

# 사물인터넷에서 주기적 센싱 데이터 제어를 위한 스마트 디바이스 집합 구성 방안

성윤영\*, 우현제\*, 이미정<sup>o</sup>

## Organizing the Smart Devices' Set for Control of Periodic Sensing Data in Internet of Things

Yoon-young Sung\*, Hyun-je Woo\*, Mee-jeong Lee<sup>o</sup>

### 요약

인터넷에 연결된 사물이 인간의 명시적 개입 없이 정보를 생성하고, 다른 사물, 사람, 시스템과 연동하여 동작하는 사물인터넷(Internet of Things) 기술이 주목받고 있다. 2020년에는 각종 센서와 네트워크 기능이 탑재된 스마트 디바이스가 260억대에 이를 것으로 전망되고, 급증하는 스마트 디바이스로부터 생성되는 센싱 데이터는 기하급수적으로 증가하여 인터넷 연결 트래픽 증가와 무선 채널 혼잡을 야기할 수 있다. 이에 본 논문에서는 사물인터넷에서 주기적인 센싱 데이터를 수집·전달하기 위해 스마트 디바이스를 싱크노드로 활용하는 방안을 제안하고, 스마트 디바이스가 실행 중인 응용의 요구사항을 만족시키는 최적의 싱크노드 집합을 구하는 것은 NP-hard문제[1]이므로 휴리스틱 방법을 이용하여 싱크노드 집합을 구성하는 알고리즘을 제안한다. 제안하는 휴리스틱 알고리즘의 시간복잡도는  $O(m^3)$ 으로 최적의 싱크노드를 구하는 알고리즘보다 빠른 시간에 싱크노드 집합을 구할 수 있다.

**Key Words** : Internet of Things (IoT), sensing data, periodic, sink nodes' set, heuristics

### ABSTRACT

IoT paradigm which makes a information without direct intervention of a human and interworks with other objects, humans and systems is attracting attention. It will be expected the number of smart devices equipped with sensors and wireless communication capabilities is reached to about 260 billion by 2020. With the vast amount of sending data generated from rapidly increasing number of smart devices, it will bring up the traffic growth over internet and congestion in wireless networks. In this paper, we utilize the smart device as a sink node to collect and forward the sensing data periodically in IoT and propose a heuristic algorithm for a selection of sink nodes' set with each sink node satisfies the QoS its applications because a selection of optimal sink nodes' set is NP-hard problem. The complexity of proposed heuristic algorithm is  $O(m^3)$  and faster than the optimal algorithm.

\* 이 논문은 2017년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No. 2015-0-00183, 초연결 자율형 IoT 서비스 지향 네트워크 인프라 기술 연구)

♦ First Author : Ewha Womans University Department of Computer Science and Engineering, syy83@ewhain.net, 학생회원

o Corresponding Author : Ewha Womans University Department of Computer Science and Engineering, lmj@ewha.ac.kr, 종신회원

\* Ewha Womans University Department of Computer Science and Engineering, hjwoo@ewhain.net, 학생회원

논문번호 : KICS2017-01-029, Received January 31, 2017; Revised March 8, 2017; Accepted April 19, 2017

## I. 서 론

사물인터넷<sup>[2]</sup>은 인간과 사물, 서비스의 세 가지 분산된 환경 요소에 대해 인간의 명시적 개입 없이 상호협력적으로 센싱, 네트워킹, 정보 처리 등 지능형 관계를 형성하는 사물 공간 연결망으로 2020년에는 각종 센서와 네트워크 기능이 탑재된 스마트 디바이스가 260억대에 이를 것으로 전망된다. 사물인터넷 환경에서 송·수신되는 데이터는 멀티미디어 콘텐츠처럼 몇 십 또는 몇 백 메가바이트 수준의 대용량이 아니라 몇 킬로바이트 수준의 적은 양이다. 향후 다양한 멀티미디어 콘텐츠들이 생산, 공유, 소비되겠지만 사물인터넷에서는 센싱 데이터 중심의 가벼운 연결이 기하급수적으로 증가하면서 멀티미디어 콘텐츠 같은 대용량 데이터의 트래픽을 뛰어넘는 환경이 될 것으로 예상된다<sup>[3]</sup>.

스마트 디바이스가 급증하는 상황에서 모든 스마트 디바이스가 인터넷 연결을 기반으로 정보를 교환하는 것은 한정된 네트워크 자원을 불필요하게 소모할 수 있으므로 스마트 디바이스에 내장된 네트워크 인터페이스를 이용하여 두 기종이 직접 통신을 하는 것이 급증하는 센싱 데이터를 제어하는 데에 효율적이고 이를 가능하게 하는 저전력 근거리 무선 통신 기술(블루투스 4.0, zigbee 등)의 표준화 진행이 활발하다. 또한 이러한 환경의 변화에 따라 LTE도 저전력 통신 기술인 LTE-M으로 진화하고 있다.

근접 거리에 있는 스마트 디바이스들이 실시간 혹은 초, 분 단위의 주기로 온도, 습도 등의 작은 크기의 데이터 혹은 비교적 큰 크기의 이미지 데이터를 수집·전송할 수 있고 이들 데이터는 재난 안전 분야나 공장·지능형 빌딩 자동화 응용<sup>[4-6]</sup> 등에 활용될 수 있다. 증가하는 IoT 트래픽 관리를 위해 네트워크 경로 정보 파악 및 병목현상 발생 노드에 대한 관리를 통해 라우팅 경로를 견고히 하여 지연 및 에너지 소모를 줄이려는 연구<sup>[7-9]</sup>들이 진행되고 있으나 무선 네트워크에서의 트래픽 부하를 줄여 링크의 품질을 향상시키려는 연구는 미비한 상태이다. 이에 본 논문에서는 일부의 스마트 디바이스가 대표가 되어 다른 디바이스들의 센싱 데이터를 취합하여 인터넷 서버에 전달하게 함으로써 무선 네트워크에서의 혼잡을 예방하고 인터넷 연결 트래픽을 줄이는 방안을 제안한다. 즉, 스마트 디바이스에게 본연의 역할을 수행함과 동시에 센싱 데이터를 수집하여 인터넷 서버에 전달하는 싱크노드의 역할도 수행하게 하여 싱크노드 역할을 수행하는 스마트 디바이스만을 인터넷에 연결시키고자

하는 것이다.

싱크노드 역할을 수행할 대표 디바이스를 선출하고 선출된 디바이스에게 센싱 데이터를 전달하는 것은 무선 센서 네트워크에서 센서들을 그룹화하여 그룹대표가 센싱 데이터를 수집하는 클러스터링 기법과 유사하다. 클러스터링은 많은 수의 센서들이 구현되는 경우, 네트워크 확장을 위해 센서들을 그룹화하여 클러스터를 만들고 그룹대표인 클러스터 헤더(Cluster Header, CH)가 데이터를 수집·전송하여 데이터 전송 횟수를 줄임으로써 에너지 효율을 높이기 위한 것이다. 센서 네트워크에서의 센서는 센싱 데이터 수집을 위한 전용 장치로써 위치 제약 없이 데이터 수집을 위한 최적의 위치에 배치<sup>[10,11]</sup>될 수 있고 CH는 이들 센서 중에서 무작위로 선택되어 해당 역할을 수행할 수 있다. 그러나 사물인터넷에서의 스마트 디바이스는 주요 응용에 따라 위치가 결정되고 주 응용 외에 다른 응용을 동시에 실행할 수 있으므로 스마트 디바이스가 실행 중인 응용의 요구사항을 만족시키면서 싱크노드 역할을 수행할 수 있는 최적의 디바이스를 선택해야 한다. 최소한의 스마트 디바이스가 싱크노드로 선택되는 것이 인터넷 연결 비용을 감소시키는 데에 가장 이상적이나 이는 NP-hard 문제이다. 이에 본 논문에서는 사물인터넷에서 주기적인 센싱 데이터를 수집·전달하기 위해 스마트 디바이스를 싱크노드로 활용하는 방안을 제안하고 휴리스틱 방법을 이용하여 싱크노드 역할을 수행할 스마트 디바이스 집합을 구성하는 알고리즘을 제안한다. 논문의 구성은 다음과 같다. 1장의 서론에 이어 2장에서는 관련 연구로 기존 센서 네트워크에서의 클러스터링 알고리즘을 소개하고 3장에서는 제안하는 알고리즘을 설명한다. 4장에서는 제안 알고리즘을 분석한 결과를 제시하고 5장에서 논문의 결론을 맺는다.

## II. 관련연구: 센서 네트워크에서의 클러스터링 알고리즘

센서 네트워크에서 센서 그룹화를 위한 클러스터링 알고리즘은 센서 배치, 부스트랩 스킴, 추구하는 네트워크 아키텍처, CH의 특징 등에 따라 다양하게 연구되었다. 클러스터링은 네트워크 확장성을 지원 할 뿐 아니라 센싱 데이터 전달 경로를 클러스터 내로 국한시킴으로써 각 센서에서 저장하는 라우팅 테이블의 크기를 감소시키고, 센서 간 중복 데이터 전송을 피할 수 있으며 클러스터 간의 상호작용을 CH에게 전적으로 일임함으로써 통신 대역폭을 절약한다. 또한 클러

스터의 멤버가 되는 센서는 CH와의 연결에만 집중하게 함으로써 에너지 효율을 높일 수 있다<sup>[12]</sup>. 센서 네트워크를 위한 클러스터링 알고리즘<sup>[13-19]</sup>은 센서와 CH가 1-hop 혹은 2-hop 관계로 연결되도록 설계하였는데 각 센서가 통신을 통해 센싱 데이터를 전달하는 것보다 CH가 멤버 센서의 데이터를 수집·처리하는 것이 에너지를 절약할 수 있기 때문이다. 그러나 사물인터넷의 스마트 디바이스는 센서와 달리 충전이 용이하거나 유선으로 전원을 공급받는 등 센서에 비해 배터리 제약을 덜 받으므로 에너지 효율을 위해 CH에 1-hop 혹은 2-hop으로 연결되는 방식의 클러스터링보다 클러스터의 수가 적어 최소한의 CH가 선출되는 것이 인터넷 연결 비용 절감에 유리하다. 이에 본 장에서는 센서 네트워크의 클러스터링 알고리즘 중에서 클러스터 구성 시 CH와 멤버 센서와의 hop 수가 고정되지 않는 알고리즘을 중심으로 관련연구를 살펴본다.

**Random Competition based Clustering (RCC)**<sup>[20]</sup>는 모바일 애드 혹 네트워크를 위해 설계되었으나 센서 네트워크에도 적용 가능한 알고리즘으로 클러스터의 안정성에 중점을 둔다. CH claim 패킷을 가장 먼저 브로드캐스트 한 센서가 CH가 되고 이 브로드캐스트를 들은 센서들이 해당 클러스터의 멤버가 되어 클러스터가 구성되는 알고리즘으로 CH는 CH claim 패킷을 주기적으로 보냄으로써 클러스터를 유지한다. 동시 브로드캐스트로 인한 충돌을 피하기 위해 랜덤 타이머를 이용하나 그럼에도 불구하고 충돌이 발생하는 경우에는 ID가 작은 센서가 CH가 된다. 그러나 CH를 결정할 때 CH 역할을 수행할 수 있는지 센서의 역량을 고려하지 않으므로 서로 다른 응용을 실행하는 이종의 스마트 디바이스들로 구성된 사물인터넷에 RCC를 적용하여 클러스터를 구성하는 것은 적합하지 않다.

**Energy Efficient Hierarchical Clustering (EEHC)**<sup>[21]</sup>는 클러스터를 계층적으로 구성한다. 센서들이 클러스터를 구성하고 각 클러스터의 CH들 간에 다시 클러스터를 구성하여 가장 상위 클러스터의 CH가 base station에 1-hop으로 연결 가능할 때까지 이를 반복한다. 각 센서가  $p$ 의 확률로 CH 역할을 수행하겠다고 알림 메시지를 보내면 이를 수신한  $k$ -hop 내의 센서들이 해당 센서의 클러스터 멤버로 가입하여 클러스터를 구성한다. 클러스터에 가입하지 않은 센서가 일정 시간 동안 CH의 멤버 모집 메시지를 수신하지 못하면 해당 센서는 스스로가 CH가 되어 멤버를 모집한다. EEHC의 주목적은 센서들의 에너지 절약이고

이는 센서가 CH 역할을 수행할 확률  $p$ 와 클러스터 크기를 결정하는  $k$ 에 의존적으로 결정된다. 사물인터넷에서의 클러스터 구성 목적은 디바이스가 실행 중인 응용의 QoS를 만족시키는 동시에 구성되는 클러스터 수를 최소화함으로써 인터넷 직접 연결 비용을 줄이는 것이므로 EEHC를 사물인터넷에 적용시키기에는 추가적으로 고려해야 하는 요소들이 있다.

**Distributed Weight-Based Energy-Efficient Hierarchical Clustering (DWEHC)**<sup>[22]</sup>는 클러스터 크기를 고르게 유지하면서 높은 에너지 효율을 유지하는데 중점을 둔 알고리즘이다. 센서의 배터리 잔량과 이웃 센서와의 거리에 대한 가중치 함수를 정의해서 가중치가 가장 큰 센서가 CH가 되고 1-hop관계의 나머지 센서들은 클러스터의 멤버가 되는 방식으로 클러스터를 구성한다. 그 후 클러스터의 멤버들이 이웃과의 거리를 기반으로 기존의 클러스터에 남을지 2-hop 이상 떨어진 다른 CH의 멤버가 될지 다시 판단하여 필요한 경우 클러스터를 옮긴다. 이러한 작업은 해당 센서의 에너지 효율을 가장 높이는 클러스터를 찾을 때까지 반복하고 반복 횟수는 알고리즘에서 미리 정의하여 제한한다. DWEHC는 센서의 배터리 잔량과 이웃 센서와의 위치 관계를 고려하여 에너지 효율을 극대화 시키나 사물인터넷의 스마트 디바이스는 센서보다 상대적으로 배터리 제약을 덜 받고 디바이스가 실행 중인 응용의 QoS를 고려해야하므로 DWEHC 방법으로 클러스터를 구성하는 것이 적합하지 않다.

### III. 주기적 센싱 데이터 제어를 위한 싱크노드 집합 구성을 위한 휴리스틱 알고리즘

주기적으로 수집한 센싱 데이터를 인터넷 서버로 전달하기 위해 스마트 디바이스를 싱크노드로 활용할 경우 최적의 싱크노드 집합을 구하는 것은 NP-hard 문제이므로, 본 장에서는 이를 해결하기 위한 휴리스틱 알고리즘을 설명한다. 제한하는 알고리즘은 무선 네트워크 및 인터넷 연결 트래픽 감소를 위해 응용의 서비스 품질을 고려하여 무선 네트워크에서의 스마트 디바이스로부터 싱크노드로의 홉 수에 대한 제약조건을 만족시키는 가능한 적은 수의 싱크노드 집합을 구성하는 것을 목표로 한다.

사물인터넷을 구성하는 모든 스마트 디바이스는 싱크노드의 역할을 수행할 수 있으나 본 논문에서는 스마트 디바이스들 중에서 싱크노드가 될 수 있는 후보 디바이스 집합이 별도로 있다고 가정하였다.

기존 센서 네트워크에서는 배터리 제약과 무선 채널에서의 혼잡도를 고려하여 최소 홉 수로 연결 가능한 싱크노드에게 데이터를 전달했다면 스마트 디바이스는 일반 센서에 비해 상대적으로 배터리 제약이 적어 실행 중인 응용의 품질을 만족시키는 한도 내에서 홉 수를 늘려 싱크노드를 결정할 수 있다.

그림 1은 초기 사물인터넷 토폴로지부터 제안하는 알고리즘을 적용하여 싱크노드 집합을 구성하는 과정을 보인 것이다. 그림 1(a)의 초기 토폴로지에서는 모든 스마트 디바이스에 다익스트라 알고리즘을 적용하면 최소 홉으로 연결할 수 있는 싱크노드를 찾을 수 있고 이 싱크노드들로 그림 1(b)와 같이 최소 홉 싱크노드 집합(이하  $C_{shortest}$ )을 구성한다.  $C_{shortest}$ 를 찾는 것은 스마트 디바이스의 효율적인 배터리 사용 및 센싱 데이터 전달의 신속성 측면에서 스마트 디바이스에게는 매우 유익하나 무선 네트워크에서의 채널 혼잡 문제 해결과 인터넷 연결 트래픽을 감소시키는 데에 유의미한 영향을 주지 못한다. 또한, 스마트 디바이스에서 요구하는 데이터 전달의 신속성은 실행 중인 응용의 품질을 만족시키는 수준이면 충분하므로,  $C_{shortest}$ 에 연결된 스마트 디바이스에 대해 응용의 품질을 만족시키는 홉 수내에서 다른 싱크노드의 우회 연결이 가능하지 판단하여 특정 싱크 노드에 연결된 모든 스마트 디바이스가 다른 싱크노드로 우회 연결이 가능하다면 해당 싱크노드는 더 이상 싱크노드의 역할을 수행하지 않게 하여 싱크노드의 수를 줄일 수 있다.  $C_{shortest}$ 의 모든 싱크노드에 연결된 스마트 디바이스에 대해 우회 연결 여부를 판단하여 불필요한 싱크노드를 줄이는 방식으로 그림 1(c)와 같이 휴리스틱 싱크노드 집합(이하  $C_{heuristic}$ )을 구성할 수 있다.  $C_{shortest}$ 로부터  $C_{heuristic}$ 을 찾음으로써 인터넷 서버에 직접 연결하는 싱크노드의 수를 줄이고 이로 인해 인터넷 연결 트래

픽을 감소시키는 것이다.

그림 2는 제안하는 알고리즘의 슈도코드를 보인 것으로 먼저 1) 다익스트라 알고리즘을 적용하여  $C_{shortest}$ 를 구하고, 2)  $C_{shortest}$ 로부터 휴리스틱한 방법으로  $C_{heuristic}$ 을 구한다.

### 3.1 최소 홉 수로 연결 가능한 싱크노드 집합 ( $C_{shortest}$ ) 구하기

모든 스마트 디바이스에 대해 다익스트라 알고리즘을 적용시키면 싱크노드가 될 수 있는 스마트 디바이스 가운데 최소 홉으로 연결 가능한 싱크노드를 찾을 수 있고 이들 싱크노드로  $C_{shortest}$ 를 구성한다.  $C_{shortest}$ 를 구성하는 싱크노드(이하  $c_i$ )는 자신에게 최소 홉으로 연결된 스마트 디바이스 집합 (이하  $S_i$ )과 이들 스마트 디바이스의 연결 홉 수 가운데 가장 큰 값 (이하  $d_i$ )을 알 수 있다.

### 3.2 응용의 품질을 만족시키는 최소한의 휴리스틱 싱크노드 집합( $C_{heuristic}$ ) 구하기

$C_{heuristic}$ 은 두 단계를 거쳐 결정되는데 1)  $C_{shortest}$ 의 한 싱크노드를 선택하여 초기  $C_{heuristic}$ 을 구성하는 단계와 2) 초기  $C_{heuristic}$ 을 기준으로  $C_{shortest}$ 로부터 최종  $C_{heuristic}$ 을 구성하는 단계이다.

#### 3.2.1 초기 $C_{heuristic}$ 구성

초기  $C_{heuristic}$ 은 공집합이다.  $c_i$ 에 연결된 스마트 디바이스 가운데 응용의 품질 만족을 위한 최대 홉 수 (이하  $delay\_cons$ )로 연결된 스마트 디바이스가 존재하는 경우, 해당 디바이스는 다른 싱크노드로의 우회 연결이 불가능하므로  $c_i$ 를  $C_{shortest}$ 에서 제외시키고  $C_{heuristic}$ 에 포함시켜 초기  $C_{heuristic}$ 을 구성한다(1-4). 초기  $C_{heuristic}$ 을 구성하는  $c_i$ 는 연결 홉 수가  $delay\_cons$ 인 디바이스를 포함하여 자신에게 연결된 모든 스마트

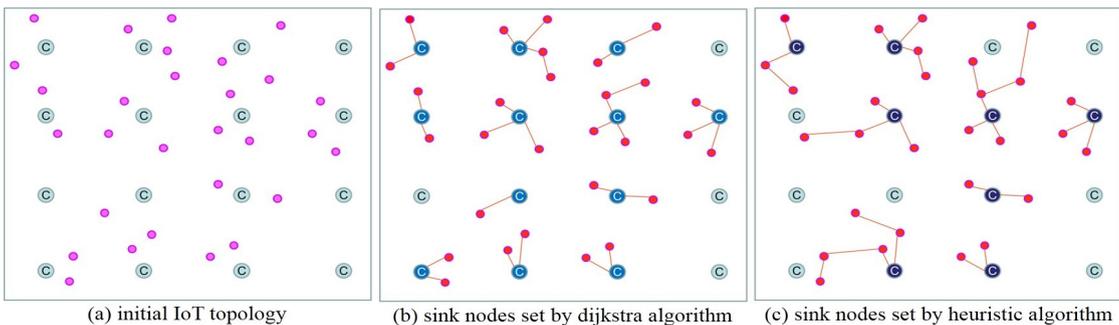


그림 1. 초기 네트워크 토폴로지로부터 다익스트라 알고리즘과 휴리스틱 알고리즘을 적용하여 싱크노드 집합을 선택하는 과정  
Fig. 1. Find heuristic sink node set by dijkstra and proposed heuristic algorithm

트 디바이스를 위해 싱크노드 역할을 수행한다.  $C_{shortest}$ 에 있는 모든  $c_i$ 에 대해 위의 과정을 적용시킨다.

$C_{shortest}$ 의 모든 싱크노드에 대해  $delay\_cons$ 로 연결된 스마트 디바이스가 존재하지 않는 경우에는 다른 싱크노드로의 우회 가능성이 가장 낮은 스마트 디바이스(즉,  $delay\_cons - d_i$ 의 값이 가장 작은 디바이스)를 찾아 해당 스마트 디바이스가 연결된 싱크노드를  $C_{shortest}$ 에서 제외하고  $C_{heuristic}$ 에 포함시킨다. 만약 다른 싱크노드로의 우회 가능성이 가장 낮은 스마트 디바이스가 여러 개이고 이들이 연결된 싱크노드끼리 다르다면 연결될 디바이스의 수가 많은 싱크노드를 우선 적으로  $C_{heuristic}$ 에 포함시킨다(5-10).

```

C={c_i | smart devices as a candidate sink node, 1≤i≤n}
S={s_j | smart devices, 1≤j≤m}

shortest_hop(s, c): shortest hop count from s to nearest c
C_shortest={c∈C | shortest_hop(s, c)≤delay_cons, ∀s∈S}
S_i={s∈S | shortest_hop(s, c_i), c_i∈C_shortest}
d_i=MAX{shortest_hop(s, c_i), ∀s∈S_i}

C_heuristic: heuristically selected sink nodes' set
            in C_shortest

#Find Initial C_heuristic - First step
Input: C_shortest, Output: C_heuristic
1: For every c_i∈C_shortest
2:   If (∃s∈S_i shortest_hop(s, c_i) == delay_cons)
3:     C_heuristic = C_heuristic ∪ {c_i}
4:     C_shortest = C_shortest - {c_i}
5: If(C_heuristic == ∅)
6:   select the c_i with MIN{delay_cons - d_i,
                           ∀c_i∈C_shortest}
7:   If(c_i is more than two)
8:     select the c_i with largest |S_i|
9:     C_heuristic = C_heuristic ∪ {c_i}
10:  C_shortest = C_shortest - {c_i}

#Find C_heuristic - Second step
List: c_i sorted by |S_i| in decreasing order,
      ∀c_i∈C_shortest
    
```

```

- c_i > c_j, if |S_i|=|S_j| and d_i > d_j,
                    any c_i, c_j∈C_shortest
- f(i)=k, c_i is kth sink node in List
s_i: ith smart device in S_i
C_t: a set of sink node that s_t can connect by
      bypassing
dist(s, c): hop count from s to c

Input: List, C_shortest, C_heuristic Output: C_heuristic
11: For k = 1 to k = |C_shortest|
12:   For each s_t∈S_{f^{-1}(k)}, t = 1 to t = |S_{f^{-1}(k)}|
13:     For each c_h∈C_heuristic, h = 1 to
                                h = |C_heuristic|
14:       d_max = delay_cons - shortest_hop(s_t, c_h)
15:       For detour = 1 to detour = d_max
16:         if(shortest_hop(s_t, c_{f^{-1}(k)}) + detour
              == shortest_hop(s_t, c_h))
17:           C_t = C_t ∪ {c_h}
18:           break
19:         End for
20:       End for
21:       If(C_t == ∅)
22:         C_heuristic = C_heuristic ∪ {c_{f^{-1}(k)}}
23:         C_shortest = C_shortest - {c_{f^{-1}(k)}}
24:         break
25:       End for
26:   If(c_{f^{-1}(k)}∈C_shortest)
27:     For each s_t∈S_{f^{-1}(k)}, t = 1 to t = |S_{f^{-1}(k)}|
28:       If(|C_t| == 1)
29:         S_h = S_h ∪ {s_t}, c_h∈C_t
30:       Else
31:         For each c_j∈C_t
32:           select the c_j with MIN{dist(s_t, c_j),
                                    ∀c_j∈C_t}
33:         S_j = S_h ∪ {s_t}
34:       End for
35:     End for
36:     C_shortest = C_shortest - {c_{f^{-1}(k)}}
37:   End for
    
```

그림 2. 제안하는 휴리스틱 알고리즘의 슈도코드  
Fig. 2. Pseudo code of proposed heuristic algorithm

### 3.2.2 최종 $C_{heuristic}$ 구성

초기  $C_{heuristic}$ 이 결정되면  $c_i$ 에 연결된 모든 스마트 디바이스에 대해  $C_{heuristic}$ 로 delay\_cons 내에 우회하여 연결 가능한지 판단한다. 우회 연결이 가능한 경우  $C_{heuristic}$ 이 해당 싱크노드의 역할을 대신하고 그렇지 않은 경우  $c_i$ 를  $C_{heuristic}$ 으로 포함시키는 방법으로  $C_{heuristic}$ 을 구성해 나간다. 가능한 최소한의 휴리스틱 싱크노드를 선택하기 위해 연결된 스마트 디바이스의 수가 많은  $c_i$ 를 우선적으로 판단하고, 이를 위해  $C_{shortest}$ 의 모든  $c_i$ 를  $|S_i|$ 에 대한 내림차순로 정렬한  $c_i$  리스트(이하 List)를 만든다.  $|S_i|$ 가 같은 경우에는  $d_i$ 가 큰 싱크노드를 우선한다. 다음으로 List의 모든  $c_i$ 에 대해  $C_{heuristic}$ 의 싱크노드로 우회하여 연결할 수 있는지 확인한다.  $C_{heuristic}$ 에 우회하여 연결할 수 없는 스마트 디바이스가 하나라도 존재하는  $c_i$ 는  $C_{shortest}$ 에서 제외하여  $C_{heuristic}$ 에 포함시키고(21-24),  $c_i$ 에 연결된 모든 디바이스는 여전히  $c_i$ 를 싱크노드로 여겨 데이터를 전달한다.  $c_i$ 에 연결된 모든 디바이스가  $C_{heuristic}$ 의 싱크노드로 우회하여 연결할 수 있다면 이때는 각 디바이스를 우회 연결이 가능한  $C_{heuristic}$ 으로 연결시키는데, 우회 할 수 있는 싱크노드가 여러 개인 경우 연결 홉 수가 가장 작은 싱크노드로 연결한다(30-34). 이후  $c_i$ 는  $C_{shortest}$ 에서 제거한다(36). 이 과정을 List의 모든 싱크노드에 적용하여 최종  $C_{heuristic}$ 을 구한다.

## IV. 휴리스틱 알고리즘 분석 및 실험

제안하는 휴리스틱 알고리즘은 우선적으로 다익스트라 알고리즘을 이용해  $C_{shortest}$ 를 구하는데  $m \times (m+n)^2$ 의 시간이 필요하고,  $C_{heuristic}$ 을 구하는 첫 단계에서  $n$ 의 시간, 두 번째 단계에서 List를 구하는  $n^2$ 의 시간을 포함하여  $n^2 + n \times (m \times n \times d_{max} + m)$ 의 시간이 필요하므로 최악의 경우  $O(m^3)$ 의 복잡도를 갖는다. [1]에서 분석한 최적의 싱크노드 집합을 구하는 알고리즘은 지수시간 복잡도 ( $O(2^m)$ )를 갖는데, 그림 3은 지수시간의 복잡도와 다항시간의 복잡도를 비교하여 보인 것으로  $m$ 이 10 이상일 때 다항시간 복잡도가 지수시간 복잡도 보다 항상 빠름을 알 수 있다. 즉, 사물인터넷을 구성하는 스마트 디바이스의 수가 10대 이상이라고 가정하는 것에 큰 문제가 없으므로 제안하는 휴리스틱 알고리즘의 수행 시간이 최적의 알고리즘보다 빠르다고 결론지을 수 있다.

한편, 실험을 통해 저전력 근거리 무선 통신 프로토콜인 블루투스 4.0, zigbee를 사용하는 경우에 대해

휴리스틱하게 싱크노드 집합을 구성하였다. 사물인터넷 토폴로지 크기는  $1,000m \times 1,000m$ 이고, 제안 방안의 가정대로 싱크노드가 될 수 있는 후보 디바이스는 토폴로지 내에 균일한 간격으로 가로 8개, 세로 8개씩 총 64개를 배치하였다. 스마트 디바이스는 시나리오 별로 각각 200, 300개이며 토폴로지 내에 무작위로 배치한다. 스마트 디바이스가 실행 중인 응용의 지연 제약은 응용의 종류에 따라 다르나 실험에서는 모두 같다고 가정하여 지연 제약을 4~7 홉으로 변화시켜가며 싱크노드를 선택하였다. 싱크노드 선택 시 싱크노드 역할을 수행할 스마트 디바이스가 현재 처리하고 있는 데이터양은 특별히 고려하지 않는다. 1 홉은 통신 프로토콜의 통신 가능 범위로 블루투스 4.0은 60m, zigbee는 75m로 설정하여 실험하였다.

휴리스틱 알고리즘을 이용하여 가능한 적은 수의 싱크노드가 선택되는 것을 보이기 위해 시나리오 별로 선택된 싱크노드의 수를 측정하였고, 이 때, 각 싱크노드 별 연결된 센서의 수와 각 센서들로부터 연결된 싱크노드까지의 홉 수를 측정하였다. 모든 측정치는 실험을 10번 반복하여 평균을 낸 값이다.

그림 4는 블루투스 4.0을 이용할 때 선택되는 싱크노드의 수로 64개의 싱크노드 중 다익스트라 알고리즘을 적용하여 최단 홉으로 연결되는 싱크노드는 시나리오 별로 50개, 59개이고, 응용의 지연 제약을 증가시켜가며 다른 싱크노드에 우회하여 연결하면 선택되는 싱크노드는 42.6개에서 37개, 50.2개에서 40.5개로 감소함을 볼 수 있다. 그림 5는 zigbee를 이용할 때 선택된 싱크노드의 수로 최단 홉으로 연결된 싱크노드는 시나리오 별로 56.3개, 61.3개이고 휴리스틱 알고리즘을 적용하여 우회연결을 하면 지연 제약이 증가함에 따라 선택되는 싱크노드의 수가 37.4개에서

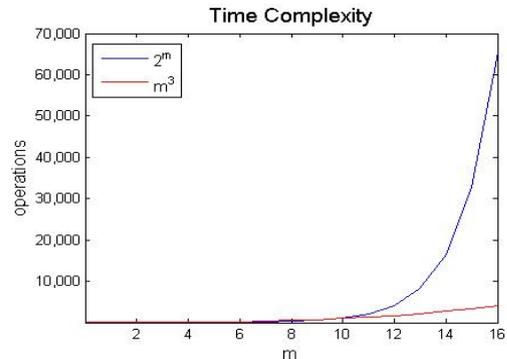


그림 3.  $2^m$ 과  $m^3$ 의 시간복잡도  
Fig. 3. time complexity of  $2^m$  and  $m^3$

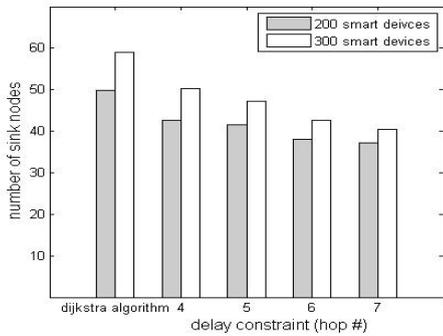


그림 4. 선택된 싱크노드의 수 (블루투스 4.0)  
Fig. 4. number of selected sink nodes by bluetooth 4.0

29.3개, 38.7개에서 28개로 감소함을 볼 수 있다.

그림 6과 그림 7은 각각 블루투스 4.0과 zigbee를 이용하는 경우, 휴리스틱 알고리즘을 이용해 싱크노드를 선택할 때 싱크노드로부터 연결된 스마트 디바이스의 연결 홉 수의 최대값과 모든 스마트 디바이스에서 싱크노드에 연결된 홉 수의 평균을 다익스트라 알

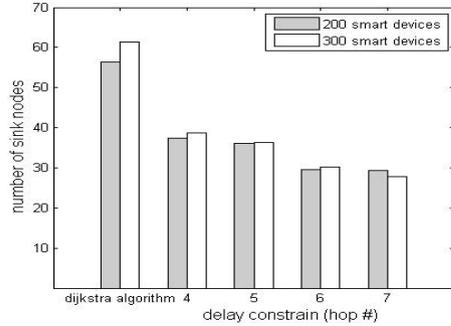


그림 5. 선택된 싱크노드의 수 (zigbee)  
Fig. 5. number of selected sink nodes by zigbee

고리즘을 적용했을 때와 비교하여 보인 것이다. 지연 제약이 증가할수록 우회하여 연결이 가능한 싱크노드가 늘어나므로 연결 홉의 최대값과 평균 연결 홉 수가 증가하나 연결 홉 수의 증가폭이 많이 크지 않음을 볼 수 있다.

그림 8과 그림 9는 각각 블루투스 4.0과 zigbee를 사용할 때, 각 싱크노드에 연결된 스마트 디바이스의

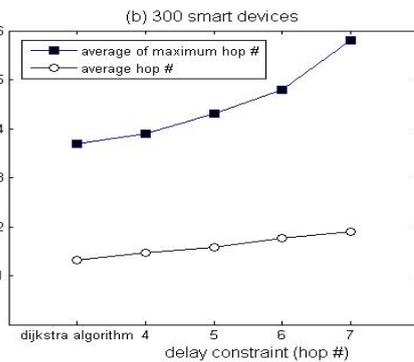
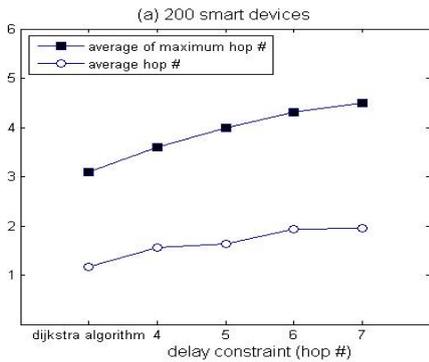


그림 6. 각 스마트 디바이스로부터 싱크노드로의 연결 홉 수 (블루투스 4.0)  
Fig. 6. number of hops from smart devices to its sink node by bluetooth 4.0

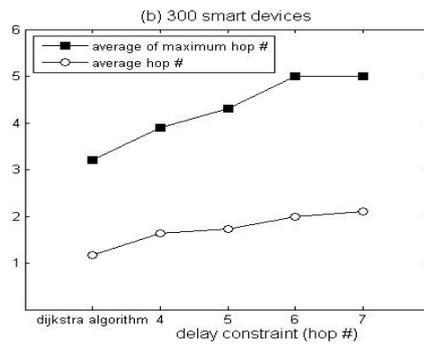
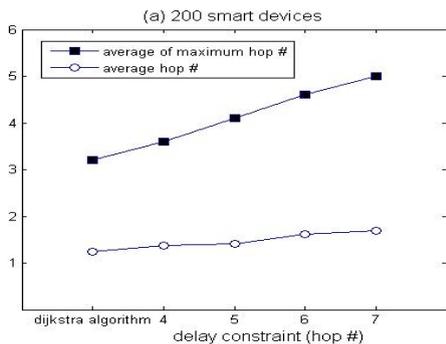


그림 7. 각 스마트 디바이스로부터 싱크노드로의 연결 홉 수 (zigbee)  
Fig. 7. number of hops from smart devices to its sink node by zigbee

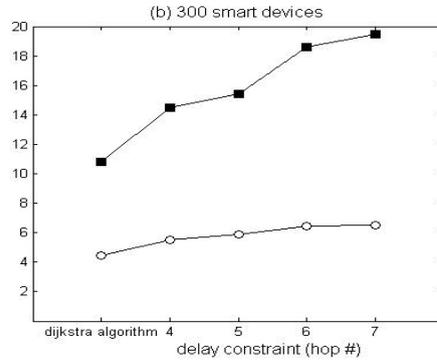
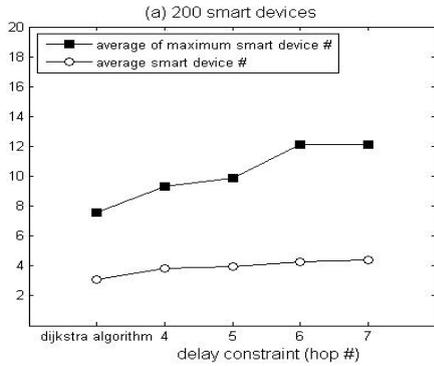


그림 8. 싱크노드에 연결된 스마트 디바이스의 수 (블루투스 4.0)  
Fig. 8. number of smart devices per sink nodes by bluetooth 4.0

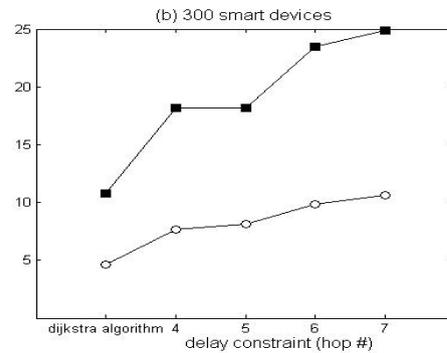
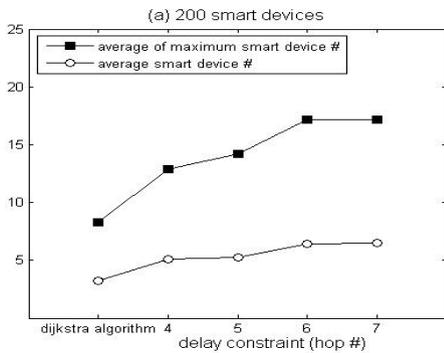


그림 9. 싱크노드에 연결된 스마트 디바이스의 수 (zigbee)  
Fig. 9. number of smart devices per sink nodes by zigbee

수를 보인 것이다. 지연 제약이 증가함에 따라 선택되는 싱크노드의 수가 감소하므로 각 싱크노드가 서비스해야 하는 디바이스의 수가 증가하고, 실험을 통해 이를 확인할 수 있었다.

## V. 결 론

본 논문에서는 사물인터넷 환경에서 급증하는 센싱 데이터의 효율적인 제어를 위해 스마트 디바이스를 싱크노드로 활용하는 방안을 제안하고, 휴리스틱하게 싱크노드를 선택하는 알고리즘을 제안하였다. 제안하는 휴리스틱 알고리즘의 복잡도는  $O(m^3)$ 으로 최적의 싱크노드 집합을 구하는 알고리즘의 지수시간 복잡도 ( $O(2^m)$ )보다 빠르게 싱크노드 집합을 구할 수 있다. 블루투스 4.0과 zigbee를 사용하는 환경에서 제안하는 휴리스틱 알고리즘을 적용시켜 응용의 지연 제약을 변경해가며 선택되는 싱크노드의 수를 비교하였다. 지연 제약이 증가할수록 우회 연결이 가능한 싱크노드의 수가 증가하므로 휴리스틱하게 선택되는

싱크노드의 수는 감소하고 이로 인해 각 싱크노드가 서비스하는 스마트 디바이스의 수는 증가하는 것을 확인할 수 있었다. 제안하는 휴리스틱 알고리즘은 다른 싱크노드로 우회하여 연결할 때 응용의 지연 제약만을 고려하는데 향후 싱크노드의 처리 가능한 데이터양을 함께 고려하고 스마트디바이스와 싱크노드의 이동성을 반영할 수 있도록 제안하는 알고리즘을 확장할 계획이다.

## References

- [1] H. J. Woo and M. J. Lee, "A study on the use of smart devices for effective control of the periodic sensing data in inter of things," in *Proc. KICS Winter Conf.*, pp. 1052-1053, Gangwon-do, Korea, Jan. 2016.
- [2] K. S. Min, *Internet of Things(2012)*, Retrieved Jan., 12, 2017, from <http://www.kisa.or.kr>
- [3] D. H. Shin, *IoT era, three things are different*

- from mobile era(2014), Retrieved Jan., 12, 2017, from <http://www.lgeri.com>
- [4] C. S. Oh, M. S. Seo, J. H. Lee, S. H. Kim, Y. D. Kim, and H. J. Park, "Indoor air quality monitoring systems in the IoT environment," *J. KICS*, vol. 4, no. 5, pp. 886-891, May 2015.
- [5] K. B. Kang, H. K. Ahn, H. S. Kim, S. H. Lee, and J. W. Jwa, "Development of vending machine for electricity based on Z-Wave mesh network," *J. KICS*, vol. 41, no. 10, pp. 1256-1262, Oct. 2016.
- [6] J. W. Cha, J. Y. Lee, and J. H. Lee, "Secure smart safety system using streetlight infrastructure," *J. KICS*, vol. 40, no. 5, pp. 851-856, May 2015.
- [7] J. H. Kim and J. Y. Lee, "A study about problem of routing with MPTCP for advanced performance in homegeneous network for IoT," in *Proc. KICS Int. Conf. Commun.*, pp. 1043-1044, Jeju island, Korea, Jun. 2014.
- [8] S. H. Lee, J. H. Hong, N. U. Kim, and T. M. Chung, "Application of SDN technologies for IoT traffic management in MLN," in *Proc. KICS Int. Conf. Commun.*, pp. 262-263, Jeju island, Korea, Jun. 2015.
- [9] I. J. Jung, S. H. Chung, and S. J. Lee, "A routing scheme considering bottleneck and route link quality in RPL-based IoT wireless networks," *J. KIISE*, vol. 42, no. 10, pp. 1268-1279, Oct. 2015.
- [10] E. I. Oyman and C. Ersoy, "Multiple sink network design problem in large scale wireless sensor networks," in *Proc. ICC 2004*, pp. 3663-3667, Paris, France, Jun. 2004.
- [11] S. Dai, C. Tang, S. Qiao, K. Xu, H. Li, and J. Zhu, "Optimal multiple sink nodes deployment in wireless sensor networks based on gene expression programming," in *Proc. ICCSN 2010*, pp. 355-359, Singapore, Feb. 2010.
- [12] A. A. Abbasia and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Commun.*, vol. 30, no. 14-15, pp. 2826-2841, Oct. 2007.
- [13] D. J. Baker and A. Ephremides, "The architectural organization of a mobile radio network via a distributed algorithm," *IEEE Trans. Commun.*, vol. 29, no. 11, pp. 1694-1701, Nov. 1981.
- [14] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 7, pp. 1265-1275, Sept. 1997.
- [15] R. Nagpal and D. Coore, "An algorithm for group formation in an amorphous computer," in *Proc. PDCS'98*, Las Vegas, Oct. 1998.
- [16] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "Application specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wirel. Commun.*, vol. 1, no. 4, pp. 660-670, Oct. 2002.
- [17] M. Demirbas, A. Arora, and V. Mittal, "FLOC: a fast local clustering service for wireless sensor networks," in *Proc. DIWANS'04*, Florence, Italy, Jun. 2004.
- [18] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mob. Comput.*, vol. 3, no. 4, pp. 366-379, Oct.-Dec. 2004.
- [19] H. Huang and J. Wu, "A probabilistic clustering algorithm in wireless sensor networks," in *Proc. IEEE VTC-2005-Fall*, vol. 3, pp. 1796-1798, Dallas, TX, Sept. 2005.
- [20] K. Xu and M. Gerla, "A heterogeneous routing protocol based on a new stable clustering scheme," in *Proc. MILCOM 2002*, vol. 2, pp. 838-843, Anaheim, CA, Oct. 2002.
- [21] S. Bandyopadhyay and E. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Proc. INFOCOM 2003*, vol. 3, pp. 1713-1723, San Francisco, California, Apr. 2003.
- [22] P. Ding, J. Holliday, and A. Celik, "Distributed energy efficient hierarchical clustering for wireless sensor networks," in *Proc. DCOSS'05*, pp. 322-339, Marina Del Rey, CA, Jun. 2005.

**성 윤 영 (Yoon-young Sung)**



2007년 2월 : 이화여자대학교 컴  
퓨터학과 졸업  
2009년 8월 : 이화여자대학교 컴  
퓨터정보통신공학과 석사  
2009년 9월~현재 : 이화여자대  
학교 컴퓨터공학과 박사과정

<관심분야> 무선 이동 네트워크, VANETs, 사물인터  
넷, etc.

**우 현 제 (Hyun-je Woo)**



2004년 : 이화여자대학교 컴퓨  
터학과 졸업  
2006년 : 이화여자대학교 컴퓨  
터학과 석사  
2007년~현재 : 이화여자대학교  
컴퓨터정보통신공학과 박사  
과정

<관심분야> 무선 이동 네트워크, Ad-hoc 네트워크,  
VANETs, 사물인터넷, etc.

**이 미 정 (Mee-jeong Lee)**



1987년 : 이화여자대학교 전자  
계산학과 졸업  
1989년 : University of North  
Carolina at Chapel Hill 컴  
퓨터학과 석사  
1994년 : North Carolina State  
University 컴퓨터공학과 박사

1994년~현재 : 이화여자대학교 공과대학 컴퓨터공학  
과 교수

<관심분야> 프로토콜 설계 및 성능분석, 멀티미디  
어 전송을 위한 트래픽 제어, 인터넷 QoS, 트래  
픽 엔지니어링, 무선 이동 네트워크, VANETs,  
etc.