

Research Article

Reliable Flying IoT Networks for UAV Disaster Rescue Operations

Taemin Ahn,¹ Jihoon Seok,² Inbok Lee,³ and Junghee Han⁴ 

¹Samsung Electronics Co., Ltd., Yong-tong-gu, Suwon, Gyung-gi-do, Republic of Korea

²SK Hynix Co., Ltd., 2091 Gyeongchung-daero, Bubal-eup, Icheon-si, Gyeonggi-do, Republic of Korea

³Department of Software, Korea Aerospace University, 6 Hanggongdaehang-ro, Deokyang-gu, Goyang-si, Gyeonggi-do 412-791, Republic of Korea

⁴Department of Telecommunication and Computer Engineering, Korea Aerospace University, 76 Hanggongdaehang-ro, Deokyang-gu, Goyang-si, Gyeonggi-do 412-791, Republic of Korea

Correspondence should be addressed to Junghee Han; junghee@kau.ac.kr

Received 15 December 2017; Accepted 28 June 2018; Published 6 August 2018

Academic Editor: Omprakash Gnawali

Copyright © 2018 Taemin Ahn et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, UAVs (unmanned air vehicles) have been developed with high performance, and hence, the range of system utilizing UAVs has also been widening. UAVs are even considered as connected mobile sensors and are claimed to be the future of IoT (Internet of Things). UAVs' mission fulfillment is relying on the efficiency and performance of communication in a FANET (Flying Ad hoc NETwork) environment where UAVs communicate with each other through an ad hoc network without infrastructure. Especially, for mission-critical applications such as disaster rescue operations, reliable and on-time transmission of rescue information is very critical. To develop the reliable FANETs, a realistic network simulation platform for UAV communication has become an important role. Motivated by this observation, this paper first presents a study on realistic FANET environment simulation platform. On top of the proposed platform, we also design a stable UAV communication protocol with high packet delivery and bounded end-to-end communication delay.

1. Introduction

Various types of UAV applications and services have been emerged, which range from commercial applications such as a real estate photography to public safety services. In the past, UAVs were used mainly in the field of defense including border surveillance and ground attack, but now they are used in various industrial fields such as logistics, IT, and agriculture. They have also been used in suppression, disaster relief, traffic monitoring, government departments, volcanic eruptions, and wildlife surveillance [1]. For example, Google's Rune Project, which uses airship-based UAVs to expand its wireless Internet penetration network, Amazon's next-generation delivery system using unmanned helicopters, Yamaha's industrial UAV, for use in agriculture, and AirDroid's personal drones that capture images and photographs are actively under development.

With emergence of smart and converged services for these autonomous UAVs, there has been rapid increase in the need for reliable connection among UAVs and control centers. However, the communication environment between UAVs

changes frequently due to the fast movement of the nodes. In addition, since it moves in three-dimensional space, the surrounding topography changes frequently as well. This can cause frequent disconnection of the network link among UAVs and the infrastructure. To support reliable communication, reliable FANETs play important roles.

Inspired by this observation, this paper develops a new technology for reliable UAV networks. Main features of the proposed scheme are as follows:

- (i) *We design and implement a realistic UAV simulation platform:* In order to develop a technology optimized for FANET environment, realistic network simulation is critical. To do this, we design a platform with combining a network simulator NS-3 and a robot simulator Gazebo together. Using these two tools, we develop a realistic UAV simulation platform that can realize and simulate UAV operations in the realistic FANET environment.
- (ii) *We propose a reliable FANET routing methodology:* We examine problems of traditional routing schemes

and we design a stable UAV communication algorithm with high packet delivery and bounded end-to-end communication delay. Specifically, the proposed routing algorithm has three main features: (1) hierarchical node clustering, (2) time division packet transmission, and (3) real-time estimation of transmission delay.

The rest of the paper is organized as follows: Section 2 introduces existing simulation platforms and routing protocols with discussing problems and limitations of existing platforms and routing protocols in FANETs. In Section 3, we introduce a design of the proposed simulation platform and a routing protocol developed for FANET environment. Section 4 explains the experimental procedure and the results for the performance evaluation of the proposed approach. Finally, Section 5 concludes this paper with discussion.

2. Related Works

2.1. Simulation Platforms. To develop reliable UAV systems and applications, it is important to be able to simulate UAV operations in a realistic way. So far, many network simulators have been developed, such as NS-2 [2], NS-3 [3], QualNet [4], and OMNet [5]. To use these simulators for mobile networks, either a mobility model like Gauss-Markov [6] is specified or trace files created by other software tools like SUMO [7] are given as inputs for the network simulators. Although SUMO software can produce very realistic motion patterns of nodes, there are some limitations that it can only make these patterns a priori, not on the fly. Also, it can only generate movement traces in 2D space. In contrast, Gauss-Markov mobility [6] model in NS-3 can make the movement of nodes in 3D space. However, a mobility model is based on statistical probability, which implies that previous movements of nodes influence the next movement decision in the model. Therefore, it has fundamental limitations to simulate UAV motions for disaster search and rescue operations. More importantly, these network-based simulators cannot reflect UAVs' realistic motions and behaviors affected by physical and mechanical features and its surrounding environment.

On the other hand, robot simulators like Gazebo [8] can control the physical movement of UAVs. Gazebo is a tool widely used for the development of robots with various physical characteristics, such as conveyor belts, unmanned probes, and line tracers. This tool can incorporate various sensor modules for these robots, and so we can make a UAV equipped with Wi-Fi signal detection sensor or a camera sensor. Also, it is possible to implement a collision avoidance algorithm inside Gazebo for controlling motion of the robot. Gazebo is an OpenGL-based simulation tool, and it can visualize the behavior of the robot as shown in Figure 1(b).

Overall, solely using the existing network simulators or robot simulators, we can test only either network performance or robot motion operations, not both considered. To handle this issue, this paper proposes a new FANET platform with combining the network simulator NS-3 and the robot

simulator Gazebo together. To incorporate these two tools, we adopt ROS (Robot Operating System) [9] as a middleware. ROS provides various development tools as well as various functions for hardware abstraction, subdevice control, and interprocess message passing for software development. As shown in Figure 2, ROS handles a process that performs computation as a node. Each node sends and receives messages to the publisher-subscriber structure through a channel called *Topic*. Through this, it is possible to exchange information with nodes or to share desired information with a desired node.

2.2. Traditional Routing Protocols for MANETs. In this section, we introduce some of ad hoc routing protocols that are commonly used in MANET (Mobile Ad hoc NETWORK) or VANET (Vehicular Ad hoc NETWORK) environments and examine applicability of the existing protocols to UAV system in a FANET environment. An ad hoc routing protocol is specially designed to cope with link disconnection or topology change due to high mobility feature in an ad hoc network environment.

First, ad hoc routing protocol is a method to cope with link disconnection or topology change due to node mobility in ad hoc network environment where wireless communication is performed between nodes scattered without AP. There are three types of routing protocols: (1) table-driven proactive routing protocols, (2) on-demand reactive routing protocols, and (3) hybrid routing protocols. A proactive routing protocol is a method of preliminarily computing a route to create a routing table and updating it periodically by an algorithm. Typical proactive routing protocols are DSDV [10] and OLSR [11]. On the other hand, reactive routing protocols (e.g., DSR [12] and AODV [13]) are also called on-demand routing protocols. The reason for this is that, contrary to the proactive routing protocol, there is no operation if the packet transmission request does not occur, but when the packet transmission request is generated, the protocol for determining the forwarding path gets started only at that time. The hybrid routing protocol is a routing method that combines the advantages of proactive routing and reactive routing. Depending on the network conditions, proactive routing and reactive routing may be used in combination.

In addition, most VANET routing algorithms can be classified as either topology-based or geography-based. Due to its simplicity, geography-based routing has been proved to be more suited for VANET. The most critical issue in location-based approaches is that all nodes in VANET environment including source, destination, and forwarding nodes can move dynamically at a high speed. Hence, the positional information of destination node might need to be propagated by neighbors to all nodes at short intervals. Also, these geography-based algorithms have inherent limitations in the sense that packets cannot be delivered due to network disconnection or partition under low traffic density. To overcome these limitations, delay tolerant network- (DTN-) based methods have been proposed. In DTN protocols like vehicle-assisted data delivery (VADD) [14], a moving vehicle carries a packet and waits until a new vehicle moves into its vicinity.

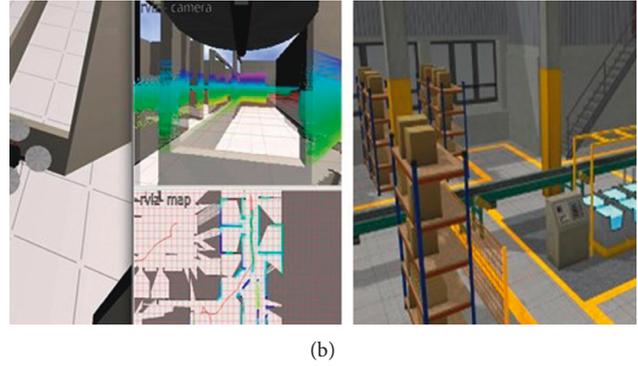
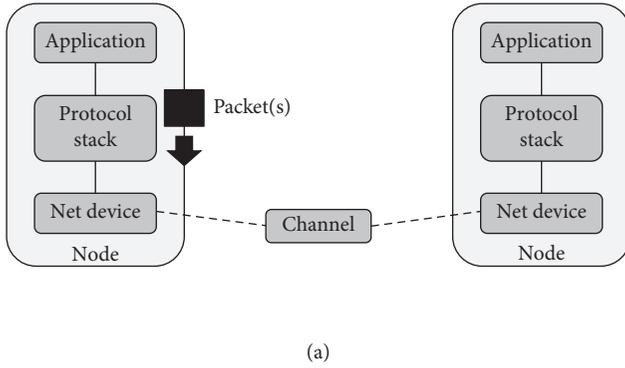


FIGURE 1: Existing network and robot simulators. (a) NS-3 structure [3]. (b) Gazebo [8].

3. Proposed Approach

Although various routing protocols have been developed so far, applying these protocols for FANETs is a challenging task due to high speed mobility of UAVs. Furthermore, the network simulation platform reflecting the UAV’s physical features is not available yet, to the best of our knowledge. In what follows, this paper identifies important issues to solve for FANETs and proposes the integrative simulation platform. Then, we present the new routing protocol for FANETs with complementing the limitations of traditional methods.

3.1. Problem Statement. Based on the above observations, we identify a target scenario of UAV systems for disaster relief as follows:

- (i) *Need for ad hoc flying networks:* if a disaster such as an earthquake, a landslide, or a tsunami occurs, it is not easy to use existing infrastructure in the area. Therefore, there is a need for a method of networking and communicating between UAVs without the help of an AP or a base station.
- (ii) *Need for collaboration among UAVs:* even if the performance of the UAV improves day by day, there is a limit to finding a large number of survivors in a short period of time with only one high-performance UAV. Therefore, we assume a scenario in which a plurality of UAVs looks for survivors.
- (iii) *Reliable data sharing among UAVs:* UAVs are fast enough to reach tens to hundreds of kilometers per hour and are not as dense as the nodes in MANET or VANET environments. In this environment, it is most important to guarantee packet transmission within a limited time in order to quickly and accurately transmit the message related to collision avoidance, flight control message, or survivor discovery information according to mission execution to other nodes.
- (iv) *Need for UAVs with various sensors:* for effective disaster rescue operations with UAVs, various sensors such as camera sensor, Wi-Fi sensor, voice signal sensor are needed to detect survivors. Especially, in case of survivors buried because of disasters, it is

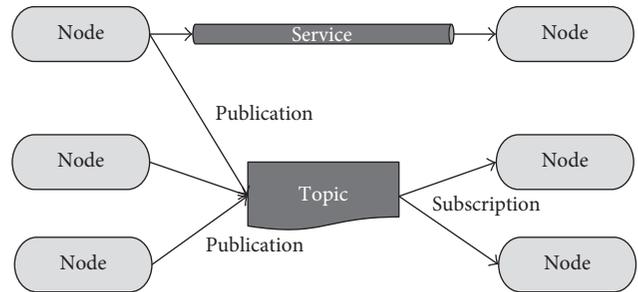


FIGURE 2: Robot Operating System (ROS) [9].

difficult to detect survivors only with camera sensors. Therefore, each UAV should be able to attach a Wi-Fi signal detection sensor or a voice signal detection sensor to detect the location of a survivor and perform rescue operation.

3.2. Development of FANET Simulation Platforms. As we mentioned before, it is important to be able to simulate UAV operations in a realistic way. Solely using the existing networks simulators or robot simulators, we can test only either network performance or robot motion operations, not both considered. To overcome this limitation, this paper proposes a holistic UAV simulation platform by integrating network simulators and robot simulators together. Specifically, we develop a new simulation platform to realistically implement the FANET environment as described in Figure 3. By adopting a robot simulator Gazebo [8], the proposed platform is able to represent the realization of UAV’s mechanical movement such as cluster flight, collision avoidance, and 3-dimensional flight. Furthermore, the proposed platform with Gazebo can support operations of various sensors including a radar and a camera attached UAVs as well. On the other hand, we also integrate a conventional network simulator [2, 3] in the proposed platform to realistically analyze communication features among UAVs. Since multiple UAVs collaborate with each other to perform mission critical operations, successful on-time communications among UAVs are very essential. Using the NS-3 network simulator with Gazebo, we can accurately analyze communication performance of UAVs while simulating their flight motion and sensor operations at the same time.

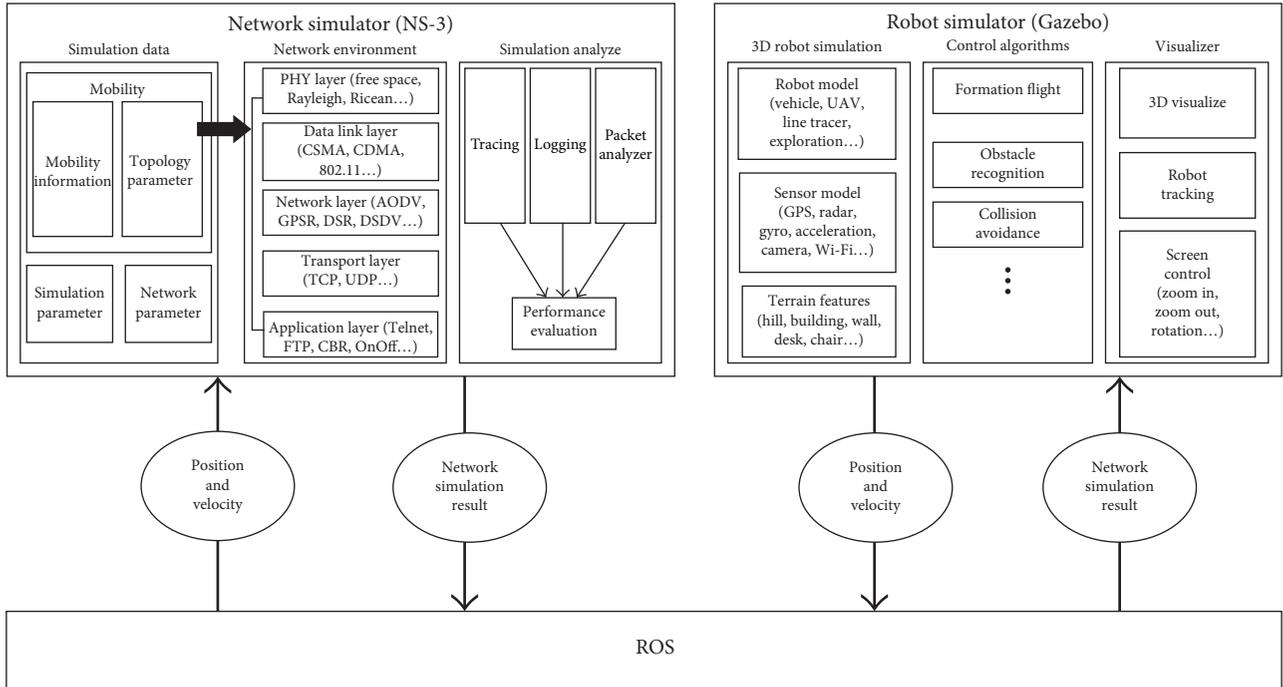


FIGURE 3: The proposed architecture of a FANET simulation platform.

To make both NS-3 and Gazebo interoperate with each other, we use a middleware called ROS [9], which is widely used in a robot control field. For example, Gazebo simulates a realistic robot motion in the simulator, making the same movement as a real UAV. UAV motion-related information is transmitted to the NS-3 network simulator through the ROS, which is illustrated as the right-most arrow line in Figure 3. In this platform, Gazebo transmits UAVs' current locations and velocities to NS-3. Based on this information, NS-3 can set up the nodes dynamically, each of which corresponds to each UAV in Gazebo. Specifically, initial locations of these nodes are determined based on this information from Gazebo, and these nodes move in NS-3 simulation with considering the velocities of UAVs transmitted from Gazebo on the fly. Overall, this proposed simulation platform not only enables the evaluation of the communication performance according to the realization of the UAV's mission in NS-3, but also realistically measures the UAV's performance according to the current communication state.

In order to integrate a network simulator NS-3 and a robotic simulator ROS, the two simulators must exchange information with each other. Specifically, NS-3 needs motion information of the UAVs created by the robot simulator, and the robot simulator requires the result of the communication from the network simulator. In the proposed platform, these two different processes in NS-3 and Gazebo communicate with each other via ROS interface to exchange information. In this system, a robot created by Gazebo becomes a subscriber node. ROS generates a publisher node to control the UAV and sends a UAV motion control message (twist) to the subscriber node. The physical movement information of the UAV created through this

process can be transmitted to the subscriber node of the network simulator through the publisher node of the robot simulator and conversely the mission execution message of the UAV generated at the publisher node of the network simulator can be transmitted to the subscriber node of the robot simulator (in the current implementation, these two processes are designed to update the data every 0.1 simulation second. The cycle can be adjusted depending on the volume of the exchanged data and degree of urgency) as described in Figure 3.

The proposed simulation platform is largely divided into a network simulator (NS-3), a robot simulator (Gazebo), and an interface part (ROS), each of which will be described in detail. NS-3 consists of three main parts: simulation data, network environment, and simulation analysis parts. The simulation data includes parameters related to the topology such as the size of the topology, the number of nodes, network layer parameters, simulation time, simulation environment setting, and finally mobility information related to node movement. Among them, the mobility information may be created by the mobility model. But the node in the proposed platform uses the ROS to acquire the motion information generated from the Gazebo instead of theoretical models. In addition, the result of the communication made by the realistic network simulation of the NS-3 is transferred to the robot simulator through the ROS. The 3D robot simulation part of the robot simulator is composed of various robot models such as UAVs and unmanned probes, as well as sensor models such as camera, radar, and GPS which are essential for the task of the robot. Furthermore, terrain features of 3D environment are also included. In addition, Gazebo incorporates an obstacle avoidance scheme with a visualizer part. The network simulation results

received earlier in the NS-3 are used to determine the next movement of UAVs, and the next movement of the resulting UAV is sent to the NS-3 via the ROS in turn. In this way, two simulation tools interact to control the movement of the UAV according to the current communication situation and conversely analyze the communication performance according to the movement of the current UAV.

3.3. FANET Routing Protocol. Thanks to the proposed simulation platform in Section 3.2, we are able to realistically analyze the problems of existing routing protocols in FANETs. The proposed platform not only enables the evaluation of the communication performance in NS-3, but also realistically measures the UAV's performance reflecting physical environments provided by Gazebo tools. With this FANET simulator, we are able to design routing protocol optimized for the mission of UAV for disaster relief by realistically considering FANET environment. Also, we are able to effectively verify the performance of the proposed algorithm compared with traditional routing protocols based on the proposed simulation platform.

As mentioned in Section 2, both traditional reactive and proactive routing algorithms have their own drawbacks to be applied to FANETs. Since most information of UAVs is mission critical and so requires to be delivered (1) in high success ratio and (2) in time. To achieve these goals, the proposed algorithm adopts flooding-based routing methodology and modifies it with the following features:

- (i) *Hierarchical node clustering*: the problems of flooding-based routing methods happen mainly due to the high volume of packets, which is inherent limitation of broadcasting or flooding based packet transmission. To ameliorate this problem, the proposed routing method proposes clustering-based mechanism so that it can reduce the amount of packet transmission.
- (ii) *Time division packet transmission*: to avoid further packet congestion, we transmit packets in time division manner on top of clustering policy.
- (iii) *Real-time estimation of end-to-end delay*: when constructing clustered FANETs, we perform real-time estimation of packet transmission delay. In FANET environments, on-time information transmission is really essential. With the proposed real-time estimation procedure, we are able to predict the worst case end-to-end delay for FANET communications and check the feasibility of a given set of clusters with a given time interval. Thanks to this procedure, we can guarantee the worst-case packet transmission delay so that we can provide reliable communication service in FANETs.

3.3.1. Hierarchical Node Clustering. The biggest problem of a traditional flooding mechanism [15] is that as the number of nodes increases, the packet overhead in the network exponentially increases. This causes collision between

transmitted packets. In order to solve this problem, we propose a clustered network structure that hierarchically groups nodes—nodes are clustered in two layers, edge nodes and cluster heads. Hierarchical node clustering refers to a method of dividing a node into several groups and selecting a head node of each group and communicating through this node as shown in Figure 4. In this example, nodes in the overall topology are divided into four clusters, and each cluster has one cluster head node. Bold red solid arrows and dotted arrows indicate unicast transmission from edge nodes and flooding transmission among cluster heads, respectively. In particular, a drone on the network edge carrying out a disaster relief mission can send this information to each cluster head in a unicasting mechanism. When a cluster head drone receives unicast packets, it floods this packet to other cluster heads to share this information. In the traditional flooding mechanism, the edge node can trigger flooding while only cluster heads initiate flooding in the proposed clustering methodology. By doing this, we can significantly reduce the packet volumes on the networks and therefore can improve packet delivery rate. In Section 4, we verify this strategy using the proposed FANET platform.

3.3.2. Time Division Flooding. Although hierarchical node clustering makes it possible to reduce the total number of packets transmitted and received to a certain extent, if the number of clusters is large and/or the amount of data to be shared is huge, the proposed hierarchical method alone is not enough to support reliable FANET communication. Motivated by this observation, this paper incorporates additional feature, called time-division, in order to guarantee real-time and reliability of communication even when a large number of UAVs and drones perform many applications including collision avoidance and survivor search as assumed in this paper. Specifically, we divide the time axis of packet transmission into several time slots and allocate them to each cluster head to send packets only during their own time slots. The TDMA-based communication method of accessing the channel only to each time slot by dividing the time axis has already been studied extensively by various network protocols [16, 17]. For UAV and drone networks, a TDMA-based MAC layer protocol has been applied to a single-hop or multihop UAV network [16, 17] for improving the reliability of communication in FANETs. However, most of these TDMA algorithms are implemented in the MAC layer, which cause modification of existing MAC devices.

In this paper, we propose an algorithm that improves communication performance by adjusting the timing of sending flooding packets at the network layer, which can be compatible with existing MAC devices such as Wi-Fi and Zigbee. In the proposed protocol, each cluster head is assigned a certain time slot for packet transmission while edge nodes send unicast packets to their cluster head in CSMA/CA scheme. Figure 5 illustrates this idea with the time-axis representation of packet transmission at each cluster. In this example, we assume that five edge nodes (N_1 , N_2 , N_3 , N_4 , and, N_5) have detected the survivor's

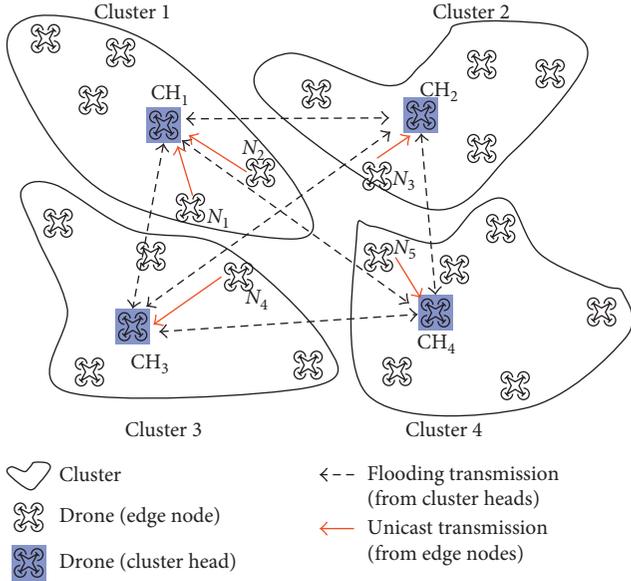


FIGURE 4: Hierarchical network structure.

signal, and they transmit this information to their own cluster heads as in the previous cluster network example of Figure 4.

As shown in this figure, unicast packets sent to the cluster head are not immediately forwarded. Instead, they are queued in the packet queue at corresponding cluster head node waiting for the time slot during which they can send the packets. For example, cluster₁ is assigned a time slot at t_3 and hence the packets received from N_1 and N_2 are queued and flooded at time t_3 . Note that the packet from N_5 waits in the queue for a long time for a next time slot because its previous time slot has just passed before a packet arrives.

Overall, the proposed scheme with network-layer time division feature can improve packet delivery rate by avoiding collision among flooding packets without requiring modification of existing MAC layer devices.

3.3.3. Worst-Case Delay Estimation. So far, we have proposed hierarchical node clustering and time division schemes to reduce the number of packets and avoid collision of flooded packets. For mission critical disaster rescue operations of UAVs, it is also important to guarantee packet delivery within some time bounds (i.e., transmission deadline). Furthermore, since the moving speed of the UAV is as high as several tens to several hundred km/h, it is necessary to secure the packet transmission within the time limit in order to avoid collision between UAVs. For such applications, correctness of the systems depends on temporal behavior as well as functional correctness. We call such applications as real-time systems. Real-time system usually refers to hardware and software systems with time constraints. Specifically, real-time system requires that the system should response to events within specified deadlines [18, 19].

From this aspect, we now present a new methodology that guarantees real-time constraints of a target FANET system. Related to this timing delay issue, we answer the following three questions:

- (i) *How to organize the cluster?* Depending on the hierarchy of clustered networks and the number of clusters, packet collision can be increased. Packet collision causes uncertainty of successful packet transmission and therefore makes it difficult to guarantee packet delivery time within deadlines. For example, as the number of clusters increases, the time slot interval of each cluster head becomes shorter, so that the risk of collision of the flooding packet sent by the head increases.
- (ii) *How to assign time slots to cluster heads?* The second problem to answer is about parameters related to the time division packet transmission. The most important advantage of the time-division packet transmission is the fact that it can minimize packet collision and its periodicity can enable us to predict the worst transmission delay time. In this paper, we present the theoretical analysis for predicting worst case timing delay of packet transmission under the proposed hierarchical clustered networks.

Figure 6 illustrates the process for estimating the worst-case packet transmission delay time. In this example, the node N_1 generates the first packet q_1 at time t_1 and the second packet q_2 at t_2 , and the cluster head (i.e., CH_1) floods q_1 and q_2 received from N_1 during its assigned time slot (i.e., t_3) (for simplicity of explanation, we assume propagation delay is zero in this example). These packets are received by CH_2 at t_3 assuming no propagation delay, and they have been queued waiting for the time slot for CH_4 . Suppose the time slot for CH_4 is not long enough to accommodate these two packets. In this scenario, the packet q_1 might be flooded at t_4 first and then the second packet q_2 needs to wait for the next time slot at t_5 . As a consequence, the destination node N_0 would receive q_1 and q_2 at t_4 and t_5 , respectively.

$$WD = \sum_{\text{cluster } c \in \text{path}} WD_c. \quad (1)$$

In the scenario shown in Figure 6, the worst end-to-end transmission delay (i.e., WD) is determined by the sum of the worst-case transmission delay times of each cluster (i.e., WD_c) on the transmission path as shown in (1). If the packet generation period is p , the generation time of the q th packet is $(q-1) * p$. Suppose that the time required to transmit all q packets waiting for transmission is defined as $w(q)$. In this case, since the arrival time of the q th packet is $(q-1) * p$, the transmission delay time is calculated as shown in the following equation:

$$w(q) - (q-1) * p. \quad (2)$$

Therefore, since the worst transmission delay time in the corresponding hop is the largest one of the transmission delay time, WD_c can be written as

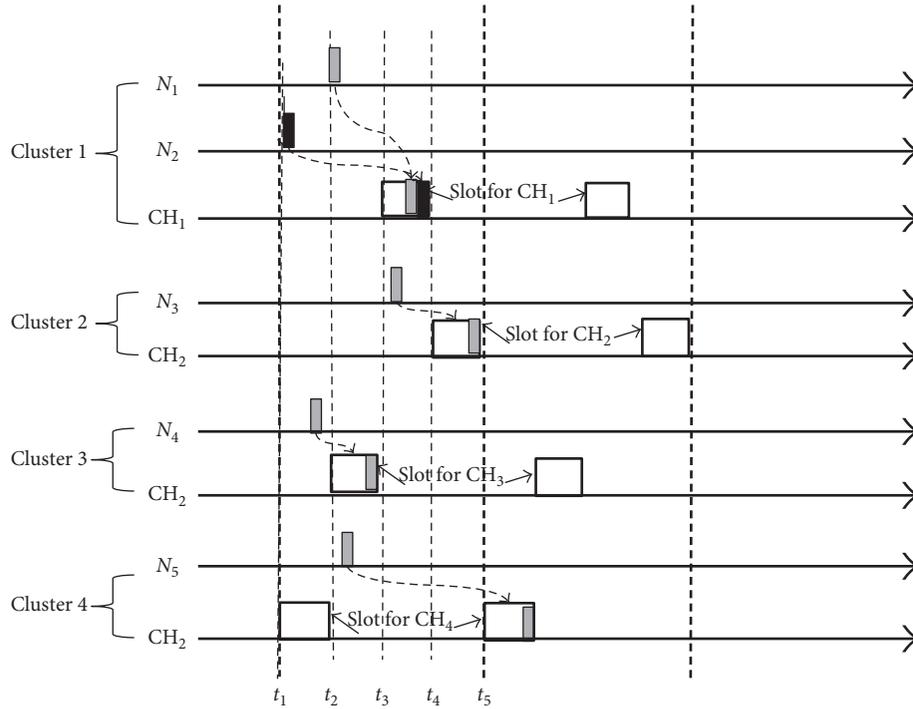


FIGURE 5: TDMA-based flooding at cluster heads.

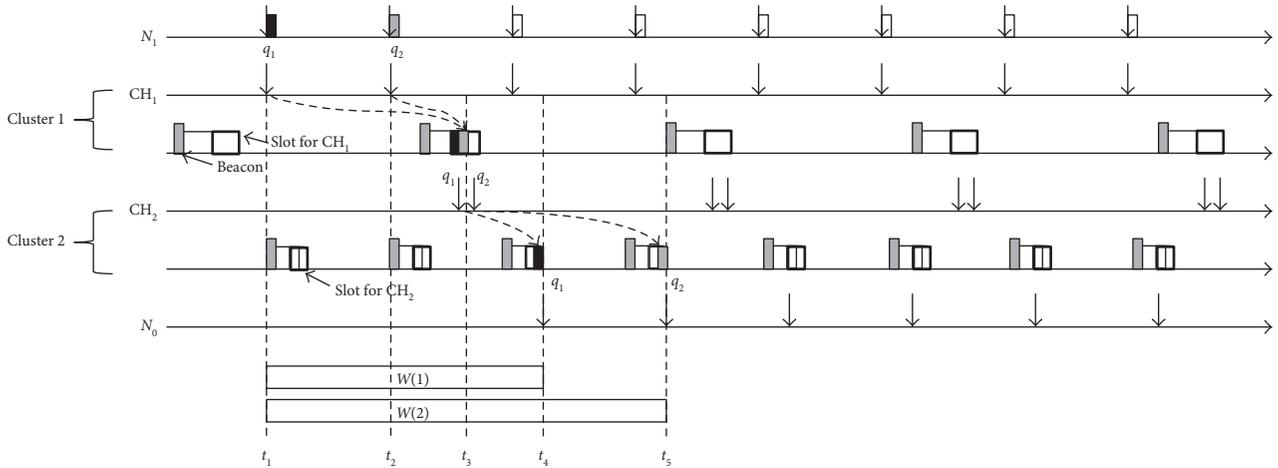


FIGURE 6: Worst-case delay estimation.

$$WD_c = \max_{q=1,2,\dots,Q} (w(q) - (q-1) * p), \quad (3)$$

where Q is the first integer satisfying $w(q) \leq Q * p$, and the worst transmission delay time, WD_c can be predicted by knowing $w(q)$ using the following equation:

$$w(q) = \left\lceil \frac{q * b/R}{S} \right\rceil * p_c S = \frac{b/R}{p} * p_c. \quad (4)$$

In (4), b/R means the time when b bits are sent to the rate R , that is, the time taken to send one packet. Therefore, S means the time taken to send all packets generated within one period, and p_c means the period for packet flooding of

cluster c . This equation is a variation of worst case busy period theorem [20]. In order to predict the worst-case transmission delay, it is necessary to know not only the packet size and the transmission rate of the link but also the period of packet generation in the application, the period of packet flooding in the cluster head, and the size of the time slot of each cluster head. In other words, the packet transmission period and the time slot allocated to each cluster head must be determined so that the worst transmission delay time can be calculated.

Using this procedure, we are able to predict the worst case end-to-end delay for FANET communications and check the feasibility of a given set of clusters with a given time interval. With this real-time feasibility check scheme,

we will verify two-level hierarchical clustering which can improve PDR and end-to-end delay in the next section.

4. Evaluation

In this section, we develop the proposed FANET simulation platform with NS-3 and Gazebo tools to realistically evaluate network performance in the FANET environment as presented in Section 3.2. On top of the platform, we verify the FANET routing protocol proposed in Section 3.3. This paper conducted two types of evaluation. We first focused on the primitive communication performance of the proposed routing protocols by examining packet delivery ratio and delay in Section 4.3. In what follows, we also measured UAV's ability to successfully localize and approach survivors in Section 4.4.

4.1. Simulation Scenario. The simulation scenario for measuring the performance of the proposed routing protocol is shown in Figure 7. The figure is a screenshot from a Gazebo tool during the simulation. The red star represents the locations of two survivors and the red circle represents the cluster head UAV. Also, the white box in the center represents the leader node controlling all drone members. The entire nodes except the leader node are divided into four clusters as shown in Figure 7.

Nodes that have detected the survivors' Wi-Fi signals send this information to their cluster head node. Then, the cluster head node collects the received packets waiting for its time slot, and the head node floods them at once during the time slot. Any UAV node receiving flooded survivor detection information from three or more different nodes changes its flight direction to get near the survivors.

The parameters for the simulation are shown in Table 1. Reflecting the fact that the maximum Wi-Fi range is about 250 meters in general, the distance between UAV nodes is set to 200 meters for maximum coverage. The average moving speed of each UAV is 5~10 m/s, and 16 survivors are randomly distributed in the area.

4.2. Performance of Existing Mobile Routing Protocols. Before we introduce a new routing protocol, we first analyze limitations of the traditional routing protocols in FANET environments in more detail. For realistic simulations, we conduct evaluation with the above integrative simulation platform in Section 3.2. Figure 8 first shows the performance of a reactive routing protocol AODV [13] and a proactive routing protocol OLSR [11] (AODV and OLSR are well-known representative protocols among the various routing protocols introduced in Section 2).

As shown in Figure 8(a), for both AODV and OLSR, PDR (packet delivery rate) performance decreases dramatically as nodes move faster in FANET. Also, even at very low mobility rates, the network is still suffering from low PDR. Note that AODV shows better performance than OLSR. This is explained by the fact that AODV is an on-demand routing protocol, which builds a path for transmitting packets when a packet transmission request occurs.

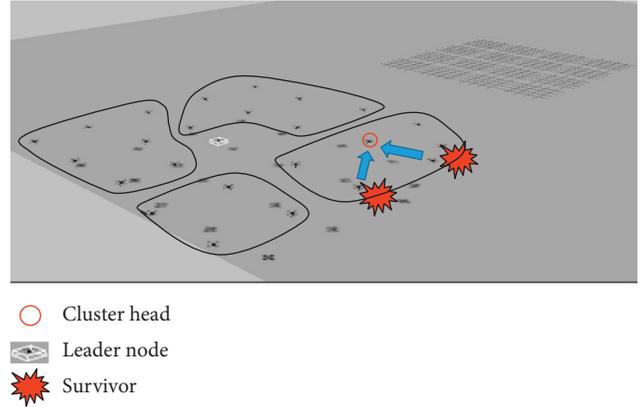


FIGURE 7: Simulation scenario.

TABLE 1: Simulation parameters [6, 8].

Parameter	Value
Channel	Wi-Fi
Application	On-off application
Traffic type	UDP
MAC protocol	802.11
Packet size	1024 (byte)
Data rate	DsssRate 11 Mbps
Topology size	2500 × 2500 (m ²)
Internode spacing	200 (m)
Simulation time	100 (s)
Number of nodes	45
Node movement speed	5~10 (m/s)
Number of survivors	16
Number of clusters	2, 4, 22, 44
Packet generation period	0.1(s), 1(s)
Packet flooding period	0.5(s), 1(s), 5(s)

This is trivial that OLSR is proactive and so it is more suitable for static networks. On the other hand, when the AODV routing protocol is used, the end-to-end delay time increases as the node moves faster. This is because the AODV, which is a reactive routing protocol, floods a large amount of control packets into the network every time the network topology changes, refinds the route, and finally transmits the data packet only after receiving the RREP packet. As a consequence, the frequent topology changes cause long end-to-end delay of the transmission packets.

Overall, the PDR and end-to-end delay performance of AODV and OLSR are not good enough to be applied in fast moving FANET environments. Note that most messages required for a UAV to operate are semibroadcast (or multicast) messages which all flying nodes in the topology should listen for inter-UAV collaboration. From this perspective, we believe that a flooding-like scheme that propagates information to all nodes in the topology might be more appropriate. However, a traditional flooding algorithm shows packet overhead problems as shown in Figure 9. As the number of source nodes that send packets is increased, PDR decreases due to packet overheads. Specifically, overhead causes an increase in the number of collisions among

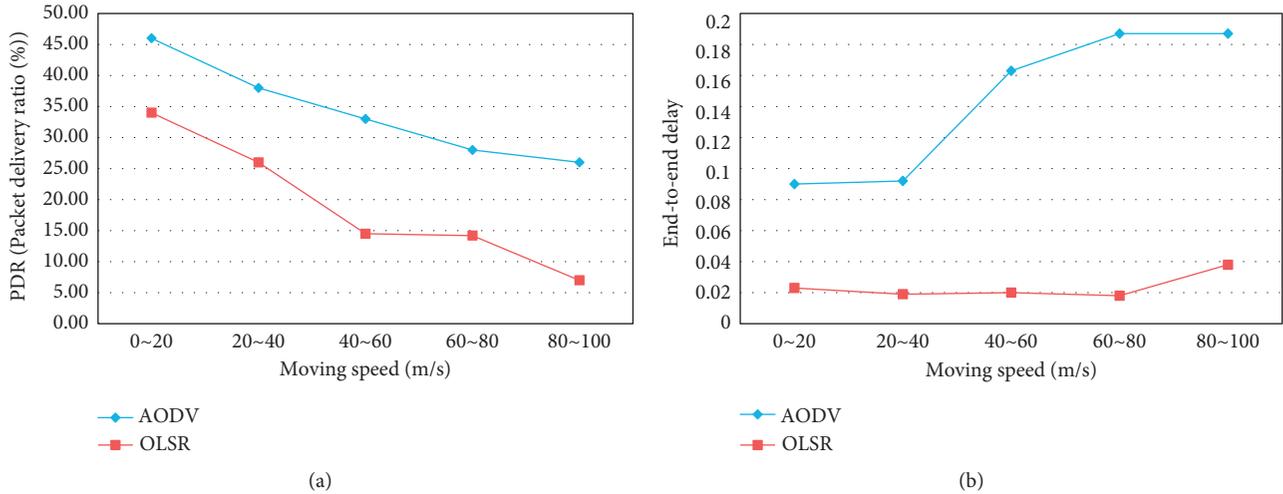


FIGURE 8: Limitations of existing MANET protocols in drone networks. (a) Packet delivery rate. (b) End-to-end delay.

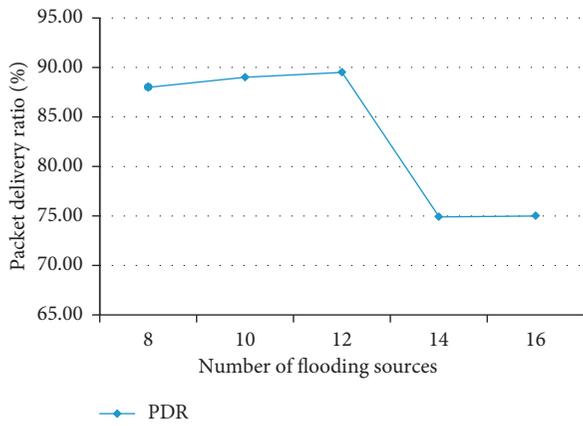


FIGURE 9: How about a flooding protocol?

packets, which means that the number of packet retransmissions in the MAC layer increases.

4.3. FANET Routing Performance. As we mentioned before, we first examine performance of the proposed FANET routing mechanism in two metrics, PDR and end-to-end delay. The experiments are all conducted with the proposed simulation platform with Gazebo and NS-3. This implies that the provided results are reflecting communication performance with realistically considering UAVs’ movement.

4.3.1. Packet Delivery Rate. Figure 10 shows PDR (packet delivery rate) results according to the number of clusters and the number of flooding cycles, respectively. In this environment, the number of clusters has been varied to 2, 8, 22, and 44. Since the number of normal drones except a group leader is 44, in the case of 44-cluster scenario, each cluster has only one drone as a member and the only group member acts as a cluster head as well. In other words, all drones are cluster heads and so they all flood packets during the

assigned time slot without sending packet in unicast way at all.

From this experiment, we first observe that the more the clusters exist, the better the PDR is achieved. Theoretically, the smaller the number of clusters, the longer the time slot of one cluster head node, which means that the risk of collision between flooding packets flooding in the cluster head is reduced. However, the graph shows that the highest data rate is provided when the number of clusters is 22 or 44. The reason for this is explained by the fact that when the number of clusters is low, lots of unicast packets are sent to the same cluster head node. This causes these unicast packets collide with each other. Note that when the flooding rate is less than 1 s, the PDR is relatively low because the number of flooding packets is way too high and so they end up colliding with each other.

4.3.2. Transmission Delay. Now, we examine the end-to-end delay of the proposed FANET routing protocol. Figure 11 shows the histogram of the packet transmission delay of the traditional flooding and the proposed algorithm with clustering and time-division features. In this experiment, the flooding period is 0.5 second with 8 clusters, and the application packet generation period is 0.1 second.

First thing we have noticed is that the average delay of the traditional flooding is better than the proposed routing scheme. However, the distribution of delays shown in Figure 11 shows a very important point. In the conventional flooding method, we found that some packets experience long delays as shown as a long tail in the left figure. This is explained by the fact that in the traditional flooding, packets are flooded as they are received and so many packets are subject to be retransmitted due to packet collision.

In contrast, the proposed algorithm shows that most packets are delivered within 0.5 seconds thanks to its TDMA feature. In other words, the proposed scheme is able to bind the transmission delay within a certain value. The worst-case delay guarantee is very important feature for mission-critical applications such as disaster rescue operations with time

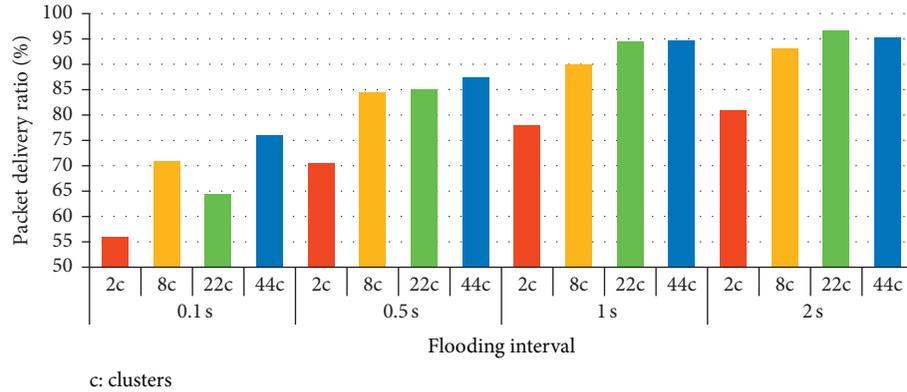


FIGURE 10: PDR (packet delivery rate) with 2, 4, 22, and 44 clusters for various flooding periods.

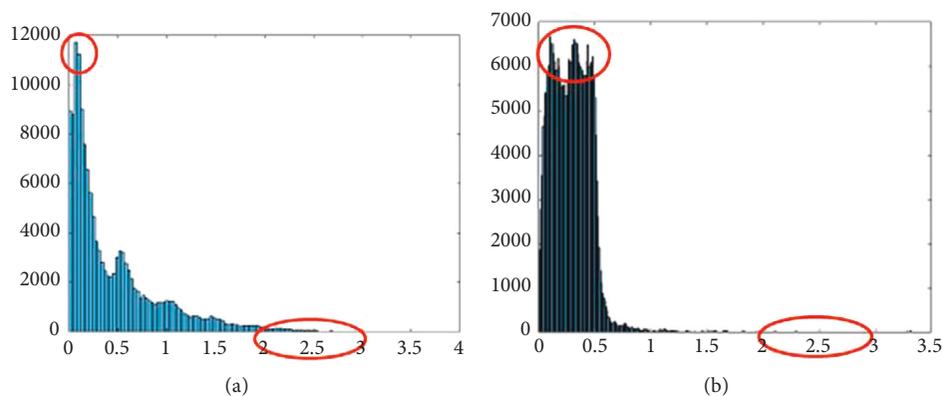


FIGURE 11: Transmission delay with original flooding scheme (a) and with the proposed algorithm (b).

constraints. Note that since the proposed TDMA transmission operation is performed at the network layer, not at MAC layer, it cannot completely guarantee that the packet is transmitted only in a predetermined time slot. This fact explains a few number of packets with long delays in the right graph (in order to guarantee hard deadline satisfaction, a MAC layer model supporting GTS (guaranteed time slot) such as Zigbee [17] should be incorporated with the proposed scheme).

4.4. UAV Mission Performance. In this section, we evaluate UAV mission performance by measuring how quickly each UAV approaches survivors. For this evaluation, we assume two types of applications running on UAVs, app_1 and app_2 as shown in Table 2. Recall that the main goal of the proposed rescue system is to locate survivors as quickly as possible in a stable communication environment. To do this, each UAV detecting a survivor’s Wi-Fi signal is supposed to spread this information in order to make neighboring UAVs move near survivors in the proposed rescue scenario. Once these multiple UAVs arrive around the survivor, they cooperate with each other to accurately locate the exact survivor’s location together using several localization metrics such as triangulation [21, 22]. So, the faster the time to approaching the survivor, the better the mission performance.

TABLE 2: Application parameters.

Parameter	app_1	app_2
Packet size (byte)	1024	512
Packet generation interval (sec)	1	0.1, 0.5, 1
Flooding interval (sec)	5	5
Number of UAVs	45	45

In this paper, we focus on how quickly each UAV can move near the survivor (localization algorithms are beyond this paper’s scope).

Figure 12 shows a subset of our experimental results as a cumulative distribution function (CDF). The x axis and y axis represent the time to approach and the number of UAVs, respectively. From these results, we first observe the performance of the traditional flooding mechanism drops dramatically as the packet transmission rate increases, as shown in Figures 12(a)–12(c). This is because the network becomes more congested due to the faster period of packet transmission. Therefore, the probability that a neighbor node can successfully receive a packet is lowered and hence the time for the UAV to start approaching the survivor gets delayed. As an extreme example, when the packet generation cycle is 0.1 second, any node cannot receive app_2 packets and so no node approaches the survivor at all until the end of the

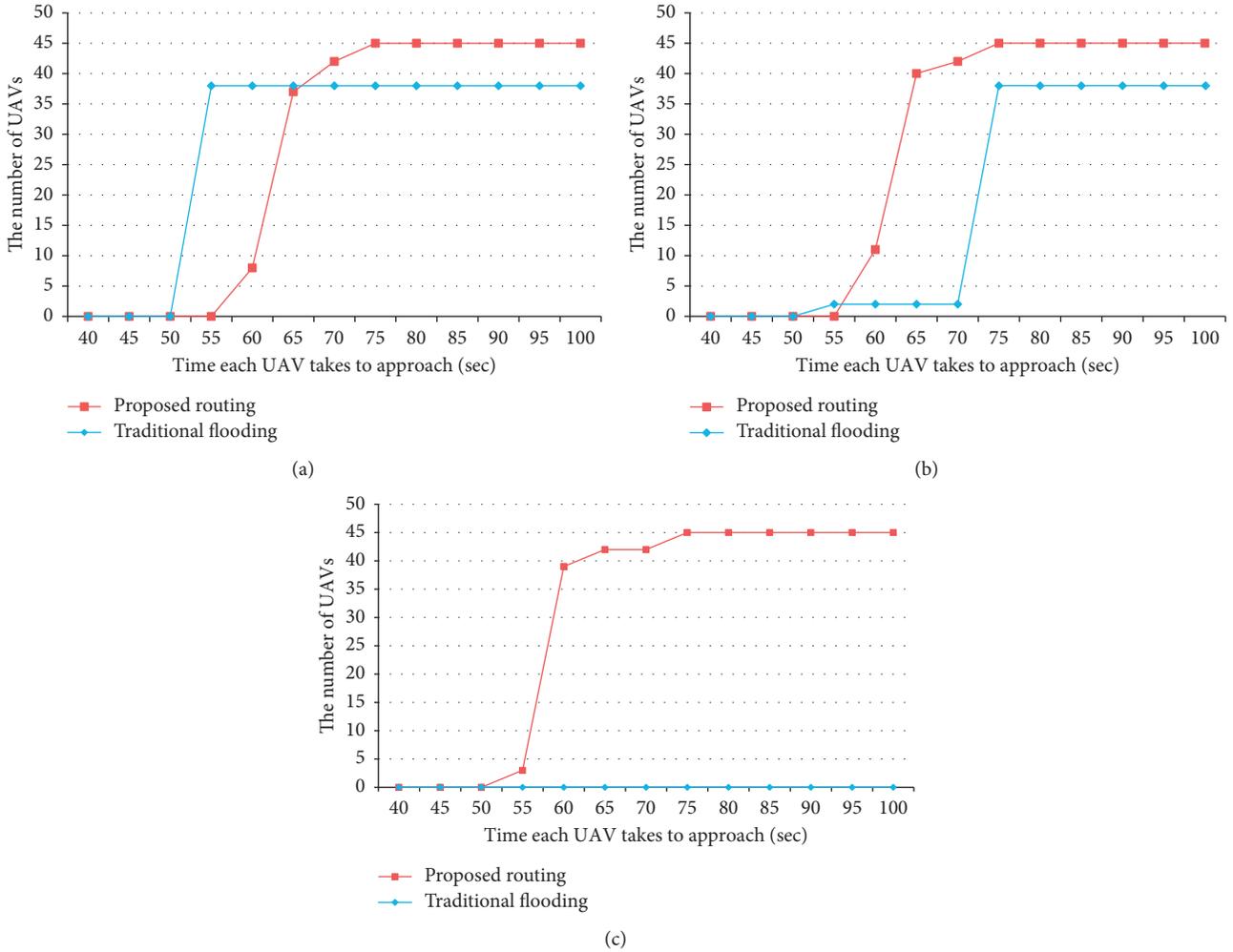


FIGURE 12: Mission performance comparison: (a) app_2 's transmission interval = 1 s, (b) app_2 's transmission interval = 0.5 s, and (c) app_2 's transmission interval = 0.1 s.

simulation. On the other hand, the performance of the proposed routing method is hardly affected by the application's packet interval. This is because the proposed algorithm reduces packet congestion through clustering and time-division transmission, as shown in the red-dot graph. In all of these three scenarios, we can see that most nodes start approaching the survivor in 50~70 seconds. One interesting thing is that in the first scenario with flooding interval 1 sec, the traditional flooding seems to provide better performance at the beginning than the proposed scheme. For example, after 50~55 seconds, around 38 UAVs approach the survivor in the traditional scheme whereas less than 5 UAVs move toward the survivor in the proposed method, as shown in Figure 12(a). This can be explained by the fact that when the packet generation cycle is 1 second, the network is not that congested at the beginning. So in this case, the traditional scheme can be beneficial because it floods packets as soon as it is ready. In contrast, the proposed method is supposed to wait for its time slot even when the network is not congested. As a consequence, transmission delay due to the time division transmission can affect the

mission performance of this simulation. However, as time goes, the flooding packet volume gets increased and so the network becomes more congested than before. In the latter part of the simulation, the proposed scheme ends up performing better than the traditional flooding.

5. Discussion

In this paper, we first propose a new simulation platform to overcome the disadvantages of existing network simulator and robot simulator and to create a realistic FANET environment for communication simulation. The proposed platform can realize not only realistic network simulation but also physical movement of robot and sensor operation by using network simulator and robot simulator together. Based on this simulation platform, we propose a new FANET routing protocol that overcomes the disadvantages of existing routing protocols. Since the existing unicast routing protocol is not suitable for carrying out the UAV mission, which requires the propagation of various information with time constraints, this paper develops the cluster-based flooding algorithm with

a TDMA feature as well. In this paper, we verified that even the proposed simple two-level hierarchical clustering with real-time feasibility check methodology can improve PDR and end-to-end delay. We believe that more efficient clustering method will further improve overall performance. To do this, in the next study, we are going to analyze different clustering methodologies and develop the customized clustering method for FANETs.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Ministry of Science, ICT and Future Planning (Award no. NRF-2015R1C1A2A01055444).

References

- [1] H. Cai and Q. Geng, "Research on the development process and trend of unmanned aerial vehicle," in *Proceedings of the Vehicle International Industrial Informatics and Computer Engineering Conference*, Xi'an, Shaanxi, China, January 2015.
- [2] M. Herrera, *NS2-Network Simulator*, Springer, Valparaiso, IN, USA, 2004.
- [3] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the ns-3 simulator," in *Proceedings of the SIGCOMM Demonstration*, Seattle, Washington, USA, August 2008.
- [4] M. Subramanya Bhat, D. Shwetha, and J. T. Author Devaraju, "A performance study of proactive, reactive and hybrid routing protocols using QualNet simulator," *International Journal of Computer Applications*, vol. 28, no. 5, pp. 10–17, 2011.
- [5] J. L. Kuo, C. H. Shih, and Y. C. Chen, "Performance analysis of real-time streaming under TCP and UDP in VANET via OMNET," in *Proceedings of the 13th International Conference on ITS Telecommunications (ITST)*, pp. 116–121, Tampere, Finland, November 2013.
- [6] D. Broyles and A. Jabbar, "Design and analysis of a 3-D Gauss-Markov model for highly dynamic airborne networks," in *Proceedings of the International Telemetry Conference*, International Foundation for Telemetry, San Diego, CA, USA, October 2010.
- [7] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent Development and Applications of SUMO-Simulation of Urban Mobility," *International Journal on Advances in Systems and Measurements*, vol. 5, no. 3, pp. 128–138, 2012.
- [8] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. V. Stryk, "Comprehensive simulation of quadrotor UAVs using ROS and gazebo," in *Proceedings of the International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, pp. 400–411, Springer, Tsukuba, Japan, November 2012.
- [9] M. Quigley, K. Conley, B. Gerkey et al., "ROS: an open-source robot operating system," in *Proceedings of the ICRA Workshop on Open Source Software*, vol. 3, no. 3, Kobe, Japan, May 2009.
- [10] G. He, *Destination-Sequenced Distance Vector (DSDV) Protocol*, Networking Laboratory, Helsinki University of Technology, Espoo, Finland, 2002.
- [11] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," Report No. RFC 3626, IETF, Fremont, CA, USA, 2003.
- [12] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: the dynamic source routing protocol for multi-hop wireless Ad Hoc networks," *Ad Hoc Networking*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.
- [13] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," Report No. RFC 3561, IETF, Fremont, CA, USA, 2003.
- [14] J. Zhao and G. Cao, "VADD: Vehicle-assisted data delivery in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1910–1922, 2008.
- [15] J. K. Taek, M. Gerla, V. K. Varma, M. Barton, and T. R. Hsing, "Efficient flooding with passive clustering—an overhead-free selective forward mechanism for ad hoc/sensor networks," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1210–1220, 2003.
- [16] D. T. Ho and S. Shimamoto, "Highly reliable communication protocol for WSN-UAV system employing TDMA and PFS scheme," in *Proceedings of the GLOBECOM Workshops (GC Wkshps)*, pp. 1320–1324, Houston, TX, USA, December 2011.
- [17] P. Kinney, "Zigbee technology: wireless control that simply works," in *Proceedings of the Communications Design Conference*, vol. 2, pp. 1–7, San Jose, CA, USA, 2003.
- [18] K. G. Shin and P. Ramanathan, "Real-time computing: a new discipline of computer science and engineering," *Proceedings of the IEEE*, vol. 82, no. 1, pp. 6–24.
- [19] C. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard real-time environment," *Journal of the ACM*, vol. 20, no. 1, pp. 46–61, 1973.
- [20] K. Tindell and J. Clark, "Holistic schedulability for distributed hard realtime systems," *Microprocessing and Microprogramming*, vol. 40, no. 2-3, pp. 117–134, 1994.
- [21] A. Savvides, C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in Ad-Hoc networks of sensors," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 166–179, Rome, Italy, July 2001.
- [22] K. F. Ssu, C. H. Ou, and H. C. Jiau, "Localization with mobile anchor points in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 3, pp. 1187–1197, 2005.



Hindawi

Submit your manuscripts at
www.hindawi.com

