International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 13, Number 2, pages 216–241, July 2023

Delivery Routing to Reduce Calculation Load of Drones on Divided Logistics Areas for Drone Logistics Networks

Kazuya Matsutani
Graduate School of Science and Technology, University of Tsukuba
Tsukuba, Ibaraki, 305-8573, Japan

and

Shigetomo Kimura
Faculty of Engineering, Information and Systems, University of Tsukuba
Tsukuba, Ibaraki, 305-8573, Japan

Received: February 15, 2023
Revised: May 5, 2023
Accepted: May 26, 2023
Communicated by Ryo Nakamura

Abstract

In a drone logistics system, a message delivery system in which the drone delivers messages for other users on the way to a parcel delivery destination has been proposed. To reduce the complexity of message delivery routes, this paper proposes a message delivery method that divides logistics areas and determines the message delivery routes in each area. The method also makes it possible for a later departure drone and an early departure drone to exchange information to add, cancel, and/or exchange their messages and delivery points. The simulation experiments show that compared with the previous method, the proposed method has lower computational complexity and in contrast, the average travel distance increases by a maximum of 53.6% to 77.9% and so forth.

Keywords: Message Routing Method, Divided Logistics Areas, Drone Logistics Networks, DTN

1 Introduction

In recent years, unmanned aerial vehicles (UAVs), such as drones, have attracted attention as Internet of Things (IoT) devices. In particular, a number of companies are attempting to build parcel delivery systems using drones, which automatically deliver products ordered via the Internet or some other network. For example, in the U.S., Amazon is already providing a practical service [1]. In Japan, several companies, including Japan Post, are conducting practical experiments to implement such a service by 2023 [2].

In these drone logistic networks, in addition to their primary function of parcel delivery, various systems have been proposed that allow each drone to send messages to users to provide various services while on the way to the delivery points. In a drone network, since the nodes move rapidly, routing control needs to differ from that for the ordinary Internet. For drone logistic networks, Iranmanesh et al. proposed a weighted flight path planning (WFPP) algorithm [3] for optimizing the flight paths of drones for message transmission.

When one drone comes into close proximity with another drone, this algorithm redistributes the remaining message delivery points of the two drones to optimize their flight paths. However, the problem is that these drones consume large computational resources. When the density of drones is low, since each drone may not come into contact with another drone, the algorithm may not work effectively.

To solve this problem, this paper incorporates a deterministic relay forwarding method into WFPP and then proposes a method that divides up the parcel delivery area to determine message delivery points in each divided area [4] [5].

The proposed method makes it possible for a later departure drone to exchange information with an early departure drone to add, delete, or exchange their remaining message delivery points within the current divided area. Finally, simulation experiments show that it is possible to estimate that the average delivery distance in the proposed method is almost the same as for the previous method, in spite of the fact that the proposed method requires less computation. The average delivery distance is reduced by 0.7 % to 3.5 % when there are many message delivery points and these points are exchanged between the early and later drones.

The remainder of this paper is organized as follows. Section 2 describes a delay tolerant network (DTN), where since nodes such as drones move at high speed, the network paths are not fixed. Then, this section explains the WFPP algorithm, which is the target of this paper, in order to solve the problem of the high computational load for each drone in the WFPP. Section 4 proposes a message routing method based on divided logistics areas for drone logistics networks. Section 5 describes simulation experiments to verify the effectiveness of the proposed method. Finally, Section 7 concludes the paper and describes future research work.

2 Delay Tolerant Network (DTN)

As shown in Figure 1, when nodes move around at high speed and the connections between their nodes are intermittently interrupted, the network is called a delay tolerant network (DTN). A delay tolerant network is also sometimes referred to as disruption or disconnection tolerant network. The drone logistics network in this paper is also included in a DTN because it is difficult to establish permanent links between nodes. This section describes the communication methods commonly used in a DTN.

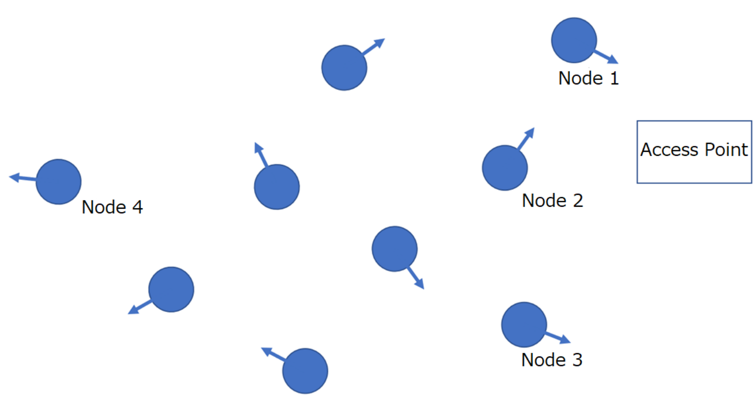

Figure 1: Overview of DTN.

In Figure 1, the blue circles indicate nodes (e.g. drones) and the arrows indicate the direction of the nodes' movements. An access point is a fixed communication base, such as a parcel collection point, or can be a rendezvous point for all nodes, but cannot be assumed to exist. In this figure, Node 1 and Node 2 are sufficiently close to be able to communicate directly with each other. Nodes 3 and 4, however, are moving in almost opposite directions and are not expected to approach to each other closely. It is too difficult to determine which node can relay messages between Nodes 3 and 4, since the direction of movement of all nodes varies.

This type of network significantly differs from the ordinary Internet, which is built on the assumption that links are rarely disconnected. Since existing TCP/IP-based network protocols cannot be used as they are, new protocols are required. Routing protocols for DTNs are classified into two types, i.e., deterministic relay forwarding and probabilistic relay forwarding [6]. The former deterministic relay forwarding scheme "is an approach in which an end-to-end route is determined in advance and relaying is performed" [6]. In particular, when the destination point of a receiver node is known or can be estimated, the sender node is programmed to move to some point where it can communicate on the way to the destination point.

The latter probabilistic relay forwarding scheme "is an approach in which relay forwarding is performed while determining the next relay node on a hop-by-hop basis without determining the end-to-end route in advance" [6]. This method is used when any node may incidentally come into contact with the receiver node, since the destination point of all the other nodes cannot be predicted.

An epidemic routing [7] method is a typical protocol for the probabilistic relay forwarding scheme. In this protocol, whenever two or more nodes stay within communication range where they can communicate with each other, they always copy all own messages to the other nodes. By repeating this process, messages spread throughout the network like an epidemic.

However, since the routing result differs depending on which paths the nodes traverse, we can say that the behavior is stochastically variable. Therefore, in epidemic routing, avoiding copying messages to unnecessary nodes is a challenging problem. For example, Pi et al. proposed reputation-based distributed routing (RBDR) algorithm [8] where each node is assigned a reputation and only nodes with low reputations can copy to nodes with high reputations. Since a high reputation node cannot send messages to low reputation node, the message cannot be sent from the high reputation nodes when these nodes congregate. To solve this problem, Matsutani et al. introduced adjusting nodes to improve the method [9].

3 WFPP

On the other hand, Iranmanesh et al. proposed weighted flight path planning (WFPP) [3] for the probabilistic relay forwarding method that adopts an algorithm to optimize the flight paths of drones. In this method, when two drones communicate to redistribute their message delivery points, each drone finds an approximate solution with high accuracy to the traveling salesman problem (TSP) by applying Christofides' algorithm [10] to obtain their new flight paths.

3.1 Algorithm of WFPP

In WFPP, two drones n_1, n_2 establish their connection when each one enters the other's communication range. The delivery order of all the remaining message delivery points is reset. Then, over the connection, they exchange their remaining message delivery points according to the following procedure to determine their route after the exchange. Assume that n_1 executes the following in cooperate with n_2, and m_0 and d_1 are its origin (base) and parcel delivery destination, respectively. m'_0 and d_2 are n_2's origin (base) and parcel delivery destination, respectively.

- Input G: A set of all the remaining message delivery points of the both drones n_1, n_2.
- Output $M_1 = \{m_0, m_1, m_2, \ldots, m_q, m_0\}$: A list of the n_1's ordered message delivery points. Note that n_2 also outputs $M_2 = \{m'_0, m'_1, m'_2, \ldots, m'_r, m'_0\}$ that satisfies the following: $G \cup \{d_1, d_2\} = \{m_0, m_1, \ldots, m_q, m'_0, m'_1, \ldots, m'_r, \}$.

1. Initialize $M_1 = \{m_0, d_1, m_0\}$.

2. If G is empty, output M_1.

3. Otherwise, from the elements of set G, select the point $\aleph$ with the highest weight ω.

   The weight ω_i of a transit point m_i is calculated by the following equation,

   $\omega_i = \mu_i - (T_i + E_{send})$

   where $\mu_i (0 \le \mu_i \le 10)$ is the priority of each message (the higher the value, the higher the priority), T_i is TTL of the message, E_{send} is the power consumed by the node when sending the message.

4. Calculate the TSP algorithm for $M_1 \cup \{\aleph\}$ to solve the approximate *cost* of the shortest path through $\aleph$ and all transit points in M_1, where both of the start and end points are m_0 to finally return to the base m_0.

   n_2 also calculates the same TSP algorithm for $M_2 \cup \{\aleph\}$.

5. Then, send their *cost* with each other. If n_1's *cost* is smaller, then M_1 is updated to the result of the TSP algorithm as $M_1 = \{m_0, \ldots, \aleph, \ldots, m_0\}$, but M_2 is not. Otherwise, M_2 is updated.

   However, if this new *cost* causes that the n_1's total flight distance exceeds the maximum flight distance $L_{\max}$ or if n_1 is unable to deliver the message during the TTL, then M_2 is updated. But, n_2 also matches one of the above conditions, both n_1 and n_2 give up to visit to $\aleph$. That is why WFPP cannot ensure to visit all message delivery points.

6. Eliminate $\aleph$ from G as $G = G - \{\aleph\}$.

7. Go to 2.

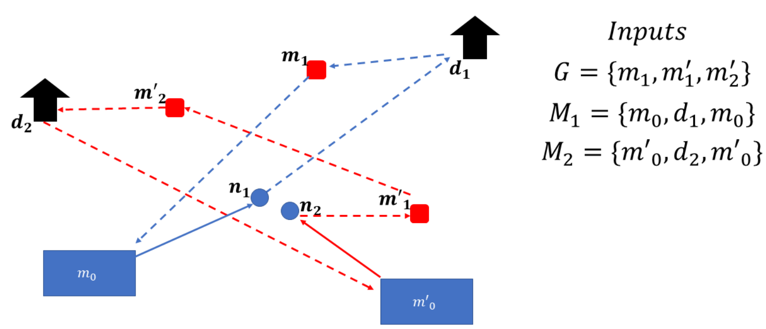

Figure 2: An Example of WFPP (STEP 1).

Figure 2 shows an example of WFPP's procedures. In this figure, two drones n_1 and n_2 established their connection, where n_1's remaining message delivery point is $\{m_1\}$ and n_2's ones are $\{m_1', m_2'\}$. After exchange their message delivery points, $G = \{m_1, m_1', m_2'\}$ and $M_1 = \{m_0, d_1, m_0\}$, $M_2 = \{m_0', d_2, m_0'\}$ at STEP 1 of WFPP. Since $G \neq \emptyset$, STEP 2 is skipped.

At first time of STEP 3, as in Figure 3, each node must select a point with the highest weight as $\aleph$. Suppose that m_1' is selected as in the figure. Then, at STEP 4, WFPP solves the TSP to obtain the shortest path among the message delivery point, i.e., n_1 solves the TSP for $M_1 = \{m_0, m_1', d_1, m_0\}$ and n_2 does for $M_2 = \{m_0', m_1', d_2, m_0'\}$. However, because the TSP is NP-hard, WFPP uses a TSP solver based on Christofides' algorithm to obtain a highly accurate approximate solution, which requires $O(n^3)$ where n is the number of message delivery points.At STEP 5, n_1 and n_2 exchange each result. Figure 3 supposes that n_1 won the competition, since the approximate solution for M_1 is shorter than that for M_2. Then, M_1 is updated as the TSP's result. At STEP 6, m'_1 is eliminated from G.

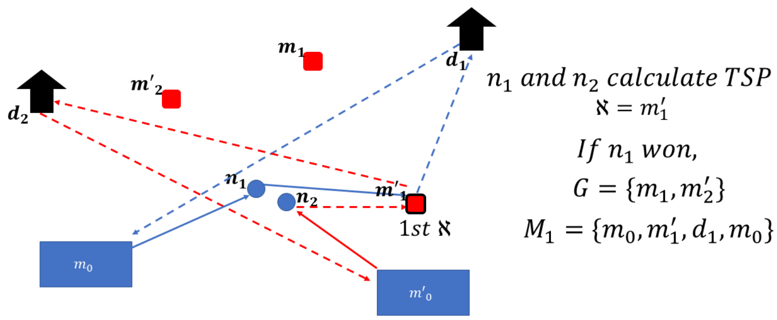

Figure 3: An Example of WFPP (First Time of STEPs 3–6).

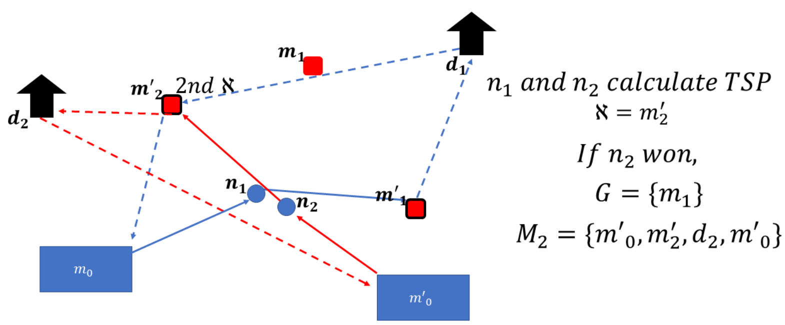

Figure 4: An Example of WFPP (Second Time of STEPs 3–6).

For the second iteration, as in Figure 4, m'_2 is selected and added to M_2. For the third iteration, as in Figure 5, m_1 is selected and also added to M_2.

As the above procedures, the calculation of the approximate solution from STEP 2 to STEP 6 is repeated $O(n^2)$ times, because of the combination of message delivery points to be exchanged. Thus, overall computational complexity in WFPP is $O(n^2) \times O(n^3) = O(n^5)$.

Since each drone must perform calculations every time the drone comes into contact with other drones, it is known that each drone consumes large computational resources. At the calculation time, each drone must stay at the same position, although they may have a heavy parcel and the weather condition is bad such as heavy rain or strong window. To contact to the other drones, each drone must periodically scan the other drones. When the density of drones is low, since there are few exchanges that each drone can make in contact with the other drones, the algorithm may not work effectively.

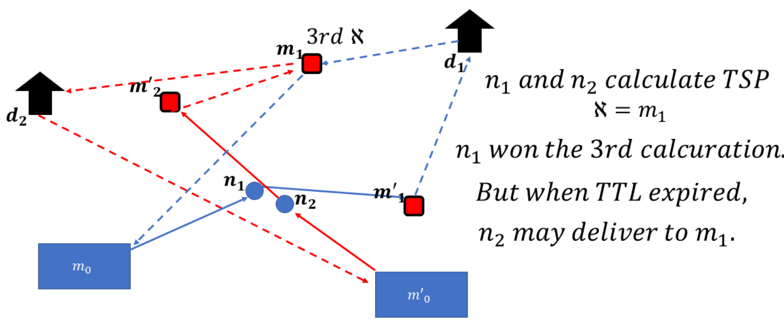

Figure 5: An Example of WFPP (Third Time of STEPs 3–6).

4 Proposed Method

In order to improve the WFPP problem, this section introduces the concept of a deterministic relay forwarding scheme for WFPP and proposes a message routing method to reduce delivery distance with smaller computational complexity.

4.1 Preconditions

Figure 6 shows an overview of a drone logistics network for the parcel and message delivery assumed in the proposed method.

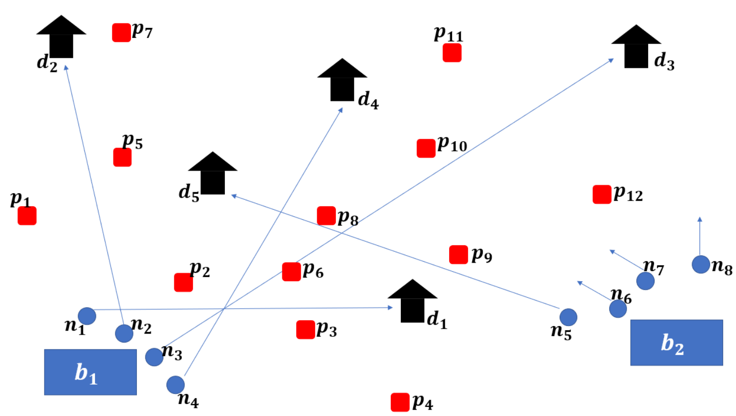

Figure 6: Overview Network Assumed in the Proposed Method.

- Let $N = \{n_1, n_2, \ldots, n_v\}$ be a set of v drones and $D = \{d_1, d_2, \ldots, d_v\}$ be a set of parcel delivery destinations. Each drone n_i $(1 \leq i \leq v)$ carries only one parcel and delivers it to the delivery destination d_i. That is, for one parcel delivery, a drone and its parcel delivery point correspond to one-to-one.
- A parcel warehouse for delivery is called a base. Let $B = \{b_1, b_2, \ldots, b_m\}$ $(m \leq v)$ be a set of bases. Each drone n_i belongs to only one base b_j $(1 \leq j \leq m)$ and delivers a parcel from b_j.

Multiple drones can belong to a single base. There is a one-to-many correspondence between bases and drones.

- Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of the message delivery points distributed throughout the logistics network. P is divided into v subsets $P_1, P_2, \ldots, P_v$ where any two different subsets have no common element, i.e., $P = P_1 \cup P_2 \cup \cdots \cup P_v$, and if $i \neq j$ then $P_i \cap P_j = \emptyset$. A drone n_i leaves from its own base b_j, visits all message delivery points belonging to P_i to send a message to the user at this point. After the drone has delivered all messages, it delivers a parcel to a parcel delivery destination d_i and returns to its own base.
- The flight path of each drone n_i is determined when the drone leaves the base. From the departure time, flight speed, and flight path of the drone, the point where the later drone can contact n_i can be determined.
- After the early drone n_i leaves, if an addition or cancellation of a message delivery is required in P, a drone departing after n_i will contact n_i to add or delete a message that n_i is scheduled to deliver after the contact time, and can even exchange their message delivery points. In this case, these drones must calculate the modified flight path by themselves.
- All drones are assumed to be able to fly on any path and be unaffected by geographical or topographical factors.

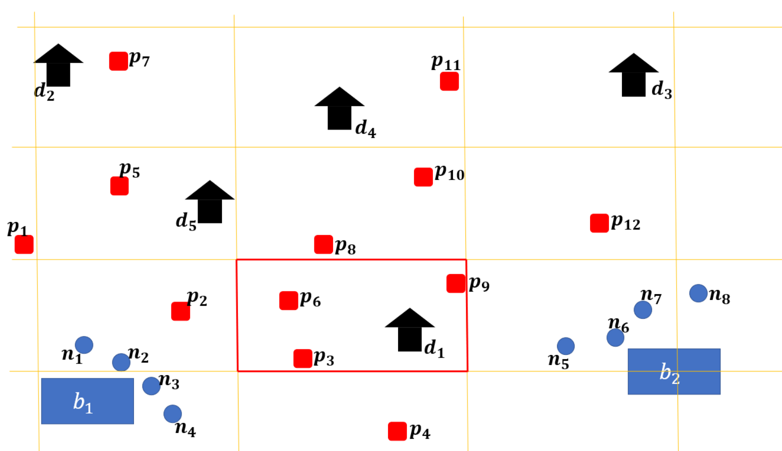

Figure 7: Dividing a Logistics Network.

As an example, in Figure 6, n_1 to n_8 are drones. n_1 to n_4 belong to base b_1 and n_5 to n_8 belong to base b_2. The final destination of each drone n_i is the parcel delivery destination point d_i, but d_6, d_7 and d_8 of n_6, n_7 and n_8 are omitted due to space constraints. The red points in the figure are message delivery points that at least one drone needs to visit once to deliver a message. The arrows in the figure show the flight paths that each drone will traverse if it goes straight from its own base to its final parcel delivery destination. This path is optimal if the drone does not visit any message delivery point.

4.2 Proposed Algorithm

The proposed algorithm involves distributing message delivery points, determining flight paths for message delivery points, and redistributing the delivery points between the early departure drone and the later departure drone. The main purpose of this algorithm is not to find the shortest flight path, but to reduce the computational complexity compared to the previous method. This is especially important when redistributing the points between drones, since the paths must be calculated by the drones.

4.2.1 Dividing the Logistics Network

First, a network manager divides the entire logistics network into a grid as shown in Figure 7. In the following, a square piece of the grid is referred to as a cell. Let C_i be the i-th cell and $P_{C_i}^j$ denote the j-th message delivery point in C_i. The cell size depends on the number of message delivery points, the number of drones, and the range of the entire network.

4.2.2 Distributing Message Delivery Points in each Cell among Drones

Before drones leave their bases, the manager connects a straight line for each drone n_i between the drone's base b_j and the parcel delivery destination point d_i. If the line passes through a cell C_k, the drone transits C_k and is responsible for delivering some messages in C_k. Among all drones that transit the same cell, all message delivery points in the cell are distributed according to the following procedure.

In Figure 8, Figure 9, and Figure 10, the 8 message delivery points $P_{C_i}^1, P_{C_i}^2, \ldots, P_{C_i}^8$ in a cell C_j are distributed to drones n_l, n_m, and n_n that are assumed to transit C_j in this order. The solid arrows in Figure 8 are straight lines connecting the drone's base and the parcel delivery destination point.

First, let $\#(P_{C_i})$ be the number of message delivery points in C_i and

$\#(n_{C_i})$ be the number of drones that transit C_i. If $N = \#(P_{C_i})/\#(n_{C_i})$ is divisible, each drone will visit N message delivery points. Otherwise, let R be the remainder of N and the first R drones will visit $\lfloor N \rfloor + 1$ message delivery points in the order of entering to the cell or $\lfloor N \rfloor$ for others.

In Figure 8, the $\#(P_{C_i}) = 8$ message delivery points are distributed among $\#(n_{C_i}) = 3$ drones. $N = 8/3$ is not divisible and the remainder of N is $R = 2$. Since n_l and n_m will enter the cell sooner than n_n, the first R drones n_l and n_m will visit $\lfloor N \rfloor + 1 = 3$ message delivery points and the final n_n will visit $\lfloor N \rfloor = 2$points.

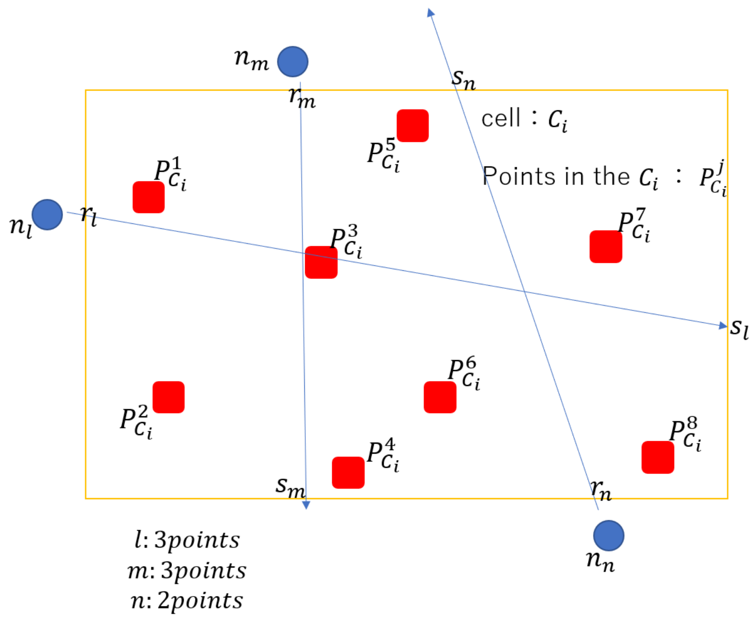

Figure 8: Counting Message Delivery Points that each Drone Visits.

Next, in the order of entering cell C_i, each drone selects its visit point one by one as follows to let the drone add its message delivery points between the entrance point r_i and the exit point s_i of C_i. The detail of the algorithm is as follow.

- Let r_i and s_i be the intersections of the cell boundary and the straight line between the base b_j of a drone n_i and the parcel delivery point d_i, where r_i is the entrance point of cell C_i and is nearer than the exit point s_i from b_j. In the order of entering the cell, each drone selects the nearest message delivery point from the intersection s_i. For example, in Figure 9, a black point is s_i $(i = l, m, n)$ when the dotted line is the corresponding straight line l_i between the base b_i and the parcel delivery destination point d_i. Since the order of entering C_i is n_l, n_m, and n_n, the first n_l selects $P_{C_i}^7$, then n_m selects $P_{C_i}^4$, and finally n_n selects $P_{C_i}^5$.

- For the remaining message delivery points, in the order of entering cell C_i, each drone selects the nearest message delivery point from the corresponding straight line l_i up to the number of n_i's message delivery points. Each drone can calculate the nearest point as follows. First, it shifts all the remaining message delivery points in parallel as r_i moves to the origin point. Then, it rotates all the points around r_i as s_i is on the positive region of the x-axis. Finally, the shifted and rotated point whose absolute value of y-coordinate is minimum is selected as the solution. For example, in Figure 10, n_l first selects $P_{C_i}^1$, and then $P_{C_i}^3$.

  The drone n_i will visit the message delivery points in the order of the smaller value of x-coordinate on the shifted and rotated coordinate. As a result, each drone follows a zigzag path across the corresponding straight line, as shown in Figure 10.

- After a drone n_i visits all selected message delivery points in cell C_i, it goes to the intersection s_i to enter the next cell. If a parcel delivery point d_i exists in C_i, it does not go to s_i, but delivers a parcel to d_i and goes back to base b_j along the corresponding line l_i.

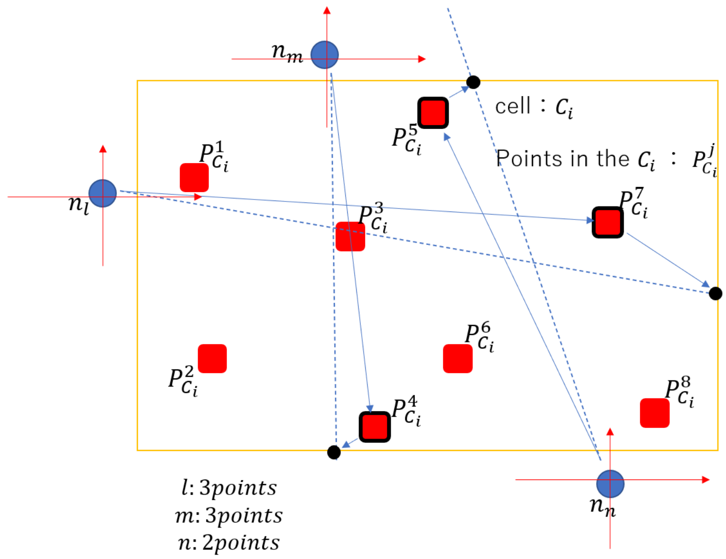

Figure 9: Determining the Last Point to Visit.

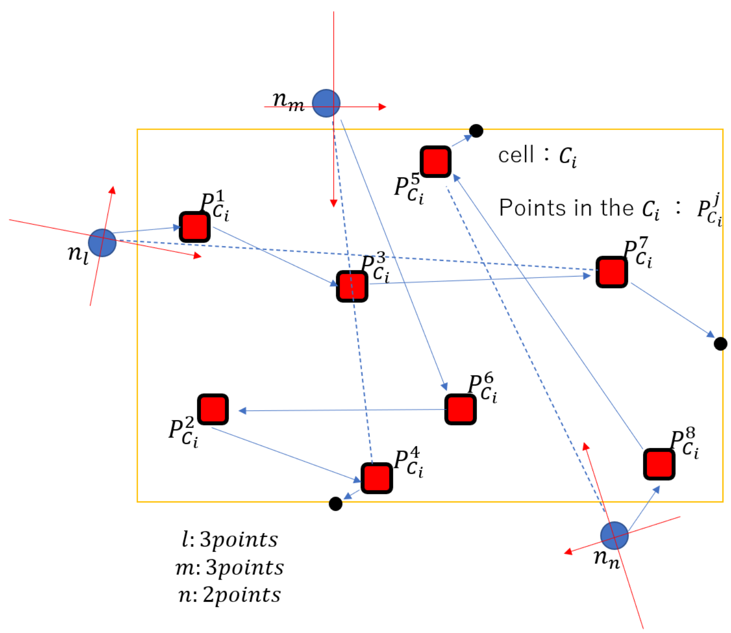

Figure 10: Determining the Remaining Points to Visit.

4.2.3 When a Later Departure Drone Contacts an Early Departure Drone

Suppose that a later departure drone n_m contacts an early departure drone n_l at a contact point c in the current cell C. These drones exchange their message delivery points that may reduce their total delivery distance. The details of the algorithm are as follow.

After c, n_l was scheduled to visit message delivery points $l_1, l_2, \ldots$ and n_m to visit $m_1, m_2, \ldots$. Let $c = l_0 = m_0$. The next point of each final message delivery point is the exit point from C, i.e., s_m and s_n.

Shift all message delivery points etc. in C in parallel as the entrance point r_l of n_l is the origin point. Then rotate them around r_l as the exit point s_l is on the positive region of the x-coordinate. When there exists a message delivery points m_j whose x-coordinate is between the x-coordinates of adjacent message delivery points l_i and l_{i+1}, these points may be exchanged as follows.

- If the y-coordinates of l_i and l_{i+1} are the same sign, or one or both are 0, and m_j is also the same sign or 0, then m_j becomes a message delivery point of n_l.
- If the signs of the y-coordinates of l_i and l_{i+1} differ, then unconditionally m_j becomes a message delivery point of n_l.
- Otherwise, m_j remains a message delivery point of n_m.

From the above procedure, if n_m has a message delivery point that does not intersect the straight line between the n_l's entrance point and exit point of the current cell, then the message delivery point becomes n_l's message delivery point. After all n_m's message delivery points are checked, the n_l's message delivery points are also checked by the above procedure and some are changed to n_m's if they satisfy the condition. Through this exchange, the proposed method may be realized without a high level of computational complexity by reducing n_l's and n_m's flight distances.

Figure 11 to Figure 14 are an example of exchanging message delivery points. In Figure 11, a later departure drone n_l contacts an early departure drone n_m.

After exchanging their message delivery points $\{l_1, l_2\}$ and $\{m_1, m_2\}$, n_l shifts and rotates all the message delivery points etc, as r_l is the origin point and the line between r_l and s_l matches to the x-coordinates matches as shown in Figure 12. Since the y-coordinates of l_1 and l_2 are the same (+), n_l deliveries these points. n_m also shifts and rotates the message delivery points etc., as shown in Figure 13. Since the signs of m_1 and m_2 differ, n_l deliveries m_1.

After the above exchanging, message deliver points are modified as shown in Figure 14.

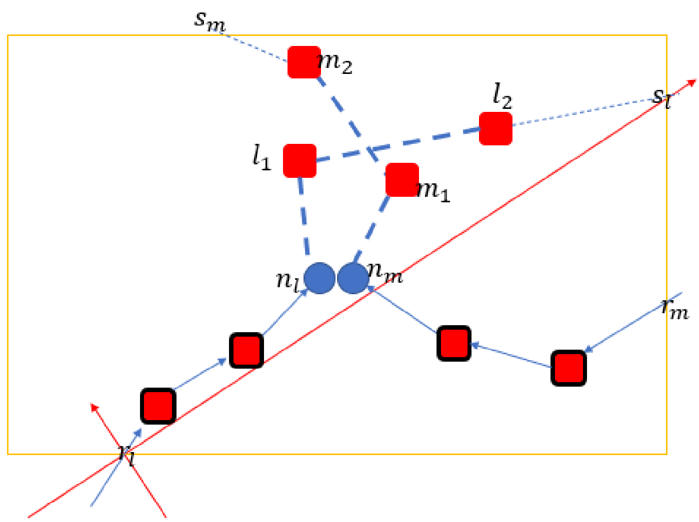

Figure 11: Message Delivery Route Before Exchange Message Delivery Points

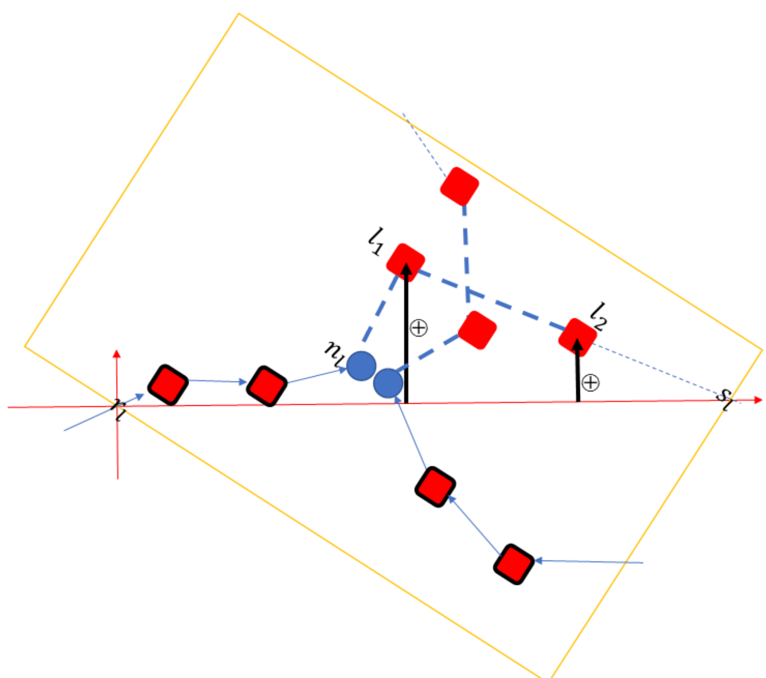

Figure 12: Shifted and Rotated Message Delivery Points for n_l

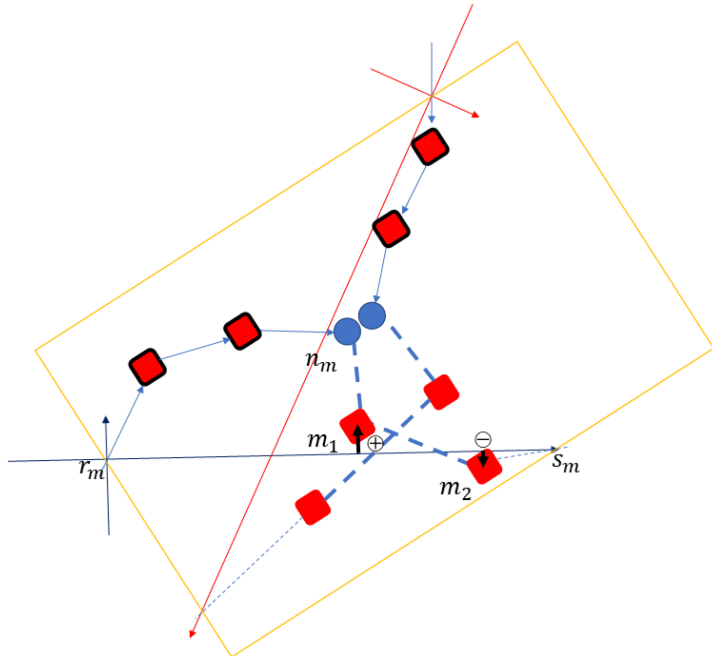

Figure 13: Shifted and Rotated Message Delivery Points for n_m

4.2.4 For Cells Not Traversed by Drones

From the algorithm in Section 4.2.2, if no drones transit a cell, then they also visit all message delivery points in the cell. For such message delivery points if any, the drone whose parcel delivery point is closest to their points visits after the drone delivers its parcel. This ensures 100% delivery rate because every message delivery point is visited at least once.

However, if the number of such cells is very large, this simple method is not expected to work well. In particular, we should consider the drone's remaining battery power to ensure that the drone can return to its own base.

4.3 Computational Complexity of Proposed Method

In the proposed algorithm, the computational complexity is high, when a cell contains all message delivery points. Let V be the number of drones and P the number of points. For the algorithm in Section 4.2.2, first, all drones are sorted in the order of shortest distance from their cell exit point. Then, in the next step, all message delivery points must be sorted in the order of shortest distance from the line connecting their entrance point and exit point. Since the calculation of $O(P \log P)$ is required for each drone, the overall computational complexity is $O(VP \log P)$.

Since each base can calculate the above algorithm, no computational complexity is incurred by each drone. However, as mentioned in Section 4.2.3, the early departure drone and the later departure drone need to perform calculations to exchange their information. Since all message delivery points are already sorted in the order of delivery, each drone needs $O(P)$ to exchange. In general, since the message delivery points are distributed into many cells, the number of message delivery points in one cell is limited and the computational complexity is lower.

On the other hand, in WFPP, each drone needs $O(P^5)$ to calculate TSP, each time the drone contacts the other drone. However, the approximation accuracy of TSP is better than that of the proposed algorithm. The next section will evaluate how much the proposed method increases the drone's flight distance in the case where drones only deliver parcels.

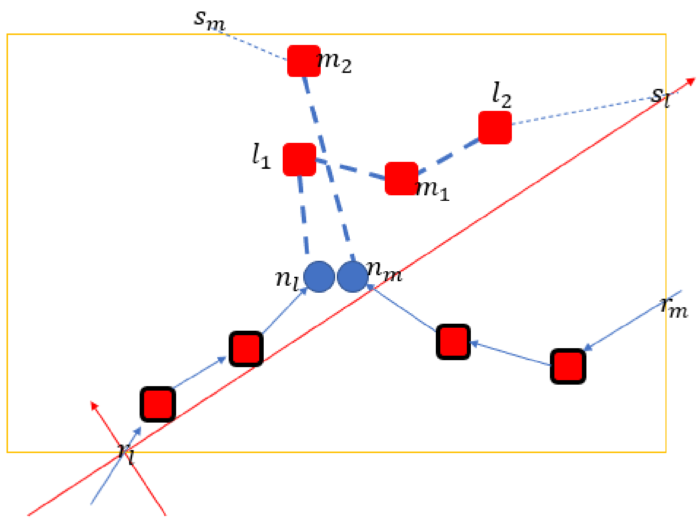

Figure 14: Message Delivery Route After Exchange Message Delivery Points

5 Simulation Experiments

In this section, we describe the following simulation experiments conducted using a Java program.

Particularly, in Experiment 4, we use The One Simulator [11] which is the network simulator for DTN environment.

5.1 Details of Simulations

Table 1 shows the specifications of the computers used in Experiments 1–3, and 5, and Table 2 shows those used in Experiment 4. Experiment 4 used a different computer than Experiments 1–3, and 5 because we conducted a larger network simulation.

Table 1: Computer using in Experiments 1–3, and 5

element	**value**
CPU	Intel Core i5-6200U CPU @ 2.30 GHz–2.40 GHz
RAM	8GB

Table 2: Computer using in Experiment 4

element	**value**
CPU	AMD Ryzen 7 3700X 8-Core Processor
RAM	16GB

5.2 Experiment 1

This experiment was conducted using the simulation parameters shown in Table 3, but the number of cells was fixed to 9 and the cell size was 3000 m $\times$ 3000 m.

Figure 15 shows the network topology divided into nine cells. The initial entry points of drone 1 (n_1) and drone 2 (n_2) were in cells C_0 and C_6, respectively. Their parcel delivery points were

Table 3: Simulation Parameters in Experiments 1–3, and 5

parameter	**value**
The number of drones	2
The range of entire network	9000m × 9000m
The number of cells (Experiments 1, 2, and 5)	9
The number of cells (Experiment 3)	9, 16, 25
The cell size (Experiments 1, 2, and 5)	3000m × 3000m
The cell size (Experiment 3)	3000m × 3000m, 2250m × 2250m, 1800m × 1800m,
The number of message delivery points (Experiments 1, 2, and 3)	5, 10, 15, 20
The number of message delivery points (Experiment 5)	20, 40, 60, 100

Table 4: Simulation Parameters in Experiment 4

parameter	**value**
The number of drones	20, 40, 60, 80, 100
The number of bases	1
The speed of drones	100Km/h
link speed	54Mbps
transmission range	70m
Buffer size of drones	50MB
Message TTL	5 hours
Message	Random between 500KB - 1MB
The range of entire network	9000m × 9000m
The number of cells	9
The cell size	3000m × 3000m
The number of message delivery points	20, 40, 60, 80, 100
The seed value of The One Simulator	35

d_1 in cell C_8 and d_2 in cell C_2, respectively. Their parcel delivery points were randomly assigned from cells C_2 and C_8. The message delivery points were not restricted to any particular cell, but were randomly placed within the entire network. Experiments 2 and 5 satisfy above the initial entry points and the message delivery points. Experiment 3 also satisfies only the condition of the message delivery points.

Two different deliveries were executed. In the first scenario, two drones delivered messages and each parcel in accordance with the proposed algorithm. In the second scenario, they do not exchange messages and only deliver on each parcel directly from their base to allow their average delivery distance to be compared.

5.3 Experiment 2

Same as Experiment 1, but the early depart drone (drone 1) and the later depart drone (drone 2) exchange their message delivery points in accordance with proposed algorithm.

5.4 Experiment 3

Same as Experiment 2, but the cell size is changed to evaluate the impact of the number of cells.

In Experiment 3, like 3 times 3 cells, we defined the top-left cell is cell numbers C_0, and increasing the number toward the bottom-right, the bottom-right cell is C_{15} in 4 times 4, and the bottom-right cell is C_{24} in 5 times 5.

In the case of 4 times 4 cells, The initial entry points of drone 1 (n_1) and drone 2 (n_2) were in cells C_0 and C_{12}, respectively. Their parcel delivery points were d_1 in cell C_{15} and d_2 in cell C_3, respectively.

In the case of 5 times 5 cells, The initial entry points of drone 1 (n_1) and drone 2 (n_2) were in cells C_0 and C_{20}, respectively. Their parcel delivery points were d_1 in cell C_{24} and d_2 in cell C_4, respectively.

5.5 Experiment 4

This experiment was conducted with the simulation parameters shown in Table 4 using The One Simulator. We compared the average latency when the number of message delivery points and the number of drones are changed. The latency is the difference between the time that the drones start to flight and the time that all messages are delivered to their destination nodes.

In this experiment, there are nine cells and only one base is in C_2. The parcel delivery points and the message delivery points were not restricted to any particular cell, but were randomly placed within the entire network.

5.6 Experiment 5

This experiment was conducted comparing proposed method and existing method WFPP, by average delivery distance and average actual computation time. The actual computation time is the time to calculate only the new flight path in each contacting drone after exchanging its message delivery points. The time does not include any other processes such as finding drones to contact.

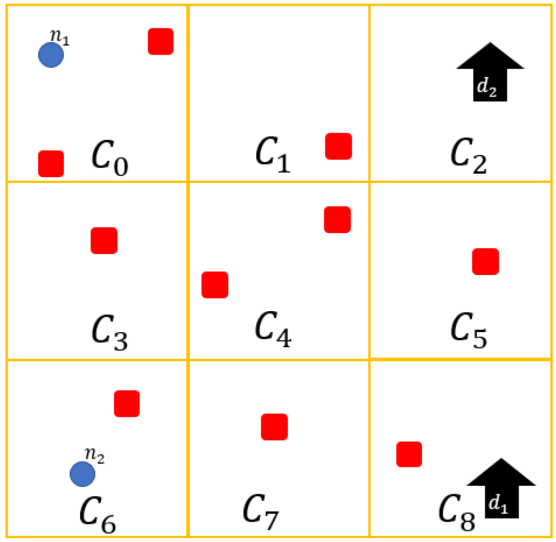

Figure 15: The Network Topology in the Experiments 1, 2, and 5.

6 Simulation Result

6.1 Simulation Result in Experiment 1

In Experiment 1, the delivery distance of each drone is measured 100 times for each number of message delivery points. Note that the distance is required where a drone delivers all messages (if these messages exist) and a parcel in the proposed algorithm from the drone's own base.

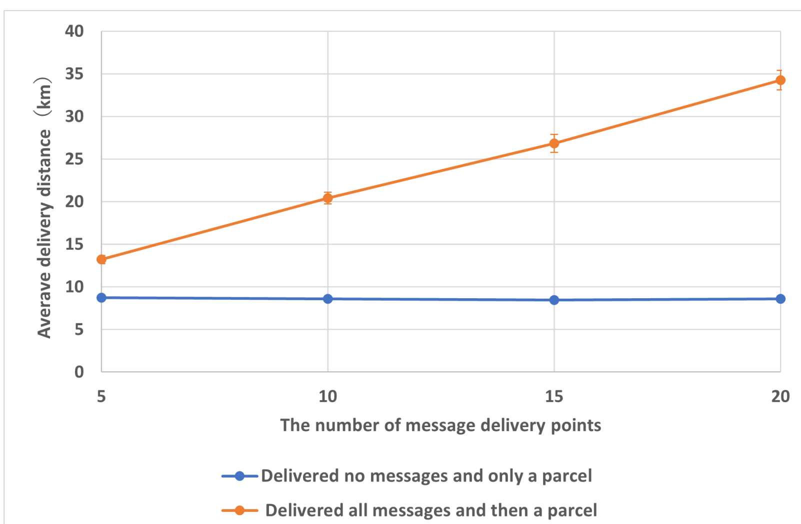

Figure 16: The Average Delivery Distance in Experiment 1.

Figure 16 shows the average delivery distance when only drone n_1 made deliveries. Figure 17 shows the average delivery distance when both drones n_1 and n_2 made deliveries. All the following graphs have error bars to indicate the standard error of each value. From Figure 16, when there is only one drone, the average delivery distance with the proposed method increases according to the number of message delivery points, and for 5, 10, 15, and 20 message delivery points, their average distances are 1.54, 2.44, 3.23, and 4.10 times greater than those when there is no message delivery, respectively.

As in Experiment 1, the delivery distance of each drone is measured 100 times for each number of message delivery points with two drones, but these drones do not exchange their message delivery points at this stage.

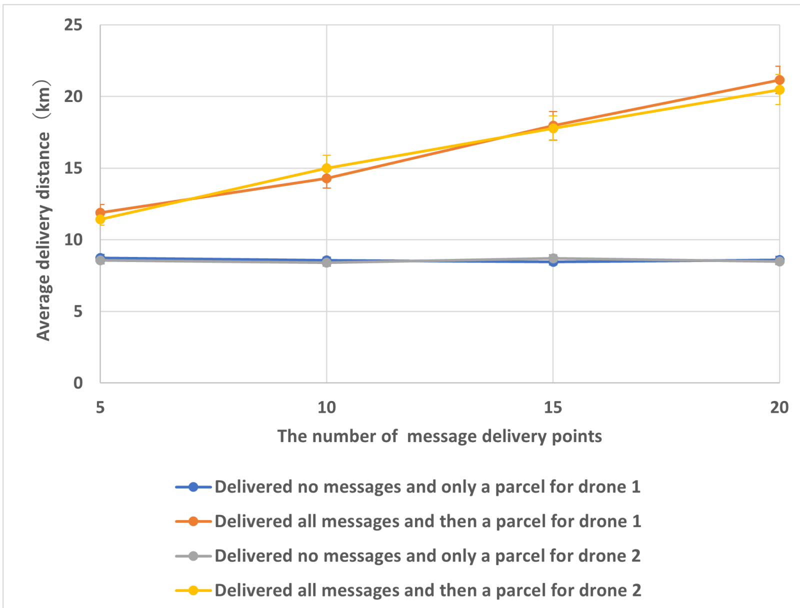

Figure 17: The average delivery distance without exchanging message delivery points for two drones in Experiment 1.

Figure 17 shows the average delivery distance. The average delivery distances with and without message deliveries are almost same in n_1 and n_2. As in Figure 16, the average delivery distance increases according to the number of message delivery points, but the average delivery distances with message deliveries for a drone are reduced compared to those in Figure 16 since the message delivery points are distributed to two drones.

6.2 Simulation Result in Experiment 2

Next, Experiment 2 examined the effect when two drones exchange their message delivery points. In fact, no message delivery points were exchanged in many of the trials. Therefore, the next experiment continues trials until message delivery points are exchanged in 10 trials.

Table 5: The number of trials to exchange message delivery points in 10 trials.

The number of message delivery points	**The number of trials**
5	190
10	143
15	74
20	90

Table 5 shows the results of the total number of trials for each number of message delivery points. From this table, the more message delivery points there are, the greater the rate at which message

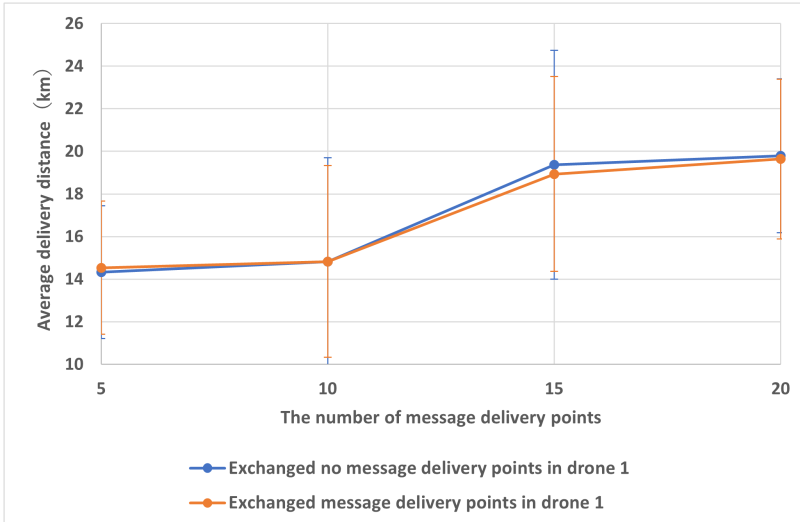

Figure 18: The average delivery distance of drone 1 with and without exchanging message delivery points in Experiment 2.

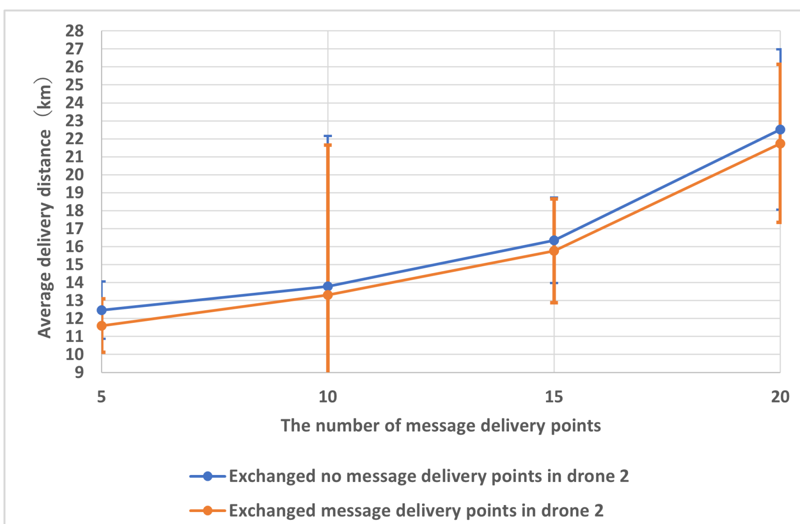

Figure 19: The average delivery distance of drone 2 with and without exchanging message delivery points in Experiment 2.

delivery points are exchanged. This is because that for many message delivery points, it tends to be easy to satisfy the conditions for the exchange of message delivery points and the contactee drone has more candidate message delivery points that may become message delivery points for the contactor drone and vice versa.

Figure 18 and Figure 19 show the average delivery distance and the confidence intervals at the 95% confidence level for the two drones for the case when they did not exchange any message delivery points and for the case when they did make exchanges.

These figures show that the average delivery distance is shortened by the exchange of message delivery points. In Figure 18, when the number of message delivery points is 20, the average delivery distance with exchange is about 0.7 % shorter than that without exchange. Figure 19 also shows that the average delivery distance with exchange is about 3.5 % shorter than that without exchange when the number of message delivery points is 20. However, many cases exist where the total delivery distance of the two drones increases due to the exchange of message delivery points. As the confidence interval indicated, there is great variability in the average delivery distance. From this perspective, WFPP has advantages since it calculates the approximate distance.

For the problem shown in Section 4.2.4, Table 6 shows the average, maximum, and minimum number of message delivery points that no drones visit. The average numbers in this table are not very large. However, since the maximum numbers indicate that there are multiple cells not transited by any drones, this problem cannot be ignored.

Table 6: The average, maximum, and minimum number of message delivery points that no drones visit.

The number of message delivery points	Average	Max	Min
5	0.2	1	0
10	0.7	4	0
15	0.25	1	0
20	1	5	0

Although the current simulation program does not implement, a drone whose parcel delivery point is closest to their points visits after the drone delivers its parcel, as mentioned in Section 4.2.4.

Another solution is to use more drones only for delivery messages without a parcel, when the base finds that there are cells not traversed from any drones after the flight path plan is made.

6.3 Simulation Result in Experiment 3

Experiment 3 was conducted to examine the effect of a smaller cell size for exchanging message delivery points. As shown in Table 3, 3×3, 4×4, and 5×5 cells are prepared for the grid in the same network range.

Figure 20 and Figure 21 show the average delivery distance of drone 1 and drone 2, respectively, when message delivery points are exchanged. These results show that the average delivery distance of 3×3 cells is greater than that of 4×4 and 5×5, however, those of 4×4 and 5×5 cells are almost the same. Thus, 4×4 is suitable for up to 20 message delivery points. Since the number of message delivery points in a cell decreases for smaller cell size, the calculation time to exchange message delivery points may also decrease. 5 × 5 cells may be suitable for more than 20 message delivery points.

In general, if the whole network area is divided into N equal-sized cells and M message delivery points are uniformly distributed, then the average message delivery point in each cell is M/N. When N is large, i.e., the cell size is small, the expected flight distance in a cell is short, but each drone may traverse many cells. Therefore, some minimum flight distance in a whole network seems to be obtained, but it depends on parcel delivery points for the proposed algorithm. If the parcel delivery point for a drone is near the base, then the drone cannot pass through cells that are far from the

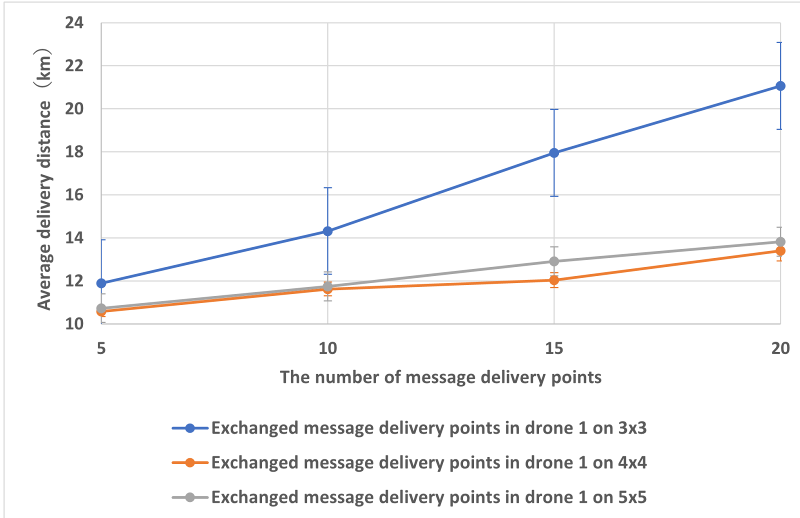

Figure 20: The average delivery distance of drone 1 with exchanging message delivery points for different cell size in Experiment 3.

base. Moreover, if the cell size is too small, many cells may not be traversed any drones. The proposed algorithm should be enhanced to ensure that all cells are passed by at least one drone.

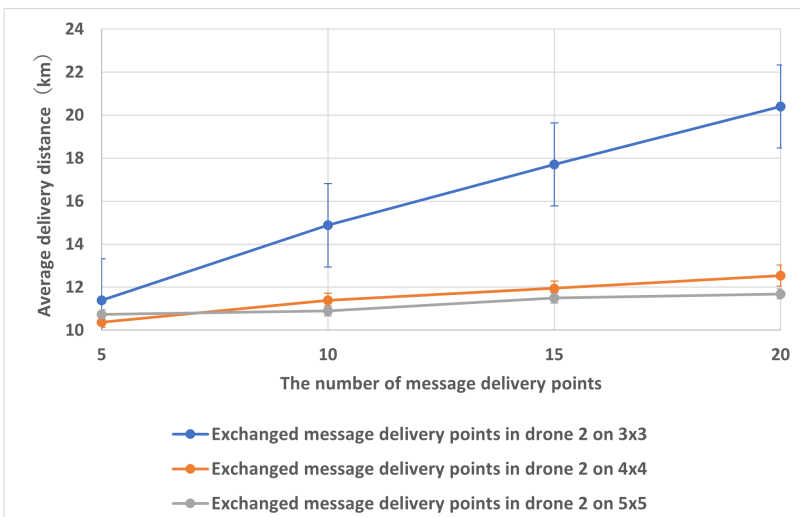

Figure 21: The average delivery distance of drone 2 with exchanging message delivery points for different cell size in Experiment 3.

6.4 Simulation Result in Experiment 4

Next, Experiment 4 examined. Figure 22 is this experiment's network topology that is a screenshot of the animation tool in The One Simulator. The screenshot is captured when 100 drones take off from the delivery base circled in red. There are also 100 destinations and 100 message delivery points.

In this figure, a blue word is a node name. A green circle around the name's first letter represents its communication range. A node whose name begins "Base" represents a delivery base, "Dest" represents a delivery destination, "DataPoint" represents a message delivery point, and "Drone" represents a drone. Drones in the orange and blue polygonal areas are flying toward the lower left corner and the upper right corner in the figure, respectively.

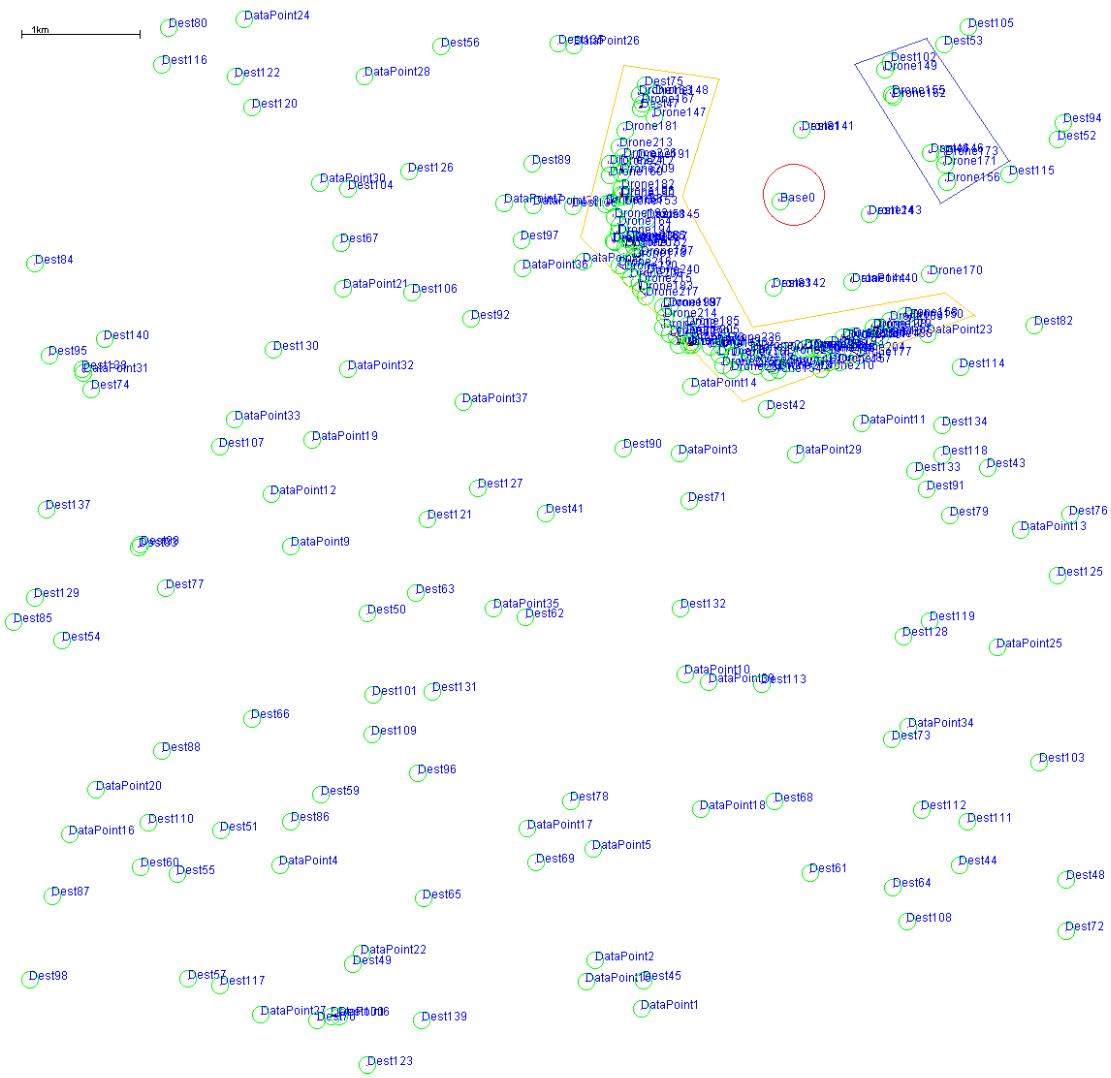

Figure 22: The network topology of Experiment 4 in The One Simulator.

Figure 23 shows that the average latency when the number of message delivery points is changed for the different number of drones. From this figure, the average latency decreases as the number of drones increases and as the number of message delivery points decreases.

For example, for 20 message delivery points, the average latency is 297.2 seconds for 20 drones, while 254.0 seconds to 272.2 seconds for 40 or more drones. In the latter cases, since more than 20

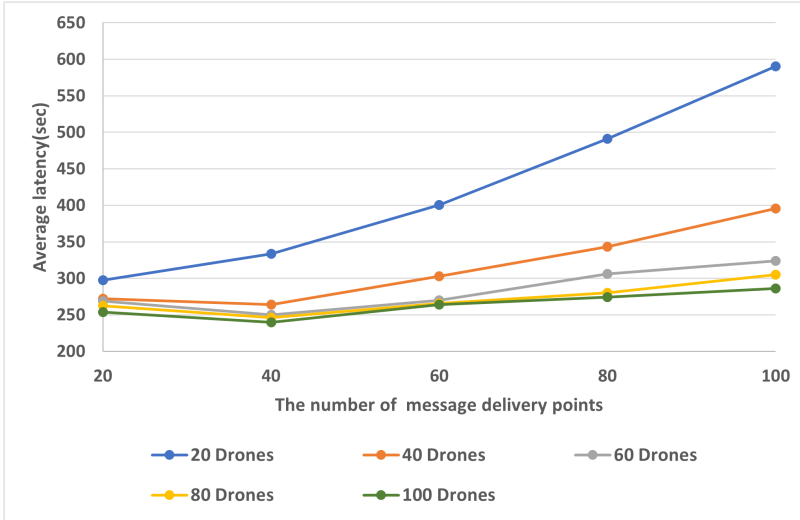

Figure 23: The average latency with changing the number of message delivery points for the different number of drones in Experiment 4.

drones have only 20 message deliveries, some of them just fly to the parcel delivery point and just return to the base. Thus, in the latter case, drones cannot reduce their enough latency from the former case. This is why the average latency for 20 drones is shorter than that for 40 drones.

In contrast, for 80 message delivery points, the average latency is 306.1 seconds for 60 drones, while 274.1 to 280.4 seconds for 80 or more drones. Moreover, when the number of message delivery points is 100, the average latency for 100 drones is 38.3 % of that for 40 drones, since for 100 message delivery points, each drone delivers to only one message delivery point for the former case, but to two or three message delivery points for the latter case.

However, as the number of drones becomes bigger, their flight and maintenance costs are also bigger. Since the battery capacity of the drones limits the flight distance, there also exists the minimum number of drones required for the number of message delivery points. Therefore, the trade-off between the average latency and the cost of the drones exists, and an appropriate number of drones should be used for these factors.

6.5 Simulation Result in Experiment 5

Experiment 5 compared the proposed method with the previous method WFPP from the average delivery distance and the average actual computation time.

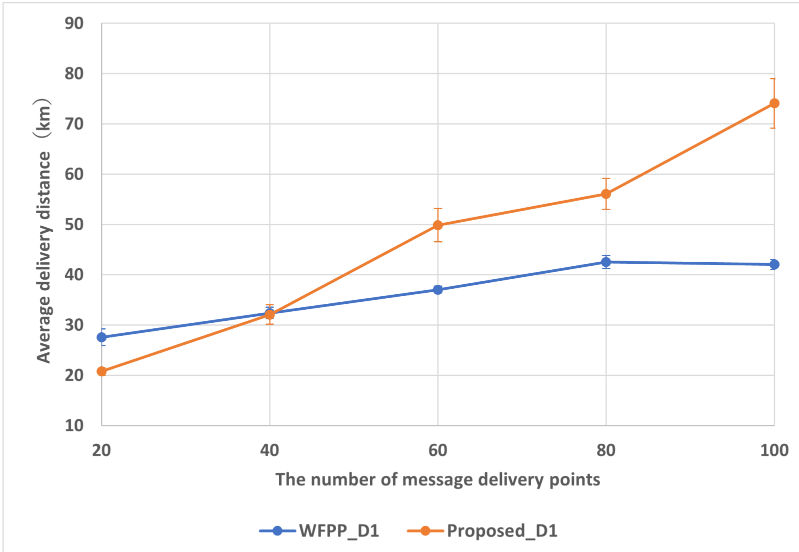

Figure 24: The average delivery distance in the proposed method and WFPP of Drone 1 in Experiment 5.

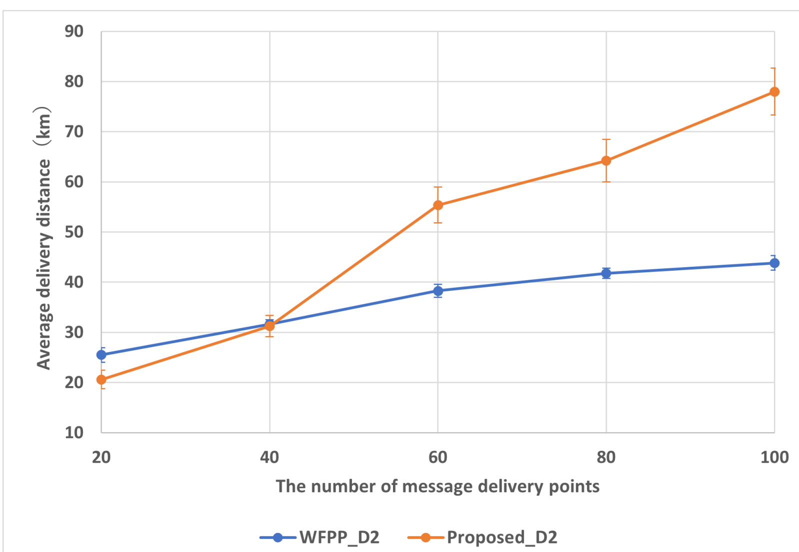

Figure 25: The average delivery distance in the proposed method and WFPP of Drone 2 in Experiment 5.

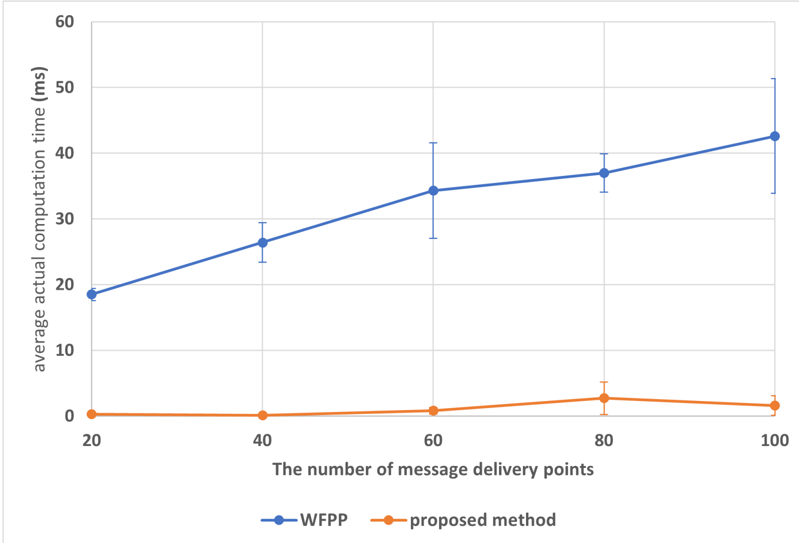

Figure 26: The average actual computation time in the proposed method and WFPP in Experiment 5.

Figure 24 and Figure 25 shows that when the number of message delivery points is small, the proposed method sometimes has a shorter average travel distance, but there is not much difference between the proposed method and the existing method.

However, when the number of the points increases, the average travel distance increases about 1.54 to 1.78 times. This is because the existing method requires higher accuracy flight paths by using Christofides' algorithm.

Figure 26 shows that the average actual computation time of the proposed method is from 0.0 ms to 1.0 ms, while that of the existing method is about 20.0 ms to 40.0 ms, and then the difference is between 20.0 to 40.0 times. The impact of the computation time for one exchange seems not to be so large in a practical environment, since it may take a few seconds from contacting a drone to exchanging message delivery points. Moreover, each drone must periodically scan the other drones in WFPP. However, if the density of drones is sufficiently high and each drones may exchange message delivery points multiple time then the impact becomes large. In the proposed method, the contact is scheduled and can be reduced if the density is high. Since the entire network is divided into cells, the number of exchanged message delivery points is also limited.

This result indicates that the proposed method does not need too much resources for calculations than WFPP.

7 Conclusion

This paper proposed a method for message delivery in a drone logistics networks. Compared to WFPP, the proposed method significantly reduces the complexity of computation performed by drones.

Every message delivery point is visited at least once so that delivery rate is 100%.

The results of the simulation experiments show that the average delivery distance increases according to the number of message delivery points, but decreases, when message delivery points are exchanged compared with the case that no points are exchanged. The average delivery distance and the average actual computation time are compared between the proposed method and the previous method WFPP. The average travel distance increased by 1.54 to 1.78 times at the maximum, in contrast, the computation time was reduced by a factor of 40.

In the future work, the authors should improve the proposed method to reduce the average latency and solve the problem for cells not traversed by drones described in Section 4.2.4.

References

[1] Amazon.com, Inc., “Amazon Prime Air FAQs — Amazon Customer Service,” https://www.amazon.com/gp/help/customer/display.html?nodeId=T3jxhuvPfQ629BOIL4, 14 Feb 2023.

[2] Japan Post Co., Ltd., “Implementation of Delivery Trials in Cooperation between Drones and Delivery Robots, ” https://www.post.japanpost.jp/notification/pressrelease/2021/00_honsha/1130_02.html, 16 Jan 2022. (in Japanese)

[3] Saeid Iranmanesh, Raad Raad, Muhammad Salman Raheel, Faisel Tubbal, and Tony Jan, “Novel DTN Mobility-Driven Routing in Autonomous Drone Logistics Networks,” IEEE Access, Vol. 8, pp. 13661–13673, 2020.

[4] Matsutani Kazuya and Kimura Shigetomo, “Message Routing Based on Divided Logistics Areas for Drone Logistics Networks,” IEICE Technical Report, Vol. 122, No. 48, pp. 1–6, 2022.(in Japanese)

[5] Kazuya Matsutani and Shigetomo Kimura, “Delivery Routing to Reduce Calculation Load of Drones on Divided Logistics Areas for Drone Logistics Networks,” Proceedings of 2022 Tenth International Symposium on Computing and Networking Workshops (CANDARW), pp. 44–47, 2022.

[6] Tsuru Masato, Uchida Masato, Takine Tetsuya, Nagata Akira, Matsuda Takahiro, Miwa Hiroyoshi, Yamamura Shinya, “Delay Tolerant Networking Technology – The Latest Trends and Prospects,” IEICE Communications Society Magazine, Vol. 2011, No. 16, pp. 57–68, 2011. (in Japanese)

[7] Amin Vahdat and David Becker, “Epidemic Routing for Partially-Connected Ad Hoc Networks,” Technical Report, Duke University, 2000.

[8] Wenchao Pi, Hao Hao, Shujie Yang, Yong Liao, and Changqiao Xu, “RBDR: A Distributed Routing Algorithm in High Mobility UAV Network,” Proceedings of 2019 International Conference on Networking and Network Applications (NaNA), pp. 45–50, 2019.

[9] Matsutani Kazuya and Kimura Shigetomo, “Improvement of the Reputation-Based Distributed Epidemic Routing Algorithm with Adjusting Nodes,” Proceedings of 83rd National Convention of IPSJ, Vol. 2021, pp. 295–296, 2021. (in Japanese)

[10] Nicos Christofides, “Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem,” Carnegie-Mellon Univ Pittsburgh PA Management Sciences Research Group, 1976.

[11] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen, “The ONE Simulator for DTN Protocol Evaluation,” SIMUTools ’09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques, 2009.