Dynamic Topology Reconstruction Protocol for UAV Swarm Networking

Minsoo Park¹, SeungGwan Lee² and Sungwon Lee³,*

¹ Department of Computer Science and Engineering, Kyung Hee University, Gyeonggi-do 17104, Korea; ms0511@khu.ac.kr
² Humanitas College, Kyung Hee University, Gyeonggi-do 17104, Korea; leesg@khu.ac.kr
³ Department of Software Convergence, Kyung Hee University, Gyeonggi-do 17104, Korea
* Correspondence: drsungwon@khu.ac.kr; Tel.: +82-31-201-3841

Received: 14 May 2020; Accepted: 1 July 2020; Published: 3 July 2020

Abstract: Along with the fourth industrial revolution, the use of unmanned aerial vehicles (UAV) has grown very rapidly over the past decade. With this rapid growth, studies using UAVs are underway in various areas. UAVs are more economical and effective when utilizing several nodes rather than operating a single aircraft. In general, UAVs collect and transmit information to the control center (CC), and act on control commands from the CC. Communication between UAVs and the control center is usually achieved using modules such as radio frequency (RF), Bluetooth, Wireless Fidelity (Wi-Fi) and cellular. However, when multiple UAVs communicate directly with the CC, the limitations of communication technologies and problems with increasing nodes occur. To address these points, several studies have constructed an ad-hoc network of UAVs to address the limitations of Wi-Fi and Bluetooth communication range, or the high cost of cellular systems. However, previous studies have constructed fixed topology ad-hoc networks. These studies did not take into account the problem of changing network topology due to the rapid mobility and frequent formation changes of UAVs. Due to this, limits occurred such that UAVs moved only in the pre-built topology. In this paper, we propose a dynamic topology construction protocol for UAV swarms to address this problem. The main contents of this paper are as follows. We first look at the research and limitations of existing UAV communications, and propose a protocol to solve this problem. This paper proposes a protocol for UAV construction of ad-hoc networks when trying to perform missions using multiple UAVs, and also describes how the changed network topology is reconstructed when network topology changes due to changes in flight formation. Finally, we establish a situation and apply the proposed protocol, analyze the results and describe further required research.

Keywords: dynamical networks; network reconstruction; unmanned aerial vehicle; UAV swarm; ad-hoc network

1. Introduction

Along with the recent fourth industrial revolution, the use of unmanned aerial vehicles (UAVs), also known as drones, has grown very quickly over the past decade. UAVs were first developed for military purposes, and are now being used for civilian use in the fields of photography, leisure, agriculture and transportation [1]. The global UAV market in the private sector is expected to grow from an annual $5 billion in 2019 to $14.5 billion in 2028 [2].

As interest in UAVs is growing, research using them is underway in various fields, such as smart factories and smart farms [3–5]. For UAVs used in various fields, it is more economical and effective if several UAVs are used rather than operating a single aircraft [1,6,7]. Studies using UAVs have recently been conducted in various areas, such as sensing, transport, tracking, search and rescue [8].
In general, several UAVs collect and transmit information to the control center (CC). Some problems arise when UAVs communicate directly with the remote CC [9]. When all nodes communicate directly with the CC, an increasing node number may result in an increase in packets that the CC has to process, resulting in a large amount of overhead and a delay. Radio frequency (RF), Bluetooth, and Wireless Fidelity (Wi-Fi) are mainly used for CCs and UAV communication [10]. With these modules, if the UAV is out of coverage range, communication is not possible. To solve this problem [8], the limitation of communication distance is overcome by constructing a UAV internal network and applying a hierarchical tree topology. However, existing studies do not consider formation changes during group flights of UAVs that must operate in a limited range, based on the communication limit of adjacent nodes, according to the initially configured topology.

Therefore, this paper proposes the dynamic topology reconstruction protocol of the UAV swarm network to solve the above problems. The problem that the formation change is difficult as the topology is fixed can be solved by dynamically reconstructing the topology of the UAV swarm. In addition, since the formation can be changed freely, the communication distance limitation between the CC and the UAV can be solved. This paper proposes a protocol for how multiple UAVs construct ad-hoc networks when trying to perform missions. The main contents of this paper are as follows. First, Section 2 discusses communication for UAV swarms and existing studies about UAV swarms, and introduces the limitations of existing methods. It also provides an overview of the mobile ad-hoc network, known as MANET. In Section 3, we introduce the concept of the protocol proposed in this paper, and describe how the network topology is formed. After describing the algorithms used, the format of messages used in the protocol is introduced. Section 4 describes the simulated environment, and shows the process of constructing a network topology by applying the proposed protocol through experiments, and analyzes and discusses the results.

2. Related Works

This section introduces the concepts and issues involved in UAV communication.

2.1. UAV Wireless Communications

UAVs are divided into two main types [10]. The first is the type in which a person is directly responsible for remote control of the UAV, and the other is autonomously flying UAVs. At this time, UAVs send flight data, such as speed, altitude and coordinates, as well as video data from the camera, to the CC. Communication between the control center and the UAV or between UAVs is important when several UAVs fly together. Wireless communication methods used for UAV control amongst civilians include Bluetooth, Wi-Fi and cellular [11]. Bluetooth is a near field communication (NFC) standard that uses the Industrial Scientific and Medical (ISM) band of 2400 to 2483 MHz. As interference results in low-power communication, the advantage of Bluetooth is that it is suitable for the control of UAVs with limited data communication. Furthermore, UAV battery limitations can be addressed [12]. On the other hand, transmission speed is slower than that of other telecommunication systems, and the transmission of high-capacity data is difficult. Wi-Fi is a wireless technology that is applied to High-Fidelity (Hi-Fi) and wireless local area network (LAN)s, which are wired networks. The Institute of Electrical and Electronics Engineers (IEEE) has established a wireless LAN standard of 802.11. While Wi-Fi has the advantage of being able to transmit high-capacity data at high speed, it is not suitable for long-distance communication due to its limited transmission power [13]. A cellular system is a technology in which a base station divides a service area into areas called cells, and is responsible for communication. Because the network is installed in advance, the advantages of high transmission speed and the assurance of good communication quality are advantageous for the communication of highly mobile objects. However, the use of cellular systems is highly restricted, as they are charged with communication, and there is no network in the air [14]. Table 1 compares the advantages and disadvantages of the types of communication.
When operating multiple UAVs together, the shortcomings of UAV communication described above become more important. Because Wi-Fi limits transmission and reception distance due to limited output, it is difficult to operate UAVs at great distances. Bluetooth has difficulty transmitting large data, such as image data. In addition, in cellular systems, there are no problems with the transmission of high-capacity data or restrictions on transmission distances. However, it is impossible to use cellular systems when ground networks collapse, or when it is difficult to obtain information from base stations. The configuration of the ad-hoc network of UAVs provides a solution to the above shortcomings [15]. Therefore, this paper addresses these shortcomings by forming an internal network with an ad-hoc network, and communicating with only one master node to the CC.

<table>
<thead>
<tr>
<th>Communications</th>
<th>Advantage</th>
<th>Disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bluetooth</td>
<td>Low power communication</td>
<td>Low transmission speed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Difficult to transfer high capacity</td>
</tr>
<tr>
<td>Wi-Fi</td>
<td>High transmission speed</td>
<td>High capacity data transfer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High transmission speed</td>
</tr>
<tr>
<td>Cellular system</td>
<td>High capacity data transfer</td>
<td>High cost</td>
</tr>
</tbody>
</table>

2.2. UAV Swarm Research

This section introduces existing studies on UAV swarm networks conducted on various topics. The main areas of study are as follows.

First, a study on the communication of UAV swarms. In [6], the overall Wireless Mesh Network was overviewed, and compared communication range and latency between routing and flooding techniques of UAV swarms. Next is a study on transmitting data using UAV relays over a distance where communication with each other is impossible. [16] introduced a method of relaying data sent from a base station to a destination by floating several UAVs. Lastly, this research was conducted by building an ad-hoc network using UAVs. Neeraj Chhabra, et al. [10] applied the hierarchical tree topology to the drone network, and then compared a single UAV and a UAV swarm tree topology. This study does not cover cases where the initially set topology was changed due to UAV mobility. Therefore, in this paper, the study is performed assuming the situation when the topology is changed.

2.3. Mobile Ad-Hoc Network

A mobile ad-hoc network is a network that is autonomously configured among mobile nodes without the aid of fixed networks [17]. In traditional ad-hoc network routing protocol, when nodes with greater mobility than fixed-node networks form networks, the problem arises in which the routing tables continue to change as nodes move. To solve these problems, a mobile ad-hoc network known as MANET is used. MANET’s protocols can be largely divided into two types: proactive (table-drive); and reactive (on-demand) [18].

The pro-active routing protocol, also known as table-driven, is the way all nodes in the network maintain the routing information of the other nodes [19]. Because the topology of the MANET is dynamic, these routing tables are updated whenever the network topology changes [20]. Maintaining routing information for all nodes is suitable for low latency and a low number of nodes. However, because the items in the table have to maintain path information to other nodes, the routing table can become too large, so it is not suitable for large networks. Typical pro-active protocols include DSDV (destination sequenced distance vector routing protocol) and GSR (global state routing) [21].

The reactive routing protocol, also known as on-demand, only performs the routing procedure when a request for the path occurs [22]. This is suitable for mobile ad-hoc networks, but each path setting requires a search of the entire network, making it inappropriate for real-time communication
due to long delays and the generation of a lot of control traffic. Key reactive protocols include DSR (dynamic source routing protocol) and AODV (ad-hoc on-demand vector routing protocol) [23].

3. Proposal

This section introduces the dynamic topology reconstruction protocol (DTRP) proposed in this paper.

3.1. Dynamic Topology Reconstruction Protocol

3.1.1. Goal of DTRP

The goals of the DTRP proposed in this paper are as follows. The objective is to see real-time network topology changes due to the large mobility of UAVs flying in clusters, and to reconstruct those topologies over the internal ad-hoc network. Figure 1 schematizes the network topology of the UAV swarm.

![Figure 1. Unmanned aerial vehicle (UAV) network topology.](image)

3.1.2. Architecture and Components of the DTRP

There are three main components of DTRP: the CC, which controls the entire swarm and manages the network; the master node (MN), which controls the internal network of the swarm in the air and the nodes of the CC; and the slave node (SN), which is responsible for the transmission of sensing and flight information. The architecture of the entire protocol is as follows in Figure 2. In addition, each node has its own node ID, which distinguishes it.

![Figure 2. Architecture of the proposed idea.](image)
3.2. Topology Reconstruction Using DTRP

This section discusses the topology reconstruction process of the DTRP.

3.2.1. The Entire Process

The entire process of the DTRP is as follows. The most basic process is the base topology reconstruction process, which describes how to construct a network topology. The first piece of the entire process is power on registration, which constructs the initial network topology and operates the initial settings after the operation of the swarm starts. Then, there is the periodic check operation process that checks whether a formation change in the entire network has occurred. In this process, we check if the formation of the network has changed, and if so, proceed to the topology reconstruction process. The third process is the control commands offer & operation process, which controls the operation of the swarm. This process sends control commands from the CC to the SNs in the entire network via the MN, and performs those commands.

3.2.2. Base Topology Reconstruction Process

This section describes the underlying DTRP topology reconstruction algorithm. This process is implemented to form the entire network of topology link tables. The initial link table generates the upper ID field as ‘-1’ and the link field as empty. Each node has its own link table. An example of the link table is shown in Figure 3. The link table shows which nodes are linked, and the linked node is divided using ‘,’ as well as square brackets, ‘[‘,’]’.

<table>
<thead>
<tr>
<th>Node ID</th>
<th>Upper ID</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>[1, [3], [4]], [2, [5], [6], [7], [8]]</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>[3], [4]</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>[5], [6], [7], [8]</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>[7], [8]</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 3.** Example of the link table.

The following describes the algorithms of the topology reconstruction process. First, the MN broadcasts the init message to the surrounding SN. The nodes that receive the init message check their link table, and store the sender’s node ID in their link table’s upper node ID if the upper node is ‘-1’.

Next, the SN that received the init message sends the acknowledge (ACK), which means that they received the init message from the MN. The parent node that receives the ACK considers the source ID of the message to be its link node, and stores it in the ACK list to confirm the end of the entire process.

After sending the ACK messages, the SN changes the source ID of the init message to its own node ID for linking with the nodes outside the communication coverage of the MN node, and then broadcasts the init message again. If each node does not receive the ACK within 10 seconds after the init message is broadcast, it considers itself the lowest node, and sends a link message to configure the link table.

A node that receives a link message considers the source ID of that message as its own sub-link, adds it to its link table using the add link algorithm summarized in Algorithm 1, and sends the entire link list back to the parent node.
Algorithm 1 Add Link Algorithm

Parameter \( t \): link table, \( l \): link of source node, \( s \): node ID of source
1: \( t += \left[ \right] \)
2: \( t += s \)
3: If \( l \) is not empty
4: \( t += \right\) \)
5: \( t += l \)
6: \( t += \left] \)
7: Return \( t \)

The MN checks the saved ACK list and receives a link message from all linked nodes, completes its link table and sends it to the CC. The following is Algorithm 2, based on the above rules.

Algorithm 2 Topology Construction Algorithm

1: MN broadcasts the init message
2: For each node
3: If upper ID is NULL then
4: store source ID to upper ID
5: Send ACK to upper node
6: Broadcast the init message
7: If node not receive ACK in 10 sec then,
8: send link message to upper node
9: If MN received all link message
10: Send link table to CC

3.2.3. Power on Registration

The power on registration operates in two processes before and after the flight. First, when the pre-flight process is executed, each node identifies its own node ID. The MN then checks the communication with the CC. Once all work is completed, the network topology is formed through the base topology reconstruction process to make communication possible.

3.2.4. Periodic Check Operation

The periodic check operation process is a task to verify that the entire link is maintained.

The MN sends a link check message to the SN that is linked to it once every three minutes. The SNs that receive the link check message deliver the ACK to the parent node, and send the link check message after changing the source ID field of the link check message to their linked node.

If the bottom node received the link check message, it transmits an ACK to the upper node. If the node received an ACK from all of its linked nodes, it sends an ACK to the upper node.

If one has not received the ACK of a particular node for one minute, re-send the link check message to that node three times every 10 seconds. If the ACK is not received after retransmission, put the ID of the error node in the error message, and send the error message to the parent node.

The master node waits until it receives an ACK from all its linked nodes, and proceeds with the topology reconstruction process if the check operation is not completed within three minutes.

The following is Algorithm 3, based on the above rules.
Algorithm 3 Link Check Operation Algorithm

1: MN sends the link check message
2: For each SN
3: If bottom node, then,
4: Send ACK to upper node
5: Else then,
6: Resend link check message to linked node
7: If ACK not received, then,
8: For three times every 10 sec
9: Resend link check message
10: If ACK received = false, then,
11: send error message
12: If ACK received from all linked node then,
13: Send ACK to upper node
14: If after three min, MN has not received ACK, then,
15: Do topology reconstruction process

3.2.5. Control Commands Offer & Operation

This section is a process that works when controlling the swarm. If there is a node that requires operation control, CC transmits the control offer message to the appropriate node through the MN. Nodes that receive the control offer message save the CC to the command list, and send an ACK. Each node then retransmits the offer message if it has not received the ACK within 10 seconds. When the ACK for the entire offer is received by the CC, the control operation message is sent over the MN. The node that receives the control operation message executes the CC that was saved. Then, we check the network topology change through the link check operation process, and perform the topology reconstruction process if the swarm formation has changed.

3.3. Message Format for a DTRP

This section discusses the format of messages used in each of the processes in the DTRP. Each message has a message type that is used to identify the messages.

3.3.1. Init Message

The init message is used during the topology reconstruction process above, and is broadcast so only the source ID field exists. Figure 4 shows the format of the init message.

![Figure 4. Init message format.](image)

3.3.2. Link Message

The link message is shown in Figure 5. With the init message, the link message is used in the topology reconstruction process, and has the payload of that linked node’s link table. As shown in Figure 6, transport messages with link tables are stored as a string type.
3.3.3. ACK

Figure 7 shows the format for an ACK. An ACK is used as a response to a message. No separate payload exists, as it is just a message for processing responses.

3.3.4. Link Check Message

The link check message, shown in Figure 8, is the message used in the link check operation process. The source ID field distinguishes senders to receive an ACK and the destination ID field for transfer of the entire data are required, and no separate payload exists.

3.3.5. Error Message

The error message is used for checking errors in the link check operation. As can be seen in Figure 9, there is an error node ID field that stores the ID of the node that did not send an ACK.
3.3.6. Control Offer & Operation Message

Figures 10 and 11 show the format of the control message to control the motion of the swarm. The command length of the control offer message is a field to indicate the total length of commands stored in the CC field, and the CC field stores commands to control the behavior of the UAV. Figure 12 shows an example of the control offer message.

![Control offer message format](image1)

**Figure 10. Control offer message format.**

![Control operation message](image2)

**Figure 11. Control operation message.**

![Control offer message example](image3)

**Figure 12. Control offer message example.**

4. Results

In this section, it is shown how the network topology is formed during a UAV swarm flight using DTRP.

4.1. Simulation Environment

Experiments on the operation of the UAV swarm are carried out in a simulation, as it is difficult to implement in real-life situations. The simulation program used in this experiment was Veins (Vehicles in Network Simulation). This software is a vehicle network simulation based on OMNeT++ and SUMO (Simulation of Urban Mobility). Veins is an open source framework that links the traffic simulator SUMO and the network simulator OMNeT++ [24].

To specify the formation of UAVs, virtual zones in a grid form were established [25]. The nodes on the Veins run along the roads in the zone. Figure 13 shows the shape of the virtual zone used in the experiment.

The variables used in the experiment are shown in Table 2. Each side of the grid is 100 m long, and the grid has a total of 10 * 10 sections. One side of the total area is 1000 m long, and each node travels and communicates on that grid. For the experiment, communication between the master node and the CC did not set a distance limit, and communication between nodes, including the master node, limited the distance to a maximum of 150 m.
4.2. Testbed Scenario

This section shows the test bed scenario and its results. For the test, UAVs performed two-dimensional simulations without considering the altitude. The test obtains base topology by running the network topology construction on the base size of the UAV. Then, we move the UAV to a suitable location for the test case, and check if the desired type of topology is completed. In the experimental process, the distance between the surrounding nodes was set to the maximum communication range of 150 m, enabling only communication with nearby nodes. In addition, the change of MN was not considered in this experiment, and the initial set MN will act as the master until the end of the experiment.

4.2.1. Base Case

The base case was tested with five nodes. The expected values before the experiment are shown in Table 3. The resulting values after the experiment are as follows in Figure 14. Figure 15 is an experimental screenshot, and Figure 16 is a network topology based on the result values.

Table 3. Expected result of base case.

<table>
<thead>
<tr>
<th>Number of Nodes</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expected result</td>
<td>[1],[2],[3],[4]</td>
</tr>
</tbody>
</table>
4.2.2. Test Case 1

Test case 1 is the result of the topology construction after moving the node from the base case. The expected values before the experiment are shown in Table 4. The resulting values after the experiment are as follows in Figure 17. Figure 18 is an experimental screenshot and Figure 19 is a network topology based on the result values.
Table 4. Expected result of test case 1.

<table>
<thead>
<tr>
<th>Number of Nodes</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expected result</td>
<td>[1,[3]], [2,[4]]</td>
</tr>
</tbody>
</table>

Figure 17. Result of test case 1.

Figure 18. Simulation screenshot of test case 1.

Figure 19. Network topology of test case 1.

4.2.3. Test Case 2

Test case 2 was tested by adding four more nodes to the base case. The expected values before the experiment are shown in Table 5. The result values shown after the experiment are as follows in Figure 20. Figure 21 is an experimental screenshot, and Figure 22 is a network topology based on the result values.

Table 5. Expected result of test case 2.

<table>
<thead>
<tr>
<th>Number of Nodes</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expected result</td>
<td>[1,[5]], [3,[6]], [2,[7]], [4,[8]]</td>
</tr>
</tbody>
</table>
Figure 20. Result of test case 2.

Figure 21. Simulation screenshot of test case 2.
4.2.4. Total Number of Messages per Case

The following is the number of total messages for each case. For the base case, the number of init messages is four, and the number of ACK messages is four. There are also four link messages for the entire link table formation, and therefore the total number of messages is 12. For test case 1, the number of init messages is four and the number of ACKs is four. The number of link messages is also four, for a total of 12 messages. For test case 2, the number of init messages is eight and the number of ACKs is eight. The number of link message is eight and the total number of messages is 24. Figure 23 is a graph of the number of messages per case.

4.3. Comparison Between the Central Control System and the Proposed System

This section attempts to compare the number of control message transmissions in a centralized system, in which the CC communicates directly with the UAVs, with the system proposed in this paper. When delivering control commands to the entire swarm, for a centralized system, the number of messages that the CC processes is proportional to the number of nodes. In contrast, for the proposed system, the number of messages processed by the CC is one, regardless of the number of nodes, since the MN transmits all messages over the entire swarm. The following graph in Figure 24 shows the
number of messages processed by the CC of the centralized system and the proposed system, based on the number of nodes.

![Comparison between Central and Proposed](image)

**Figure 24.** Comparison between central and proposed.

5. Conclusions

With the recent fourth industrial revolution, UAVs are being used in a variety of fields, and demand is expected to grow. UAVs are more economical and effective when utilizing several aircrafts than when using only one. However, excessive overhead and delay can occur, as UAVs communicate directly with the CC via modules such as RF, Bluetooth, and Wi-Fi. There is also a problem with the limitations of the communication range of UAVs resulting in limitations of distance.

To overcome this, several studies were conducted, including ad-hoc network and relay, among others. Studies of these internal network configurations have been able to overcome problems with overhead or delays caused by direct communication with the CC and limitations in communication distance. However, existing studies have limitations in which they operate within a limited scope with a fixed topology structure.

In this paper, we proposed a protocol that dynamically constructs a network topology to overcome the above limitations. However, the experiments in this paper were conducted in a simulated environment, rather than a real environment. In real-world UAV communication, limitations exist, such as battery efficiency problems in UAVs and interference problems in radio wave communication in nodes. When conducting an experiment in a real environment, it may be difficult to obtain accurate experiment results due to the communication interference problem. However, the experiment was conducted in a simulation environment, and in future research, we would like to solve problems that occur when experimenting in a real environment.

In the exploding field of UAVs during the Fourth Industrial Revolution, this study focused only on the construction of the network, but could lead to more advanced research if studied in combination with the physical problems that affect UAV communication.

**Author Contributions:** Conceptualization, M.P. and S.L. (Sungwon Lee); methodology, M.P. and S.L. (Sungwon Lee); software, M.P. and S.L. (Sungwon Lee); writing—original draft preparation, M.P.; writing—review and editing, S.L. (SeungGwan Lee) and S.L. (Sungwon Lee); supervision, S.L. (SeungGwan Lee) and S.L. (Sungwon Lee). All the authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.
References


