2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)

To Route or To Ferry: A Hybrid Packet Forwarding Algorithm in Flying Ad Hoc Networks

Cong Pu Logan Carpenter
Weisberg Division of Computer Science
Marshall University, Huntington, WV 25755, USA
Email: {puc, carpenter190}@marshall.edu

***Abstract*—The capabilities and roles of unmanned aerial vehicles, a.k.a. drones, have been rapidly evolving as a result of the advances in processing, sensing, communicating, and networking technologies of robotic systems. Because of the versatility, flexibility, easy installation, and relatively small operating expenses of drones, Flying Ad Hoc Networks (FANETs) consisting of a fleet of drones endowed with sensing, computing, and wireless communicating capabilities are promptly proliferating and representing a key enabler for Internet-of-Drones and its applications. Unfortunately, reliable packet forwarding in FANETs is not always guaranteed because of unstable link quality and intermittent connectivity caused by high mobility of drones. In this paper, we propose a hybrid packet forwarding algorithm, named $HYBD^{fwd}$, to efficiently and reliably deliver data packets to ground destination in FANETs. The $HYBD^{fwd}$ consists of two schemes: end-to-end routing and delay-tolerant forwarding. In end-to-end routing, the drone initiates a route discovery procedure to find an end-to-end routing path to deliver data packets to ground destination. In case no end-to-end routing path exists, delay-tolerant forwarding is applied, where the drone forwards data packets to the ferry drone that is moving to ground destination or it carries data packets and moves to ground destination to deliver data packets. We evaluate the proposed hybrid packet forwarding algorithm through extensive simulation experiments using OMNeT++ and compare its performance with a prior motion-driven packet forwarding algorithm, and experimental results indicate that the proposed hybrid packet forwarding algorithm can be a viable approach in FANETs.**

***Index Terms*—Unmanned Aerial Vehicles, Drones, Packet Forwarding Algorithm, Flying Ad Hoc Networks, Internet-of-Drones**

I. Introduction

Unmanned aerial vehicles (UAVs), often referred to as drones, have been increasingly popular among hobbyists, researchers, and investors. The U.S. Federal Aviation Administration projects that the country's commercial, small, non-model fleet of drones will grow to 451,800 in 2022, and small model hobbyist drones will be doubled from an estimated 1.1 million vehicles in 2017 to 2.4 million units by 2022. The economic implications for commercial drone industry, including aerial surveying and mapping, aerial surveillance and security, aerial inspection of infrastructure, aerial delivery, etc., are also indisputable in the U.S. It has been estimated that drone integration within national air space will account for $82.1 billion in job creation and economic growth over the 10-year span from 2015 to 2025. As the number of drones rapidly increases, Internet-of-Drones (IoD) and its applications are expeditiously proliferating, where a myriad of multi-sized and heterogeneous drones seamlessly interact with each other through zone service providers to realize the goal of coordinating the access of drones to controlled airspace and providing navigation services [1]. With the ongoing advances in processing, sensing, communicating, and networking technologies, drones will find many new ways to improve our lives further in the realm of IoD.

978-1-7281-2522-0/19/$31.00

Drones are small flying robots with the capabilities of photographing, sensing, computing, and wireless communicating, and ready to gather information and transmit often large-sized data to a ground station [2]. The communication of a single drone scenario has been well investigated, for example, a load-carry-and-deliver single-hop routing protocol has been proposed to utilize a single drone to relay messages between two distant ground locations [3]. However, as a major part of speedily emerging IoD, Flying Ad Hoc Networks (FANETs) with the advantages of faster multitasking capability, longer network lifetime, and higher scalability pose new research challenges. For example, how a fleet of drones efficiently and collaboratively route data packets to a destination in order to achieve the goal of sharing information and knowledge and coordinating decisions. Due to the unique characteristics of FANETs, such as high mobility and continuous movement, drastically changing network topology, and intermittently connected communication links, routing protocols and communication algorithms that were specifically designed for Mobile Ad Hoc Networks (MANETs) and Vehicular Ad Hoc Networks (VANETs) fail in the highly dynamic aerial environment [4], [5]. For example, wireless connectivity among drones is challenged by the high mobility of drones, which leads to continuous variation of mutual distances between drones and drastic network topology changes, resulting in a significant degradation of network performance [6], [7].

In this paper, we propose a novel hybrid packet forwarding algorithm to efficiently and reliably deliver data packets to ground destination in FANETs. Unlike prior routing schemes such as traditional source routing (e.g., DSR), distance vector routing (e.g., AODV), or link state routing (e.g., OLSR) that solely rely on either end-to-end routing path or a certain degree of link stability, the proposed hybrid packet forwarding algorithm incorporates end-to-end routing technique with delay-tolerant forwarding operations to efficiently and reliably deliver data packets. Our major contribution is briefly summarized in twofold:

- We categorize and analyze the existing routing protocols and communication algorithms in FANETs in terms of static routing, proactive routing, reactive routing, hybrid routing, position-based routing, cluster routing, and other approaches. The main motivation of categorization and comparative analysis is to help network engineers choose appropriate routing protocols based on specific scenario where FANETs will be deployed.
- We propose a hybrid packet forwarding algorithm, also referred to as $HYBD^{fwd}$, which consists of end-to-end routing scheme and delay-tolerant forwarding scheme, to efficiently and reliably deliver data packets to ground destination in FANETs. In end-to-end routing, the drone initiates a route discovery procedure to find an end-to-end routing path to deliver data packets to ground destination. In case no end-to-end routing path exists, delay-tolerant forwarding is applied, where the drone forwards data packets to the ferry drone that is moving to ground destination or it carries data packets and moves to ground destination to deliver data packets.

We develop a customized discrete event driven simulation framework by using OMNeT++ [8] and evaluate its performance through extensive simulations in terms of packet delivery ratio, packet delivery latency, and number of ferried data packets. We also revisit prior motion-driven packet forwarding algorithm [9], and modify it to work in FANETs for performance comparison. The simulation results show that the proposed hybrid packet forwarding algorithm can not only improve the packet deliver ratio, but also reduce the packet delivery latency, indicating a viable approach in FANETs.

The rest of paper is organized as follows. Prior schemes are provided and analyzed in Section II. A system model and the proposed hybrid packet forwarding algorithm are presented in Section III. Section IV focuses on simulation results and their analyses. Finally, concluding remarks with future research direction are provided in Section V.

II. Related Work

In this section, we categorize and analyze the existing routing protocols and communication algorithms in FANETs and similar environments in terms of static routing, proactive routing, reactive routing, hybrid routing, position-based routing, cluster routing, and other approaches.

Static Routing: In static routing, drones are used as packet carriers which transfer packets between source and destination. Before the task starts, a static routing table is computed and loaded to drones, and cannot be updated during the task operation. The advantage of static routing is lightweight, however, the drawbacks are also obvious that the systems deployed with static routing are not fault tolerant or suitable for dynamically changing environment. In [10], drones are used as message ferries to store, carry, and forward messages to destination when it is impossible or impractical to directly transmit the messages through intermediate nodes in highly partitioned ad hoc networks. Since static routing protocol is not fault tolerant and appropriate for dynamic environments, the route planning problem is of great importance to drones.

Proactive Routing: Proactive routing is also called active routing, where routing information is updated and shared periodically among drones in advance. Thus, the routing path between every pair of drones in the network can be selected to transmit packets immediately without a long waiting time. However, the disadvantages are also undeniable, for example, a large amount of control packets are needed to keep the routing information up-to-date, which introduces a higher communication overhead. In addition, proactive routing protocols are not suitable for high-mobility networks. In [11], a speed-aware predictive-optimized link state routing protocol (P-OLSR) exploiting GPS information is proposed to assist routing operations in FANETs, where the relative speed between two drones can be obtained based on GPS information, and is taken into account as a factor in the calculation of the expected transmission count metrics.

Reactive Routing: Reactive routing is also called on-demand routing, where a routing path is built on demand when packets need to be sent. Compared to proactive routing, reactive routing has a lower communication overhead by effectively reducing the number of control packets, but introduces a higher communication latency due to the discovery of end-to-end routing path. Dynamic source routing (DSR) [12] is a classic reactive routing protocol for multi-hop wireless mesh networks, where a source node first floods a route request packet throughout the network when it has data packets to send. When the destination node receives the route request packet, it replies a route reply packet piggybacked with the complete route of the destination node to source node. Ad hoc on-demand distance vector (AODV) routing [13] is another representative reactive routing, which is designed based on DSR and DSDV [14]. It uses the periodic beaconing and sequence numbering procedure of DSDV and a similar route discovery procedure as in DSR.

Hybrid Routing: Hybrid routing is a combination of proactive routing and reactive routing to overcome the problem of high control message overhead in proactive routing and high end-to-end delay in reactive routing. In [15], a hybrid routing framework suitable for a wide variety of mobile ad hoc networks is proposed, where each node proactively maintains routes within a local region, also referred to as the routing zone. The knowledge of routing zone topology is leveraged by routing framework to improve the efficiency of a globally reactive route query/reply mechanism. In [16], a routing protocol named rapid-reestablish temporally ordered routing algorithm (RTORA) is proposed for FANETs, where a reduced-overhead mechanism is adopted to overcome adverse effects caused by link reversal failure. In the reduced-overhead mechanism, a large amount of useless control packets from flooding are prevented.

Position-based Routing: In position-based routing, the geographic position of destination contained in the packet or the position of neighbor nodes of forwarding node is utilized to make routing decision. In [17], a geographic position mobility oriented routing, called GPMOR, is proposed for FANETs, where the movement of drones is predicted with Gaussian-Markov mobility model to eliminate the impact of highly dynamic movement. The GPMOR selects the next-hop node according to the mobility relationship in addition to Euclidean distance to make more accuracy decision.

Cluster Routing: Cluster routing places drones into groups, also called clusters, and performs hierarchical routing between these clusters. The routing is primarily established with some proactive planned routes at the higher levels, and then helps the request from triggered drones through reactive routing at the lower levels. In [18], the concept of multicluster FANETs employing IEEE 802.15.4 MAC layer protocol for drone-to-drone communication is proposed to reduce communication cost and optimize network performance.

Other Approaches: The [19] proposes adaptive communication protocols including a position-prediction-based directional MAC protocol and a self-learning routing protocol based on reinforcement learning in FANETs. In [20], a predictive routing protocol based on three-dimensional estimation with a fast update mechanism for the flying path is proposed in FANETs, where prediction mechanism is employed to determine the drone location and its trajectory to enhance the efficiency of routing protocol. In [21], an aerial network management protocol built on top of a software defined networking (SDN) architecture is proposed to address the needs of efficient and robust end-to-end data relaying in FANETs, where each drone becomes a SDN switch that performs under directives sent by a centralized controller.

In summary, over the past few years, various routing protocols and communication mechanisms have been proposed in FANETs and similar environments. Through careful analysis and comparison, it is found that each protocol has its own definite strengths and weaknesses, and suitable for specific situation. Most prior approaches focus on the load-carry-and-deliver, the shortest path, the path with the best link quality and lightest traffic load, the geographic position of destination or next-hop node, or mobility prediction. However, little attention has been paid to hybrid packet forwarding algorithm incorporating with end-to-end routing and delay-tolerant forwarding in FANETs.

III. The Proposed Hybrid Packet Forwarding Algorithm

In this section, we first introduce system model and assumptions, link expiration time model, and link throughput model, and then propose a hybrid packet forwarding algorithm, called $HYBD^{fwd}$, to efficiently and reliably deliver data packets to ground destination in FANETs.

A. System Model and Assumptions

In this paper, a search and rescue mission in a post-disaster area is considered, where a set of drones identified by its drone id is deployed to inspect the targeted area and provide high resolution overview images in order to spot a missing person or an object [22]. Due to the limited capability of on-board computer of drones, the collected real-time aerial photography are sent to a stationary ground destination for further processing in time. We assume that each drone is equipped with a Global Positioning System, Inertial Measurement Units, and digital map to obtain its current geographical position and mobility information [23]. In most of drone-based services and applications, drones like small quad-copters usually do not fly at high altitudes [24], therefore, we assume that all drones have the same constant and low altitude during the flight. We also assume that drones have no energy restrictions since they are equipped with rechargeable batteries which can be recharged from wireless recharging stations [1] or environmental energy resources [25].

It is a well-known fact that separating data and control channels in wireless networks gives rise to performance benefits [9]. In addition, drones usually require a reliable communication channel for control and telemetry data for safe flight operation, so the existence of an out-of-band-channel is a reasonable assumption. Thus, it is assumed that each drone uses high-throughput radio technology with the limited communication range and low-throughput channel with long communication range for data traffic and control traffic, respectively. Through periodically exchanged drone status (i.e., relaying or ferrying, and location and motion information) control messages transmitted via the low-throughput control traffic channel, the ground destination is aware of current status of all drones and the global network topology. For the vast majority of outdoor mission, for example search and rescue operations, drones can be considered to operate in flat areas, thus, wireless links can be assumed to have a free space propagation model and line-of-sight characteristics. In other words, the received signal strength solely depends on the distance between the sender and the receiver. In order to reduce high deployment and operational costs, sparse drone (later node) deployment is considered in FANETs. In addition, motion speed and heading direction information can be obtained from equipped Global Positioning System and Inertial Measurement Units, which will be used to estimate the link expiration time as well as the link throughput.

B. Modeling Link Expiration Time and Link Throughput

First, we propose a link expiration time (LET) prediction method utilizing the geographical location and mobility information based on [26]. We assume that two nodes n_i and n_j are located within the communication range r of each other, and the two-dimensional position coordinates of n_i and n_j are denoted as (x_i, y_i) and (x_j, y_j), respectively. We also suppose that v_i and v_j are the moving speeds, and θ_i and θ_j ($0 \leq \theta_i$, $\theta_j < 2\pi$) are the moving directions of n_i and n_j, respectively. According to [26], the duration of time that n_i and n_j will stay connected, denoted by $T_{cont}^{\{i,j\}}$, is represented by

$$T_{cont}^{\{i,j\}} = \frac{-(a \cdot b + c \cdot d) + \sqrt{(a^2 + c^2) \cdot r^2 - (a \cdot d - b \cdot c)^2}}{a^2 + c^2}, \quad (1)$$

where

$$a = v_i \cdot \cos\theta_i - v_j \cdot \cos\theta_j, \quad (2)$$

$$b = x_i - x_j, \tag{3}$$

$$c = v_i \cdot \sin\theta_i - v_j \cdot \sin\theta_j, \tag{4}$$

$$d = y_i - y_j. \tag{5}$$

Note that when n_i and n_j are moving at the same speed and direction, where $v_i = v_j$ and $\theta_i = \theta_j$, $T_{cont}^{\{i,j\}}$ becomes ∞. Thus, the predicted connection time $T_{cont}^{\{i,j\}}$ is the link expiration time (LET) between n_i and n_j.

In addition, according to [9], the link throughput can be measured as a function of the known geographical distance between the packet sender and the packet receiver. Thus, the derived empirical link throughput between node n_i and node n_j, denoted by $T_{put}^{\{i,j\}}$, is given as follows

$$T_{put}^{\{i,j\}} = 10^6 \cdot (-9.09 \cdot \log_2(dist(i,j)) + 72.58), \tag{6}$$

where $dist(i,j)$ is the spatial distance between n_i and n_j. Here, the unit of spatial distance $dist(i,j)$ and link throughput $T_{put}^{\{i,j\}}$ is meter and bit/second, respectively.

C. Hybrid Packet Forwarding Algorithm

In this subsection, we propose a hybrid packet forwarding algorithm, also referred to as $HYBD^{fwd}$, to efficiently and reliably deliver data packets to ground destination in FANETs. In the $HYBD^{fwd}$, end-to-end routing scheme is first applied to find the potential end-to-end routing path with the maximum output between source node and ground destination. In case the network is partitioned and no end-to-end routing path exists, delay-tolerant forwarding scheme is used, where the source node forwards data packets to the ferry node with a higher data ferry rate that is moving to ground destination to deliver data packets. If there is no available ferry node with a higher data ferry rate, the source node carries data packets from its current position and physically moves to ground destination to deliver data packets.

1) End-to-End Routing Scheme: When the source node has data packets to deliver to ground destination, it initiates the route discovery procedure by broadcasting a route request (RREQ) packet. The RREQ packet contains source node id (ID_{src}), packet sequence number (PKT_{seq}), source route record (S_{route}), list of position coordinates of nodes in source route record (L_{coord}), list of moving speeds of nodes in source route record (L_{spd}), and list of moving directions of nodes in source route record (L_{dir}). Any intermediate node located between the source node and the ground destination receives a RREQ packet for the first time, it caches the packet sequence number PKT_{seq} along with a timestamp when the RREQ packet is received. To save storage space, the cached PKT_{seq} with the timestamp less than T_{cur} - ϖ is evicted, where T_{cur} and ϖ are the current system time and a system parameter, respectively. In addition, it appends its node id in the source route record S_{route}, adds its position coordinate in the list of position coordinates L_{coord}, adds its moving speed in the list of moving speeds L_{spd}, adds its moving direction in the list of moving directions L_{dir}, and then rebroadcasts the RREQ packet. When an intermediate node receives a duplicated RREQ packet, it first checks whether its node id is in the piggybacked source route record S_{route} of the received RREQ packet or not. If its node id is in the S_{route}, it drops the received RREQ packet directly. Otherwise, it appends its node id in the S_{route}, adds its position coordinate, moving speed, and moving direction in the L_{coord}, L_{spd}, and L_{dir}, respectively, and then rebroadcasts the RREQ packet. When the intermediate node that has a direct connection to ground destination (or is within the communication range of ground destination) receives the RREQ packet, it appends its node id in the S_{route}, adds its position coordinate, moving speed, and moving direction in the L_{coord}, L_{spd}, and L_{dir}, respectively. And then, it replies a route reply (RREP) packet piggybacked with the PKT_{seq}, S_{route}, L_{coord}, L_{spd}, and L_{dir} back to the source node along the reverse route of the RREQ packet.

When the source node receives the first RREP packet, it records the piggybacked PKT_{seq}, S_{route}, L_{coord}, L_{spd}, and L_{dir} in the route table. In addition, the source node starts a timer and waits for a certain time period, T_{wait}, to receive more RREP packets and learn all possible routes. When T_{wait} expires, based on the recorded S_{route}, L_{coord}, L_{spd}, and L_{dir} in the route table, the source node calculates the link expiration time and the link throughput of all links along each candidate route in the route table according to Eq. 1 and 6, respectively. And then, the source node calculates an estimated route output of each candidate route in the route table. In this paper, the route output represents the maximum amount of data that can be transmitted before the candidate route is broken and becomes invalid, and is the comprehensive judgment factor of the candidate route in terms of link expiration time and link throughput. The estimated route output of the candidate route $\{route\}$ can be represented as $R_{output}^{\{route\}} = T_{cont^{min}}^{\{n_i,n_j\}} \cdot T_{put^{min}}^{\{n_k,n_m\}}$. Here, $T_{cont^{min}}^{\{n_i,n_j\}}$ and $T_{put^{min}}^{\{n_k,n_m\}}$ is the smallest link expiration time between node n_i and node n_j, and the smallest link throughput between node n_k and node n_m along the candidate route $\{route\}$, respectively. Based on the calculated $R_{output}^{\{route\}}$, the source node selects the route with the largest $R_{output}^{\{route\}}$ as the forwarding path, and then sends data packets to ground destination.

Certain link of the route can be disconnected frequently because of high mobility. In the $HYBD^{fwd}$, if a node continuously fails to deliver data packets to next-hop node along the forwarding path, i.e., not overhearing implicit acknowledgment or receiving explicit acknowledgment [27], it considers the link to be disconnected and sends a route error (RERR) packet piggybacked with disconnected link to source node. Upon receiving the RERR packet, the source node stops sending data packets along the broken route, and then initiates the route discovery procedure again to find a new route to deliver data packets or apply delay-tolerant forwarding scheme if end-to-end routing path does not exist. In classic reactive routing protocols, for example DSR [12], route caching through unconditional overhearing is one of the major features to improve routing performance [28]. Whenever an intermediate node forwards or overhears a packet, such as RREQ, RREP, or data

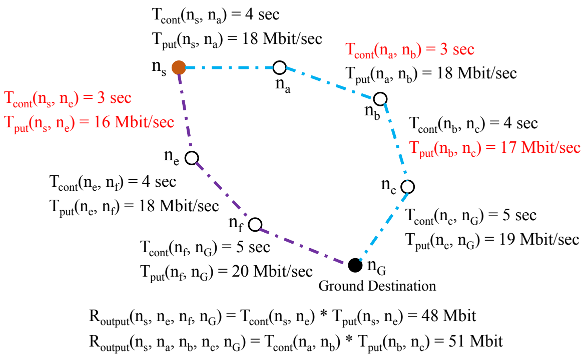


Fig. 1. A snapshot of the network, where two end-to-end routing paths, denoted by color purple and blue, are available between source node n_s and ground destination n_G. $T_{cont}(n_i, n_j)$ and $T_{put}(n_i, n_j)$ is the link expiration time and the link throughput between node n_i and node n_j, respectively.

packet, it caches the piggybacked route information in its route table. If an intermediate node forwards or overhears a RERR packet, it removes any cached route containing the broken link from its route table. However, in this paper, the overhearing feature is not considered. In particular, intermediate nodes are not designed to cache the piggybacked route information when they receive or overhear any on-flying packets. The rational behind this design is that the nodes in FANETs have high mobility and the network topology changes frequently, thus, the cached routes can become unavailable and stale quickly.

For example, as shown in Fig. 1, suppose that two end-to-end routing paths are available between source node n_s and ground destination n_G, and source node n_s has already received the RREP packets along these two routes. Thus, n_s first computes the link expiration time and the link throughput of all links along these two routes, and then calculates the estimated route output of each route. Here, $R_{output}^{\{n_s,n_a,n_b,n_c,n_G\}} = T_{contmin}^{\{n_a,n_b\}} * T_{putmin}^{\{n_b,n_c\}}$ = 51 Mbit, and $R_{output}^{\{n_s,n_e,n_f,n_G\}} = T_{contmin}^{\{n_s,n_e\}} * T_{putmin}^{\{n_s,n_e\}}$ = 48 Mbit, respectively. Thus, the route $\{n_s, n_a, n_b, n_c, n_G\}$ with the larger estimated route output will be chosen as forwarding path. The major operations of the proposed end-to-end routing scheme is shown in Fig. 2.

2) Delay-Tolerant Forwarding Scheme: Since geographic position and mobility information of nodes are available, routing concepts employing location and motion awareness could be potential packet forwarding candidate in FANETs, where data packets are forwarded to nodes that are physically moving closer to ground destination. In this paper, not only nodes can act as communication relay nodes in end-to-end routing, but also they may act as communication ferry nodes that carry and move data packets physically to ground destination [29]. Thus, we propose a delay-tolerant forwarding scheme to deliver data packets to ground destination in case no end-to-end routing path is available.

When the source node has data packets to deliver, however, the route discovery procedure fails to find the end-to-end routing path to ground destination, the source node first checks whether there is a ferry node that is currently moving to ground destination within its communication range. If there is a ferry node within its communication range, the source

---

Notations:

• ID_{src}, PKT_{seq}, S_{route}, L_{coord}, L_{spd}, L_{dir}, T_{wait}, and $R_{output}^{\{route\}}$: Defined before.

• $RREQ[ID_{src}, PKT_{seq}, S_{route}, L_{coord}, L_{spd}, L_{dir}]$, $RREP$, $RERR$, $cache^i_{seq}$, $coord_i$, spd_i, dir_i, RT_i, and CR_i: *RREQ* packet, *RREP* packet, *RERR* packet, cached packet sequence numbers of *RREQ* packets at node n_i, position coordinate of node n_i, moving speed of node n_i, moving direction of node n_i, route table of node n_i, and communication range of node n_i.

Event-Driven End-to-End Routing Scheme:

◇ When a source node n_s has data packets to deliver:
  Broadcast *RREQ* packet;
◇ When an intermediate node n_i receives a *RREQ* packet:
  if n_i is within the CR_G of ground destination n_G
    Append n_i in $RREQ.S_{route}$; Add $coord_i$ in $RREQ.L_{coord}$;
    Add spd_i in $RREQ.L_{spd}$; Add dir_i in $RREQ.L_{dir}$;
    Reply $RREP$ packet with $RREQ.\{S_{route}, L_{coord}, L_{spd}, L_{dir}\}$;
  else
    if $RREQ.PKT_{seq} \notin cache^i_{seq}$ /∗ Receive $RREQ$ first time. ∗/
      Cache PKT_{seq} in $cache^i_{seq}$;
      Append n_i in $RREQ.S_{route}$; Add $coord_i$ in $RREQ.L_{coord}$;
      Add spd_i in $RREQ.L_{spd}$; Add dir_i in $RREQ.L_{dir}$;
      Rebroadcast $RREQ$ packet;
    else /∗ Receive duplicated $RREQ$. ∗/
      if $n_i \in RREQ.S_{route}$
        Discard $RREQ$ packet;
      else
        Append n_i in $RREQ.S_{route}$; Add $coord_i$ in $RREQ.L_{coord}$;
        Add spd_i in $RREQ.L_{spd}$; Add dir_i in $RREQ.L_{dir}$;
        Rebroadcast $RREQ$ packet;
◇ When a source node n_s receives a *RREP* packet:
  Cache $RREQ.S_{route}$ in RT_s; Start a timer T_{wait};
◇ When timer T_{wait} expires at n_s:
  for each $\{route\} \in RT_s$
    Compute $R_{output}^{\{route\}}$;
  Send data packets along $\{route\}$ with the maximum $R_{output}^{\{route\}}$;
◇ When an intermediate node n_i detects a link failure of $\{route\}$:
  Send $RERR^{\{route\}}$ packet back to source node;
◇ When a source node n_s receives a $RERR^{\{route\}}$ packet:
  Stop sending data packet along $\{route\}$;
  Initiate route discovery procedure;

Fig. 2. The pseudocode of the proposed end-to-end routing scheme.

node sends a request-to-send (RTS) packet to the ferry node. When the ferry node receives the RTS packet, it replies a clear-to-send (CTS) packet. After receiving the CTS packet, the source node forwards its data packets to the ferry node who will carry and deliver source node's data packets along with its own data packets to ground destination. For example, as shown in Subfig. 3(a), suppose that there is no end-to-end routing path to ground destination n_G. However, there is a ferry node n_f within the communication range of source node n_s. Instead of carrying data packets and moving to ground destination n_G by itself, n_s can forward its data packets to n_f that is moving to ground destination n_G to deliver data packets. Thus, n_s and n_f first exchange RTS and CTS packets, and then, n_s forwards its data packets to n_f, which will carry and deliver data packets to ground destination n_G. By doing so, n_s can save its energy and spend more time on mission-driven tasks instead of consuming them by moving closer to ground destination n_G for data packet delivery.

If there is no ferry node within the communication range of the source node, the ground destination can instruct a nearby ferry node to change its original air route and move closer to the source node to load data packets. First, the ground destination calculates the data ferry rate of rerouting the ferry node to move to the source node to load data packets and then move to ground destination for data packet delivery. In addition, the ground destination calculates the sum of the data

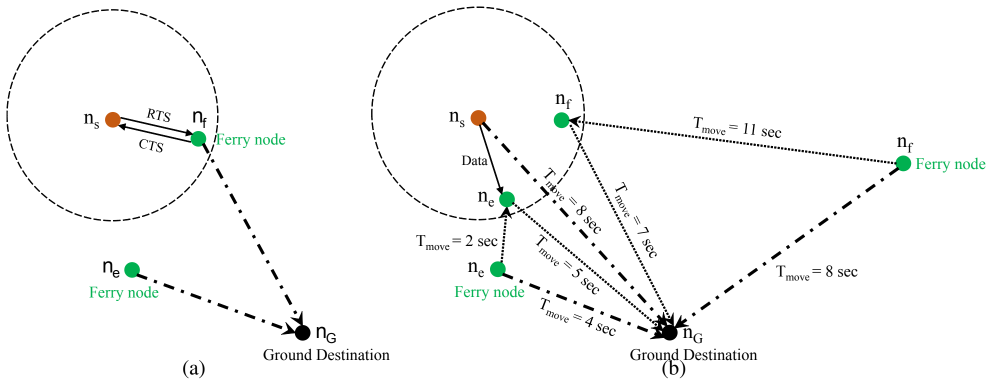


Fig. 3. The proposed delay-tolerant forwarding scheme, where solid lines, dash-dotted lines, dotted lines, and dash lines represent the packet transmission, original air route, changed air route, and communication range, respectively. T_{move} is the traveling time when moving from one location to another location.

ferry rate of the source node and the ferry node carrying their own data packets and moving to ground destination for data packet delivery individually. In this paper, the data ferry rate represents the amount of data packets that can be moved and delivered to ground destination successfully by the ferry node in a given time period. The larger the data ferry rate is, the more data packets can be delivered by the ferry node within a time period. Here, the data ferry rate is calculated as $R^{\{i\}}_{ferry} = \frac{W^{\{i\}}_{data}}{T^{\{i\}}_{travel}}$, where $W^{\{i\}}_{data}$ and $T^{\{i\}}_{travel}$ is the total amount of data packets carried and delivered to ground destination and the total amount of traveling time required to move to ground destination for node n_i, respectively. If the data ferry rate of rerouting the ferry node is larger than the sum of the data ferry rate of individual delivery by the source node and the ferry node, the ground destination will instruct the ferry node to change its original air route and move to the source node to load data packets. Otherwise, the source node and the ferry node carry their own data packets and move to ground destination to deliver data packets individually. If multiple ferry nodes can provide a larger data ferry rate of rerouting than that of individual delivery, the ferry node that provides the highest data ferry rate will be chosen.

For example in Subfig. 3(b), suppose that two ferry nodes n_e and n_f are near the source node n_s, and each of them has 10 data packets to deliver. The ground destination n_G first calculates the data ferry rate of rerouting the ferry node n_e and n_f, which is $R^{e}_{ferry*} = \frac{W^{\{s\}}_{data}+W^{\{e\}}_{data}}{T^{\{e\}}_{travel*}} = \frac{10+10}{2+5} = 2.86$ pkt/sec, and $R^{f}_{ferry*} = \frac{W^{\{s\}}_{data}+W^{\{f\}}_{data}}{T^{\{f\}}_{travel*}} = \frac{10+10}{11+7} = 1.18$ pkt/sec, respectively. And then, the ground destination n_G calculates the sum of the data ferry rate of individual delivery for the source node n_s and the ferry node n_e, which is $R^{\{s,e\}}_{ferry} = \frac{W^{\{s\}}_{data}+W^{\{e\}}_{data}}{T^{\{s\}}_{travel}} = \frac{10+10}{8} = 2.5$ pkt/sec, and for the source node n_s and the ferry node n_f, which is $R^{\{s,f\}}_{ferry} = \frac{W^{\{s\}}_{data}+W^{\{f\}}_{data}}{T^{\{f\}}_{travel}} = \frac{10+10}{8} = 2.5$ pkt/sec, respectively. Here, the larger traveling time is considered in the calculation of the sum of the data ferry rate of individual delivery. For example, in 8 seconds, the source node n_s and the ferry node n_e can move to ground

Notations:
- S_{ferry}, C_{ferry}: The set of current ferry nodes and the ferry node candidate.
- CR_i, *RTS*, *CTS*: Defined before.

Event-Driven Delay-Tolerant Forwarding Scheme:

◇ When a source node n_s has data packets to deliver, however, end-to-end routing path doesn't exist:

```
if a ferry node n_f is within the CR_s of n_s
    n_s and n_f exchange RTS and CTS packets;
    n_s sends data packets to n_f;
else
    for n_i ∈ S_ferry
        Calculate R^i_ferry* of rerouting ferry node n_i;
        Calculate the sum of R^s_ferry and R^i_ferry for individual delivery;
        if R^i_ferry* > (R^s_ferry + R^i_ferry) and R^i_ferry* > R^max_ferry
            R^max_ferry = R^i_ferry*; C_ferry = n_i;
    if C_ferry ≠ ∅
        Instruct ferry node C_ferry to load source node n_s's data packets;
    else
        Source node n_s carries and delivers data packets to ground destination;
```

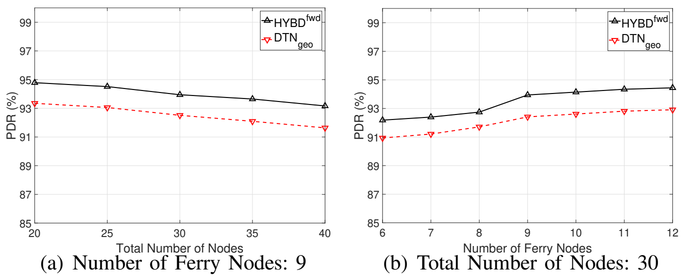
Fig. 4. The pseudocode of the proposed delay-tolerant forwarding scheme.

destination and deliver all their data packets. Since the data ferry rate of rerouting the ferry node n_e is larger than that of rerouting the ferry node n_f, thus, the ground destination n_G will instruct the ferry node n_e to change its original air route and move to the source node n_s to load data packets, and then deliver data packets to ground destination n_G. And the ferry node n_f will follow its original air route and move to ground destination n_G as planned. In the worst scenario, if there is no ferry node that can provide a higher data ferry rate of rerouting than that of individual delivery, the source node n_s will carry its data packets and physically move to ground destination n_G for data packet delivery. The major operations of the proposed delay-tolerant forwarding is shown in Fig. 4.

IV. Performance Evaluation

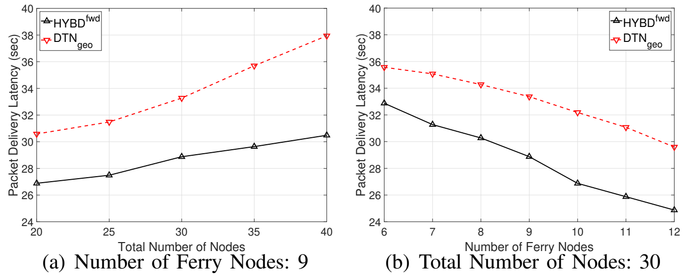
A. Simulation Testbed

We conduct extensive simulations using OMNeT++ [8] for performance evaluation and analysis. 20-40 nodes are uniformly distributed in a 1500×1500 (m^2) square network area, and a single ground destination is placed in the middle of bottom side of network area, where nodes will hover above the targeted area, and then collect and deliver data packets (i.e., high resolution overview images) to ground destination through the proposed $HYBD^{fwd}$ approach. Each node makes use of high-throughput radio technology with the limited



(a) Number of Ferry Nodes: 9 (b) Total Number of Nodes: 30

Fig. 5. The performance of packet delivery ratio (PDR) against the total number of nodes and the number of ferry nodes.



(a) Number of Ferry Nodes: 9 (b) Total Number of Nodes: 30

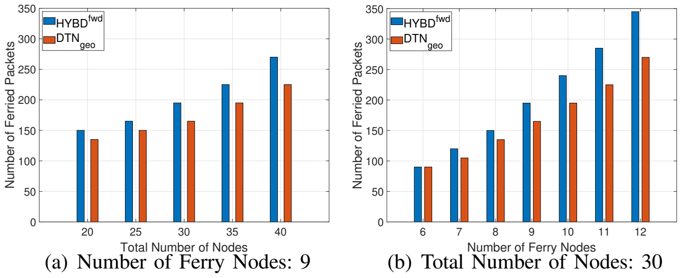
Fig. 6. The performance of packet delivery latency against the total number of nodes and the number of ferry nodes.

communication range (200 (m)) and low-throughput channel with large communication range (1000 (m)) for data traffic and control traffic, respectively. The throughput between two nodes is modeled as a function of distance derived from measurements according to Eq. 6. In addition, each node generates total 15 data packets to be addressed to ground destination. The channel error rate is randomly changed from 0 to 5%. The path-planned mobility model [30] is deployed in the network, where 6-12 ferry nodes are randomly selected and travel with a constant speed of 20 meter/sec toward a waypoint by following a pre-planned path without taking any random direction. In order to let all nodes in the network be able to physically move to ground destination to deliver data packets, the total simulation time is set to 100 seconds. Each simulation scenario is run 10 times with different randomly generated seeds to obtain steady state performance metrics. In this paper, we measure the performance in terms of packet delivery ratio, packet delivery latency, and number of ferried data packets by changing key simulation parameters including the total number of nodes and the number of ferry nodes. For performance comparison, we compare the $HYBD^{fwd}$ with a motion-driven packet forwarding algorithm DTN_{geo} in [9].

B. Simulation Results and Analysis

Fig. 5 shows the packet delivery ratio (PDR) against the total number of nodes and the number of ferry nodes in the network. As shown in Subfig. 5(a), the PDRs of both $HYBD^{fwd}$ and DTN_{geo} slightly decrease as the total number of nodes increases in the network. Since more nodes exist in the network, more data packets are generated to be delivered to ground destination. Thus, more data packets could be forwarded to ground destination though end-to-end routing path. However, a few amount of data packets could get lost during the transmission due to bad channel quality or link error, which results in a decreasing PDR. In addition, a higher PDR is observed by $HYBD^{fwd}$ compared to DTN_{geo}. This is because DTN_{geo} does not consider the link expiration time of end-to-end routing path, some data packets actually cannot be delivered to ground destination through pre-selected shortest path due to broken links. Finally, a less number of data packets are received by ground destination. However, $HYBD^{fwd}$ selects the end-to-end routing path in terms of link expiration time and link throughput. When a certain link is to be broken along the forwarding path, the source node will stop sending data packet, and try to find another end-to-end routing path or apply delay-tolerant forwarding scheme. In Subfig. 5(b), as the number of ferry nodes increases, the PDRs of both schemes slightly increase. Since more ferry nodes are available in the network, more data packets could be ferried and delivered to ground destination through ferry nodes instead of end-to-end routing paths, which results in an increasing PDRs. In particular, a larger number of data packets is delivered by $HYBD^{fwd}$ compared to DTN_{geo}. This is because the ferry nodes in $HYBD^{fwd}$ can actively move to other nodes and load data packets, more data packets can be physically moved to ground destination for delivery. On the other side, as less amount of data packets are forwarded through end-to-end routing paths, less data packets could get lost during the transmission due to bad channel quality or link error, resulting in a higher PDR.

In Fig. 6, the packet delivery latency is observed by varying the total number of nodes and the number of ferry nodes in the network. In Subfig. 6(a), as the total number of nodes in the network increases, the packet delivery latency of both $HYBD^{fwd}$ and DTN_{geo} increase significantly. As the total number of nodes increases, more nodes exist in the network, and more data packets are generated and required to be delivered to ground destination. Since the end-to-end routing path does not exist between every node and ground destination, more data packets have to be carried and delivered to ground destination through ferry nodes, or physically moved to ground destination by nodes themselves, which results in a longer traveling time for data packets. Finally, the overall packet delivery latency increases. $HYBD^{fwd}$ shows a lower packet delivery latency than that of DTN_{geo}, this is because the ferry nodes in $HYBD^{fwd}$ can reroute their air flights to other nodes to load data packets and deliver to ground destination, which results in a lower packet delivery latency. In Subfig. 6(b), the packet delivery latency significantly decreases as the number of ferry nodes increases in the network. Except for delivering data packets through end-to-end routing path, more data packets can be delivered to ground destination through ferry nodes, thus, the total amount of time for carrying and moving data packets to ground destination is significantly decreased. As a result, a lower packet delivery latency is observed. $HYBD^{fwd}$ shows a lower packet delivery latency compared to DTN_{geo}, this is because more ferry nodes can reroute to other nodes to load data packets, they do not need to physically move to ground destination to deliver data packets, and a large amount of traveling time can be reduced.



(a) Number of Ferry Nodes: 9 (b) Total Number of Nodes: 30

Fig. 7. The performance of number of ferried packets against the total number of nodes and the number of ferry nodes.

The number of ferried data packets are observed against the total number of nodes and the number of ferry nodes in the network in Fig. 7. As shown in Subfig. 7(a), for both $HYBD^{fwd}$ and DTN_{geo}, the number of ferried data packets slightly increases as the total number of nodes changes from 20 to 40. This is because more nodes exist in the network, however, not all nodes can find the end-to-end routing path to deliver data packets. Thus, more data packets have to be carried and delivered to ground destination by ferry nodes, resulting in an increment in the total number of ferried data packets. In particular, a larger number of ferried data packets is achieved by $HYBD^{fwd}$ compared to that of DTN_{geo}. This is because the ferry nodes in $HYBD^{fwd}$ can change its original air flight to other nodes to load data packets, more data packets will be ferried and delivered to ground destination. In Subfig. 7(b), $HYBD^{fwd}$ still outperforms DTN_{geo} in terms of the number of ferried data packets. In addition, as the number of ferry nodes increases in the network, the difference of the number of ferried data packets between $HYBD^{fwd}$ and DTN_{geo} increases. This is because more ferry nodes in $HYBD^{fwd}$ can actively move to other nodes and load more data packets. However, the ferry nodes in DTN_{geo} only accept data packets from one-hop neighbor nodes.

V. Conclusion and Future Work

In this paper, we propose a hybrid packet forwarding algorithm, which consists of end-to-end routing and delay-tolerant forwarding, to efficiently and reliably deliver data packets to ground destination in FANETs. We develop a customized discrete event driven simulation framework by using OMNeT++ and evaluate its performance through extensive simulation experiments in terms of packet delivery ratio, packet delivery latency, and number of ferried data packets. The simulation results indicate that the proposed hybrid packet forwarding algorithm is a viable approach for data transmission and communication in FANETs. Due to radio propagation and its channel dynamics cannot easily be captured by simulation model and framework, as a future work, we plan to develop a small-scale testbed with small quad-copters, e.g, Crazyflie 2.0, and deploy a real outdoor environment to see the full potential of the proposed approach.

References

[1] T. Long, M. Ozger, O. Cetinkaya, and O. Akan, "Energy Neutral Internet of Drones," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 22–28, 2018.

[2] T. Andre *et al.*, "Application-Driven Design of Aerial Communication Networks," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 129–137, 2014.

[3] C. Cheng, P. Hsiao, H. Kung, and D. Vlah, "Maximizing Throughput of UAV-Relaying Networks with the Load-Carry-and-Deliver Paradigm," in *Proc. IEEE WCNC*, 2007, pp. 4417–4424.

[4] M. Asadpour *et al.*, "Routing in a Fleet of Micro Aerial Vehicles: First Experimental Insights," in *Proc. ACM Airborne*, 2014, pp. 9–10.

[5] C. Pu, "Jamming-Resilient Multipath Routing Protocol for Flying Ad Hoc Networks," *IEEE Access*, vol. 6, pp. 68 472–68 486, 2018.

[6] M. Kang *et al.*, "Mobility Aware Hybrid Routing Protocol for Mobile Ad hoc Network," in *Proc. IEEE ICOIN*, 2012, pp. 410–414.

[7] M. Asadpour *et al.*, "Micro Aerial Vehicle Networks: An Experimental Analysis of Challenges and Opportunities," *IEEE Commun. Mag.*, vol. 7, no. 52, pp. 141–149, 2014.

[8] A. Varga, *OMNeT++*, 2014, http://www.omnetpp.org/.

[9] M. Asadpour *et al.*, "Route or Carry: Motion-Driven Packet Forwarding in Micro Aerial Vehicle Networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 3, pp. 843–856, 2017.

[10] M. Le, J. Park, and M. Gerla, "UAV Assisted Disruption Tolerant Routing," in *Proc. IEEE MILCOM*, 2006, pp. 1–5.

[11] S. Rosati, K. Krużelecki, G. Heitz, D. Floreano, and B. Rimoldi, "Dynamic Routing for Flying Ad Hoc Networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 3, pp. 1690–1700, 2016.

[12] D. Johnson and D. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Springer Mobile Computing*, 1996, pp. 153–181.

[13] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," in *Proc. WMCSA*, 1999, pp. 90–100.

[14] C. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *ACM Comp. Commun. Rev.*, vol. 24, no. 4, pp. 234–244, 1994.

[15] Z. Haas *et al.*, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," *Internet Draft, draft-ietfmanet-zone-zrp-04.txt*, July 2002.

[16] Z. Zhai, J. Du, and Y. Ren, "The Application and Improvement of Temporally Ordered Routing Algorithm in Swarm Network with Unmanned Aerial Vehicle Nodes," in *Proc. IEEE ICWMC*, 2013, pp. 7–12.

[17] L. Lin, Q. Sun, J. Li, and F. Yang, "A Novel Geographic Position Mobility Oriented Routing Strategy for UAVs," *Journal of Computational Information Systems*, vol. 8, no. 2, pp. 709–716, 2012.

[18] W. Zafar and B. Khan, "A reliable, delay bounded and less complex communication protocol for multicluster FANETs," *Digital Communications and Networks*, vol. 3, no. 1, pp. 30–38, 2017.

[19] Z. Zheng, A. Sangaiah, and T. Wang, "Adaptive Communication Protocols in Flying Ad Hoc Network," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 136–142, 2018.

[20] G. Gankhuyag *et al.*, "Robust and Reliable Predictive Routing Strategy for Flying Ad-hoc Networks," *IEEE Access*, vol. 5, pp. 643–654, 2017.

[21] G. Secinti, P. Darian, B. Canberk, and K. Chowdhury, "SDNs in the Sky: Robust End-to-End Connectivity for Aerial Vehicular Networks," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 16–21, 2018.

[22] N. Zhao, W. Lu, M. Sheng, Y. Chen, J. Tang, F. Yu, and K. Wong, "UAV-Assisted Emergency Networks in Disasters," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 45–51, 2019.

[23] C. Pu, "Link-Quality and Traffic-Load Aware Routing for UAV Ad Hoc Networks," in *Proc. IEEE CIC*, 2018, pp. 71–79.

[24] Y. Zhou, N. Cheng, N. Lu, and X. Shen, "Multi-UAV-Aided Networks: Aerial-Ground Cooperative Vehicular Networking Architecture," *IEEE Veh. Technol. Mag.*, vol. 10, no. 4, pp. 36–44, 2015.

[25] C. Pu, T. Gade, S. Lim, M. Min, and W. Wang, "Lightweight Forwarding Protocols in Energy Harvesting Wireless Sensor Networks," in *Proc. IEEE MILCOM*, 2014, pp. 1053–1059.

[26] W. Su *et al.*, "Mobility prediction and routing in ad hoc wireless networks," *Int. J. Network Mgmt*, vol. 11, no. 1, pp. 3–30, 2001.

[27] C. Pu, S. Lim, B. Jung, and J. Chae, "EYES: Mitigating Forwarding Misbehavior in Energy Harvesting Motivated Networks," *Elsevier Computer Communications*, vol. 124, pp. 17–30, 2018.

[28] C. Pu, S. Lim, C. Jinseok, and J. Byungkwan, "Active Detection in Mitigating Routing Misbehavior for MANETs," *Springer Wireless Network*, vol. 25, no. 4, pp. 1669–1683, 2017.

[29] M. M. T. Bin, M. Ammar, and E. Zegura, "Message Ferry Route Design for Sparse Ad Hoc Networks with Mobile Nodes," in *Proc. ACM MobiHoc*, 2006, pp. 37–48.

[30] A. Bujari, C. Calafate, J. Cano, P. Manzoni, C. Palazzi, and D. Ronzani, "Flying ad-hoc network application scenarios and mobility models," *Int. J. Distrib. Sensors Netw.*, vol. 13, no. 10, pp. 1–17, 2017.