



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

2015년 8월  
석사학위 논문

무선 센서 네트워크의  
클러스터 내 전송을 위한 노드  
밀도 기반 트랜스포트  
프로토콜

조선대학교 대학원

컴퓨터공학과

백 철 현

무선 센서 네트워크의  
클러스터 내 전송을 위한 노드  
밀도 기반 트랜스포트  
프로토콜

A Node-Density-Aware Transport Protocol for  
Intra-Cluster Transmissions in Wireless Sensor  
Networks

2015년 8월 25일

조선대학교 대학원

컴퓨터공학과

백 철 현

무선 센서 네트워크의  
클러스터 내 전송을 위한 노드  
밀도 기반 트랜스포트  
프로토콜

지도교수 모 상 만

이 논문을 컴퓨터공학 석사학위신청 논문으로  
제출함

2015 년 4 월

조선대학교 대학원

컴퓨터공학과

백 철 현

## 백철헌의 석사학위논문을 인준함

위원장	조선대학교 교수	강 문 수 (인)
위 원	조선대학교 교수	모 상 만 (인)
위 원	조선대학교 교수	정 일 용 (인)

2015 년 5 월

조선대학교 대학원

ABSTRACT .....	vi
제1장 서론 .....	1
제2장 관련 연구 .....	3
제1절 신뢰성 기반 전송 제어 .....	4
1. PSFQ 프로토콜 .....	4
2. RMST 프로토콜 .....	5
3. ESRT 프로토콜 .....	6
제2절 트래픽 기반 혼잡 제어 .....	6
1. CODA 프로토콜 .....	6
2. STCP 프로토콜 .....	7
3. ECN 프로토콜 .....	8
제3절 자원 기반 혼잡 제어 .....	10
1. TARA 프로토콜 .....	10
2. TADR 프로토콜 .....	10
제4절 DCA 프로토콜 .....	12
제5절 LEACH 클러스터링 .....	12
제3장 노드 밀도 기반 트랜스포트 프로토콜 .....	14
제1절 듀티 사이클 .....	14
제2절 클러스터 헤드 리소스 제어 .....	16
제3절 클러스터 헤드 에너지 조절 .....	18
제4장. 성능 평가 .....	21
제1절. 시뮬레이션 환경 .....	21

제2절. 시뮬레이션 결과 .....	22
제5장. 결론 .....	27
참고 문헌 .....	28

## 그림 목 차

그림 1. 무선 센서 네트워크 프로토콜의 분류 .....	3
그림 2. PSFQ의 전송방식 .....	4
그림 3. RMST 프로토콜의 구성 .....	5
그림 4. CODA의 전송방식 .....	7
그림 5. 혼잡제어 핫스팟 시나리오 .....	11
그림 6. LEACH 클러스터링 결과 .....	13
그림 7. 주기적인 센서 활성화 방식 .....	14
그림 8. 클러스터 헤드 리소스 제어 흐름 .....	17
그림 9. 클러스터 헤드의 자원 제어 방식 .....	18
그림 10. 클러스터 헤드 에너지 값 전송 .....	19
그림 11. 클러스터 헤드 에너지 조절 흐름도 .....	20
그림 12. 라운드 수에 따른 살아있는 노드의 수(초기 노드수 = 200) .....	22
그림 13. 라운드 수에 따른 살아있는 노드의 수(초기 노드수 = 100) .....	23
그림 14. 클러스터 내 노드 수에 따른 혼잡도 .....	23
그림 15. 클러스터 내 노드 수에 따른 패킷 전송 실패율 .....	24
그림 16. 패킷 송신율에 따른 혼잡도 .....	24
그림 17. 패킷 송신율에 따른 패킷 전송 실패율 .....	25
그림 18. 클러스터 수에 따른 클러스터 헤드의 데이터 전송 실패율 .....	25
그림 19. 클러스터 수에 따른 혼잡도 .....	26



## 표 목 차

표 1. STCP 헤더정보 .....	8
표 2. STCP의 헤더정보(ACK/NACK) .....	8
표 3. 수식에 사용되는 파라미터 .....	15
표 4. 시뮬레이션 파라미터 .....	21

# A B S T R A C T

## A Node-Density-Aware Transport Protocol for Intra-Cluster Transmissions in Wireless Sensor Networks

Cheolheon Baek

Advisor: Prof. Sangman Moh, Ph.D.

Department of Computer Engineering

Graduate School of Chosun University

In wireless sensor networks (WSNs), a transport protocol plays an important role of reliable transmission and congestion control. The efficient design of a transport protocol contributes to energy conservation as well as performance improvement in WSNs. In particular, an efficient congestion control protocol is necessarily required for reliable transmission and energy saving. In this thesis, a node-density-aware transport protocol (NDTP) for intra-cluster transmissions in WSNs is proposed, which takes node density into account to mitigate congestion in intra-cluster transmissions. Because cluster heads do more works and consume more energy than normal sensor nodes, the maximum active time and queue length of cluster heads are restricted to reduce energy consumption in NDTP. According to the performance evaluation results, the proposed NDTP significantly outperforms the conventional protocol in terms of network lifetime, congestion frequency, and packet error rate.

## 제1장 서 론

무선 센서 네트워크(wireless sensor network: WSN)는 센서로 센싱이 가능하고 수집된 정보를 가공하는 프로세서가 달려 있으며 이를 전송하는 소형 무선 송수신 장치이다. 무선 통신 기능을 갖추고 배터리로 동작하는 센서 노드(sensor node)와 이를 수집하여 외부로 내보내는 싱크 노드(sink node)로 구성된 네트워크이다. 기존의 네트워크와 다르게 의사소통의 수단이 아니라 자동화된 원격 정보 수집을 기본 목적으로 한다.

무선 센서 네트워크는 사람들이 접근하기 어려운 장소에서의 물리 현상 감시 뿐만 아니라 과학적, 의학적, 군사적, 상업적 및 실생활에 이르기까지 광범위하게 응용되고 있다[1]. WSN은 일반 네트워크와 달리 에너지 공급이 제한적이고 이로 인해 네트워크 수명이 짧다. 따라서, 센서 노드 및 프로토콜 설계의 가장 중요한 기준은 에너지 효율성이다[2].

WSN에서 트랜스포트 프로토콜(transport protocol)은 패킷 전송의 신뢰성 회복과 혼잡을 제어하는 중요한 역할을 한다[3]. 네트워크에서 메시지 송수신의 종단 노드는 전체 네트워크의 안정성을 보장하기 위해 반드시 혼잡 상황에 대해 반응해야 된다[4]. WSN에서의 프로토콜 역시 혼잡 제어를 효과적으로 수행해야 하고 신뢰성 있는 전송을 보장할 수 있어야 한다.

WSN의 트랜스포트 프로토콜에서 신뢰성 기반으로 작동하는 프로토콜은 PSFQ (Pump Slowly Fetch Quickly)[5], RMST (Reliable Multi-Segment Transport)[6], ESRT (Event-to-Sink Reliable Transport)[7] 가 있다. 이 프로토콜들은 혼잡제어 보단 데이터의 온전한 전송을 목적으로 두고 전송하는 방식이다. 그리고 무선 센서 네트워크의 혼잡 제어 프로토콜은 크게 트래픽 기반과 자원 기반으로 나뉜다. 트래픽 기반 혼잡제어 프로토콜들은 CODA (COngestion Detection and Avoidance)[8], STCP (Sensor Transmission Control Protocol)[9], ECN (Explicit Congestion Notification)[10] 등이 있다. 자원 기반의 프로토콜은 TARA (Topology-Aware Resource Control)[11], TADR (Traffic-Aware Dynamic Routing)[12], DCA (Duty-cycle based Congestion Avoidance)[13] 프로토콜이 있다. 트래픽 기반의 혼잡제어는 발생하는 트래픽을 낮춰서 혼잡을 제어하는 방식의 기법이고 자원 기반의

혼잡제어는 데이터의 송수신량을 제어하여 혼잡을 제어하는 방식으로 볼 수 있다.

본 논문에서는 WSN에서 노드 밀도를 기반으로 한 혼잡 제어 기법을 이용하여 에너지 효율적인 혼잡제어 프로토콜을 제안한다. 제안한 프로토콜은 NDTP (Node-Density-aware Transport Protocol)로 명명되며, 자원 기반 프로토콜로 분류될 수 있다. 본 논문에서 기술하는 자원 기반제어는 송수신 데이터의 양은 노드 각각이 보내는 데이터를 줄인다는 의미보단 무선 센서 네트워크의 클러스터 내에서 헤드가 받는 수신 데이터의 양을 줄임으로써 혼잡을 제어하는 방식이다. DCA 프로토콜은 자원 기반 프로토콜인 TARA와 TADR 프로토콜의 성능을 개선하여 만든 프로토콜이므로 본 논문에서는 DCA 프로토콜과 성능평가를 비교 하였다. 성능평가 결과에 의하면 기존의 DCA 방식에 비해 클러스터 내의 패킷 전송율을 감소시키고 노드의 생존시간이 3라운드 만큼 늘어났다. 그리고 노드 밀도가 높을수록 혼잡 빈도가 기존 방식에 비해 크게 감소하였다.

본 논문은 다음과 같이 구성된다. 제 2장에서는 무선 센서 네트워크에서 혼잡을 제어하는 목적으로 연구되어 온 종래의 방식들을 검토 분석한 결과를 기술한다 제 3장에서는 본 논문에서 제안한 새로운 혼잡제어의 원리와 동작 측면에서 상세히 기술한다. 제 4장에서는 시뮬레이션을 통한 성능 평가 결과를 기술하고 성능 향상 효과를 검증한다. 마지막으로 제 5장에서는 본 논문의 결론을 기술한다.

## 제2장 관련 연구

그림 1은 무선 센서 네트워크에서 프로토콜의 분류를 나타낸 그림이다. 크게 신뢰성 기반 프로토콜과 혼잡제어 프로토콜로 나뉘고 혼잡제어 프로토콜에서는 또 트래픽 기반 혼잡제어 방식과 자원 기반 혼잡제어 방식으로 나뉜다. 신뢰성 기반의 대표적 프로토콜은 PSFQ, RMST, ESRT가 있고 트래픽 기반 혼잡제어에는 CODA, STCP, ECN이 있다. 마지막으로 자원 기반 혼잡제어 방식에는 TARA, TADR, DCA 프로토콜이 있다. 이 절에서는 기존의 프로토콜들의 기법을 나열하고 장단점을 설명한다.

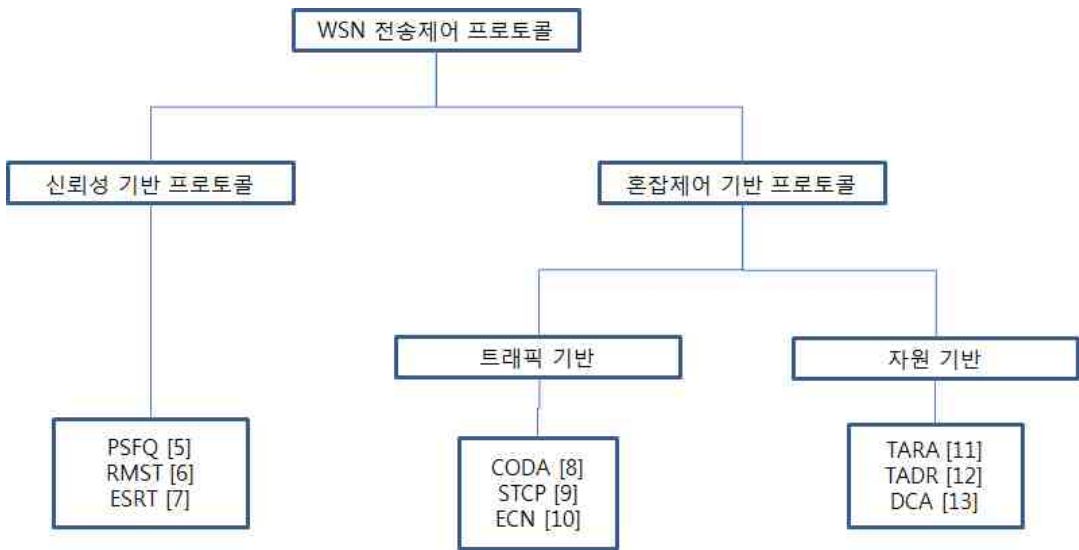


그림 1. 무선 센서 네트워크 프로토콜의 분류

## 제1절 신뢰성 기반 프로토콜

### 1 PSFQ 프로토콜

PSFQ 프로토콜은 신뢰성 있는 전송을 최우선으로 고려한다[5]. PSFQ는 상당한 오버헤드가 존재하기 때문에 upstream (센서로부터 싱크로 전송)로의 전송이 아닌 downstream (싱크로부터 센서로 전송)에서 주로 사용하며 노드에 의한 제어나 관리를 목적으로 한다. 하지만 이 방식은 중간 노드에서 손실된 패킷이 완전히 복구될 때까지 손실 패킷 이후 전송된 패킷을 버퍼에 저장하기 때문에 버퍼 오버플로우 발생 가능성이 높다. 그림 2는 PSFQ 프로토콜의 전송방식을 나타낸 그림이다. 그림처럼 2번 패킷이 전송이 되지 않았을 경우 NACK (negative acknowledgement)를 보내 패킷이 유실되었다는 신호를 이전 노드에게 보낸다. 그리고 노드는 2번 패킷이 도착할 때까지 다음 패킷인 3번을 전송하지 않고 기다렸다가 2번 패킷이 도착하면 2번 패킷을 먼저 보낸 후에 3번 패킷을 보내는 걸 나타낸다.

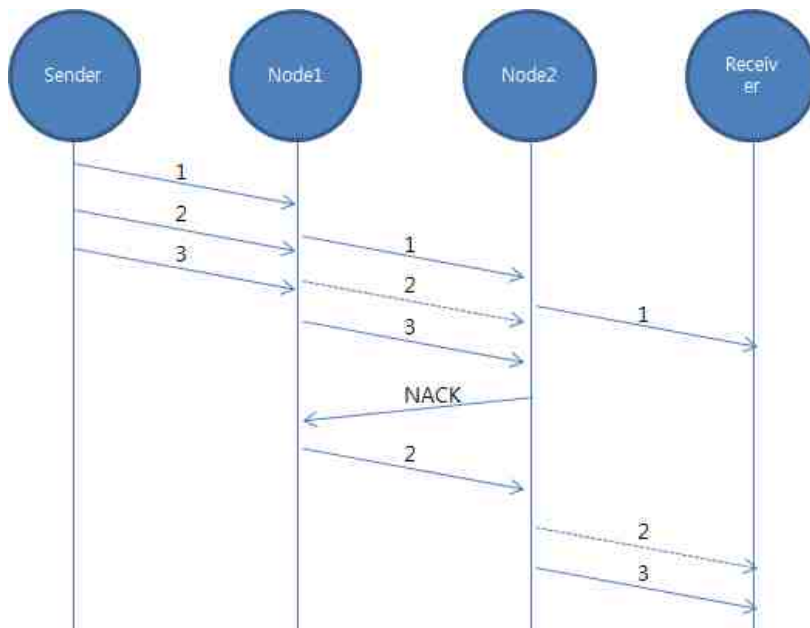


그림 2. PSFQ의 전송방식

## 2. RMST 프로토콜

RMST 프로토콜[6]은 upstream 방향의 전송에 대해 신뢰성을 보장하는 프로토콜이다. 이 기법은 end-to-end 데이터 패킷을 전송하는데 신뢰성을 제공하며 캐싱 및 복구를 위한 S-NACK (Selective Negative ACKnowledgement) 기반 프로토콜이다.

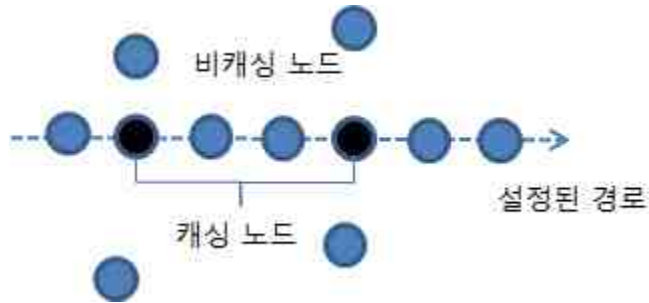


그림 3. RMST 프로토콜의 구성

RMST는 목적 노드까지의 경로가 설정되고 전송중인 노드를 캐싱 모드인 경우와 비 캐싱 모드인 경우로 나누고 캐싱 모드가 재전송의 책임을 담당하도록 한다[14]. 중간에 데이터의 불연속성이 발견되거나 손실이 발견되었을 경우 중간에 있는 수신자는 처음 sender에게 재전송을 요구하는 것이 아닌 중간의 캐싱 노드에게 재전송을 요구함으로써 빠른 재전송이 이루어지도록 한다. 시퀀스의 빈 곳이 발생하는 경우는 NACK 메시지를 통해 빈 곳에 대한 패킷을 요구하는 방법으로 복구를 하고 시퀀스가 잘려나가는 경우(다음 단편을 받기 위해 기다리는 타이머가 완료되었을 경우에 발생)는 전체 패킷을 다시 재전송하는 방법을 취한다.

RMST는 몇 가지의 문제점을 지니고 있다. 첫째, 전송순서가 보장되지 않는다. 따라서 실시간 전송을 보장하지 못한다. 둘째, 다른 기법들에 비해 에너지 소비가 크다. 셋째, 응용 수준에서는 신뢰성을 제공하지 못하는 특성이 있다.

### 3 ESRT 프로토콜

ESRT 프로토콜[7]은 기본적으로 센서 네트워크에서 모든 데이터가 센서 노드에서 싱크 노드로 집중된다는 상황을 가정한다. 무선 센서 네트워크에서 최소한의 에너지 소비와 혼잡 제어를 기반으로 신뢰성 있는 이벤트 싱크 전송을 제공한다[15]. 싱크 노드의 관심사는 어떤 노드가 어떤 데이터를 보낸다는 것보다는 어떠한 이벤트가 발생했는가 하는 것이 된다.

ESRT의 신뢰성은 싱크 노드에서 이벤트가 발생했다고 확신할 수 있는 센서 노드들로부터의 알림의 정도라고 할 수 있다. 따라서 ESRT에서는 이벤트의 발생을 알리는 리포트 전송률을 실제 싱크에서 기대하고 있는 신뢰성에 따라서 조절하도록 한다. 또한 센서가 보고하는 빈도수의 조절에 의해 희망했던 신뢰성을 보장하고자 end-to-end 전송을 한다.

ESRT의 문제점은 같은 값을 사용하여 모든 센서들의 보고 횟수를 조절하기 때문에 서로 다른 값을 사용할 때보다 성능이 떨어지는 문제점을 가지고 있다.

#### 제2절 트래픽 기반 혼잡 제어

##### 1. CODA 프로토콜

CODA 프로토콜은 UPSTREAM 방향의 프로토콜이다. CODA는 3가지 주요 컴포넌트가 있다[8]. 3가지 주요 컴포넌트는 congestion detection, open-loop hop-by-hop backpressure, closed-loop end-to-end multi-source regulation 을 말한다.

CODA는 현재 버퍼 점유량과 무선 채널 로드 모니터링에 의해 혼잡 검출을 시도한다. 버퍼 점유량 또는 무선 채널 로드가 임계치를 초과하게 되는 것은 혼잡이 발생하였음을 의미한다. 혼잡을 검출한 노드는 open-loop hop-by-hop backpressure 방법으로 전송률을 감소시키기 위해 그 노드의 downstream 방향의 이웃한 노드들에게 피드백 메시지를 공지한다. CODA는 closed-loop end-to-end 기법을 통해 multi-source 속도를 노드 스스로 조절할 수 있다.

센서 노드는 싱크 노드로부터 전송률을 유지하기 위해 주기적으로 ACK를 받는다.



만약 혼잡이 발생되면 ACK를 정해진 시간 내에 받지 못하게 된다. 이경우에 노드 스스로 전송률을 낮춘다.

CODA 프로토콜의 문제점은 센서들에서 싱크로의 단일 방향 제어만 한다. 데이터 신뢰성을 고려하지 않는다. Close-loop multi-source 환경에서 싱크에서 발생된 ACK 메시지의 손실 가능성으로 인한 네트워크 혼잡 때문에 지연 또는 응답 시간은 증가한다. 같은 문제점들이 있다.

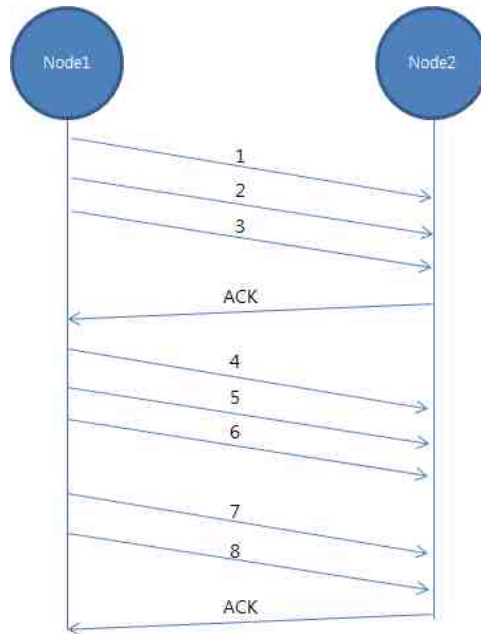


그림 4. CODA의 전송방식

## 2 STCP 프로토콜

STCP는 제어된 가변 신뢰성, 혼잡 탐지와 회피를 제공하는데 있어서 싱크 노드가 대부분 기능을 수행한다. 각 센서 노드는 흐름 종류나 전송률, 요구되는 신뢰성 등을 고려하여 싱크 노드들에게 데이터를 전송한다[9].

STCP는 응용마다 다른 특성을 고려하여 각 노드는 다양한 응용에 적합할 수 있도록 다양한 데이터 흐름을 제공하며, 혼잡 제어와 회피, 신뢰성 같은 기능을 추가적으로 수행한다. STCP는 센서 노드가 센싱 데이터를 보내기 전에 세션 초기 패킷을 싱

크 노드에게 전송한다.

세션 초기 패킷은 데이터 흐름, 지역 클럭 정보(전송시간), 데이터 흐름의 타입, 전송률, 신뢰성 등으로 구성되어 있다. 싱크 노드에서는 세션 초기 패킷의 정보를 저장하고 ACK신호를 각 센서 노드에게 전송하여 실제 센싱 데이터를 전송시킨다.

STCP의 헤더정보에는 데이터 패킷의 순서를 위한 시퀀스 넘버 필드가 있고 패킷의 특성을 확인하고, 흐름 종류를 나타내는 Flow ID, 혼잡 탐지와 회피를 지원하기 위해 CN (Congestion Notification) 비트가 있다.

표 1. STCP의 헤더 정보

Sequence Number (16 bits)	Flow ID (6 bits)	CH (1bit)	Option (7 bits)	Clock(32 bits)
------------------------------	---------------------	--------------	--------------------	----------------

Clock 필드에는 패킷이 전송되는 지역 시간 정보가 있다. 센서 노드에게 데이터를 전송받은 싱크 노드는 ACK 패킷을 센서 노드에게 전송한다.

표 2. STCP의 헤더정보(ACK/NACK)

Sequence Number (16 bits)	Flow ID (6 bits)	CH (1 bit)	ack/ nack( 1 bit)	Option (6 bits)	Clock(32 bits)
------------------------------	---------------------	---------------	-------------------------	--------------------	----------------

STCP의 흐름에는 두 가지(연속, 이벤트 기반)이 있다. 첫 번째로 연속 흐름은 다음과 같은 순서를 따른다.

1. 싱크노드가 센서노드의 전송률을 센서 노드로부터 받아서 기대한 시간 내에 도착하지 않는 패킷에 대해 NACK 신호를 센서 노드에게 전송한다.
2. 센서 노드에서는 전송되는 패킷을 버퍼에 저장한다.
3. 센서 노드에서는 버퍼의 오버플로우를 예방하기 위해 버퍼 타이머를 유지시킨다.
4. 버퍼 크기가 임계값에 도착되면, 버퍼 타이머와 버퍼를 클리어 시킨다.
5. 버퍼 타이머는 현재 네트워크 상태와 패킷 전송률을 고려하여 설정된다.
6. 만약 패킷이 빈번하게 손실되면 재전송을 위한 버퍼 타이머가 증가된다.

흐름-이벤트기반은 다음 3가지 순서를 따른다.

1. 센서 노드는 요구되는 신뢰성을 기반으로 전송되어진 패킷의 ACK 신호를 기다리게 된다.
2. 센서 노드는 각 전송 패킷을 ACK 신호를 받을 때까지 버퍼에 저장하며 ACK 신호가 도착하면, 대응되는 패킷을 버퍼에서 삭제하게 된다.
3. 버퍼 타이머는 주기적으로 클리어 된다.

STCP의 혼잡 명시 기법은 다음과 같다.

1. 센서 노드의 버퍼가 임계치에 도착하면 CN 비트 필드를 설정하고, 전송되는 모든 패킷을 CN비트가 설정된 값으로 전송한다.
2. CN비트가 설정된 패킷을 받은 싱크 노드는 혼잡이 발생했다고 판단하여 ACK 신호 패킷에 CN 비트를 설정하여 경로에 있는 노드들에게 혼잡된 경로를 전송하여 알린다.
3. ACK 신호에 혼잡 비트가 설정된 패킷을 받은 노드는 다른 경로를 선택하거나 전송률을 낮추어 전송한다. STCP는 혼잡제어와 신뢰성을 동시에 만족시키지만 복잡성 때문에 에너지소모량이 크다는 단점이 있다.

### 3 ECN 프로토콜

ECN은 통신망의 혼잡 발생 가능성을 미리 탐지하여 이를 송신측에 명시적으로 알려 전송속도를 조절하는 방식이다[10]. 혼잡 발생이 예상되면 도착하는 패킷 헤더의 CE (Congestion Experienced) 비트를 1로 마킹하여 목적지로전송 패킷이 목적지에 도착하면, 수신자는 중간 노드의 혼잡정보를 송신자에게 알려주기 위해 CE 비트를 헤더에 존재하는 ECN-Echo플래그에 그대로 복사하여 ack 패킷 전송 ECN-Echo 플래그가 세팅된 ack를 수신한 송신자는 자신의 송신 윈도우의 크기를 반으로 감소하는 등의 적절한 혼잡 제어 메커니즘을 수행하고 다음에 전송할 패킷에 Congestion Window Reduced 플래그를 세팅하여 보냄으로써 혼잡에 대비하여 전송을 감소시켰다는 것을 수신자에게 알려주어, 수신자가 더 이상 혼잡을 예고하는 ack을 보내지 않도록 한다.

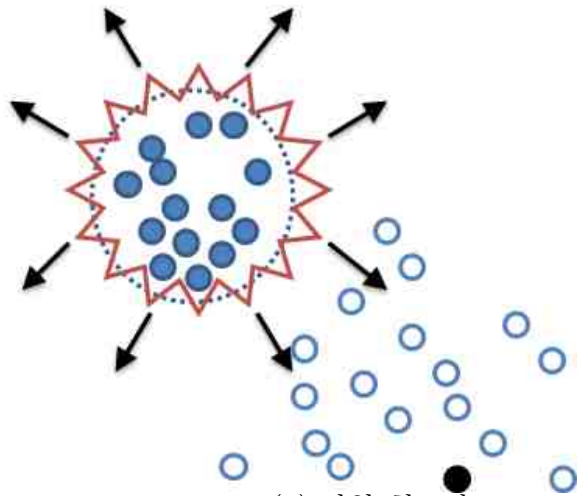
### 제3절 자원 기반 혼잡 제어

## 1 TARA 프로토콜

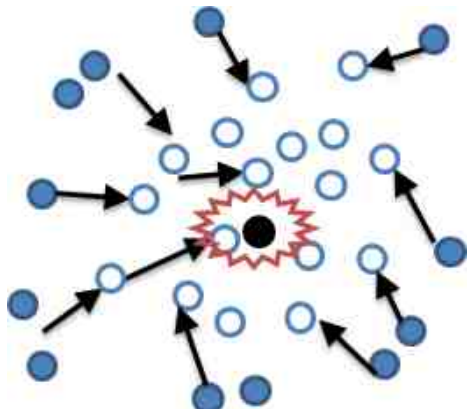
TARA는 리소스 제어 기법을 사용한 혼잡 제어 기법으로 혼잡 제어를 위하여 휴면 중인 임의의 수의 노드를 활성화 시켜서 우회경로로 사용한다. 네트워크상의 휴면 노드를 이용하기 위해서는 네트워크의 토폴로지를 인지하는 것이 중요하다. TARA는 다양한 혼잡 발생 토폴로지를 가정하여 그에 알맞은 트래픽 분산 방법을 제시한다. 혼잡상태는 그림 5와 같은 상황에서 발생한다. 그림 5의 (a)는 센싱 노드들이 한 곳에 집중되어 혼잡을 발생시키는 상황이고 (b)는 싱크 노드로 데이터가 몰리면서 발생하는 핫스팟 상황 마지막으로 (c)는 데이터를 전송할 때 겹치는 부분이 생기면서 발생하는 핫스팟이다. 그러나 토폴로지 인지를 위한 오버헤드가 크고, 트래픽 분산을 위한 정확한 알고리즘이 제시되지 않았다는 단점을 갖는다[11].

## 2. TADR 프로토콜

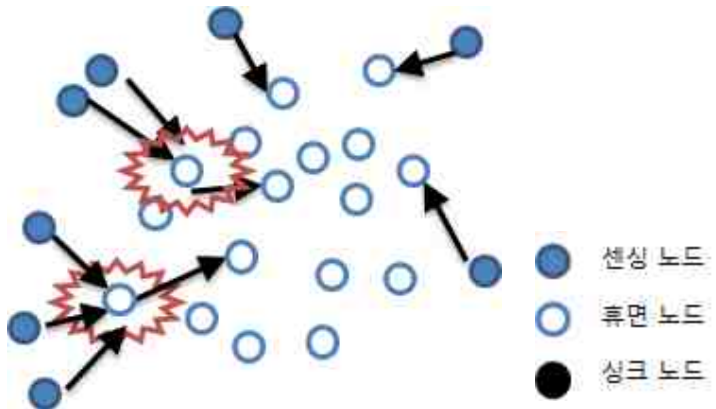
TADR은 포텐셜 기반의 트래픽 인지동적라우팅 알고리즘 네트워크를 포텐셜 필드로 정의하고 혼잡발생시 혼잡 지점을 인식하여 이를 우회하는 경로를 사용하여 혼잡을 회피한다. 기본적인 라우팅 백본 설정을 위해 각 노드의 깊이를 사용하고 혼잡 감지 및 트래픽 인지를 위하여 큐의 길이를 사용한다. TADR은 혼잡의 회피를 위해 하나의 대체경로만을 사용하게 되므로 네트워크 토폴로지의 리소스를 최대한 사용하지 못하는 단점을 갖는다[12].



(a) 자원 핫스팟



(b) 싱크 핫스팟



(c) 교차 핫스팟

그림 5. 혼합제어 핫스팟 시나리오

#### 제4절 DCA 프로토콜

AS-MAC[16] 프로토콜에서는 듀티사이클을 정의하기 위해 두 가지 파라미터를 고려한다. Uptime은 노드가 활성상태를 유지하는 시간을 의미하고, Wake-up Interval은 두 활성상태 사이의 시간 간격을 의미한다.

각 센서 노드는 주기적으로 활성 상태와 수면 상태를 반복하면서 전력을 공급하여 통신을 수행하고 수면 상태일 때는 무선 송수신기의 전력을 차단하여 통신에 의한 에너지 소모를 줄일 수 있다.[17]

Uptime이 짧을수록 에너지 효율성은 증가하며 Wake-up Interval은 길수록 더 많은 시간동안 수면 상태를 유지할 수 있어 에너지 효율성이 증가한다.

DCA기법은 듀티사이클 조절을 이용한 혼잡제어 기법으로 MAC 계층의 수신 패킷 정보와 듀티사이클 정보를 이용하여 동작한다. DCA 기법에서의 MAC 프로토콜은 비동기 듀티사이클에 기반을 둔 프로토콜로 가정한다. 또한 네트워크는 트리 기반의 토폴로지로 구성된다고 가정하여 부모 노드는 자식 노드들의 데이터 전송률에 기반을 두어 혼잡을 감지하고 자신의 듀티사이클을 조정하거나 자식의 전송률을 조절하여 혼잡을 회피한다[13].

#### 제5절 LEACH 클러스터링

LEACH (Low Energy Adaptive Clustering Hierarchy)[18]는 대표적인 클러스터 기반 라우팅 프로토콜로써, 노드사이의 에너지 소모를 균형있게 만들기 위해서 정규 분포 확률 기반으로 균등한 횟수로 클러스터 헤드를 선정한다. 클러스터 헤드는 TDMA 스케줄을 작성하여 멤버노드들에게 알려주고, 데이터를 수신 받아 싱크 노드에 전송한다. LEACH에서의 데이터 송수신은 CDMA방식으로 중간 노드 중계 없이 직접 전송한다. 그러나 LEACH는 헤드가 확률적으로 선출되므로 센서 노드가 에너지를 균형적으로 소모되지 못하고 네트워크 확장시에 노드간의 전송거리에 따른 에너지 소모가 더욱 커지며 클러스터 내의 멤버 노드들의 에너지 소모도 크게 증가한다. 헤드의 확률적 선출은 수식 (1)과 (2)에 의해 선출된다. P는 전체노드에서 헤드가 선출되는 확률을 나타내고 집합 G는 헤드로 선출되지 않은 노드들의 집합이다. 전체 노드들은 0과 1사이의 난수를 생성하고 임계값인 T(n)보다 작게 되면 클러스터 헤드로 선출된다. T(n)은 수식 (1)로 계산한다. 수식(1)에서 n은 집합 G에 들어있는 노드

의 수이고  $r$ 은 매 라운드를 의미하고  $P$ 는 전체 노드 수에서 헤드로 선출된 확률을 의미한다. 그림 6처럼 수신에너지가 가장 큰 근접 노드들이 선출된 헤드와 클러스터링을 이루게 된다.

$$T(n) = \frac{P}{1 - P(r \bmod \frac{1}{P})}, \forall n \in G \quad (1)$$

$$T(n) = 0, \forall n \notin G \quad (2)$$

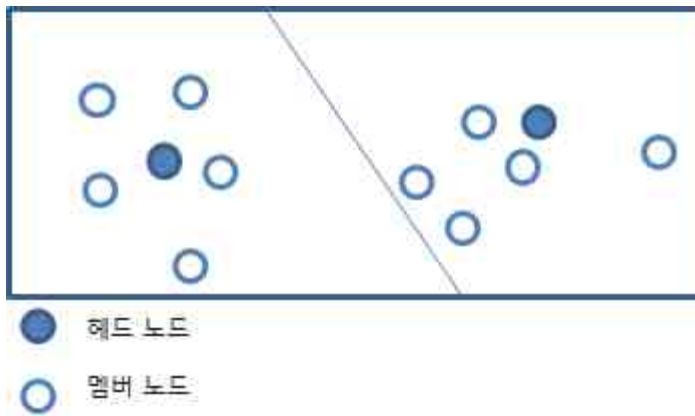


그림 6. LEACH 클러스터링 결과

### 제3장 노드 밀도 기반 트랜스포트 프로토콜

#### 제1절 듀티 사이클

듀티 사이클은 MAC 프로토콜에서 에너지 소모를 줄이는 효과적인 방법으로 널리 사용되고 있다. 무선 센서 네트워크에서 에너지를 절약하기 위하여 센싱이나 전송을 하지 않는 시간 동안 주기적으로 노드를 수면 상태로 만들어 에너지 소모를 줄이게 되는데, 주기적으로 깨어날 때 활성화 되어 있는 시간을 주기 시간으로 나눈 값을 듀티 사이클(duty cycle)이라고 한다. 그림 7에서 듀티 사이클은  $T_a$ 와  $T_p$ 의 비율 즉 ( $T_a/T_p$ )이다. 백분율로도 표기하는데 이 경우 ( $T_a/T_p$ ) × 100%이다.

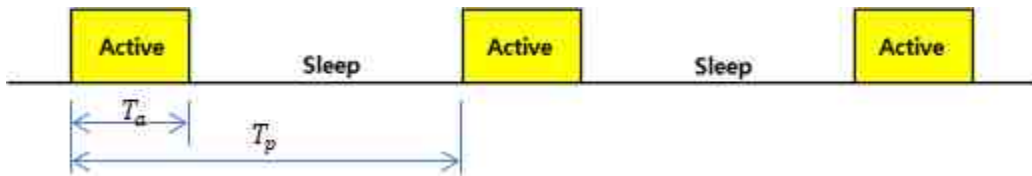


그림 7. 주기적인 센서 활성화 방식

그림 7에서  $T_a$ 는 노드의 활성화상태로 이 시간만큼 노드는 작업을 수행한다. 그리고 수면 상태는 전력공급이 중단되어 노드는 비활성상태로 된다.

표 3은 기존의 듀티사이클에서 사용되는 파라미터와 본 논문이 제시하는 혼잡제어의 파라미터가 모두 포함 되어있다.  $T_{a\_max}$ 는 노드의 최대 활성화상태이다. 노드의 작업량이 많아질 경우 최대  $T_{a\_max}$ 까지 활성화상태의 시간이 된다. 반대로  $T_{a\_min}$ 은 노드의 최소 활성화상태이다. 노드가 작업을 처리하는 최소의 시간이  $T_{a\_min}$ 이 된다.  $T\_inter\_arrival$ 는 클러스터 멤버노드로부터 클러스터 헤드노드까지의 전송되는 패킷의 평균 도달 시간이다. 이 값은 노드의 이벤트 발생횟수 또는 혼잡상태에 따라 변한다.  $P$ 는  $T_a$ 시간동안 처리할 수 있는 패킷량이다.



표 3. 수식에 사용되는 파라미터

파라미터	의미
$T_a(x)$	클러스터 헤드 x의 활성화 주기 내 활성화 시간(sec)
$T_p(x)$	클러스터 헤드 x의 활성화 주기
$T_{a\_max}(x)$	클러스터 헤드 x의 활성화 주기 내 최대 활성화 시간(sec)
$T_{a\_min}(x)$	클러스터헤드 x의 최소 활성화 시간(sec)
$T_{a\_next}(x)$	클러스터 헤드 x의 다음 활성화 시간(sec)
$s(x)$	노드x에서 요구되는 서비스 시간(sec)
$T\_inter\_arrival(i)$	노드 I로부터 수신되는 패킷의 평균 도달 시간(sec)
$P(x)$	$T_a(x)$ 동안 클러스터 헤드 x가 싱크 노드에 전송한 패킷의 수
$N(x)$	클러스터 헤드 x의 멤버 노드의 수
$d(x)$	$T_a(x)$ 에서 $s(x)$ 를 뺀 여분의 잔여시간 ( $d(x) = T_a(x) - s(x)$ )
$E\_battery(x)$	클러스터 헤드 x의 에너지(Joule)
$E\_avg$	클러스터 헤드들의 평균 잔여 에너지(Joule)
$Q_{length}(x)$	클러스터헤드 x의 queue 길이
$Q_{length\_next}(x)$	클러스터헤드 x의 queue의 다음 길이

## 제2절 클러스터 헤드 자원 제어

NDTP는 Duty-cycle을 기본 베이스로 사용한다. 노드 $x$ 에서 요구되는 서비스 시간을  $s(x)$ 라고 하였을 때 다음과 같은 식으로 계산한다.

$$s(x) = \sum_i^N \frac{1}{T_{\cap-arrive(i)}} * T_a(x) * \frac{1}{P} \quad (3)$$

$$d(x) = T_a(x) - s(x) \quad (4)$$

$d(x)$ 가 0보다 작을 경우는 트래픽이 초과되어 혼잡이 발생 되었다고 볼 수 있다.  $d(x)$ 를 계산 후에 클러스터 헤드는 4가지 판단을 한다. 수식 (3)과 같이  $d(x)$  0보다 클 경우에는  $s(x)$ 를  $T_{a\_min}(x)$ 와 비교하여  $s(x)$ 가 더 클 때 다음  $T_a(x)$ 의 값은  $s(x)$ 가 되고 작거나 같을 경우에는 다음  $T_a(x)$ 의 값은  $T_{a\_min}(x)$  값이 된다. 수식 (4)와 같이  $d(x)$ 가 0보다 작거나 같을 경우에는 혼잡 상태로 보고  $s(x)$ 를  $T_{a\_max}(x)$ 와 비교하여  $s(x)$ 가 더 작을 때 다음  $T_a(x)$ 의 값은  $s(x)$ 가 되고 크거나 같을 경우에는 클러스터 헤드노드의 queue 크기를 줄여 헤드가 많은 데이터를 받지 않도록 리소스를 조절한다.

노드들의 전송을 막는 방법이 아닌 헤드의 queue의 길이를 조절하기 때문에 노드의 밀도에 큰 영향이 없다. 클러스터 내 노드의 수가 많아서 노드 밀도가 높아져도 헤드에서는 정해진 숫자의 데이터만 받고 나머지 데이터를 무시하기 때문에 헤드에 과부하가 걸릴 일이 적어지게 된다. 또한 클러스터 헤드의 queue 길이를 줄이면 클러스터 헤드가 처리하는 데이터 량이 감소되므로 그만큼의 에너지를 아낄 수 있다. 그리고 queue 길이가 최소가 되었을 때 헤드에 과부하가 걸릴 경우 클러스터 내의 노드 밀도가 높은 것으로 인식하고 클러스터 내에서 임의의 노드들을 수면 상태로 만들어서 혼잡을 조절 한다. queue의 길이를 줄여서 들어오는 데이터가 적어지지만 주변의 센싱 데이터들을 파악하는 것에는 큰 문제가 없을 것으로 보고 있다. 트래픽이 과도한 클러스터 그룹은 대체로 노드들의 밀도가 높기 때문이다. 그래서 몇 몇의 클러스터 멤버노드의 데이터를 받지 않더라도 주변의 다른 노드들의 데이터를 통해 충분한 주변 환경을 센싱할 수 있다.

$$\begin{aligned} & \text{if } d(x) > 0 \\ & \left\{ \begin{array}{l} T_{a\_next}(x) = s(x) \quad , s(x) > T_{a\_min}(x) \\ T_{a\_next}(x) = T_{a\_min}(x) , s(x) \leq T_{a\_min}(x) \end{array} \right. \quad (5) \end{aligned}$$

$$\text{if } d(x) \leq 0 \begin{cases} T_{a\_next}(x) = s(x) & , s(x) < T_{a\_max}(x) \\ Q_{length\_next}(x) = Q_{length}(x) * \frac{1}{2} & , s(x) > T_{a\_max}(x) \end{cases} \quad (6)$$

그림 8은 위 클러스터 헤드에서 어떠한 흐름으로 수식 (5), (6)을 적용하는지를 나타낸 그림이다. 혼잡도 계산 후 수식에 따라  $T_{a\_next}(x)$ 를 변화 시킨 후 데이터 송수신을 진행 한 후 미리 설정한 일정한 주기에 따라  $s(x)$ 와  $d(x)$ 를 다시 계산 한 후 다음  $T_{a\_next}(x)$  값을 정한다.

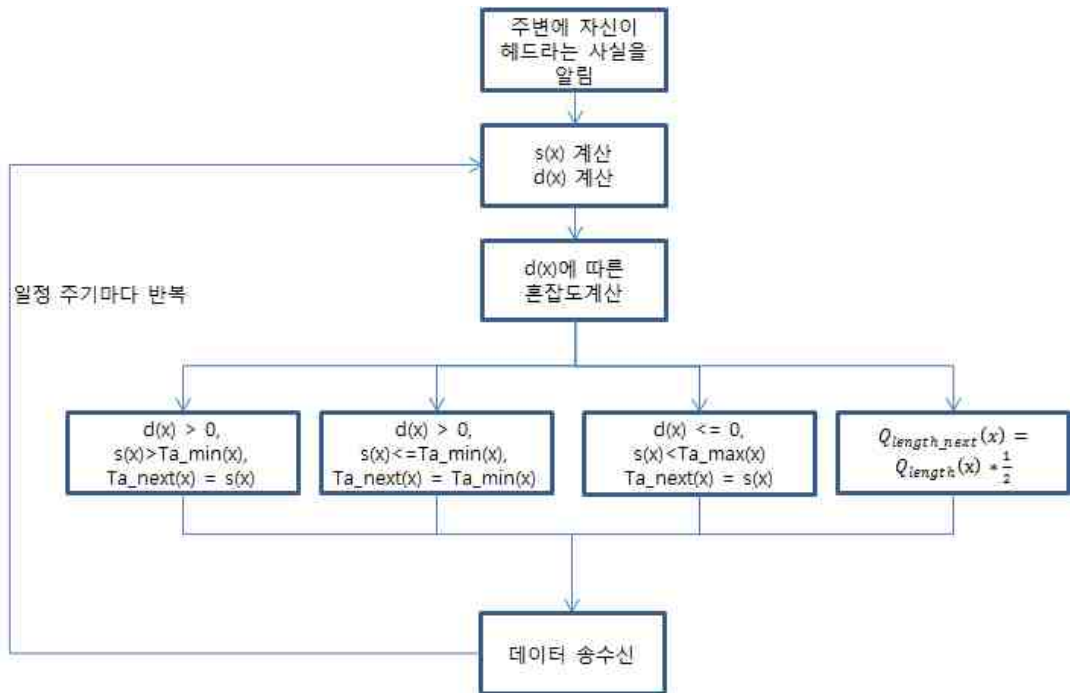


그림 8. 클러스터 헤드 자원 제어 흐름

NDTP의 혼잡제어 방식에는 AIMD방식을 적용하여  $T_a$  값이 최대값이 될 경우 클러스터 헤드에서 받는 리소스를 절반으로 줄여 바로 혼잡을 회피할 수 있도록 하였다. 그림 9과 같이 클러스터 헤드가 몇몇 노드의 데이터를 받지않고 무시함으로써 헤드에 오버플로우가 발생하지 않아 에너지소모를 줄이고 전송성공율을 높일 수 있다.

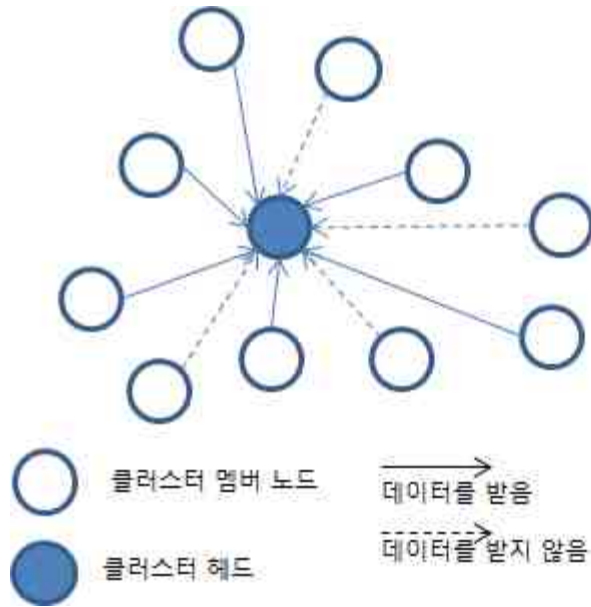


그림 9. 클러스터 헤드의 자원 제어 방식

### 제3절 클러스터 헤드 에너지 조절

에너지 효율은 무선 센서 네트워크에서 네트워크 수명에 큰 영향을 주기 때문에 가장 중요한 지표중 하나이다[19]. 무선 센서는 환경상 회수가 불가능하기 때문에 모든 노드들이 동시에 수명이 다하는 것을 이상적으로 보고 있다. 그러므로 노드들의 에너지 체크를 통한 에너지 밸런스 유지를 통해 노드들의 에너지 밸런스를 맞추는게 중요하다. 그림 10과같이 클러스터 헤드들은 주기적으로 싱크 노드에게 자신의 에너지 값을 전달한다. 노드의 1라운드마다 에너지를 체크하여 싱크 노드에게 데이터를 보낼 때 energy값을 보내게 된다. 싱크 노드는 받은 에너지 값 데이터들을 비교하여 에너지 잔여량이 적은 클러스터 헤드에게 ack메시지에 1비트를 이용하여  $T_{a\_max}(x)$ 의 값을 줄이도록 제어한다. 싱크 노드에서 E\_battery의 값을 비교하여 ack메시지에 감소(1)또는 유지(0) 비트를 담아서 노드들의 D\_max(x)값을 감소시킨다. D\_max(x)값을 줄인 클러스터 헤드는 다른 헤드들에 비해 활성화하는 최대 시간이 적어지게 되어 다른 헤드들과의 에너지 밸런스가 유지된다.

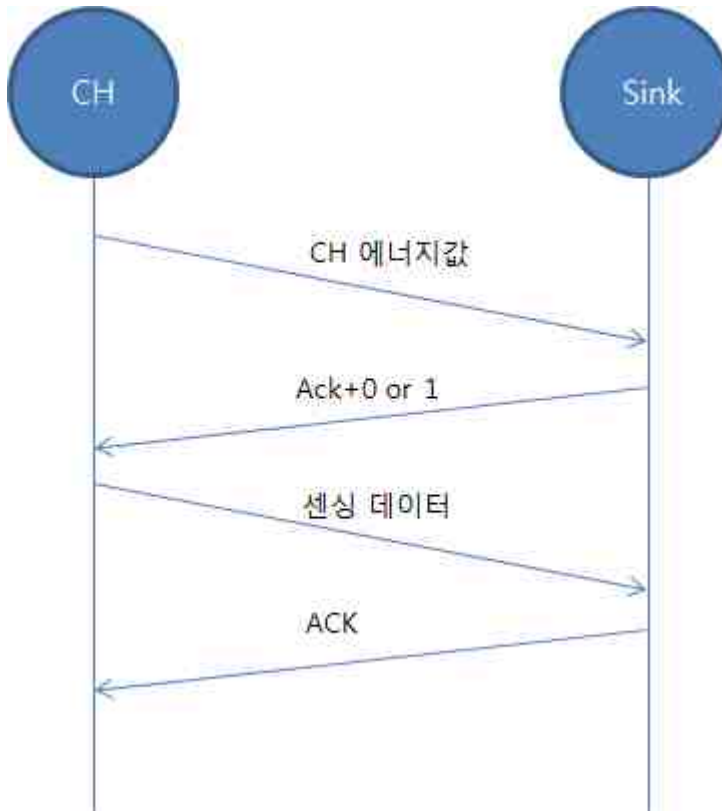


그림 10. 클러스터 헤드 에너지값 전송

그림 11은 클러스터 헤드의 에너지조절 흐름도를 나타낸 그림이다. 각 라운드마다 헤드로 선출되면 주변에 자신이 헤드인걸 브로드캐스팅을 통해 알리고 싱크 노드에 자신의 에너지 값을 전송한다.

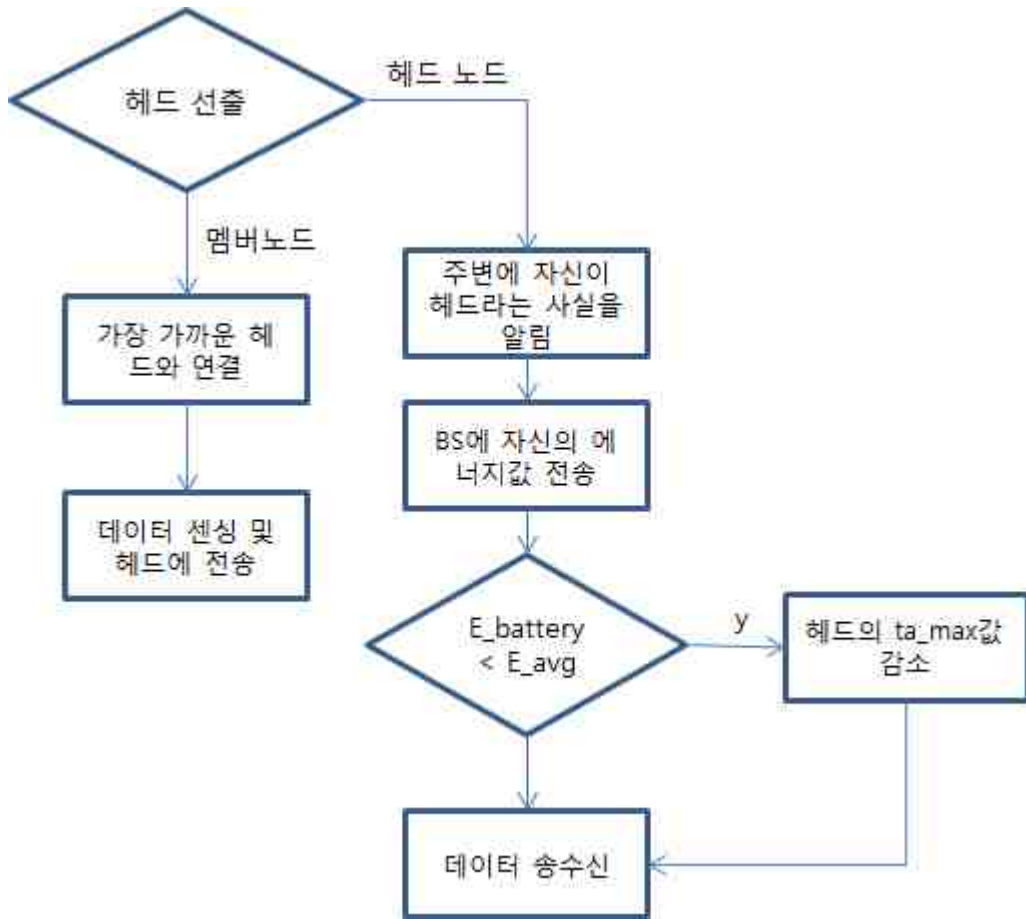


그림 11 클러스터 헤드 에너지 조절 흐름도

## 제4장 성능 평가

이 장에서는 NDTP의 성능평가를 기술한다. LEACH 클러스터링 방식을 적용시켜 기존의 DCA 방식과 NDTP 프로토콜을 같은 환경에서 비교한다. 라운드 수와 밀도에 따른 노드의 생존 수와 클러스터 헤드 수에 따른 데이터 전송실패율을 비교하였다.

### 제1절 시뮬레이션 환경

성능평가 파라미터는 표 4에 요약되어 있다. 평가는 라운드 수, 클러스터 수, 클러스터 내 노드 수, 초당 패킷 송신량에 따른 노드 수  $T_{a,max}$  도달횟수, 헤드의 패킷 전송 실패율, 클러스터 내 패킷 전송실패율을 나타낸다. 본 실험에서는 LEACH source 코드에 의해 제공되는 에너지 소비 모델[20]을 사용하였다. 클러스터 헤드 노드와 싱크 노드간의 거리를 측정하여 거리가 일정거리 이상일 경우 거리의 4제곱을 곱하고 일정 거리 이하 일 경우에는 거리의 제곱을 곱한다. 시뮬레이션은 Java 프로그래밍과 matlab을 이용하였다.

표 4. 시뮬레이션 파라미터

파라미터	값
$E_{da}$	5 nJ/bit/signal
$E_{elect}$	50 nJ/bit
$E_{sense}$	5 nJ/bit
$\epsilon_{fs}$	10 pJ/bit/ $m^2$
$\epsilon_{mp}$	0.0013 pJ/bit/ $m^4$
Initial energy	2 J
Sink Location	(125, 75)
Network area	100 × 100 $m^2$
Number of nodes	100, 200
Transmission range	136 m
packet outgoing rate	1, 3, 5, 7, 9 packet/sec

$$E_{Tx}(k,d) = E_{tx - elect}(k) + E_{Tx - amp}(k,d) \quad (5)$$

$$= \begin{cases} kE_{elect} + k\epsilon_{fs}d^2, & d < d_0 \\ kE_{elect} + k\epsilon_{mp}d^4, & otherwise \end{cases}$$

식(5)에서 elect는 디지털 코딩, 변조, 필터링 된 신호의 확산 등의 요인에 따른 무선 전자 전송/수신 에너지 이다.  $\epsilon_{fs}d^2$ 와  $\epsilon_{mp}d^4$ 허용 가능한 수신기 및 비트 에러율 가의 거리에 따른 증폭기 에너지의 상수 값이다.

### 제2절 시뮬레이션 결과

그림 12와 13은 라운드 수(시간)에 따른 살아있는 노드의 수를 나타낸다. 초기 노드를 100개와 200개로 시작하여 비교하였다. NDTP의 에너지 효율 방식과 혼잡제어 방식을 통해 기존의 DCA보다 노드들의 수가 좀 더 오래 살아있는 걸 볼 수 있고 또한 에너지 균형을 맞춰줌으로써 노드들의 사망 시간이 대부분 일정한 것으로 나타난다.

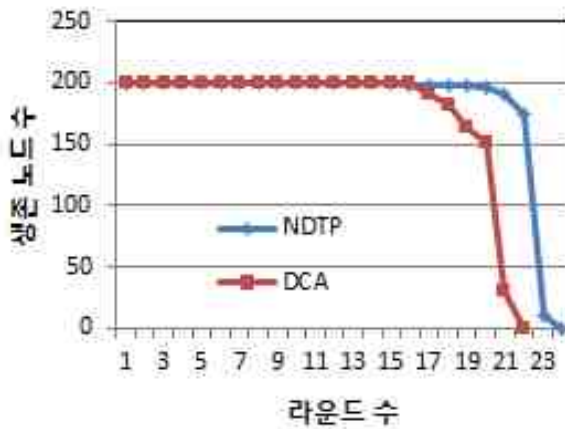


그림 12. 라운드 수에 따른 생존 노드의 수(초기 노드수 = 200)



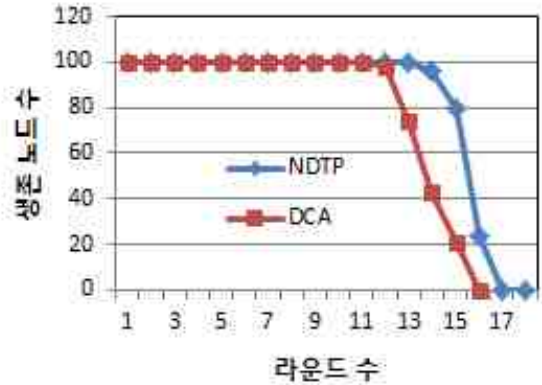


그림 13. 라운드 수에 따른 생존 노드의 수(초기 노드수 = 100)

그림 14는 NDTP와 DCA의 cycle에 따른  $T_a$  값의 max값 도달 횟수를 나타낸 그림이다. DCA방식에서는 밀도가 높을수록  $T_a$ 가 max값에 도달하는 횟수가 늘어나는 반면 NDTP는 앞서 나온 내용처럼 밀도의 영향을 크게 받지 않고 혼잡을 제어하는 모습을 볼 수 있다.

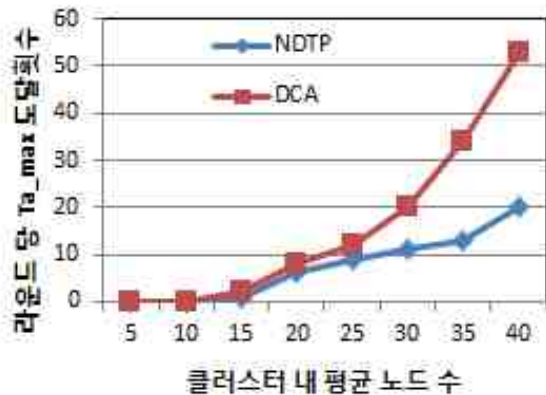


그림 14. 클러스터 내 노드 수에 따른 혼잡도

그림 15는 클러스터 내 노드 수에 따른 패킷 전송 실패율을 나타낸다. 기존의 방식인 DCA방식과 NDTP방식의 큰 차이가 없음을 볼 수 있는데 그 이유는 같은 환경에서 NDTP와 DCA는 같은 전송 방식을 사용하고 NDTP는 패킷 전송 실패율을 낮추는 기법이 아니라 queue 길이를 조절하여 일부 데이터를 받지않는 방식이기 때문에

클러스터 내 패킷 전송 실패율은 큰 차이가 없음을 볼 수 있다.

그러나 그림 16의 그래프를 보면 초당 패킷 송신량이 많아지게 될 경우 NDTP가 DCA방식에 비해 좀 더 낮은 패킷 전송 실패율을 보였다. 초당 패킷 송신량이 많아지게 되면 네트워크의 혼잡 빈도가 높아지게 되는데 이 때 혼잡 제어를 효율적으로 하는 NDTP가 기존의 방식에 비해 낮은 패킷 전송율을 보이게 된다. 초당 패킷 송신량에 따른 혼잡 빈도는 그림 17을 통해 볼 수 있다.

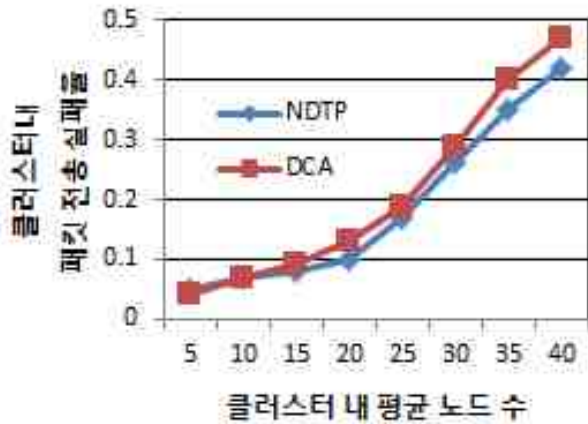


그림 15. 클러스터 내 노드 수에 따른 패킷 전송 실패율

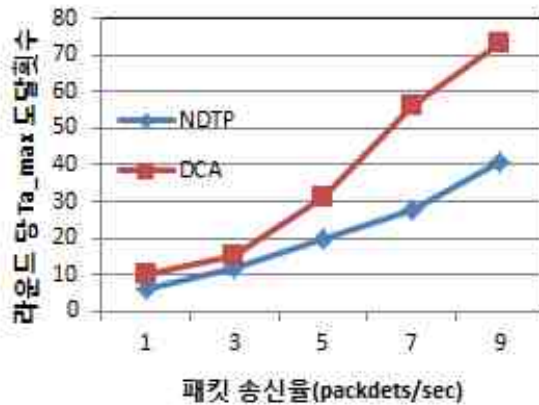


그림 16 패킷 송신율에 따른 혼잡도

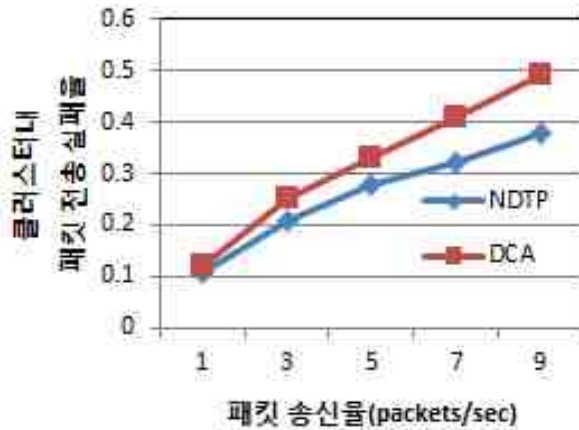


그림 17. 패킷 송신율에 따른 패킷 전송 실패율

그림 18은 클러스터 수에 따른 헤드와 싱크 노드간의 전송 실패율을 나타낸다. 멤버노드가 클러스터 헤드 노드에게 전송하는 정보보다 클러스터 헤드가 싱크 노드에 전송하는 데이터가 더 중요성이 높기 때문에 클러스터 헤드 수에 따른 헤드의 패킷 전송실패율을 측정하였다. 헤드 수가 작은 경우는 멤버의 노드 수가 많기 때문에 전송실패율이 높은 편이고 헤드 수가 높은 경우는 멤버 노드 수가 작기 때문에 헤드의 부하가 적어 전송실패율이 낮다. NDTP는 DCA에 비해 노드 밀도에 크게 영향을 받지 않기 때문에 헤드에 오버플로우 같은 무리가 가지 않아 전송 실패율이 현저하게 낮은 걸 볼 수 있다.

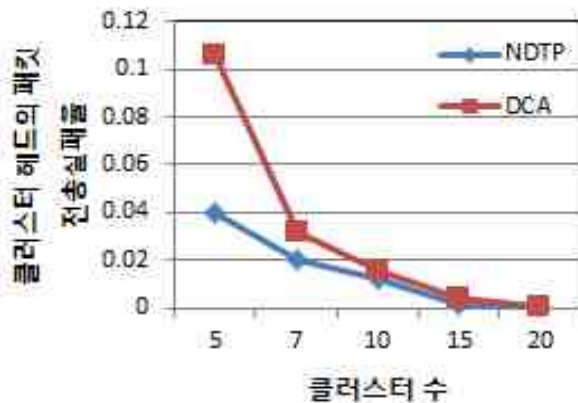


그림 18. 클러스터 수에 따른 클러스터 헤드의 패킷 전송 실패율

그림 19는 클러스터 수에 따른 Ta\_max 도달횟수를 나타낸다. 클러스터 수가 적으면 그만큼 한 클러스터에 속한 멤버노드 수가 많아지기 때문에 네트워크가 자주 혼잡하다. 그러나 queue 길이를 조절하여 혼잡제어를 하는 NDTP가 기존의 DCA 방식보다 훨씬 더 안정된 네트워크 환경을 구성하는 걸 볼 수 있다. 기존의 방식인 트래픽 기반 제어는 전송속도를 늦춰서 혼잡을 제어하지만 클러스터 내 노드 수가 많을 경우 그에 대한 한계가 존재한다. 그러나 자원 기반 방식을 통하여 노드 수가 많은 네트워크 내에서 혼잡 빈도가 확실히 줄어드는 걸 볼 수 있다.

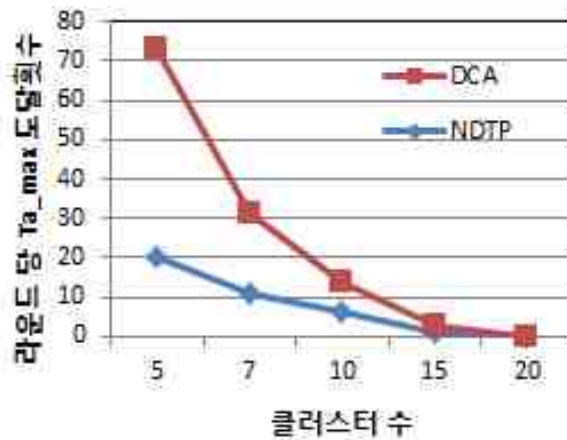


그림 19. 클러스터 수에 따른 혼잡도

## 제5장 결 론

본 논문에서는 노드의 밀도에 관계없이 클러스터 헤드가 에너지를 관리하고 또한 클러스터 헤드의 에너지를 체크하여  $Ta_{max}$ 값을 조절함으로써 클러스터 헤드의 수명을 좀 더 늘릴 수 있는 방안을 연구하였다. 클러스터 헤드는 일반 노드들 보다 에너지 소모량이 많기 때문에 다른 노드들보다 클러스터 헤드의 수명은 더 빨리 소모하게 된다. 그렇기 때문에  $Ta_{max}$ 값 조절 뿐 아니라 클러스터 헤드의 queue 길이 까지 제어하여 좀 더 에너지소모를 줄이는 방식을 구현하였다. queue 길이가 제어됨으로 인해 혼잡을 제어함과 동시에 클러스터 헤드와 싱크 노드간의 데이터 전송실패율도 낮출 수 있다. 제안한 NDTP 프로토콜은 트래픽 기반 제어방식이 아닌 자원 기반 혼잡제어 기법을 적용함으로써, 네트워크내의 혼잡을 줄이고 결과적으로 데이터의 전송 성공률을 높이는 효과를 보였다. NDTP는 기존의 DCA방식에 비해 클러스터 내 패킷 전송 실패율을 11%정도 감소 시키고 노드의 생존시간이 3라운드 만큼 늘어났다. 또한 NDTP의 핵심인 센서 노드 밀도가 높을수록 혼잡 빈도가 30%이상 감소하고 헤드의 패킷 전송실패율도 크게 감소하였다. 헤드의 패킷 전송율은 클러스터 내의 노드 전송율보다 더 중요한 부분이기 때문에 헤드의 패킷 전송 실패율의 감소는 필수적이다.

본 연구의 향후 과제로서 클러스터 내 노드의 전송 실패율을 개선하기위한 네트워크의 효율적인 전송방식의 프로토콜을 연구할 예정이다.

## 참고문헌

- [1] N. Aschenbruck, "STMP - Sensor data Transmission and Management Protocol," *Proc. of IEEE 36th Conf. on Local Computer Networks*, pp. 475-483, 2011.
- [2] V. C. Gungor, "A Real Time and Reliable Transport (RT)<sup>2</sup> Protocol for wireless Sensor and Actor Networks," *IEEE/ACM Trans. on Networking*, Vol. 16, No. 2, pp. 359-370, 2008.
- [3] F. Yunus, "Proposed Transport Protocol for Reliable Data Transfer in Wireless Sensor Network (WSN)," *Proc. of 2011 IEEE 4th Int. Conf. on Modeling Simulation and Applied Optimization*, pp. 1-7, 2011.
- [4] 송병훈, "고속 인터넷을 위한 새로운 TCP 혼잡 제어에 관한 연구," 광운대학교 대학원 박사학위 논문, 2004.
- [5] C. Wan, A. Campbell, and L. Krishnamurth, "Pump-slowly, Fetch-Quickly (PSFQ): A Reliable Transport Protocol for Sensor Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 23, No 4, pp.862-872, 2005.
- [6] F. Stann and J. Heidemann, "RMST : Reliable Data Transport in Sensor Networks, " *Proc. of 1st IEEE Int Workshop on Sensor Net Protocols and Application*, pp. 102-112, 2003.
- [7] Y. Sankarasubramaniam, O. Akan, and L. Akyildiz, "ESRT : Event-to-Sink Reliable Transport in Wireless Sensor Networks," *Proc. of MobiHoc 2003*, pp. 177-188, 2003.
- [8] C. Wan, S. Eisenman, and A. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks," *Proc. of Int. Conf. on Embedded Networked Sensor Systems*, pp. 266-279, 2003.
- [9] Y. G. Iyer, S. Gandham, and S. Venkatesan, "STCP: A Generic Transport Layer Protocol for Wireless Sensor Networks," *Proc. of IEEE ICCN 2005*, pp. 449-454, 2005.
- [10] K. K. Ramakrishnan, Sally Floyd, "A proposal to add Explicit Congestion Notification (ECN) to IP", Internet Draft-kksjf-ecn-03.txt, October, 1998. Congestion Notification", *Computer Communication Review*, V.24 N.5, pp.10-23, 1994.
- [11] J. Kang, Y. Zhang, and B. Nath, "TARA: Topology-Aware Resource

- Adaptation to Alleviate Congestion in Sensor Networks,” IEEE Transaction on Parallel and Distributed System, Vol.18, issue. 7, pp.919-931, 2007.
- [12] T. He, F. Ren, C. Lin, and S. Das, “Alleviating Congestion Using Traffic-Aware dynamic Routing in Wireless Sensor Networks,” IEEE SECON'08, pp.233-241, 2008.
- [13] 이동호, 정광수 “무선센서 네트워크에서 듀티사이클 조절을 통한 혼잡 제어 기법”, 한국통신학회논문지 제 35권 제1호 네트워크 및 서비스, pp154-161, 2010
- [14] 이원진, 김승천, “유비쿼터스 센서 네트워크에서 신뢰성 향상 기법,” 한국통신학회 학술대회논문집, pp. 1353-1356, 2008.
- [15] 박홍진, 장재복, “무선 센서 네트워크에서 신뢰성 있는 데이터 전송을 위한 혼잡 탐지와 회피,” 한국항행학회논문지 제14권 제3호, pp. 343-350, 2010.
- [16] B. Jang, J. Lim, and M. Sichitiu, “AS-MAC: An Asynchronous Scheduled MAC Protocol for Wireless Sensor Networks,” IEEE MASS'08, pp.434-441, 2008.
- [17] W. Ye, J. Heidemann, and D. Estrin, “Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks,” IEEE/ACM Transaction on Networking, Vol.12, No.3, pp. 493-506, 2004.
- [18] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” Proc. of Hawaii International Conference on System Science, Maui, Hawaii, pp. 10. 2000.
- [19] D. Kim. and J .Cho. “A method to support mobile sink node in a hierarchical routing protocol of wireless sensor networks,” Journal of the KICS, Vol.33, No.1, pp. 48-57. 2008.
- [20] W. Bo, H. Hu and F. Wen. "An improved LEACH protocol for data gathering and aggregation in Wireless Sensor Networks," Computer and electrical engineering, pp. 398-401, 2008.