KIPS Tr. Software and Data Eng.
Vol.4, No.9 pp.385~392 pISSN: 2287-5905

http://dx.doi.org/10.3745/KTSDE.2015.4.9.385

A Time Synchronization Protocol for Energy-Constrained Wireless Networks

Shi-Kyu Bae†

ABSTRACT

In IoT(Internet of Things), it is important for wireless networks to communicate data created among resource-constrained wireless nodes, where time synchronization is needed for meaningful data creation and transmission. Time Synchronization by flooding is one of the mostly used protocols for WSN(Wireless Sensor Networks). Even though this type of scheme has some advantages over other types (i.e. a simple algorithm and independency of topology and so on), too many data transmission is required, leading to large power consumption. So, reducing transmission data is an important issue for energy efficiency in this kind of networks. In this paper, a new Flooding-based time synchronization protocol is proposed to use energy efficiently by reducing a transmitted traffic. The proposed scheme's performance has been evaluated and compared with an representative scheme, FTSP(Flooding Time Synchronization Protocol) by simulation. The results are shown that the proposed scheme is better than FTSP.

Keywords : IoT(Internet of Things), Time Synchronization, Flooding, Wireless Sensor Network, Energy Efficiency

에너지 제한적인 무선 네트워크에서 동작하는 시각 동기화 프로토콜

배 시 규†

요 약

사물인터넷에서 전력이나 컴퓨팅 등 자원이 한정된 무선 노드 사이에 데이터를 효율적으로 전송하는 무선 네트워크 기술이 중요하며, 의미 있는 데이터 생성과 전송을 위해서 모든 노드 사이에 시각 동기화가 필요하다. 무선 센서 네트워크에서 여러 방식의 시각 동기화 프로토콜이 제안되었는데, 플러딩에 의한 방안도 그중의 하나이다. 이 방식은 다른 방식에 비하여 알고리즘이 간단하고, 토폴로지 변화에 영향을 받지 않는 장점을 가지고 있음에도 불구하고 지나치게 많은 데이터 전송이 요구되어 결과적으로 전력 소모가 많다. 전력이 한정적인 무선 노드에서는 가능한 성능을 저하시키지 않으면서 전력 소모를 줄이는 것이 중요하므로, 이 논문에서는 플러딩에 기반한 에너지 효율적인 시각 동기화 프로토콜을 제안한다. 시뮬레이션을 통하여 가장 대표적인 프로토콜인 FTSP(Flooding Time Synchronization Protocol)와 성능을 비교하고, 에너지 효율 면에서 우수한 프로토콜임을 보인다.

키워드 : 사물인터넷, 시각 동기화 프로토콜, Flooding, 무선 센서 네트워크, 에너지 효율

1. 서 론

사물인터넷(IoT: Internet of Things)이 가능하기 위해서는 기존의 다양한 기술들이 복합적으로 사용되므로, 여러 기술 및 서비스 응용 분야에서 사물인터넷이 주목을 받고 있다. IoT의 주요 기술에는 센서기술, 유무선 통신 및 네트워크 인프라 기술, IoT 서비스 인터페이스와 같이 크게 3가지 기술로 이루어져 있다[1]. 그중 센서 기술은 임베디드, 스마트 기기, 유무선 통신 기술 등과 함께 중요한 부분을 차지하고 있다. 센서 기기들 중에는 스마트 폰과 같은 고성능의 장비도 있으나, 센서 노드나 RFID 같은 저성능의 센서 기기도 많이 사용되므로 이들 센서로부터 자료를 수집, 처리, 전송하여 인터넷과 연동시키는 무선 센서 네트워크(WSN: Wireless Sensor Network) 기술은 IoT에서 중요한 역할을 한다[2].

† 정 회 원 : 동양대학교 사이버보안학과 교수
Manuscript Received : July 29, 2015
Accepted : August 24, 2015
* Corresponding Author : Shi-Kyu Bae(skbae@dyu.ac.kr)

WSN의 연구도 여러 분야에서 다양하게 이루어지고 있으며, WSN을 구성하는 센서 노드들이 고정되어있는 형태를 고정식(static)이라고 하고, 센서 노드들이 이동할 수 있으면 동적(dynamic) 또는 이동식(mobile) 네트워크라고 한다. 일반적으로 동적 WSN은 토폴로지가 자주 변화하며 이동하는데 전력이 소모되므로 상대적으로 고정식보다 비용이 높다. 그리고 각 노드가 이동이 가능하므로 네트워크를 재배치하거나 재구성할 수 있는 장점이 있다[3]. 이 두 방식을 절충하여, 기본적으로 고정식으로 구성하되, 전체가 아닌 일부 센서 노드만 이동 기능을 부여하는 방식이 사용[4]되기도 한다. 이를 혼합형(hybrid) 혹은 반고정식이라 부르기로 한다.

무선 네트워크 분야 중에서 시각 동기화에 관한 연구가 있으며, 이것은 최근에 새로 시작된 것이 아니라 예전 인터넷이 개발된 초기부터 있어왔으나, 무선 네트워크 환경, 특히 무선 센서 네트워크 분야에서는 새로운 요구사항과 고려사항이 필요하여 관심을 끌게 되었다. 대표적으로 인터넷과 같은 유선 네트워크에서 동작하는 NTP(Network Time Protocol)[5]가 일찍이 개발되었다. WSN에서의 각 센서 노드는 서로 협업을 하기 위하여 공통 클럭을 유지하는 것이 중요하며, 이를 이용하는 응용에서는 네트워크 전체 혹은 각 센서 노드 간에 시각 동기화가 유지되는 것을 전제로 한다.

지금까지 개발된 여러 시각 동기화 프로토콜들은 송신자와 수신자 간의 메시지를 주고받는 형태에 따라 송신-수신형과 수신-수신형으로 나뉜다. 송신-수신형 동기화는 한 노드가 송신자가 되어 데이터를 전송하며, 송신자를 제외한 다른 노드들은 수신자가 되어 보내진 데이터를 수신하며, TPSN(Timing-sync Protocol for Sensor Networks)[6]과 FTSP(Flooding Time Synchronization Protocol)[7]이 여기에 속하고, 글로벌 참조 클럭을 전체 네트워크에 전달할 수 있다. 이와 대조적으로 수신-수신형 동기화에서는 참조 노드가 특정한 패킷을 한 번 전송하면, 이후에는 전송한 노드를 제외한 다른 노드들이 서로 협력하여 동기화를 이루는 방식을 갖는다. 이 종류에 해당하는 대표적인 프로토콜이 RBS(Reference Broadcast Synchronization)[8]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 상호 간의 클럭 오프셋을 기록한다. 이 RBS는 가까운 거리에 위치한 여러 노드 간에 상대적인 클럭 오프셋을 비교적 정확하게 파악할 수 있지만, 특정 노드의 클럭(혹은 글로벌 클럭)으로 네트워크 전체를 동기화할 수는 없다.

이들 대표적인 시각 동기화 프로토콜 중 FTSP는 루트 노드로부터 시작하여 모든 노드에 확산하는 방식으로 동기화를 한다. 즉, 별도의 계층 구조를 만들지는 않지만 루트 노드로부터의 거리에 따라 계층 구조가 형성되어 중심에 있는 노드로부터 통신 영역 내에 있는 모든 노드에게 시간 정보의 전송이 이루어지며, 루트 노드로부터 시작하여 단계적으로 모든 노드에게 전송이 이루어진다. 전달되는 순서는 미리 정해진 구조가 없으므로 중심 노드로부터의 거리에 따라 결정이 된다. 플러딩으로 인하여 지나치게 많은 메시지가 전송되는 것을 방지하기 위하여 모든 노드는 반드시 한 번만 전송을 하도록 하고 있다[9]. FTSP는 다른 방식과 비교할 때 별도의 계층구조가 필요 없고 알고리즘이 간단한 장점이 있으나, 네트워크 전체의 동기화에 지나치게 많은 데이터 전송이 필요하다. 이는 전력 소모가 빨리 일어나 네트워크의 수명에도 영향을 끼친다.

이 논문에서는 플러딩 방식으로 시각 동기화를 수행하는 프로토콜에서 전송 메시지 수를 줄여 전력 소모를 줄이고 네트워크 전체의 수명을 늘리는 WSN용 시각 동기화 프로토콜을 제안한다. 논문의 구성은 다음과 같다. 2절에서 플러딩을 사용하는 FTSP의 동작을 살펴본다. FTSP의 문제점을 찾아 개선된 알고리즘을 3절에서 제안한다. 4절에서 제안한 알고리즘을 적용하여 성능을 분석하며, 마지막으로 5절에서 결론을 다룬다.

2. 관련 연구

2.1 플러딩 기반 시각 동기화 프로토콜

플러딩을 사용하는 시각 동기화 프로토콜은 루트 노드로부터 전체 노드를 동기화하는 글로벌 동기화 형태(예, FTSP)와 네트워크의 일부 노드들끼리 동기화를 하는 분산형 형태가 있다. 여기서는 글로벌 동기화에 중점을 둔다.

시각 동기화에 플러딩을 사용하는 것은 다른 형태의 응용(데이터 수집, 라우팅 프로토콜)과는 차이점을 가지고 있다. 그 차이점을 정리하면 다음과 같다.

① 데이터 수집 응용은 여러 노드로부터 데이터를 수집하여 하나의 노드(싱크 노드)로 전달되며, 라우팅 프로토콜은 특정한 두 노드(근원지 노드와 목적지 노드) 사이에 데이터 전송이 목적이다. 하지만, 글로벌 시각 동기화에서는 전송 노드가 루트 노드 한곳으로 고정이 되며 수신 노드는 루트 노드를 제외한 전체 노드이다. 루트 노드가 변경되는 경우가 프로토콜에 따라 발생하기는 하지만, 일정 시간 동안은 하나의 루트 노드를 중심으로 나머지 전체 노드와의 전송 형태가 된다. 이런 측면에서는 데이터 분배 응용과 공통점이 있다.

② 다른 응용에서는 근원지 노드에서 보내진 데이터가 목적지 노드까지 그대로 전달되거나 중간 전달 노드에서 변형된 데이터가 근원지 노드에 전달된다. 하지만, 글로벌 시각 동기화에서는 루트 노드로부터 전송되는 데이터가 전송 과정에서 발생하는 지연 시간과 관련이 있으므로 전송노드와 수신 노드 사이에 밀접한 관련이 있다.

③ 다른 응용에서는 데이터 전송 과정에서 신뢰성이 반드시 중요하지만, 무선 네트워크에서의 글로벌 시각 동기화는 어느 정도의 데이터 손실이나 지연이 허용되기도 한다.

④ 다른 응용에서는 이벤트에 따라 데이터 전송이 1회 혹은 가끔 발생하지만, 무선 네트워크에서의 글로벌 시각 동기화에서는 정기적으로 또는 일정 조건에 따라 동일한 동작이 반복된다.

그러므로 기존의 데이터 수집 또는 라우팅에서 제안된 플

러딩 방법과는 다른 시각 동기화에 적합한 플러딩 방안이 필요하다. [10]에서는 두 노드의 링크 신뢰도를 고려하여 사이에 있는 여러 노드 중 하나를 중계 노드로 선택하는 방법을 제안하였으나 지나치게 처리하는 정보량이 많고 동작이 복잡하여, 무선 노드에게는 과도한 부담이며, 시각 동기화에서는 어느 정도의 전송 오류가 허용이 되므로 적합하지 않다. [11]에서는 센서 네트워크의 수명을 늘리기 위하여 클러스터를 구성하고 라우팅 하는 방안을 제안하였다. [12]의 저자들은 하위 계층에서 이웃 노드들의 정보와 수신 신호를 수집하여 근원지 노드에서 목적지 노드까지 중간 전달 노드의 재전송을 줄여 에너지를 절약하는 알고리즘을 제안하였으나, 이웃 노드 간의 주기적인 데이터 교환과 이웃 노드 리스트 관리에 많은 부하가 소요된다. [13]의 저자는 FTSP를 기반으로 루트 노드를 제외한 각 노드는 새로 전송을 계속하기 전에 자신이 전달할 수 있는 영역 내의 이웃 노드들의 위치를 계산하고 다른 노드로부터 데이터의 전달이 이미 이루어졌는지 여부를 확인하여 선택적으로 재전송하는 알고리즘을 제안하였다. 하지만, 전체 노드의 위치 정보를 알고 있어야 하며, 이웃 노드의 전송 데이터 내용을 계속적으로 모니터링 하는 부하가 지나치게 크다. 또한 다른 이웃 노드에서 전송한 데이터 내용만으로 자신의 전송 영역이 완전히 커버되었는지 확신할 수 없다.

2.2 FTSP의 동작

FTSP의 동작을 설명하기 위하여 Fig. 1에 예를 들어 나타내었다. n1, n2…는 노드의 ID이고, n1을 루트 노드로 여러 노드가 그림의 위치에 배치되어있다고 하자. 루트 노드(n1)에서 처음으로 동기화 정보를 방송하면 방송 영역 내에 있는 n2, n3, n4, n5, n7 노드에 신호가 전달되며, 루트 노드로부터의 거리에 따라 도착 시간이 약간 차이가 발생한다. 여기서는 n5가 가장 가까운 거리에 있다고 가정하면 가장 먼저 루트 노드로부터 데이터를 수신한다. 다음 이들 노드들이 각각 다시 동기화 정보를 방송하게 되는데, n5가 가장 먼저 전송을 시작하여 데이터가 n6, n8에 도착한다. n2와 n3가 다시 전송한 데이터는 루트 노드에만 도착하지만 루트 노드는 이를 무시한다. n4가 전송한 데이터는 n1, n5, n8에 도착하지만, n1는 상위 레벨이고, n5는 n4와 같은 레벨이므로 역시 무시한다. 그리고 n8은 n4로부터 데이터를 받기는 했지만 이미 n5로부터 데이터를 받아서 한 단계 낮은 레벨로 설정하였기 때문에 추가로 받은 데이터를 무시한다.

이렇게 모든 노드에서 한 차례씩의 방송을 실시하고 나면 모든 노드는 상위 레벨 노드로부터 동기화 정보를 수신하였고 네트워크 동기화가 완료된다. 이 과정을 동기화 라운드라고 부르기로 한다. 즉, FTSP에서는 모든 노드가 1회씩 방송 메시지를 전송하므로 네트워크 전체의 동기화에는 전체 노드 수만큼의 전송이 필요하다.

FTSP에서는 이 라운드를 연속적으로 몇 차례(예, 8회) 실시하여 시각 정보를 추정한다. 그리고 이러한 몇 차례의 연속적인 라운드를 일정 기간마다 반복하여 동기화 오류가 일정 범위 내로 유지되도록 한다.

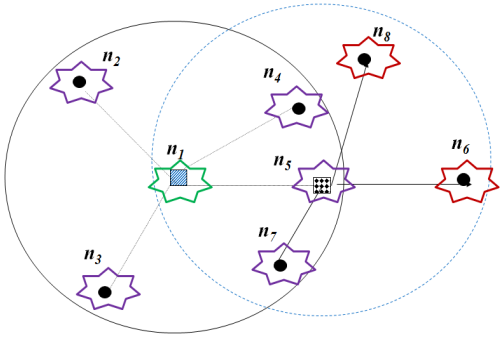

Fig. 1. Operation of FTSP

3. 제안 알고리즘

FTSP에서는 루트 노드로부터 이웃 노드로 플러딩 방식으로 데이터를 전달하여 네트워크의 모든 노드로 데이터를 전달하는 과정을 반복적으로 수행한다. 매 라운드마다 데이터를 전달하는 노드들 간의 관계는 별도로 인식하지 않으며 매 라운드마다 노드의 위치에 따라 달라질 수 있다. 다시 말하면 네트워크의 토폴로지가 변화하더라도 FTSP의 동작은 구분 없이 동작할 수 있으며 이 점이 다른 동기화 방식과 비교할 때 장점으로 부각되기도 한다. 하지만, 이런 동작은 에너지 효율 면에서는 반대로 단점이 된다. 왜냐하면 매 라운드마다 모든 노드는 최소한 한 번 이상의 데이터 송신을 하므로 노드의 수가 증가할수록, 그리고 라운드가 자주 이루어질수록 데이터 전송 횟수가 증가하고 이것은 과도한 전력 소비로 나타난다.

본 논문에서는 대부분 토폴로지의 변화가 거의 없으나 가끔 토폴로지의 변화가 발생하는 환경을 고려하여 전력 소모를 줄이는 시각 동기화 프로토콜을 제안한다. 본 논문에서 제안하는 동기화 방안은 기본적으로 FTSP에서처럼 루트 노드로부터 이웃 노드로 플러딩 방식으로 데이터를 전달하여 네트워크의 모든 노드로 데이터를 전달하는 방식을 사용한다. 그러나 이런 동작은 매 라운드마다 모든 노드가 최소한 한 번 이상의 데이터 송신을 하므로 노드의 수가 증가할수록, 그리고 라운드가 자주 이루어질수록 데이터 전송 횟수가 증가하고 이것은 과도한 전력 소비로 나타난다.

본 논문에서는 초기에 또는 토폴로지의 변화가 있을 때는 FTSP와 동일하게 동작하나, 토폴로지의 변화가 없는 안정적인 네트워크에서는 일부 노드들의 데이터 전송을 생략하도록 하여 메시지 전송 횟수를 줄여 에너지 소비를 줄이는 방법을 제안하고, 이 방식은 e-FTSP(Energy-efficient FTSP)라고 한다.

우선 e-FTSP는 두 개의 동작 모드로 이루어진다. 즉, 초기화(Initialize)(또는 트리 구성) 모드와 동기화(Synchronization) 모드로 나뉜다((Fig. 2) 참조). 트리 구성 모드는 초기에 e-FTSP가 시작할 때나 토폴로지가 안정적으로 유지되다가 변화가 감지되어 새로 토폴로지를 파악할 필요가 있을 때 동작한

다. 동기화 모드는 초기에 불안정 모드가 동작한 후 토폴로지의 변화가 없는 동안에 동작한다.

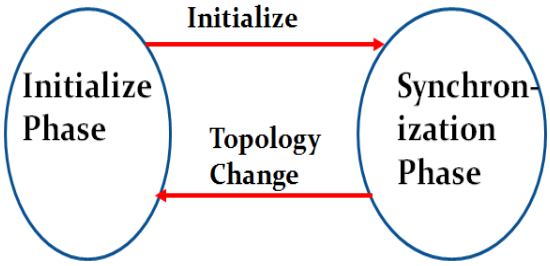

Fig. 2. Operation Phases of e-FTSP

3.1 e-FTSP 트리 구성 모드

트리 구성 모드는 초기에 e-FTSP가 시작할 때나 토폴로지가 안정적으로 유지되다가 변화가 감지되어 새로 토폴로지를 파악할 필요가 있을 때 동작한다. 기본적인 동작은 FTSP와 유사하나 노드 간의 부모-자식 관련 정보를 생성하여 보관하는 기능이 추가된다. 다시 말하면, FTSP는 데이터 전송과 함께 루트 노드로부터의 계층 레벨만을 인지할 뿐 노드 간의 관계는 고려하지 않는다. 하지만, e-FTSP에서는 이 과정에서 부모-자식 노드 간의 관련 정보를 생성하고 보관하여 동기화 모드에서 활용한다.

트리 구성 모드에서 트리를 생성하는 과정을 자세히 알아보면 다음과 같다. 트리 생성은 루트 노드를 중심으로 시작되며, 루트 노드는 트리 레벨을 0으로 설정한 후 {NodeID : level# : packet_type(Init) : parent NodeID}로 이루어진 "Init" 패킷을 방송함으로써 트리 구성 모드의 동작을 시작한다. 여기서 트리 레벨은 루트 노드로부터 형성되는 트리의 깊이를 나타내며 0에서 시작하여 홉이 증가할수록 증가되는 값으로 표시된다. 루트 노드는 부모 노드가 없는 유일한 노드로 부모 노드 ID를 '-1'로 설정한다. 루트 노드를 제외한 다른 모든 노드는 Init 패킷을 받으면 송신한 노드를 부모 노드로 설정하고 레벨을 1 증가한 값으로 설정한 후, 다시 다음 단계의 노드에게 Init 패킷을 방송한다. 이때 모든 노드는 최초로 패킷을 수신한 때에 부모 노드로부터 레벨을 설정하며 이후에 추가로 수신하는 패킷에 대해서는 더 이상 방송을 하지 않는다. 즉, 모든 노드는 단 한 번의 방송만을 시도한다. 부모 노드를 결정한 후에 계속 수신되는 Init 패킷으로부터 추가로 후견자(Guardian) 노드 정보를 설정하며, 부모 노드와 동등하거나 더 높은 레벨의 노드만이 후견자 노드가 될 수 있다. 후견자 노드는 토폴로지의 변화가 발생할 때 부모 노드를 대신할 노드들이며 자세한 내용은 동기화 모드에서 설명한다.

각 노드가 자식 노드의 정보를 가지기 위해서 자식 노드로부터의 확인 패킷이 필요하다. 여기서는 전송 패킷 수를 최소화하기 위해 추가의 전송 과정 없이 자식 노드가 다음 하위 레벨의 자식 노드를 설정하기 위해 Init 패킷을 방송할 때 이것을 암묵적 확인(Implicit Acknowledgement)으로 활용한다. 즉, 자식 노드가 Init 패킷을 방송할 때 자신이 설정했던 부모 노드 ID를 패킷에 포함시켜 방송하면 부모 패킷이 이를 수신하면서 자식 노드들을 인식하도록 한다. 이렇게 1차로 루트 노드에서 네트워크의 모든 노드에게 시각 정보를 전달하면서 동시에 각 노드 간의 관계(즉 트리)가 형성이 되고 각 노드는 트리에서 자신의 부모 및 자식 노드의 정보를 파악하게 된다.

이때 일부 노드는 자식 노드 정보가 없는 노드가 되며, 이것을 단말(또는 leaf 노드)이라고 한다. Fig. 3은 생성된 트리의 예를 보인 것으로, 노드를 부모 노드(루트 노드 포함)와 단말 노드의 두 종류로 크게 나누고 각각 노란색과 붉은색으로 표현하였다. 여기서 단말 노드는 일단 자신에게 자식 노드가 없다는 것을 파악하고 나면 다음 라운드부터는 시각 정보를 전달하기 위해 데이터 전송을 할 필요가 없다. 그러므로 트리 생성 모드가 종료되고 동기화 모드로 들어가면 다음 라운드부터 단말 노드들은 동기화를 위해 데이터 전송에 참여하지 않는다. 자세한 내용은 동기화 모드에서 설명한다.

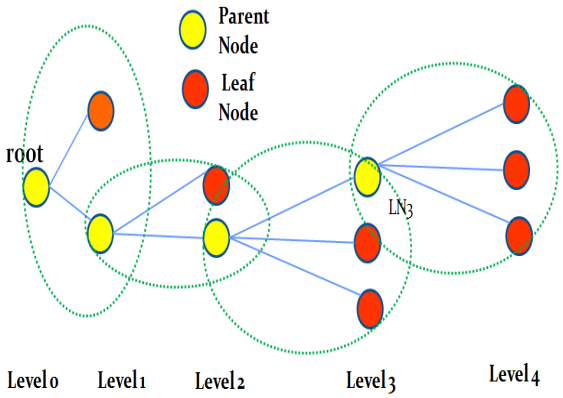

Fig. 3. Example Tree Created at the Syncronization Phase of e-FTSP

3.2 e-FTSP 동기화 모드

동기화 모드에서 한 라운드가 종료되면 전체 노드 중 일부 노드는 자신이 단말 노드라는 것을 인식하게 되어, 다음 라운드부터 시각 정보를 전달하기 위한 데이터 전송에 참여하지 않더라도 네트워크 전체 동기화에 전혀 이상이 없다. 결과적으로 전송 메시지 수가 줄어들어 전력 소모를 감소시키는 효과가 발생한다.

Fig. 3의 트리의 예를 들어보자. FTSP에서는 한 라운드의 동기화를 위해 11번의 데이터 전송이 필요하다. 반면에, e-FTSP에서는 11개 노드 중 7개의 단말 노드를 제외한 4개의 부모 노드에게만 메시지를 전송하게 되어 FTSP보다 7/11의 에너지를 절약하게 된다.

WSN의 구성이 고정적이어서 토폴로지의 변화가 없다면 대부분의 라운드에서 이 정도의 에너지 효율이 높아질 것이다. 하지만 노드가 이동하거나, 일부 노드에 이상이 발생하

여 토폴로지의 변화가 생긴다면 이전에 사용하던 토폴로지 정보는 소용이 없게 된다. 여기서 토폴로지 변화에 따라 동작 모드를 적절하게 조정할 필요가 있다.

몇 가지 시나리오를 살펴보면 다음과 같다.

① 토폴로지가 자주 변화하는 동적인 네트워크 구성인 경우에는 트리 구성 모드만으로 동작시킨다.

② 대부분 또는 상당 기간 동안 토폴로지의 변화가 없으며 가끔 노드의 이동이나 일부 노드의 에너지 상태에 따라 토폴로지가 변할 수 있다면, 동기화 모드에서 토폴로지 변화 감지 기능을 추가로 실행하여 토폴로지 변화 여부에 따라 동기화 모드를 계속할지 또는 트리 구성 모드로 상태 변이를 할지 결정한다. 토폴로지 변화 감지 기능은 다음에 설명한다.

③ 토폴로지 변화 감지 기능으로 인한 오버헤드가 지나치고, FTSP처럼 선형 추정(linear regression)을 위해 몇 차례의 연속적인 라운드가 필요하면 연속적인 라운드 동안 한 차례의 트리 구성 모드와 몇 차례의 동기화 모드를 실행한다. Fig. 4는 FTSP에서의 동기화 기간을 나타낸 것으로, 일정 주기(예, 30초)마다 동기화를 다시 수행하며 한 동기화 구간 동안 시각 정보 수집과 추정을 위해 여러 번의 동기화를 짧은 시간 내에 연속적으로(예, 8회) 실시한다. 이러한 경우, 8차례의 연속적인 동기화 라운드에서 트리 구성 모드 1회+동기화 모드 7회 단위로 동작시킨다.

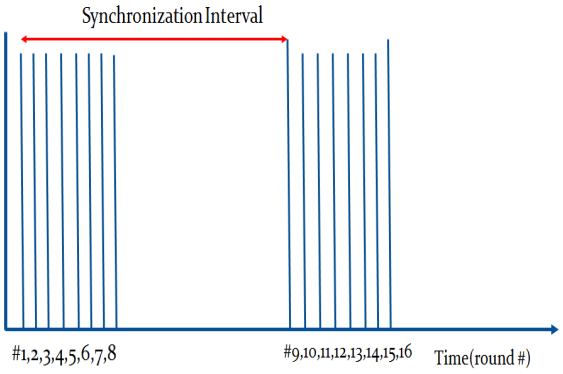

Fig. 4. Synchronization Interval of FTSP

3.3 토폴로지 변화 감지

e-FTSP가 동기화 모드에서 동기화를 실시하다가 토폴로지 변화를 감지하면 트리 구성 모드로 상태를 천이하는데, 토폴로지 변화를 어떻게 감지할 수 있을까? 각 노드가 네트워크 전체 혹은 이웃 노드들의 정보 없이 토폴로지의 변화를 감지하기는 어렵다. 하지만 간접적으로 각 노드에게 일정 시간(동기화 주기+α) 이상의 시간이 지났음에도 동기화 정보가 전달되지 않는다면 동기화에 문제가 있음을 알 수 있다. 이 문제의 원인은 부모 노드로부터 전송된 데이터가 충돌이나 손실 등으로 정확히 전달되지 않았거나, 노드의 이동, 노드의 사망 등의 토폴로지 변화에 기인한 것일 수 있다.

① 데이터의 손실이 원인이라면 다음 동기화 주기 내에 다음 데이터가 도착할 가능성이 높다. 동기화 주기가 크다면 다음 동기화 주기까지 기다리지 않고, 부모 노드에게 데이터 전송을 재요청하여 확인할 수 있다.

② 부모 노드에게 재전송을 요청했는데 응답이 없다면 부모 노드와의 링크에 문제가 발생했다는 것을 알 수 있다. 이 현상을 토폴로지 변화로 판단하되 전체 토폴로지의 재구성으로 진행하기 전에 한 가지 더 가능성을 타진한다.

③ 부모 노드와의 링크에 문제가 발생하면 트리의 형태가 변경된다. 하지만, 부모 노드가 아닌 부모를 대신할 노드(후견자 노드)가 존재한다면 부모 노드를 대체하여 두 노드 사이의 관계만 변경하여 나머지 대부분의 트리는 그대로 유지하여도 가능하다. 후견자 노드 정보는 트리 구성 모드에서 Init 패킷을 사용하여 트리 정보를 형성할 때 부모 노드 정보뿐 아니라 다른 상위 레벨의 노드로부터 데이터를 수신할 때 기록해둔다.
- 부모 노드로부터 문제가 발생하면 후견자 노드로부터 수신한 기록을 살펴보고 부모 노드를 임시로 후견자 노드로 대체한다.
- 부모 노드뿐 아니라 후견자 노드로부터 정상적인 데이터를 수신하지 못하면 후견자 노드와 차례로 접속을 시도하여 성공하면 새 부모 노드로 설정한다.
- 부모 노드뿐 아니라 후견자 노드 모두와도 연결이 실패하면 최종적으로 토폴로지 변화로 결정하며, 이웃 노드로 토폴로지 재구성을 요청하는 패킷을 방송하여 루트 노드까지 전달하도록 한다.

④ 토폴로지 재구성 요청을 받은 루트 노드는 Init 패킷을 방송하여 트리 구성 모드를 실행한다.

Fig. 5는 제안 프로토콜의 알고리즘을 유사 코드로 나타낸 것이다.

4. 시뮬레이션

알고리즘의 동작을 검증하기 위하여 Fig. 6과 같은 노드를 배치하여 토폴로지를 설정하고 NS2(Network Simulator)-2.35[14]을 사용하여 실험하였다. 공통되는 시뮬레이션 환경은 노드 간 거리 100m, 전송 영역 반경 550m이고, 노드에서의 전력 변수는 송신전력(0.660W), 수신전력(0.395W), 대기전력(0.035W)으로 설정하였다. 사용된 MAC은 802.11, 그리고 라우팅 프로토콜은 DSDV이다.

4.1 잔여 전력

Fig. 6의 배치에서 FTSP와 동일한 방식이지만 e-FTSP의 Init 모드를 실행하면 Fig. 7과 같은 트리가 생성되고, 각 노드에서는 부모 노드, 후견자 노드, 자식 노드 등의 정보가 Table 1에 나타난 것처럼 인지가 된다. 여기서 #1 노드가 루트 노드이고 부모 노드는 #1, #5, #7, #10, #11, 나머지 노

드는 단말 노드이다. 각 부모 노드는 한 개 이상의 자식 노드를 가지고 있고, #9~#17 노드는 부모 노드 외에 하나 이상의 후견자 노드를 가지고 있다.

```
if (root node)
Broadcast a Init or Sync packet according to the mode (Init or Synchronization)
For all nodes except root node {
If (packet received && firstly received) {
  if (Init packet) Set level=parent node's level+1;
  If( Init packet ||
       (Synch packet && Parent node))
     Broadcast Synch packet
}
If (packet received from child)
     update children_list
}
```

Fig. 5. Pseudo Code for Algorithm of e-FTSP

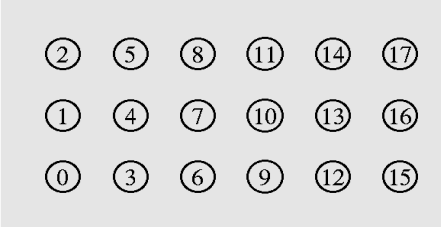

Fig. 6. Simulation Topology

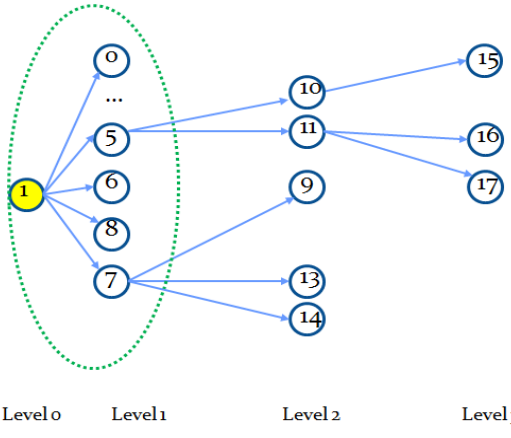

Fig. 7. Tree Created at Simulation Topology of Fig. 6.

여기서 FTSP와 e-FTSP를 8라운드 실행하고 각 노드의 잔류 에너지를 비교해보았다(Fig. 8 참조).

초기 에너지 2 Joule을 전 노드에 동일하게 설정하고 실험한 결과, FTSP의 전력 소비가 전체 노드에서 e-FTSP보다 훨씬 많으며 모든 노드가 전송에 참여하므로 각 노드별 전력 소모가 비슷함을 알 수 있다. e-FTSP의 노드별로 에너지 분포를 살펴보면, 루트 노드 (#1), #1, #5, #7, #9, #10는 부모 노드이어서 다른 단말 노드보다 상대적으로 전력 소비가 더 많게 나타났다.

Table 1. Relationship Among Nodes Created at Simulation Topology of Fig. 6

node#	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Parent	1	1	1	1	1	1	1	1	1	7	5	5	7	7	7	10	11	11
Child	0	0,2,3,4,5,6,7,8	0	0	0	10,11	0	9,12,13,14	0	0	15	16,17	0	0	0	0	0	0
Guardian	0	0	0	0	0	0	0	0	0	3,4,6,8	3,4,6,7,8	4,6,7,8	6	6,8	8	9,12,13,14	9,10,12,13,14	10,12,13,1

4.2 전송 데이터 크기 및 전력 소모

FTSP의 동작과 동일하게 비교하기 위하여, FTSP와 e-FTSP를 8라운드를 실시한 후 전송한 메시지의 수를 측정하였다. 각 패킷의 크기는 20바이트로 동일하게 하였으므로 데이터 량은 20*메시지 수*라운드로 산출할 수 있다. 실험은 동일한 영역 내에 노드 수를 증가시켜 노드 밀도를 변경하면서 8라운드씩 실행하고 전송 메시지 수를 측정하였다. Fig. 9에서 나타나듯이, FTSP에서는 노드의 수가 증가할수록(즉, 노드 밀도가 증가할수록) 전송 메시지 수는 비례하여 증가한다. 이는 FTSP에서 모든 노드에서 반드시 한 번씩 전송을 하기 때문이다. 대조적으로 e-FTSP에서는 노드의 수가 증가할수록 전송 메시지 수가 증가하지만 비례하지 않고 기울기가 상대적으로 완만하다. 실험에서는 노드 수가 72(그림에서 #n=72로 표시)인 경우에 노드 수가 54인 경우보다 오히려 전송 메시지 수가 줄어든 것으로 나타났다. 이는 전송 노드 수가 늘어나더라도 토폴로지에 따라(트리 정보에서 단말 노드의 수가 증가한다면) 전송에 참여하는 노드의 수가 줄어들므로 전송 메시지 수가 감소할 수 있음을 보여준다. 결론적으로 e-FTSP가 FTSP보다 전송 메시지 수가 상대적으로 적으며, 특히 노드 밀도가 높을 때 효과적임을 알 수 있다.

전송한 데이터의 양이 적으면 소비 전력이 낮을 것으로 예상할 수 있으며[15], 실제로 전력 소비를 측정해보았다. Fig. 10에서 확인할 수 있듯이, e-FTSP의 전력 소비가 FTSP보다 낮은 것을 알 수 있으며, 두 방안에의 격차는 노드의 밀도가 증가할수록 더욱 커졌다.

동기화는 1회로 끝나는 것이 아니라, 일정한 시간 간격마다 혹은 특정한 조건이 감지되면 여러 번 반복된다. Fig. 4에서 설명한 것처럼, FTSP는 한 번에 짧은 시간 내에 8회의 동기화 정보를 전송하고, 일정한 시간 간격 후(동기화 주기) 다시 재동기를 반복한다. 동일한 조건에서 FTSP와 e-FTSP를 더 오랜 시간 동안 실행하며 전력 소모를 비교해보았다. Fig. 11에 나타난 것처럼, 동기화 라운드가 계속되면 당연히 전력 소모가 증가하며, e-FTSP가 FTSP보다 라운드가 계속될수록 전력 소모가 더 줄어 에너지 효율적임

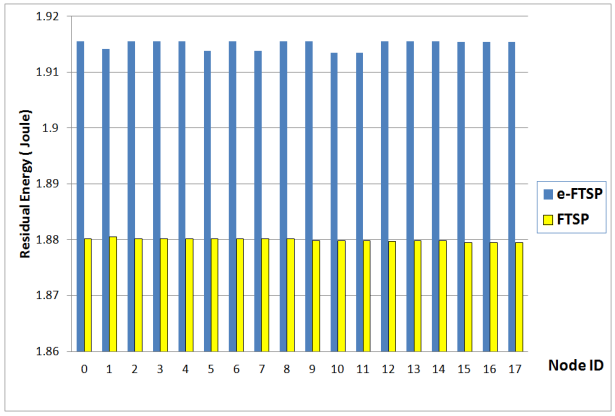

Fig. 8. Residual Energy at Each Node After Performing 8 Rounds

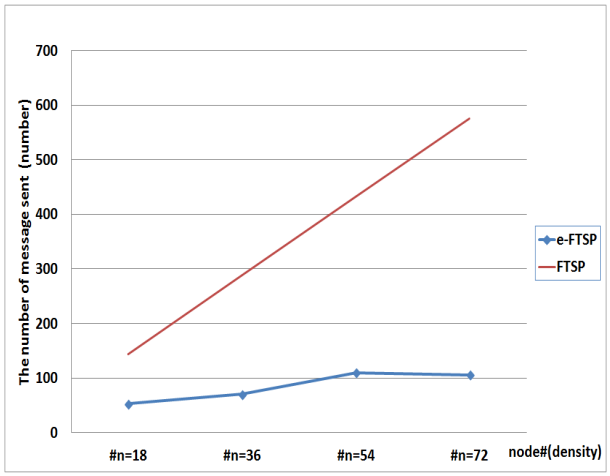

Fig. 9. The Number of Messages Sent for 8 Rounds

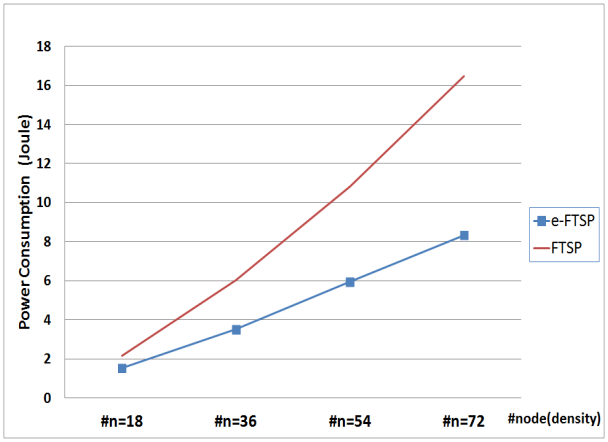

Fig. 10. Power Consumption Comparison for Two Schemes During 8 Rounds

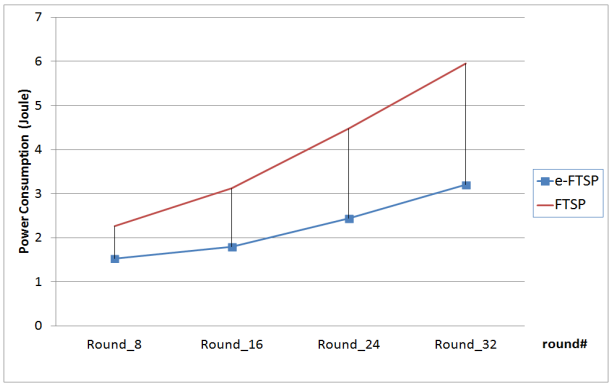

Fig. 11. Comparison of Power Consumptions of Two Schemes for Several Resynchronization

을 확인할 수 있다. 이 결과로 볼 때, 네트워크 토폴로지가 고정적이거나, 짧은 8회의 동기화 정보가 전송되는 동안에 토폴로지의 변화가 없는 네트워크에서는 e-FTSP가 FTSP보다 더 효율적임을 알 수 있다.

또한 e-FTSP의 트리 구성 모드의 동작은 전력 소비 면에서 FTSP와 동일하므로 토폴로지의 변화가 짧은 시간 내에서 어떠한 심한 조건에서도 FTSP보다 동등하거나 더 효율적이다.

5. 결 론

사물인터넷에서 전력이나 컴퓨팅 등의 자원이 한정된 노드 사이에 데이터를 전송하는 무선 네트워크에서는 시각 동기화가 중요하다. 무선 센서 네트워크용으로 이러한 시각 동기화 프로토콜이 많이 제안되었는데, 대표적인 방식이 플러딩에 의한 것이다. 이 방식은 다른 방식에 비하여 알고리즘이 간단하고, 토폴로지 변화에 영향을 받지 않는 장점을 가지고 있음에도 불구하고 지나치게 많은 데이터 전송이 요구되어 결과적으로 전력 소모가 많다. 전력이 한정적인 무선 노드에서는 전송 데이터(즉, 전력)를 줄이는 것이 중요하므로, 이 논문에서는 플러딩에 기반한 에너지 효율적인 시각 동기화 프로토콜을 제안하였다. 시뮬레이션을 통하여 가장 대표적인 프로토콜인 FTSP와 성능을 비교하고, 에너지 효율이 우수함을 보였다.

References

[1] H. Yang, S. Lee, K. Lee, and H. Bang, “A Trend of Sensor Technology for IoT,” *Korea Information Processing Society Review*, Vol.21, No.2, 2014.

[2] J. Jeon, “A study on the Techniques Trends and Prospects for Internet of Things,” *Journal of Korean Convergence Security Association*, Vol.14, No.7, pp.65-73, Dec., 2014

[3] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer Networks*, Vol.52 No.12, 2008.

[4] G. Anastasi, M. Conti, M. Francesco, and A. Passarella, “Energy Conservation in wireless sensor networks: A survey,” *Ad Hoc Networks*, Vol.7 No.3, 2009.

[5] D. Mill, “Computer Network Time Synchronization: the

Network Time Protocol on Earth and in Space," 2nd Ed., CRC Press, 2011.

[6] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-Synch Protocol for Sensor Networks," *Proceeding of ACM Sensys*, USA, 2003.

[7] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The Flooding Time Synchronization Protocol," *Proceeding of ACM SenSys'04*, 2004.

[8] J Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *Proceeding of ACM OSDI*, 2002.

[9] 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, Jun., 2014.

[10] S. Byeon, H. Seo, and J. Kim, "Contribution-Level-Based Opportunistic Flooding for Wireless Multihop Networks," *Journal of KIISE*, Vol.42, No.6, Jun., 2015.

[11] H. Choi and S. Kang, "A Method for Constructing Multi-Hop Routing Tree among Cluster Heads in Wireless Sensor Networks," *Journal of Korean Institute of Communications and Information Sciences*, Vol.39B, No.11, 2014.

[12] S. Choi and S. Yoo, "An Efficient Flodding Algorithm with Adaptive Retransmission Node Selection for Wireless Sensor Networks," *Journal of Korean Institute of Communications and Information Sciences*, Vol.32, No.11, 2007.

[13] Y. Wang, "Energy-balanced Time Synchronization Algorithm Based on Flooding for Multi-hop Wireless Sensor Networks," *International Journal of u-and e-Services, Science and Tecknology*, Vol.8, No.1, 2015.

[14] "The Network Simulator ns-2: Documentation," [Internet] http://www.isi.edu/nsnam/ns/ns-documentation.html, Nov., 2011.

[15] S. Bae, "Power Consumption Analysis of Prominent Time Synchronization Protocols for Wireless Sensor Networks," *JIPS(Journal of Information Processing Systems)*, Vol.10 No.2, pp.300-313, Jun., 2014

배 시 규

e-mail : skbae@dyu.ac.kr

1993년 경북대학교 컴퓨터공학과(석사)

1998년 경북대학교 컴퓨터공학과(박사)

1995년~현 재 동양대학교 사이버보안학과 교수

관심분야 : Computer Networks, Mobile Networks, Internet Applications, IoT