

다중 드론 애드혹 네트워크를 위한 단계적 자율적인 망 복구 기술

김 건 환*, 조 유 제^o

A Stepwise Network Self-Recovery Scheme for Multi-drone Ad Hoc Networks

Geon-Hwan Kim*, You-Ze Cho^o

요 약

드론은 재난 상황에서의 인명구조와 수색, 군사 분야에서의 정찰임무, 농업 분야에서의 농작물 관리 및 감지와 같은 다양한 활용 분야를 가진다. 이와 같은 폭 넓은 활용 분야를 위해서는, 다수의 드론을 운용하는 다중 드론 시스템이 전체 임무의 효율성을 크게 향상시킬 수 있다. 게다가, 다수의 드론을 사용하여 다중 홉 통신을 구성하는 경우에는 드론의 활동 영역을 넓힐 수 있다. 다중 드론 시스템에서는 드론과 컨트롤러 사이에 안정된 네트워크를 구성하는 것이 무엇보다 중요하다. 드론과 컨트롤러 사이의 연결이 불안정한 경우에는 임무의 실패뿐만 아니라 드론의 제어가능 범위 이탈로 인한 드론의 추락과 같은 2차 사고가 발생할 수 있다. 본 논문은 다중 드론 애드혹 네트워크 환경에서의 자율적인 네트워크 복구 기술을 제안한다. NS-3를 이용한 시뮬레이션을 통해서 제안하는 다중 드론 애드혹 환경에서의 단계적인 자가 복구 기술은 네트워크 비 복구 환경에 비해 패킷 전달률과 평균 네트워크 단절 시간 측면에서 높은 성능 향상을 보였다.

Key Words : FANETs, multi-drone networks, neighbor locations, self-recovery, drone

ABSTRACT

Drones have various applications, such as rescue and exploration in disaster situations, surveillance in military, monitoring and management of crops in agriculture field. For such a wide variety of applications, a multi-drone system using several drones can further improve the efficiency of the tasks. In addition, the multi-hop communication using multiple drones can enhance the radius of operation. In multi-drone systems, it is important to establish a stable network between the drone and controller. This paper focuses on a network self-recovery scheme in multi-drone ad hoc network environments. We compare the proposed scheme with the network non-recovery environment, and confirm that the proposed scheme outperforms the network non-recovery scheme in terms of the packet delivery ratio and the average network disconnection time.

I. 서 론

드론과 관련된 다양한 연구의 진행에 따라 광범위

한 기술적 진보가 이루어졌고 실생활 및 군사 응용 분야에서의 드론 활용이 더욱 다양해지고 있는 추세이다. FANET(Flying Ad Hoc Network)에 대한 지금까

※ 본 논문(저서)은 2017학년도 경북대학교 연구년 교수 연구비에 의하여 연구되었음.

• First Author : Kyungpook National University, School of Electronics Engineering, kgh76@ee.knu.ac.kr, 학생회원

◦ Corresponding Author : Kyungpook National University, School of Electronics Engineering, yzcho@ee.knu.ac.kr, 중신회원

논문번호 : KICS2017-09-274, Received September 28, 2017; Revised December 6, 2017; Accepted December 12, 2017

지의 연구는 비행체간에 데이터를 효율적으로 전달하기 위한 최적의 라우팅 알고리즘과 군집 드론 운용 분야에서의 군집 드론에 대한 경로 계획 방법에 주로 중점을 두어서 진행되었다. 실제로 드론의 높은 이동성과 급변하는 주변의 환경으로 인해 드론들 사이에 링크 단절이 자주 발생함에도 불구하고, 드론들 간의 네트워크 연결성에 중점을 둔 연구는 많이 진행되지 않았다¹¹⁾.

본 논문에서는 FANET 환경에서의 자율적 네트워크 복구 방법에 초점을 두고 있다. 일반적인 단일 드론 시스템에서의 네트워크 복구는 드론과 컨트롤러 사이에 연결이 끊어지는 상황이 발생하고 이를 해결하는 과정으로 볼 수 있다. 이 때의 단일 드론 시스템은 하나의 컨트롤러와 하나의 드론으로 구성되는 시스템이다. 단일 드론 시스템에서는 “리턴 투 홈” 기술을 통해 네트워크 복구가 이루어진다. 드론이 컨트롤러의 제어 가능 범위를 벗어나거나 시야에서 사라지는 경우에 “리턴 투 홈” 버튼을 눌러서 드론의 초기 이륙 지점으로 되돌아오게 한다.

본 논문에서 고려하는 FANET 환경은 다수의 드론과 하나의 지상제어 센터를 갖는 다중 드론 시스템이다. 단일 드론 시스템과는 달리, 다중 드론 시스템은 많은 수의 드론을 활용하여 이들의 협업을 통해 더 넓은 작업범위를 갖게 하고 임무수행의 효율성 또한 증가시킨다¹²⁾. 드론의 작업범위 확대는 다중 홉 통신을 이용하여 드론 간의 협업을 통해 이루어진다. 다중 홉 통신을 이용하면 특정한 장애물 뒤에 위치한 드론과의 통신도 가능하게 되고 다중 드론을 활용한 분야에서의 활용성을 더욱 확장시킬 수 있다¹³⁾. 단일 드론 시스템에서 사용되는 “리턴 투 홈” 기술이 다중 드론 시스템에서 적용된 상황을 가정해보면 드론의 불필요한 이동을 예상할 수 있다. 드론이 컨트롤러와 연결이 끊어지게 되면 “리턴 투 홈” 동작에 의해 초기 이륙 지점까지 되돌아가야 하고 이는 과도한 배터리의 소모를 초래하고 임무의 효율 또한 저하시킨다.

일반적으로 드론은 매우 동적인 환경에서 작동하기 때문에 언제든지 통신 단절이 발생할 수 있다. 따라서 다중 드론 시스템으로의 운용 상황에서 드론과 컨트롤러 간의 단절을 고려하지 않는다면, 드론의 분실 및 추락과 같은 사고가 발생할 수 있다. 다중 드론 시스템이 FANET 구조를 지원할 수 있다면 다른 드론과의 지속적인 애드혹 통신을 통해 전체적인 연결성을 유지할 수 있으며, 다중 홉을 거쳐 컨트롤러와도 통신을 할 수 있다. 이러한 연결성은 다중 드론 시스템의 신뢰성을 향상시키고 급변하는 환경에서의 운용에 있

어서 예상하기 힘든 상황을 처리할 수 있게 한다¹⁴⁾.

따라서, 본 논문에서는 다중 드론 기반의 FANET을 위한 네트워크 자가 복구 기술을 제안한다. 이 기술은 드론 간의 동적인 애드혹 네트워크 환경에서 위치 정보를 사용하여 단계적 절차를 거쳐 네트워크에서 이탈된 드론이 자율적으로 네트워크 복구를 수행하는 기술이다. 본 논문은 다음과 같이 구성된다. 2장에서는 본 연구와 유사한 연구 및 단일 드론 시스템에서의 “리턴 투 홈”과 관련된 연구들을 고찰하고 3장에서는 제안하는 단계적 네트워크 자가 복구 기술을 제안한다. 4장에서는 NS-3를 사용한 시뮬레이션 환경과 제안 기술에 대한 성능 비교 결과를 기술 및 분석하고 마지막 5장에서 결론을 맺도록 한다.

II. 관련연구

2.1 애드혹 네트워크의 다중 드론 제어 기술

대다수의 상업용 드론은 하나의 컨트롤러와 하나의 드론으로 구성되며 이는 단일 드론 시스템으로 구분된다. 단일 드론 시스템에서 드론의 비행은 컨트롤러에 의한 제어를 기반으로 하며 다양한 활용을 위해서는 드론과 컨트롤러 사이의 신뢰성 있는 연결이 매우 중요하다. 하지만 실제 운용에서는 통신 장애 및 주변 환경의 급격한 변화로 인해 컨트롤러와 드론 간의 연결이 자주 끊기는 상황이 발생한다¹⁵⁾.

다른 제어 기술로는 주로 군사용 목적으로 사용되는 사전에 입력된 데이터에 따른 드론 운용이다. 사전에 입력 및 프로그래밍 된 제어 방법은 이전의 데이터를 기반으로 동작하는 자동 비행을 의미한다. 이러한 방식의 단점으로는 실시간 제어 및 실시간 데이터 전송이 불가능하거나 어려운 점이다. 일반적으로, 사전에 입력된 정보에 따라서 드론이 운용되기 때문에 예정된 임무를 완수할 때 까지는 제어가 불가능하다¹⁶⁾.

더 넓은 분야와 다양한 목적으로의 활용을 위해서는 다수의 드론들을 제어할 수 있어야 한다. 그러므로 드론 간 다중 홉 통신 기술을 적용하게 되면 지상제어 센터와의 직접적인 연결 없이 다중 홉을 통해서 간접적으로 연결을 이룰 수 있고 이를 통해 다수의 드론들이 제어될 수 있다. 뿐만 아니라 임무 수행을 위한 드론의 운용 범위 또한 확대된다. 본 연구를 진행하기에 앞서, FANET을 기반으로 한 다중 드론 실시간 제어 기술에 대해서 제안했으며¹⁷⁾, 이는 지상제어 센터의 통신 가능 범위를 벗어난 드론을 제어하기 위한 기술이다.

그럼 1에서 지상제어 센터와 직접적으로 연결된 하

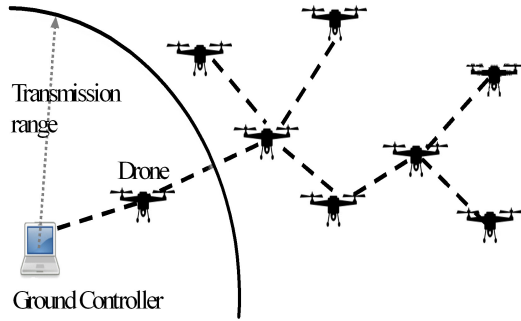


그림 1. 단일 컨트롤러와 다중 드론 제어 시스템
Fig. 1. A single-controller and multi-drone control system

나의 드론을 제외한 나머지 드론들은 지상제어 센터의 전송 범위 밖에 위치하고 있다. 전송 범위 밖에 위치한 드론들은 지상제어 센터와 직접적으로 연결되지 않았기 때문에 애드혹 네트워크를 사용해 서로 연결을 맺고 다중 홉을 거쳐 지상제어 센터와 통신이 가능하게 된다. 이 때, 다중 홉을 거치지 않고 각 드론들이 지상제어 센터와 직접 연결을 맺는다면 드론 운용 범위가 크게 줄어드는 문제점이 발생한다. 각각의 드론들은 GPS(Global Positioning System)에서 수집된 위치정보를 지상제어 센터로 전달한다. 지상제어 센터는 제어하고자 하는 드론에 대한 제어 메시지를 전송하고, 다른 드론들은 단순히 릴레이 역할을 수행한다. 이 제어 메시지에는 각 드론의 IP주소가 포함되어 있고 이를 바탕으로 자신에 대한 메시지인지 판별한다. 제어 메시지의 목적지 주소가 자신의 IP 주소와 일치하지 않으면, 메시지를 자신의 1 홉 이웃 드론에게 전달한다. 특정 드론이 자신에 대한 제어 메시지를 받은 경우에는 전달 받은 제어 명령에 따라 동작을 수행한다. 따라서 애드혹 네트워크 구성을 통하여 단일 컨트롤러만으로 다수의 드론을 제어 할 수 있다.

2.2 단일 드론 시스템에서의 복구 기술

앞서 언급한 대다수의 상업용 드론은 하나의 드론과 컨트롤러로 구성되는 단일 드론 시스템이며 드론 운용 중 사용자의 시야에서 드론이 사라진 경우, “리턴 투 홈” 기술을 통해 드론이 비행을 시작한 지점으로 복귀시킨다. 이러한 복귀는 GPS 위치 정보를 이용해 이루어지며 어느 정도의 오차를 가질 수 있다. 따라서 GPS 위치 정보에 셀 분할 알고리즘과 확장 칼만필터를 사용하여 드론의 위치 정보를 이용한 위치 추정 정밀도를 향상 시키는 연구가 이루어졌다⁸⁾.

또한, 단일 드론이 조종 범위를 벗어나게 되면 스스로 이를 인지하고 조종 가능 범위로 다시 돌아오도록

하는 자동 회기 드론 연구가 진행되었다. 자동 회기 드론은 수행된 이동 명령어의 종류, 명령어 수행 시점의 배터리 잔량을 기억하여 조종 범위를 벗어났을 때 이를 역으로 수행하는 방식이다. 이 방안은 GPS 없이 데이터베이스를 이용한 백트래킹 알고리즘에 초점을 맞추었으며, 퍼지 제어를 통한 보상값으로 신호를 증폭시켜서 오차를 줄이는 방법을 사용하였다⁹⁾.

기존의 연구와는 다르게, 본 논문에서는 FANET 환경에서 실제의 드론 운용을 고려한 네트워크 자가 복구 기술을 제안한다.

III. 제안방안

이 절에서는 이웃 테이블을 생성하는 과정과 특정 드론이 네트워크를 이탈했을 때, 이를 인지하고 네트워크 자가 복구 기술이 동작하는 과정에 대해서 소개한다. 제안한 기술에서는 모든 드론이 GPS를 가지고 있으며, 지상제어 센터의 위치를 알고 있다고 가정한다.

3.1 Hello message를 통한 이웃 테이블 생성

각 드론은 자신의 IP주소와 GPS 위치 정보가 포함된 hello message를 주기적으로 broadcast한다. 모든 드론이 주기적으로 전송하는 hello message는 이웃 테이블을 생성하고 유지 및 관리를 목적으로 사용된다. 주기적인 hello message의 교환으로 각 드론은 자신의 이웃 테이블을 갱신한다. 이웃 테이블이 가진 항목으로는 이웃 드론의 IP 주소, 이웃 드론으로부터 받은 위치 정보, 그리고 그 정보의 유지 시간을 나타내는 TTL(Time-to-live) 이 있다. 이웃 드론의 IP 주소 항목은 새로운 이웃인지의 여부를 판별하고 이웃 테이블을 관리하기 위해 사용된다. GPS 데이터는 위도, 경도, 고도 값으로 표현되며 각 드론의 위치 정보를 나타낸다. 마지막으로 TTL 값은 hello message의 전송 주기인 hello interval의 3배로 설정되며, 이웃 테이블의 엔트리 유지 관리를 위해 사용된다. 아래의 표 1은 이웃 테이블의 예를 나타낸다.

특정 드론이 이웃 드론으로부터 hello message를 전달받았을 때, 자신의 이웃 테이블에서 hello message를 보낸 드론의 IP 주소가 존재하는지를 가장 먼저 확인한다. 이전에 hello message를 보낸 적이 있는 이웃이라면 자신의 이웃 테이블 내에 정보가 존재할 것이고 그렇지 않다면 이웃 테이블에 해당 정보가 존재하지 않을 것이다. 새로운 이웃인 경우에는 전달 받은 hello message로부터 드론의 IP 주소와 위치 정

표 1. 이웃 테이블의 예
Table 1. Example of neighbor table

Drone id	Location (lat, lon, alt)	TTL(sec)
-	35.8832°,128.6129°,14.3m	-
192.168.3.3	35.8805°,128.6129°,14.8m	2.34
192.168.3.5	35.8878°,128.6132°,15.4m	2.51
192.168.3.7	35.8874°,128.6136°,15.2m	2.02

보를 얻어서 이웃 테이블의 새로운 엔트리에 저장한다. 새로운 이웃 정보의 저장과 함께 해당 엔트리의 TTL 값을 hello 전송 주기의 3배 값으로 설정한다. 만약 이웃 테이블에 이미 존재하는 이웃으로부터 hello message를 전달받은 경우에는 해당 이웃에 대한 기존의 위치 정보를 새롭게 전달받은 hello message를 토대로 갱신한다. 마찬가지로, 해당 엔트리의 TTL 값도 함께 갱신한다. 이는 기존에 hello message를 주고받던 이웃 드론과 계속해서 연결되어 있음을 나타낸다.

3.2 TTL 값에 따른 링크 단절 감지

드론은 자신에게 주어진 임무를 수행하기 위해 계속 이동하게 된다. 일반적으로 드론은 주위가 급변하고 장애물이 존재하는 환경에서 운용된다. 실제 운용에서는 장애물이나 날씨와 같은 주변 영향으로 인해 드론 간 연결이 약해지거나 단절되는 상황이 발생할 수 있다. 뿐만 아니라, 통신 범위를 고려하지 않은 조종으로 인한 드론의 네트워크 이탈 또한 발생할 수 있다. 이처럼 특정 드론이 이웃 드론과의 전송 범위를 벗어나서 서로 간에 hello message를 주고받을 수 없게 되면 네트워크에서 이탈된 상태로 간주한다. 이 경우 네트워크에서 이탈된 드론의 이웃 테이블에 저장되어 있던 특정 이웃 드론의 TTL 값은 머지않아 만료된다. TTL 값이 hello message 전송주기의 3배로 설정되었기 때문에, 3번의 hello message를 전달받지 못하면 해당 드론에 대한 TTL 값이 만료되게 된다. 이웃 테이블의 특정 엔트리에 대한 TTL 값이 만료되면, 이웃 테이블 내에 다른 이웃이 존재하는지를 확인한다. 만약 이웃 테이블 내에 다른 이웃이 아직 존재한다면, TTL 값이 만료된 이웃 드론에 대한 엔트리를 삭제한다. 이는 특정 드론과의 연결은 끊어졌지만 다른 이웃과의 연결이 유지된 상태이므로 네트워크에서 이탈되지 않은 상태이다. 하지만 이웃 테이블 내에 다른 이웃이 존재하지 않는다면, 이는 자신이 네트워크로부터 이탈된 상태이므로 네트워크를 복구하기 위해 본 논문에서 제안한 네트워크 자가 복구 모드를 시작한다. 그림 2는 기본 동작 및 링크 단절 감지 과정에

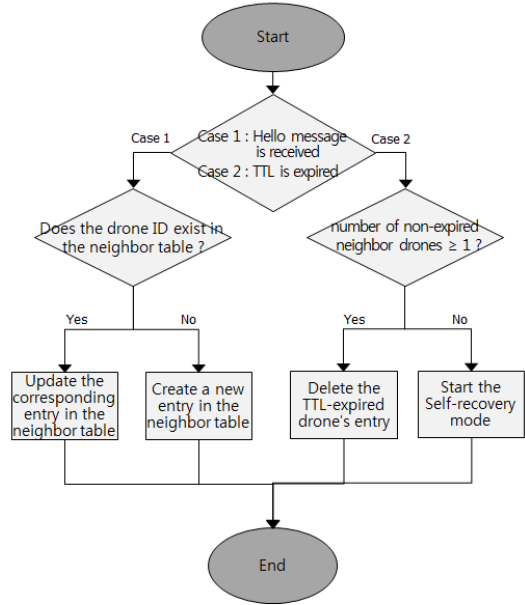


그림 2. 제안 기술의 전체 동작 흐름도
Fig. 2. Flow chart of the proposed scheme

대한 흐름도를 나타낸다.

3.3 네트워크 자가 복구 절차

드론이 주어진 임무 수행을 위해 계속해서 이동을 하다보면, 네트워크의 말단에 위치하게 되는 상황이 발생한다. 이 때 지상제어 센터 쪽으로의 네트워크를 유지해주는 이웃 드론과 반대의 방향으로 이동을 수행하면, 곧이어 네트워크 이탈이 발생하게 된다. 이외에도 주변 장애물에 의해 일시적으로 통신신호가 약화되는 상황이 발생하면 특정 드론들은 네트워크 이탈 상태가 될 수 있다. 자신의 이웃 테이블에 오직 하나의 이웃 드론에 대한 정보만 남아있고, 그 하나만 남아있던 드론에 대한 TTL 값이 만료되면 네트워크에서 이탈된 상태로 간주된다. 이 경우 해당 드론은 다시 지상제어 센터와의 연결을 위한 네트워크 자가 복구 절차를 시작한다. 이 절차는 총 세 단계로 구성되며 제안하는 네트워크 자가 복구 절차에 대한 동작 흐름도는 그림 3과 같다.

본 논문에서의 제안 기술은 드론이 지상제어 센터로부터 전달된 이동 명령에 따라서 해당 위치로 이동하며 임무를 수행하는 상황을 고려한다.

1) 복구 1 단계 : 네트워크 자가 복구 절차의 첫 번째 단계는 이웃 테이블의 최상단에 저장된 자신의 이전 위치로의 이동이다. 드론이 현재 위치에서 지상제

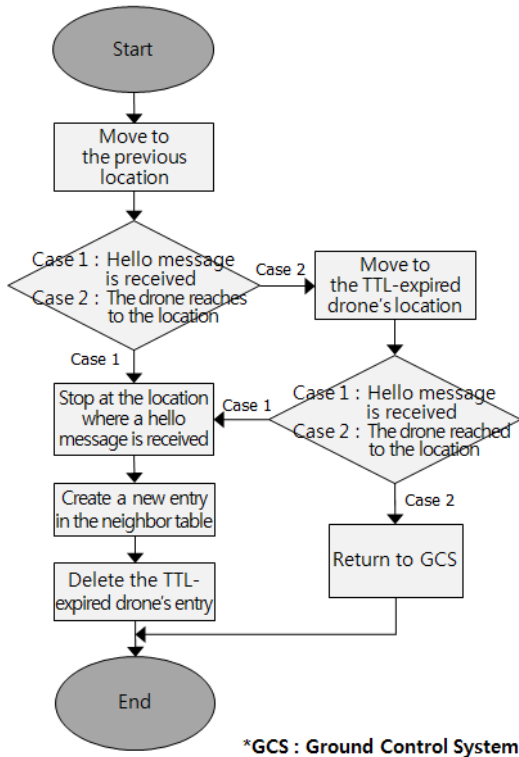


그림 3. 제안한 네트워크 자가 복구 동작 흐름도
Fig. 3. Flow chart of the network self-recovery mode

어 센터로부터의 명령에 따라서 이동을 시작하게 되면 자신의 현재 GPS 위치정보를 이웃 테이블의 첫 번째 엔트리에 저장한다. 이 위치정보는 드론이 다른 위치로 이동할 때마다 갱신되며 임무를 위해 계속해서 이동을 하다가 네트워크로부터의 이탈이 발생하게 되면 네트워크 자가 복구 1단계를 시작한다. 이웃 테이블의 첫 번째 엔트리에 저장해두었던 자신의 이전 위치로 이동을 수행하고, 이동 중에 다른 드론으로부터 hello message를 전달 받으면 네트워크에 복구된 상태로 복구 과정을 중지한다. 하지만 자신의 이전 위치에 도달할 때 까지 다른 드론과 만나지 못하는 경우에는 네트워크 자가 복구 1단계는 실패한 것으로 간주하고 복구 2단계를 시작한다.

2) 복구 2 단계 : 앞서 3.2에서는 이웃 테이블 내에 마지막으로 완료된 이웃 외에 다른 이웃이 존재하지 않는다면 네트워크 자가 복구 모드를 수행한다고 설명하였다. 네트워크 자가 복구 2단계는 이웃 테이블에 마지막으로 남아있었던 이웃 위치 정보를 이용한 복구이다. TTL 값이 만료된 이웃의 위치로 이동을 수행하는 동안에 다른 드론을 만나게 되면 복구 동작을 멈

춘다. 이처럼 복구가 성공하는 경우에는 복구 1, 2단계 모두 복구 도중에 만난 새로운 드론의 위치 정보를 이웃 테이블 내에 저장하고, 기존에 남아있던 TTL값이 만료된 이웃 드론의 엔트리는 제거한다. 하지만 이웃이 있었던 위치에 도달할 때 까지 다른 드론과 만나지 못하는 경우에는 2단계 또한 실패한 것으로 간주하고 마지막 단계인 3단계를 시작한다.

3) 복구 3 단계 : 복구 2단계가 실패하면 마지막 단계인 3단계를 수행한다. 복구 2단계에서는 드론이 자신의 이웃 드론이 있던 위치로 이동했지만 해당 이웃 드론은 임무를 위해 다른 방향으로 이동한 경우에는 2단계 과정 또한 실패할 수 있다. 2단계까지의 복구에 실패한 드론은 초기에 드론이 비행을 시작한 지상제어 센터가 있는 위치로 이동을 수행한다. 이 과정에서 다른 드론을 만나게 되면 복구를 멈추고, 그 위치에서 다시 계획된 임무를 다시 수행한다.

IV. 성능분석

제안하는 이웃 위치정보 기반 단계적 네트워크 자가 복구 기술을 평가하기 위한 시뮬레이션 툴로서 NS-3(Network Simulator 3)^[10]를 사용하였고, 패킷 전달률과 평균 네트워크 단절 시간을 성능 평가 지표로 활용하였다. 또한, 이웃 테이블의 엔트리 관리를 위해 사용되는 TTL값의 변화에 대한 정찰 범위의 영향을 확인하기 위해 정찰된 셀의 개수를 지표로 활용하였다. 제안한 자율적 네트워크 복구 기술과 네트워크 복구 절차를 수행하지 않는 네트워크 비 복구 기술과의 성능 비교 시뮬레이션을 수행했다.

4.1 시뮬레이션 환경

시뮬레이션 시나리오에서는 네 대의 드론이 대각으로 배치되어 서로의 전송 범위 내에 고정되어 있으며 한 대의 드론만 시뮬레이션 영역 내에서 정의된 이동성 모델에 따라 이동한다. 시뮬레이션 영역은 5,000m × 5,000m이고 모든 드론 및 지상제어 센터의 통신거리는 1,000m로 설정되었다.

본 시뮬레이션에서의 각 드론은 Random Waypoint Mobility Model^[11]에 따라 움직이게 되는데, 드론은 다음으로 이동할 웨이포인트를 무작위로 선택하고 설정된 일정한 속력으로 이동하게 된다.

아래의 그림 4는 이동성 모델에 따른 드론의 이동을 나타낸다. 전체 시뮬레이션 영역은 500m × 500m 크기의 셀로 나누어지고 각 셀의 중심은 드론이 이동하는 웨이포인트로 설정되었다. 따라서 본 시뮬레이션

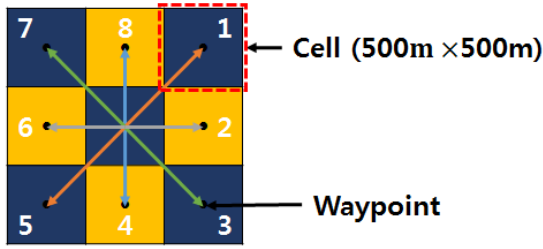


그림 4. 이동성 모델에 따른 드론의 이동
Fig. 4. Movement of drone according to the mobility model

에서는 총 100개의 셀이 존재하며 드론이 특정 셀의 웨이포인트에 위치한 경우에는 해당 셀의 전체영역을 정찰한 것으로 가정한다. 그림 4에서, 중심 셀에 위치한 드론은 둘러싼 8개의 셀 중에서 임의로 하나의 웨이포인트를 다음 목적지로 선정하여 이동한다.

네트워크 비 복구 기술의 경우에는 시뮬레이션 동안 계속해서 무작위로의 이동을 수행하는 반면, 제안한 네트워크 복구 기술의 경우에는 네트워크 이탈이 발생하지 않은 상태일 때에만 무작위로의 이동을 한다. 네트워크 이탈이 발생하면, 제안한 네트워크 자가 복구 기술에 따라 복구 절차를 수행한다. 시뮬레이션에서 이동성 드론의 속력은 5, 15, 25, 35, 45, 55 m/s로 바꾸어가며 비교하였다. 지상제어 센터와 나머지 드론은 (0, 0), (500, 500), (1,000, 1,000), (1,500,

표 2. 시뮬레이션 파라미터
Table 2. Simulation parameters

파라미터	값
Map size	5,000m × 5,000m
Cell size	500m × 500m
Simulation time	1,800 sec
Routing protocol	OLSR, AODV
Number of drones	5
Drone speed	5, 15, 25, 35, 45, 55 m/s
Transmission range	1,000 m
Loss model	Friis propagation loss model
Packet size	1,000 bytes
Transmission rate	1 Mbps
Mobility model	RandomWaypointMobilityModel
Data packet interval	1 sec

1,500) 지점에 대각선 형태로 고정되었다. (0, 0)에 위치한 지상제어 센터는 매 초 1,000 bytes의 패킷을 전송하며 이 패킷은 고정된 드론들을 거쳐 이동성 드론에게 전달된다. 라우팅 프로토콜로는 OLSR^[12] 과 AODV^[13]를 고려했다. 동일한 상황에서 제안한 네트워크 복구 동작을 수행하는 경우와 그렇지 않은 경우에 대한 패킷 전달률과 평균 네트워크 단절 시간을 비교하였다. 시뮬레이션에 적용된 파라미터는 표 2와 같다.

4.2 성능 메트릭

시뮬레이션을 통한 제안 기술의 성능을 비교하기 위해 아래의 네 가지 메트릭을 사용하였다.

4.2.1 패킷 전달률

패킷 전달률은 지상제어 센터에서 전송된 패킷 중 목표지인 이동 드론에게 성공적으로 전달된 패킷의 수의 비로 정의된다.

4.2.2 평균 네트워크 단절 시간

평균 네트워크 단절 시간은 어떤 드론이 이동으로 인해 네트워크에서 이탈한 시간에 대한 평균값으로 정의된다.

4.2.3 정찰된 셀의 개수

정찰된 셀의 개수는 전체 시뮬레이션 영역을 이루는 셀 중에 드론이 한 번이라도 이동한 셀들의 합으로 정의된다. 하나의 셀은 500m × 500m의 크기를 가지며 셀의 중심인 웨이포인트까지 도달한 경우에 해당 셀을 정찰한 것으로 간주한다. 이 메트릭은 4.4절의 시뮬레이션에서 사용된다.

4.2.4 지연 시간

지연 시간은 지상제어 센터에서 전송된 패킷이 목적지인 이동성 드론에게 전달되기까지 걸린 시간으로 정의된다.

4.3 시뮬레이션 결과

그림 5는 OLSR 라우팅 프로토콜로 동작하는 드론의 속도 변화에 따른 패킷 전달률을 나타낸다.

드론의 속도가 5m/s일 때는 비 복구 기술의 패킷 전달률이 약 70%로 나타났고 각 드론들의 통신범위에 비해 드론 자체의 속도가 매우 낮았기 때문에 상대적으로 높은 패킷 전달률을 보였다. 드론의 속도가 증가함에 따라서 패킷 전달률은 감소하는 모습을 확인할 수 있다. 속도가 15m/s가 되었을 때 약 50%의 패

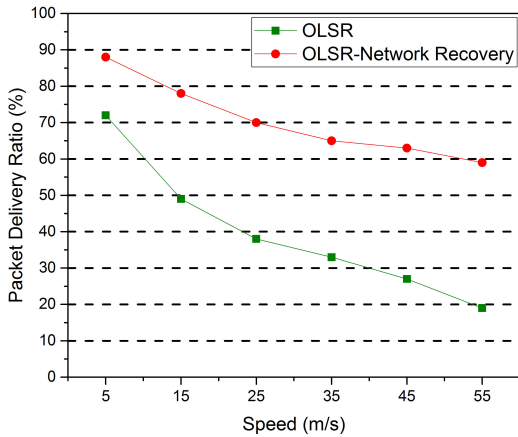


그림 5. 드론 속도에 따른 패킷 전달률 비교(OLSR)
Fig. 5. Packet delivery ratio according to drone speed with OLSR

킷 전달률을 보였고, 최대 속도인 55m/s일 때는 불과 20%의 패킷 전달률을 보였다.

그러나 제한한 네트워크 복구 기술의 경우, 드론의 속도가 5m/s일 때의 패킷 전달률은 90%로 나타났다. 뿐만 아니라, 드론의 속도가 증가함에 따라 패킷 전달률은 감소했지만 그 감소폭은 네트워크 비 복구 기술에 비해 현저히 낮았다. 속도가 15m/s일 때 약 80%의 패킷 전달률, 최대 속도인 55m/s일 때 60% 정도의 패킷 전달률을 보였다. 따라서 네트워크 복구 기술은 네트워크 비 복구 기술보다 저속에서는 약 20%, 고속에서는 약 40% 더 높은 패킷 전달률을 가졌다.

그림 6은 드론의 속도 변화에 따른 평균 네트워크 단절 시간을 나타낸다. 드론의 속도와 상관없이, 네트

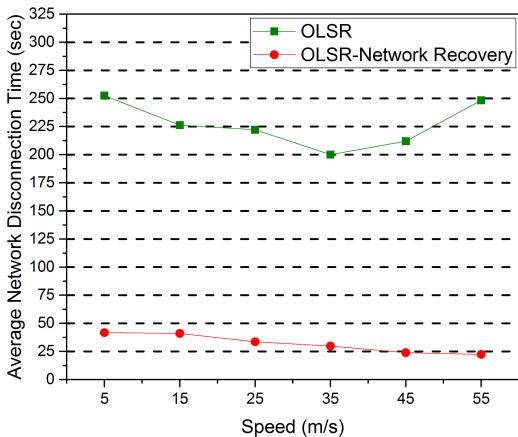


그림 6. 드론 속도에 따른 평균 네트워크 단절 시간 비교(OLSR)
Fig. 6. Average network disconnection time according to drone speed with OLSR

워크 비 복구 기술은 네트워크 복구 기술보다 훨씬 긴 네트워크 단절 시간을 보였다. 이는 네트워크 비 복구 기술의 경우, 드론이 무작위로 이동하기 때문에 드론의 속도와는 무관하게 긴 시간동안 네트워크와 단절되는 상황이 자주 발생하여 전체적으로 긴 시간의 네트워크 단절 시간을 나타냈다. 다만, 비 복구 상태로 동작하는 상황에서 관찰할 수 있는 사실은 드론의 속도가 최저일 때는 드론의 느린 속도로 인해 네트워크 단절 시간의 편차가 컸으며 최대 단절 시간이 높은 쪽으로 치중되어 있었다. 반면, 드론이 최대 속도일 때는 네트워크 단절 시간의 편차는 작으나 그 횟수가 더 많았기 때문에 높은 평균 네트워크 단절 시간을 확인할 수 있었다.

이어서 네트워크 복구 기술의 경우에는 드론의 속도에 따라 약간의 차이를 보였다. 드론의 속도가 느릴 때는 약 40초의 평균 네트워크 단절 시간을 보였으나 최고 속도인 55m/s가 되었을 때, 25초 정도의 네트워크 평균 단절 시간을 보였다. 드론의 속도가 느릴 때는 네트워크 단절을 인지하고 복구를 위한 이동을 수행하는 것이 속도가 느리기 때문에 긴 시간이 걸리는 반면, 속도가 빠를 때는 복구를 위한 이동에 걸리는 시간이 줄어들기 때문에 단절 시간이 짧게 나타났다. 따라서 속도가 증가함에 따라 평균 네트워크 단절 시간은 조금씩 감소하는 것을 확인했다.

그림 7은 AODV 라우팅 프로토콜로 동작하는 드론의 속도 변화에 따른 패킷 전달률을 나타낸다. 드론의 속도가 5m/s일 때는 비 복구 기술의 패킷 전달률이 약 75%로 나타났고 OLSR 라우팅 프로토콜로 동작할 때와 마찬가지로 각 드론들의 통신범위에 비해

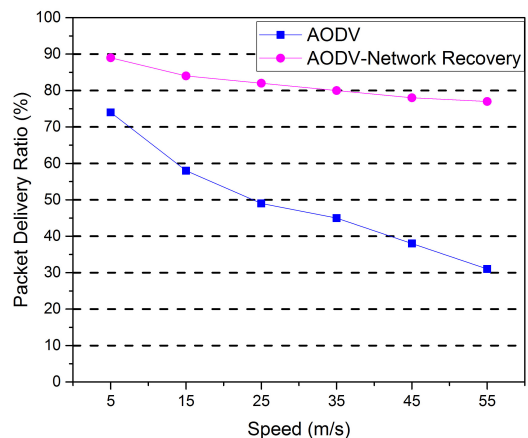


그림 7. 드론 속도에 따른 패킷 전달률 비교(AODV)
Fig. 7. Packet delivery ratio according to drone speed with AODV

드론 자체의 속도가 매우 낮았기 때문에 상대적으로 높은 패킷 전달률을 보였다. 속도가 15m/s가 되었을 때 약 60%의 패킷 전달률을 보였고, 최대 속도인 55m/s일 때는 30%의 패킷 전달률을 보였다.

AODV 라우팅 프로토콜이 적용된 네트워크 복구 기술의 경우, 드론의 속도가 5m/s일 때의 패킷 전달률은 대략 90%로 나타났다. 드론의 속도가 증가함에 따라 패킷 전달률은 감소했지만 그 감소폭은 네트워크 비 복구 기술에 비해 현저히 낮았다. 속도가 35m/s일 때 약 80%의 패킷 전달률, 최대 속도인 55m/s일 때 77%정도의 높은 패킷 전달률을 보였다. 따라서 AODV 라우팅 프로토콜이 적용된 네트워크 복구 기술은 네트워크 비 복구 기술보다 저속에서는 약 15%, 고속에서는 약 47% 더 높은 패킷 전달률을 가졌다.

그림 8은 AODV 라우팅 프로토콜로 동작하는 드론의 속도 변화에 따른 평균 네트워크 단절 시간을 나타낸다. 네트워크 비 복구 기술은 네트워크 복구 기술보다 훨씬 긴 네트워크 단절 시간을 보였다. 네트워크 비 복구 기술은 드론이 무작위로 이동하기 때문에 드론의 속도와는 무관하게 긴 시간동안 네트워크와 단절되는 상황이 발생했다. OLSR 라우팅 프로토콜이 적용된 경우와 마찬가지로, 드론의 속도가 최저일 때 느린 속도로 인해 네트워크 단절 시간의 편차가 컸으며 단절 시간의 최댓값이 높은 쪽으로 치중되어 있었다. 반면 드론이 최대 속도일 때는 네트워크 단절 시간의 편차는 속도가 느릴 때에 비해서는 작으나 그 횟수가 더 많았기 때문에 높은 평균 네트워크 단절 시간을 보였다.

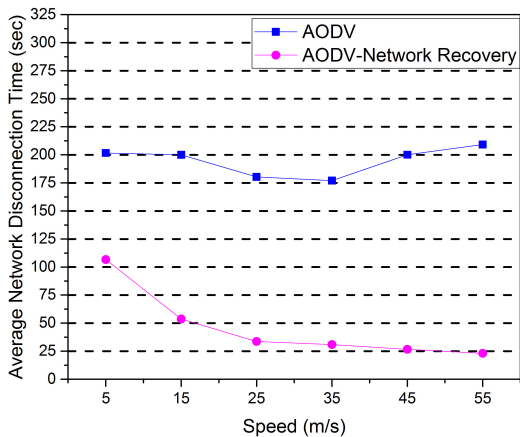


그림 8 드론 속도에 따른 평균 네트워크 단절 시간 비교 (AODV)
Fig. 8. Average network disconnection time according to drone speed with AODV

반면에 AODV 라우팅 프로토콜로 동작하는 네트워크 복구 기술은 드론의 속도에 따라 차이를 보였다. 드론의 속도가 낮을 때는 약 100초의 평균 네트워크 단절 시간을 가졌으나 최고 속도인 55m/s가 되었을 때, 25초 정도의 평균 단절 시간을 가졌다. 드론의 속도가 낮을 때는 네트워크 단절을 인지하고 복구를 위한 이동을 수행하는 것이 오래 걸리는 반면, 속도가 빠를 때는 복구를 위한 이동에 걸리는 시간이 줄어드는 것을 확인할 수 있었다. 따라서 속도가 증가함에 따라 평균 네트워크 단절 시간이 감소하는 것을 확인했다.

각 라우팅 프로토콜에 대한 패킷 전달률에는 약간의 차이가 있었다. 드론이 저속으로 동작하는 경우에는 거의 비슷한 값을 가졌다. 하지만 드론의 속도가 증가함에 따라 AODV 라우팅 프로토콜은 OLSR 라우팅 프로토콜보다 약 19% 더 높은 패킷 전달률을 보였다. 드론의 속도가 저속에서 고속으로 증가함에 따라 점차 패킷 전달률도 차이가 커지는 것을 확인할 수 있었다. 결국, 저속 환경에서는 AODV와 OLSR이 유사한 성능을 보였지만 고속 환경에서는 AODV가 더 높은 패킷 전달률을 가지는 것을 확인할 수 있었다.

4.4 TTL 값에 따른 정찰 범위의 변화

이웃 테이블의 엔트리 관리를 위해 사용되는 TTL 값의 변화에 따른 정찰 범위의 영향을 확인하기 위한 시뮬레이션을 수행했다. 시뮬레이션을 위한 대부분의 파라미터는 표 2와 동일하지만, 라우팅 프로토콜은 AODV, 드론의 속력은 55m/s로 고정되었고 TTL 값은 기존 값에서 2.5, 5, 7.5, 10배로 증가시키며 비교하였다.

그림 9는 TTL 값이 7.5초일 때의 드론의 이동 경로를 나타낸다. 그림 9에서의 드론 이동 경로는 (1,500, 1,500) 지점에 위치한 드론을 중심으로 왼쪽 위로 밀집되어 있음을 확인할 수 있다. TTL 값이 최소일 때, 드론은 먼 거리를 이동하지 못한 채 네트워크 자가 복구를 시작하기 때문에 비교적 좁은 범위의 지역에서만 이동을 한다.

그림 10은 기존의 TTL 값인 7.5초의 10배인 75초일 때의 드론의 이동 경로를 나타낸다. 그림 9에서는 (3,000, 3,250) 영역 내에서만 이동이 이루어 졌지만, 그림 10에서는 (4,500, 5,000)까지의 범위 내에서 이동한 것을 알 수 있다. TTL 값이 증가했기 때문에 드론은 이전보다 더 먼 거리를 이동하게 되고 네트워크 로부터의 이탈 이후 뒤늦게 복구 동작을 시작한다. 네

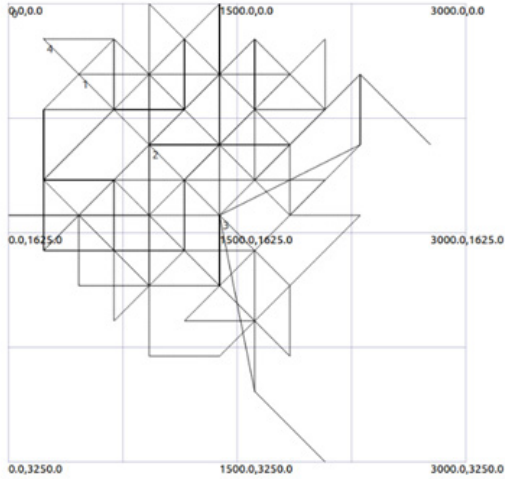


그림 9. TTL = 7.5초일 때 드론의 이동 경로
Fig. 9. The trajectory of the drone when TTL = 7.5 sec

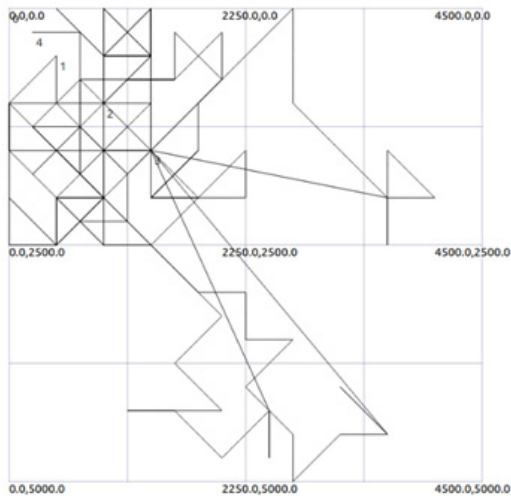


그림 10. TTL = 75초일 때 드론의 이동 경로
Fig. 10. The trajectory of the drone when TTL = 75 sec

트위크 이탈을 뒤늦게 감지하기 때문에 네트워크 단절 시간은 증가하게 되지만 더 넓은 영역에서의 임무 수행이 가능하다.

그림 11은 TTL 값에 따른 패킷 전달률과 드론에 의해 정찰된 셀의 개수를 나타낸다. 전체 100개의 셀 중에서 TTL이 7.5초일 때는 약 47개의 셀을 정찰했으며 75초일 때는 약 65개의 셀을 정찰하였다. 이로부터 이웃 테이블의 엔트리를 관리하는 TTL 값이 증가함에 따라 드론이 정찰한 셀의 수는 증가하는 것을 알 수 있다. 하지만 TTL 값이 증가함에 따라 네트워크로

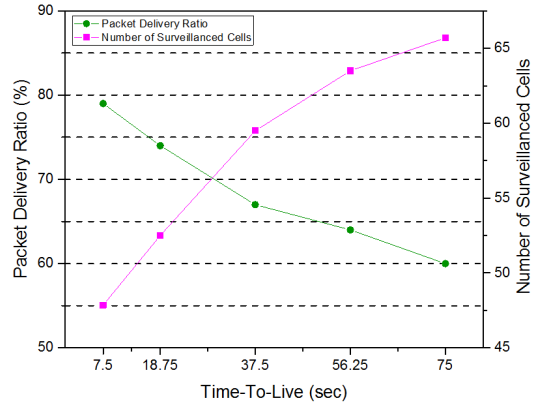


그림 11. TTL 값에 따른 패킷 전달률과 정찰된 셀의 개수
Fig. 11. Packet delivery ratio and number of surveilled cells according to TTL

부터의 단절된 시간이 증가하므로 패킷 전달률은 점차 감소하게 된다. TTL 값이 7.5초인 경우에는 약 80%의 패킷 전달률을 보였지만 75초일 때는 20% 감소된 60%의 패킷 전달률을 보였다. 따라서 TTL 값의 증가에 따라서 패킷 전달률은 감소하지만, 더 넓은 범위의 정찰이 가능해진다. 또한, 두 그래프가 교차하는 지점으로부터 최적의 TTL 값을 얻을 수 있다.

4.5 라우팅 프로토콜에 따른 지연시간

드론 애드혹 네트워크에서는 지상제어 센터로부터의 제어 정보의 신속한 전달이 무엇보다 중요하기 때문에 각 라우팅 프로토콜에 따른 지상제어 센터로부터 이동 드론까지의 지연시간에 대한 성능 비교를 수행했다.

그림 12는 이동성 드론의 속력 변화에 따른 AODV와 OLSR의 지연시간을 나타낸다. 정적 라우팅 프로토콜인 OLSR의 경우에는 동적 라우팅 프로토콜인 AODV보다 낮은 지연시간을 보였다. 두 가지의 라우팅 프로토콜 모두, 드론의 속력이 증가함에 따라 지연시간이 점차 감소했지만 약간의 변동을 나타냈다. AODV의 경우에는 드론의 속력이 15m/s에서 25m/s로 증가할 때 지연시간의 큰 변화폭을 보였으며 25m/s 이후에는 근사한 값의 지연시간을 보였다. OLSR의 경우, AODV와는 다르게 저속과 고속상황에서의 지연시간 변동 폭이 크지 않았으며 드론의 속력이 증가함에 따라 지연시간의 표준편차가 점차 줄어들었다. 하지만 AODV는 속력의 변화와는 무관하게 일정하지 않은 지연시간의 표준편차를 보였다. 결론적으로 지연시간 측면에서는 정적 라우팅 프로토콜인 OLSR이 더 나은 성능을 보였다.

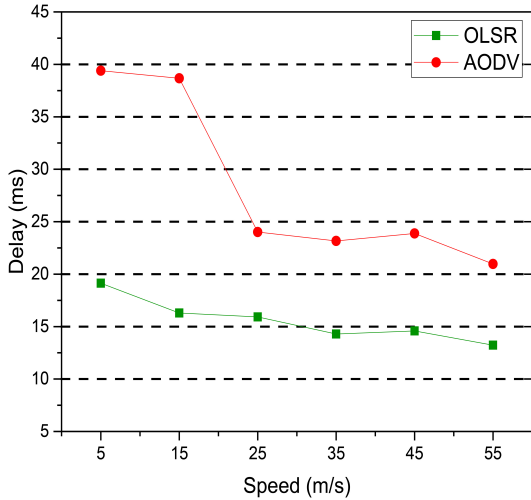


그림 12. 라우팅 프로토콜에 따른 지연시간
Fig. 12. Delay according to routing protocols

V. 결 론

본 연구에서는 다중 드론 시스템 환경에서 드론의 네트워크 이탈이 발생하는 경우, 이웃 위치를 기반으로 한 단계적 자가 복구 기술에 대해 제안하였다.

시뮬레이션 결과는 다중 드론 네트워크 환경에서 제안한 기술이 네트워크 비 복구 기술보다 낮은 평균 네트워크 단절 시간과 높은 패킷 전달률을 보였다. 제안한 기술은 드론의 속도가 증가함에 따라 패킷 전달률이 조금 감소했지만, 네트워크 비 복구 기술보다는 훨씬 높은 성능을 보였다. 또한, 평균 네트워크 단절 시간 측면에서는 약 200초정도 더 낮은 단절 시간을 보였다. AODV와 OLSR 라우팅 프로토콜의 적용에 따른 비교로는 AODV 라우팅 프로토콜의 경우에 고속 환경에서 더 높은 패킷 전달률을 가졌고 저속 환경에서는 높은 지연시간을 보였다. OLSR 라우팅 프로토콜의 경우는 저속 환경에서 더 낮은 평균 네트워크 단절 시간을 보였으며 전반적으로 AODV 라우팅 프로토콜에 비해 낮은 지연시간을 보였다. 또한, TTL 값이 정찰된 셀의 개수와 패킷 전달률에 미치는 영향을 살펴보았다. TTL 값이 증가함에 따라 패킷 전달률은 감소했지만 정찰된 셀의 개수는 증가하는 것을 확인했다.

차후 연구에서는 본 연구에서 제안한 기술을 향상시키기 위한 hello message의 전송 세기 조절 방안에 대해서 연구하고 애드혹 네트워크망과 지상제어 센터와의 연결성을 보장할 수 있는 개선된 알고리즘에 대한 성능평가를 수행할 예정이다.

References

- [1] M. H. Tareque, M. S. Hossain, and M. Atiquzzaman, "On the Routing in Flying Ad hoc Networks," in *Proc. FedCSIS*, 2015.
- [2] I. Bekmezci, O. K. Sahingoz, and S. Temel, "Flying ad-hoc networks (FANETs): A survey," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1254-1270, 2013.
- [3] W. Zafar and B. M. Khan, "Flying ad-hoc networks: Technological and social implications," *IEEE Technol. Soc. Mag.*, vol. 35, no. 2, pp. 67-74, Jun. 2016.
- [4] E. W. Frew and T. X. Brown, "Networking issues for small unmanned aircraft systems," *J. Intell. Robot. Syst.*, vol. 54, no. 1, pp. 21-37, 2008.
- [5] O. K. Sahingoz, "Networking models in flying ad-hoc networks(FANETs): Concepts and challenges," *J. Intell. Robot. Syst.*, vol. 74, no. 1, pp. 512-527, 2014.
- [6] S. Wilerson, C. Korpela, and K. Chang, "Aerial swarms as asymmetric threats," in *Proc. ICUAS*, pp. 381-386, Jun. 2016.
- [7] G. H. Kim, J. C. Nam, I. Mahmud, and Y. Z. Cho, "Multi-drone control and network self-recovery for flying Ad Hoc Networks," in *Proc. ICUFN*, pp. 148-150, 2016.
- [8] K.-J. Kim, Y.-K. Kim, S.-H. Choi, and J.-M. Lee, "Outdoor localization for returning of quad-rotor using cell divide algorithm and extended kalman filter," *J. IKEEE*, vol. 17, no. 4, pp. 440-445, 2013.
- [9] K. M. Sun, S. J. Jang, D. S. Kwon, and J. S. Ahn, "Self-Traceable Drone," in *Proc. KICS Fall Conf.*, pp. 463-465, 2015.
- [10] NS-3 (Network Simulator - 3), <https://www.nsnam.org>.
- [11] K. Singh and A. H. Verna, "Applying OLSR routing in FANETs," in *Proc. ICACCCT*, pp. 1212-1215, May 2014.
- [12] T. Clausen and P. Jacquet, *Optimized Link State Routing Protocol (OLSR)*, RFC 3626, 2003.
- [13] C. E. Perkins and E. M. Royer, *Ad Hoc On*

Demand Distance Vector (AODV) Routing,
Draft-ietf-manet-aodv-02.txt, 1998.

김 건 환 (Geon-Hwan Kim)



2015년 2월 : 경북대학교 전자
공학부 졸업
2017년 2월 : 경북대학교 전자
공학부 석사
2017년 3월~현재 : 경북대학교
전자공학부 박사과정

<관심분야> 차세대 이동 네트워크, 무선 애드혹 네
트워크, 드론 ICT 융합 기술

조 유 제 (You-Ze Cho)



1982년 : 서울대학교 전자공학
과 졸업
1983년 : 한국과학기술원 전기
전자공학 석사
1988년 : 한국과학기술원 전기
전자공학 박사
1989년~현재 : 경북대학교 전자
공학부 교수

1992년~1994년 : Univ. of Toronto, 방문교수
2002년~2003년 : 미국 국립표준연구소(NIST), 객원
연구원

<관심분야> 차세대 이동 네트워크, 무선 애드혹 네
트워크, 이동성 관리 기술, Future Internet