

# 유비쿼터스 센서 네트워크에서 연관된 데이터의 효율적인 처리방안

## Efficient Processing Scheme for Correlated Data in Ubiquitous Sensor Networks

류 제택\*, 허남호, 유승화, 김기형

(Jea-Tek Ryu, Nam-Ho Heo, Seung-Wha Yoo, and Ki-Hyung Kim)

**Abstract :** In now days, Ubiquitous technology grow up, so the variety service are developed. Sensor networks' purpose is collection information about environment and geographic. But sensor network has limit in power, cost and so on. There is much restriction. Some sensor networks purpose is monitoring environment. And there is some relation in sensing data. Sensor nodes sense information by periods. First sensing data correlate with neksensing data. At this point, this paper suggest power saving method. Some data are same, the other data are similar.

**Keywords:** Data fusion, Data skipping

### I. 서론

최근 무선 네트워크 기술의 발전에 따라 여러 분야에서 센서네트워크를 응용하여 여러 가지 서비스를 제공하려 하고 있다. 센서네트워크는 특성상 저 전력 기반으로 설계되어 있기 때문에 여러 서비스를 제공하는데 있어 상당한 제약이 따르고 또한 그 응용프로그램을 개발하는데 있어서 또한 많은 어려움이 있다. 하지만 센서네트워크의 특성상 저비용으로 구성할 수 있기 때문에 다양한 정보를 센싱하여 모니터링하는 네트워크등에 다양한 서비스들을 제공하고 있으며, 이를 통한 유비쿼터스 도시 등을 통해 실생활에 더욱 가까이 다가오고 있다. 기술적인 발전으로 점점 저비용으로 노드를 제작할 수 있어지면서 다양한 시도가 이루어지고 있지만 센서네트워크의 특성은 크게 변하지 않기 때문에 항상 전력소모와 저전력 저 비용에 대한 고려가 필요하다. 또한 서비스에 따라 6loWPAN과 Zigbee 등의 플랫폼에 대한 고려역시 필요하며 이러한 플랫폼에 따라 다양한 방식의 전력제어기술 네트워크 효율성증가의 방법 또한 달라진다.

본 논문에서는 센서네트워크의 효율성을 향상시키기 위하여 주기적인 정보를 수집하는 네트워크에 초점을 맞추어 데이터의 시간적인 연관관계를 분석하는 방법을 사용한다. 일반적으로 아무런 데이터 퓨전(Data Fusion)을 사용하지 않은 네트워크와 자신의 데이터를 코딩하여 네트워크 트래픽을 감소시키는 자기 트래픽 코딩(Self Coding), 자신이 포워딩하는 데이터를 코딩하는 외부 트래픽 코딩(Foreign Coding)등을 사용하는 네트워크 그리고 현재 논문에서 제안하는 시간관계를 분석하여 네트워크의 데이터 효율성을 높이는 데이터 스킵핑(Data Skipping)을 사용하는 센서네트워크를 시물레이션 한다.

이 논문의 구성은 다음과 같다. 제 2장에서는 현재 제안된 데이터퓨전관련 연구들에 대해서 알아보고, 3장에서는 본 논문에서 제안하는 데이터생략에 대해서 알아본다. 4장에서는 본 논문에서 사용한 데이터 퓨전에 대한 네트워크 모델에 대한 내용을 정리한다. 5장에서는 시물레이션을 통해 네트워크 효율성을 검증한다. 6장에서는 본 논문의 결론에 대해 정리한다.

문에서 제안하는 데이터생략에 대해서 알아본다. 4장에서는 본 논문에서 사용한 데이터 퓨전에 대한 네트워크 모델에 대한 내용을 정리한다. 5장에서는 시물레이션을 통해 네트워크 효율성을 검증한다. 6장에서는 본 논문의 결론에 대해 정리한다.

### II. 데이터 퓨전(Data Fusion)

센서네트워크는 구성노드 하나하나가 독립된 배터리로 작동된다. 하지만 그 네트워크를 관리하는 코디네이터(Coordinator)는 전원이 지속적으로 공급되는 형태를 가지는 것이 보편적인 네트워크의 구성이다. 하지만 이는 보통의 센서네트워크이며 특별한 네트워크에서는 코디네이터역시 배터리 전원으로 작동된다. 배터리를 통해 전원을 공급받는 이러한 노드들을 통해 네트워크를 구성하기 때문에 센서네트워크에서는 전체 네트워크의 전력제어가 필요하다. 데이터 퓨전은 이러한 네트워크에서 발생하는 트래픽을 경감시키기 위한 방법 중에 하나이다.

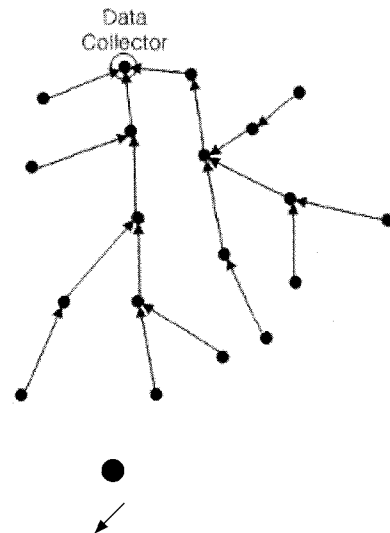


그림 1. 모니터링 센서 네트워크

\* 책임저자(Corresponding Author)

논문접수 : 2008xx      채택확정 : 2008xx

류제택, 허남호 : 아주대학교 정보통신전문대학원

([ricman@u.ac.kr](mailto:ricman@u.ac.kr)), ([jdorak@u.ac.kr](mailto:jdorak@u.ac.kr))

※ 본 연구는 교육부 학술연구조성비의 지원을 받아 연구되었음.

그림 1에서는 역 멀티캐스트 라우팅 경로(Reverse Multicast Routing Root)를 통해 데이터수집노드에게 데이터가 전송되는 모니터링 센서네트워크를 나타낸 그림이다. 각각의 점은 센서노드를 의미한다. 이 라우팅 경로는 퓨전에서 중요한 의미를 가진다. 그림의 화살표로 센서의 데이터가 전송되며 이때 전송되는 데이터의 센싱데이터퓨전을 실행하여 트래픽을 절감시킨다. 데이터퓨전은 각각의 개별적인 데이터간의 연관성을 토대로 해당데이터를 압축하는 방법이다.

이때 물리적으로 가까운 위치에 있는 노드의 센서데이터의 연관관계를 통해서 전체 네트워크에 전송되는 데이터의 크기를 줄이는 방법과 각각의 노드가 포워딩할 데이터를 코딩하는 방법등이 있다. 또한 TinyDB[2]와 Congar projects[3]에서 사용하는 쿼리 처리(Query Processing) 방식은 평균, 합, 최대값, 최소값등의 SQL 언어와 유사한 방식의 쿼리 처리를 통해 전체 네트워크에서의 효율성을 향상시키는 방법이 있다.

일반적인 센서네트워크는 주로 해당지역의 정보를 수집하고 그 정보를 수집하는 형태를 가진다. 따라서 각각의 센서노드들은 정보를 주기적으로 센싱하며 그 센싱한 데이터를 수집노드에게 전송한다. 이러한 형태의 모니터링 네트워크에서는 네트워크안의 대부분의 패킷들의 목적지 주소가 데이터를 수집하는 노드가 되며 해당 트래픽이 주기적으로 센싱되어 발생하는 특징을 가지게 된다. 또한 지리적으로 근접한 노드들의 온도, 습도 등 일반적인 센싱 데이터가 연관성 가지게 된다. 이런 연관성을 가진 데이터를 통해 데이터를 코딩하는 것을 Data Aggregation 이라 한다. 센서노드는 이런 Data Aggregation을 통해 수집노드로 데이터를 전송하기 전에 데이터를 코딩하여 전송한다. 이런 Data Aggregation에는 크게 자신의 데이터를 코딩하는 Self-Coding과 자신을 통해 포워딩되는 데이터를 코딩하는 Foreign-Coding이 있다. [4] 하지만 이 논문에서는 전체 데이터를 전송하는 것이 아닌 데이터의 시간에 따른 관계를 파악하여 동일한 데이터의 전송을 노드 간의 약속에 따라 데이터를 생략할 수 있는 방법을 제안한다.

### III. 데이터 스킵핑(Data Skipping)

#### 1. 데이터 스킵핑(Data Skipping)

센서노드가 주기적으로 센싱하는 데이터의 값을 확인하며 그 데이터가 이전에 센싱한 데이터와 일정한 오차범위 이내에 있다면 그 값을 전송하지 않도록 하는 방법이다. 네트워크의 모든 노드들은 주기적으로 데이터를 수집하는 노드에게 트래픽을 발생 시킨다. 이때 센서노드가 센싱하는 주기가 대단히 길지 않다면 그 센싱 데이터 값은 이전의 데이터 값과 큰 차이를 나타내지 않는다. 간단하게 데이터를 센싱하는 다음과 같은 역 멀티캐스트 경로 네트워크를 통해 설명한다.

A-G까지의 7개의 노드로 구성된 센서네트워크이다. 데이터를 수집하는 노드는 A이며 B-G의 노드는 주기적으로 각각의 센싱데이터를 A로 전송하며 전송하는 라우팅 경로는 다음과 같은 실선으로 표시한다. 1홉의 거리에 위치한 B노드는 다음과 같은 트래픽을 발생시킨다. 실제 데이터가 전송되었을 경우 실선으로 표시하였으며 DataSkipping 이 이루어진 트래픽은 점선을 포함하여 표시하였다. 다음 예시의 데이터 변동 바운드값은 0.2로 가정하였다. 일반적인 경우의 바운드값은

$$\frac{(\max(\text{센싱 데이터}) + \min(\text{센싱 데이터}))}{100} \quad (1)$$

100

(1)

으로 계산한다. 두 번째 트래픽은 처음트래픽과의 차이가 0.2이기 때문에 바로 생략과정없이 전송되었다. 하지만 세 번째 트래픽은 0.1의 차이가 발생하였기 때문에 전송되지않고 데이터가 생략되었다.

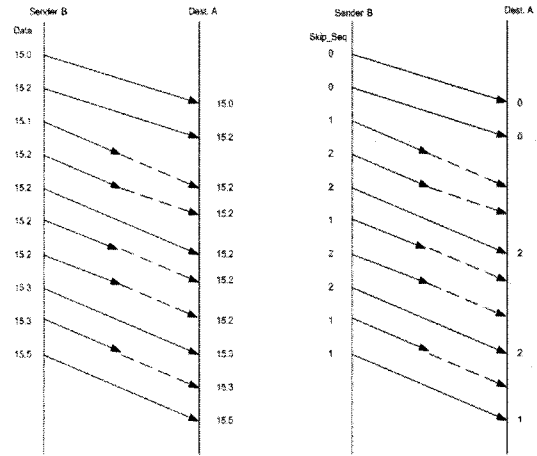


그림 2.1홉 거리에서의 데이터 스킵핑 데이터 흐름

하지만 모든 트래픽이 같다고 해서 모든 트래픽을 생략한다면 해당 노드가 동작하고 있는지 그렇지 않은지 확인할 수 없기 때문에 최대 대기횟수를 설정할 수 있다. 위의 그림에서는 2번의 생략 이후엔 전송의 방법을 통해서 데이터를 전송하였다. 오른쪽의 SkipSeq 는 해당 트래픽이 몇 번의 생략 이후에 전송되는 데이터인지 확인하는 값이다. 해당 데이터가 생략될 때마다 SkipSeq 를 하나씩 올리며 해당 값이 전송될 때 목적지에서 해당 데이터가 몇 번이 생략되었는지 확인할 수 있도록 한다. 위의 경우 총 10회의 트래픽 중 5회가 생략되었다. 하지만 이를 1홉에 있는 노드가 아닌 2홉 위치에 있는 D노드의 위치에서 살펴본다.

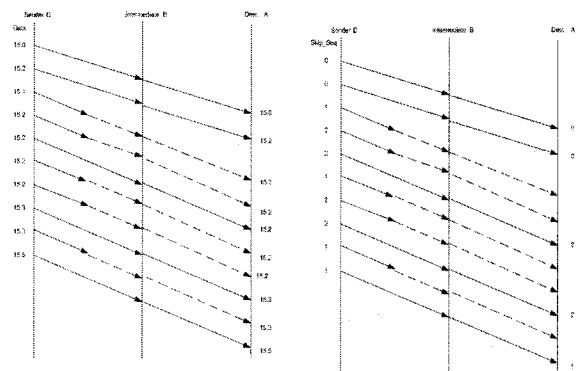


그림 3.2홉거리의 데이터 스킵핑 데이터 흐름

D노드에서는 A까지 2홉을 걸쳐서 트래픽이 전송된다. DataSkipping의 특징은 여러 홉을 거쳐서 트래픽이 전송된다면 그 데이터가 포워딩되는데 중간노드들 역시 포워딩을 하지 않기 때문에 중간노드들의 전력소모 역시 막을 수 있다는 점이다.

하지만 이러한 알고리즘을 적용하기 위해서는 몇 가지 미리 정의되어야 하는 것이 있다. 데이터를 수집하는 노드인 A는 각각의 노드의 트래픽을 1회 저장하고 있어야 하며 또한 해당 노드에서 발생한 트래픽이 오류로 인해 소실되었던지 노드가 작동을 정지하였는지의 상황에 대처하기 위해 수집 노드 A는 타이머로 해당 노드의 상태를 체크해야 한다.

2. 데이터 스키핑의 데이터 수집노드

데이터를 수집하는 노드는 데이터 스키핑을 적용한 네트워크에서 많은 연산을 하게 된다. 만약 데이터를 수집하는 노드가 전원이 들어오지 않는 배터리로 작동한다면 본 논문의 데이터 퓨전 방법을 적용한다면 가장먼저 배터리가 소모되어 전체 네트워크가 작동이 불가능해진다. 하지만 반대로 FFD(Full Function Device)로 가정한다면 센서네트워크의 효율성이 커진다.

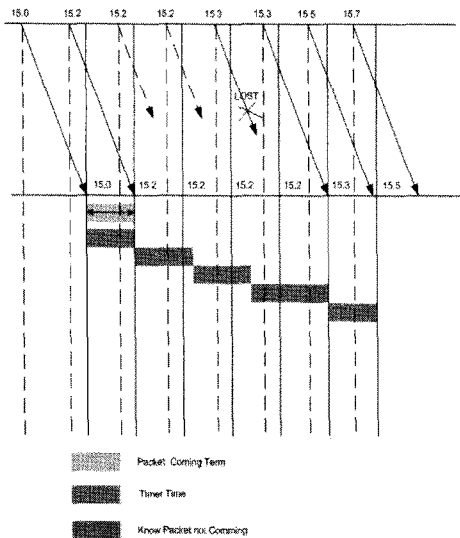


그림 4. 데이터 수집 노드의 타이머 작동

처음 패킷이 도착하면 타이머를 세팅하여 해당 노드에서 패킷이 도착하는 주기를 세팅한다. 이때 타이머는 패킷이 도착을 하면 리셋을 하게 된다. 하지만 데이터가 생략되어 전송이 되지 않았을 경우에는 타이머가 종료되면 해당 데이터가 생략되었다는 것은 인지한다. 이때 타이머는 정확하게 해당 패킷의 발생주기로 한다면 딜레이에 의해 지체 되는 경우에 생략된 데이터로 인지하게 되는 문제가 있기 때문에 최대 패킷 생략 가능 개수(예시에서는 2)를 감안하여 다음의 딜레이 계산방식을 적용한다.

$$\text{타이머 값} = \left[ \frac{1}{\text{최대생략가능트래픽의수} + 1} \right] + [\text{센서노드에서패킷이도착하는주기}] \quad (2)$$

해당 예시에서는 패킷이 도착하는 주기의 1.3배의 타이머가 세팅된다. 두 번의 생략 가능한 트래픽에 해당하는 타이머가 종료되고 패킷을 기다리게 된다. 해당 하는 패킷이 도착하지 않는다면 해당 구간에서 타이머가 종료되고 나서 노드가 살아 있는지 확인하는 메시지를 해당 노드에게 전송한다. 그리고 다음 패킷이 도착하면 다시 타이머를 리셋 하여 이전의 동작방식으로 동작하게 된다.

이러한 타이머를 관리하는 목적지 노드는 전력소비역시 커지며 모든 노드에 대한 타이머의 계산이 필요해진다. 하지만 목적지 노드는 센서노드와는 달리 전력이 안정적으로 공급되는 형태의 장치인 경우가 대다수이기 때문에 전체 네트워크의 전력소비를 줄이기 위해서는 위와 같은 방법이 도움이 된다.

하지만 데이터의 송신 값과 수신 값이 완전히 일치하지 않는다. 이는 중복패킷의 패킷 전송 비운드 값을 0으로 세팅하였을 경우 완전히 동일한 센싱 값일 경우에만 작동하는 방법이 있다.

3.3 네트워크 모델

일반적으로 센서네트워크는 그래프  $G=(V, E)$  로 나타낸다. V는 센서네트워크의 노드들의 집합이며 E는 해당 센서들 간의 링크들의 집합을 의미한다. V에 속하는 집합 S에 속하는  $(S \subset V)$  k개의 노드들은 데이터를 수집하는 노드  $(t \in V)$  향해 데이터를 주기적으로 전송한다.  $e=(u,v)$  는 센싱하여 데이터를 전송하는 노드는 u 이 노드에서 전송된 데이터의 목적지 노드를 v라 할 때의 경로를 의미한다.  $w(e)$ 는 이러한 경로에 매 라운드의 전송에 필요한 비용을 나타낸다.

링크e를 통해 데이터를 전송하는데 드는 비용  $t(e)$  는  $w(e)$ 를 통해 전송된다. 여기서의  $\alpha(u,v)$  이다.)[6]

$$t(e) = w(e)\alpha(e) \quad (3)$$

와 같이 나타낼 수 있으며 여기서의  $\alpha(e)$ 는 한 유닛의 데이터를 e링크를 통해 전송하는데 소모되는 에너지양을 의미한다.

$q(\cdot)$  데이터퓨전에 소모되는 에너지양은  $f(e)$ 로 나타낸다. 목적지노드까지 전송되면서 해당 데이터를 퓨전을 수행하는데 소모된 노드들의 에너지 값으로 나타낸다. 한 유닛을 퓨전 하는데 드는 에너지양을  $q(e)$ 라고 한다. 이때 데이터 퓨전된 데이터들은 중간노드들에서 자신의 트래픽과 자기가 포워딩하는 트래픽을 다시 퓨전을 하게 된다. u와 v노드간의 v노드에서의 데이터 퓨전 비용은 로 나타낸다.

$\tilde{w}(\cdot)$ 는 현재 데이터 퓨전이후의 비용을 의미한다. 그러면  $u$ 와  $v$ 노드의 데이터퓨전 비용을 (4) 같이 나타낼 수 있다.

$$f(e) = q((w(u) + \tilde{w}(v))) \quad (4)$$

$r_{u,v}$ 이때 데이터 퓨전을 통해 데이터 자체의 크기가 줄어들기 때문에 이때 줄어드는 데이터의 크기에 대한 고려가 필요하다. 노드  $u,v$ 를 통해 데이터를 전송할 때 데이터 퓨전을 통해 줄어드는 데이터의 크기를 이라 한다. 그러면 위의 데이터 퓨전 양에 이를 적용하면 (5)와 같이 나타낼 수 있다.

$$w(v) = (w(u) + \tilde{w}(v))(1 - r_{u,v}) \quad (5)$$

$S_{u,v}$  하지만 이때 데이터 스키핑이 적용된 네트워크라면 스키핑으로 인해 절약되는 트래픽의 확률을 이라 가정한다면 (6)과 같이 나타낼 수 있다.

$$w(v) = w(v)(1 - S_{u,v}) \quad (6)$$

이를 통해 전체 네트워크에 전송되는 데이터양을 유추할 수 있다. 노드들의 집합  $S$ 와 데이터를 수집하는 싱크노드  $t$ 가 있다.  $S$ 에 속하는 모든 노드들은 싱크노드  $t$ 를 향해 각 라운드마다 트래픽을 생성한다. 해당 네트워크의 최소한의 데이터 전송량을 가지는 값을 찾는것이 본논문의 목표이므로 각각의 서브그래프로 나누어 본다

$G = (V, E) \subseteq G \subseteq V \subseteq t \subseteq V$  를 만족하는 서브그래프를 생성할 수 있다. 이때, 아래와 같이 네트워크의 에너지소모량을 계산할 수 있다.

$$\sum_{e \in E} (f(e) + t(e)) \quad (7)$$

$S_{u,v}$  본 논문의 목표는 을 극대화 하여 발생하는 트래픽을 최소화 하여 전체 네트워크의 효율성을 향상시킨다. 다음 5장의 시뮬레이션을 통해 위에 네트워크 에너지 모델을 실험한다.

#### IV. 실험 결과

시뮬레이션은 다음과 같은 값을 기준으로 시뮬레이션 되었다. 각 노드는 멀티 홉으로 세팅되었다. 각각의 실험은 50회씩 실험된 평균값을 사용하였다. 각각의 노드의 전송 거리는 20m로 세팅하였다. 또한 uniform위치에 노드를 배열할 때 전송거리 때문에 연결을 맺지 못하는 노드가 없도록 조정하였다. 따라서 세팅 값에 있는 노드의 개수는 실제 데이터를 센싱하여 트래픽을 만들어내는 노드이다.  $r_{u,v}$ 패킷을 생성하는 주기는 10초주기로 생성하였으며 각각의 센서데이터의 크기는 2byte이다. 전체 패킷은 40byte의 IEEE 헤더와 8byte의

UDP헤더를 포함하여 50byte의 패킷을 생성하여 1시간동안의 네트워크에 부가되는 트래픽에 대한 실험이다.

표 1. 시뮬레이션 설정 값

시뮬레이션	Qualnet 4.5
에너지모델	User
TX, RX cost	40nJ/bit
전송거리	40m
$r_{u,v}$	0.5
fusion cost	1 ~ 40nJ/bit
라디오타입	802.15.4 Radio
MAC프로토콜	802.15.4
노드의 배치	1000m x 1000m, Uniform
노드 개수	10 ~ 100 개
실험횟수	50회
시뮬레이션 시간	3600s

해당실험에서는, 즉 데이터퓨전으로 얻어지는 데이터 압축률은 50%로 가정하였으며, 이는 전체 패킷에 대해서 적용하였다. 또한 DataSkipping의 경우는 다음센싱 데이터가 전송을 실행하지 않는 범위 안에 존재할 확률을 50%로 세팅하여 실험하였다. None-Coding은 대조군으로 설정하였으며 이 값은 데이터 값뿐만 아니라 라우팅관련 패킷 등 전체 패킷들의 총합으로 세로축의 단위는 Kbyte이다. 코딩하지 않은 대조군과 비교하여 10개의 노드일 경우 큰 차이를 보이지 않는다. 하지만 노드의 개수가 늘어남으로 인해 데이터를 수집하는 노드와 노드들의 홉 수가 점차 증가하게 되어 네트워크에 패킷의 수가 급격히 증가하게 된다. 자기트래픽 코딩과 외부트래픽 코딩은 50노드의 실험에서 거의 유사한 값을 가지게 되지만 코딩을 수행하는 데이터퓨전 노드의 수는 50%로 세팅하였다. 모든 노드에서 코딩을 하게 된다면 데이터 트래픽은 확실히 줄어들지만 데이터퓨전에 따른 노드의 전력낭비가 커지기 때문에 50%이상의 값은 특수한 상황이 아니라면 적절하지 않다. 노드의 수가 100개로 더 많아짐으로 포워딩되는 데이터의 크기가 커져 외부트래픽 코딩을 이용하였을 때 더 좋은 결과를 보여준다. DataSkipping은 10초간의 트래픽 생성주기로 인해 스킵 되는 패킷의 수가 많기 때문에 가장 좋은 효율을 보여준다.

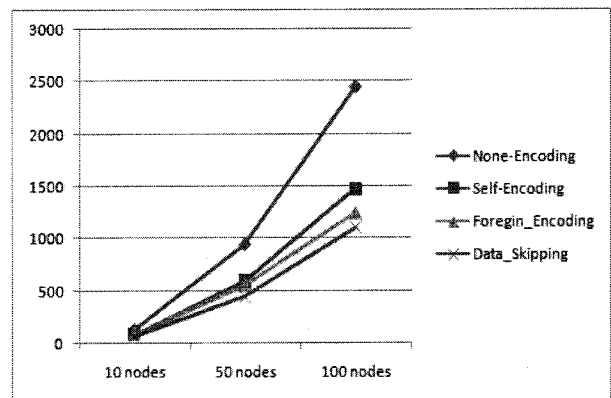


그림 5 전체 네트워크 트래픽 양

데이터 퓨전을 위해 드는 비용을 증가시켰을 때 네트워크가 소모하는 에너지양의 변화를 나타낸 그래프이다. X축은 1 ~ 40nJ/bit 의 에너지 단위를 의미하며 세로의 값은 J이다. 100개의 노드에서 10초마다 트래픽을 발생시켰으며 1시간을 시뮬레이션 하였다. NoneCoding 은 대조군으로 설정하였으며 퓨전관련을 전혀 수행하지 않기 때문에 퓨전에 소모되는 에너지양은 0J이다. 위의 데이터 퓨전에 관련된 함수가 작동할 때 카운트 한 횟수를 기준으로 노드의 계산에 따른 에너지 소모량을 나타낸다. 자기트래픽 코딩의 경우는 자신이 발생하는 트래픽을 코딩할 때 데이터퓨전을 사용하며 외부트래픽 코딩은 자신이 포워딩할 트래픽을 코딩할 때 데이터퓨전을 사용한다. DataSkipping 의 경우, 자신이 각각 센싱데이터를 저장하고 비교할 때 마다 각각 데이터퓨전을 하게 된다. 또한 DataSkipping 은 각각의 패킷이 수집노드에 도착할 때 마다 데이터퓨전관련 함수를 실행하기 때문에 총 에너지 소모량이 월등히 높다. 또한 수집노드의 경우 DataSkipping 은 항상 타이머를 체크하고 해당 패킷마다 도착시간을 계산하기 때문에 엄청난 부하를 가지게 되지만 반대로 단순 센서노드들은 이전 센싱 값과 현재 센싱 값을 비교만 하기 때문에 전체 데이터퓨전에 드는 비용의 계산 현실 값과 차이가 생긴다. 따라서 FFD(Full Function Device)에서의 함수의 비용을 제외한다면 다음과 같은 결과를 가지게 된다. 또한 이는 해당 알고리즘을 수행하는 오버헤드로 볼 수 있다.

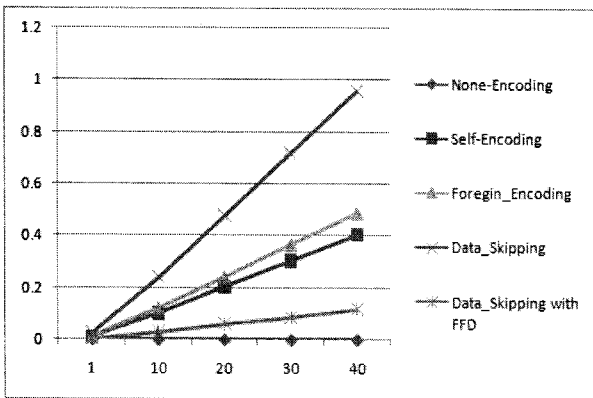


그림 6. 데이터 퓨전 비용의 증가 폭

DataSkipping 은 해당 구간의 데이터가 전송 바운드 이내에 있을 경우 해당 값을 전송하지 않는다. 이는 데이터 수집노드의 응용프로그램 계층에서 계산 될 수 있다. 하지만 이러한 방법을 적용하였을 경우 센서노드가 센싱 하는 데이터와 수집노드에서 수집되는 데이터간의 차이가 발생한다. Y축은 해당 데이터의 정확도를 나타내며 해당값은 센서에서 발생한 데이터와 수집노드에서 수집한 데이터의 값의 차이를 개수로 환산하여 %로 나타낸다. X축은 DataSkipping 을 적용하였을 때 센서데이터가 바운드 내에 존재할 확률이다. DataSkipping 을 하는 최대 횟수는 3회로 세팅하였으며 75%는

모든 트래픽이 전송바운드 내에 있어서 1/4만의 트래픽을 발생시킬 때를 나타낸다. 해당 데이터 트래픽의 종류에 따라 실험데이터는 달라지지만 기상청데이터베이스에서 제공하는 시간상의 온도데이터를 사용하였다.

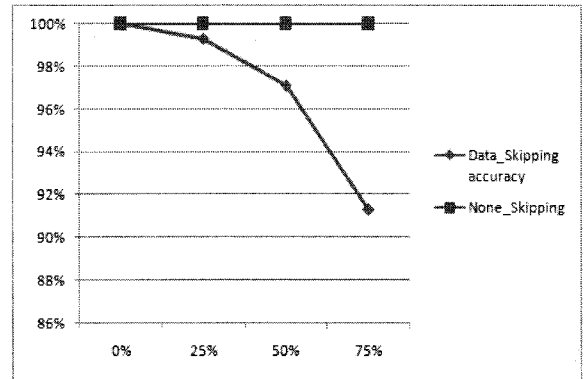


그림 7. 데이터 스킵핑으로 일어나는 데이터 일치도

### VI. 결론

센서네트워크는 주로 주변 환경을 모니터링 하는데 사용되며 실제적인 광대역 통신처럼 다양한 기능을 수행하기는 한계가 있다. 따라서 본 논문의 DataSkipping 은 모니터링이라는 관점에서 네트워크의 지속시간을 최대한 확보하려는 방법을 제안하였다. DataSkipping 의 가장 주된 목적은 센서네트워크에 구성된 노드가 센싱하는 데이터의 시간 관련성에 따라 해당 데이터전송의 효율성을 증가시키는 것이다. 하지만 이는 FFD(Full Function Device)측, 전력이 공급되는 데이터 수집노드가 존재할 때, 이 기능이 극대화된다. 데이터 수집노드는 항상 전력이 공급되기 때문에 해당 센서데이터의 패킷들을 항상 모니터링하면서 노드의 상태와 데이터 등에 대한 연산을 수행하게 된다. 하지만 데이터 수집노드가 관리하는 데이터가 크게 증가할 경우 모든 패킷을 모니터링 해야 하기 때문에 부하가 크게 증가하며, 실제로 적용 했을 경우 문제가 생길 수 있다. 또한 해당 논문은 센서노드의 이동성은 전혀 고려하지 않았다. 센서 노드를 이동한다면 데이터 퓨전 작업을 하는 노드를 결정하는 방법자체에 문제가 생긴다.

### 참고문헌

- [1] B. Krishnamachari, D. Estrin, and S. Wicker, "Impact of Data Aggregation in Wireless Sensor Networks," " Proc. 22nd Int' l. Conf. Distrib. Comp. Sys., Vienna, Austria, pp. 575 - 578, July 2002
- [2] B. J. Bonfils and P. Bonnet, "Adaptive and Decentralized Operator Placement for In-Network Query Processing, " " Proc. 2nd Int' l. Wrksp. Info. Processing in Sensor Networks, Apr. 2003.
- [3] Y. Yao and J. Gehrke, "The Cougar Approach to In-Network Query Processing in Sensor Networks, " " ACM SIGMOD Record, vol. 31, no. 3, 2002.

- [4] P. Rickenbach and R. Wattenhofer, "Gathering Correlated Data in Sensor Networks," Proc. ACM Joint Wksp. Foundations of Mobile Comp., Philadelphia, PA, Oct. 2004.
- [5] R. Cristescu, B. Beferull-Lonzano, and M. Cerlerli. On Network Correlated Data Gathering. In Proc. of the 23rd Conference of the IEEE Communications Society (INFOCOM), 2004.
- [6] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In Proc. of the 33rd Hawaii International Conference on System Sciences (HICSS), 2000.
- [7] I. Solis and K. Obraczka. The Impact of Timing in Data Aggregation for Sensor Networks. In Proc. of the IEEE International Conference on Communications (ICC), 2004.



류 제택

2005년 아주대학교 정보 및 컴퓨터 공학부 (공학사).  
 2007년 아주대학교 정보통신 전문대학원 정보통신공학과(공학석사).  
 2007년~현재 아주대학교 정보통신 전문대학원 박사과정 재학중. 관심분야 센서 네트워크



김 기 형

1990년 한양대학교 졸업(공학사)  
 1992년 한국과학기술원 졸업(공학석사)  
 1996년 한국과학기술원 졸업(공학박사)  
 1997년 2005년 영남대학교 정보통신공학부 교수

2005년 현재 아주대학교 정보및컴퓨터공학부 교수 관심분야 6LoWPAN, 센서네트워크, IPv6, 임베디드시스템



허 남 호

2006년 아주대학교 정보 및 컴퓨터 공학부 (공학사).  
 2006년 ~ 현재 현재 아주대학교 정보통신 전문대학원 석사과정 재학중. 관심분야 센서 네트워크



유 승 화

1972 서울대학교 응용 수학 이학사  
 1980 University of Kansas Computer Science 공학석사  
 1983 University of Kansas Computer Science 공학박사  
 1983-1988 AT&T Bell Labs 연구원

1988-1989 Amdahl Corporation 수석연구원  
 1989-1999 삼성전자 정보통신 전무  
 1999-현재 아주대학교 정보통신 전문대학원 교수 관심분야 Operating Systems, Computer Network, IP, 네트워크 관리