The Energy-Aware Multi-UAV Dispatch and Handoff Algorithm for Maximizing the Event Communication Time in Disasters

Ing-Chau Chang 1,*, Chi-Sheng Liao 1 and Chin-En Yen 2

1 Department of Computer Science and Information Engineering, National Changhua University of Education, Changhua 50007, Taiwan; suboy@hotmail.com.tw
2 Department of Early Childhood Development and Education, Chaoyang University of Technology, Taichung 41349, Taiwan; ceyen@cyut.edu.tw

* Correspondence: icchang@cc.ncue.edu.tw; Tel.: +886-4-723-2105

Abstract: For handling the broken-down communication infrastructure when a disaster event happens, this paper proposes to dispatch the unmanned aerial vehicle (UAV) to the disaster area as the relay node, which further forms a Flying Ad hoc Network (FANET). Since the UAV only owns limited energy and a disaster event may need multiple UAVs to cover its area, an efficient multi-UAV dispatch algorithm is critical to recover the communication link of the disaster area. In this paper, we adopt the mobile ground control station (GCS) to transport UAVs to the boundary of the disaster area first. According to the UAV energy consumption rate during flight and two communication modes, the UAV charging progress, and the number of required UAVs of the event, the mobile GCS then executes the proposed energy-aware multi-UAV dispatch algorithm (EAMUD) to dispatch multiple UAVs to this disaster area for building the FANET. Hence, the broken-down link in the disaster area is recovered after the FANET connects to nearby network infrastructure. Further, we propose the multi-UAV handoff scheme and exception handling processes to replace energy-exhausted UAVs for maximizing the event communication time of the disaster event. Finally, we execute simulations for related work and four EAMUD variants under different parameter values in the real scenario. These results exhibit that EAMUD with the Postpone method (EAMUD-P) achieves the highest event communication time among all these schemes.

Keywords: unmanned aerial vehicle (UAV); mobile ground control station; energy-aware multi-UAV dispatch and handoff algorithm; VANET; FANET

1. Introduction and Related Work

In recent years, the unmanned aerial vehicles (UAV) applications have grown rapidly [1,2] because the UAV owns several characteristics [3,4] such as higher mobility, less interference by fewer obstacles in the air, more flexible capability by equipped with different sensors, etc. Research in [5,6] deployed UAVs in 5G cellular networks as relay stations to improve signal coverage. In [7–10], several routing protocols of the Vehicular Ad hoc Network (VANET) were proposed. Further, the UAV was integrated with VANET as a temporary gateway whenever the network infrastructure malfunctioned after natural disasters like the earthquake or mudslide destroyed backbone communication [11–14]. Several UAV image applications for disaster management have been proposed in [15–18]. However, most research focuses on how to deploy UAVs to fully cover the disaster area [19–28]. Fewer of them work on the scheduling policies to dispatch new UAVs for replacing old ones and extending the communication lifetime before they have drained all their battery energy.

By considering the limited battery energy, a scheduling framework proposed in [22] dispatches a group of UAVs to their pre-assigned destinations. It further formulated these missions as a mixed integer linear programming (MILP) problem, which focuses on
minimizing the total energy consumed by all UAVs. This work assumes the starting and finishing time of all missions must be known in advance. However, this assumption may not be fulfilled in real cases. Moreover, it has not mentioned how to dispatch new UAVs to hand over the energy-drained ones for continuing the missions. In [23], two types of the Ground Control Stations (GCS), i.e., fixed and mobile ones, were proposed to control, receive and process all the data of the UAV system. The optimal UAV flying speed has been measured and analyzed in [24]. It formulates the total energy consumption $E_{UAV}$ as the sum of the energy, i.e., $E_{travel}$, consumed for flying to the destination, that, i.e., $E_{hover}$, for hovering over the destination and that, i.e., $E_{radio}$, for communicating with the underwater sensor network. Hence, the energy model is formulated as $E_{UAV} = E_{travel} + E_{hover} + E_{radio}$.

Three communication scenarios between vehicles, between vehicle and UAV, and between UAVs have been modeled as ground-to-ground (G2G), air-to-ground (A2G)/ground-to-air (G2A), and air-to-air (A2A) communication, respectively [25]. According to the height and density of buildings, the A2G and G2A channels can be modeled as the line-of-sight (LoS) or non-line-of-sight (NLoS) propagation with corresponding probabilities. Oppositely, the A2A channel between UAVs is modeled as the LoS propagation because far fewer obstacles existed in the air. Based on these models, this paper will compute the signal-to-noise ratio (SNR) and the consumed signal power.

By using the MAVLink protocol, a UAV replacement algorithm proposed in [26] executes uninterrupted multi-UAV waypoint mission for an Unmanned Aerial System (UAS). It also presented the mission continuity coefficient of the battery replacement scheme as being higher than the battery recharging one. However, it lacks analysis about how the number of batteries for replacement influences the performance of mission continuity and how much additional costs are. Similarly, a set of MAVLink messages presented in [27] supports UAV replacement automatically as the UAV has depleted its battery energy. However, it does not describe how to choose a new UAV to replace the energy-depleted one. Two Linear Integer Problem (LIP) optimization solutions for an IoT platform proposed in [28] execute a specific IoT task with the minimum energy consumption. For selecting appropriate UAVs for a task, diverse criteria, e.g., what sensors the UAVs own, how much energy they have charged, and how close they are to the task position, are considered in this paper. However, USP does not support the UAV replacement scheme when the UAV runs out of its battery energy.

In this paper, we consider the problem when multiple roads in the disaster suffer communication link failures, called events here, simultaneously. Due to the limited communication range and available battery energy of each UAV, the ground control station (GCS) proposed in this paper adopts the proposed energy-aware multi-UAV dispatch algorithm (EAMUD) to dispatch different numbers of UAVs to corresponding locations on each link-broken road. After every UAV arrives at its assigned location, it first forms the flying ad hoc networks (FANET) with nearby UAVs and then builds a two-layer network with existing VANET or 4G/5G cellular network to recover communication links of each event with the maximum event communication time.

Table 1 compares the aforementioned work in terms of important characteristics for dispatching UAVs to recover the communication link of the disaster area. Most of them focus on dispatching a UAV to serve a single event without considering multiple events, UAV energy consumption, or subsequent handover process. Only the proposed EAMUD provides solutions for all of them. It has the following contributions:
Table 1. Comparison of related work.

| Research | Features | Handle Multiple Events and Dispatch Multiple UAVs for an Event | Consider the UAV Charging Progress and Energy Consumption for Flying and Communication | Check whether the Dispatched UAVs Can Maximize the Event Communication Time | Consider the UAV Handover Process and Handle Exceptions | Propose the LP Optimization Equations to Compute the Optimal Event Communication Time |
|----------|----------|---------------------------------------------------------------|--------------------------------------------------------------------------------|--------------------------------------------------------------------------------|------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| Ref. [22]| No       | Yes                                                          | No                                                                               | No                                                                                | No                                                                     | No                                                                               |
| Ref. [26]| No       | No                                                           | No                                                                               | No                                                                                | No                                                                     | No                                                                               |
| Ref. [27]| No       | No                                                           | No                                                                               | No                                                                                | No                                                                     | No                                                                               |
| Ref. [28]| No/Yes   | No                                                           | No                                                                               | No                                                                                | No                                                                     | No                                                                               |
| EAMUD    | Yes      | Yes                                                          | Yes                                                                              | Yes                                                                               | Yes                                                                    | Yes                                                                               |

1. We first propose the mobile GCS to carry UAVs to the disaster area for recovering broken links. GCS adopts the UAV pool to record information of all UAVs and schedules all events stored in the proposed event queue for efficient multi-UAV dispatching.

2. Then we propose the energy-aware multi-UAV dispatch algorithm (EAMUD) to dispatch multiple UAVs to the disaster area for building the FANET, by considering the UAV battery charging progress and energy consumption for flying and communication. Further, EAMUD performs the proposed Event-Based Dispatching (EBD) algorithm to check whether communication durations of all dispatched UAVs overlap or not to maximize the event communication time.

3. For continuing communication of the broken link before the dispatched UAV drains its remaining energy, we propose the UAV handover process of EAMUD such that GCS can dispatch an optimal UAV to hand over the dispatched one. Further, whenever GCS cannot find enough available UAVs in the UAV pool to serve the current sub-event, two exception handling methods, i.e., Drop and Postpone, are proposed in EAMUD to schedule the scarce UAV resources efficiently, which in turn extends the event communication time.

4. For given events, we propose the linear programming optimization equations to compute the optimal event communication time as the performance benchmark for dispatching UAVs in the ideal scenario.

The rest of the paper is organized as follows. Section 2 describes details of the proposed event structure and flows of the EAMUD and EBD algorithms. Section 3 describes how to formulate linear programming equations for maximizing the event communication time in the ideal scenario. Section 4 shows simulation results of EAMUD in the ideal and real scenarios. Section 5 concludes this paper.

2. Energy-Aware Multi-UAV Dispatch Algorithm (EAMUD)

2.1. The Two-Layer FANET-VANET Network in Disaster

Whenever the disaster breaks two roads as shown in Figure 1, corresponding communication paths like VANET links through them will disconnect. In this paper, we denote these two broken VANET link as Event A, i.e., Ev\textsubscript{A}, and B, i.e., Ev\textsubscript{B}, respectively. For continuing communication for these events, we propose to drive a mobile GCS, which carries multiple UAVs and charging faculties, to a position near these broken roads. The GCS will first connect to existing VANET. Then it will dispatch several UAVs to corresponding locations on each link-broken road to build a temporary FANET in the air by creating UAV-to-UAV (U2U) links between dispatched UAVs. After that, the UAV located at the edge of FANET will connect to the vehicle in the boundary of VANET through the UAV-to-Vehicle (U2V) link, which in turn forms a two-layer FANET-VANET network to recover communication link of each event. According to the energy model of UAV battery described below, GCS and UAVs are able to calculate energy consumption for flying, hovering, and communication. Excluding the energy for flying back to GCS for recharging the battery, the remaining energy of each dispatched UAV is consumed on hovering over its assigned position and maintaining the U2U/U2V communication. Based on the EAMUD algorithm proposed later, GCS can dispatch an optimal UAV to hand over the dispatched one for
continuing communication of the event before it drains its remaining energy. In this way, this two-layer FANET-VANET network with EAMUD achieves the longest communication duration, i.e., the maximum event communication time, for each broken road. In Figure 1, GCS dispatches $U_1, U_2$ and $U_3$ to their corresponding positions, which are denoted as the sub-event $Ev_{A,1}$, $Ev_{A,2}$, and $Ev_{A,3}$, respectively. Hence, the communication link of $Ev_A$ is recovered by the FANET, i.e., $U_2 \leftrightarrow U_3 \leftrightarrow U_1$, and two U2V links $V_1 \leftrightarrow U_2$ and $U_1 \leftrightarrow V_3$.

![Figure 1](image.png)

**Figure 1.** The two-layer Flying Ad hoc Network-Vehicular Ad hoc Network (FANET-VANET) network in disaster.

### 2.2. The Energy Model of UAV Battery

#### 2.2.1. Recharging of the UAV Battery

Whenever a UAV stays on the dock of GCS, GCS continues charging this UAV until its battery has been fully charged or it has been dispatched by GCS. Assume the battery capacity of the UAV is denoted as $E_{Max}$ in Joule (J). The maximum time, i.e., $T_{Char}$, in hours to fully charge the UAV battery is formulated as follows:

$$T_{char} = \frac{E_{Max} \times (1 + \alpha_{loss})}{I_{char} \times V_{char}} \tag{1}$$

where $I_{char}$ is the charge current in Amp, $V_{char}$ is the current voltage in V and $\alpha_{loss}$ is the efficiency loss coefficient, e.g., 10%.

#### 2.2.2. Energy Consumption of the UAV Battery

There are two kinds of energy consumption on UAV communication. One is the transmission power of the U2U link between adjacent UAVs and the other is that of the U2V link for connecting the edge UAV in FANET and the edge vehicle in VANET. In the following, we will discuss how to convert the transmission power into corresponding energy consumption for the U2U and U2V links, respectively.

- **Energy consumption for the U2U communication**

  Assume there are no obstacles between UAVs, hence the U2U link belongs to the LoS type. For two adjacent UAVs to maintain an U2U link, the required transmission power, i.e., $P_{U2U}$, in unit of dBm at the U2U sender must be larger than the SNR threshold, i.e., $SNR_{thU2U}$, at the receiver, which is formulated as follows [25]:

$$P_{U2U} \geq SNR_{thU2U} + PL + N_{U2U} \tag{2}$$

where $PL$ is the free space loss between UAVs and $N_{U2U}$ is the noise power at the U2U receiver. The value of $PL$ is equal to $10\log\left[\left(\frac{4\pi \times f_1}{C}\right)^2 \times d_{U2U}^2\right]$, where $f_1$ is the carrier frequency, $C$ is the light speed and $d_{U2U}$ denotes the Euclidean distance between UAVs.
Finally, the required transmission power \( P_{LU2V} \) in unit of dBm is converted to \( E_{LU2U} \) in unit of watt for the U2U sender as follows:

\[
E_{LU2U}(W) = \left(1 mW\right)^{10\left(\frac{P_{LU2V}}{10}\right)} \times 10^{-3}
\]

\[\text{Energy consumption for the U2V communication}\]

Because the U2V connection between the edge UAV and vehicle is a combination of LoS and NLoS communication, the required transmission power, i.e., \( P_{LU2V} \), in LoS and that, i.e., \( P_{NU2V} \), in NLoS consumed by the U2V sender can be formulated as follows, respectively [25].

\[
P_{LU2V} \geq SNR_{lu2v} + PL + \xi_{LOS} + N_{U2V}
\]

\[
P_{NU2V} \geq SNR_{nu2v} + PL + \xi_{NLOS} + N_{U2V}
\]

For connecting the edge UAV and vehicle through LoS communication, Equation (4) means that \( P_{LU2V} \) in dBm must be no less than the sum of the SNR threshold, i.e., \( SNR_{lu2v} \), at the U2V receiver, the free space loss \( PL \), the shadow fading with normal distribution \( \xi_{LOS} \sim N(\mu_{LOS}, \sigma_{LOS}^2) \) and the noise power \( N_{U2V} \) over the LoS U2V connection. Please note that \( \mu_{LOS} \) and \( \sigma_{LOS}^2 \) are the mean and the standard deviation of the shadow fading for the LoS link. Similarly, Equation (5) formulates the relationship among \( P_{NU2V}, SNR_{nu2v}, PL, \xi_{NLOS} \) and \( N_{U2V} \), where \( \xi_{NLOS} \sim N(\mu_{NLOS}, \sigma_{NLOS}^2) \) is the shadow fading with normal distribution over the NLoS U2V connection. According to [25], the probability of the LoS U2V connection, i.e., \( P_{LoS} \), is computed as follows:

\[
P_{LoS} = \frac{1}{1 + B \cdot \exp[-C \cdot (\theta - B)]}
\]

where \( B \) and \( C \) are constant values for different propagation environment. \( \theta \) is the elevation angle between the ground and the U2V link. Hence, the probability of the NLoS U2V connection, i.e., \( P_{NuLOS} \), is equal to 1 - \( P_{LoS} \). Then the average required U2V transmission power \( P_{LU2V} \) in dBm is modelled as follows:

\[
P_{LU2V} = P_{LU2V} \times P_{LoS} + P_{NU2V} \times (1 - P_{LoS})
\]

where \( P_{LU2V} \) is the weighted sum of \( P_{LU2V} \) and \( P_{NU2V} \), depending on \( P_{LoS} \) and 1 - \( P_{LoS} \), respectively. Finally, the transmission power \( P_{LU2V} \) in unit of dBm is converted to \( E_{LU2V} \) in unit of watt for the U2V sender using Equation (3).

2.3. The Event Queue and UAV Pool of GCS

In this paper, we design an array, called the event queue (\( Ev_q \)), to store all new and handover events chronologically for UAV dispatching. Data structure of \( Ev_q \) consists of four components as shown in Figure 2. Component A denotes \( Ev_q \) adopts the time of occurrence or that of handover, i.e., \( t \), as its array index. This time will be called as the processing time of the sub-event in this paper. Component B presents new or handover event \( j \), i.e., \( Ev_{ij} \), which will be processed by the EAMUD algorithm at time \( t \). As mentioned above, every event may contain one or more sub-events that represent specific destinations of dispatched UAVs. As shown by component C in Figure 2, \( Ev_{ij} \), denotes sub-event \( i \) of event \( j \) at time \( t \). There are three states, i.e., New, Ho and Res, for each sub-event. The sub-event in state New will be dispatched a new UAV by EAMUD. The sub-event in state Ho is the one which will dispatch a handover UAV to replace the energy-drained one. Finally, the sub-event in state Res means it has reserved a charging UAV of GCS. Component D records information of each sub-event, including its ID, state, \((x, y)\) location, ID of the reserved UAV \( (U_{Res}) \).
Figure 2. Data structure of $Ev_q$ (A) the event time, (B) the event $j$ at time $t$, (C) the sub-event $i$ of event $j$ at time $t$, (D) information of each sub-event.

GCS adopts a UAV pool to record information of all UAVs, as shown in Table 2. Each UAV is in one of the four states, i.e., Available, Charging, Dispatched and Reserved. If a UAV in GCS has been charged to an energy level higher than dispatch energy threshold $E_{DTH}$, which means this UAV could be dispatched, its state is Available. However, the UAV state is Charging before its energy level has not reached $E_{DTH}$. If a UAV has been dispatched to the location of a sub-event, its state is Dispatched. Finally, the state of a UAV that has been reserved for dispatching to a sub-event in the future is Reserved.

| UAV Pool | UAV Id | State      | $E_{Bat_u}$ | Sub-ev ID | $E_{Com_u}$ |
|----------|-------|-----------|-------------|-----------|-------------|
| U_1      | Available | 100      |             |           |             |
| U_2      | Available | 100      |             |           |             |
| U_3      | Available | 93       |             |           |             |
| U_4      | Charging  | 40       |             |           |             |
| U_5      | Dispatched | 97      | $Ev_{A,1}$ | $(x_1^A, y_1^A)$ | 3           |
| U_6      | Dispatched | 70.5    | $Ev_{A,2}$ | $(x_2^A, y_2^A)$ | 3           |
| U_7      | Reserved  | 45       |             |           |             |

Figure 3 shows how to calculate the handover interval, i.e., $\Delta t$, for the original dispatched UAV. The yellow line is its time axis, starting from the time $t$ when GCS dispatches this UAV to serve sub-event $Ev_{j,i}$. Orange point 1 indicates the time when the dispatched UAV has reached the position of $Ev_{j,i}$ with the flying duration $f_{j,i}$. Orange point 2 is the time when the original UAV has consumed its energy to the level only enough for flying back to GCS. Hence, the original UAV serving $Ev_{j,i}$ could communicate with its neighbors for the maximum duration of $com_{j,i}$, which is between orange points 1 and 2. For replacing the original UAV and continuing the link communication of the event, GCS has to dispatch the handover UAV at time $t_{next}$, which is equal to $t + \Delta t$ as indicated by orange point 3 such that the handover UAV could reach the position of $Ev_{j,i}$ at the time of orange point 2. As a result, the duration from time $t$ to $t_{next}$ is defined as the handover interval, i.e., $\Delta t$, of the original dispatched UAV. In this paper, we assume the original and handover UAVs have the same flying durations $f_{j,i}$. Hence, $\Delta t$ is equal to $com_{j,i}$, which is calculated as follows:

$$
\Delta t = f_{j,i} + com_{j,i} - f_{j,i} = com_{j,i} = \frac{E_{Rem}}{E_{com}^{u} + E_{Ho}^{u}} = \frac{E_{Bat_u}^{u} - 2 \times E_{Fly}^{u}}{E_{com}^{u} + E_{Ho}^{u}}
$$

(8)
Figure 3. The time $t_{next}$ for dispatching the handover unmanned aerial vehicle (UAV) to replace the original one.

Further, the remaining energy $E_{Rem}^u$ for the original UAV to consume on communication and hovering at the position of $Ev_{ij}$ is the difference of its energy $E_{Bat}^u$ before dispatching and that, i.e., $2 \times E_{Fly}^u$, for its round-trip flight. Hence, $com_{ij}$ and $\Delta t$ are the quotient of $E_{Rem}^u$ divided by the unit energy, $E_{Com}^u + E_{Ho}^u$, consumed on communication and hovering. Please note that the unit energy consumed on communication, i.e., $E_{Com}^u$, is equal to the sum of energy consumed for the U2U and U2V communication, i.e., $E_{tU2U}$ and $E_{tU2V}$. Finally, GCS inserts a handover sub-event into its event queue at time $t_{next}$, i.e., $t + \Delta t$, for dispatching the handover UAV to replace the original one by EAMUD.

2.4. Details of the EAMUD Algorithm

The main concept of the EAMUD algorithm is as follows. Before the GCS dispatches a UAV for a sub-event recorded in its event queue when the processing time of this sub-event has come, it will first check whether it could dispatch the required UAVs for all sub-events, which belong to the parent event of the current sub-event, at their processing time or not. In this paper, every sub-event with the same parent event of this sub-event is called as its sibling one. If the GCS fails to fulfill the UAV requirement for any sibling sub-event, this sub-event has to be postponed. In the following, two different cases to postpone the sub-event are discussed.

1. If this sub-event is not the last sibling one of its parent event, the GCS will postpone it to the processing time of next sub-event, which has not been checked, of the same parent event: As shown in Figure 4, $Ev_A$ has three sub-events $Ev_{A,1}$, $Ev_{A,2}$ and $Ev_{A,3}$. According to Equation (8), the GCS calculates the handover interval, i.e., $\Delta t$, of the original dispatched UAV for $Ev_{A,1}$ and knows it has to dispatch a handover UAV to replace the energy-drained one at time 2. For maximizing the event communication time of the whole event, the EAMUD algorithm has to check whether the GCS could dispatch UAVs for all other sub-events, i.e., $Ev_{A,2}$ and $Ev_{A,3}$, during time 2 to time 6, which indicates the time for the GCS to dispatch the next handover UAV for replacing the preceding dispatched one at time 2. If the GCS is not able to dispatch a handover UAV at the processing time of $Ev_{A,2}$, i.e., time 3, which will introduce a communication break for this event later and waste the handover UAV dispatched for $Ev_{A,1}$ at time 2. Hence, the EAMUD algorithm postpones these two sub-events, $Ev_{A,1}$ and $Ev_{A,2}$, to the processing time of next sub-event $Ev_{A,3}$ at time 4 and then performs EAMUD again.
Figure 4. The scenario to postpone a sub-event to the processing time of its next sub-event.

2. If this sub-event is the last sibling one of its parent event, the GCS will postpone all sub-events to its processing time plus a delay interval $t_{delay}$. As the case mentioned above, if the GCS can dispatch a handover UAV at time 3 to $Ev_{A,2}$ but cannot dispatch another handover UAV at time 4 to the last sub-event $Ev_{A,3}$, this scenario means that both handover UAVs, which are dispatched at time 2 and 3, are wasted. Hence, as shown in Figure 5, the EAMUD algorithm postpones these two sub-events, $Ev_{A,1}$ and $Ev_{A,2}$, to time 4, i.e., the processing time of the last sub-event $Ev_{A,3}$, plus $t_{delay}$.

After that, the GCS performs EAMUD again. All sub-events are first sorted according to their Euclidean distance to the GCS. Then by averaging the flying time difference of any two adjacent sub-events of an event as follows:

$$ t_{delay, j} = \frac{\sum_{i=1}^{N_j-1} \left( \left( d_{ji+1} - d_{ji} \right) / S \right)}{N_j - 1} \quad (9) $$

where $S$ is the average flying speed of the UAV, $d_{ji}$ is the Euclidean distance between the GCS and the location of the $i$-th sub-event $Ev_{ji}$ of event $Ev_j$ and $N_j$ is the sub-event number of $Ev_j$, the value of $t_{delay, j}$ is calculated accordingly.

Figure 5. The scenario to postpone all sub-events to the processing time of the last sub-event plus a delay interval $t_{delay}$.

Figure 6 is the complete flow chart of EAMUD, which is explained below. Assume EAMUD starts at time $T$.

Step 1. EAMUD first checks if there is any sub-event in $Ev_q$ has to be processed at time $T$. If yes, EAMUD executes Step 2. Otherwise, the current time will advance to $T + 1$ to execute Step 1.1.
Step 1.1 If there is any sub-event in $\mathcal{Ev}_q$ has to be processed, EAMUD goes to Step 1. Otherwise, EAMUD stops its execution because it has processed all sub-events in $\mathcal{Ev}_q$.

Step 2. EAMUD checks if the current time is equal to the end time of this sub-event. If yes, which means the parent event of this sub-event has finished, all sibling sub-events of the same parent event have to be removed from $\mathcal{Ev}_q$. Otherwise, EAMUD goes to Step 3.

Step 3. EAMUD checks whether this sub-event has reserved a UAV. If no, it goes to Step 4. Otherwise, it executes the algorithm in Step 5 to check UAV dispatching, based on $t_{next}$ of this sub-event.

Step 4. If the number of UAVs needed for this sub-event is less than or equal to the number of UAVs which states are Available in the UAV Pool, EAMUD will dispatch this number of Available UAVs from the UAV pool for this sub-event, with the highest battery powers first criterion. After that, EAMUD goes to Step 5. If the number of UAVs needed for this sub-event is higher than the number of UAVs which states are Available in the UAV Pool, EAMUD must handle this exception by jumping to Step 6.

Step 5. At this step, EAMUD performs the event-based dispatching (EBD) algorithm to check whether communication durations of UAVs already dispatched for all other sibling sub-events could overlap with the communication duration of the UAV dispatched for this sub-event. If yes, the communication link failure of this event has been recovered during the overlap duration, which raises the event communication time. The flow of the EBD algorithm is shown in Figure 7 and its details are explained as follows:

Step 5.1 If the next handover time $t_{next}$ of every other sibling sub-event is later than that of this sub-event, which means that each sub-event has reserved its handover UAV at its corresponding handover time $t_{next}$, EBD goes to Step 5.2. Otherwise, it performs Step 5.4.

Step 5.2 Insert the next handover time $t_{next}$ of this sub-event into $\mathcal{Ev}_q$ and goes to Step 5.3.

Step 5.3 Dispatch the reserved UAV to the location of this sub-event for continuing the event communication. At this time, the EBD algorithm has completed its checking process and goes back to Step 7 of EAMUD.

Step 5.4 If this sub-event has not reserved a UAV, this EBD algorithm goes to Step 5.5. Otherwise, it goes back to Step 5.1.

Step 5.5 According to status of all UAVs in the UAV pool of the GCS, EBD can decide if there are enough Available or Charging UAVs could be reserved for this sub-event. If yes, the GCS will reserve the UAV having the highest battery energy for this sub-event and this algorithm goes back to Step 5.1. If the GCS fails to reserve any UAV for this sub-event, it will cancel UAV reservations of all sub-events which have been checked in Step 5, return these reserved UAVs to the UAV pool and go to Step 5.6.

Step 5.6 If this sub-event is the last sibling one of its parent event, EBD goes to Step 5.8. Otherwise, it goes to Step 5.7.

Step 5.7 Because this sub-event is not the last sibling one of its parent event, the GCS will postpone all checked sub-events to the processing time of next sibling sub-event, which has not been checked by this algorithm, as mentioned in case 1 above. EBD then goes to Step 7.

Step 5.8 Because this sub-event is the last sibling one of its parent event, the GCS will postpone all sibling sub-events to its processing time plus a delay interval $t_{delay}$. EBD then goes to Step 7.

Step 6. Two exception handling methods are proposed here.

- The Drop method:
  - a. Remove this sub-event from $\mathcal{Ev}_q$ to stop dispatching the handover UAV for this sub-event.
b. For maintaining the link communication of an event, each sub-event of this event must be served by a dispatched UAV. Since this sub-event has been removed from \( Ev_q \), all sibling sub-events must be removed from \( Ev_q \) to avoid wasting UAV energy.

- The Postpone method:
  a. Assume the processing time of this sub-event \( i \) is \( t_i \). This method is to postpone its processing time to \( t_i + t_{\text{delay}} \) and modifies its information in \( Ev_q \).
  b. Each sibling sub-event with its processing time earlier than \( t_i + t_{\text{delay}} \) is postponed to \( t_i + t_{\text{delay}} \). All corresponding information in \( Ev_q \) are updated accordingly.

![Figure 6. The flow of EAMUD.](image)

![Figure 7. The flow of the event-based dispatching (EBD) algorithm.](image)
After that, EAMUD goes to Step 7.

Step 7. According to results of Step 5 and Step 6, GCS updates the corresponding UAV state in the UAV pool as Available, Charging, Reserved, or Dispatched, respectively. Then, the current time will advance to \( T + 1 \) and EAMUD continues to check other sub-events in \( Ev_q \) by going back to Step 1.

3. Linear Programming Formulation for Maximizing the Event Communication Time in the Ideal Scenario

As mentioned above, the UAV is a scarce resource for recovering communication of the broken link in disaster. How CN dispatches UAVs to serve all sub-events of each event with the EAMUD algorithm depends on which UAV in the UAV pool is available when the event happens. Since each event may occur at different time in the real disaster scenario, it is complex to model the real UAV dispatching process, not to mention the subsequent UAV handover processes and UAV dispatch exceptions. In the following, for maximizing the aggregate event communication time of all given events for a disaster duration, the UAV dispatching process in the ideal scenario is formulated and solved by the linear programming technique under four constraints to find its optimal solution. Please note that the ideal scenario here assumes all events occur simultaneously when the simulation begins, which is rare in the real scenario. Further, it only considers how CN dispatches UAVs to serve each event in the first round and ignores the subsequent UAV handover processes and UAV dispatch exceptions. Assume \( Ev = \{ Ev_1, Ev_2, \ldots, Ev_j \} \) is the set of all given events, where \( Ev_j \) denotes the \( j \)th event in this set. \( N_j \) is the number of sub-events in \( Ev_j \) and the \( i \)th sub-event of \( Ev_j \) is denoted as \( Ev_{ji} \). Assume \( UAV_A \) is the set of all available UAVs.

The objective function to maximize the sum of the event communication time \( \chi^t_{ji} \) of each given \( Ev_j \) at each time \( t \) in the ideal scenario is formulated as follows:

\[
\max \sum_i \chi^t_{ji} \tag{10}
\]

Subject to:

\[
\text{com}_{ji} = \begin{cases} 
E_{u}^{\text{Bat}} \geq E^{\text{DTH}}, & \forall u \in UAV_A \\
\left( \frac{E_{u}^{\text{Bat}} - 2 \times E_{ji}^{\text{Fly}}}{E_{u}^{\text{Com}} + E_{u}^{\text{Ho}}} \right) > 0, & \forall u \in UAV_A 
\end{cases}
\]

\[
\chi^t_{ji} = \begin{cases} 
1, & \text{if UAV } u \text{ has been dispatched to serve } Ev_{ji} \text{ at time } t \\
0, & \text{Otherwise}
\end{cases}
\]

\[
\chi^t_j = \begin{cases} 
1, & \text{if } \sum_i \chi^t_{ji} = N_j, \forall Ev_{ji} \in Ev_j \\
0, & \text{Otherwise}
\end{cases}
\]

Equation (10) has four constraints. Constraint (1) is to guarantee that the energy \( E_{u}^{\text{Bat}} \) of any available UAV \( U_u \) has been charged to an energy level equal to or higher than threshold \( E^{\text{DTH}} \). Constraint (2) formulates the second constraint, where means the maximum communication time \( \text{com}_{ji} \) of any available UAV must be larger than zero. As mentioned in Section 2, the energy \( E_{u}^{\text{Rem}} \) for UAV \( U_u \) to consume on communication and hovering at the position of \( Ev_{ji} \) is the difference of \( E_{u}^{\text{Bat}} \) and \( 2 \times E_{ji}^{\text{Fly}} \). Hence, the maximum communication time \( \text{com}_{ji} \) of \( U_u \) for \( Ev_{ji} \) is equal to the quotient of \( E_{u}^{\text{Rem}} \) divided by the unit energy, \( E_{u}^{\text{Com}} + E_{u}^{\text{Ho}} \). Here, \( \chi^t_{ji} \) in Constraint (3) is a decision variable. If an available UAV, i.e., \( U_u \in UAV_A \), has been dispatched to serve sub-event \( Ev_{ji} \) at time \( t \), \( \chi^t_{ji} \) is set as 1. The last constraint is formulated in Constraint (4). If the sum of \( \chi^t_{ji} \) for each sub-event \( Ev_{ji} \) at time \( t \) is equal to the number of sub-events of \( Ev_j \), i.e., \( N_j \), which means that each
sub-event $Ev_{ij}$ at time $t$ has been served by a dispatched UAV, the communication link of $Ev_j$ at time $t$ has been recovered. Hence, $\chi_{tj}^j$ is set to 1. Otherwise, $\chi_{tj}^j$ is 0.

4. Performance Evaluation

In this paper, we adopt Eclipse [29] to run the simulation with parameters listed in Table 3. This simulation is conducted on the road network ($10.87 \text{ km} \times 10.60 \text{ km}$) of Shanghai, China, which is shown in Figure 8, by randomly choosing several road segments to suffer communication link failures, i.e., the event, in the disaster. The event percentage ($EP$) parameter in Table 3 is the ratio of road segments, denoted by two adjacent blue bullet points, suffered communication link failures in Figure 8. Hence, the number of events ($N_{ev} = N_{rs} \times EP$) in this simulation is equal to the product of the number of total road segments ($N_{rs}$) and the event percentage $EP$. According to [25], $\mu_{\text{LOS}}$ and $\mu_{\text{NLOS}}$ for calculating the value of Equations (4) and (5) are $1 \text{ dB}$ and $20 \text{ dB}$, respectively in an urban environment. Further, B and C in Equation (6) are set as 9.6 and 0.28, respectively. The following figures show average simulation results of five schemes on two metrics for twenty times. These five schemes are EAMUD with the Drop exception handling method ($EAMUD-D$), EAMUD with the Postpone method ($EAMUD-P$), EAUD with the Drop method ($EAUD-D$), EAUD with the Postpone method ($EAUD-P$) and the related work USP, respectively. Here, EAUD is a variant of EAMUD without performing the EBD algorithm in Step 5 when the number of UAVs needed for this sub-event is less than or equal to the number of available UAVs. Hence, EAUD is not able to check whether communication durations of UAVs already dispatched for all other sibling sub-events will overlap with the communication duration of the UAV dispatched for this sub-event. Consequently, it cannot achieve the high event communication time as EAMUD does. The following two metrics, i.e., the average ratio of event link communication time and its histogram, are depicted with respect to the UAV dispatch energy threshold, the number of UAVs and the event percentage. As mentioned in Section 2.3, if the UAV in the GCS has charged to an energy level, which is not less than the UAV dispatch energy threshold ($E_DTH$), it can be dispatched to serve a sub-event if needed. As mentioned in Section 3, if each sub-event $Ev_{ij}$ of $Ev_j$ at time $t$ can be served by a dispatched UAV, the communication link of $Ev_j$ at time $t$ is said to be recovered. Then, the event link communication time of $Ev_j$ is formulated by Equation (10) and its ratio of event link communication time is equal to $\left(\frac{\sum_{i=1}^{N_{ev}} T_{ev} \chi_{tj}^j}{N_{ev}}\right) / T_{ev}$.

Further, the average ratio of event link communication time is defined as the average ratio value of all events in the simulation, which is formulated as follows:

$$\frac{\sum_{i=1}^{N_{ev}} \left(\frac{\sum_{t=1}^{T_{ev}} \chi_{tj}^j}{T_{ev}}\right)}{N_{ev}}$$

(11)

where $T_{ev}$ is the event duration and $N_{ev} (= N_{rs} \times EP)$ is the number of events.

Figure 8. Shanghai map in the simulation.
Table 3. Simulation parameters.

| Item                                                                 | Value                       |
|----------------------------------------------------------------------|-----------------------------|
| Simulation time                                                      | 172,800 s                   |
| Event duration ($T_{ev}$)                                            | 57,600 s                    |
| The number of total road segments ($N_{rs}$)                         | 245                         |
| UAV flying speed ($S_u$)                                             | 10 m/s                      |
| UAV communication range ($d_{u2u}$)                                 | 300 m                       |
| U2U carrier frequency ($f_1$)                                       | 1 GHz                       |
| U2V carrier frequency ($f_2$)                                       | 5.9 GHz                     |
| The unit energy consumed on flying ($E_{Fly}^u$)                    | 115 J/s                     |
| Charge current ($I_{char}$)                                          | 3 A                         |
| Charge current voltage ($V_{char}$)                                 | 4.2 V                       |
| SNR threshold ($SNR_{thU2U}$)                                       | 2 dB                        |
| Noise powers of U2V and U2U ($N_{U2V}, N_{U2U}$)                    | $-80$ dBm                   |
| Required transmission powers of U2V and U2U ($P_{tU2V}, P_{tU2V}$)  | $27$ dBm                    |
| UAV dispatch energy threshold ($E_{DTH}$)                           | 50, 62.5, 75, 87.5 (default), 100% |
| Number of UAVs                                                       | 20, 40, 80 (default), 160, 240, 320, 400 |
| Event percentage ($EP$)                                              | 20, 30 (default), 40, 50, 60 % |
| UAV battery capacity ($E_{Max}$)                                     | 251,748 J                   |

4.1. Analytical Results of the Linear Programming Equations in the Ideal Scenario

In this section, we first use the linear programming equations formulated in Section 3 to calculate analytical results of the objective function, i.e., Equation (10) for the aggregate event communication time, as the performance benchmark in the ideal scenario. For solving the optimal result of Equation (10), information, like locations, numbers of sub-events, etc. of all events must be known in advance. Hence we assume that the number of all events and their sub-events are given using default values in Table 3. They happened at the same time and their locations are fixed. Here, we compare analytical optimal results of Equation (10), which are denoted as the OPT scheme below, and simulation results of the aggregate event communication time of EAUD and EAMUD. These analytical optimal results are solved by the JAVA API, provided by CPLEX [30], on Eclipse.

Figure 9 illustrates the average ratio of the event link communication time with respect to the number of UAVs in the ideal scenario. As the number of UAV raises, the probability that the GCS can dispatch UAVs to serve all sub-events of the same parent event becomes higher, which increases average ratio of the event link communication time accordingly for all schemes until the number of UAVs is high enough, i.e., larger than 240, to satisfy all UAV requirements in the event queue. When the number of UAVs is smaller than 240, OPT achieves the highest ratio of the event link communication time, then EAUD and EAMUD follow. This is because OPT can dispatch as more UAVs as possible to events having the highest event link communication time among all given events, according to four constraints of Equation (10). In this ideal scenario without considering UAV handovers and dispatch exceptions, EAUD and EAMUD achieve the same event link communication time, which are a little lower than those of OPT.

Figure 9. The average ratio of the event link communication time vs the number of UAVs in the ideal scenario.
Figure 10 exhibits the relation between the average ratio of the event link communication time and the event percentage. As the event percentage grows, all events and their corresponding sub-events in the event queue of GCS increase accordingly. Hence, the probability for dispatching a UAV to serve a sub-event is getting lower under a fixed number of UAVs, which in turn degrades the average ratio of the event link communication time of each scheme. Among these three schemes, OPT can choose a better event to dispatch UAVs when the event percentage, i.e., the number of events and sub-events, raises, such that it can achieve the highest ratio of the event link communication time. In summary, EAUD and EAMUD approximate performance behaviors and results of OPT with respect to the number of UAVs and the event percentage in the ideal scenario.

Figure 10. The average ratio of the event link communication time vs the event percentage in the ideal scenario.

4.2. Performance Results in the Real Scenario

In this section, we compare performance results of five schemes, which have to handle subsequent UAV handover processes and UAV dispatch exceptions, in the real scenario, where all events last for the fixed event duration ($T_{ev}$) and the starting time of each event is randomly chosen during the simulation time. Figure 11 shows how different UAV dispatch energy thresholds result in different average ratios of the event link communication time for five schemes. As the UAV dispatch energy threshold rises, the probability for UAVs in the GCS to become available to be dispatched is getting lower such that ratios of the event link communication time for USP and EAMUD-D decrease slowly. The reason for EAMUD-D to outperform USP is because whenever UAVs in the GCS are not available to be dispatched to the current sub-event, USP only drops this sub-event without dropping all other sibling ones as EAMUD-D does. Oppositely, ratios of the event link communication time of other schemes increase as the UAV dispatch energy threshold grows.

Figure 11. The average ratio of the event link communication time vs the UAV dispatch energy threshold.

As soon as EAUD-D cannot reserve a handover UAV in the GCS for a sub-event, it will drop this sub-event and all sibling ones but EAUD-P will postpone this sub-event and every sibling one with its processing time earlier than $t_i + t_{delay}$ to $t_i + t_{delay}$. By observing Figure 12a,b, which illustrates histograms of the average ratio of the event link communication time with respect to UAV dispatch energy threshold, EAUD-D has two peaks on the 0–0.05 and 0.85–1.0 ranges but more events with EAUD-P span on the larger 0.05–0.40 range.
Hence, EAUD-D achieves higher average ratios of the event link communication time than EAUD-P does, which is shown in Figure 11.

![Figure 13](image1)

**Figure 13.** Histograms of the average ratio of the event link communication time with respect to UAV dispatch energy threshold.

When the UAV dispatch energy threshold is smaller than 75%, EAMUD-P suffers the lower ratio of the event link communication time than EAUD-D. The reason of this behavior is because EAMUD-P postpones the sub-event, which cannot reserve a handover UAV, and every sibling sub-event with its processing time earlier than \( t_{ij} + t_{delay} \) to \( t_{ij} + t_{delay} \). Though EAMUD-P gives this sub-event a second chance to reserve a UAV, this postpone method has a higher probability of reserving the UAV, whose battery energy may just exceed the UAV dispatch energy threshold. Hence, this kind of UAV will drain out its battery energy soon after being dispatched such that the number of UAV handover grows, which results in the lower ratio of the event link communication time for EAMUD-P in this case. Oppositely, as the UAV dispatch energy threshold grows larger than 75%, the dispatched UAV owns higher battery energy to extend its communication duration. Further, as shown in Figure 12, EAMUD-D and EAMUD-P have similar behaviors as EAUD-D and EAUD-P on their histograms of the ratio of the event link communication time. As compared to EAMUD-D, EAMUD-P is able to distribute UAV resources to most events for raising the ratio of the event link communication time, especially for the larger UAV dispatch energy threshold when \( E^{DTH} = 87.5\% \) in Figure 12b. Consequently, EAMUD-P achieves the highest average ratio of the event link communication time among these five schemes.

Figure 13 shows the relation between the number of UAV and the average ratio of the event link communication time for five schemes. As the number of UAV raises, the probability that the GCS can dispatch UAVs to serve all sub-events of the same parent event becomes higher, which increases average ratio of the event link communication time accordingly for all schemes until the number of UAVs is high enough, i.e., larger than 240, to satisfy all UAV requirements in the event queue. As explained above, when the number of UAVs is smaller than 240, EAMUD-P achieves the highest ratio of the event link communication time, then EAUD-D/EAMUD-D and EAUD-P follow, but USP suffers the lowest one in Figure 13. Figure 14 illustrates histograms of the ratio of the event link communication time with respect to the number of UAVs. When the number of UAVs is
small, as shown in Figure 14a, EAUD-D has two peaks on the 0 and 0.95–1.0 ranges but most events with EAUD-P suffer from lower ratios of the event link communication time, which spans on the 0.05–0.30 range. Hence, EAUD-D achieves higher average ratios of the event link communication time than EAUD-P does, which is shown in Figure 13. Similarly, EAMUD-D also has two peaks on the 0 and 0.95–1.0 ranges as EAUD-D. However, most events with EAMUD-P achieve higher ratios of the event link communication time in the 0.65–0.90 range, which means that EAMUD-P is the best one among these five schemes to recover from the communication link failure in the disaster when the UAV resource is scarce. When the number of UAV grows larger than 240, the average ratio of the event link communication time of each scheme approaches 1 for all schemes, as shown in Figure 13, which can be justified by histograms of the ratio of the event link communication time, as shown in Figure 14b.

Figure 13. The average ratio of the event link communication time vs the number of UAVs.

Figure 14. Histograms of the average ratio of the event link communication time with respect to the number of UAVs.

Figure 15 exhibits the relation between the average ratio of the event link communication time and the event percentage. As the event percentage grows, all events and their corresponding sub-events in the event queue of GCS increase accordingly. Hence, the probability for dispatching a UAV to serve a sub-event is getting lower under a fixed number of UAVs, which, in turn, degrades the average ratio of the event link communication time of each scheme. As mentioned above, because EAMUD-P executes the EBD algorithm to check whether the communication link failure of an event can be recovered by overlapping
communication durations of UAVs already dispatched for all sibling sub-events and that of the UAV dispatched for current sub-event, its average ratio of the event communication time is highest among these five schemes.

![Figure 15. The average ratio of the event link communication time vs the event percentage.](image)

Figure 15. The average ratio of the event link communication time vs the event percentage.

Figure 16 illustrates histograms of the average ratio of the event link communication time with respect to the event percentage. As the event percentage grows from 30% to 60%, EAUD-D and EAMUD-D suffer from higher peaks on the 0–0.20 range, which means more events are unable to resume their link communication. Both of them have similar histograms in Figure 16a,b such that they have almost the same average ratio of the event link communication time, as shown in Figure 15. However, major events with EAUD-P span on the 0.05–0.40 range, but those with EAMUD-P distribute over the 0.65–0.95 range. These two histograms explain why the average ratio of the event link communication time of EAMUD-P is much higher than that of EAUD-P.

![Figure 16. Histograms of the average ratio of the event link communication time with respect to the event percentage.](image)

In summary, these three simulation parameters have different influences on these two metrics. The average ratio of event link communication time of every scheme is proportional to the number of UAVs but it is inversely proportional to the event percentage when other simulation parameters are fixed. On the other hand, only three schemes achieve higher values for the average ratio of event link communication time with the larger UAV dispatch energy threshold. As mentioned above, these performance behaviors can be justified by histograms of the average ratio of the event link communication time with respect to these parameters.
three parameters. Finally, two variants of the proposed EAMUD, i.e., EAMUD-P and EAMUD-D, perform best on the average ratio of the event link communication time among these five schemes in the real scenario. Whenever the UAV dispatch energy threshold is larger than 75%, EAMUD-P achieves the highest values of the average ratio of the event link communication time. Oppositely, EAMUD-D should be adopted as the UAV dispatching scheme when the UAV dispatch energy threshold is smaller than 75%.

As mentioned in the beginning of Section 4, the number of events \((N_{ev} = N_{rs} \times EP)\) in this simulation is equal to the product of the number of total road segments \((N_{rs})\) and the event percentage \(EP\). Hence, the total number of sub-events in the simulation is equal to the sum of the number of sub-events of each event, i.e., \(\sum_{j=1}^{N_{ev}} N_{j}\), where \(N_{j}\) is the number of sub-events of \(Ev_j\). In general, if two UAVs, one dispatched to serve the sub-event and the other one charged fast enough in GCS for replacing the dispatched one when needed, can be allocated to each sub-event, the average ratio of event link communication time can reach 1, as shown in Figure 13 when the number of UAVs is larger than 240. However, if the number of UAVs in GCS is lower than twice the total number of sub-events, the average ratio of event link communication time decreases faster as the ratio of the number of UAVs over the total number of sub-events becomes smaller, due to the more serious UAV contention and dispatch exception. We will try to model this complex relationship in the future.

5. Conclusions

In this paper, we proposed the mobile GCS to execute the energy-aware EAMUD with its associated UAV handover process such that it can dispatch multiple UAVs for all sub-events to recover the broken link and maximize the event communication time in the disaster area. With EBD, EAMUD is able to confirm that the UAV dispatch of current sub-events is feasible by assuring that communication durations of all dispatched UAVs are overlapped. We also proposed the Drop and Postpone exceptional handling methods. Further, we formulated the linear programming optimization equations to dispatch UAVs for given events in the ideal scenario. As compared to EAMUD-D, EAUD-P, EAUD-D, and the traditional USP, simulation results show EAMUD-P achieves the highest average ratio of the event link communication time with respect to different UAV dispatch energy thresholds, number of UAVs, and event percentages in the real scenario.

Author Contributions: Conceptualization, I.-C.C. and C.-S.L.; methodology, I.-C.C. and C.-S.L.; software, C.-S.L.; validation, I.-C.C., C.-S.L. and C.-E.Y.; formal analysis, C.-S.L. and C.-E.Y.; investigation, C.-S.L.; resources, I.-C.C.; data curation, C.-S.L.; writing—original draft preparation, I.-C.C. and C.-S.L.; writing—review and editing, I.-C.C. and C.-E.Y.; visualization, C.-S.L.; supervision, I.-C.C.; project administration, I.-C.C.; funding acquisition, I.-C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ministry of Science and Technology, Taiwan, grant number MOST 106-2221-E-018-008. The APC was funded by MOST 109-2221-E-018-017.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: We further exclude the Data Availability Statement because this study did not report any data.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.
Abbreviations

The following abbreviations are used in this manuscript:

- A2A: Air-to-air
- A2G: Air-to-ground
- EAMUD: energy-aware multi-UAV dispatch algorithm
- EAMUD-D: EAMUD with the Drop method
- EAMUD-P: EAMUD with the Postpone method
- EAUD-D: EAUD with the Drop method
- EAUD-P: EAUD with the Postpone method
- EBD: Event-Based Dispatching
- $E_v$: Event queue
- FANET: Flying Ad hoc Network
- G2A: Ground-to-air
- G2G: Ground-to-ground
- GCS: Ground control station
- LIP: Linear Integer Problem
- LoS: Line-of-sight
- MILP: Mixed integer linear programming
- NLoS: Non-line-of-sight
- SNR: Signal-to-noise ratio
- U2U: UAV-to-UAV
- U2V: UAV-to-Vehicle
- UAS: Unmanned Aerial System
- UAV: unmanned aerial vehicle
- VANET: Vehicular ad hoc network

References

1. Sahingoz, O.K. Networking models in flying ad-hoc networks (FANETs): Concepts and challenges. *J. Intell. Robot. Syst.* 2014, 74, 513–527. [CrossRef]
2. Menouar, H.; Guvenc, I.; Akkaya, K.; Uluagac, A.S.; Kadri, A.; Tuncer, A. UAV-enabled intelligent transportation systems for the smart city: Applications and challenges. *IEEE Commun. Mag.* 2017, 55, 22–28. [CrossRef]
3. Aasen, H. UAV spectroscopy: Current sensors, processing techniques and theoretical concepts for data interpretation. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 8809–8812.
4. Rosner, D.; Trifu, C.; Tranca, C.; Vasilescu, I.; Stancu, F. Magnetic field sensor for UAV power line acquisition and tracking. In Proceedings of the 17th RoEduNet Conference: Networking in Education and Research, Cluj-Napoca, Romania, 6–8 September 2018; pp. 1–5.
5. Zhao, H.; Wang, H.; Wu, W.; Wei, J. Deployment algorithms for UAV airborne networks toward on-demand coverage. *IEEE J. Sel. Areas Commun.* 2018, 36, 2015–2031. [CrossRef]
6. Sharma, V.; Srinivasan, K.; Chao, H.-C.; Hua, K.-L.; Cheng, W.-H. Intelligent deployment of UAVs in 5G heterogeneous communication environment for improved coverage. *J. Netw. Comput. Appl.* 2017, 85, 94–105. [CrossRef]
7. Qureshi, K.N.; Abdullah, A.H.; Lloret, J.; Altameem, A. Road-aware routing strategies for vehicular ad hoc networks: Characteristics and comparisons. *Int. J. Distrib. Sens. Netw.* 2016, 16, 1–19. [CrossRef]
8. Hajlaoui, R.; Guyennet, H.; Moulahi, T. A survey on heuristic-based routing methods in vehicular ad-hoc network: Technical challenges and future trends. *IEEE Sens. J.* 2016, 16, 6782–6792. [CrossRef]
9. Huang, C.-F.; Jhang, J.-H. Efficient RSU selection approaches for load balancing in vehicular ad hoc networks. *Adv. Technol. Innov.* 2020, 5, 56–63. [CrossRef]
10. Zhou, Y.; Cheng, N.; Lu, N. Multi-UAV-aided networks: Aerial-ground cooperative vehicular networking architecture. *IEEE Veh. Technol. Mag.* 2015, 10, 36–44. [CrossRef]
11. Wang, X.; Fu, L.; Zhang, Y.; Gan, X.; Wang, X. VDNet: An infrastructure-less UAV-assisted sparse VANET system with vehicle location prediction. *Wirel. Commun. Mob. Comput.* 2016, 16, 2991–3003. [CrossRef]
12. Ouibbati, O.S. CRUV: Connectivity-based traffic density aware routing using UAV for VANETs. In Proceedings of the International Conference on Connected Vehicles and Expo (ICCVE), Shenzhen, China, 19–23 October 2015; pp. 68–73.
13. Ouibbati, O.S.; Lakas, A.; Lagraa, N. UVAR: An Intersection UAV-assisted VANET Routing Protocol. In Proceedings of the IEEE Wireless Communications and Networking Conference, Doha, Qatar, 3–6 April 2016; pp. 1–6.
14. Yim, J.; Park, H.; Kwon, E.; Kim, S.; Lee, Y.-T. Low-power image stitching management for reducing power consumption of UAVs for disaster management system. In Proceedings of the IEEE International Conference on Consumer Electronics, Las Vegas, NV, USA, 12–14 January 2018; pp. 1–3.
16. Ezequiel, C.A.F.; Cua, M.; Libatique, N.C.; Tangonan, G.L.; Alampay, R.; Labuguen, R.T.; Favila, C.M.; Honrado, J.L.E.; Canos, V.; Devaney, C.; et al. UAV aerial imaging applications for post-disaster assessment, environmental management and infrastructure development. In Proceedings of the International Conference on Unmanned Aircraft Systems, Orlando, FL, USA, 27–30 May 2017; pp. 274–283.

17. Rokhmana, C.A.; Andaru, R. Utilizing UAV-based mapping in post disaster volcano eruption. In Proceedings of the International Annual Engineering Seminar, Yogyakarta, Indonesia, 1–3 August 2016; pp. 202–205.

18. Erdelj, M.; Natalizio, E. UAV-assisted disaster management: Applications and open issues. In Proceedings of the International Conference on Computing, Networking and Communications, Kauai, HI, USA, 15–18 February 2016; pp. 1–5.

19. Hayat, S.; Yanmaz, E.; Muzaffar, R. Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. *IEEE Commun. Surv. Tutor.* 2016, 18, 2624–2661. [CrossRef]

20. Jeong, D.; Park, S.Y.; Lee, H.J. DroneNet: Network reconstruction through sparse connectivity probing using distributed UAVs. In Proceedings of the IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications, Hong Kong, China, 30 August–2 September 2015; pp. 1797–1802.

21. Erdelj, M.; Natalizio, E.; Chowdhury, K.R. Help from the sky: Leveraging UAVs for disaster management. *IEEE Pervasive Comput.* 2017, 16, 24–32. [CrossRef]

22. Ghazzai, H.; Kadri, A.; Ghorbel, M.B.; Menouar, H. Optimal sequential and parallel UAV scheduling for multi-event applications. In Proceedings of the IEEE 87th Vehicular Technology Conference, Porto, Portugal, 3–6 June 2018; pp. 1–6.

23. Erdelj, M.; Król, M.; Natalizio, E. Wireless sensor networks and multi-UAV systems for natural disaster management. *Comput. Netw.* 2017, 124, 72–86. [CrossRef]

24. Palmer, J.; Yuen, N.; Ore, J.-P. On air-to-water radio communication between UAVs and water sensor networks. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 5311–5317.

25. Jia, S.; Zhang, L. Modelling unmanned aerial vehicles base station in ground-to-air cooperative networks. *IET Commun.* 2017, 11, 1187–1194. [CrossRef]

26. Erdelj, M.; Saif, O.; Natalizio, E.; Fantoni, I. UAVs that fly forever: Uninterrupted structural inspection through automatic UAV replacement. *Ad Hoc Netw.* 2017, 94, 101612. [CrossRef]

27. Dietrich, T.; Andryeyev, O.; Zimmermann, A.; Mitschele-Thiel, A. Towards a unified decentralized swarm management and maintenance coordination based on MAVLink. In Proceedings of the International Conference on Autonomous Robot Systems and Competitions, Bragança, Portugal, 4–6 May 2016; pp. 124–129.

28. Motlagh, N.H.; Bagaa, M.; Taleb, T. UAV selection for a UAV-based integrative IoT platform. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA, 4–8 December 2016; pp. 1–6.

29. Eclipse IDE for Java Developers. Available online: https://www.eclipse.org/downloads/packages/release/2020-09/r/eclipse-ide-java-developers (accessed on 31 October 2020).

30. Cavalcante, T.R.; Bessa, F.L.V.; Cordeiro, L.C. Planning and evaluation of UAV mission planner for intralogistics problems. In Proceedings of the Brazilian Symposium on Computing Systems Engineering (SBESC), Paraná, Brazil, 7–10 November 2017; pp. 9–16.