shown to be better than the original algorithm used in TPSN.

▶ Keyword : Time Synchronization, TPSN(Timing-sync Protocol for Sensor Networks), Wireless Sensor Network, Collision, Polling.

I. Introduction

무선센서네트워크(WSN : Wireless Sensor Networks)의 연구는 여러 분야에서 다양하게 이루어져오고 있는데, 그 중에 하나인 시각 동기화는 최근에 새로 시작된 것이 아니라 인터넷 초기 시절부터 전통적으로 행해져 왔으나, 무선 네트워크 환경, 특히 무선 센서 네트워크 분야에서의 새로운 요구사항을 고려하여 다른 시각에서 이루어지고 있다.

인터넷과 같은 유선 네트워크에서의 시각 동기화 프로토콜로는 대표적으로 NTP (Network Time Protocol)[1]가 개발되었다. WSN을 구성하는 각 센서 노드는 서로 협동작업을 하기 위하여 공통 클럭을 유지하는 것이 중요하며, 응용에서는 네트워크 전체 혹은 일부 센서 노드간에 시각 동기화가 유지되는 것을 요구 한다. 그리고 무선 센서 네트워크를 설계할 때 공통적으로 전력 소모 최소화가 중요하며, 시각 동기화 프로토콜을 설계할 때에도 마찬가지로 전력 소모를 최소한으로 줄이는 것이 필요하다[2].

WSN에서 시각 동기화 문제를 해결하기 위하여 여러 프로토콜이 개발되었는데, TPSN(Timing-sync Protocol for Sensor Networks)[3]도 그 중에 하나이며, NTP를 기반으로 한다. TPSN은 동기화 전에 레벨 탐색이라는 과정을 통해 계층 구조를 생성하며, 이 계층 구조를 사용하여 네트워크의 전체 노드를 순차적으로 동기화한다. 전체 네트워크를 동기화하는 과정에서 낮은 레벨에 위치하는 센서 노드들은 한 레벨 높은 위치의 노드와 패킷을 동시에 교환하며 서로 경쟁하게 되고 충돌이 발생할 수 있다. 여러 노드들이 패킷을 전송하는 과정에서 충돌이 발생하면 전체 네트워크를 동기화하는데 걸리는 시간이 길어질 뿐 만 아니라 더 많은 전력이 소모된다.

이 논문에서는 TPSN에서의 충돌을 줄여 네트워크 동기화 시간과 전력 소모를 줄인 개선된 TPSN 알고리즘을 제안한다.

논문의 구성은 다음과 같다. 2장에서 기존의 무선센서네트워크용 시각 동기화 프로토콜과 TPSN의 동작을 살펴본다. TPSN의 문제점을 정리하고 개선된 TPSN 알고리즘을 3장에서 제안한다. 4장에서 제안한 알고리즘을 적용하여 성능을 분

• First Author: Shi-Kyu Bae, Corresponding Author: Shi-Kyu Bae
*Shi-Kyu Bae(skbae@dyu.ac.kr), Dept. of Cyber Security, DongYang University
• Received: 2015. 06. 24, Revised: 2015. 07. 12, Accepted: 2015. 08. 18.

석하며, 마지막으로 5장에서 결론을 다룬다.

II. Related Works

2.1 WSN Time Synchronization Protocols

지금까지 개발된 여러 시각 동기화 프로토콜들은 송신자와 수신자간의 메시지를 주고받는 형태에 따라 송신-수신형과 수신-수신형으로 나뉜다.

송신-수신형 동기화는 한 노드가 송신자가 되어 데이터를 전송하며, 송신자를 제외한 다른 노드들은 수신자가 되어 보내어진 데이터를 수신하며, TPSN(Timing-sync Protocol for Sensor Networks)[3]과 FTSP(Flooding Time Synchronization Protocol)[4]이 여기에 속하고, 글로벌 클럭을 전체 네트워크에 전달할 수 있다. 이와 대조적으로 수신-수신형 동기화에서는 참조 노드가 특정한 패킷을 한 번 전송하면, 이 후에는 전송한 노드를 제외한 다른 노드들이 서로 협력하여 동기화를 이루는 방식을 갖는다. 이 종류에 해당하는 대표적인 프로토콜이 RBS(Reference Broadcast Synchronization) [5]로서, 한 노드(송신자)가 다른 수신 노드들이 동시에 수신하도록 비이컨 메시지를 방송하면, 송신자를 제외한 다른 수신자들이 이 비이컨 메시지를 수신한 각각의 로컬 클럭 값을 서로 교환하여 상호간의 클럭 오프셋을 기록한다. 이 RBS 프로토콜은 가까운 거리에 위치한 여러 노드간에 상대적인 클럭 오프셋을 비교적 정확하게 파악할 수 있지만, 특정 노드의 클럭(혹은 글로벌 클럭)으로 네트워크 전체를 동기화할 수는 없다. RBS와 FTSP의 장점을 결합한 혼합형 방안[6]이나, TPSN의 성능을 개선하기 위한 여러 제안들이 있었다 [7,8]. [7]의 논문에서는 클러스트의 레벨을 제한시켜 트리의 깊이를 줄임으로써 성능을 개선시키는 방안을 발표하였다. [8]의 저자들은 자식 노드가 가장 낮은 레벨의 노드를 부모 노드로 선택하는 방안을 제안하였다.

2.2 Operation of TPSN[3]

TPSN은 전형적으로 트리 구조에서 동작하며, 루트 노드로부터 시작하여 상위 레벨의 노드와 연결된 하위 레벨의 노드들간의 메시지 교환을 통해 동기화를 이루고, 순차적으로 전체 네트워크를 동기화한다. 그림1에 나타난 것처럼 루트 노드(Level 0로 표시)와 연결된 Level 1 노드 간에 동기화가 이루어진 후, 연속적으로 Level 1 노드와 연결된 Level 2 노드들 간에 동기화가 이루어져, 최종적으로 네트워크의 모든 노드가 동기화가 된다.

TPSN의 동작을 좀 더 구체적으로 살펴보면, TPSN은 두 개의 단계, 즉, 레벨 탐색 단계와 동기화 단계로 이루어진다. 레벨 탐색 단계에서는 트리 구조가 생성된다. 트리 생성은 플러딩(flooding)을 약간 변형한 방식으로 이루어진다. 동기화 단계에서 모든 노드는 상위 레벨의 노드와 양방향 통신을 통해서 동

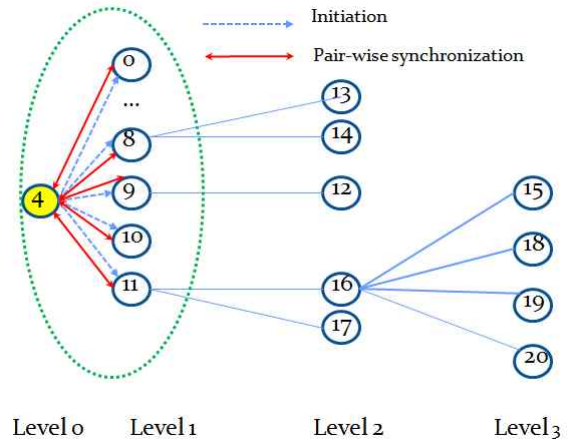


Fig. 1 Operation of TPSN

기화를 이루며 궁극적으로 모든 노드는 루트 노드와 동기화를 이룬다. 각 레벨에서 부모 노드와 연결된 자식 노드 간에는 메시지 교환이 이루어지는데, 부모노드의 시작 메시지를 시작으로 각 자식 노드가 적절한 시기에 부모 노드에게 동기화 요청 메시지를 보내고 이에 대한 응답을 부모노드로부터 받음으로서 그 노드의 동기화가 이루어진다. 동기화가 완료된 노드는 다시 연결된 자식 노드와 동기화를 시도한다. 이 때, 여러 노드가 한 부모 노드와의 메시지 전송을 동시에 시도하면 충돌이 일어나므로, TPSN에서는 각 노드에서 전송을 시도하기 전에 임의의 대기 시간을 갖도록 하고 있다.

III. Revised TPSN Algorithm

TPSN에서 적용하고 있는 각 노드의 임의의 대기 시간은 적정 값을 정하기 어려우므로 충돌을 완전히 해결할 수 없어 충돌에 의한 대기 지연 시간 및 재전송 등으로 인한 추가적인 네트워크 동기화 지연과 전력 소모의 오버헤드가 발생한다.

실사 충돌이 일어나지 않더라도 이 대기 시간은 각 노드의 동기화 완료 시간을 지연시키고 최종적으로 전체 네트워크의 동기화 완료 시간에 영향을 끼치며, 네트워크 동기화 완료 시간이 길어짐으로서 이에 따라 전력 소모도 커지게 된다. 네트워크 전체 동기화가 완료되기 전에 발생하는 이벤트의 시간 정확도는 응용에 영향을 끼치게 되므로, 네트워크 전체 동기화 완료 시간은 가능하면 작을수록 좋다.

이 논문에서 제안하는 방안은 TPSN의 동작에서 부모 노드와 자식 노드간의 동기화에서 메시지 전송을 경쟁 방식이 아닌 폴링 방식을 사용함으로써 노드간의 불필요한 충돌을 줄이고 지연 시간을 절약하여 네트워크 전체 동기화 완료 시간을 줄이기 위한 것으로, P-TPSN(Polling-based TPSN)이라고 부르기로 한다.

P-TPSN도 TPSN과 유사하게 트리 구성과 동기화의 두 개의 동작 상태를 갖는다.

3.1 Operation of P-TPSN : Tree Construction

TPSN의 탐색 과정에서의 트리 구성은 간단한 플러딩 (flooding) 방식을 사용하는데, 생성된 트리에서 동기화를 실행하면 레벨 수가 증가할수록(즉, 트리의 깊이가 클수록) 루트 노드를 기준으로 한 말단 노드의 클럭 오차는 점점 커지며, 전체 노드의 평균 동기화 오류도 증가하게 된다. 따라서 트리의 구성은 깊이가 얇을수록 동기화 오류가 개선된다. 그림 1에서 예를 들면, 레벨 2에 있는 노드 12와 노드 16은 노드 레벨 1에 있는 노드 0이나 노드 11보다 덜 정확하고, 가장 낮은 레벨에 있는 노드 15나 노드 20이 가장 오류가 크다.

P-TPSN에서는 가능한 깊이가 얇은 트리를 구성하기 위하여 [9]에서 제안한 트리 구성 알고리즘과 유사하지만 다음과 같이 트리 구성 알고리즘을 작성하였다.

```

Radius : radius of broadcast range
if (root node)
Broadcast a discovery-level packet (node-id, level#)
For all nodes except root node {
  If (packet received && firstly received) {
    Set level=parent node's level+ 1;
    Distance = measure distance
      between parent node and node itself;
    Wait (k*Radius/distance) seconds;
    Broadcast additional discovery-level packet
      including its id to notify the parent;
    // Send its id to the parent to notify a real child;
  }
}
    
```

< Tree Construction Algorithm.>

트리 구성 알고리즘이 실행되면 그림 1과 같은 트리가 구성되는데, 루트 노드를 포함한 각 부모 노드는 자식 노드의 정보를 가지게 되며, 말단에 있는 노드(leaf nodes)는 자식 노드로부터 통지를 받지 않으므로 부모 노드가 되지 않는다 (그림 1에서 예를 들면, 노드 12, 노드 17, 노드 15 등은 부모 노드가 아니다).

TPSN의 동작에서는 부모 노드인지 혹은 아닌지(즉, 말단 노드인지)의 역할만 인식하고, 자식 노드들은 부모 노드의 ID를 알고 있어야 한다. 하지만, P-TPSN에서는 부모 노드가 자식 노드들의 ID 리스트를 알고 있어야 하므로, 트리 형성 과정에 이 부분이 추가되었다.

3.2 Operation of P-TPSN : Synchronization Phase

부모 노드와 자식 노드사이의 동기화는 다음과 같이 동작한다. TPSN에서는 부모 노드가 동기화 시작을 자식 노드들에게 통지하면, 각 자식 노드들은 임의의 지연 시간 후 부모 노드와 메시지 교환을 시작한다. 각 자식 노드는 부모 노드에게 요청 메시지를 보내고 부모 노드로부터 응답 메시지를 받으면 메시지에 포함된 시각 정보를 사용하여 부모노드에 대하여 동기화한다. 동기화가 완료된 노드들은 다시 자식노드들에게 부모 노드 역할을 하여 동기화를 연속적으로 실시한다. 모든 말단 노드까지 동기화가 끝나면 네트워크 전체의 동기화가 완료된다. 이 과정에서 각 노드들은 인근 노드들과의 충돌을 겪게 되고 재전송함으로써 추가적인 전력 소비와 지연 시간이 증가하게 된다.

P-TPSN에서는 각 부모 노드가 자식 노드들에게 순차적으로 메시지 교환을 할 수 있도록 해당 자식 노드를 지정하여 메시지 요청을 하도록 한다.

그림 2는 부모노드와 자식 노드간의 동기화 메시지 교환 시의 시간 관계를 나타낸 것이다. 부모 노드가 "Initiate" 패킷을 보내어 자식 노드들이 순서대로 "REQ" 패킷을 보내도록 한다. 이때 자식 노드의 리스트를 함께 보내어 각 자식 노드가 자신의 요청 순서를 알 수 있도록 한다.

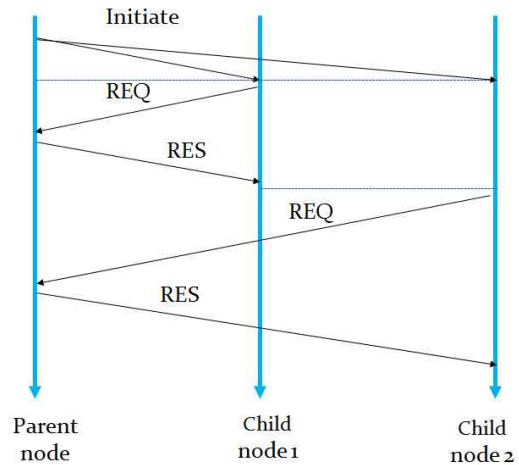


Fig. 2 timing diagram for message exchanges among parent and child nodes.

트리를 T라고 하고, 각 노드의 집합을 V, 두 노드간의 연결 링크를 E라고 할 때, $T = (V, E)$ 로 나타내기로 하자.

노드 i의 자식 노드 집합 C는 노드 i로부터 통신 반경 내에 있고 다음과 같이 정의할 수 있다.

$$C(i) = \{j \in V : i \neq j \text{ and } (i,j) \in E\} \quad (1)$$

노드 j의 부모 노드를 P(j)라고 하면, 식 (1)에서 모든 j에 대하여 $P(j) = i$ 이다.

한 부모 노드와 자식 노드간의 동기화 과정을 설명하면 다음과 같다.

(a) 부모 노드는 자식 노드들에게 동기화를 시작할 것을 알리는 "Initiate" 메시지를 자신의 ID와 함께 자식 노드들 리스트를 실어 전송한다.

(b) 자식 노드들은 부모 노드로부터 받은 "Initiate" 패킷 속의 리스트를 확인하여 자신이 "REQ" 패킷을 보낼 순서를 확인한다.

(c) 리스트의 첫 번째에 해당하는 자식 노드가 "REQ" 패킷을 보내어 부모 노드의 응답을 요청한다.

(d) 부모 노드는 첫 번째 자식 노드로부터 "REQ" 패킷을 받은 시간과 응답 시간을 기록하여 "RES" 패킷으로 요청한 자식 노드에게 다시 보낸다. 이 "RES" 패킷에는 다음에 동기화를 시도할 자식 노드의 ID를 함께 piggyback하여 첫 번째 자식 노드는 동기화를 완료하고, 동시에 두 번째 자식 노드가 동기화 요청을 하도록 한다 (그림 2 참조).

(e) 모든 자식 노드들이 순서대로 (c),(d) 동작을 반복하여 마지막 자식 노드까지 동기화가 완료되면 한 부모 노드와 하위 레벨에 있는 자식 노드간의 동기화 과정이 완료된다. 마지막 자식 노드로부터 "REQ" 패킷을 받은 부모 노드는 "RES" 패킷에 마지막 자식 노드를 위한 시간 정보뿐 아니라, 모든 자식 노드들의 동기화가 완료됨을 알리는 "FINISH" 영역 정보를 설정하여 보냄으로써 다른 자식 노드들도 동일 레벨이 동기화가 완료됨을 인지하도록 한다.

속할 것이다. 만일, 상위 레벨의 동기화 과정(즉, (e)의 동작)이 끝나기 전에 미리 현재 레벨의 동기화가 완료되고 다음 레벨의 동기화를 시도하면(즉, (d)동작) 충돌이 발생할 수 있다. 또한 다음 레벨의 동기화를 여러 현재 자식 노드들이 동시에 부모 노드로서 동기화를 Initiate하려고 하면 충돌이 발생할 수 있다. 이런 경우에서도 미리 동기화가 끝난 자식 노드는 자신과 동일 레벨에 있는 다른 자식 노드들의 정보를 알고 있으므로, 마지막 동일 레벨 자식 노드의 동기화가 끝날 때까지 기다렸다 다음 하위 레벨이 동기화를 시작함으로써 동일 레벨 자식 노드들과의 충돌을 미리 피할 수 있다. 다음은 제안한 알고리즘을 유사 코드로 나타낸 것이다.

IV. Simulation

알고리즘의 동작을 검증하기 위하여 그림 3과 같은 토폴로지를 설정하고 NS2(Network Simulator)-2.35[10]을 사용하여 그림 1의 트리를 생성하고 실험하였다.

공통되는 시뮬레이션 환경은 노드간 거리 100m, 전송 영역 반경 550m이고, 노드에서의 전력 변수로는 송신전력(0.660W), 수신전력(0.395W), 대기전력(0.035W)으로 설정하였다. 사용된 MAC은 802.11 그리고 라우팅 프로토콜은 DSDV이다.

```

< P-TPSN Algorithm >
Use the current Tree
For root node {
    send "INIT" with children list
    If (received "REQ" from first child)
        send "RES" with time info, next_turn child_ID.
}
For child node {
    If (received "INIT" from parent && when its turn)
        send "REQ"
    If (received "RES" from parent for its REQ)
        Do Synchronize
    If (received "RES" from parent && when its turn)
        send "REQ"
}
For Parent node except root {
    If (received "INIT" from parent && when its turn)
        send "REQ"
    If (received "RES" from parent for its REQ)
        Do Synchronize
    If (received "RES" from parent && when last child)
        send "INIT" //
}
    
```

< Pseudo code of the proposed algorithm.>

동기화를 완료한 각 자식 노드들은 다시 하위 레벨에 자식 노드가 있는 부모 노드일 경우 앞에서 언급한 동기화 과정을 계

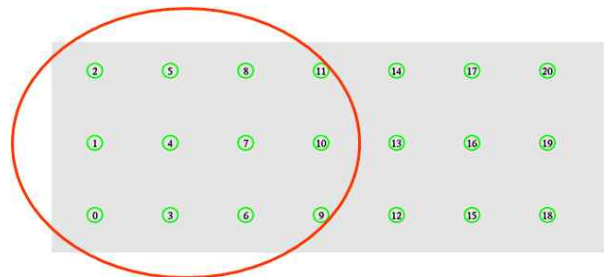


Fig. 3 Simulation Topology

4.1 Synchronization Convergence Time

PSTN에서는 각 노드가 부모 노드와의 패킷 전송을 시도할 때 이웃 노드와의 충돌을 피하기 위해 즉시 시도하지 않고 임의의 시간동안 기다린 후 전송을 시도한다. 그러므로 노드마다 다른 임의의 대기 시간에 의해 각 노드의 동기화 완료 시간이 결정된다.

시뮬레이션을 통해 그림 1의 트리를 사용하여 여러 번의 TPSN 동작을 실행하며 각 노드의 동기화 완료 시간을 측정하였다. 그림 4는 TPSN의 대기 시간을 (0~2.0 sec) 사이의 Uniform 함수에서 발생시켜 여러 번 시도한 것 중 3가지 결과만을 골라 비교한 것이다. 최대 대기 시간을 1.0 sec 이하로 실험했을 때는 충돌로 인한 전송 실패로 전체 노드가 동기화되지 않는 결과가 자주 발생하였다. 최대 대기 시간을 2.0 sec으로 했을 때에도 실제 실험에서는 6번 중 1번은 네트워크 전체 동기화에 실패하였다. 그림에서 보듯이 대기 시간이 임의로 설

정되므로 각 노드가 동기화되는 시간과 순서가 매번 실험할 때마다 다르게 나타났다. 노드 #4는 기준이 되는 루트 노드이고, 네트워크의 동기화 완료(즉, 수렴 시간)은 가장 늦게 동기화 완료되는 노드(마지막 노드)에 의해 결정되며, TPSN_1에서는 노드 #18, TPSN_2에서는 노드 #19, TPSN_3에서는 노드 #20

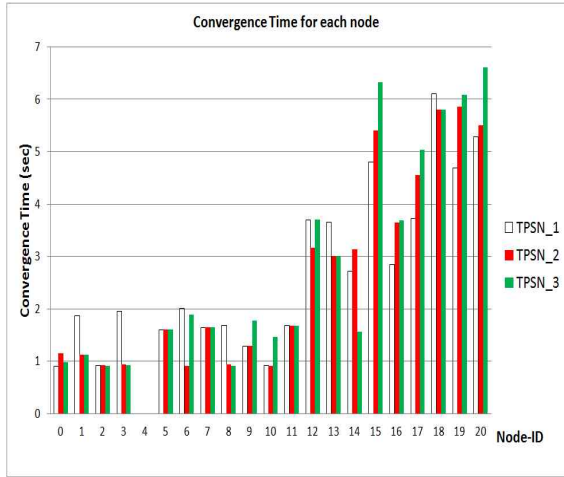


Fig. 4 Simulation results for several TPSN operations.

이 각각 마지막 노드이다.

그림 5는 여러 TPSN 실험 결과 중 동기화 완료 시간이 가장 빠른 TPSN_2와 본 논문에서 제안한 P-TPSN의 동기화 완료 시간을 비교한 것이다. P-TPSN에서는 TPSN과 달리 임의의 대기 시간을 사용하지 않으므로 결과는 항상 일정하다. 대부분의 노드에서 동기화 완료시간이 P-TPSN이 TPSN_2보다 빠름을 확인할 수 있다. P-TPSN의 동기화 완료 시간을 더 자세히 보기 위해 P-TPSN의 결과만을 다시 나타내면 그림 6과 같다.

그림 6을 자세히 살펴보자. 노드 #1, #2, #3, #5, #6, #7, #8, #9, #10, #11은 루트 노드(#4)의 자식 노드로서 순서대로 동기화 되었다. 다음 레벨 1 노드 #8의 자식 노드인 #13, #14, 노드 #9의 자식 노드인 #12, 노드 11의 자식 노드인 #16, #17이 동기화 되었다 (레벨 2 노드들의 동기화 완료). 그 다음 레벨 2인 노드 #16의 자식 노드인 노드 #15, #18, #19, #20이 차례대로 동기화되어, 마지막으로 노드 #20의 동기화가 완료됨으로써 전체 네트워크의 동기화가 완료되었다. 여기서 TPSN_2와 P-TPSN의 네트워크 동기화 완료 시간은 각각 5.8519 sec와 1.8171 sec이다.

4.2 Power Consumption

그림 7에서는 5회 실시한 TPSN과 본 논문에서 제안한 P-TPSN을 1 라운드 실행했을 때 걸리는 네트워크 동기화 완료 시간과 이 때 소요된 전력을 나타내었다. P-TPSN의 성능이 TPSN을 여러 번 실행했을 때 어떤 결과보다도 우수함을 확인할 수 있다.

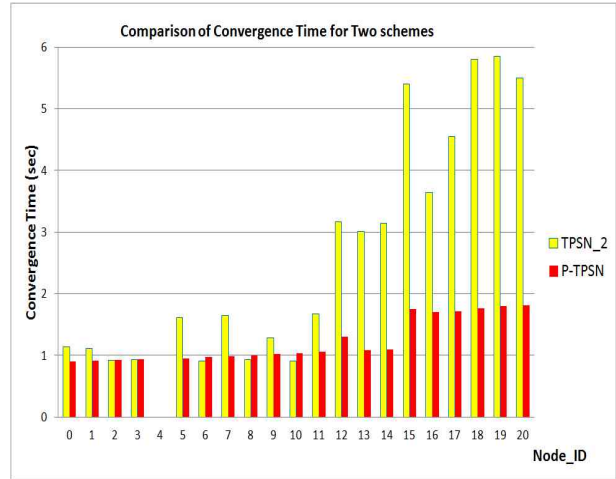


Fig. 5 Comparison of Convergence Times for Two synchronization Schemes.

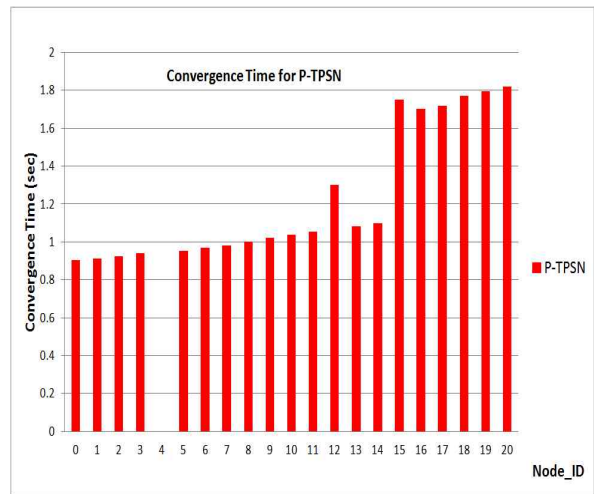


Fig.6 Convergence Time of P-TPSN.

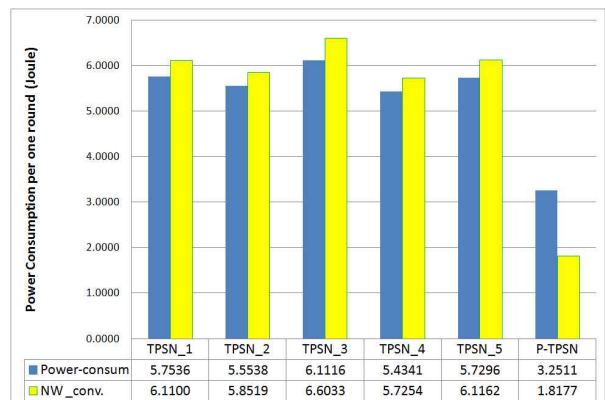


Fig. 7 Comparison of Power-consumption and Network Convergence Time for P-TPSN and TPSNs.

IV. Conclusions

TPSN은 무선 센서 네트워크에서 동작하는 대표적인 시각 동기화 프로토콜로서, 오늘날까지 많은 시각 동기화 프로토콜에서 활용되거나 참조가 되고 있다. 하지만, TPSN은 동기화 과정에서 여러 노드가 경쟁적으로 데이터 전송을 시도함으로써 충돌로 인하여 동작이 지연되거나 불필요한 전력이 소모된다. 이 논문에서는 TPSN에 경쟁방식이 아닌 폴링 방식을 도입하여 불필요한 충돌을 줄임으로써, 전력 소모를 줄일 뿐 아니라 네트워크 전체의 동기화 완료 시간을 절약하기 위한 방안을 제안한다. 시뮬레이션을 사용하여 제안된 방안의 성능을 검증하고 원래의 방안과 비교하였으며, 원래의 방안보다 더 우수한 성능 결과를 확인하였다.

REFERENCE

- [1] D. Mill, Computer Network Time Synchronization: the Network Time Protocol on Earth and in Space, Second Edition, CRC Press, 2011.
- [2] S. Bae, "A Survey on Time Synchronization Protocols for Wireless Sensor Networks", Journal of The Korea Society of Computer and Information, Vol. 19, No.6, June 2014.
- [3] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-Synch Protocol for Sensor Networks, in Proceeding of ACM Sensys, USA, 2003.
- [4] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The Flooding Time Synchronization Protocol", in Proceeding of ACM SenSys'04, 2004.
- [5] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts", in Proceeding of ACM OSDI, 2002.
- [6] S. Bae, "An EIBS Algorithm for Wireless Sensor Network with Life Time Prolongation", Journal of the Korea Society of Computer and Information, Vol. 19, No. 9, Sep. 2014.
- [7] D. Liu, Z. Zheng, Z. Yuan, and W. Li, "An Improved TPSN Algorithm for Time Synchronization in Wireless Sensor Network", in Proc. Of Int. Conf. on Distributed Computing Systems Workshops, pp. 279-284, 2012.
- [8] A. Kulaki and K. Erciyes, "Time Synchronization Algorithm based on Timing-Synch Protocol in Wireless Sensor Networks", in Proc. ISCIS(Int. Symposium on Computer and Information Science), pp.1-5, 2008.
- [9] S. Bae, "Enhanced Tree Construction Algorithm for Timing-sync Protocol for Sensor Networks", Journal of Advances in Computer Networks, Vol. 3, No.1, Mar. 2015.
- [10] "The Network Simulator ns-2: Documentation", <http://www.isi.edu/nsnam/ns/ns-documentation.html>, Nov. 2011.

Authors



Shi-Kyu Bae received the B.S. degree in Electronic Engineering, and M.S. and Ph.D. degrees in Computer Engineering from Kyungpook National University, Korea, in 1986, 1994 and 1998, respectively

Dr. Bae has worked for Samsung Electronics Co., and joined the dept. of computer engineering at Dongyang University, Korea, in 1995. He is currently serving as a professor He is interested in computer networks, mobile networks, and multimedia networks.