

# 자원 효율적인 네트워크 리프로그래밍 프로토콜 설계<sup>†</sup>

(Design of resource efficient network reprogramming protocol)

최 락 현\*, 홍 원 기\*\*

(Rock-Hyun Choi and Won-Kee Hong)

**요약** 네트워크 리프로그래밍은 센서 필드에 배치된 센서 노드들의 효율적인 유지 보수를 위해 코드 업데이트와 오류 수정을 원격으로 수행할 수 있도록 하는 기술이다. 센서 노드들 간에 소량의 데이터를 전달하는 일반적인 무선 센서 네트워크 통신과는 달리 네트워크 리프로그래밍에서는 대용량 데이터의 신뢰성 있는 전달이 요구된다. 기존의 네트워크 리프로그래밍 기법은 신뢰성 있는 데이터 전달을 위해 데이터 손실 발생 시 복구를 위한 비용이 많이 들며 이로 인한 에너지 소모가 급증하는 문제점을 안고 있다. 본 논문에서는 센서 네트워크에 적합한 클러스터 기반의 리프로그래밍 기법을 제안한다. 센서 필드를 클러스터로 나누고 각 클러스터에 대표 노드를 선정하여 중복 전송과 불필요한 경쟁을 최소화함으로써 에너지 소모를 줄인다. 또한 테이블을 이용하여 노드들의 상태 정보를 통해 정확한 오류 복구를 수행함으로써 신뢰성을 높이고 있다.

**핵심주제어** : 네트워크 리프로그래밍, 코드 업데이트, 무선 센서 네트워크, 신뢰성, 클러스터

**Abstract** Network reprogramming is a technology that allows several sensor nodes deployed in sensor field to be repaired remotely. Unlike general communication in sensor network where small amount of data is transferred, network reprogramming requires reliable transfer of large amount of data. The existing network reprogramming techniques suffers high cost and large energy consumption to recover data loss in node communication. In this paper, a cluster based network reprogramming scheme is proposed for sensor network. It divides sensor field into several clusters and chooses a cluster header in charge of data relay to minimize duplicated transmission and unnecessary competition. It increases reliability by effective error recovery through status table.

**Key Words** : Network reprogramming, Code update, Wireless sensor network, Reliable communication, Network cluster

## 1. 서 론

무선 센서 네트워크(WSN : Wireless Sensor

Network)는 그 용도와 배치되는 환경에 따라서 수십 개에서 수천 개 이상의 센서 노드들이 사용된다. 배치된 노드들이 환경 데이터를 수집하고 노드들 간의 네트워크를 통하여 사용자에게 전달하게 된다. 이러한 센서네트워크는 유비쿼터스 시스템에서 가장 기본적인 기반시설에 이용되고 있다. 현재 다양한 분야에서 적용되고 있는데 대표적으로 군사, 생태관찰, 환경감

<sup>†</sup> 이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업 연구임(No. 2009-0077716).

\* 대구대학교 정보통신공학과 석사과정

\*\* 대구대학교 정보통신공학과 부교수, 교신저자

시, 재난재해 감지, 사회기반시설의 안정성 확인, 지능형 도시, 의료, 홈네트워크 등 일상생활에서부터 광범위하고 위협한 지역까지 다양하게 활용되고 있다. 이러한 다양성으로 인해 노드들은 다양한 환경에서 적용할 수 있어야 한다. 하지만 배치 초기에 모든 환경의 변수들을 파악하기란 쉽지 않다. 그렇기 때문에 배치이후에도 관리가 필요하다. 그렇지만 관리를 위해서 노드들을 수거하여 업데이트를 진행하고 오류를 수정하는 등의 작업은 시간적, 물리적으로 어려움이 있다. 이러한 문제점을 해결하기 위해서 네트워크를 통한 업데이트 데이터를 전달하고 업그레이드 할 수 있도록 하는 것이 네트워크 리프로그래밍이다[1].

이러한 네트워크 기법으로는 TinyOS에서 제공하는 단일 홉 전송방식인 XNP[2]가 있으며 이를 바탕으로 하여 대규모 센서네트워크에 적합한 다양한 기법들이 연구되었다. MOAP(Multi-hop Over Air Programming, UCLA)[3]은 리플 전달 프로토콜을 이용하여 멀티 홉으로 코드를 전달할 수 있으며, UC Berkeley에서 개발한 Deluge[4]는 업데이트 데이터를 페이지와 패킷으로 나누어 전달하는데 이를 통해서 대규모 네트워크에서 유리한 파이프라이닝 기법을 적용하였다. MNP(multihop network Reprogramming, MSU)[5]는 응답 메시지의 수를 집계하고 전송노드를 선정하고 다양한 상태모드를 정의하여 효율적인 노드관리를 통해 에너지 소모를 줄이는데 중점을 둔 기법이다. 이 외에도 다양한 기법들이 연구 및 개발되고 있다.

센서네트워크에서 사용되는 네트워크 리프로그래밍 기법에서 가장 중요한 것은 신뢰성 확보와 에너지 소모를 최소화 하는 것이다. 일반적인 변수나 센싱 데이터를 처리하는 것과는 달리 시스템 프로그래밍이기에 100% 신뢰성이 보장된 코드로 업데이트가 이루어지지 않으면 그 노드는 회수되지 않는 한 재사용이 불가능할 수도 있다. 또한 한번 필드에 배치된 노드는 회수가 어렵기 때문에 효율적인 에너지 관리는 필수이다. 하지만, 기존의 리프로그래밍 기법들을 보면 노드들을 업데이트하기 위해 주변 노드들의 정보를 활용해서 적절한 량의 데이터를 주고받는 게 아니라, 중복된 영역에 한 노드가 업데이트 되어 있지 않으면 같은 영역에서 다수의 노드들이 전달자가 되어 코드를 전달한다. 이로 인해서 무분별한 중복전송이 이루어져 에너지 낭비 문제에 노출된다. 또한 같은 영역에서 노드들이 통신을 확보하기위해 벌어지게 되는 노드들 간

의 경쟁도 발생해서 자원의 낭비는 심화된다.

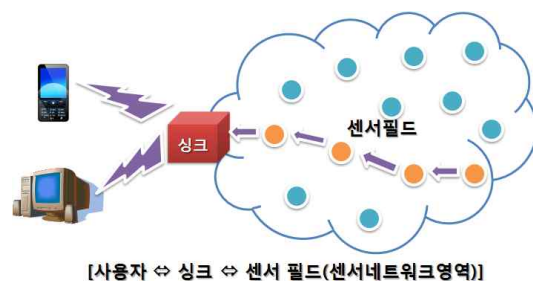
본 논문에서는 위에서 언급한 문제점들을 해결하기 위해 노드들을 일정단위로 묶고 대표노드를 선정하여 노드들의 정보에 대한 테이블을 생성 관리하여 자원 효율적인 업데이트를 진행 할 수 있도록 설계하였다. 제안된 코드업데이트 기법은 CBPD(Cluster Base Dissemination protocol)로 불리고 실험은 시뮬레이션을 통해 Deluge 기법보다 전체 데이터 전송량은 9.7%, 데이터 전달 완료시간은 5%, 그리고 에너지 소비량은 11% 가량 감소한 것을 확인 할 수 있었다.

2장은 관련연구로 기존의 리프로그래밍 기법에 대해서 알아보고 문제점을 분석한다. 3장에서는 기존의 리프로그래밍의 문제점을 해결하는 방법을 제시하고 제안하는 클러스터기반의 네트워크 리프로그래밍 프로토콜을 설계한다. 이어서 4장에서는 시뮬레이션을 통하여 기존의 기법과의 성능 평가를 시도한다. 마지막으로 5장에서 결론을 맺는다.

## 2. 관련연구

### 2.1 센싱 데이터 전달 라우팅 프로토콜

(그림 1)에서와 같이 일반적으로 센서네트워크에서는 노드가 주변 환경정보를 수집하고 노드들 간의 네트워크를 통하여 싱크노드로 정보를 보내게 된다. 이러한 동작은 모든 노드들이 시도한다. 이렇게 싱크에 모인 정보들은 네트워크 서버를 통하여 사용자에게 전달하게 된다. 이때 노드들은 에너지 소모를 최소화 하고 신속한 데이터 전달을 위해서 싱크까지 최적의 경로를 선정하는 기술이 필요하게 된다.



(그림 1) 센서네트워크 데이터 전달 체계

데이터 전달 과정 체계에 따라 크게 평면적 라우팅, 계층적 라우팅, 위치기반 라우팅으로 분류할 수 있으며, 이를 정리하면 <표1>과 같다[6].

<표 1> 센서 네트워크의 라우팅 프로토콜 분류

분류	라우팅 프로토콜
평면적	flooding, SPIN, Directed Diffusion
계층적	LEACH, PEGASIS
위치 기반	GAF, GEAR

일반적으로 라우팅 프로토콜들은 센서 노드의 에너지 소모를 최소화하고 효율적인 데이터 전달을 고려하고 있다. 데이터 전달의 가장 기본적인 플러딩 기법에서 발생할 수 있는 데이터 중복을 줄이는 기법, Ad-hoc 라우팅 프로토콜 기반에서의 제어에 대한 오버헤드를 줄이는 연구, 최단 경로 설정을 통한 최소의 홵(hop)을 통한 데이터 전달 연구가 활발히 진행되었다.

최근에는 노드들 간의 균형적인 에너지 소비를 통해 네트워크의 생명력을 오래 지속 시키는 연구도 활발하게 진행 중이다. 이를 위한 에너지 인식 기반의 데이터 전달 프로토콜로는 EADR, SAFE, LAMaR 등이 있다[8].

또한 최근에는 IETF의 ROLL Working Group에서 RPL(IPv6 Routing Protocol for Low power and Lossy Networks)에 대한 논의가 활발하게 진행 중에 있다. RPL은 현재 DAG(Directed Acyclic Graph)을 바탕으로 한 경로 설정과 데이터 전송을 시도하고 있다[9]. 이 표준화 작업을 통해 네트워크 라우팅에 전반적인 조건을 고루 만족 시켜줄 수 있는 프로토콜을 설계 중이다.

또한 대규모 네트워크에서 효율적이고 원활한 데이터 전송을 위해서 클러스터 기법을 사용하기도 하는데 이러한 기법으로는 LEACH(Low-Energy Adaptive Clustering hierarchy) 프로토콜이 대표적이다[10].

## 2.2 센서네트워크 리프로그래밍

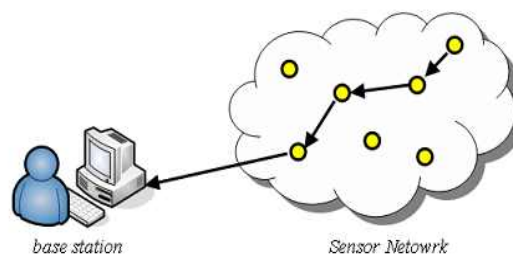
### 2.2.1 리프로그래밍 데이터 전달 방식

무선 센서 네트워크에서 일반적인 데이터 전달은

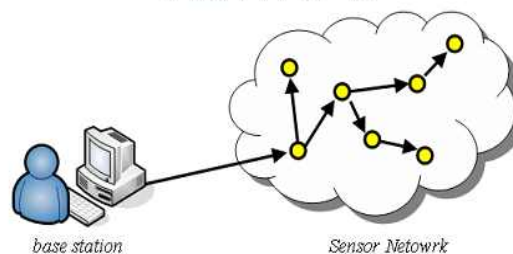
하나의 노드에서 수집한 데이터를 싱크노드까지 여러 노드를 거쳐 데이터를 전달하는 방식이다. 한마디로 다수의 노드의 데이터들이 결국에는 싱크노드 하나로 모이게 되는 방식을 취하고 있다.

하지만 리프로그래밍에서는 반대의 성향을 띄고 있다. 사용자측에서 네트워크 내의 모든 노드로 보낼 업데이트를 위한 데이터를 전달하게 되고 그 데이터들이 싱크노드로부터 전체 네트워크로 전달되어야 한다.

(그림 2)와 같이 데이터 전달에 있어서 사용하는 네트워크 경로는 같지만 대상과 방식이 다르다. 그렇기 때문에 기존의 데이터 전달 프로토콜을 리프로그래밍을 위한 데이터 전달 프로토콜에 바로 적용하기에는 어려움이 있다.



(a) 센싱 데이터 전달 과정



(b) 코드 업데이트 데이터 전달 과정

(그림 2) 데이터 전달 과정 비교

리프로그래밍을 위한 데이터는 센서가 수집하는 정보보다 많은 양의 데이터를 전달해야 된다. 또한 시스템 파일로써 그 중요도가 매우 높기 때문에 100%의 신뢰성을 요구하게 된다. 이렇듯 리프로그래밍을 위한 데이터 전송에는 다양한 요구조건이 명시되어야 한다. 대표적인 요구조건을 나열하면 다음과 같다.

1) 센서 네트워크 내의 모든 노드들에게 100% 전달할 수 있도록 신뢰성이 보장되어야 한다. 이것은 해당하는 네트워크 내의 모든 노드들에게 업데이트 데이

터를 전달해야 되고, 무선 통신으로 손실되는 데이터에 대한 복구 기능이 요구된다. 이는 일반적인 데이터와는 달리 노드를 운영하는 필수 데이터로써 그 중요도가 매우 높기 때문이다.

2) 업데이트를 위한 데이터 전송으로 인한 에너지 소모를 최소화하여 설계해야 한다. 센서 노드들은 임베디드 시스템으로써 제한적인 에너지를 가지고 있기 때문에 에너지 소비는 센서 노드의 수명 시간과 직결되는 가장 중요한 요소이다.

3) 메모리와 저장 공간의 효율적인 사용이 요구된다. 이는 노드가 제한적인 자원을 가지고 있기 때문이다. 또한 일반적인 센싱 데이터 전송과 업데이트를 위한 데이터 전송에 있어서 주고받는 데이터의 크기가 다르기 때문에 이를 고려하여 메모리와 저장 공간을 효율적으로 사용할 수 있도록 설계하는 것이 중요하다.

### 2.2.2 대표적인 네트워크 리프로그래밍

무선 센서 네트워크에서 노드의 원격 코드 업데이트 기능을 지원하는 네트워크 프로그래밍 기법에 대하여 많은 연구가 진행 되어 왔다.

대표적인 프로그래밍 기법으로는 먼저 단일 홉(hop) 내 노드의 TinyOS[7] 커널(kernel)과 응용(application) 코드를 업데이트 하는 XNP(Crossbow Network Programming, Crossbow 社)[2], 리플 전달 프로토콜(Ripple Dissemination Protocol)을 사용하여 멀티 홉(multi-hop)으로 코드 업데이트 데이터를 전달할 수 있는 MOAP (Multi-hop Over Air Programming, UCLA)[3], XNP를 확장하여 멀티 홉을 지원하고 코드 업데이트 데이터를 페이지와 패킷으로 표현하여 파이프라이닝 형태로 데이터를 전달하는 Deluge(UB Berkeley)가 있다[4]. 먼저 XNP는 기존의 ISP(In System Programming) 기능을 단지 무선으로 프로그래밍이 가능하도록 되어 있어 전달가능한 데이터는 1 홉(hop)으로 제약되어 있다. MOAP는 윈도우 슬라이딩과 비슷한 비트 벡터 테이블 비트 플래그 조작으로 패킷 손실이 발생하면 재전송 요구 메시지를 해당 상위 코드 버전 노드에게 송신한다. MNP(Multihop Network Reprogramming, MSU)[5]는 응답 메시지의 수를 집계하여 전송노드를 선택하는 기법을 정의하고 노드들의 상태 모드를 정의하여 효율적인 코드 업데이트 데이터를 전달하는 프로토콜이다. 전송 중 발생한 오류를 복구하기 위해 MissingVector을 이용해 하

위 노드들의 오류 영역을 확인하여 ForwardVector에 복구 데이터를 담아 일괄적으로 재전송하는 복구 방법으로 데이터 전송 횟수를 효과적으로 줄였다. Deluge는 페이지와 패킷으로 분할 된 데이터를 이용해 전달한다. 페이지 단위로 패킷을 전달함으로써 파이프라이닝(Pipelining) 형태로 전달할 수 있다. 분할된 데이터를 파이프라이닝 형식으로 보내기에 좀 더 효율적으로 안정적인 데이터 전달을 할 수 있다. 또한 패킷 손실이 발생했을 때는 MOAP의 슬라이딩 윈도우 기법과 비슷한 비트 벡터 테이블의 비트 플래그 조작으로 재전송 요구 메시지를 상위 코드 버전 노드에 송신한다.

그 외에도 센서네트워크에서의 리프로그래밍을 위한 다양한 기법들이 연구되고 있다. <표 2>에서는 리프로그래밍의 기법들을 나열하고 그 기법들의 Encoding/Decoding 방식, MAC 종류, 전송 가능한 Hop, 네트워크 내에서의 Scope, Pipelining 기법의 적용 유무의 특징을 정리하였다[1].

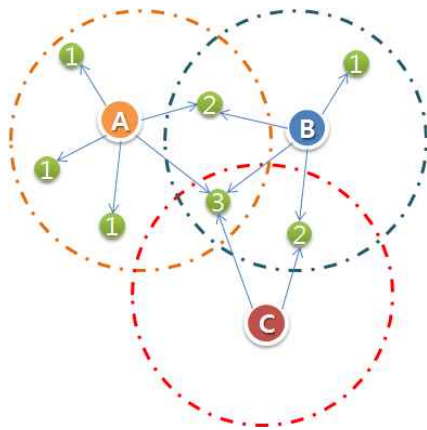
<표 2> 리프로그래밍의 종류와 특징[1]

Name	Encoding/decoding	MAC	Hop	Scope	Hierachy	Pipelining
XNP	Complete program	CSMA	Single-hop	Whole network	No	No
Reijers approach	Platform-dependent patch	CSMA	Single-hop	Whole network	No	No
Incremental	Platform-dependent patch	CSMA	Single-hop	Whole network	No	No
Trickle	Metascript	CSMA	Multihop	Whole network	No	No
MOAP	Complete program	CSMA	Multihop	Whole network	No	No
Deluge	Complete program	CSMA	Multihop	Whole network	No	Yes
MNP	Complete program	CSMA	Multihop	Whole network	No	Yes
Sprinkler	Complete program	TDMA	Multihop	Whole network	Yes	No
Firecracker	Complete program	CSMA	Multihop	Whole network	Yes	No
Aqueduct	Complete program	CSMA	Multihop	Selected nodes	No	Yes
TinyCubus	Modular update	CSMA	Multihop	Selected nodes	No	No

### 3. 클러스터기반의 리프로그래밍

#### 3.1 기존 노드 선정 및 전송의 문제점

일반적으로 센서네트워크 리프로그래밍의 기본적인 철학은 주변 노드의 정보를 고려하지 않고 스스로 판단하여 업데이트를 진행 한다. 하지만 이러한 방식은 전송 노드 선정 및 데이터 전송에 있어서 통신범위안에서 중복 데이터 전송 및 충돌을 발생 시킬 가능성이 매우 크다.



(그림 3) 중복 영역의 문제점

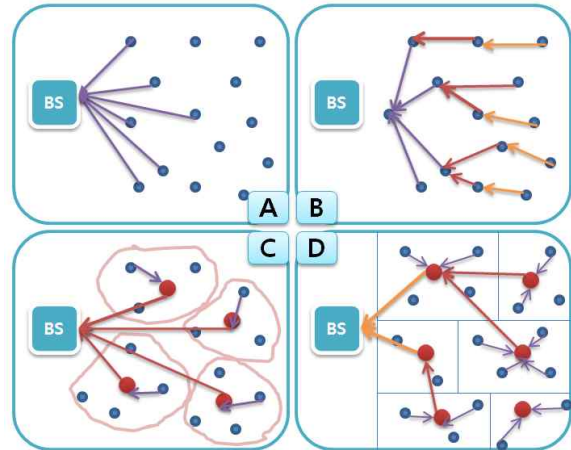
(그림 3)에서 보는 것과 같이 일반적으로 노드들은 전송노드로부터 하나의 데이터만 받아야 하지만 때에 따라서는 전송을 두 번 혹은 3번 받게 되는 경우도 발생하게 된다. 그림에서는 A와 B사이에서 통신범위가 겹치는 노드가 발생하게 되고 노드는 이로 인해 데이터를 중복으로 받게 되거나 데이터들 간의 충돌과 결합으로 인하여 데이터전송에 문제가 발생할 수 있다. 이러한 문제가 발생하는 이유는 노드들 사이에 의사소통과 노드들의 소속이 없기 때문이다.

또한 노드들은 언제 네트워크 전체가 업데이트가 완료 되었는지를 알 수가 없다. 노드들은 같은 버전의 프로그램으로 동작하는 것이 가장 신뢰성을 높일 수 있기 때문에 이 문제 또한 중요한 문제가 될 수 있다. 그리고 불필요한 대기시간이 길어 질수 있다.

이러한 문제점을 해결하기 위해서 본 논문에서는 클러스터기반의 네트워크 리프로그래밍을 위한 데이터 전달 프로토콜을 제안한다.

#### 3.2 클러스터 기반의 데이터 전달 기법

클러스터 데이터 전달 방식은 노드 각자가 데이터를 목적지까지 전달하는 것이 아니라 대표노드를 통해서 데이터를 전달하는 것이다[11].

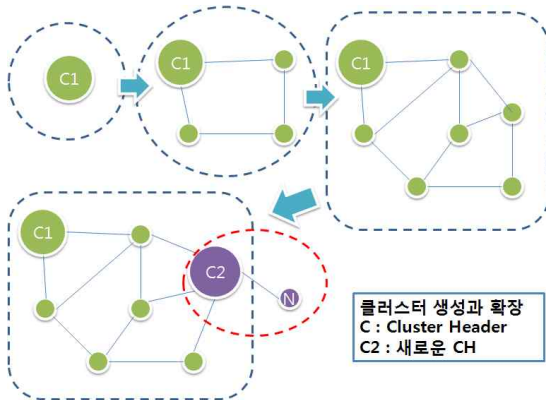


(그림 4) 센서네트워크에서의 데이터 전달 방식 (A)단일 홉의 전송, (B)다중홉 전송, (C) 클러스터기반의 단일홉, (D) 클러스터기반의 다중홉 전송

(그림 4)는 일반적인 데이터 전달 방식과 클러스터를 기반으로 하는 데이터 전달 방식을 도식화하고 있다. 클러스터 기반의 데이터 전송은 대역폭의 효율적인 관리와 클러스터 내의 데이터를 병합을 수행할 수 있기 때문에 효율적인 구조다. 대규모 네트워크에 이용했을 경우 유용하다. 이는 노드들이 밀집된 곳에서는 데이터 중복으로 인한 트래픽이 증가 될 수 있는 특성을 가지기 때문에 효율적인 운영을 위해서 클러스터 기반의 데이터 전송이 많이 사용되고 있다.

(그림 5)는 클러스터의 생성과 확장을 도식화 한 그림이다. 하나의 대표 노드가 포함할 수 있는 노드의 수 이상이 될 때에 가장 멀리 있는 노드를 새로운 대표노드로 선정하여 새로운 클러스터를 형성하게 된다. 클러스터 기반의 데이터 전달을 위해서는 우선적으로 네트워크 내의 노드들을 클러스터링 하여야 한다. 이를 위한 과정은 기존의 데이터 전달 프로토콜을 활용하여 Advertisement-Request-Data 과정을 통해서 클러스터를 생성한다. Advertisement를 통해서 대표노드의 ID를 함께 전달하고 전달받은 노드들은 데이터전송 수락과 함께 클러스터 가입을 신청하게 된다. 대표





(그림 5) 클러스터 생성과 확장

노드는 응답메시지를 토대로 클러스터의 테이블을 만들고 노드들의 상태 정보를 저장하고 관리하게 되며, 이 때 주고받는 데이터는 아래 <표 3>과 같다.

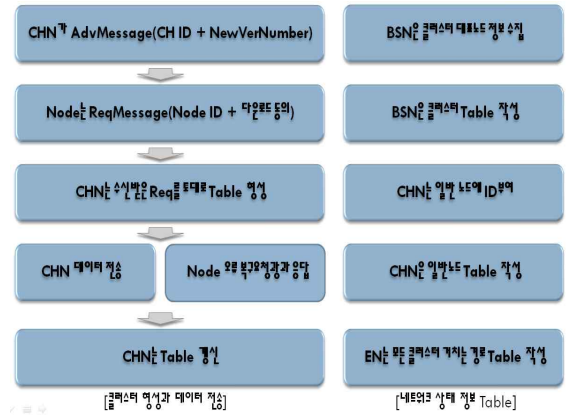
<표 3> 컨트롤 메시지

Adv Message	Req Message
New Ver. Number + 대표노드 ID	수락메시지 + 자신의 ID

대표노드는 수신 받은 Req Message를 바탕으로 자신의 ID를 대표로하는 클러스터를 생성하게 된다. 또한 클러스터에 가입된 노드들에 대한 테이블을 작성하게 되는데 이는 네트워크의 크기와 노드의 배치간격을 고려하여 설정하게 된다. 노드하나의 RF반경 내에 배치될 수 있는 노드의 수보다는 크며 최대 포함 가능한 노드의 수는 255개로 제한된다. 클러스터 테이블이 생성되는 전체적인 과정은 (그림 6)과 같다.

이렇게 생성된 테이블은 노드의 상태를 0과 1을 이용하여 데이터 전달의 완료 상태를 표시하게 된다. 기본적으로 1로 초기화 되어있으며 데이터 송신 후 0으로 변하게 된다. 재전송 요청된 노드는 1이 되고 일정 시간 대기 이후 더 이상의 재전송 요청이 없으면 1이 된 노드들에 해당하는 데이터를 전송해주게 된다. 재전송 받은 노드들이 복구완료 메시지를 대표노드에게 전달하게 되면 대표노드의 테이블은 모두 0이 되면 1로 리셋 되고 다음 단계로 넘어가게 된다.

클러스터 생성이후 클러스터 대표노드는 자신의 정보를 베이스스테이션으로 보내게 되고 베이스스테이



(그림 6) 수행과정

션은 이를 조합하여 테이블을 생성하고 관리하게 된다.

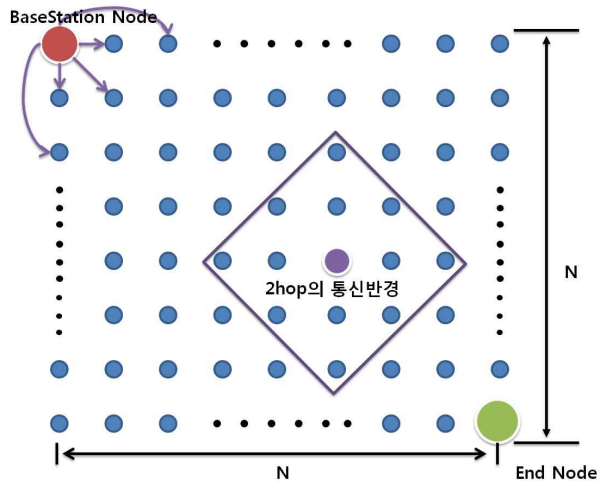
또한 전체적인 완료 상태를 주기적으로 확인하기 위해서 EN(EndNode)를 배치한다. EN는 노드 배포 초기에 베이스스테이션으로부터 가장 멀리 떨어진 노드를 선정한다. EN는 모든 클러스터를 거치는 경로를 가지게 되며 이 경로를 통해 주기적으로 클러스터들의 상태 정보를 확인하고 자신에게 업데이트 정보가 왔을 때에는 전체적인 노드의 완료 상태를 확인하여 베이스스테이션에 전달하게 된다. 이를 통해서 전반적인 업데이트 진행 상황을 파악하게 되고 빠르게 상태 전환을 시도하여 기존의 환경정보 수집 상태로 돌아갈 수 있게 해준다.

클러스터 기반의 데이터 전송을 적용하는 것은 노드들 간의 중복 전송을 방지하기 위해서 하나의 노드가 주변 정보를 수집하고 사용하는 것이다. 그렇기 때문에 노드들은 광고메시지를 수신 받게 되면 최초의 광고메시지에게 응답을 하고 응답이후에는 다른 광고메시지가 오더라도 받지 않는다. 그 외의 기본적인 전송관련 기법들은 Deluge를 기본 바탕으로 하여 데이터 전달을 진행한다.

#### 4. 실험

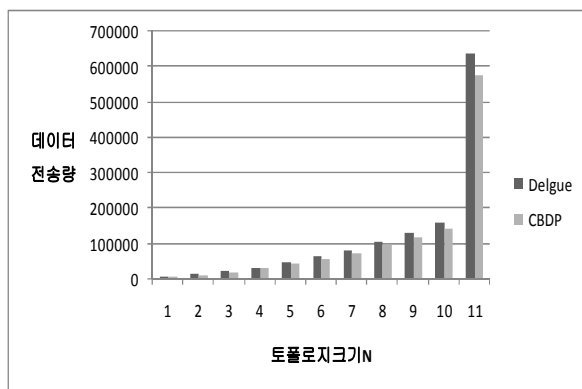
클러스터 기반의 리프로그래밍 기법의 성능평가를 위해서 Deluge와의 비교 분석을 하였다. 이를 위해서 N×N 크기의 격자 형태의 토폴로지 (그림 7)를 구성하고 크기에 따라 각 데이터 전달 프로토콜의 데이터

메시지 전송 횟수, 데이터 전달 소용시간 에너지 소비량을 평가항목으로 선정하고 시뮬레이션을 실시하였다.



(그림 7) 실험환경 토폴로지

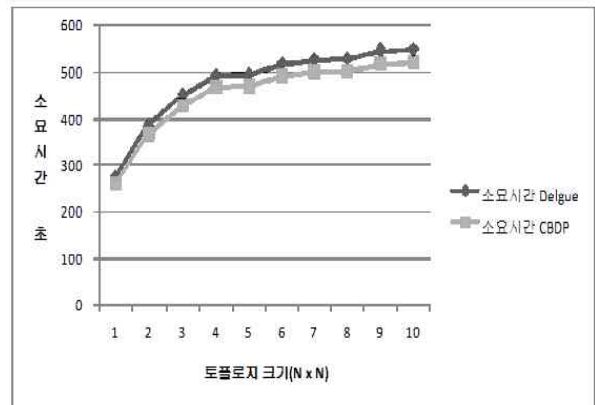
실험을 위해서 TinyOS에서 제공하는 TOSSIM 시뮬레이터[12] 환경 변수 값을 참고 하였으며 무선 통신에 따른 패킷 손실률을 10%로 일정하다고 가정하였다. 데이터 전송속도는 120kbps, RF 전송 범위는 2hop 거리로 가정하였다. 데이터의 크기는 22080byte의 파일을 20page×48packet×23byte로 구성하였다.



(그림 8) 토폴로지 크기에 따른 데이터 전송량

Deluge와의 데이터 전송량을 비교하였을 때 토폴로지의 크기가 증가할수록 제안하는 클러스터 기법이 적은 양의 데이터를 전송한 것을 알 수 있다. 이러한 결과는 클러스터링을 통한 데이터의 중복 전송이 없

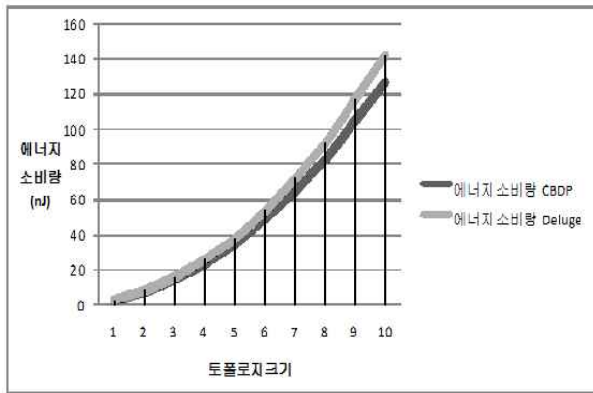
어지고 데이터 충돌 및 경쟁으로 인한 데이터 손실 복구를 위한 재전송이 줄어들었기 때문에 실험 결과 (그림 8)을 통해 17%가량 전송량이 감소된 것을 확인했다.



(그림 9) 데이터 전달 완료 소요시간 비교

(그림 9)는 업데이트 데이터 전달에 소요되는 시간을 비교한 그래프이다. 제안하는 기법이 노드들의 상태 테이블을 통하여 데이터 충돌 및 경쟁으로 일어날 수 있는 전송 지연을 감소시켜 Deluge보다 5%가량 시간을 단축시킬 수 있었다. 또한 EndNode 설정을 통하여 전체 노드들의 전송완료시간을 빠르게 인식할 수 있었다. 하지만 클러스터링 과정과 테이블 생성과 확인하는 과정에 소모된 시간으로 인해서 많은 차이를 나타내지는 못하였다.

(그림 10)은 전송완료까지의 시간과 데이터 전송량을 및 메모리 사용을 고려하여 에너지 소비량을 비교하였다. 클러스터링과 EndNode를 통하여 데이터의 중복 전송 및 오류 복구를 위한 재전송을 감소시키고 불필요한 대기시간을 감소시킨 결과 Deluge보다 11%가량의 에너지 소모를 감축시킬 수 있었다. 하지만 클러스터 형성과정에서 연산과 테이블 생성 및 관리에 에너지 소모가 컸기 때문에 많은 성능 향상이 이루어지지 않았다.



(그림 10) 에너지소비량 비교

## 5. 결론

무선 센서네트워크에서 한번 배치된 노드들은 다양한 변수에 의해서 시스템 장애가 발생할 수 있으며 성능 개선을 위한 네트워크 프로그래밍 기법을 필요로 하게 된다. 이러한 네트워크 리프로그래밍은 일반적인 센서네트워크에서의 데이터와는 달리 시스템 데이터로써 데이터 전송에 있어서 높은 신뢰도를 요구하게 된다. 이는 네트워크 내의 모든 노드들에게 동일한 업데이트 데이터를 완벽하게 전달하는 것을 말하며 또한 네트워크의 수명에 영향을 최소화 하여야 한다.

본 논문에서는 기존의 네트워크 리프로그래밍 기법에서의 문제점 이었던 중복 전송과 과도한 경쟁, 데이터 충돌로 인한 전송 지연 문제를 해결하기 위해서 클러스터 기반의 데이터 전송을 제안하였다.

제안한 프로토콜은 노드들을 전송 가능한 홉 이내의 노드들을 묶어 클러스터 단위로 관리하고 대표노드를 선정하여 데이터 전송을 주관하며 노드들의 상태정보를 수집하게 된다. 이를 통해서 데이터 전송에 있어 노드들 간의 경쟁을 피하고 중복전송 및 데이터 충돌을 없앤다. 또한 EndNode를 선정하여 주기적으로 전체 클러스터들의 상태를 베이스스테이션으로 알려주어 업데이트 시점을 결정하고 불필요한 대기시간을 최소화해준다.

성능평가를 위해서 TinyOS에서 제공하는 TOSSIM 시뮬레이터 환경 변수 값을 참고하여  $N \times N$ 의 격차 토폴로지의 크기를 변화해가면서 데이터의 전송량과 완

료시간, 에너지 소비량을 확인하고 Deluge와 비교 하였다. 이 결과 클러스터 기반의 데이터 전송이 전송량 9.7%, 데이터 전달 완료시간 5%, 에너지 소비량은 11%가량 감소한 결과를 보였다. 향후 보다 다양한 실험과 추가 연구를 통해 고정된 토폴로지에서 에너지 소비 예측 경향 파악을 통해 제2차 클러스터 형성 시 통신 횟수를 줄일 경우 에너지 소모를 보다 줄일 수 있는 클러스터 네트워크 리프로그래밍 프로토콜을 설계할 수 있을 것이다.

## 참 고 문 헌

- [1] Qiang Wang, Yaoyao Zhu, and Liang Cheng, Lehigh University, "Reprogramming Wireless Sensor Networks:Challenges and Approaches", IEEE Network, May/June 2006.
- [2] Crossbow Technology Inc. "Mote In-Network Programming User Reference Version 20030315", <http://webs.cs.berkeley.edu/tos/tinyos-1.x/doc/Xnp.pdf>, 2003.
- [3] Stathopoulos, T. and et. al., "A Remote Code Update Mechanism for Wireless Sensor Network", Technical Report CENS-TR-30, UCLA Center for Embedded Networked Computing, 2003.
- [4] Hui, J. W. and et. al., "Deluge: Dissemination Protocols for Network Programming at Scale", <http://www.cs.berkeley.edu/~jwhui/research/Deluge/cs262/cs262a-report.pdf>, 2003.
- [5] SS Kulkarni, L Wang, "MNP Multihop Network Reprogramming Service for Sensor Networks", 2nd international conference on Embedded networked sensor systems, pp.285-286, 2004.
- [6] 이승일, "무선 센서 네트워크에서 저비용의 코드 업데이트 데이터 전달 프로토콜", 대구대학교 대학원 공학석사 학위 논문, 2006.
- [7] U.C. Berkeley as a part of the TinyOS Project. "Deluge 2.0 - TinyOS Network Programming", <http://www.cs.berkeley.edu/~jwhui/Deluge/documentation.html> July 28, 2005.
- [8] Akkaya K. and et. al., "A Survey on Routing



Protocols for Wireless Sensor Network”, Elsevier Ad Hoc Network Journal, Vol3/3, pp.325-349, 2005.

- [9] T. Winter and et. al., “RPL: IPv6 Routing Protocol for Low power and Lossy Networks”, draft-ietf-roll-rpl-05, June 10, 2010.
- [10] Vinayak Naik, Anish Arora, Prasun Sinha, and Hongwei Zhang, “Sprinkler: A Reliable and Energy Efficient Data Dissemination Service for Extreme Scale Wireless Networks of Embedded Devices” IEEE TRANSACTIONS ON MOBILE COMPUTING, Vol. 6, No. 7, July 2007.
- [11] S.G. Foss and S.A. Zuvey, “On a certain segment process with Voronoi clustering”. INRIA, Rapport de Recherche No. 1993.
- [12] Levis, P. and et. al., “TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications”, In Proceedings of the First ACM Conference on Embedded Networked Sensor Systems, 2003.



**최 락 현** (Rock-Hyun Choi)

- 학생회원
- 2010년 2월 : 대구대학교 통신공학과 공학사
- 2010년 3월 : 대구대학교 정보통신학과 공학석사과정
- 관심분야 : WSN, 임베디드 시스템, IMU



**홍 원 기** (Won-Kee Hong)

- 종신회원
- 2001년 2월: 연세대학교 컴퓨터공학과 공학박사
- 2001-2002년: University of California, Irvine, Post-doctrual Fellows
- 2004년~현재 : 대구대학교 정보통신공학부 부교수
- 관심분야 : WSN, 임베디드 시스템

논문 접수 일 : 2010년 08월 05일  
 1차수정완료일 : 2010년 09월 13일  
 게재 확정 일 : 2010년 09월 14일