

SAN 환경에서 클러스터 기반의 멀티채널 알고리즘

공준익*, 이성로^o

Cluster-Based Multi-Channel Algorithm in SAN Environments

Joon-Ik Kong*, Seong Ro Lee^o

요약

선박 네트워크는 다양한 장치들을 네트워크로 연결하여 선박 상태를 실시간 모니터링 하고, 유지보수 비용을 최소화할 수 있다. 특히, 선박 네트워크에서 저비용, 저파워, 다기능의 센서 노드를 활용한 무선 센서 네트워크의 연구는 많은 장점이 있다. 본 논문에서는 선박 내의 무선 센서 네트워크 환경에서 에너지 효율을 고려한 클러스터 기반의 멀티채널 알고리즘을 제안한다. 제안하는 알고리즘은 분산적이고, 동적인 채널 할당 알고리즘을 이용해 클러스터 단위로 채널을 할당하기 때문에 클러스터 간의 간섭을 회피하고 처리율 및 에너지 효율의 향상의 결과를 보인다.

Key Words : Ship Area Network, Wireless Sensor Network, Clustering Algorithm, Multi-Channel, Channel Allocation Algorithm

ABSTRACT

Ship Area Network(SAN) can monitor the status of ship in real time and minimize the maintenance costs by connecting various devices to the network. In particular, among researches on SAN, Wireless Sensor Network using sensor nodes that is low-cost, low-power, and multifunctional has a number of advantages. In this paper, we propose cluster-based multi-channel algorithm considering the energy efficiency in wireless sensor network in a ship. The proposed algorithm shows the result of improvement of throughput and energy efficiency, because it reduces interference between clusters by using channel allocation algorithm that is distributed and dynamic.

I. 서론

최근 IT 융합 기술의 발전으로 지능형 선박 기술 및 선박 네트워크(SAN : Ship Area Network)에 대한 연구가 진행되고 있다¹⁻²⁾. 선박 네트워크는 다양한 선박 장치 및 기기들을 하나의 네트워크로 연결하여 안전한 선박 운행 및 효율적인 유지 보수를 위한 실시간 정보 교환을 목적으로 하고, 보통 전력선, 이더넷, 케이블 등의 유선 네트워크와 Zigbee, WLAN,

Bluetooth, RFID 등의 무선 기술을 하나의 네트워크로 연결하여 원격 제어할 수 있도록 한다. 이로써 선박 상태에 대한 실시간 모니터링으로 각종 사고를 미연에 방지할 수 있고, 유지보수 비용을 최소화할 수 있다.

하지만 선박 네트워크를 위한 유선 네트워크는 인프라 구축에 있어서 몇 가지 문제점이 있다³⁻⁴⁾. 첫째로, 선박이 건조되기 전에 미리 설계하여 구축되어야 한다. 이미 건조된 선박에 유선 인프라를 구축하는 것

* 본 연구는 2014년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(NRF-2009-0093828)과 미래창조과학부 및 정보통신산업진흥원의 IT융합 고급인력과정 지원사업(NIPA-2014-H0401-14-1009)의 지원을 받아 수행된 연구결과임.

• First Author : Institute of Information Science and Engineering Research, Mokpo National University, joonik2000@gmail.com, 정회원

◦ Corresponding Author : Department of Electronics Engineering, Mokpo National University, srlee@mokpo.ac.kr, 정회원

논문번호 : KICS2015-04-109, Received April 2, 2015; Revised May 13, 2015; Accepted May 13, 2015

은 어려움이 따른다. 둘째로, 노드 수의 변화에 따른 대처가 어렵다. 셋째로, 불안정한 전압 및 전류로 인한 오작동으로 안전사고가 발생할 수 있다. 넷째로, 느린 전송속도로 인해 사고 발생 시 신속하고 효과적인 대처가 어려울 수 있다. 마지막으로, 선로 구축에 대한 비용적인 측면에서 비효율이 발생할 수 있다.

위와 같은 문제점으로 인해 노드의 설치가 쉽고 사용하기 편리한 무선 네트워크에 대한 연구가 필요하다. 특히 무선 센서 네트워크(WSN : Wireless Sensor Network)는 저비용, 저파워, 다기능의 센서 노드들이 센서로부터 획득한 데이터를 서로 협력하여 목적지까지 전송하고, 사용자가 다루기 쉽기 때문에 상품의 위치 및 재고관리, 주차장 관리, 산불·홍수와 같은 자연 재해 모니터링, 야생 동물들의 이동 경로 추적 등의 다양한 분야에서 활용된다^{5,6)}. 따라서 본 논문에서는 선박 네트워크에서 활용되는 무선 센서 네트워크를 고려한다.

무선 센서 네트워크는 제한된 에너지를 이용하기 때문에 센서 노드의 에너지 효율을 우선적으로 고려해야 한다. 센서 노드에서 소비 되는 에너지는 MCU에서 데이터 처리할 때 소모되는 에너지 보다 다른 노드와 데이터 통신을 할 때 더 많은 에너지가 소모된다. 이 원인은 Collision, Overhearing, Idle Listening, Control Packet Overhead로 구분할 수 있다. 따라서 이와 같은 에너지 소모 원인을 고려한 효율적인 알고리즘이 필요하다.

클러스터링 알고리즘은 네트워크를 클러스터 단위로 나누어 에너지 효율을 향상시키는 대표적인 알고리즘이고, 이 기법을 활용한 연구가 진행되어 왔다⁷⁾. 클러스터링 알고리즘은 각 센서 노드들 중에서 클러스터 헤드와 그의 멤버들을 결정하여 센서 네트워크를 여러 개의 군집 형태로 묶어 통신하는 방법이다. 클러스터 헤드는 멤버들로부터 수집된 데이터를 통합하여 전송하기 때문에 네트워크의 에너지 효율을 향상시킨다.

하지만 싱글 채널을 사용하는 클러스터링 알고리즘은 랜덤하게 분포되어 있는 센서 노드들이 클러스터를 구성하면서 클러스터 간의 통신 범위가 중첩되어 간섭이 발생할 수 있다. 클러스터 간의 간섭은 데이터가 전송될 때 충돌이 발생하여 데이터 전송의 실패를 유발하고, 그로 인한 데이터의 재전송으로 에너지 낭비가 발생한다.

이와 같은 인접한 클러스터 간의 간섭을 해결하기 위해 본 논문에서는 클러스터 기반의 멀티채널 알고리즘을 제안한다. 제안하는 채널 할당 알고리즘은 분

산적이고, 동적으로 동작하기 때문에 클러스터 간의 간섭을 회피하고 처리율 및 에너지 효율의 향상 측면에서 좋은 성능을 보인다.

본 논문은 다음과 같이 구성된다. 2장에서는 멀티채널 알고리즘에 대한 관련 연구를 보이고, 3장에서는 기존 클러스터링 방식의 문제점을 해결하기 위해 클러스터 단위로 서로 다른 채널을 분산적이고, 동적으로 할당하는 알고리즘을 제안한다. 4장에서는 자기조직화에 대해 설명하고, 5장에서는 제안하는 알고리즘의 실험 결과를 보인다. 마지막으로 6장에서 결론을 맺는다.

II. 관련연구

멀티채널 알고리즘은 다음과 같이 분류할 수 있다. 첫째로 송수신기의 수에 따라 구분한다. 하나의 송수신기를 사용한다면 하드웨어의 비용은 저렴하지만 복잡한 알고리즘이 요구된다. 보통의 센서 노드들은 송신과 수신을 동시에 할 수 없는 하나의 송수신기가 장착되어 효율적으로 데이터 전송을 하기 위한 알고리즘이 요구된다. 반대로 여러 개의 송수신기를 사용하면 알고리즘은 단순해지지만 하드웨어 비용의 증가와 상당한 에너지 소모가 발생한다. 둘째로 채널을 할당하는 방법에 따라 구분한다. 미리 정의된 규칙에 의해 채널을 할당하는 방법^{8,9)}과 기지국에서 직접 채널을 할당하는 방법¹⁰⁾이 있다. 셋째로 채널이 할당되는 노드의 수에 따라 구분한다. 예를 들어, 채널을 각 노드에 개별적으로 할당하는 방법과 여러 개의 노드들을 그룹으로 묶어 할당하는 방법^{11,12)}이다.

최근 무선 센서 네트워크에서 활용되는 멀티채널 알고리즘들이 연구되어 왔다. MMSN⁸⁾은 센서 네트워크에서 하나의 송수신기로 멀티채널 MAC을 적용한 첫 번째 알고리즘이다. 또한 RTS/CTS를 사용하는 것 대신에 미리 정의된 경량의 채널 할당 방법을 제안한다. 그러나 이 알고리즘은 채널 할당이 고정되어 있고, 채널의 활용도가 제한되는 단점이 있다.

MC-LMAC⁹⁾은 스케줄 기반의 멀티채널 알고리즘이다. 이것은 시간 슬롯을 할당하기 위해 ‘occupied slots vector’라고 하는 비트 벡터를 이용한다. 즉, 노드의 연결에 따라 비트의 길이를 설정하고 시간 슬롯을 사용하면 1을, 그렇지 않으면 0을 설정한다. 시간 슬롯을 설정하고자 하는 노드는 주변 노드의 비트 벡터를 수집하여 OR 연산을 하면 사용하지 않는 시간 슬롯을 설정할 수 있다. 만약 임의의 시간 슬롯에서 충돌이 감지되었을 때는 그것을 기억하고 있다가

해당 시간 슬롯을 사용하는 노드에게 보고하여 노드가 충돌을 확인하도록 한다. 그 후, 노드는 해당 시간 슬롯을 해제하고 다른 시간 슬롯을 선택하는 알고리즘을 수행한다. 그러나 비트 벡터의 사용은 데이터 패킷의 전송 오류 등으로 임의의 비트가 반전될 수 있다. 따라서 충돌이 발생하지 않는 시간 슬롯임에도 불구하고 시간 슬롯 선택 알고리즘을 다시 수행해야 하는 문제가 있다.

HyMAC^[10]은 센서 노드로부터 데이터를 수집하여 기지국으로 전송하는 hybrid TDMA/FDMA MAC 프로토콜이다. 이 알고리즘은 TDMA를 기반으로 Scheduled Slot과 Contention Slot으로 구성된 다수의 프레임이 반복되어 동작한다. 노드들은 기지국이 정해진 시간 슬롯과 채널을 이용하여 Scheduled Slot에서 데이터 전송을 하지만, 스케줄 되지 않은 노드들은 자신의 시간 슬롯과 채널을 결정하기 위해 Contention Slot에서 무작위로 기지국으로 HELLO 메시지를 전송하고, 기지국은 SCHEDULE 메시지를 전송하여 모든 노드들이 할당된 시간 슬롯과 채널에서 데이터를 전송한다. 그러나 이 알고리즘은 기지국에서 시간 슬롯과 채널을 할당하기 때문에 오버헤드가 발생하고, HELLO 메시지를 수신하지 못하면 해당 노드는 자신의 시간 슬롯과 채널을 할당 받을 수 없다. 또한 기지국을 기준으로 하는 트리 구조이기 때문에 상위 레벨로 올라갈수록 트래픽이 증가하여 고정된 시간 슬롯에서 모든 데이터 전송이 어려운 단점이 있다.

TMCP^[11]는 서브 트리 단위로 서로 다른 채널을 설정하여 노드들 사이에 채널을 바꾸는 횟수를 줄인다. 이로써 서브 트리 간에 간섭을 제거하고, 병렬적으로 데이터 전송하기 때문에 네트워크의 처리율을 높이고 패킷 손실을 줄일 수 있다. 그러나 TMCP는 임의의 노드가 다른 채널을 사용하는 인접 노드에게 데이터를 전달할 수 없는 문제가 발생한다. 또한 같은 채널을 사용하는 서브 트리에서는 노드들 간의 통신을 위해 서로 경쟁하는 문제가 발생할 수 있다. 마지막으로 채널이 할당된 서브 트리에서 중간 노드가 장애가 생길 경우 기지국까지 전달하는 경로가 단절될 위험이 있다.

본 논문에서는 클러스터 기반의 멀티채널 알고리즘과 분산적인 채널 할당 알고리즘을 제안한다. 이 알고리즘은 클러스터 단위로 채널을 할당하기 때문에 같은 클러스터 내의 노드들의 데이터를 클러스터 헤드가 통합하고, 통합된 데이터는 클러스터 헤드 간의 통신으로 기지국까지 전달한다. 이와 같은 방법으로 위에서 언급된 TMCP의 단점을 해결하였다. 그리고 제

안하는 채널 할당 알고리즘은 임의의 대표 노드를 선정하여 동적인 방식으로 채널을 할당하고, 분산적인 방법으로 최소의 채널을 할당합니다. 이로써 패킷 손실을 최소화하고 병렬적인 데이터 전송으로 처리율 및 에너지 효율의 향상 효과를 얻을 수 있다.

III. 클러스터 기반의 멀티채널 알고리즘

3.1 멀티채널 클러스터링 알고리즘

현재까지 무선 센서 네트워크에서 클러스터링 방법(LEACH^[13] 등)은 싱글 채널에서 클러스터 헤드와 클러스터 멤버를 선출하고, 클러스터 헤드를 중심으로 데이터를 수집 및 통합하는 방법으로 연구되어 왔다. 그러나 이런 방법은 인접한 클러스터 간의 간섭으로 인해 전송 지연뿐만 아니라 비효율적인 에너지 소모가 발생한다. 따라서 이 문제를 해결하기 위해 본 논문에서는 각 클러스터 단위로 서로 다른 채널을 할당하는 방식을 제안한다.

그림 1은 제안하는 알고리즘의 동작 모습을 표현한 그림이다. 이 알고리즘에서는 Setup Phase와 Steady State Phase가 반복적으로 동작한다. Setup Phase는 클러스터 구성, 채널 할당, 할당된 채널로 이동하는 3 단계로 구성된다. 첫째로, 클러스터 구성 단계는 에너지 효율을 고려한 기존의 클러스터 헤드 선출 방법을 택하여 클러스터를 구성한다. 본 논문에서는 클러스터 구성 방법을 고려하지 않는다. 둘째로, 채널 할당 단계는 인접한 클러스터 간 겹치지 않는 채널을 할당하기 위한 알고리즘이 동작하는 단계로써 분산적이고 동적으로 채널을 할당할 수 있는 알고리즘은 3.2절에서 설명한다. 마지막 단계인 할당된 채널로 이동하는

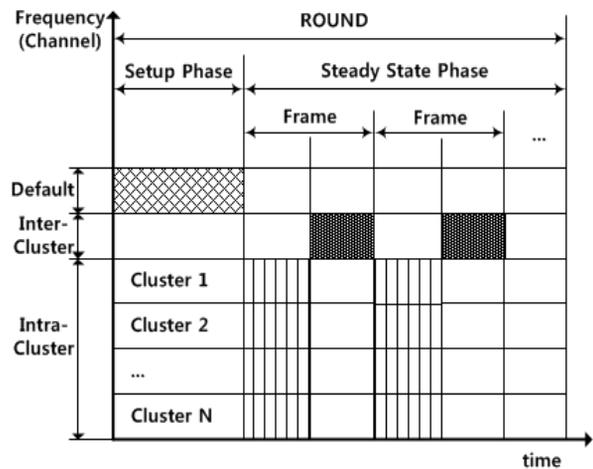


그림 1. 클러스터 기반의 멀티채널 알고리즘
Fig. 1. Cluster-based multi-channel algorithm

단계는 클러스터 헤드가 할당 받은 채널을 클러스터 멤버들이 그 채널을 인지하고 해당 채널로 이동하는 단계이다. Steady State Phase는 Intra-Communication과 Inter-Communication으로 구성된 프레임이 반복된다. Intra-Communication에서는 TDMA 방식으로 클러스터에 할당된 채널을 이용하여 클러스터 멤버들이 병렬적인 데이터 전송을 한다. 이로써 인접한 클러스터 간의 간섭을 방지할 수 있기 때문에 처리율과 에너지 효율을 향상시킬 수 있다. Inter-Communication에서는 클러스터 헤드가 수집한 데이터를 멀티 홉으로 기지국으로 전송한다.

다음으로 세로축은 네트워크에서 사용되는 채널로써 각각 Default Channel, Inter-Cluster Channel, 그리고 Intra-Cluster Channel로 나뉜다. Default Channel은 센서 노드가 초기 상태일 때 할당되는 채널로, 이 채널을 통해 스스로 네트워크를 구성한다. Inter-Cluster Channel은 클러스터 헤드가 멤버들로부터 수집한 데이터를 멀티 홉으로 기지국으로 전송하기 위한 채널이다. 이때 클러스터 멤버들은 슬립 모드 상태로 에너지를 비축한다. 마지막으로 Intra-Cluster Channel은 클러스터 내부에서 클러스터 헤드와 클러스터 멤버가 서로 데이터 통신을 하는 채널로써 인접한 클러스터 간에 서로 다른 주파수 채널을 할당하여 인접한 클러스터와 간섭이 발생하지 않도록 한다.

3.2 채널 할당 알고리즘

각 클러스터에 서로 다른 채널을 할당하는 알고리즘은 센서 네트워크에서 클러스터 구성이 끝난 후에 동작한다. 클러스터 단위로 서로 다른 채널을 할당하는 문제는 그래프 색칠 문제^[14]로 모델링할 수 있다. 그래프 색칠 문제는 임의의 그래프에 색을 칠할 때 인접한 정점들이 서로 다른 색을 갖도록 하는 것으로 채널 할당 알고리즘에 적용하면 정점은 클러스터 헤드, 간선은 클러스터 헤드 간 연결된 링크로 표현할 수 있다. 즉, 오직 클러스터 헤드들이 서로 메시지를 통해 겹치지 않는 채널을 할당하고, 클러스터 멤버들은 슬립 모드로 전환하여 에너지 소모를 줄인다.

채널 할당 알고리즘을 위해 모든 클러스터 헤드는 사용 가능한 모든 채널을 자신의 리스트에 저장하고 한 개 이상의 Seed Node를 무작위로 선출한다. 채널 할당은 Seed Node로부터 시작해서 주변의 클러스터 헤드들로 전파되어 분산적이고 동적으로 채널을 설정할 수 있다. 그리고 멀리 떨어져 있는 클러스터는 같은 채널을 재사용할 수 있기 때문에 적은 수의 채널로 서로 겹치지 않는 독립적인 채널을 설정할 수 있다.

그림 2는 채널 할당 알고리즘의 슈도코드이다. Seed Node로 선택된 노드는 채널 리스트에서 첫 번째 채널을 선택하고 이 정보를 이웃에게 전송한다. 그러면 이웃 노드들은 같은 채널이 할당되는 것을 방지하기 위해 자신의 채널 리스트에서 해당 채널을 지운다. 만약 채널을 설정하는 과정에서 이웃 노드가 자신의 채널과 같은 채널을 사용할 경우 노드마다 설정된 고유 ID를 비교하여 ID가 작은 노드에게 우선권을 주어 채널 충돌을 해결한다. 이 과정은 모든 클러스터 헤드가 채널을 설정할 때까지 반복된다.

그림 3은 임의의 토폴로지를 고려하여 채널이 할당되는 과정을 나타낸다. 먼저, 하나의 Cycle은 각 클러스터 헤드가 자신의 채널을 할당하고, 할당된 채널을 이웃들에게 알려 리스트를 갱신하는 것으로 정의한다. Cycle 1은 Seed Node로 선정된 노드 5와 노드 8이 채널 1을 설정하고 이웃들에게 전송한다. 따라서 Seed Node와 이웃한 노드는 채널 1을 리스트에서 삭제한다

```

IF(Seed Node)
    Assign Channel 1
    Broadcast the information to the neighbors
    Break
ENDIF

WHILE(Setup Phase)
    IF(receiving channel assignment packet)
        IF(received Channel = my Channel)
            IF(my ID < sender ID)
                Assign the channel to my Channel
                Broadcast the information to the neighbors
                Continue
            ELSE
                Delete an element in the channel list
                Assign another channel k
                Broadcast the information to the neighbors
                Continue
            ENDF
        ENDF
    Delete an element in the channel list
    Assign another channel k
    Broadcast the information to the neighbors
    ENDF
ENDWHILE
    
```

그림 2. 분산적 채널 할당 알고리즘을 위한 슈도코드
Fig. 2. Pseudo-code for distributed channel allocation algorithm

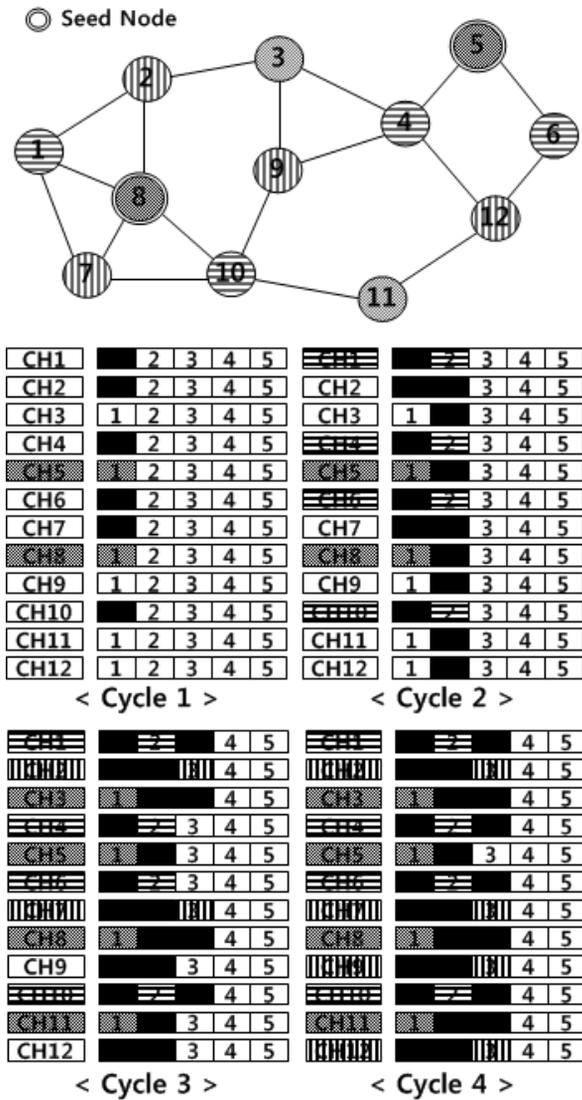


그림 3. 특정 토폴로지에 대한 채널 할당 과정
Fig. 3. Channel allocation process for a specific topology

다. Cycle 2에서는 Seed Node와 1홉 거리에 있는 노드들이 채널 2를 설정하고 이웃에게 전송한다. 이때, 노드 1과 이웃인 노드 7에서 채널 충돌이 발생하지만, ID가 작은 노드에 우선권을 부여하였으므로 노드 7은 채널 2를 설정하지 못하고 리스트에서 채널 2를 삭제한다. Cycle 3과 Cycle 4에서는 위와 같은 과정을 반복한다. 여기에서 주의 깊게 살펴볼 것은 Cycle 3의 노드 9와 노드 11은 자신의 리스트에서 채널 1을 사용할 수 있다는 정보를 갖고 있기 때문에 채널 1을 재사용한다는 점이다. 결과적으로 해당 토폴로지에서는 총 4단계의 Cycle과 4개의 서로 다른 주파수 채널을 사용한다. 이와 같은 과정으로 설정된 채널은 인접한 클러스터에게 간섭을 미치지 않기 때문에 에너지 낭비를 막고 처리율을 향상시킬 수 있다.

제안하는 알고리즘은 각 클러스터에 채널을 할당하기 위해서 클러스터 헤드를 중심으로 동작하고, 공간의 재사용으로 클러스터 멤버들 간에 채널의 간섭이 드물게 발생할 수 있다. 이를 해결하기 위해 각 클러스터에 이미 설정되어 있는 채널을 바꾸는 것은 비효율적인 일이다. 따라서 Setup Phase의 마지막 단계인 클러스터의 모든 노드들이 해당 채널로 이동할 때, 클러스터 멤버들 간에 간섭이 발생하는지 간단하게 체크할 수 있는 방법이 있다. 예를 들어 그림 4와 같이 클러스터 1과 클러스터 2의 멤버들이 간섭이 발생한다고 가정해보자. 클러스터 헤드 1(CH1)이 설정한 채널 및 자신의 멤버가 사용할 시간 슬롯에 대한 정보를 자신의 클러스터 멤버에게 전송한 후 클러스터 멤버로부터 ACK 메시지를 수신 받는다. 이 ACK 메시지를 통해 주변에 있던 클러스터 멤버 2(CM2)가 간섭 발생 여부를 판단할 수 있다. 그리고 자신의 클러스터 헤드에게 간섭이 발생한다는 메시지를 전송하고 자신의 시간 슬롯을 재할당 받아 간섭을 피할 수 있다.

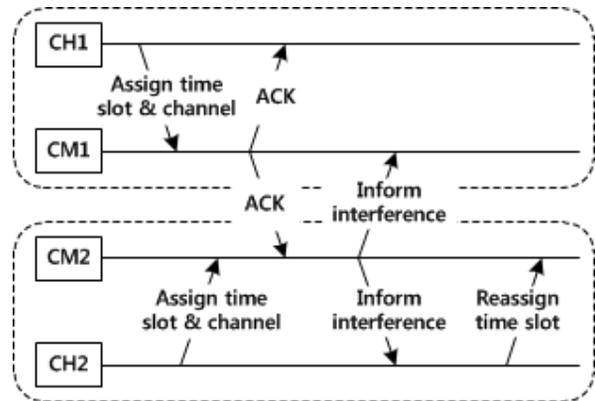


그림 4. 다른 클러스터에 속한 클러스터 멤버들 간의 간섭 해결 방법
Fig. 4. Solution for interference between cluster members in different clusters

3.3 자기조직화(Self-Organization)

3.3.1 센서 노드의 장애

무선 센서 네트워크에서 센서 노드는 배터리 소모나 주변 환경의 요인 등으로 기능의 장애가 발생할 수 있다. 이러한 기능 장애가 네트워크에 미치는 영향을 고려해야한다. 제안 알고리즘에서는 센서 노드들이 클러스터 헤드와 클러스터 멤버로서 역할을 수행한다. 먼저, 클러스터 멤버의 장애는 알고리즘의 구조적인 문제가 발생하지 않아 무시하면 된다. 하지만, 클러스터 헤드의 장애는 자신의 클러스터 멤버들의 데이터를 수집할 수 없고, 다른 클러스터의 데이터가 기지국

까지 전송되는 경로를 단절시킬 수 있다. 예를 들어, 그림 5에서 클러스터 2의 클러스터 헤드(CH2)가 장애가 발생한다면, 인접한 클러스터와 다른 채널을 사용하는 클러스터 2의 멤버(CM2)들은 해당 채널에 고립되어 데이터를 전송할 수 없고, 클러스터 헤드 1(CH1)에서 클러스터 헤드 3(CH3)으로 가는 경로도 끊어진다.

따라서 일정 시간 동안 자신의 클러스터 헤드와 통신을 하지 못하게 되면 클러스터 멤버들은 클러스터 헤드가 장애가 발생한 것으로 인지하고, 고립된 센서 노드들이 클러스터 헤드 선출 과정을 거쳐 다음 Setup Phase가 돌아올 때까지 새로운 임시 클러스터 헤드를 선정한다. 이렇게 선정된 임시 클러스터 헤드는 기존의 클러스터 헤드와 같은 역할을 수행해 고립되었던 클러스터 멤버들을 다시 통신에 참여할 수 있게 하고, 다른 클러스터 헤드와 멀티 홉으로 연결되어 기지국까지의 데이터 전송 경로를 복구한다.

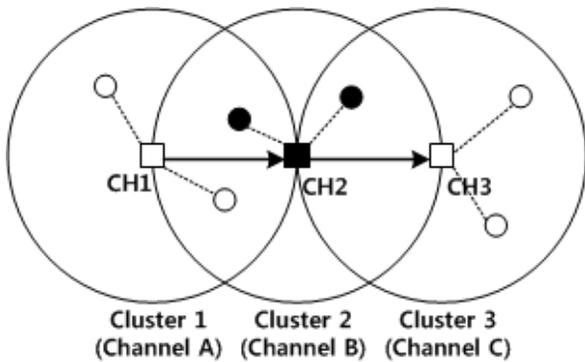


그림 5. 센서 노드의 장애
Fig. 5. Failure of the sensor node

3.3.2 신규 센서 노드의 삽입

센서 네트워크가 동작하고 있는 상태에서 새로운 노드들이 그 네트워크에 삽입이 될 수 있다. 삽입되는 노드의 수는 한 개의 노드가 삽입될 수 있고, 다수의 노드가 삽입될 수 있다. 네트워크가 Setup Phase 상태 일 때는 네트워크의 모든 노드들이 Default Channel에서 동작하기 때문에 신규 노드는 다른 노드들과 함께 클러스터를 구성하면 된다. 그러나 Steady State Phase 상태일 때는 삽입된 노드들은 Default Channel을 사용하고 주변 노드들은 다른 채널을 사용하기 때문에 서로 통신을 할 수 없다. 하지만 제안하는 알고리즘에서는 클러스터 헤드가 수집한 데이터를 기지국으로 전송하기 위해 Inter-Cluster Channel을 빈번히 사용하므로, 새로 삽입된 노드는 Default Channel과 Inter-Cluster Channel에서 HELLO 메시지나 주변 데

이터의 엿듣기를 통해 다른 노드의 존재 여부를 판단할 수 있다.

먼저 한 개의 센서 노드가 삽입된 경우, Default Channel에서 엿듣기를 성공했다면, 현재 네트워크는 Setup Phase를 수행하고 있다는 의미이므로 다른 노드들과 함께 클러스터를 구성하면 되고, Inter-Cluster Channel에서 엿듣기를 성공했다면, 동작하고 있는 근처의 클러스터 헤드들을 발견했다는 의미이다. 이때에는 엿듣기한 데이터 패킷에 대해 수신 강도, 사용하고 있는 Intra-Cluster Channel, 그리고 해당 클러스터 헤드에 속한 클러스터의 멤버 수 등의 정보를 검출하여 최적의 클러스터 헤드를 선택하고, 수신했던 데이터 패킷에서 검출한 정보를 이용하여 해당 클러스터 시간 슬롯의 마지막 슬롯으로 신규 노드를 할당한다. 만약 주변의 모든 클러스터가 각각 최대의 클러스터 멤버를 갖고 있다면 신규 노드는 어느 클러스터에도 편입되지 못할 수 있다. 따라서 일정한 시간이 지난 후 어느 곳에도 편입되지 못하면 클러스터 멤버를 갖지 않는 독립적인 클러스터 헤드가 되고, 다음 Setup Phase를 통해 새롭게 클러스터를 구성한다.

다음으로 다수의 센서 노드가 추가된 경우를 고려해보자. Default Channel에서 엿듣기를 성공했다면 앞의 경우와 같이 다른 노드들과 함께 클러스터를 구성하면 되고, 엿듣기를 성공하지 못했다면 네트워크는 Steady State Phase로 간주할 수 있다. 따라서 Default Channel에서는 새로 추가된 노드들만 존재하게 되고, 각자 전송한 HELLO 메시지로 주변의 신규 노드들을 발견하여 새로운 클러스터를 구성한다. 그 후 클러스터 헤드는 Inter-Cluster Channel로 이동하여 주변 클러스터의 채널 정보를 수집하고 자신의 채널을 설정한다. 이렇게 설정된 클러스터는 다음 Setup Phase까지 유지한다.

IV. 성능평가

본 장에서는 제안한 알고리즘의 성능 평가를 위해 표 1과 같은 환경을 구성한 NS-2^[15]를 이용해 분석하였다. 모든 노드는 같은 양의 에너지를 가지며 동기화가 되어있다. Setup Phase에서 노드들은 기존의 클러스터 헤드 선정 방법을 이용하여 스스로 클러스터가 구성되고, 제안한 채널 할당 알고리즘을 이용하여 각 클러스터에 채널을 할당하였다. 이때 노드들이 사용할 수 있는 채널의 수는 10개로 제한하였다. Steady State Phase에서는 클러스터 멤버가 Intra-Cluster Channel로 설정된 각각의 독립된 채널에서 100byte의 크기의

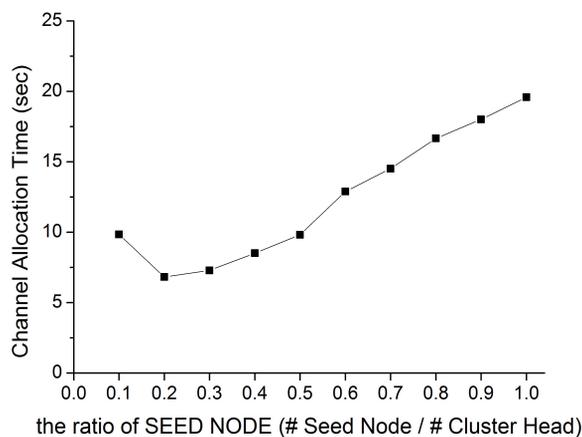
표 1. 실험 환경

Table 1. Simulation parameters

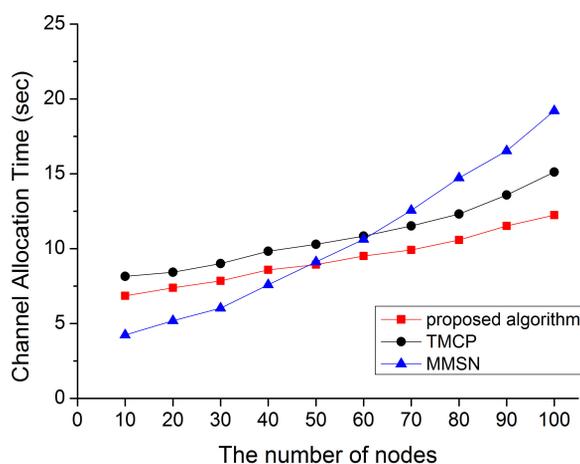
Parameter	Value
Network size	1000×1000
Number of nodes	10~100
Data size	100 bytes
Radio range	50 m
Idle power	35 mW
Received power	395 mW
Transmit power	660 mW
Number of available channels	10
Maximum number of members in each CH	10

데이터를 클러스터 헤드로 전송하고, 클러스터 헤드는 이를 수집 및 통합하여 Inter-Cluster Channel에서 기지국으로 전송하기를 반복한다. 그리고 성능 비교를 위하여 기존의 알고리즘과 채널 할당 시간, PDR, 처리율, 지연 시간, 평균 에너지 소모 측면에서 분석하였다.

제안하는 채널 할당 알고리즘에서는 몇몇의 클러스터 헤드가 Seed Node로 설정되어야 한다. 이 Seed Node로부터 분산적이고 동적으로 채널이 모든 노드에 할당된다. 이때, 네트워크에서 얼마나 많은 Seed Node가 선정되어야 하는지 결정하는 것은 매우 중요하다. 그림 6(a)는 Seed Node의 비율에 따라 모든 노드가 채널이 할당될 때까지 걸린 시간을 나타낸다. 대략 20%의 비율로 Seed Node가 선정되었을 때 최적의 채널 할당 시간이 된다는 것을 확인할 수 있다. 반면에 Seed Node가 20%보다 적을 경우에는 Seed Node가 충분하지 않아 모든 클러스터에 채널을 할당하는 시간이 많이 걸리고, Seed Node가 20%보다 많을 경우에는 Seed Node로부터 전송된 메시지들의 충돌로 재전송이 많이 발생하여 시간이 많이 걸린다. 그림 6(b)는 노드 수의 따른 채널 할당 시간을 기존의 알고리즘과 비교하였다. 먼저 MMSN은 노드 수에 따라 채널 할당 시간이 선형적으로 증가하는데, 이것은 상당한 컨트롤 오버헤드가 발생한다는 것을 나타낸다. 반면에 제안하는 알고리즘과 TMCP의 채널 할당 시간은 상대적으로 완만한 증가를 보인다. 이것은 네트워크를 구성하는 노드의 수가 많을 때 더 효율적이라는 의미이다. 특히, 제안하는 알고리즘은 TMCP보다 빠르게 채널을 할당할 수 있는데, 이것은 채널을 할당하기 위해 전송되는 컨트롤 패킷의 오버헤드를 줄인 결과이다.



(a)



(b)

그림 6. 채널 할당 시간 (a) Seed node의 비율 (b) 다른 알고리즘과 비교

Fig. 6. Channel allocation time (a) the ratio of seed nodes, (b) the comparison of the algorithms

그림 7에서는 초당 30 패킷을 보내는 소스의 수를 변화시키며 측정된 PDR을 나타내고, 80% 이상이 유효한 것으로 판단한다. 싱글채널 클러스터링 알고리즘의 PDR은 소스의 수가 증가할수록 급격하게 감소한다. MMSN은 소스의 수가 4개를 넘어가면서 80%이하의 낮은 PDR을 갖는다. TMCP는 소스의 수가 6개일 때까지 조건을 만족하였고, 제안하는 알고리즘은 소스의 수가 8개일 때까지 높은 PDR을 유지한다. 이것은 그룹으로 채널을 할당하는 것이 패킷의 충돌 문제를 완화시키고, 이웃 노드들 간의 간섭을 줄이는 효과를 얻을 수 있다.

그림 8은 처리율을 나타낸다. 처리율은 CBR (Constant Bit Rate) 스트림을 주기적으로 생성하고, 스트림의 수를 변화시키며 기지국에서 수신한 데이터를 분석하여 측정하였다. 알고리즘들은 CBR 스트림

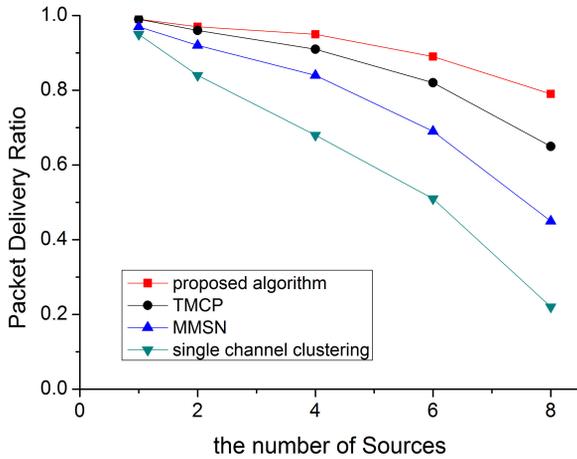


그림 7. 패킷 전송률
Fig. 7. Packet delivery ratio

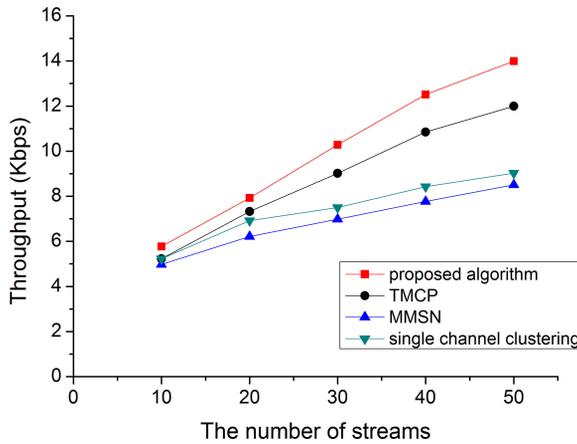


그림 8. CBR 스트림의 변화에 따른 처리율
Fig. 8. Throughput with changes in the CBR streams

이 증가하는 경우에 처리율이 큰 차이를 보인다. MMSN은 패킷 손실과 노드들 사이의 채널을 조절하는 시간 때문에 낮은 처리율을 갖는다. 싱글 채널 클러스터링 알고리즘은 CBR 스트림이 증가하면 인접한 클러스터 사이의 간섭 때문에 패킷 손실이 증가하기 쉬워 MMSN과 비슷한 성능을 보인다. TMCP는 앞의 두 알고리즘보다 좋은 성능을 보이지만, 서브 트리의 중간 노드가 오작동 등으로 성능 감소가 발생할 수 있다. 제안하는 알고리즘은 클러스터 멤버들의 데이터를 서로 다른 채널에서 통합해서 패킷 손실을 최소화 하고, 각 클러스터에서 TDMA로 병렬적으로 통신하고, 서로 다른 채널을 사용하여 간섭을 줄임으로써 성능을 높인다.

그림 9는 중단 지연을 보여준다. 이것은 모든 알고리즘이 비슷한 결과를 보인다. 하지만, TMCP는 각각의 서브 트리가 소스에서 목적지까지 미리 독립적인 채널을 설정하여 경로가 설정되어 있기 때문에 빠른

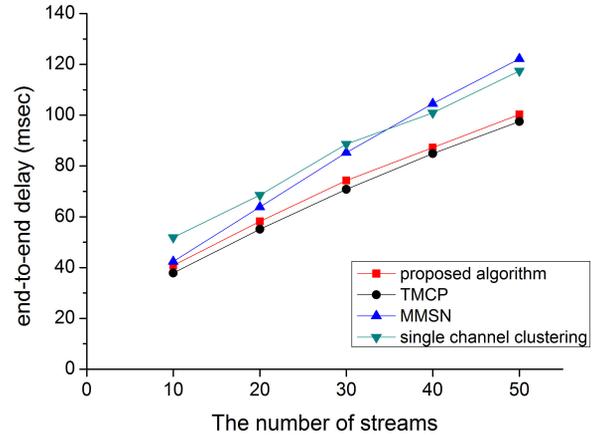


그림 9. CBR 스트림의 변화에 따른 지연시간
Fig. 9. End-to-end delay with changes in the CBR streams

데이터 전송을 할 수 있어 다른 알고리즘들에 비해 미세하게 좋은 결과를 보였다. 제안하는 알고리즘은 클러스터 멤버들이 미리 정의된 TDMA 스케줄에 따라 통신하기 때문에 TMCP 보다 중단 지연이 높게 나타난다. MMSN은 다른 노드들과 통신하기 위해 빈번하게 채널을 변경하기 때문에 중단 지연이 증가하기 쉽다. 그리고 싱글 채널 클러스터링 알고리즘은 인접한 클러스터와의 간섭과 충돌로 인해 재전송이 발생하고, 데이터 통합으로 인해 중단 지연이 증가한다.

그림 10은 에너지 효율을 나타낸다. MMSN은 다른 노드들과 통신을 하기위해 채널을 빈번히 바꾸기 때문에 상대적으로 많은 에너지를 소모한다. 싱글 채널 클러스터링 알고리즘은 MMSN과 비교하여 대체로 좋은 에너지 효율을 보인다. 그러나 CBR 스트림이 증가하면 인접한 클러스터 간에 간섭과 충돌로 에너지 소모가 증가한다. 제안하는 알고리즘과 TMCP는 그룹으로 채널을 할당함으로써 에너지 효율을 향상시

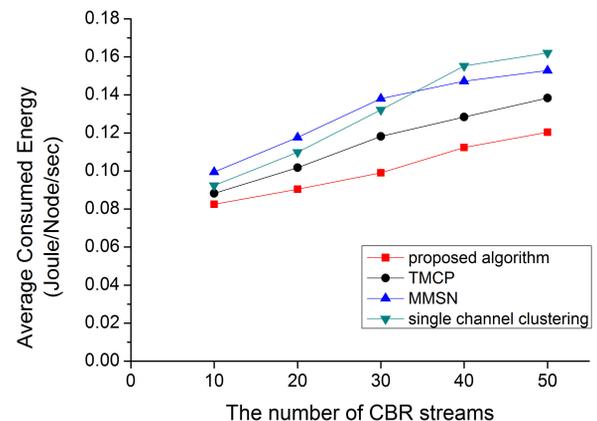


그림 10. CBR 스트림의 변화에 따른 평균 에너지 소비량
Fig. 10. Average consumed energy with changes in the CBR streams

킨다. 더욱이, 제안하는 알고리즘은 인접한 클러스터들이 서로 다른 채널을 사용하여 병렬적으로 데이터를 전송하여 간섭을 회피하고, 클러스터 헤드가 데이터 통합하여 기지국으로 전송하기 때문에 에너지 효율을 높일 수 있다. 하지만, TMCP는 데이터 통합 없이 모든 노드가 데이터 전송하기 때문에 제안 알고리즘 보다 에너지 소모가 크다.

V. 결 론

선박 네트워크는 융합된 유무선 네트워크를 통해 장치들을 하나의 네트워크로 연결하여 각 종 정보를 실시간으로 확인하고 선박의 안전한 운행 및 유지비용의 효율성을 높인다. 특히, 선박 네트워크에서 소형의 센서 노드를 활용한 무선 센서 네트워크의 연구는 많은 장점이 있다. 본 논문에서는 선박 네트워크 환경에서 에너지 효율을 고려한 무선 센서 네트워크에 관한 연구를 진행하였다. 제안하는 알고리즘은 분산적이고 동적인 채널 할당 알고리즘을 적용해 클러스터 단위로 서로 다른 채널을 할당하여 성능을 향상시켰다. 클러스터 내부에서는 클러스터 멤버들이 TDMA 스케줄에 따라 동작하므로 불필요한 시간에는 슬립을 통해 에너지를 절약할 수 있고, 인접한 클러스터와의 간섭도 최소화할 수 있다. 따라서 기존의 알고리즘에 비해 처리율 및 전송 지연, 에너지 효율 측면에서 향상된 성능을 보였다. 그리고 센서 노드가 장애가 발생하거나 새로 추가되는 경우에도 내고장성과 확장성을 보장한다. 그러나 제안하는 알고리즘은 정의된 채널 및 시간 슬롯에 따라 통신이 이루어지기 때문에 기지국으로 빠르게 전송해야 할 데이터를 처리할 수 없다. 따라서 QoS 보장을 위한 연구가 계속 진행되어야 할 것이다.

References

[1] K. I. Lee, J. H. Park, W. S. Choi, and K. D. Moon, "International standardization trend in SAN," *TTA J.*, no. 126, pp. 45-51, Nov. 2009.

[2] S. H. Lee, J. H. Kim, K. D. Moon, K. I. Lee, and J. H. Park, "Performance analysis on integrated ship area network," *J. KICS*, vol. 38C, no. 03, Mar. 2013.

[3] M. S. Choi, S. J. Pyo, J. S. Lee, S. H. Yoon, and S. R. Lee, "Energy efficiency routing algorithm for vessel ubiquitous sensor network

environments," *J. KICS*, vol. 36, no. 5, Nov. 2012.

[4] Y. S. Moon, Y. C. Bae, J. K. Park, and S. H. Roh, "The implementation of wire and wireless integration module of zigbee and oprical communication for ship area network (SAN)," *J. KICS*, vol. 5, no. 5, Oct. 2010.

[5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, 393-422, Mar. 2002.

[6] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292-2330, Aug. 2008.

[7] X. Liu, "A survey on clustering routing protocols in wireless sensor networks," *Sensors*, vol. 12, no. 8, Aug. 2012.

[8] G. Zhou, C. Huang, T. Yan, J. A. Stankovic, and T. F. Abdelzaher, "MMSN: multi-frequency media access control for wireless sensor networks," *IEEE INFORCOM*, Apr. 2006.

[9] O. D. Incel, L. van Hoesel, P. Jansen, and P. Havinga, "MC-LMAC: A multi-channel MAC protocol for wireless sensor networks," *AdHoc Networks*, vol. 9, no. 1, Jan. 2011.

[10] M. Salajegheh, H. Soroush, and A. Kalis, "HYMAC: Hybrid TDMA/FDMA medium access control protocol for wireless sensor networks," *18th Annu. IEEE Int. Symp. Pers., Indoor and Mob. Radio Commun.(PIMRC'07)*, pp. 1-5, Athens, Sept. 2007.

[11] Y. Wu, J. A. Stankovic, T. He, and S. Lin, "Realistic and efficient multi-channel communications in wireless sensor networks," *IEEE INFOCOM*, Apr. 2008.

[12] J. I. Kong, K. Hur, and D. S. Eom, "A cluster-based multi-channel MAC protocol for wireless sensor networks," in *Proc. Korea Multimedia Soc.*, vol. 12, no. 2, Nov. 2009.

[13] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wirel. Commun.*, vol. 1, no. 4, Oct. 2002

- [14] en.wikipedia.org/wiki/Graph_coloring
- [15] The Network Simulator NS-2, http://nsnam.isi.edu/nsnam/index.php/Main_Page

공 준 익 (Joon-Ik Kong)



2008년 2월 : 고려대학교 전산학과 및 전자공학과 졸업
2010년 2월 : 고려대학교 전자공학과 공학석사
2015년 2월 : 고려대학교 전자공학과 공학박사
2015년 4월~현재 : 목포대학교

정보산업연구소 박사후 연구원

<관심분야> 디지털통신시스템, MAC, Routing, IoT, 임베디드시스템

이 성 로 (Seong Ro Lee)



1987년 2월 : 고려대학교 전자공학과 공학사
1990년 2월 : 한국과학기술원 전기 및 전자공학과 공학석사
1996년 8월 : 한국과학기술원 전기 및 전자공학과 공학박사
1997년 9월~현재 : 목포대학교

공과대학 정보전자공학과 교수

<관심분야> 디지털통신시스템, 이동 및 위성통신시스템, USN/텔레미틱스응용분야, 임베디드시스템