

# 센서 네트워크를 위한 효율적인 애드-혹 라우팅 알고리즘 설계

## The Efficient Ad-Hoc Routing Algorithm Design for Sensor Network

\*이 민 구 \*\*이 상 학  
Lee, Min Goo Lee, Sang Hak

**Abstract** - The non-ideal characteristics of wireless communication are found in sensor network. And sensor network must also address new raised issues. The efficient ad-hoc routing algorithm is considered the nice solution for new raised sensor network problems. To design this efficient ad-hoc routing algorithm, we study and evaluate new components in routing algorithm. Namely, new components are Link estimator, Neighbor table and Parent selection. We have tested this related experiment using the TIP-30C. TIP-30C is sensor network node that is designed by KETI(Korea Electronic Technology Institute).

**Key Words** : Sensor Network, TIP-30C, Ad-Hoc, Link Estimator

### 1. 장 서 론

미래 컴퓨팅 세계에서 핵심 기술로 인정받고 있는 새로운 테크놀로지인 센서 네트워크(Sensor Network)의 의미를 정리해 보면, 단어의 의미 그대로 기존의 센서(Sensor) 기능에 무선 통신 기능을 추가 부여한 새로운 분야의 기술이라 할 수 있다.

이와 같은 센서 네트워크 분야에서는 현재 TinyOS가 적합한 OS로 각광받고 있으며, 전 세계 300여개 이상의 센서 네트워크 연구 그룹들에 의해서 채택되어 사용되고 있다. 특히, 국내에서도 KETI를 비롯한 연구기관들과 대학교 연구실들을 중심으로 TinyOS를 기반으로 한 센서 네트워크 분야에 대한 연구가 다양하게 시작되고 있다.

이처럼 센서 네트워크(Sensor Network)와 같은 네트워크 임베디드(Embedded) 시스템들을 타깃으로 고안된 OS가 TinyOS이다. 그리고 TinyOS에서 구현하고자 하는 개념들을 지원하기 위해 고안된 새로운 프로그래밍 언어가 NesC이다.

본 논문에서는 TinyOS와 NesC의 전반적인 개념과 특성에 대한 소개를 제공하며, 전자부품연구원(KETI)에서 개발한 TIP-30C 센서 노드를 이용하여 개발된 라우팅 알고리즘을 테스트 하였다.

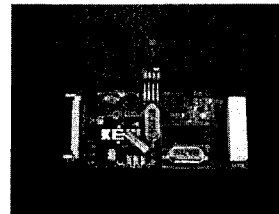
이를 위해 본 논문은, TIP-30C, TinyOS & NesC, 알고리즘에 사용되어진 핵심 컴포넌트 등에 대한 설명으로 구성되어 있으며, 결론 부분에서는 본 논문에서 개발한 라우팅 알고리즘의 성능에 대해 논하였다.

### 2. 장 본 론

#### 2.1 절 TIP-30C

전자부품연구원에서 개발한 TIP-30C는 915MHz의 RF 대역을 사용하여 센서 네트워크를 구성하는 무선 센서 네트워크 노드이다. 다양한 센서(습도, 온도, 빛 센서)가 부착되어져 이를 활용하여 센서 네트워크의 다양한 어플리케이션을 구현할 수 있다.

TIP-30C는 자체적으로 네트워크를 형성함과 동시에 통신 채널을 설정할 수 있는 Stand-Alone 컴퓨팅 노드이다. 노드들 사이의 원활한 무선 통신을 위해 915MHz 대역의 RF를 사용하였으며, UC Berkeley의 TinyOS를 기반하고 있다. 그리고 전자부품연구원(KETI)에서 개발한 Inverted F-Type 세라믹 안테나를 채택하여, 센서 네트워크 노드의 소형화라는 궁극적인 목표를 달성하고자 하였다.



<그림> TIP-30C

TIP-30C는 Atmega 128, 8비트 프로세서를 사용하였으며, 128KB의 플래시 메모리, 38K Baud의 데이터 전송속도, 컴퓨터와의 통신을 위한 UART 인터페이스, 3.0V의 전원(1.5V AAA Size 적렬연결), 15m의 RF 레인지질을 갖는다.

TIP-30C를 포함한 센서 네트워크 노드들이 반드시 확보해야 할 특성은 다음과 같다.

- ① 센서를 통한 환경 정보의 습득

#### 저자 소개

- \* 正 會 員 : 전자부품연구원 디지털미디어 연구센터
- \*\* 正 會 員 : 전자부품연구원 유비쿼터스 컴퓨팅 연구센터

② 무선 네트워크를 통한 통신(송/수신)

이를 위해 기존의 OS와 프로그래밍 언어들을 센서 네트워크 분야에 적용하려 했으나, 부적합함이 증명되었다. 따라서 여러 제약 조건에서도 복잡한 분산 알고리즘들을 처리하는데 효율적인 OS와 프로그래밍 언어를 위해 개발되어진 결과물이 TinyOS와 NesC이다.

2.2 절 TinyOS & NesC

2.2.1 절 TinyOS

TinyOS는 센서 네트워크와 같은 네트워크 임베디드 시스템들을 위해 특별히 고안된 아주 단순한 OS이다. 이는 단순 스케줄러 기능만을 제공하도록 그 기능이 제한된 초소형 용량의 OS를 만들기 위해 고안되었다.

TinyOS는 재사용이 가능한 컴포넌트 기반의 구조이다. 즉, 어플리케이션들이 구현에 필요한 각각의 컴포넌트들을 Wiring을 이용하여 연결한다. 이같이 컴포넌트 기반으로 이루어진 구조에서는, 다른 OS 서비스들로 구분된 컴포넌트들을 다른 어플리케이션에서 반복적으로 사용하지 않아도 되는 장점을 갖는다.

그리고 TinyOS는 동시성을 확보하기 위해 태스크(Task)와 이벤트(Event) 개념을 사용한다. 이 두 가지 요소들의 차이는 "선점"에 대한 여부이다. 즉, 태스크들은 서로를 선점하지 않는 반면, 이벤트들은 태스크들이나 이벤트들의 실행에 대한 선점이 가능한 차이점이 있다.

2.2.2 절 NesC

NesC는 위에서 언급된 TinyOS의 특성들을 지원하기 위해 고안된 프로그래밍 언어이다. 즉, 컴포넌트와 양방향성의 인터페이스들로 구성되고, 태스크와 이벤트들을 근간으로 동시성 모델을 제공하는 특징이 있다.

NesC는 결국 C 언어의 확장이다. C 언어에 근거를 둔 이유는, C 언어가 마이크로 컨트롤러들을 위한 효율적인 코드 생성이 가능하고, 하드웨어에 액세스하기 위해 필요한 기본적인 특성들의 지원이 가능하며, 기존의 C 코드와의 상호작용이 단순하다는 장점 때문이다. 물론 기존의 많은 프로그래머들이 C 언어에 익숙해져 있다는 점도 NesC가 C 언어에 근거를 둔 큰 이유이다.

NesC는 컴포넌트로 구성되고, Module과 Configuration이라는 두 개의 타입으로 존재한다. 모듈은 어플리케이션의 코드를 제공하거나, 한 개 이상의 인터페이스들을 수행하며, Configuration은 컴포넌트가 사용하는 인터페이스들을 연결(Wiring)한다.

컴포넌트들은 인터페이스들을 제공(Provide)하거나 사용(Use)한다. 그리고 인터페이스는 양방향이라는 특성을 갖는데, 이는 커맨드와 이벤트를 포함함으로써 가능하다. 예를 들어, 제공자(Provider)는 커맨드를 구현하는 반면에 사용자(User)는 이벤트를 구현한다.

그리고 NesC의 동시성이라는 특성을 확보하기 위해서 태스크(Task)와 Atomic 구문 두 가지의 방법을 사용하고 있다.

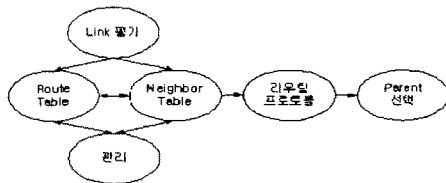
2.3 절 핵심 컴포넌트

성능 좋은 센서 네트워크의 라우팅 알고리즘을 설계하기 위해서는 노드들의 수가 아무리 많을지라도 링크의 상태와

라우팅 정보를 저장 할 수 있는 테이블(Neighbor/Routing Table), 그리고 이를 주기적으로 유지/운영 할 수 있는 테이블 관리 컴포넌트, 효율적인 링크 평가를 할 수 있는 Link 평가 컴포넌트, 적절하게 Parent를 선택하는 컴포넌트 등이 필요하다.

성능 좋은 센서 네트워크 라우팅 알고리즘을 설계하는 최종 목표는, 센서를 통해 획득한 데이터들을 Parent 노드에게 전달하기 위해 가장 효율적이고 성능이 우수한 중간 노드들을 찾아 가는 것(중간 Parent 선택)이라고 바꾸어 말할 수 있다.

본 논문에서 설계한 전체적인 라우팅 알고리즘의 개념도는 아래와 같다.



<그림> 설계된 라우팅 알고리즘

2.3.1 절 Link Estimator

위의 라우팅 알고리즘 그룹에서, 링크 평가는 소스 노드에서 타겟 노드를 선택하는 과정에서 사용한다. 각각의 노드들은 패킷들의 송/수신 이벤트(Success/Fail)를 지속적으로 관찰하여, 무선 링크의 품질을 평가한다. 발생 할 수 있는 변화에 대해 빠른 계산과, 신속한 대응, 안정성이 확보되는 Link 평가를 설계 목표로 하였다.

- ① Neighbor 테이블의 수신 평가 값의 존재 여부 확인
- ② 5회 수신된 수신 평가 값을 이용하여 얻은 새로운 수신 평가 값을 Neighbor 테이블에 저장한다.

2.3.2 절 Neighbor 테이블 관리

Neighbor 테이블에서, 관련 데이터를 관리하여, 상태가 우수한 Neighbor들의 수를 테이블 내에 일정하게 유지하며, 라우팅 테이블 내에 Sorting 된 값들을 넣는 작업을 수행한다.

테이블 관리는 다음 세 가지의 동작으로 구성된다. 즉, "삽입/삭제/보강"이다. 입력되는 패킷들에 대한 Neighbor가 어느 것인지에 대한 분석(Link 평가)이 수행 된 후, 입력된 소스가 삽입 될 것인지 혹은, 보강될 것인지에 대해 고려된다. 이에 대한 동작은 소스가 테이블 내에 있으면, 보강 기능이 수행되고, 소스가 없거나, 테이블이 가득 차 있으면, 노드는 테이블로부터 또 다른 노드를 제거할 것인지, 혹은 소스와 관련된 있는 정보를 제거 할 것인지를 선택한다.

2.3.3 절 Parent 선택

Parent의 선택은 Neighbor 테이블에서, 조건에 가장 부합하는 Neighbor를 선택하여, Parent로 선정한다. 그리고 이러한 Parent의 선정 변화가 적응수록 더욱 안정적인 Parent로 예측 가능하다.

Parent 선택을 위해 구체적으로 적용한 개념이 Cost이다. 이를 계산하는데 사용된 요소는 다음과 같다.

- ① 송/수신 평가값

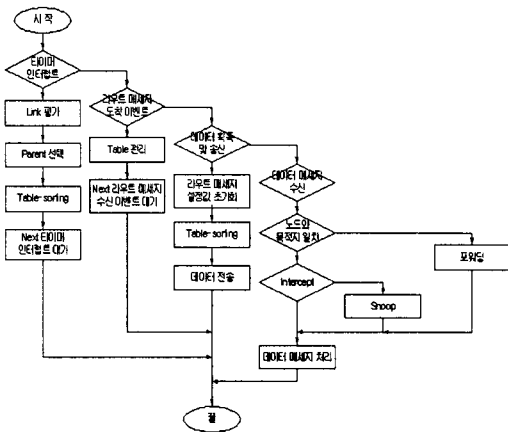
- ② 홉 카운트 수
- ③ 기준 코스트의 범위 만족 여부

### 2.3.3 절 전체 알고리즘

본 논문에서 설계된 센서 네트워크 라우팅 알고리즘은, 다음과 같은 경우 실행되도록 설계되었다.

- ① 타이머 인터럽트에 의한 주기적인 실행
- ② 라우팅 메시지 도달
- ③ 데이터를 획득한 후, 데이터 메시지 전송
- ④ 데이터 메시지를 수신

전체 라우팅 프로세스 순서도는 아래와 같다

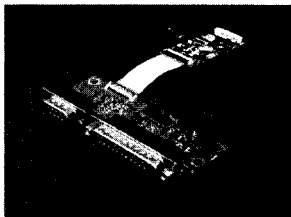


<그림> 전체 라우팅 프로세스 순서도

## 3. 장 결론

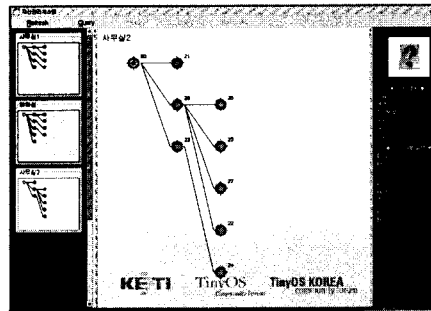
### 3.1 절 실험 결과

위와 같은 라우팅 알고리즘을 설계한 후, 이에 대한 성능을 테스트하기 위해서 본 논문에서는 다음 사진과 같은 환경에서 테스트를 실행하였다. 즉, TIP-30G가 TIP-30C 와 연결되어져 각각의 노드들이 센서를 통해 획득한 데이터를 PC로 전달하여 저장한 후 이를 분석하였다.



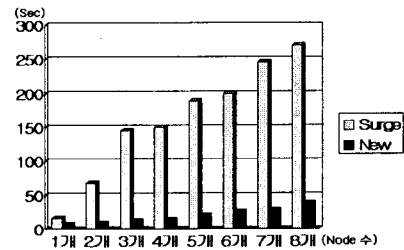
<그림> TIP-30C와 TIP-30G 연결 사진

다음의 화면 캡처는 본 논문에서 실험에 이용한 센서 네트워크 노드들이 실제 멀티홉 애드-홉 네트워크 토폴로지를 정상적으로 구성한 모니터 화면을 보여준다.



<그림> 네트워크 토폴로지 실제 화면

본 논문에서 구현된 반응 속도가 빠른 라우팅 엔진에 대한 성능을 비교하기 위해, UC Berkeley에서 배포한 Surge 어플리케이션과의 네트워크 토폴로지 형성 시간에 대해서 다음과 같이 비교/분석했다.



<그림> 실험 결과값

## 4. 장 결론

이상에서 살펴보았듯이, 센서 네트워크 분야에 있어서 링크의 평가, Neighbor 테이블 관리, Parent 선택 등은 성능 좋은 센서 네트워크 라우팅 알고리즘을 설계함에 있어서 핵심 모듈이며 이 모듈의 개선을 통해 좋은 실험 결과 값을 얻을 수 있었다. 하지만, 센서 네트워크의 다양한 라우팅 엔진은 Application Oriented 되었다고 할 수 있다. 즉, 어플리케이션의 특성에 맞게, 라우팅 알고리즘은 그 기능이 맞춤형으로 설계되어야 한다.

본 논문에서는 KETI에서 개발된 TIP-30C라는 센서 네트워크 모듈과 TIP-30G라는 인터페이스를 사용하여 관련 실험을 진행했다. 하지만, 다양한 어플리케이션에 적합한 센서 네트워크 모듈이 여러 기관이나 업체에서 개발되어 이에 대한 다양한 응용과 연구가 더욱 활발히 이루어져야 할 것이다.

## 참 고 문 헌

[1] David Gay, Philip Levis, David Culler, Eric Brewer. NesC 1.1 Language Reference Manual. May 2003.  
 [2] J.Hill, R.Szewezyk, A.Woo, D.E.Culler, and K.S.J.Pister. System Architecture Directions for Networked Sensors. In Architectural Support for Programming Languages and Operating Systems, pages 93-104, 2000.