

스마트 빌딩에서 사물인터넷 기반 탄력적 소방 대피 안내 시스템

심승민*, 양태훈*, 서동영*, 김상하°

Flexible Fire Safety Navigation System Based on Internet of Things in Smart Buildings

Seungmin Sim*, Taehun Yang*, Dongyeong Seo*, Sang-Ha Kim°

요 약

실내 화재는 재산과 인명에 큰 손실을 야기하며, 특히 다양한 목적을 위해 점점 더 복잡해지는 오늘날의 건물에서는 그 문제가 더욱 심화된다. 따라서 모든 건물에는 인명과 재산을 보호하기 위한 소방 방재 시스템이 구축된다. 이러한 시스템은 화재 센서로부터 감지된 정보를 중앙 서버를 통해 관리하여, 화재의 확산을 저지하거나 사람들의 대피를 유도하기 위한 정보를 제공하는 데 사용된다. 하지만, 기존 시스템들은 중앙 서버에 의존하기 때문에 시시각각 변화하는 상황에 탄력적으로 대응하기는 쉽지 않다. 따라서 이러한 문제를 해결하기 위해 우리는 스마트 빌딩에서 IoT 기반의 탄력적 화재 안전 내비게이션 시스템을 제안한다. 제안하는 시스템은 중앙 서버에 의존하는 기존 시스템들의 문제를 완화한다. 이를 위해서, 화재안전망은 센서들 간 정보교환을 통해 화재 상황을 감시한다. 그리고 네트워크 기반의 화재 정보를 통해 대피자들의 스마트폰을 통해 대피 경로를 직접 제공할 수 있다. 실험은 제안된 계획에 대한 개념 증명을 통해 수행되며, 실험 결과는 제안된 방식이 기존 시스템보다 우수함을 보인다.

Key Words : Internet of things, Smart building, Wireless sensor network, Mobile device, Fire evacuation system

ABSTRACT

Indoor fires cause a big loss of property and human life in environments which especially increasingly complex buildings for multiple purposes. Thus, fire prevention systems are built in most of the buildings to protect people and property. Such systems provide information to prevent the spread of fire and induce people to evacuate by managing the information detected from the fire sensor through a central server. However, due to the dependence of the existing systems on the central server, it is not trivial to flexibly respond to the changing situation every moment. Therefore, to solve this problem, we propose an IoT-based elastic fire safety navigation system in smart buildings. The proposed system mitigates a problem by the dependence on a central server of existing systems. For this, a fire safety network monitors fire situations via information exchange between sensors. Through network-based fire information, evacuation paths could be directly provided to the evacuees via their smartphones. The experiments are conducted via a proof-of-concept for the proposed scheme. The results are shown that the proposed scheme is superior to the existing system.

※ 본 연구는 충남대학교 학술연구비에 의해 지원되었습니다.

• First Author : Chungnam National University Department of Computer Engineering, sminsim98@gmail.com, 정희원

° Corresponding Author : Chungnam National University Department of Computer Engineering, shkim@cnu.ac.kr, 종신희원

* Chungnam National University Department of Computer Engineering, {thyang@cclab.cnu.ac.kr, dyseo.tm@gmail.com}, 학생회원
논문번호 : 202110-277-B-RN, Received October 4, 2021; Revised December 3, 2021; Accepted December 28, 2021

1. 서론

사물인터넷의 출현과 다양한 센서들 및 정보통신 기술의 발달은 건물을 스마트하게 만들고 있다. 이를 스마트 빌딩이라고 부른다. 오늘날 스마트 빌딩에는 풍향계, 풍속계, 공기질 센서, 화재 감지 센서, 온도 센서, 습도 센서 등 다양한 센서들이 설치 및 매립되어 지고, 이들로부터 얻은 정보를 기반으로 건물 내 공기 순환을 위한 공조 시스템, 화재 대피 시스템 등 다양한 시스템들이 운용되고 있다. 특히, 화재는 건물의 자산뿐만 아니라 인명피해로 이어질 수 있기 때문에, 화재 대피 시스템은 건물의 안전성을 위하여 매우 중요한 시스템이다^{1,2}.

그림 1에서처럼, 건물 내 화재가 발생하게 되면 화재 경보 시스템에 의해 경보음이 울리고, 사람들은 건물 밖으로 대피하게 된다. 이때, 사람들은 비상구를 찾아 긴급하게 이동하는 것이 일반적인 대피 형태이다. 그러나 이러한 대피 형태는 안전하지 않고 효과적이지 않다. 화재가 발생한 위치를 알 수 없기 때문에 화재가 발생한 방향으로 이동할 수도 있고, 구조가 복잡한 대형 건물의 경우 자신의 위치와 비상구의 위치를 알기 어렵다. 따라서, 오늘날 스마트 빌딩에 다양한 화재 대피 안내 시스템들이 제안되어왔다.

오늘날 화재 대피 안내 시스템들은 공통적으로 화재를 감지하는 센서와 이에 대한 정보를 수집하는 중앙 서버, 그리고 화재 정보를 바탕으로 사람들의 대피를 유도하기 위한 피난 유도등 또는 스마트폰과 같은 디바이스들로 구성된다. 이러한 구조에서, 화재 대피 안내 정보는 항상 중앙 서버를 통해 제공되며, 대피

유도 방식은 크게 두 가지로 구분된다. 하나는 사용자가 선택할 수 있는 모든 대피 경로 정보를 제공하여 대피를 유도하는 방식³⁻⁶이고, 다른 하나는 중앙 서버에서 결정된 대피 경로를 제공하여 대피를 유도하는 방식⁷⁻⁹이다. 그러나, 화재 대피 안내 서비스를 제공하기 위해서, 기존 화재 대피 시스템들은 중앙 서버의 화재 정보를 기반으로 한다. 즉, 중앙 서버 중심의 서비스가 제공되기 때문에, 중앙 서버와 통신이 두절 및 물리적 손상은 시스템 전체에 대한 장애를 발생시킨다. 이러한 중앙 서버의 의존성은 화재 대피 시스템의 강건성과 그로 인한 화재 안내 서비스에 대한 신뢰성을 더욱 저해시키는 요소로 작용할 수 있다. 시시각각 변화하는 화재 상황 속에서, 신뢰적인 화재 대피 안내 서비스를 제공하기 위해서는 다음과 같은 요구사항을 만족해야 한다. 1) 화재 대피자에게 자신의 위치와 화재의 위치를 제공해야 한다. 2) 위치 정보를 바탕으로, 화재로부터 가장 안전한 비상구로 도달할 수 있는 대피 경로를 제공해야 한다. 3) 위의 요구사항들은 외부와 통신이 단절된 상황에서도 만족해야 한다.

본 논문은 기존 시스템들이 갖는 문제점들을 개선하고 신뢰적 화재 대피 안내 서비스를 제공할 수 있는 새로운 화재 안전 내비게이션 시스템을 제안한다. 제안된 시스템은 화재 센서들로 구축된 화재 안전 네트워크, 건물 내 화재 정보를 수집하는 화재 안전 서버, 그리고 화재 정보를 서버로 전달하고 내비게이션 서비스를 제공하는 화재 안전 내비게이션 디바이스로 구성된다. 디바이스는 화재 안전 네트워크를 구성하는 센서 노드의 신호 정보를 통해서 건물 내 자신의 위치를 알 수 있고, 중앙 서버에 수집된 정보를 바탕으로 화재의 위치를 알 수 있다. 뿐만아니라, 두 정보를 활용하여 가장 안전한 대피 경로를 계산할 수 있는 가중치 기반의 알고리즘을 개발한다. 그리고 센서 간 멀티홉 통신으로 화재 안전 네트워크 전체에 전파된 화재 정보를 내비게이션 디바이스가 직접적으로 활용함으로써, 화재 안전 서버와 통신이 단절된 상황에서도 화재 안전 내비게이션 서비스를 제공할 수 있다.

본 논문의 구성은 다음과 같다. 2장은 화재 대피 안내 시스템과 관련된 기존 연구들에 대해 살펴보고, 시스템 구조에 대해 분석한다. 3장은 본 논문에서 제안하는 화재 안전 내비게이션 시스템을 설명한다. 4장은 제안된 시스템의 검증 및 평가를 수행한다. 5장은 본 논문의 결론 및 향후 연구로 마친다.

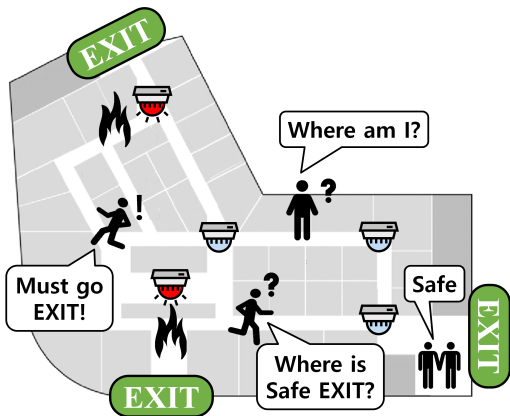


그림 1. 스마트 빌딩에서 화재
Fig. 1. Fire accident in smart building

II. 관련 연구

본 장은 화재 대피 안내 시스템과 관련된 기존 연구들에 대해 탐구한다. 기존 연구들은 시스템 구조에 따라 총 다섯 가지로 분류된다.

논문 [3]은 비콘 기반의 지능형 화재 대피 안내 시스템을 제안한다. 중앙 서버는 건물 내 배치된 화재 센서들의 정보를 수집하고, 이에 대한 정보를 사용자의 스마트폰에 전달한다. 그리고 나서 스마트폰은 건물 내 화재 정보와 주변 센서들의 비콘 신호 정보를 활용하여, 건물 내 화재의 위치와 사용자의 위치를 사용자에게 알려준다. 이로써, 사용자가 해당 정보를 바탕으로 스스로 판단하여 대피하도록 유도하는 방안이다. 그러나 이 방안은 화재가 발생한 구역으로 가까이 가지 않도록 화재 정보만 제공해줄 뿐 직접적으로 비상구에 도달할 수 있는 어떠한 방법도 제공하지 않는다. 따라서, 구조가 복잡한 건물 내부나 구조를 잘 알지 못하는 경우 효과적이지 못한 방안이다. 즉, 대피 과정에서, 갑작스럽게 나온 경우 신속한 대피를 방해할 수 있다.

논문 [4]는 BLE (Bluetooth Low Energy) 비콘을 활용한 실내 위치 기반의 화재 대피 안내 시스템을 제안한다. 중앙 서버는 센서들로부터 화재 정보와 인구 혼잡도를 수집하여, 이를 바탕으로 화재가 발생한 구역의 정보 및 이동할 수 있는 모든 대피 경로를 파악 후 사용자의 스마트폰으로 전송한다. 사용자는 스마트폰을 통해서 건물 내 현재 위치, 화재 발생 구역의 위치, 그리고 비상구를 향해 갈 수 있는 모든 대피 경로를 알 수 있다. 그러나 화재의 위치나 위험 정도에 따라 안전하지 않은 대피 경로가 존재할 수 있다. 현재 화재가 발생하지 않은 경로만을 고려하여 생성된 경로에 대해서 안전한 지 보장할 수 없다.

논문 [5]와 [6]은 공통적으로 LoRa (Long Range) 통신 기반 화재 대피 안내 시스템을 제안한다. 장거리 통신에 특화된 LoRa 기술을 기반으로 화재를 감지하는 센서와 대피 경로를 안내하는 LED (Light-Emitting Diode) 표시등을 활용한다. 화재 정보는 센서들을 통해서 중앙 서버로 전달 된다. 중앙 서버는 수신된 화재 정보를 활용하여, 건물 내 화재 상황을 알 수 있다. 이를 통해, 중앙 서버는 각각의 센서 노드 기준에서 가장 가까운 비상구를 계산하며, 이 결과를 각 LED 표시등에게 전달함으로써 각 위치에서 대피 경로를 안내할 수 있다. 따라서, 이는 사용자가 건물 구조에 대한 지식이나 별도의 디바이스 없이도, 오로지 LED 표시등을 따라 이동함으로써 비상구

에 도달할 수 있도록 한다. 그러나 두 연구에서 활용한 LoRa는 지하나 고층으로 갈수록 데이터 전송 성공률이 저하되는 특징을 갖는다. 따라서, 지상에 층이 낮은 건물에 대해 제한적으로 시스템 적용이 가능하다. 오늘날 복합 상가와 같은 대형 건물이나 초고층 건물에 적용하기에는 통신 신뢰성에 대한 문제로 적용에 제한이 있다.

논문 [7]과 [8]은 비콘 기반 화재 대피 안내 내비게이션 시스템을 제안한다. 본 시스템은 센서, LED 피난 유도등, 그리고 스마트폰을 활용한다. 센서들은 수집된 화재 정보 및 사용자의 위치 및 혼잡도를 중앙 서버로 전송한다. 중앙 서버는 수신된 데이터를 바탕으로 가장 안전한 경로를 계산한다. 논문 [7]의 경우, 계산된 경로에 따라 LED 피난 유도등에 대피 경로를 표시하도록 정보를 전달한다. 또한, 비상구에 설치된 혼잡도 센서를 통해서, 특정 비상구에 사람이 몰리지 않도록 예방한다. 논문 [8]의 경우, 계산된 경로를 사용자의 스마트폰에 전송하고, 이와 동시에 LED 피난 유도등에도 해당 경로 정보를 전달함으로써 대피 경로를 안내한다.

논문 [9]는 메시 센서 네트워크 (MSN, Mesh Sensor Network) 기반 화재 대피 안내 시스템을 제안한다. MSN은 건물 내부의 화재 감지 센서 노드들로 이루어지며, 화재 발생 시 화재 데이터는 센서 노드 간 멀티홉 통신 방식으로 중앙 서버에 전달된다. 중앙 서버는 건물 내부 지도와 수집된 화재 데이터를 사용하여, 화재가 발생한 구역으로 향하지 않도록, 화재로부터 안전한 길만 나타낸 지도를 생성한 후 서버에 저장한다. 그리고 이에 대한 정보를 확인할 수 있는 온라인 접속 링크를 사용자의 스마트폰으로 전송한다. 사용자는 해당 링크를 통해서 접속 한 후, 자신의 위치를 직접 입력하면 현재 사용자의 위치에서 가장 가깝고 안전한 비상구로 안내한다. 본 시스템은 사용자가 자신의 스마트폰에 별도의 애플리케이션을 설치하지 않아도, 화재 안전 내비게이션 서비스를 받을 수 있다. 그러나 본 서비스를 제공받기 위해서는 사용자의 스마트폰이 항상 인터넷에 접속이 가능해야 하고, 사용자가 자신의 위치를 사전에 알고 있어야만 한다.

앞서 살펴본 바와 같이, 관련 연구들은 화재 대피를 위한 경로를 계산할 때, 건물 내 설치된 다양한 센서들로부터 얻어진 정보를 활용한다. 이러한 정보들은 중앙 서버로 수집되고, 종합적인 판단을 통해 화재 대피 경로를 제공한다. 그러나, 건물 내 설치된 셀룰러 기지국 및 Wi-Fi AP들이 화재로 인해 손실될 경우 인터넷 접속이 불가능하기 때문에 제대로 된 서비스를

제공하기가 어렵다.

III. 화재 안전 내비게이션 시스템

본 장에서는 화재 안전 내비게이션 시스템 (Fire Safety Navigation System)에 대해 설명한다. 그림 2와 같이, 화재 안전 내비게이션 시스템은 센서 노드로 이루어진 화재 안전 네트워크, 화재 안전 내비게이션 디바이스, 그리고 화재 안전 서버로 구성된다. 또한 이 시스템에서 화재 안전 출구 서비스를 제공하기 위해 필요한 2가지 알고리즘을 제안한다. 하나는 화재 안전 네트워크에서 화재 안전 정보 전파를 위한 것이고, 다른 하나는 실내 지도와 화재 정보를 활용하는 안전 출구 내비게이션을 위한 것이다.

본 장은 다음과 같이 구성된다. 3.1절에서는 무선 센서 네트워크 기반의 화재 안전 네트워크에 대해 설명하고, 3.2절에서는 화재 정보 수집 및 안전 출구를 안내하는 화재 안전 내비게이션 디바이스에 대해 설명한다. 그리고 3.3절에서는 건물 전체의 화재 정보를 관리 및 제공하는 화재 안전 서버에 대해 설명한다. 이어서, 3.4절에서는 화재 정보의 전파를 위한 화재 안전 프로토콜을 설명하고, 3.5절과 3.6절에서는 화재 안전 프로토콜로 생성된 패킷을 처리하기 위한 알고리즘 및 흐름도를 설명한다. 마지막으로, 3.7절과 3.8절에서 안전 출구 내비게이션 알고리즘 및 흐름도에 대해 설명한다.

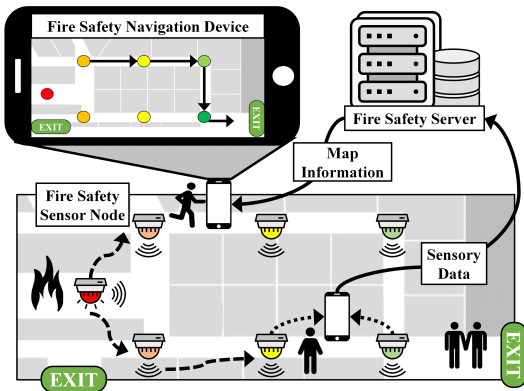


그림 2. 화재 안전 내비게이션 시스템
Fig. 2. Fire safety navigation system

3.1 화재 안전 네트워크

화재 안전 네트워크 (Fire Safety Network)는 무선 센서 네트워크 기술을 기반으로, 건물 내 발생한 화재 정보를 감지 및 수집하기 위해 구축된다. 이 화재 안

전 네트워크는 다수의 화재 안전 센서 노드 (Fire Safety Sensor Node)들로 구성되며, 이들은 스마트 빌딩 내 배치될 것이다. 화재 안전 센서 노드는 그림 3에서 볼 수 있듯이, 크게 3가지 부분으로 구성된다: 통신 (Communication), 데이터 관리 (Data management), 그리고 센싱 처리 (Sensing Process).

통신부 (Communication)는 센서 노드가 다른 센서 노드와 데이터를 교환하거나 모바일 디바이스에게 데이터를 전달하기 위해 필요하다. 이 통신부는 3가지 기능으로 구분된다: 블루투스 통신 (Communication over Bluetooth), 통신 스케줄러 (Communication Scheduler), 이웃 테이블 (Neighbor Table). 첫 번째, 블루투스 통신 기능은 인근 센서의 데이터를 수신하기 위해 사용된다. 저전력 기술이 적용되어 있기 때문에 배터리로 동작하는 대부분 디바이스들이 사용한다. 화재 안전 내비게이션 디바이스로서 모바일 디바이스를 사용하기 때문에 현실 적용을 위해 블루투스 기술을 채택한다. 두 번째, 통신 스케줄러 기능은 블루투스 통신의 동작을 결정한다. 통신 스케줄러에 정해진 일정대로 주변 신호를 수신하거나 패킷을 송신할 수 있다. 센서 노드는 데이터를 단방향 송신하는 것이 아니라, 다른 센서 노드의 신호를 수신할 수 있어야 한다. 그런데, 통신 모듈은 하나의 안테나로 송수신을 동시에 하지 못하기 때문에 송신과 수신을 위한 통신 모드를 전환하여야 한다. 따라서, 이 기능은 이러한 통신 모드의 전환에 대한 스케줄링을 담당한다. 세 번째, 이웃테이블 기능은 통신 대상에 대한 정보를 등록하고 관리한다. 센서 노드가 설치되는 환경은 일반적으로 주변에 다양하고 수많은 통신 장비들이 존재할 것이다. 이들에 대해 무분별한 수신을 하게 된다면, 불필요한 데이터 처리 과정이 일어나게 되고 이는 과도한 에너지 소모를 불러일으킬 수 있다. 게다가, 알려지지 않은 통신 장비들로부터 잘못된 정보가 수집될 수 있기 때문에 시스템에 오류를 발생시킬 수도 있다. 따라서, 센서 노드가 신호를 수신 시 해당 신호에

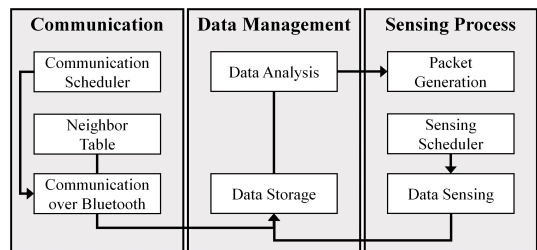


그림 3. 화재 안전 센서 노드 구조
Fig. 3. Architecture for fire safety sensor node

포함된 통신 장비 정보를 확인하기 위해 이웃테이블 기능이 필요하다.

데이터 관리부 (Data Management)는 다른 센서로부터 수신되거나 센서 노드 자체에서 생성된 센싱 데이터를 관리하는 기능을 수행한다. 이 데이터 관리부는 2가지 기능으로 구분된다: 데이터 저장소 (Data Storage)와 데이터 분석 (Data Analysis). 첫 번째, 데이터 저장소 기능은 블루투스 통신 기능을 통해 수신된 데이터와 센싱 처리부의 데이터 감지 (Data Sensing) 기능을 통해 감지된 데이터를 보관한다. 센서 노드는 제한적인 환경에서 효율적으로 동작하기 위해 만들어지기 때문에 적은 용량의 메모리를 갖는다. 따라서, 데이터는 공간 효율적인 방식으로 저장되어야 한다. 이러한 센서 노드에 적합한 파일 시스템 또는 압축 저장 방식 등이 채택된다. 두 번째, 데이터 분석 기능은 데이터 저장소에 저장된 데이터에 대한 분석을 수행한다. 수집된 혹은 감지된 데이터의 분석을 통해서, 화재 감지 여부를 판단하고 그 결과에 따른 적절한 동작을 수행한다. 데이터 분석을 위해서 본문문에서 제안하는 화재 안전 프로토콜에 따라 생성된 패킷을 처리하기 위한 알고리즘이 사용된다.

센싱 처리부 (Sensing Process)는 센서로부터 화재 정보를 감지하고, 이에 대한 정보를 패킷으로 만드는 기능을 한다. 이 센싱 처리부는 3가지 기능으로 구분된다: 센싱 스케줄러 (Sensing Scheduler), 데이터 센싱 (Data Sensing), 그리고 패킷 생성 (Packet Generation). 첫 번째, 센싱 스케줄러 기능은 센서를 통해 데이터를 얻어내는 센싱 주기를 관리한다. 센싱 주기는 센서 노드의 배터리를 사용하기 때문에, 센서 노드의 수명 연장을 위해서 센싱 주기는 스케줄러에 의해 관리되어야 한다. 이는 전원 관리부에 따라 센서 노드의 수명을 연장시키기 위해 센싱 스케줄러가 조정될 수 있다. 두 번째, 데이터 센싱 기능은 센서를 통해 데이터를 얻어내는 기능이다. 이는 센싱 스케줄러에 의해 정해진 주기대로 센서에게 감지 명령을 전달하는 과정을 수행한다. 이 과정을 통해 얻어진 데이터는 데이터 관리부의 데이터 저장소에 저장된다. 세 번째, 패킷 생성 기능은 화재 정보를 전달하기 위한 패킷을 생성한다. 패킷에는 화재 정보가 포함되며, 화재 정보를 전달하기 위해 개발된 프로토콜인 화재 안전 프로토콜(3.4절)에 따라 생성된다.

3.2 화재 안전 내비게이션 디바이스

화재 안전 내비게이션 디바이스 (Fire Safety Navigation Device)는 화재 안전 네트워크 및 화재 안

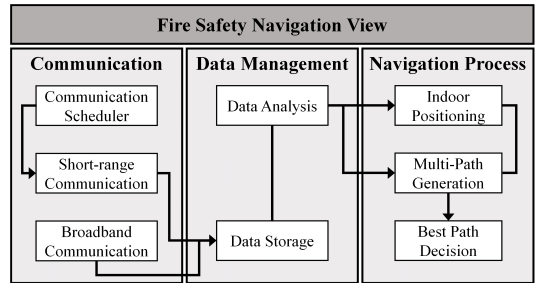


그림 4. 화재 안전 내비게이션 디바이스 구조
Fig. 4. Architecture for fire safety navigation device

전 서버로부터 수신한 정보를 기반으로 동작한다. 그림 4는 화재 안전 내비게이션 디바이스에 대한 구조를 나타낸 그림이다. 화재 안전 내비게이션 디바이스는 크게 5가지 부분으로 구성된다: 전원 관리 (Battery Power Management), 통신 (Communication), 데이터 관리 (Data Management), 내비게이션 처리 (Navigation Process), 그리고 내비게이션 화면 (View on Fire Safety Application).

전원 관리부 (Battery Power Management)는 화재 안전 내비게이션 디바이스가 동작하기 위해 필요한 전원을 관리한다. 화재 안전 내비게이션 디바이스는 사용자가 휴대하고 다니기 때문에 배터리로 동작하며, 이에 따라 배터리 전원 관리가 필요하다.

통신부 (Communication)는 인근의 화재 안전 센서 노드 및 원격의 화재 안전 서버와 통신하기 위한 역할을 한다. 이 통신부는 3가지 기능으로 구분된다: 통신 스케줄러 (Communication Scheduler), 단거리 통신 (Short-range Communication), 그리고 광대역 통신 (Broadband Communication). 첫 번째, 통신 스케줄러 기능은 화재 안전 내비게이션 디바이스와 화재 안전 센서 노드 간 통신을 위한 단거리 통신에 대한 통신 주기를 결정한다. 통신 할 때마다 에너지를 소모하기 때문에 적절한 조절이 필요하다. 두 번째, 단거리 통신 기능은 인근의 화재 안전 센서 노드로부터 화재 정보 수신에 가능하도록 한다. 통신 스케줄러에 의해 정해진 주기마다 정보 수신을 시도한다. 세 번째, 광대역 통신 기능은 원격의 화재 안전 서버와 통신을 가능하게 한다. 화재 안전 내비게이션 디바이스는 주변 화재 안전 센서 노드로부터 수신된 화재 정보를 화재 안전 서버에 전달하고, 이 화재 정보는 화재 안전 프로토콜 (3.4절)에 따라 생성된 패킷의 형태로 서버에 전달된다. 그리고 건물에 진입 시 화재 안전 서버로부터 건물 내 화재 안전 정보 및 건물 내 지도 정보는 사전에 제공된다.

데이터 관리부 (Data Management)는 통신부로부터 얻어지는 데이터를 관리하는 역할을 한다. 이 데이터 관리부는 2가지 기능으로 구분된다: 데이터 저장소 (Data Storage)와 데이터 분석 (Data Analysis). 첫 번째, 데이터 저장소 기능은 통신부의 단거리 통신과 광대역 통신 기능을 통해 수신된 데이터를 디바이스 내 저장한다. 저장되는 데이터는 주변의 화재 안전 센서 노드의 정보이거나, 원격 서버에서 수신된 건물 내 지도 및 화재 정보들이다. 두 번째, 데이터 분석 기능은 데이터 저장소에 있는 화재 정보들을 분석하여, 내비게이션 처리부와 연관된 기능들에서 요구하는 데이터를 각각 전달한다.

내비게이션 처리부 (Navigation Process)는 화재 안전 출구 서비스를 제공하기 위한 정보를 생성하는 역할을 한다. 이 내비게이션 처리부는 3가지 기능으로 구분된다: 실내 위치 측위 (Indoor Positioning), 다중 경로 생성 (Multi-Path Generation), 최선 경로 결정 (Best Path Decision). 첫 번째, 실내 위치 측위 기능은 화재 안전 내비게이션 서비스를 제공하는 데 필요한 기능이다. 사용자의 위치를 시작으로 가장 안전한 출구까지 대피 경로를 안내하기 때문이다. 본 기능은 근접성 (Proximity) 기반의 실내 측위 방식을 채택한다. 먼저, 주변 화재 안전 센서 노드가 주기적으로 발생시키는 센서 노드 정보를 수신한다. 그리고 신호 세기 (RSSI, Received Signal Strength Indicator)가 가장 높은 비콘의 정보를 선택한다. 그리고 나서 화재 안전 서버로부터 수신한 실내 지도 정보를 통해 해당 센서 노드가 설치된 위치를 파악한다. 두 번째, 다중 경로 생성 기능은 실내 위치 측위 기능을 통해 파악된 사용자의 위치로부터 건물 내 각 비상구를 목적지로 대피 경로를 생성한다. 이에 대한 결과로 비상구의 개수만큼 대피 경로가 생성된다. 비상구에 대한 정보는 화재 안전 서버로부터 제공받아 저장한다. 세 번째, 최선 경로 결정 기능은 다중 경로 생성 기능으로부터 생성된 대피 경로들 중 가장 안전한 대피 경로를 결정한다. 이에 고려사항은 다음과 같다. 대피 경로 상에 화재의 위험도가 낮아야 한다. 이는 화재가 발생한 장소로부터 가장 먼 곳, 대피 경로에 가연성이 낮은 물질이 있는 곳, 또는 대피자의 시야를 가리는 연기가 적은 곳 등을 위험도가 낮다고 판단한다. 이러한 정보들은 화재 안전 서버로부터 제공 받는다. 본 논문에서 제안하는 화재 안전 대피 경로 찾기 알고리즘은 이러한 고려사항들을 반영하기 위하여, 가중치를 적용한 개량된 다익스트라 (Dijkstra) 알고리즘을 활용한다.

내비게이션 화면부 (View on Fire Safety

Application)는 내비게이션 처리부에 의해 결정된 최선의 대피 경로를 화재 안전 내비게이션 디바이스의 화면에 나타낸다. 화면에는 현재 사용자가 위치한 건물 내 위치와 화재가 발생한 지역들을 보여준다. 게다가, 실시간으로 사용자의 위치로부터 가장 안전한 하나의 대피 경로를 화면에 나타낸다.

3.3 화재 안전 서버

화재 안전 서버 (Fire Safety Server)는 건물 전체의 화재 정보를 수집 및 관리하며, 건물 내 지도와 수집된 화재 정보를 바탕으로 건물 내 화재 안전 정보를 제공한다. 그림 5는 화재 안전 서버에 대한 구조를 나타낸다. 화재 안전 서버는 크게 2가지 부분으로 구성된다: 통신부 (Communication)와 데이터 관리부 (Data Management).

통신부 (Communication)는 화재 안전 내비게이션 디바이스로부터 데이터를 수집하고, 디바이스들에게 화재 정보를 전송하기 위한 기능을 담당한다. 이 통신부는 2가지 기능으로 구분된다: 화재 데이터 수신 (Fire Data Reception)과 화재 데이터 전송 (Fire Data Transmission). 첫 번째, 화재 데이터 수신 기능은 화재 안전 내비게이션 디바이스로부터 전송된 데이터를 수신하기 위해 사용된다. 화재 데이터는 건물 내 분포되어 있는 화재 안전 내비게이션 디바이스들에 의해 화재 안전 센서의 데이터가 상시로 수신될 수 있다. 두 번째, 화재 데이터 전송 기능은 화재가 발생한 층 또는 건물 내 존재하는 모든 화재 안전 내비게이션 디바이스들에게 화재 알림 및 대피 안내에 필요한 정보를 전달하기 위해 사용된다. 이는 Wi-Fi 또는 셀룰러 네트워크를 통해 이루어진다.

데이터 관리부 (Data Management)는 건물 내 화재 정보를 보관 및 분석하고 이를 통한 정보 처리 기능을 담당한다. 이 데이터 관리부는 3가지 기능으로 구분된다: 데이터 저장소 (Data Storage), 데이터 분석 (Data Analysis), 그리고 정보 처리 (Information Process).

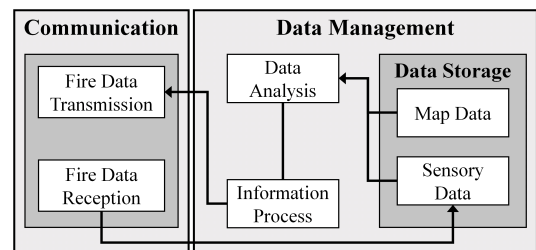


그림 5. 화재 안전 서버 구조
Fig. 5. Architecture for fire safety server

첫 번째, 데이터 저장소 (Data Storage)는 3가지 유형의 데이터를 보관한다. 지도 데이터 (Map Data)는 건물 내 각 층별 도면과 화재 안전 센서 노드가 배치된 위치 정보 등을 포함한다. 센서 데이터 (Sensory Data)는 건물 내 배치된 모든 센서들의 화재 관련 데이터이다. 두 번째, 데이터 분석 (Data Analysis)은 지도 데이터와 센서 데이터를 활용하여, 건물 전체에 대해서 화재가 발생한 지역을 분석한다. 세 번째, 정보 처리 (Information Process)는 요청된 데이터를 분석한 결과에 따라, 건물 지도 데이터를 요청하는 일부 사용자에게 해당 데이터를 전송할 지, 화재 발생에 대한 정보를 건물 내 모든 사용자에게 전송할 지에 대한 동작을 결정한다.

3.4 화재 안전 프로토콜

본 절에서는 화재 정보의 전파를 위한 화재 안전 프로토콜에 대해 설명한다. 화재 안전 센서 노드 및 화재 안전 내비게이션 디바이스 모두 본 프로토콜을 사용한다. 그림 6은 화재 안전 프로토콜에 대한 메시지 포맷을 나타낸다. 본 프로토콜은 블루투스의 BLE 메시지 포맷에서 데이터 영역에 포함된다. 제안하는 화재 안전 프로토콜은 크게 헤더와 데이터 영역으로 구분된다. 헤더 영역에는 데이터 처리를 위한 제어 정보가 포함되며, 데이터 영역에는 화재 정보를 갖는다.

화재 안전 프로토콜의 각 필드의 크기는 설치된 디바이스의 개수나 네트워크의 크기에 따라 달라질 수 있다. 화재 정보 전파는 긴급한 상황에 이루어지기 때문에, 신속하고 정확한 전송을 위해 패킷당 하나의 정보만을 포함해야 한다. 즉, 패킷의 단편화는 지원하지 않는다.

프로토콜의 필드에 관한 내용은 다음과 같다.

- **Sender ID:** 패킷을 전송한 센서 노드의 ID이며, 프로토콜의 헤더 영역에 속한다. 이 필드의 값을 이용하면, 패킷을 전송한 센서 노드를 알 수 있다. 자신

이 생성한 패킷을 전송 또는 다른 센서 노드로부터 수신한 패킷을 재전송할 때 자신의 ID로 수정한다.

- **Source ID:** 화재를 처음 감지한 센서 노드의 ID이며, 프로토콜의 헤더 영역에 속한다. 따라서 화재를 감지한 첫 번째 센서 노드의 Sender ID와 Source ID 필드의 값은 같다. 화재를 감지하지 않고, 단순 전달하는 센서 노드는 이 필드의 값을 수정하면 안 된다.
- **Urgent:** 긴급 상태를 의미하며, 프로토콜의 헤더 영역에 속한다. 이 필드의 값은 일반적으로 0이며, 화재를 감지한 경우 1이 된다.
- **Length:** 데이터 영역의 전체 길이를 의미하며, 프로토콜의 헤더 영역에 속한다. 데이터의 길이는 값에 따라 크기가 달라질 수 있기 때문에 길이를 명시적으로 지정해야 한다. 본 제안에서는 바이트 (byte) 길이를 의미한다. 예를 들어, 필드의 값이 3이면, 데이터 영역의 길이는 3 바이트이다.
- **Data:** 화재 정보를 나타내는 필드로 긴급 정도를 의미하며, 프로토콜의 데이터 영역에 속한다. 이 필드의 값은 0이상의 양의 정수이다. 최초로 화재를 감지한 센서 노드에서는 0이 되며, 패킷을 전달하는 센서 노드들은 이 필드의 값을 1씩 증가시키며 전달한다. 필드의 값이 클수록, 화재를 직접 감지한 노드로부터 더 멀어진다는 것을 의미한다.

3.5 화재 안전 센서 데이터 처리 알고리즘

화재 안전 네트워크를 구성하는 화재 안전 센서 노드들은 감지한 화재 정보를 네트워크를 통해 전파하기 위해 화재 안전 프로토콜을 통해 데이터를 교환한다. 화재 안전 네트워크는 제안된 프로토콜을 사용하여 네트워크 자체적으로 건물의 화재 정보를 저장하고 있다. 이를 위해서, 그림 7에서 보는 바와 같이, 각 센서 노드는 화재 데이터 처리 알고리즘에 따라 동작된다. 알고리즘의 변수들은 표 1에 정의되어 있다.

화재 안전 센서 노드는 통신부의 통신 스케줄러 기능을 통하여, 각 통신 상태에서 송신주기 $advIntvl$ 와 수신주기 $scanIntvl$ 를 등록한다. 그리고 주기적으로 각 통신 상태에 따른 정의된 기능을 수행한다. 이 알고리즘은 센서 노드의 배터리가 갖고 있는 에너지를 모두 소모할 때까지 동작한다. 송신 상태가 되었을 때, 자체 센서로부터 감지된 데이터와 인근 센서 노드로부터 수신된 데이터를 브로드캐스트 방식으로 전파 $advertising()$ 한다. 수신 상태가 되었을 때, 센서 노드는 이웃한 다른 센서 노드 $neighbors$ 의 데이터를 수신하기 위해 $scanning$ 을 수행한다. 수신된 데이터

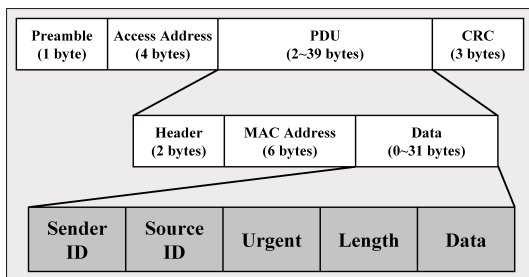


그림 6. 화재 안전 프로토콜 메시지 포맷
Fig. 6. Message format for fire safety protocol

표 1. 매개변수 목록
Table 1. List of Parameters

Term	Definition
<i>scanIntvl</i>	Scan interval of a sensor node
<i>advIntvl</i>	Advertisement interval of a sensor node
<i>T_{schedule}</i>	Scheduler for communication which includes <i>scanIntvl</i> and <i>advIntvl</i>
<i>Energy</i>	Energy of battery-powered sensor node
<i>neighbors</i>	List of adjacent sensor nodes
<i>advData</i>	Data to be advertised by the sensor node
<i>recvData</i>	Data advertised from <i>neighbors</i>
<i>newData</i>	Data refined by fire information analysis
<i>w_{Graph}</i>	Map with deployment of fire sensor nodes received from the fire server
<i>curLocation</i>	Source node which means current location of a user
<i>distance</i> []	Set of distances from <i>curLocation</i> to every sensor node
<i>routes</i> []	Set of routes which find a shortest path from <i>curLocation</i> to each sensor node on <i>w_{Graph}</i>
<i>visited</i> []	Set of boolean values about whether a sensor node is confirmed or not in shortest path search
<i>smallestIdx</i>	A sensor node which does not visit and has a shortest value on <i>distance</i> []
<i>newDist</i>	Calculated distance with a path which reaches to destination via a certain sensor node
<i>safeExit</i>	Destination node which means safest exit

*recvData*는 데이터 분석 기능을 통해 처리된다. 수신된 데이터에 포함된 정보를 분석함으로써, 센서 노드는 원격의 지역에서 발생한 화재 정보를 파악할 수 있다. 이를 위해 긴급 상태 *Urgent*의 값을 확인한다. 이 값이 1이면 화재가 발생됨을 의미하며, 0이면 그렇지 않은 상황을 의미한다. 긴급 상태가 0일 때는 센서 노

드는 아무런 동작도 하지 않으며, 1일 때는 이에 대한 정보를 다른 센서 노드에게 알리기 위해서 다음 동작을 수행한다. 다음으로, 데이터의 값을 1증가시킨 후 새로운 패킷을 생성한다. 그리고 생성된 패킷은 데이터 저장소에 저장되며, 다음 송신주기에 전송된다.

3.6 화재 데이터 처리 방안

그림 8은 화재 안전 센서 노드에서 화재 데이터 처리과정을 나타내는 흐름도이다. 센서 노드는 자신이 화재를 감지하거나 이웃 센서 노드들로부터 수신된 화재 데이터를 분석하고, 화재 발생이 감지된 경우 주변에 해당 정보를 전파한다.

센서 노드의 화재 데이터 처리 과정은 다음과 같다.

- 1) 센서 노드의 통신 모드를 확인한다.
- 2) 각 통신 모드에 적합한 동작을 수행한다.
- 2-1) 송신 모드인 경우, 센서 노드 내 저장소로부터 전송할 데이터를 가져오고, 그 데이터를 송신한다.
- 2-2) 수신 모드인 경우, 주변 센서 노드로부터 데이터를 수신한 후 분석한다.
- 2-2-1) 화재가 감지된 경우, 이웃한 센서 노드들에게 화재 정보를 포함한 데이터를 생성한 후 저장소에 넣는다.

Algorithm 1 Fire Safety Sensor Node

```

1: scanIntvl = SCAN_INTERVAL
2: advIntvl = ADVERTISEMENT_INTERVAL
3: Tschedule = commScheduler('ADV',advIntvl)
4: Tschedule = commScheduler('SCAN',scanIntvl)
5: while Energy != 0 do
6:   if Tschedule = 'ADV' then
7:     advData = DataSensing() + DataStorage.get()
8:     advertising(advIntvl,advData)
9:   end if
10:  if Tschedule = 'SCAN' then
11:    recvData = scanning(scanIntvl,neighbors)
12:    DataStorage(recvData)
13:    newData = DataAnalysis.execute()
14:    if newData.urgent = TRUE then
15:      DataStorage(newData)
16:    end if
17:  end if
18: end while
    
```

그림 7. 화재 안전 센서 데이터 처리 알고리즘
Fig. 7. Algorithm for fire safety sensor node

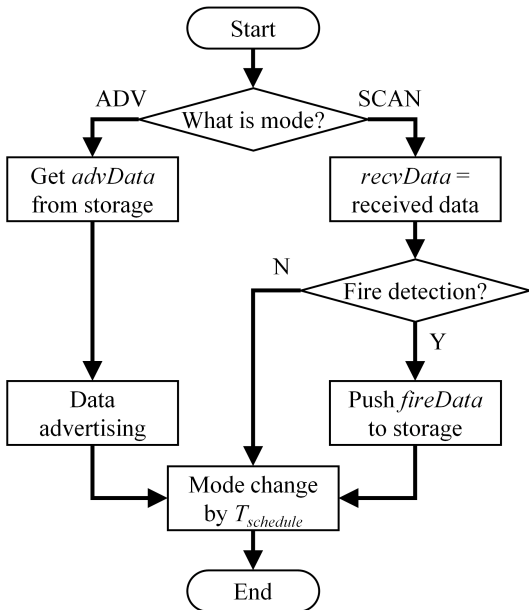


그림 8. 화재 데이터 처리 흐름도
Fig. 8. Flow chart for fire data processing

- 2-2-2) 화재가 감지되지 않은 경우, 아무런 동작도 하지 않는다.
- 3) 통신 스케줄러에 따라 송신 모드와 수신 모드는 전환된다.

3.7 안전 출구 내비게이션 알고리즘

그림 9의 안전 출구 내비게이션 알고리즘은 사용자에게 현재 위치에서 안전한 비상구에 도달할 수 있는 최단 경로를 알려주기 위한 알고리즘이 동작한다. 제안된 알고리즘은 최단 거리를 계산하는 다익스트라(Dijkstra) 알고리즘을 활용한다^[10]. 알고리즘에 사용된 변수들은 표 1에 정의되어 있다.

알고리즘에서는 이차원 배열 형태로 저장되어있는 건물 내 지도 정보를 활용한다. 이 자료구조에는 화재 센서 노드와 출구의 위치가 저장된다. 그래프의 행은 출발지 노드, 열은 목적지 노드를 뜻하며 행과 열의 조합으로 이루어진 그래프의 원소 값은 센서 간의 물리적 거리 및 출발지 노드에서 목적지 노드로 향할 때 길의 위험도를 나타낸다. 화재 발생 시 화재를 감지한 노드를 목적지로 하는 그래프 값을 화재 정도에 따라 증가시킨다. 따라서 그래프를 통해 노드에서 다른 노드로 향하는 직접적인 경로의 위험도를 알 수 있다.

이러한 정보를 담고 있는 데이터는 중앙 서버로부터 제공되며, w_{Graph} 로 표현된다. 그리고 사용자의 현재 위치 정보는 실내 측위 기법을 통해 알 수 있는 것

으로, $curLocation$ 로 표현된 (행: 1-2). 최단 거리 정보 $distance$ 배열은 현재 위치로부터 그래프 내 각각의 노드까지의 최단 거리를 저장한다. 경로 정보 $routes$ 배열은 그래프 내 각각의 노드까지의 최단 경로 정보를 저장한다. 최단 경로 노드 확인 정보 $visited$ 는 최단 경로를 구할 때 최단 경로에 포함된 노드인지 아닌지를 확인하는 배열이다. 세 배열 모두 배열의 인덱스는 목적지 노드를 뜻하며, 현재 아무것도 정해지지 않은 상태이기 때문에 각각 $INFINITE$, $UNDEFINED$, $FALSE$ 로 초기화한다 (행: 3-6). 그 다음 현재 출발지에 대한 최단 거리 $distance[curLocation]$ 은 0이므로 0으로 설정한다 (행: 8). 알고리즘은 그래프 w_{Graph} 내의 모든 노드를 방문할 때까지 반복되며, 가장 작은 거리 정보 배열의 값을 가지고 이전에 방문한 적 없는 노드부터 방문한다.

방문한 노드는 경로 포함 여부 정보가 $True$ 로 변경되며 각각의 노드를 목적지로 두었을 때 현재 방문해 있는 노드를 거치는 경우와 거치지 않는 경우에 대한 거리를 비교한다. 만약 방문한 노드를 거쳐 목적지 노드로 가는 경로가 더 작은 값일 경우 거리 정보가 갱신되며 그에 따라 경로 정보도 갱신된다 (행: 9-19). 각각의 노드를 목적지로 하는 모든 최단 경로를 모두 구한 후에 어떤 비상구가 현재 위치에서 가장 가까운지 확인한다 (행: 20). 그리고 사용자에게 해당 비상구에 도달할 수 있는 최선의 경로를 구한다 (행: 21).

Algorithm 2 Fire Safety Navigation Device

```

1:  $w_{Graph} = RecvGraph()$  // From the server
2:  $curLocation = IndoorPositioning()$ 
3: for Vertex  $v$  in  $w_{Graph}$  do
4:    $distance[v] = INFINITE$  // Init distance
5:    $routes[v] = UNDEFINED$  // Init routes
6:    $visited[v] = FALSE$  // Init visited
7: end for
8:  $distance[curLocation] = 0$ 
9: while all visited != TRUE do
10:   $smallestIdx = GetSmallestDistIdx(distance, visited)$ 
11:   $visited[smallestIdx] = TRUE$ 
12:  for Vertex  $v$  in  $w_{Graph}$  do
13:     $newDist = distance[smallestIdx] + w_{Graph}[smallestIdx][v]$ 
14:    if  $distance[v] > newDist$  then
15:       $distance[v] = newDist$ 
16:       $routes[v] = smallestIdx$ 
17:    end if
18:  end for
19: end while
20:  $safeExit = CalculateSafestExit(distance)$ 
21:  $BestPathDecision(safeExit, routes)$ 
  
```

그림 9. 안전 출구 내비게이션 알고리즘
Fig. 9. Algorithm for safe exit navigation

3.8 안전 출구 기반 대피 경로 결정 방안

그림 10은 안전 출구 내비게이션을 위한 가장 안전한 대피 경로를 결정하는 과정을 나타내는 흐름도이다. 대피 경로를 결정하기 위해서, 건물 내 화재 센서 노드 배치도를 기반으로 서버 및 주변 센서 노드들로부터 전달된 정보를 활용한다.

안전 출구 내비게이션을 위한 대피 경로 결정 과정은 다음과 같다.

- 1) 건물 내 배치되어 있는 모든 센서 노드의 정보 $nodes[]$ 와 사용자의 현재 위치 정보 v 를 활용한다.
- 2) 초기 사용자의 현재 위치에서 밀접한 센서 노드 $curr_node$ 를 선택한다.
- 3) 선택된 센서 노드를 기준으로 모든 센서 노드들까지 각각의 거리를 계산한다.
- 4) 계산된 결과에서 가장 짧은 거리 $distance$ 를 가진 센서 노드 $next_node$ 를 선정하며, $next_node$ 는 다음 경로 탐색의 대상으로 지정된다.
- 5) $next_node$ 와 현재 탐색 노드에 대한 거리 $distance[v]$ 를 비교하여, 가장 작은 값으로 갱신

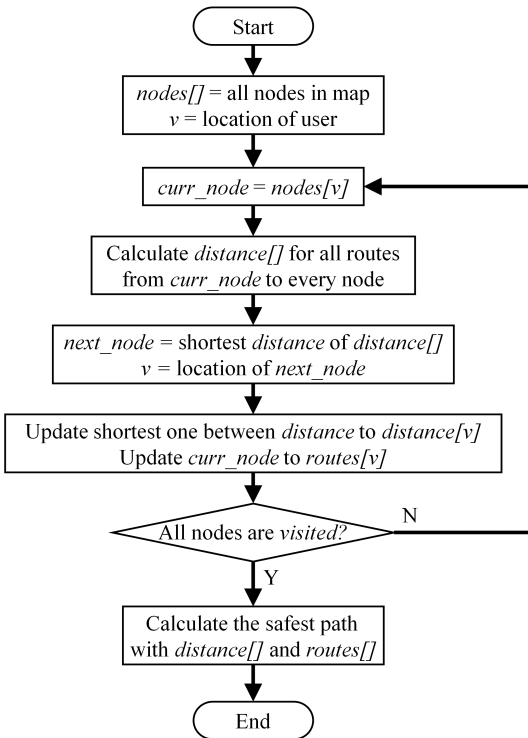


그림 10. 안전 출구 기반 대피 경로 결정 흐름도
Fig. 10. Flow chart for evacuation path decision by safe exit

한다. 이에 대응하여, $curr_node$ 를 $next_node$ 의 직전 방문 노드로 $routes[]$ 에 등록한다.

- 6) 모든 센서 노드에 대한 탐색이 종료될 때까지, 3~5 절차를 반복한다.
- 7) 탐색이 종료되면, $distance[]$ 와 $routes[]$ 를 기반으로 가장 안전한 경로를 계산한다.

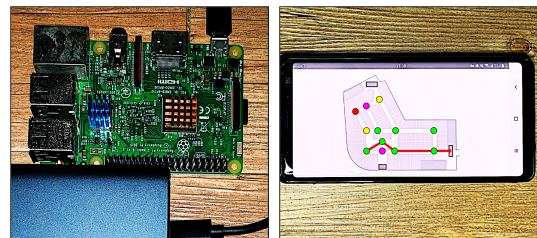
IV. 화재 안내 대피 시스템들의 비교 분석 및 성능 평가

본 장은 화재 안내 서비스를 제공하는 화재 대피 시스템의 기존 방안들과 제안 방안에 대한 성능을 비교 분석한다. 먼저, 화재 대피 시스템의 구조적 측면에서 성능 평가 및 비교 분석을 수행한다. 그리고 안내된 대피 경로에 대한 대피 실험을 수행한다. 이러한 실험을 수행하기 위해서, 프로토타입의 시스템을 개발 및 구축한다.

2장 관련연구를 통해서 다양한 화재 안내 대피 시스템에 대해서 살펴보았다. 본 실험에서는 화재 대피 경로에 대한 안내 방식에 대해서, 다음과 같은 기준을 만족하는 기존 방안들을 비교군으로 선정한다^{7,8)}. 1) 화재가 발생한 위치와 대피자의 위치 정보를 활용해야 한다. 2) 이들의 위치 정보를 바탕으로 대피 경로를 생성해야한다. 3) 생성된 대피 경로 정보는 대피자에게 제공되어야 한다.

4.1 실험 환경

본 논문에서 제안하는 시스템에 대한 성능 및 검증하기 위해 실험 환경은 다음과 같이 구성된다. 무선 센서 네트워크를 구성하는 센서 노드는 그림 11(a)에서와 같이, 블루투스 모듈 (Bluetooth 4.2)가 탑재된 라즈베리파이 (Raspberry Pi 3 Model B+)를 사용한 다¹¹⁾. 화재 안내에 사용되는 디바이스는 그림 11(b)에서와 같이, Android 10 OS가 동작하는 삼성 갤럭시



(a) 센서 노드

(b) 스마트폰

그림 11. 실험 장비
Fig 11. Experimental Devices

노트 9 (SM-N960)을 사용한다. 그리고 실험을 위해 개발한 안드로이드 애플리케이션이 동작된다. 중앙 서버는 리눅스 OS (Linux 18.04)가 동작하는 AWS (Amazon Web Service)의 EC2 인스턴스를 사용한다^[12]. 그리고 실험을 위해 장고 (Django)를 통해 개발한 서버 프로그램이 동작된다.

모든 실험은 12개의 센서 노드와 3개의 출구 노드로 구성된 네트워크를 기반으로 수행된다.

4.2 중앙 서버의 존재 유무에 따른 대피 경로 안내 영향

본 실험은 기존 방안과 제안 방안 각각에 대하여, 중앙 서버의 존재 유무가 화재 대피 안내 시스템에 의한 대피 경로 결과에 미치는 영향을 알아보기 위함이다.

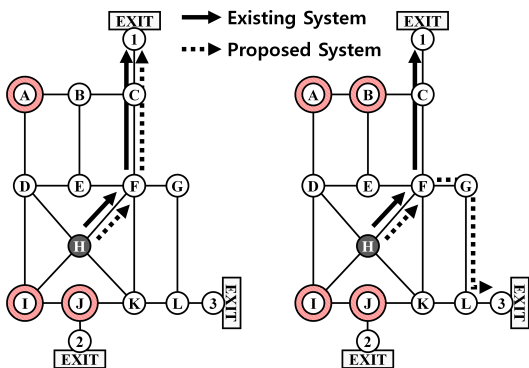
그림 12(a)는 중앙 서버가 정상 동작하며, 모든 화재 정보에 대한 정보가 중앙 서버에 제대로 수집되고 있을 때 특정 지역에서 화재가 발생한 상황을 나타낸다. 센서 노드 A, I, J는 화재 감지 센서를 통해서 화재를 감지하고, 이에 대한 정보를 화재 안전 프로토콜에 담아서 주변에 전파한다. 초기 화재를 감지한 센서 노드들이기에, 프로토콜에서 데이터 영역의 값은 0이 된다. 그리고, 대피자는 자신의 모바일 디바이스를 통해서 주변에서 신호 세기가 가장 가까운 센서 노드의 정보를 자신의 위치로 지정한다. 즉, 본 실험에서는 센서 노드 H의 주변에 있음을 알 수 있다. 이때, 기존 방안과 제안 방안은 모두 센서 노드 F와 C를 거쳐서 1번 출구 (EXIT 1)로 대피하는 경로를 나타낸다. 이 결과가 도출된 이유는 전체 화재 정보를 알고 있는 상황에서, 두 방안 모두 대피자의 현재 위치로부터 대피

할 수 있는 출구 중 최단 경로인 1번 출구를 선택하기 때문이다. 대피자의 위치에서 2번 출구 (EXIT 2)가 거리상으로는 가장 가깝지만, 화재가 발생 한 위치이기 때문에 실질적으로는 대피가 가능하지 않은 출구이다. 따라서, 대피 경로에서는 제외된다.

그림 12(b)는 중앙 서버가 화재로 인해 동작이 정지됨에 따라, 화재 정보는 중앙 서버로부터 더 이상 수집되지 않는 상황이다. 그리고 그림 12(a)의 상황에서 연속된 것으로, 화재의 확산에 따라, 센서 노드 A, I, J 뿐만 아니라, 센서 노드 B에서도 화재가 감지된다. 따라서, 센서 노드 B도 화재를 직접 감지하였기 때문에 프로토콜의 데이터 영역의 값은 0이 된다. 그러나 중앙 서버가 손실되었기 때문에 센서 노드 B의 화재 정보는 중앙 서버에 수집되지 않는다. 이때, 중앙 서버에 의한 정보를 바탕으로 화재 안전 대피 경로를 안내하는 기존 방안은 센서 노드 B에 대한 정보를 알 수 없기 때문에, 이를 반영하지 못한 결과로 그림 12(a)와 동일한 대피 경로를 나타낸다. 따라서, 대피자가 해당 대피 경로로 이동하게 된다면, 화재가 진행된 방향에 따라 인명 사고로 이어질 수 있는 위험이 있다. 이에 반해, 제안 방안은 센서 노드 F, G, 그리고 L을 거쳐 3번 출구 (EXIT 3)로 대피하는 경로를 나타낸다. 이는 센서 노드로 구성된 화재 안전 네트워크를 통해서 센서 노드 B의 화재 정보가 전파되기 때문에, 대피자의 위치가 센서 노드 B와 원거리에도 있음에도 불구하고 이에 대한 정보를 수신할 수 있다. 즉, 센서 노드 B가 전파한 프로토콜을 이웃한 센서 노드 C와 E가 수신하게 되고, 이들은 수신된 프로토콜에서 데이터 영역의 값을 증가시킨 1을 담아 재전파한다. 그리고 센서 노드 F에서는 같은 방식으로 데이터 영역의 값이 2가 된다. 게다가, 센서 노드 J에서도 화재 정보가 전파되기 때문에, 센서 노드 K는 데이터 영역의 값이 1이 되고 센서 노드 L에서는 2가 된다. 이 때, 사용자의 위치 H와 인근 센서 노드 F와 K의 데이터 값에 대한 알고리즘 2의 결과는 센서 노드 F, G, L을 거치는 경로가 된다. 이는 데이터 영역의 값이 가장 높은 곳을 안전한 경로로 택하기 때문이다. 이처럼, 제안 방안은 화재 안전 내비게이션 디바이스에서 자체적으로 안전 대피 경로를 계산하기 때문에, 중앙 서버가 없어도 대피자의 위치에서 가장 안전한 대피 경로를 제공할 수 있다.

4.3 중앙 서버의 부재에 따른 대피 경로 안내 영향

본 실험은 중앙 서버의 부재인 상황에서도, 제안 방안이 안전한 출구로 가는 대피 경로를 계산 할 수 있



(a) 중앙 서버 존재 (b) 중앙 서버 부재

그림 12. 중앙 서버의 존재 유무에 따른 대피 경로
Fig. 12. Evacuation path by existence of central server

는 지 알아보기 위함이다. 4.2절의 실험은 중앙 서버가 동작 도중 화재 사고로 인한 기능 유실이 발생한 상황을 가정한다. 그러나 처음부터 중앙 서버의 기능이 유실된 상황도 발생할 수 있다. 이러한 경우, 중앙 서버가 가진 전체 정보가 아닌 대피자 주변의 지역 정보만을 이용하여 대피 경로를 안내할 수 있어야 한다. 따라서 제안 방안은 화재 안전 네트워크를 통해 수집된 지역 정보만을 활용하여 경로를 최적화하는 알고리즘을 사용한다. 이를 위해, 동일한 화재 상황에서 전체 정보를 기반으로 경로를 최적화한 결과와 지역 정보를 기반으로 경로를 최적화한 결과를 비교하는 실험을 진행한다.

그림 13에서 (a)와 (b)는 모두 동일한 화재 상황을 나타낸다. 이를 통해서, 화재 위치는 센서 노드 A, I, J의 주변이며, 대피자의 위치는 센서 노드 F의 주변이다. 따라서 센서 노드 A, I, J는 감지한 화재 정보를 주변에 전파한다. 프로토콜의 데이터 영역은 0이 된다. 그리고 모바일 디바이스의 주변에 센서 노드 E, F, H, G가 인접하지만, 모든 센서 노드는 전파 세기가 모두 동일하기 때문에 신호 세기의 감도가 가장 높은 센서 노드가 실제 거리상 가장 가까운 센서 노드가 된다. 따라서 센서 노드 F가 자신의 위치가 된다. 그리고 안전한 출구는 1번 출구와 3번 출구이다. 그림 13(a)는 모든 센서 노드의 화재 정보를 알고 있다는 가정하에, 알고리즘 2에 의해 계산된 가장 안전한 대피 경로를 나타낸다. 센서 노드 F의 인접하는 센서 노드 G의 긴급 정도가 제일 높기 때문에, 알고리즘 2의 결과에 따라 대피 경로는 센서 노드 F, G, L, 3번 출구(EXIT 3) 순서로 결정 된다. 한편, 그림 13(b)는 현재 대피자의 위치에서 수신되는 센서 노드 F의 패킷에 포함된 정보를 통해 다른 센서 노드들의 화재 정

보를 유추한 결과이다. 중앙 서버의 정보 없이 화재 무선 센서 네트워크 전체에 대한 화재 정보를 유추하기 위해서, 대피자의 화재 안전 내비게이션 디바이스는 자신과 가장 가까운 센서 노드 F로부터 수신된 패킷을 활용한다. 이에 따라, 대피자의 디바이스는 Source ID가 A, I, J인 프로토콜을 수신하게 된다. 사전에 알고 있는 지도 데이터를 바탕으로 화재를 직접 감지한 센서 노드부터 자신의 위치까지, 데이터 값을 1씩 증가시킨다. 이 과정에서, 화재 데이터의 값이 다른 패킷이 지나갈 경우, 데이터 값이 낮은 것으로 한다. 화재는 매우 위험하기 때문에, 본 제안 방안은 최악의 상황을 고려한다. 따라서 센서 노드 F에는 센서 노드 A에서 시작된 결과로 데이터가 3이 되고, 센서 노드 I와 J에서 출발한 결과는 동일하게 데이터가 2이다. 조건에 따라, 센서 노드 F의 데이터는 2가 된다. 최종적으로, 그림 13(b)과 같은 정보를 유추할 수 있다. 그 결과, 그림 11(a)의 실제 정보와 달리, 센서 노드 B에서 화재가 감지된 걸로 판단이 된다. 그럼에도 불구하고, 유추한 결과를 바탕으로 알고리즘 2를 수행하면 그림 13(a)와 동일한 대피 경로가 나타남을 알 수 있다. 이는 긴급 정도가 높은 방향으로 대피하도록 경로가 생성되기 때문이다.

본 실험의 결과에 따라, 제안 방안이 중앙 서버로부터 수신된 정보 없이도 안전한 대피 경로를 안내할 수 있다는 것을 알 수 있다. 다시 말해서, 대피자의 주변 센서 노드로부터 수신된 데이터만 활용해도 가능하다는 것이다. 이는 제안된 화재 안전 프로토콜에 포함된 화재 발생 장소와 화재로부터 상대적 거리 등을 알 수 있기 때문이다.

4.4 안내된 대피 경로에 따른 대피 소요 시간

화재는 시간이 지날수록 그 위험도가 높아지기 때문에, 화재 위험지역에서 가능한 빠르게 벗어나는 것이 중요하다. 본 실험은 각 방안에 의해 결정된 대피 경로를 따라 대피할 때 대피 소요 시간에 대해 알아보기 위함이다. 실험에서는 출구2에서 화재가 발생한 상황을 가정한다. 따라서 대피 경로는 출구1과 3에 대하여 생성되고, 대피자는 생성된 경로를 따라 이동한다. 한편, 대피자의 신체 조건에 따라 이동 속도가 다를 수 있다. 실험에서 이를 위해 대피자의 이동 속도는 성인 평균 달리기 속도인 2~3m/s로 적용한다. 그림 14는 대피 경로 별 대피 소요 시간에 대한 실험결과를 나타낸다. 그림 14에서 가로축은 대피 실험 횟수를 나타내며, 눈금 한 칸 당 100번 수행한 결과에 대한 평균을 의미한다. 그리고 세로축은 대피 시간을 나타낸다.

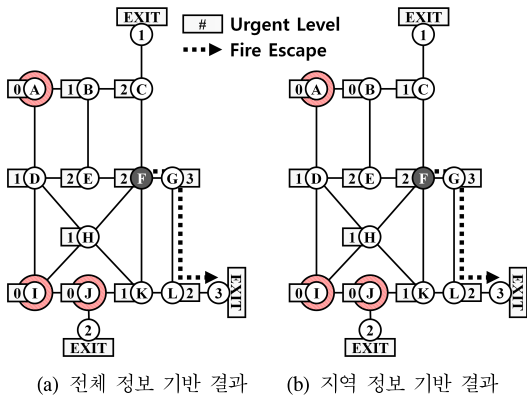


그림 13. 중앙 서버의 부재에 따른 대피 경로
Fig. 13. Evacuation path by loss of central server

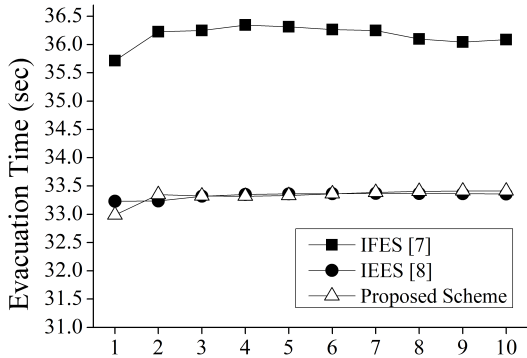


그림 14. 안내된 대피 경로에 따른 대피 소요 시간
Fig. 14. Evacuation time by guided evacuation path

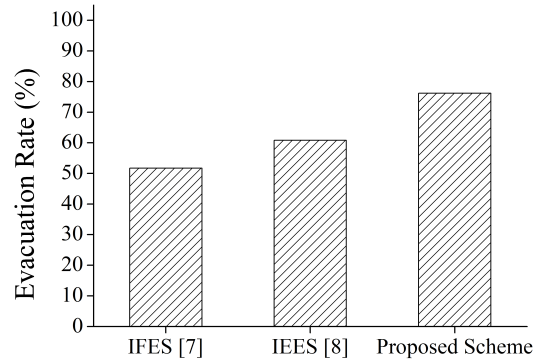


그림 15. 안내된 대피 경로에 따른 대피 성공률
Fig. 15. Evacuation rate by guided evacuation path

그림 14에서 볼 수 있듯이, IFES는 다른 방안들보다 대피 소요 시간이 가장 길다. IFES는 혼잡도에 따라 출구 별 다수의 대피 경로를 생성할 수 있다. 실험에서 대피 경로는 출구1에 대하여 3개의 경로가 생성되고, 출구3에 대하여 2개의 경로가 생성된다. 출구1에 대한 대피 거리는 90~100m가 이며, 출구3의 경우에는 80~90m이다. 따라서 IFES는 모든 경로에 대하여 대피자가 출구로 대피하기 까지 걸리는 평균 시간은 약 36초가 소요된다.

IEES와 제안 방안의 경우, 대피 소요 시간이 유사함을 알 수 있다. 이는 대피 경로를 생성할 때 IEES와 제안 방안 모두 대피 거리 및 화재 위험도를 고려하기 때문이다. IEES에서는 화재 위험 정도로서 온도 및 화재로 인한 연기 정도를 고려하며, 제안 방안에서는 화재와의 인접성을 고려한다. 이에 따라, 본 실험에서 두 방안에서 생성된 경로는 동일한 하나의 경로를 결정하게 되며, 이동 거리는 80m가 된다. 이 거리에 대해서 대피 시간을 측정했을 때, 평균 약 33초가 소요된다.

4.5 화재 대피 경로에 따른 대피 성공률

화재 대피 시스템으로부터 제공된 대피 경로를 통해 화재 지점으로부터 벗어나 안전한 장소에 도달하는 것이 매우 중요하다. 본 실험을 통해 각 방안에 따른 대피 성공률을 알아보고자 한다. 화재가 발생 후 대피자는 각 방안에 따라 결정된 대피 경로가 주어진다. 그리고 대피자는 안내된 경로를 따라 출구까지 이동한다. 이때, 화재는 시간이 지남에 따라 확산된다. 따라서 화재 확산으로 인해 출구가 사라지기 전에 대피자가 출구까지 도달하는 지로 대피 성공 여부를 판단한다. 그림 15에서 가로축은 비교 방안 및 제안 방안을 나타내며, 세로축은 대피 성공률을 나타낸다.

그림 15에서 볼 수 있듯이, IFES는 다른 방안들에 비해 매우 낮은 대피 성공률을 보인다. IFES는 대피 경로를 계산할 때 건물에 비상구별 사람의 혼잡도만 고려하여 출구 별 대피 경로를 생성한다. 이에 따라, 대피 경로는 총 3개이며, 각 경로별 이동 거리는 15m, 25m, 80m이다. 실험 결과로, 이동 거리가 80m인 경로를 따라 대피했을 경우 100%의 확률로 대피 성공률을 보인다. 반면, 거리가 짧은 15m와 25m인 경우에는 각각 30%와 33%의 대피 성공률을 보인다. 이는 화재의 확산에 따른 경로 차단이 대피 실패로 이어지는 데, 15m와 25m는 화재가 발생한 장소와 가까웠기 때문이다. 결과적으로, IFES를 통해서 제공될 수 있는 3가지 경로에 대한 대피 성공률은 약 51%가 된다.

IEES는 대피 경로를 계산할 때 대피자로부터 각 비상구까지 거리, 연기의 확산 정도 및 온도를 고려한다. IEES는 화재 상황 및 대피에 영향을 줄 수 있는 정보들을 고려하여, 계산한 다수의 대피 경로 중 가장 안전한 하나의 대피 경로를 제공한다. 본 실험에서 대피 경로는 총 2개가 되며, 출구 2와 출구 3에 대한 경로 별 이동 거리는 15m와 75m이다. 출구1은 IEES의 방안에 따라 위험도가 높기 때문에 대피 경로 계산에서 제외된다. IEES에 따라, 가장 안전한 대피 경로는 출구2에 대한 경로가 된다. 실험 결과, 이 경로는 출구와 화재 지점이 인접하기 때문에, 대피 성공률은 약 10%이다. 한편, IEES는 대피 실패 시 다른 경로를 제공한다. 출구2에 대한 경로로 대피가 실패했을 때 출구3에 대한 경로가 제공된다. 이때 새로운 경로로 대피에 성공할 확률은 약 52%가 된다. 결과적으로, IEES를 통해 제공된 모든 경로에 대한 대피 성공률은 약 61%이다.

반면, 제안 방안은 다른 방안들 보다 높은 대피 성공률을 나타낸다. 제안 방안은 IEES와 유사하게, 대피

경로의 실제 거리 및 화재 위험도를 고려한다. 그러나, 제안 방안은 IEES와 달리 대피 경로를 계산할 때 고려 요소들에 대해 우선순위를 적용한다. 대피 경로는 최단 거리로 화재 지역을 대피하는 것이 효과적인 수 있으나, 화재 위험도가 높은 지역을 지나갈 경우 대피 도중 인명 사고가 발생할 수 있다. 따라서 제안 방안은 화재 위험도를 우선 고려하는 방식을 택한다. 이에 따라, 본 실험에서 대피 경로는 출구3에 대한 경로가 제공된다. 이는 화재 발생지로부터 멀어지는 경로이기 때문에 대피에 가장 안전하다. 그러나 해당 경로에 대한 이동 거리는 가장 긴 90m이기 때문에, 대피 시간이 오래 걸릴 수 있다. 만일, 화재의 확산이 빠르게 이어질 경우 대피에 실패할 수 있다. 이에 따른 실험 결과로, 제안 방안의 대피 성공률은 약 80%가 된다.

V. 결론 및 향후 연구

본 논문은 스마트 빌딩에서 화재 발생 시 안전 출구를 안내해주는 화재 안전 내비게이션 시스템을 제안한다. 중앙 서버에 의존적인 기존 방안들은 화재로 인한 서버의 손실이 발생하면 더이상 안전 대피 경로를 안내하지 못하는 문제가 있다. 이 문제를 해결하기 위해, 본 논문에서는 무선 센서 네트워크 기술과 스마트폰 애플리케이션을 활용한 스마트 화재 안전 내비게이션 시스템이 제안된다. 실험을 통해서, 제안 방안은 기존 방안들과 다르게 중앙 서버가 동작할 수 없는 상황에서도 건물 내부에 배치된 무선 센서 노드 간의 화재정보 교환을 통해 화재 정보를 확산시킬 수 있으며, 이를 통해 대피자들에게 화재 대피에 유용한 정보를 제공할 수 있다. 그리고 제공된 화재 대피 경로에 따른 대피 실험을 수행한 결과로, 다른 방안에 비해 매우 높은 대피 성공률을 보인다.

이처럼 우수한 성능을 보임에도 불구하고, 본 논문의 제안 방안은 확장성 측면에서 한계를 갖는다. 따라서 내부 구조가 복잡하고 출입구가 여러 개인 오늘날 대형 건물들이 출현하는 환경에서 반드시 고려되어야 하는 요구사항이 될 것이다. 본 제안 방안은 네트워크 전체 정보를 기반으로 동작하기 때문에, 실시간 서비스를 제공하기 위해서는 화재 안전 네트워크의 전반에 화재 정보가 전파되어야 한다. 그러나, 네트워크 크기의 확장에 따라 전파 속도가 저하되는 문제가 있다. 이에 따라, 향후 화재 정보의 신속한 전파를 위한 방안에 관해 연구하고자 한다.

References

- [1] A. Verma, S. Prakash, V. Srivastava, A. Kumar, and S. C. Mukhopadhyay, "Sensing, controlling, and IoT infrastructure in smart building: A review," *IEEE Sensors J.*, vol. 19, no. 20, pp. 9036-9046, Oct. 2019.
- [2] N. Havard, S. McGrath, C. Flanagan, and C. MacNamee, "Smart building based on internet of things technology," *2018 12th Int. Conf. Sensing Technol.*, pp. 278-281, Limerick, Ireland, Dec. 2018.
- [3] B. Park, S. Park, Y. Kim, H. Kim, G. Jang, and S. Kim, "Beacon-based intelligent fire evacuation route guidance system," in *Proc. Symp. KICS*, pp. 440-441, Nov. 2020.
- [4] D. Kim, T. Lee, J. Kim, and J. Lee, "Design of indoor positioning system based on bluetooth low energy for fire evacuation environment," in *Proc. Symp. KICS*, pp. 400-401, Jan. 2018.
- [5] W. Wang, et al., "Dijkstra algorithm based building evacuation recognition computing and iot system design and implementation," *2019 IEEE 13th Int. Conf. Anticounterfeiting, Secur., and Identification*, pp. 229-233, Xiamen, China, Oct. 2019.
- [6] J. Shin, S. Eun, S.-S. So, and B. Kim, "Design and implementation of lora-based emergency exit guide system," *J. KIICE*, vol. 22, no. 4, pp. 569-574, Apr. 2018.
- [7] A. Khan, A. A. Aesha, J. S. Aka, S. M. F. Rahman, and M. J. Rahman, "An IoT based intelligent fire evacuation system," *2018 21st ICCIT*, pp. 1-6, Dhaka, Bangladesh, Dec. 2018.
- [8] I. A. Zualkernan, F. A. Aloul, V. Sakkia, H. A. Noman, S. Sowdagar, and O. A. Hammadi, "An IoT-based emergency evacuation system," *2019 IEEE Int. Conf. IoTAIS*, pp. 62-66, BALI, Indonesia, Nov. 2019.
- [9] S. Majumder, S. O'Neil, and R. Kennedy, "Smart apparatus for fire evacuation - An IoT based fire emergency monitoring and evacuation system," *2017 IEEE MIT*

Undergraduate Res. Technol. Conf., Cambridge, MA, pp. 1-4, Nov. 2017.

- [10] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269-271, Dec. 1959.
- [11] *Raspberry Pi 3 Model B+*, Retrieved Dec. 29, 2021, from <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>
- [12] *Amazon Web Services EC2*, Retrieved Dec. 29, 2021, from <https://aws.amazon.com/ko/ec2>

서 동 영 (Dongyeong Seo)



2015년 2월 : 숭실대학교 정보통신전자공학부 졸업
2017년 2월 : 숭실대학교 정보통신공학과 석사
2020년 3월~현재 : 충남대학교 컴퓨터공학과 박사과정

<관심분야> Wireless Sensor Networks, Internet of things

[ORCID: 0000-0002-4568-2709]

심 승 민 (Seungmin Sim)



2017년 3월~현재 : 충남대학교 컴퓨터공학과 학사과정
<관심분야> Internet of things, Wireless communication

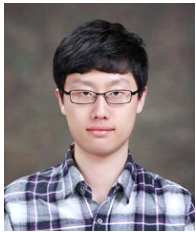
김 상 하 (Sang-Ha Kim)



1980년 : 서울대학교 학사
1984년 : University of Houston 석사
1989년 : University of Houston 박사
1992년~현재 : 충남대학교 컴퓨터공학과 교수

<관심분야> Wireless Sensor Networks, Internet Routing, MANET, Mobile Communication 등

양 태 훈 (Taehun Yang)



2014년 8월 : 충남대학교 컴퓨터공학과 졸업
2014년 9월~현재 : 충남대학교 컴퓨터공학과 박사수료
<관심분야> Mobile sensing, Wireless communication, Networking technology,

Internet of things 등

[ORCID:0000-0001-8584-8868]