

무선 센서 네트워크에서 빠른 코드분배를 위한 다수분배자 선정 방법*

김미희** · 홍준석***

A Multiple-Disseminators Determining Mechanism for Fast Code Dissemination in Wireless Sensor Networks*

Mihui Kim** · June S. Hong***

■ Abstract ■

In this paper, we propose a multiple-disseminators determining mechanism for Efficient Code Dissemination with low-delay(ECoDi) for wireless sensor networks (WSN). Code dissemination is in the spotlight as an important research issue since sensor nodes are necessary for updating new software remotely or fixing bugs dynamically. In particular, the time factor for code dissemination is the most important factor in order that the normal operation of nodes can be continuously performed as soon as finishing the dissemination. For this factor, ECoDi determines the set of disseminators through regression analysis based on the size of distributed code and the time of past unicasts and broadcasts. Then it transmits the entire code as a unicast to multiple disseminators, and the disseminators broadcast the code to the remaining neighbor nodes. Performance results on a testbed show that ECoDi reduces dissemination time significantly compared to a conventional scheme.

Keyword : Fast Code Dissemination, Mutiple Disseminators, Regression Analysis, WSN

논문투고일 : 2011년 04월 25일 논문수정완료일 : 2011년 06월 01일 논문게재확정일 : 2011년 06월 02일
* 이 논문은 2009년 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2009-352-D00286).
** 국립한경대학교 컴퓨터공학과 전임강사
*** 경기대학교 경영정보학과 부교수, 교신저자

1. 서론

최근 유비쿼터스 컴퓨팅 환경을 현실적으로 구현 하는데 사람들의 관심이 모아지면서 이에 대한 핵심 기술 중의 하나인 무선 센서 네트워크 기술의 중요성이 점점 강조되고 있다. 특히 많은 센서로 구성되어 있는 빌딩/산업 자동화를 위한 센서 네트워크나 환경 감시 등을 위하여 외부에 설치된 센서 네트워크의 경우, 운용 프로세스의 자동화 및 효율성, 범위성(scalability)은 센서 네트워크의 성공적인 적용을 위한 핵심적인 요소이다[4, 22]. 그 중에서도 센서 노드에서 실행되는 코드의 기능에 대한 업그레이드나 버그 수정을 위해 무선 환경으로 실행 코드를 배포하는 코드 분배방법은 특히 중요한 분야라고 할 수 있다[1, 8, 12].

센서 네트워크에서 효율성을 향상시킨 코드 분배방법에 관한 연구로는, 첫째 코드 분배노드에서 전송 코드의 양을 줄이기 위한 코드 사이즈를 최소화시키는 방안에 관한 연구[11, 17, 18], 둘째 효율적인 코드 분배 프로토콜로서 전송 시 에너지 효율을 증진하고 불필요한 제어 메시지 전송을 지양하며 무선 에너지를 고려한 코딩 방법에 관한 연구[1, 8, 9, 15, 20, 21], 마지막으로 센서 노드에서 수신한 코드를 실행 시 동적 링킹을 활용한 효율적인 실행 방법에 관한 연구[3, 17] 등이 있다. 이 중에서 효율적인 코드 분배 프로토콜 연구로서 다수 분배자(multiple-disseminator)에 의해 코드 분배의 지연 시간을 단축시키는 연구가 있었으나, 적절한 다수 분배자의 수를 결정하고 다수 분배자를 어떻게 선택할 것인지에 대해서는 연구되지 않아 이에 대한 연구가 필요하다.

이에 본 논문에서는 센서 네트워크를 구성하는 센서 노드의 수, 전송 시간의 환경적 요소, 분배되는 코드 사이즈에 따라 다수 분배자의 적정 수를 결정하고, 분배자 역할을 수행할 센서 노드를 결정하는 방법을 제안한다. 이러한 의사결정을 위하여 회귀분석을 이용하였으며, 다수 분배자 선택 방법을 간단하게 설계하기 위해서 영역 분할법을 사용하였다(즉, 분할된 영역의 중앙 노드가 분배자로

선택된다). 이렇게 선택된 다수 분배자까지는 유니캐스트 또는 멀티캐스트로 빠르게 전송되며, 각 분배자는 코드 분배의 기본 단위인 한 페이지를 수신하게 되면 이후 자신의 영역의 센서 노드들에게 브로드캐스팅에 의한 기존 코드 분배방법을 수행하게 된다. 본 연구를 통해 낮은 지연성을 달성하는 코드 분배를 위해 환경에 적합한 다수의 코드 분배자 선정 방법을 제안하였고, 제안된 메커니즘을 72개 센서노드로 구성된 센서 네트워크 테스트베드에서 실험하여 기존 코드 분배방법에 비해 코드 분배 지연시간이 최고 19%까지 단축됨을 보임으로써 우수성을 검증하였다.

본 논문의 구성은 다음과 같다. 제 1장의 서론에 이어 제 2장에서는 기존 코드 분배방법의 특징 및 문제점을 소개하고, 제안한 기법에서 사용하는 회귀분석에 대해 설명한다. 제 3장에서는 이를 바탕으로 제안하는 빠른 코드분배를 위한 다수 코드 분배자 선정방법에 대해 자세히 소개한다. 제 4장에서는 테스트베드에 대한 실험 결과를 분석하고, 마지막으로 제 5장에서 결론을 맺고자 한다.

2. 관련 연구

2.1 효율적인 코드분배를 위한 연구동향

센서 네트워크에서 효율적인 코드 분배를 위해서는 다음과 같은 요구사항이 충족되어야 한다. 코드 분배에 따른 오버헤드를 낮추고, 코드 분배 과정이 남은 리소스에 따라 적절히 운용되어야 하며, 플래시 메모리 리라이팅(rewriting), 센서 네트워크 라이프타임의 영향 및 프로세싱 처리량 등이 최소화되어야 한다. 또한 코드 분배에 소요되는 메모리 리소스 사용을 제한하고, 응용 프로그램에 대한 인터럽트 및 제어메시지의 전송을 제한해야 한다. 이러한 요구사항은 센서 네트워크의 특징에 의한 것으로 대부분이 센서 노드의 제한적인 자원을 효율적으로 활용하는데 초점이 맞추어져 있다[1, 2].

이러한 요구사항에 맞추어 기존 효율적인 코드 분배 방법들은 크게 전송시스템 측면, 네트워크에

서 운용되는 프로토콜 측면, 수신 시스템 측면의 세 분야로 나뉘어 연구되어 왔다. 첫 번째 연구 분야에서는 전송 에너지를 줄이기 위하여 전송시스템 측에서 전송 전에 분배할 코드의 사이즈를 최소화하기 위한 방법이 연구되었다[11, 16, 18]. 두 번째 연구 분야에서는 코드분배를 위한 제어메시지나 분배되는 코드의 양을 줄이고 손실이 발생하더라도 재전송되는 코드를 최소화하여 전송 에너지를 줄여주는 프로토콜이 연구되었다[1, 8, 15, 20, 21]. 세 번째 연구 분야에서는 수신시스템 측에서 코드 수신 후 실행 시 에너지 효율성을 위한 동적 링킹에 관한 연구가 진행되었다[3, 17].

이 중 효율적인 코드분배 프로토콜에 대한 대표적인 연구로서, Deluge[8]에서는 불필요한 중복 메시지 전송을 억제하는 방법을 제안하였고, Freshet [12]에서는 활동이 요구되지 않을 때에는 Sleep Mode로 전환하여 에너지 효율을 증진시키는 방법을 제안하였다. 또한 빠른 분배를 위해 다수의 분배자 노드의 중요성을 강조하였으며, 분배자 노드 사이에서 서로 다른 페이지(전송 코드의 단위 유닛)의 코드를 분배함으로써 빠른 분배가 가능함을 보였다. Firecracker[14]에서도 다수 분배자 노드들이 멀리 떨어져질수록 더 빠른 코드 분배가 이루어짐을 보였다.

그러나 기존 연구들에서는 몇 개의 다수 분배자 노드를 설정해야 하는지와 그 노드의 위치는 어떻게 분포되어야 하는지에 대한 연구는 구체적으로 진행되지 않았다. 이러한 문제는 다수 분배자에 의한 빠른 코드 분배를 실제로 구현하기 위해서는 필수적으로 제공되어야 할 메커니즘이다.

본 논문에서는 위와 같은 문제점을 해결하기 위하여 네트워크 토폴로지나 전체 노드의 수, 전송 지연시간, 분배되는 코드 사이즈에 따라 적절한 분배자의 수를 결정하고, 다수분배자의 적정 위치를 결정해 주는 메커니즘을 제안하고자 한다.

2.2 회귀분석

본 논문에서 제안하는 ECoDi 기법은 네트워크

의 상황, 즉 과거 네트워크에서의 전송 시간에 대한 정보를 가지고, 최적의 분배자 노드 수와 위치를 결정하게 된다. 본 절에서는 이를 위해 사용되는 회귀분석법에 대해 설명한다.

회귀분석은 통계 및 경제학 뿐만 아니라 이미지 품질 평가 또는 채널 추정 등 다양한 컴퓨터공학 분야에서 이용되었으며, 이미 알려진 독립변수들의 값으로부터 종속변수의 값을 예측하기 위하여 선형 모형을 추정하는 방법으로 많이 사용되어 왔다[5-7]. 특히 식 (1)과 같은 단순선형회귀법은 모형의 단순함으로 인해 많이 이용되었다. 이 때, x_i 는 독립 변수, y_i 는 종속변수, β_0, β_1 는 파라미터, ϵ_i 는 오차항, $i(1 \leq i \leq n)$ 는 특정 관찰값의 인덱스를 의미한다. 오차항의 분산을 줄이기 위해 $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i$ 와 같이 더 높은 차수의 방정식으로 유도하기도 한다.

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \quad (1)$$

주어진 독립변수와 종속변수의 샘플 값들을 이용하여 파라미터 β_0, β_1 을 추정하고, 식 (2)와 같은 선형회귀모형을 구한다.

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i \quad (2)$$

여기에서 모델에 의해 예측된 종속변수의 값 \hat{y}_i 와 실제 종속변수의 값 y_i 의 차를 잔차(residual) $e_i = y_i - \hat{y}_i$ 라고 한다. 이 때 사용되는 대표적 추정방법이 최소제곱(least square) 회귀추정법이라고 하며, 식 (3)과 같이 잔차 제곱의 합(SSE, sum of squared residuals)을 최소화하여 파라미터의 추정치를 구하게 된다.

$$SSE = \sum_{i=0}^N e_i^2 \quad (3)$$

이러한 최소제곱 회귀추정법에서 추정 파라미터

에 대한 공식은 식 (4), 식 (5)와 같고, \bar{x} 는 x 값의 평균, \bar{y} 는 y 값의 평균을 나타낸다.

$$\hat{\beta}_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} \quad (4)$$

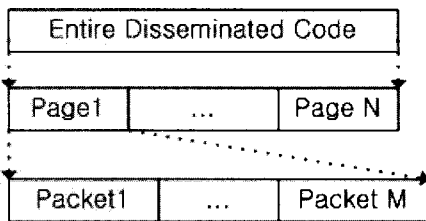
$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \quad (5)$$

3. 효율적인 코드분배 방법

3.1 기본 가정

본 논문에서는 네트워크 내에 센서 노드들의 고른 분포(uniform distribution)를 가정하고, 분배되는 전체 새로운 코드를 보유하고 있는 하나의 싱크노드가 임의의 지역에 위치해 있다고 가정한다. 분배되는 코드는 [그림 1]에서 보는 바와 같이 기존 코드 분배 프로토콜과 동일하게 페이지 단위로 나뉘어 있으며, 각 페이지는 고정 크기의 패킷으로 나뉘어 전달된다[10, 13].

싱크노드는 센서 네트워크의 영역 넓이, 센서 노드의 수와 각 센서 노드의 위치, 자신으로부터 가장 먼 노드로의 홉 수를 알고 있다고 가정한다. 센서 노드의 위치는 미리 계획되어 배치되기 때문에 정확한 정보를 알 수 있으며, 그렇지 않은 경우에도 다양한 로컬라이제이션 알고리즘에 의해서 파악이 가능하다[19].



[그림 1] 분배 코드의 구성

3.2 ECoDi의 기본

본 논문에서 제안하는 ECoDi 기법은 센서 노드

들 중에 코드 전송 시간을 최소화할 수 있는 다수 분배자 노드들을 선정한다. 선정된 분배자 노드까지는 유니캐스트로 빠르게 전송하고, 각 분배자 노드는 자신의 분할 영역 안의 센서노드로 기존 코드 분배 메커니즘에서 사용하는 브로드캐스트를 통해 전송하는 혼용 전송 방법을 사용한다. 단, 네트워크에서 멀티캐스트가 지원되는 경우, 같은 홉수의 분배자 노드들로 유니캐스트 대신 멀티캐스트를 이용하여 전송할 수 있다.

3.2.1 다수 분배자 노드 선택

가장 기본적인 ECoDi 기법을 설명하기 위해, 해당 네트워크를 [그림 2]와 같이 정사각형 모양의 격자 배치 구조와 싱크노드의 중앙 배치를 가정하여 설명한다. 본 ECoDi 기법의 목적은 코드 이미지를 전체 네트워크에 전송하는 시간을 최소화하는 것이다. 그러므로 본 기법에서는 해당 목적을 만족하기 위한 분배자 노드의 수와 그 위치를 결정해 준다. 이러한 기본 ECoDi 기법의 결과물은 식 (6)에 의해 분배자 노드의 홉 수로 간단하게 계산할 수 있다. 식에 사용된 기호는 <표 1>과 같다. 식 (6)은 분배자 노드(IU)까지의 유니캐스트 전송 시간과 이후 나머지 홉($L-IU$)까지의 브로드캐스트 전송 시간의 합을 최소화하는 IU 를 찾게 된다. 이 때, 유니캐스트와 브로드캐스트 전송 시간에 대한 함수는 제 2.2절에서 소개한 회귀분석법을 이용하여 구한다. <표 2>에서는 기본 ECoDi에서 다수 분배자 노드 선택 방법에 대해 단계별로 설명하였다.

$$\begin{aligned} \min(tU \cdot sC \cdot IU + tB \cdot sC \cdot LB) \quad (6) \\ = \min((tU \cdot IU + tB \cdot (L - IU)) \cdot sC) \\ = \min(f(IU) + g(L - IU) \cdot sC) \end{aligned}$$

위와 같이 구해진 최적 유니캐스트 홉 수(IU)를 가지고 영역을 몇 레벨로 나눌지 결정한다. 이를 단순화하기 위해 다음과 같이 단계를 늘려가며 최적 홉 수에 맞는 영역 분할을 수행한다. 예를 들어

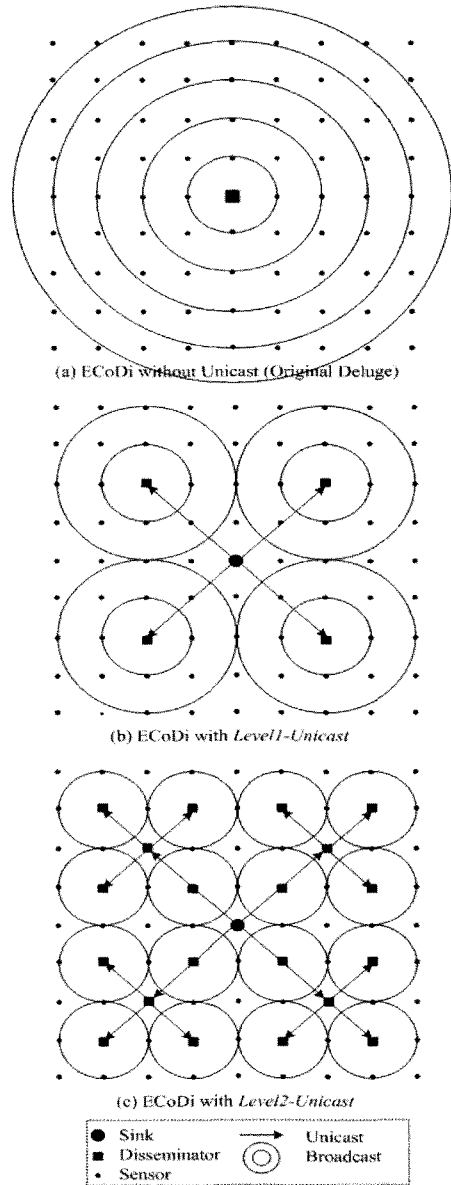
〈표 1〉 최적 유니캐스트 홉 수를 구하기 위한 식 (6)에서 사용되는 기호

기호	설명
tU, tB	한 페이지를 유니캐스트, 브로드캐스트를 통해 한 홉 전송하는데 요구되는 시간
IU, IB	유니캐스트, 브로드캐스트에 의해 전송되는 홉 수
sC	코드 사이즈(예 : C pages)
L	네트워크 길이(싱크노드로부터 가장 먼 노드까지의 최대 홉 수)
$f(x), g(x)$	홉(x) 당 유니캐스트, 브로드캐스트 전송 시간에 대한 함수

〈표 2〉 기본 ECoDi 기법의 알고리즘

Algorithm 1 : Basic ECoDi	
Input :	코드 사이즈(sC)
Output :	다수 분배자 노드 셋
Step 1 :	네트워크 길이(L , 지름 또는 반지름)를 구한다.
Step 2 :	유니캐스트, 브로드캐스트 단위 전송시간(tU, tB)을 구한다.
Step 3 :	홉 수에 따른 유니캐스트/브로드캐스트 함수식 $f(x), g(x)$ 를 식 (1)~식 (5)에 의한 회귀분석법으로 구한다.
Step 4 :	식 (6)을 만족하는 최적의 유니캐스트 홉 수(IU)를 구한다.
Step 5 :	구해진 유니캐스트 홉 수에 따라 네트워크를 개층적인 영역으로 나눈다.
Step 6 :	나누어진 영역의 중앙위치 노드를 분배자 노드로 결정한다.

[그림 2]의 센서 그리드 네트워크는 네트워크 길이가 8인 네트워크이다. [그림 2](a)에서는 IU 가 0으로 나온 경우, 싱크노드로부터 모두 브로드캐스트에 의해 전송되는 예이다. [그림 2](b)는 IU 가 4로 나온 경우(*Level1-Unicast*)로 네트워크를 싱크노드 중심으로 4개의 영역으로 나누어 각 영역 중심의 노드가 분배자 노드로 선정된다. 선정된 분배자 노드까지는 유니캐스트로 전송되고, 각 분배자 노드는 코드의 첫 페이지를 수신한 뒤 바로 자신의 영역으로 브로드캐스트하게 된다. [그림 2](c)는 IU 가 6으로 나온 경우(*Level2-Unicast*)로 [그림 2](b)에서처럼 영역이 4개로 나눈 다음, 다시 각 영역을 4개의 소영역으로 나누면, 소영역의 중앙 노드가



〈그림 2〉 유니캐스트와 브로드캐스트가 혼합된 ECoDi 전송 예

분배자 노드가 된다. 이 때 전송방법은 *Level1-Unicast*의 4개의 분배자 노드까지 유니캐스트하면서 *Level* 값과 하위 분배자 노드 리스트가 전달한다. 그러면 *Level1-Unicast*의 분배자 노드는 하위 분배자 리스트에 있는 자신의 분배자 노드들에게 수신한 페이지를 유니캐스트로 전송한다. 마지막 레

벨인 *Level2-Unicast*의 분배자 노드(각 4개)는 페이지를 분배받자마자 브로드캐스트로 자신의 영역에 코드를 전파하게 된다.

3.2.2 유니캐스트/브로드캐스트 혼용 전송

이러한 전송 방법을 일반화하여 설명하면 결정된 유니캐스트 홉 수(IU)를 가지고 분할 영역을 결정하여 레벨을 결정하고, 분배자 노드 셋을 결정한다. 이 때 싱크노드로부터 먼 곳에 위치한 분배자 노드일수록 큰 레벨값을 갖게 된다. 예를 들어 결정된 레벨 값이 $k(1 \leq k \leq L, L$ 은 싱크노드로부터 가장 먼 센서 노드까지의 홉 수)이라면 1에서 $k-1$ 까지의 레벨 값을 가진 분배자 노드는 이전 분배자 노드로부터 한 페이지를 모두 수신하면 다음 레벨의 분배자 노드로 유니캐스트 혹은 멀티캐스트를 통해 해당 페이지를 전송하고, k 레벨의 마지막 분배자 노드는 기존 코드분배 프로토콜처럼 브로드캐스트를 통해 코드를 분배한다. 그러나 다른 분포 혹은 다양한 영역의 센서 네트워크라도 싱크노드로부터의 홉 수에 기반하여 분배자 노드의 수와 그 위치, 레벨 값을 쉽게 결정할 수 있다. 이러한 적용 방법은 제 3.3절에서 자세히 기술하고자 한다.

3.3 ECoDi의 확장

네트워크 영역이 제 3.1절에서의 가정과 같이 정방형이 아닌 임의 모양이거나 싱크노드가 정중앙이 아닌 임의 지역에 위치해 있는 경우에는 <표 2>의 기본 알고리즘을 확장한 ECoDi 알고리즘(<표 3> 참조)을 통해 분배자 노드 선정을 수행할 수 있다. 이 때 기본 ECoDi 알고리즘에서 추가된 단계는 Step 1이고, 변경된 단계는 Step 7이다. 초기 Step 1에서는 싱크노드를 중심으로 정방형 가상 네트워크를 그리고, 이를 기반으로 네트워크 길이에 따라 최적 유니캐스트 홉 수를 결정하여 가상 네트워크를 분할한다(Step 2~Step 6). 분할된 영역 중 실제 네트워크 영역의 분할 영역에 대해 각 영역의 중앙위치에서 가장 가까운 노드를 분배자 노드로 선정한다.

<표 3> 확장 ECoDi 기법의 알고리즘

Algorithm 2 : Extended ECoDi

Input : 코드 사이즈(sC)

Output : 다수 분배자 노드 셋

- Step 1 : 싱크노드를 중심으로 정방형 가상 네트워크 그린다.
- Step 2 : 네트워크 길이(L , 지름 또는 반지름)를 구한다.
- Step 3 : 유니캐스트, 브로드캐스트 단위 전송시간(tU, tB)을 구한다.
- Step 4 : 홉수에 따른 유니캐스트/브로드캐스트 함수식 $f(x), g(x)$ 를 식 (1)~식 (5)에 의한 회귀분석법으로 구한다.
- Step 5 : 식 (6)을 만족하는 최적의 유니캐스트 홉 수(IU)를 구한다.
- Step 6 : 구해진 유니캐스트 홉 수에 따라 가상 네트워크를 계층적인 영역으로 나눈다.
- Step 7 : 가상 네트워크 영역 중 실제 영역에 해당하는 분할 영역에 대해, 각 영역의 중앙위치에서 가장 가까운 노드를 분배자 노드로 결정한다.

또한 선정된 분배자 노드로의 전송은 기본 ECoDi에서처럼 낮은 레벨에서 높은 레벨로 유니캐스트되고, 각 분배자는 코드의 한 페이지를 수신한 뒤 각 영역에 브로드캐스트를 통해 전송한다. 확장 ECoDi 기법의 적용은 다음 장에서 실제 테스트베드 네트워크에 적용된 예를 통해 더 자세히 설명한다.

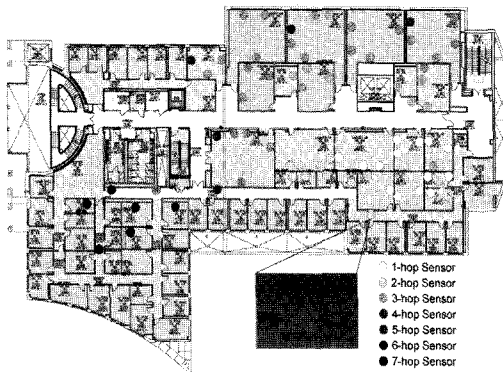
4. 실험 분석

본 논문에서 제안된 ECoDi 기법의 효율성을 입증하기 위하여 실제 환경에 배치된 WiSeNet 테스트베드[23]에서 기존의 대표적인 코드분배 기법 Seluge[10]와 ECoDi 기법을 동시에 구현하여 비교 실험하였다.

4.1 실험 환경

실험에 사용된 WiSeNet 테스트베드는 [그림 3]과 같은 72개의 센서 노드(MicaZ 모드)를 노스캐롤라이나주립대학의 공학관에 설치한 실험용 센서 네트워크이다. 해당 구현된 ECoDi 기법은 전체 네트워크에서 브로드캐스팅에 의한 코드 전파 기법

인 Seluge와 비교 실험하였으며, 전파 시 유니캐스트/브로드캐스트 혼합전파를 사용하는 것 이외의 다른 오퍼레이션은 동일하게 수행된다. 두 기법의 코드분배 시간(새로운 코드가 모든 노드에 전파 완료 되는 시간)을 성능값으로 측정하여 비교하였고, 이 결과들은 15번의 반복 실험을 통해 얻은 평균값을 이용하였다.



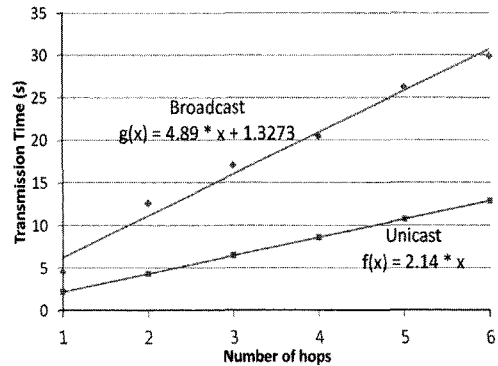
[그림 3] WISENet 테스트베드

4.2 실험 결과

실험 대상인 센서 네트워크의 크기(최대 홉 수가 7)가 작기 때문에 본 실험에서는 싱크노드를 네트워크 코너에 위치시켰다([그림 3]에서 별 기호). 이 구성은 [그림 2]와 같은 정방형 네트워크의 북서측 1/4의 영역과 같다.

<표 3>에 의한 확장 ECoDi 알고리즘에 의해 설명을 하면, 우선 가상 정방형 네트워크를 그린다(Alg. 2 : Step 1). 네트워크 길이 L 은 싱크노드로부터 전체 네트워크로의 브로드캐스트 전파에 의해 구해질 수 있다(Alg. 2 : Step 2). 본 실험 네트워크에서의 L 은 7이며, 싱크노드로부터의 홉 수가 [그림 3]의 노드 명암으로 표기되어 있다. 그림에서 처럼 노드 사이의 거리와 홉 수가 정확히 비례하지 않음을 알 수 있어 실험값에 의한 기법 적용의 중요성을 알 수 있다. 또한 구해진 L 값은 한 번의 실험으로 추후 코드 전파에 계속 사용될 수 있다.

다음으로 홉 수 따른 유니캐스트 및 브로드캐스트 전송 시간에 대한 함수인 $f(x)$ 와 $g(x)$ 를 구하기 위하여 가장 먼 노드로의 유니캐스트 전송을 실험하고, 브로드캐스트 전파 시간은 이전 단계에서 L 값을 구하기 위해 브로드캐스트 전파 시의 값을 사용하면 된다(Alg. 2 : Step 3). 이 유니캐스트 전송 시간 값은 일반적인 전송 방식이므로 전송 시 시간 정보를 측정하여 사용할 수도 있고, 한 번의 실험으로 추후 코드 전파에 계속 사용될 수 있다. 이렇게 측정된 시간을 그림으로 나타낸 것이 [그림 4]이다. 이러한 실험치를 가지고 회귀분석법에 의해 함수식 $f(x)$ 와 $g(x)$ 를 구한다(Alg. 3 : Step 4). 구해진 함수도 [그림 4]에 동시에 나타내었다. 실험 결과, 홉 당 브로드캐스트의 전송 시간이 유니캐스트 시간보다 길고, 그 증가율도 큰 것으로 나타났다. 이는 브로드캐스트 시에는 많은 노드가 함께 전송을 수행하기 때문에 전파 충돌이나 재전송 등의 수행에 의한 지연 때문인 것으로 분석된다.

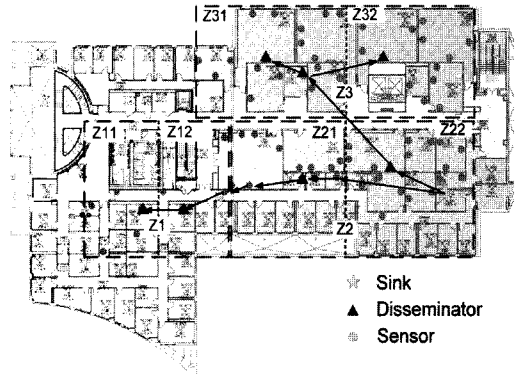


[그림 4] 측정된 한 페이지 유니캐스트 및 브로드캐스트 전송 시간과 회귀분석법에 의해 추정된 $f(x)$ 및 $g(x)$

이렇게 구해진 $f(x)$ 와 $g(x)$ 에 의해 유니캐스트 및 브로드캐스트를 혼용하여 최소의 전파 시간을 요구하는 최적의 유니캐스트 홉 수를 구한다(Alg. 2 : Step 5). <표 4>에 보는 바와 같이 $f(x)$ 와 $g(x)$ 값에 의해 최소 전파 시간을 요구하는 최적의 유니캐스트 홉 수는 6으로 결정되었다.

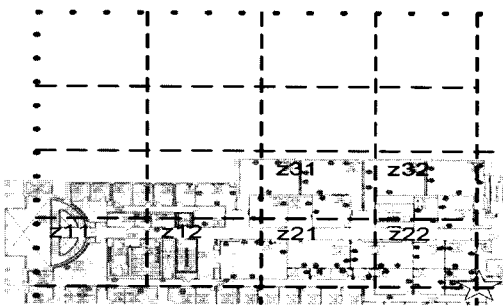
〈표 4〉 $f(x)$, $g(x)$, 식 (6)에 의해 결정된 최적의 유니캐스트 홉 수

홉 수 (x)	유니캐스트 시간 : $f(x)$	브로드캐스트 시간 : $g(x)$	식 (6)에 의한 시간
1	2.14	6.22	$f(1)+g(6) = 32.84$
2	4.29	11.12	$f(2)+g(5) = 20.31$
3	6.44	16.01	$f(3)+g(4) = 27.35$
4	8.59	20.91	$f(4)+g(3) = 24.60$
5	10.73	25.80	$f(5)+g(2) = 21.85$
6	12.88	30.69	$f(6)+g(1) = 19.11$



[그림 6] 결정된 Level3-Unicast 영역과 분배자 노드

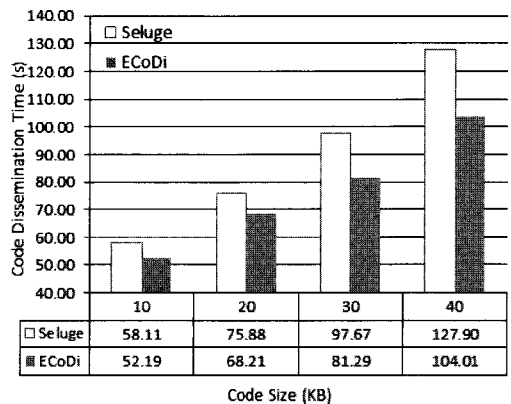
결정된 유니캐스트 홉 수인 6에 따라 가상 네트워크 영역을 분할해 간다. [그림 5]는 Step 1에서 싱크노드를 중심으로 형성한 가상 네트워크의 1/4 복서측 영역이다. 이를 홉 수 6에 맞추어 분할하면 Level3-Unicast이 되어 1/4가상 네트워크가 16개의 소영역으로 분할된다(Alg. 2 : Step 6). 실제 센서 네트워크 영역에 매핑되는 영역은 Z11에서 Z32까지 6개의 영역으로 각 영역의 중앙위치에 가까운 노드가 분배자 노드로 선정된다(Alg. 2 : Step 7). 선정된 분배자 노드들과 해당 노드들의 유니캐스트 전송 루트는 [그림 6]과 같다.



[그림 5] 싱크노드를 중심으로 형성된 가상 네트워크의 1/4 영역의 분할 내용

이렇게 결정된 영역과 각 영역의 분배자 노드의 유니캐스트 전송은 [그림 6]에서 보이고 있다. 그림에서와 같이 분배자 노드로 유니캐스트 전송이 되면, 바로 해당 분배자 노드는 자신의 영역내 브로드캐스트 전송을 통해 코드를 전파하게 된다.

지금까지 설명한 ECoDi 기법의 유니캐스트/브로드캐스트 혼합 전파에 의한 코드 전송 시간과 브로드캐스트만 사용하는 기존 기법 Seluge의 전송 시간을 측정하여 [그림 7]에 나타내었다. 결과, 기존 Seluge에 비하여 ECoDi 기법이 최대 19%까지 전파 시간을 단축시킬 수 있었다. 이 성능 향상 정도는 코드 사이즈가 커질수록 더 커짐을 보이고 있다. 이는 향후 센서 노드 기술이 발전함에 따라 현재 MicaZ 센서 노드의 코드 저장을 위한 메모리가 128KB에서 더 커질 것이며, 실행되는 코드 또한 응용이 발전함에 따라 커질 것으로 예상되어, ECoDi 기법에 의한 성능 향상 정도는 더욱 커질 것으로 예상된다.



[그림 7] 코드분배 시간

5. 결 론

본 논문에서는 다수의 무선 노드로 구성된 센서 네트워크에서 실행되는 코드의 업그레이드나 버그 수정을 위해 효율적인 코드분배를 수행하는 ECoDi 기법을 제안하였다. ECoDi 기법에서는 낮은 지연성을 제공하는 코드분배를 수행하기 위하여 적절한 다수 분배자를 선정하여 유니캐스트와 브로드캐스트를 혼용하여 전송에 사용하였다. 해당 적정 다수 분배자 수 및 위치는 네트워크의 길이, 홉 당 유니캐스트와 브로드캐스트 시간, 코드 길이 등의 정보에 따라 회귀분석법을 이용하여 결정하였다. 실제 건물에 배치된 테스트베드 실험을 통해 제안한 ECoDi 기법이 기존 브로드캐스트를 사용하는 Seluge 기법보다 최고 19% 정도의 낮은 지연을 제공하는 빠른 코드분배 방법임을 입증할 수 있었다. 이러한 코드분배의 낮은 지연성은 실제 센서 네트워크에서 짧아진 소프트웨어 업그레이드 시간을 통해 안정된 센서 네트워크 서비스 제공을 가능하게 할 것이다.

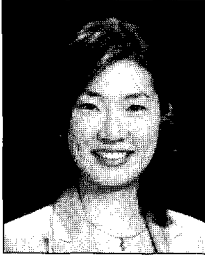
본 논문의 결과는 72개의 노드로 구성된 네트워크에서의 실험 결과이나, 향후 더 많은 노드로 구성된 네트워크 또는 다양한 형태(예 : 밀도, 토폴로지 등)의 네트워크에서 실험 또는 시뮬레이션하여 그 성능의 향상 정도를 비교하고자 한다.

참 고 문 헌

- [1] 김미희, 김지선, 김지현, 임지영, 채기준, "센서 네트워크에서 코드분배 메커니즘에 대한 조사 연구", 「한국정보처리학회논문지C」, 제16권, 제4호(2009), pp.423-438.
- [2] Brwon, S., "Updating Software in Wireless Sensor Networks : A Survey", Dept. of Computer Science, National University of Ireland, Maynooth, Tech. Rep., 2006.
- [3] Dunkels, Adam, Niclas Finne, Joakim Eriksson and Thiemo Voigt, "Run-Time Dynamic Linking for Reprogramming Wireless Sensor Networks", Embedded Networked Sensor Systems (SenSys), (2006), pp.15-28.
- [4] Egan, D., "The emergence of zigbee in building automation and industrial control", *Computing and Control Engineering Journal*, Vol.16, No.2(2005), pp.14-19.
- [5] Golberg, M. A. and H. A. Cho, *Introduction to Regression Analysis*. WIT Press, 2003.
- [6] Han, H.-S., D.-O. Kim, and R.-H. Park, "Structural information-based image quality assessment using LU factorization", *IEEE Trans. Consumer Electronics*, Vol.55(2009), pp.165-171.
- [7] Han, K.-Y., S.-W. Lee, J.-S. Lim, and K.-M. Sung, "Channel estimation for ofdm with fast fading channels by modified kalman filter", *IEEE Trans. Consumer Electronics*, Vol.50(2004), pp.443-449.
- [8] Hui, J. W. and D. Culler. "The dynamic behavior of a data dissemination protocol for network programming at scale", SenSys, Baltimore, Maryland, USA, 2004.
- [9] Hyun, S. and M. Kim, "An Efficient Code Dissemination Mechanism for Building Automation and Industrial Control Networks", ACM ICUIMC, Seoul, Korea, 2011.
- [10] Hyun, S., P. Ning, A. Liu, and W. Du, "Seluge : Secure and dos-resistant code dissemination in wireless sensor networks", *Information Processing in Sensor Networks (IPSN)*, (2008), pp.445-456.
- [11] Jeong, J. and D. Culler, "Incremental network programming for wireless sensors", *IEEE SECON*, (2004), pp.25-33.
- [12] Krasniewsk, M. D., R. K. Panta, S. Bagchi, C.-L. Yang, and W. J. Chappel, "Energy-

- efficient, on-demand reprogramming of large-scale sensor networks”, *ACM Trans. Sensor Networks(TOSN)*, Vol.4, No.2(2008).
- [13] Lanigan, P. E. and P. Narasimhan, “Sluice : Secure dissemination of code updates in sensor networks”, *IEEE ICDCS*, (2006) pp. 53-63.
- [14] Levis, P. and D. Culler, “The firecracker protocol”, *ACM SIGOPS Euro, Wksp.*, 2004.
- [15] Li, W., Y. Zhang, and B. Childers, “MCP : An energy-efficient code distribution protocol for multi-application WSNs”, 5th IEEE International Conference on Distributed Computing in Sensor Systems(DCOSS’09), Marina del Rey, CA, United States, (2009), pp.259-272.
- [16] Li, W., Y. Zhang, J. Yang, and J. Zheng, “UCC : Update-Conscious Compilation for Energy Efficiency in Wireless Sensor Networks”, *ACM SIGPLAN Conference on Programming Language Design and Implementation(PLDI)*, San Diego, California, Vol.42, No.6(2007), pp.383-393.
- [17] Marron, P. J., M. Gauger, A. Lachenmann, D. Minder, O. Saukh, and K. Rothermel, “Flexcup : A flexible and efficient code update mechanism for sensor networks”, *European Workshop on Wireless Sensor Networks(EWSN)*, (2006), pp.212-227.
- [18] Panta, R. K., I. Khalil, and S. Bagchi, “Stream : Low overhead wireless reprogramming for sensor networks”, *IEEE INFOCOM*, (2007), pp.928-936.
- [19] Patwari, N. and A. O. H. III, “Using proximity and quantized rss for sensor localization in wireless location in wireless network”, *In ACM international conference on Wireless sensor networks and applications*, (2003), pp.20-29.
- [20] Rossi, M., G. Zanca, L. Stabellini, R. Crepaldi, A. F. H. III, and M. Zorzi, “Synapse : A network reprogramming protocol for wireless sensor networks using fountain codes”, *IEEE SECON*, (2008) pp.188-196.
- [21] Tang, W., W. Kuang, B. Wang, and J. Yang, “E3NP : An Energy-Efficient, Expeditious Network Reprogramming Mechanism in Wireless Sensor Network”, *Embedded Software and Systems(ICESS)*, (2008) pp.516-523.
- [22] Willig, A., “Recent and emerging topics in wireless industrial communications : A selection”, *IEEE Trans. Industrial Informatics*, Vol.4, No.2(2008), pp.102-124.
- [23] WiSeNet, <http://netsrv.csc.ncsu.edu/twiki/bin/view/Main/Facilities.html>.

◆ 저 자 소 개 ◆



김 미 희 (mhkim@hknu.ac.kr)

현재 국립한경대학교 컴퓨터공학과 전임강사로 재직 중이며, 이화여자대학교 컴퓨터학과를 졸업하고 동 대학원 컴퓨터학과에서 석사학위와 박사학위를 취득하였다. Elsevier's Computer Communications, IEEE Transactions on Consumer Electronics, 한국정보처리학회논문지, 한국인터넷정보학회논문지 등의 국내외 학술지에 논문을 게재하였다. 주요 관심분야는 센서 네트워크, 인지 라디오 네트워크, 프로토콜 설계 및 보안 등이다.



홍 준 석 (junehong@kyonggi.ac.kr)

현재 경기대학교 경영정보학과 부교수로 재직 중이며, 서울대학교 경영학과를 졸업하고 한국과학기술원(KAIST) 경영과학과에서 석사학위를, 테크노경영대학원에서 박사학위를 취득하였다. International Journal of Information Technology and Decision Making, Expert Systems with Applications 등의 국제학술지 및 한국IT서비스학회지, 한국전자거래학회지, 지능정보연구 등의 국내학술지에 논문을 게재하였다. 주요 관심분야는 시맨틱 웹, 전자상거래 솔루션, 지능형 에이전트 등이다.