Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

A Stochastic Packet Forwarding Algorithm in Flying Ad Hoc Networks: Design, Analysis, and Evaluation

CONG PU[1], (Member, IEEE), IMTIAZ AHMED[2], (Member, IEEE), EVAN ALLEN[1], AND KIM-KWANG RAYMOND CHOO[3], (Senior Member, IEEE)

[1] Department of Computer Sciences and Electrical Engineering, Marshall University, Huntington, WV 25755, USA (e-mail: cong.pu@outlook.com, allen403@marshall.edu)

[2] Department of Electrical Engineering and Computer Science, Howard University, Washington, DC 20059, USA (e-mail: iahmed123b@ieee.org)

[3] Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249, USA (e-mail: raymond.choo@fulbrightmail.org)

Corresponding author: Cong Pu (e-mail: cong.pu@outlook.com).

ABSTRACT Over the past decade, unmanned airborne vehicles (UAVs; widely known as drones) are quickly being deployed in various civilian as well as military applications. Drones can self-organize into a connected swarm (Flying Ad Hoc Networks – FANETs) to complete various challenging missions. As the primary building block of Internet-of-Drones (IoD), FANETs have an important role to play in governing the autonomous movement of drones and supporting drone-to-everything (D2X) communications. However, factors such as flying characteristics and the highly dynamic topology of FANETs compound the challenges of packet forwarding; thus, the focus of this article. Specifically, a stochastic packet forwarding algorithm (hereafter referred to as *SPA*) is proposed for FANETs, where the data packets can be efficiently transferred to the destination. In *SPA*, the packet sender evaluates each next hop forwarding candidate drone based on different network metrics, prior to selecting the next hop forwarding drone according to the forwarding probability. In addition, an analytical model is developed to evaluate the performance of *SPA*. Then, the link expiration time and link throughput are chosen as network metrics for network simulation experiments, where *SPA* is evaluated and compared with prior schemes (e.g, DTN_{geo} and *GeoUAVs*). The performance of *SPA* is evaluated in a real-world testbed to complement the network simulation experiments. Findings from the evaluations demonstrate that *SPA* obtains better performance in terms of packet delivery ratio, packet delivery latency, the number of delivered data packets, average link lifetime, and bit error rate.

INDEX TERMS Drones, Flying Ad Hoc Networks, Entropy, Stochastic Packet Forwarding, Internet-of-Drones

I. INTRODUCTION

With their flexibility and adaptability, unmanned airborne vehicles (UAVs), also publicly known as drones, have been extensively deployed in a broad variety of applications over the past years [1]. Historically, drones were used primarily for military intelligence, surveillance and reconnaissance missions because they would not put a pilot's life at risk in combat zones. Due to relatively easy installation, small operating expenses and device miniaturization, drones are becoming more popular for their possible use in different civilian scenarios, such as aerial surveying and mapping, aerial delivery, infrastructure inspection and monitoring, and crowd management. According to ''The Drone Market Report 2021'' [2], the global commercial drone market is estimated to reach the size of approximate USD 502 billion by 2028. In the aerospace and defense industry market, the United States will continue to be ahead of the pack, with Research & Development (R&D) expenses reportedly increasing to about $3.0 billion in 2029, and procurement growing from $2.5 billion in 2020 to approximate $3.3 billion by 2030 [3].

As the proliferation of drone-based systems, Internet-of-Drones (IoD) as well as its derived applications shift into the fast lane. In IoD, a significant number of drones smoothly interact with each other to achieve the objective of managing the controlled airspace, coordinating drones, and providing

application-specific services [4]. Even though the ''Jetson lifestyle'' is not upon us yet, the age of Artificial Intelligence (AI) and Internet of Everything (IoE) has arrived. As a consequence of advanced technologies (e.g., AI, IoE, and 5G), it is envisioned that drones will totally change our point of view and perspective on the world.

Single drone has been used as wireless relay [5], flying sensor [6], or even aerial base station [7]. To complete challenging tasks efficiently and effectively, a group of drones can form a Flying Ad Hoc Network (FANET), which is dispatched to the area of interest (e.g., disaster area). Nevertheless, the special features of FANETs, for example the intermittent connectivity and uncertain wireless channels caused by drones' high mobility, unpredictable movements, and nonuniform distribution, make the packet forwarding a very challenging task. Thus, the unique characteristics of FANETs require a rethinking of packet forwarding.

1) First, since each drone is furnished with Inertial Navigation System (INS), Global Positioning System (GPS), and Inertial Measurement Units (IMUs), the location and mobility information of drones are available to make forwarding decision, e.g., the packets can be forwarded to a drone that is physically closer to the destination. However, pure geographic packet forwarding is not sufficiently competent for aerial networks that face intermittent connectivity and unpredictable topological changes. A known approach to intermittent connectivity is the estimation of link expiration time, which computes the time when the two communicating drones move out of the radio transmission range. Hence, the reliability of packet forwarding can be enhanced, since the packets can be sent before the expected link expiration time [8].
2) Second, the link throughput is another factor that needs to be considered. With the same link connection time, more data could be transmitted if the link with a higher data rate is chosen.
3) Third, the packet forwarding model should be extensible and flexible, which means that new network metrics (e.g., traffic load and battery energy) can be included to evaluate the next hop forwarding drone.
4) Last but not least, many communication algorithms and packet forwarding techniques have been proposed in FANETs. However, the difficulties to build a real-world experimental testbed have confined the evaluation of schemes to the network simulation.

This article focuses on the design, analysis, and evaluation of a packet forwarding algorithm in FANETs. Our major contributions are summarized in the following:

- A stochastic packet forwarding algorithm, hereafter referred to as *SPA*, is designed to deliver data packets efficiently and reliably in FANETs. In *SPA*, the packet sender evaluates each next hop forwarding candidate drone in terms of different network metrics, and then selects the next hop forwarding drone according to the calculated forwarding probability.
- Several network metrics are used by *SPA* to inform packet forwarding decision-making. Thus, *SPA* can nimbly and accurately detect the change of link quality and network condition. In addition, *SPA* is designed to facilitate extensibility and flexibility so that new network metrics can be conveniently included in *SPA*.

Compared to the preliminary version of the paper published in 2019 IEEE Military Communications Conference (MILCOM) [1], the following significant extensions are made:

- Prior communication schemes in FANETs and similar environments are investigated and classified in seven categories. In addition, *SPA* is compared with existing schemes and their forwarding strategies are analyzed in terms of seven criteria.
- A mathematical model is developed to analyze *SPA* in terms of four performance metrics. Link expiration time and link throughput are chosen, and extensive network simulation experiments are conducted on *SPA* using OMNeT++ [9]. For performance comparison, both motion-driven packet forwarding algorithm (DTN_{geo}) [10] and geocast routing protocol (*GeoUAVs*) [11] are implemented in the framework.
- A real-world testbed consisting of two DJI Mavic 2 Pro drones, one laptop, two ADALM Pluto Software Defined Radio (SDR) modules, two Latte Panda development boards, and two micro GPS circuits is set up. Two drones have autonomous flight and hovering capabilities and serve as potential packet forwarding candidates, while the laptop is considered as a packet sender ground station. Leveraging the real-world testbed, the performance of *SPA* is evaluated to complement the network simulation experiments.
- The design features and potential constraints of *SPA*, as well as possible extensions for future research are discussed. The potential security issues are examined and the corresponding countermeasures to secure *SPA* are proposed.

The rest of the article is organized as follows. Literature review is provided in Section II. Section III first provides the system model, and then introduces the proposed packet forwarding algorithm. In Section IV, the mathematical model and its numerical results are provided. In Sections V and VI, network simulation and real-world testbed experiments are provided and analyzed, respectively. Extensive experimental results indicate that the proposed scheme can provide better performance compared to benchmark schemes. A discussion of the proposed scheme is provided in Section VII. Finally, the conclusion is made in Section VIII.

II. RELATED WORK

Many routing protocols and communication algorithms have been developed for MANETs and VANETs during the last two decades. Due to drones' high mobility, intermittently

connected communications, and unstable network topology, however, existing approaches specifically developed for both VANETs amd MANETs might not be applicable for FANETs. In this section, the prior communication schemes in FANETs and similar systems are surveyed, and then grouped in the following seven categories.

- Static routing: Static routing tables are computed and loaded when the task starts, and these tables cannot be updated during the task operation.
- Proactive routing: Routing tables are updated and shared periodically among the drones, resulting in the availability of routing paths between every pair of drones in the network.
- Reactive routing: Reactive routing is also called on-demand routing, which can be used to find a routing path on demand when packets need to be sent.
- Hybrid routing: A combination of proactive and reactive routing to overcome the weaknesses of proactive routing and reactive routing.
- Position-based routing: The routing decision at each node is based on the destination's position contained in the packet and the position of the forwarding node's neighbors.
- Cluster routing: Cluster protocols place nodes into groups, called clusters, and perform hierarchical routing between these clusters.
- Other approaches: All others except above-mentioned techniques.

A. STATIC ROUTING PROTOCOLS

In static routing protocols, drones physically carry and move the data from source to destination along a pre-defined flying path. Thus, before the mission begins, the flying path has to be calculated and embedded into drones. The static routing protocols have the lightweight advantage. However, the disadvantages are that the flying path cannot be updated during the mission operation phase and dynamically changing environment has a huge impact on the mission completion. The authors in [12] propose a single-hop routing protocol, called load-carry-and-deliver, for the information exchange between two ground stations. In the load-carry-and-deliver, a drone collects the data from source ground station and flies to destination ground station for data delivery. In [13], the authors consider highly partitioned ad hoc networks, and assume that intermediate drones do not exist for the transmission of messages. As a result, drones are used as information carriers that can store and deliver the information to the destination. In summary, the static routing protocols are lack of flexibility and the systems adopting these protocols hardly perform well in mobile networks.

B. PROACTIVE ROUTING PROTOCOLS

The proactive routing protocols update and broadcast route information regularly within the network so that drones can have up-to-date route information. When the source plans to send the data packets, it can immediately select an appropriate routing path over which the data packets can be forwarded to the destination. However, the disadvantages are also indisputable in the proactive routing protocols, where a ton of network control packets are frequently exchanged among drones to keep timely route information. In [15], the authors first propose to estimate the link qualities to neighbor nodes and the traffic load of neighbor nodes, respectively. And then, the link quality and traffic load schemes are integrated with OLSR [14] to deliver data packets efficiently in UAV Ad Hoc Networks (UANETs). In [16], the authors propose a mobility-aware prediction-based OLSR routing protocol (P-OLSR) for FANETs. In the P-OLSR, the velocity of drones is considered to calculate the expected transmission count metrics to make routing decision. The authors in [17] propose a mobility and traffic load based routing protocol, also called ML-OLSR, for FANETs. The packet sender evaluates the speed, position, and traffic load of adjacent drones and selects the packet receiver with lower speed and light traffic load.

C. REACTIVE ROUTING PROTOCOLS

In reactive routing protocols, the routing path is established whenever required. The reactive routing protocols have a lower communication overhead (or less number of network control packets) compared to proactive routing protocols. However, there is a higher communication latency because the route from the source to the destination has to be discovered during the route discovery phase. Dynamic source routing (DSR) [18] 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. In [21], a secure AODV-based routing protocol (SUAP) is designed based on model driven development approach. The SUAP is an extension of AODV by including cryptographic scheme, secure hash function, and geographical leash. The authors in [22] propose a multipath routing protocol (JarmRout) to defend against jamming attack in FANETs. In the JarmRout, the packet source first performs the route discovery by broadcasting a route request packet. When the destination node receives more than two route request packets, it selects two node-disjoint paths with maximum distance and replies route reply packets back to source node.

D. HYBRID ROUTING PROTOCOLS

In hybrid routing protocols, reactive and proactive approaches are integrated together to increase scalability by forming certain backbones so that the route discovery overhead can be reduced. In [23], 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

TABLE 1: The comparison of prior approaches.

Approach	Type	Periodic Broadcast	Multipath	Load Balance	Route Update	Adaptability	Network Metric
LCAD [12]	Static	No	No	No	No	No	Throughput
DTN^{UAV} [13]	Static	No	No	No	No	No	Waypoints
OLSR [14]	Proactive	Yes	No	No	Periodically	No	Optimized link
LTA-OLSR [15]	Proactive	Yes	Yes	Yes	Periodically	No	Link quality & Traffic load
P-OLSR [16]	Proactive	Yes	No	No	Periodically	No	Weighted ETX$^{\divideontimes}$
ML-OLSR [17]	Proactive	Yes	Yes	Yes	Periodically	No	Mobility & Traffic load
DSR [18]	Reactive	No	Yes	No	If needed	Yes	Hop count
AODV [19]	Reactive	No	Yes	No	If needed	Yes	Hop count
TS-AODV [20]	Reactive	No	Yes	No	If needed	Yes	Time slot
SUAP [21]	Reactive	No	Yes	No	If needed	Yes	Hop count
JarmRout [22]	Reactive	No	Yes	Yes	If needed	Yes	Multiple$^{\boxtimes}$
ZRP [23]	Hybrid	Yes	Yes	No	Hybrid	Yes	Hop count
RTORA [24]	Hybrid	Yes	Yes	No	Hybrid	Yes	DAG Height
TORA [25]	Hybrid	Yes	Yes	No	Hybrid	Yes	DAG Height
LEPR [26]	Hybrid	No	Yes	No	If needed	Yes	Multiple$^{\boxtimes}$
MPCA [27]	Hybrid	Yes	No	No	Hybrid	Yes	Mobility
GPSR [28]	Greedy	No	No	No	If needed	Yes	Geometric distance
DTN^{geo} [10]	Greedy	Yes	No	No	Periodically	Yes	Throughput
GEO [29]	Greedy	Yes	No	No	If needed	Yes	Geolocation
$GeoUAVs$ [11]	Greedy	Yes	No	No	No	Yes	Geolocation
PPMAC+RLSRP [30]	Cross-layer	Yes	No	No	Periodically	Yes	Position
RARP [31]	Prediction	Yes	Yes	No	If needed	Yes	Location&Trajectory
SDN^{e2e} [32]	SDN$^{\odot}$	Yes	Yes	Yes	Periodically	No	Transmission time
SD-UAVNet [33]	SDN$^{\odot}$	Yes	No	No	Periodically	Yes	Multiple$^{\boxtimes}$
SPA	Stochastic	Yes	Yes	Yes	No	Yes	Throughput & LET$^{\circledast}$

$^{\odot}$: SDN: Software defined networking.
$^{\divideontimes}$: ETX: Expected transmission count.
$^{\circledast}$: LET: Link expiration time.
$^{\boxtimes}$: More than two network metrics are considered.

framework to improve the efficiency of a globally reactive route query/reply mechanism. The authors in [24] propose a rapid-reestablish temporally ordered routing algorithm, also called RTORA, in FANETs. The goal of RTORA is to prevent useless control packets from broadcasting and relieve negative effects caused by link failure. A link stability based preemptive routing protocol (LEPR) is proposed based on AODV for FANET in [26]. The LEPR uses the past, current, and future information of link stability to calculate multiple link-disjoint paths for data packet transmission. Here, the link stability network metrics is modeled as a function of the quality of link as well as the mobility.

E. GREEDY ROUTING PROTOCOLS

The greedy routing protocol is used to forward the packets to the drones that can minimize the number of hops or physical distance to the target destination in order to achieve the goal of reducing the packet delivery latency. To put it simply, the packet sender always sends the packets to the neighbor node that is geographically located closer to the intended destination. If such neighbor node does not exist, the perimeter forwarding should be applied so that the packets are forwarded around the perimeter of the gap region to the counterclockwise neighbor node of the packet sender. In [10], a motion-driven packet forwarding algorithm is proposed for UAV networks, where the traditional end-to-end routing and delay-tolerant forwarding are integrated together. The proposed motion-driven packet forwarding algorithm first implements Dijkstra's algorithm to find the end-to-end shortest path to the destination. If the shortest path is not available, the proposed algorithm then applies the delay-tolerant forwarding. To be specific, the sender can choose to either send the packets to a neighbor node or store the packets by itself according the virtual link weight. The authors in [29] propose a geolocation-based routing protocol in FANETs, where the routing path is established based on the geolocation information of adjacent nodes. In [11], a routing protocol named $GeoUAVs$ takes into consideration the mobility of UAVs and dynamic changing topology to deliver the data to a specific group of UAVs.

F. POSITION-BASED ROUTING PROTOCOLS

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 at each node. In [34], 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. The [35] proposes a routing protocol based on geographic positions for inter-drone communications in FANETs, where a mobility prediction method based on the Gaussian distribution function is utilized to reduce the impact of the high mobility of drones with an acceptable communication overhead. The [36] proposes a geographic load sharing strategy to fully exploit the total air-to-ground available capacity at any given time. When forwarding packets to a destination, a flight considers a set

of next-hop candidates and spreads traffic among them based on queue dynamics. In addition, load balancing is performed among Internet Gateways by using a congestion-aware handover strategy.

G. CLUSTER ROUTING PROTOCOLS

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 [37], a clustering algorithm is proposed for drone networking in near-space. First, the ground stations construct the initial clusters according to 3D coordinate information of drones. Then, the drones with higher residual energy, longer connection endurance time with neighbors, and moderate numbers of neighbor nodes are selected as cluster head. The [27] develops a cluster formation algorithm for UANETs to solve the problem of frequent cluster updates due to high-speed drones with the prediction of network topology updates. It predicts the mobility structures of drones with the help of the dictionary trie structure prediction algorithm, link expiration time, and mobility mode. In [38], 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.

H. OTHER APPROACHES

In [39], a software-defined networking based topology management (STFANET) is proposed for FANETs. STFANET is a coordination protocol that englobes both an efficient SDN-based UAV communication and a set of topology management algorithms. In [40], a routing protocol using modified AntHocNet is proposed for FANETs, where ant colony optimization technique has been adopted to achieve better dependability and performance. In [41], the authors propose another routing technique, which is the extension of AntHocNet due to mobile characteristics. In [42], the authors focus on the energy conservation problem in flying-IoT, where a routing approach for the internet of flying vehicles using DSDV is proposed. The authors in [43] describe the use of each UAV's residual energy level to ensure a high level of safety using an antbased routing technique called AntHocNet. A hybrid protocol integrated with a prediction based medium access control scheme and a reinforcement learning based routing protocol is designed for FANETs in [30] With the directional antenna, the proposed protocol can resolve the issue of directional deafness in the MAC layer with the assistance of position prediction technique. In addition, the proposed protocol can update the local routing table with the information of drones' position through a reward function. The authors in [31] propose a predictive routing protocol in FANETs. The goal of the prediction mechanism is to improve the efficiency of routing protocol by estimating the location and trajectory of drones. In [32], a software defined networking (SDN) based network management protocol is designed to achieve the end-to-end data exchange in FANETs. In the network, drones act as SDN switches and operate with commands or instructions received from a centralized controller. In [33], a software-defined UAV communication framework, also called SD-UAVNet, is proposed to resolve the issues of UAV collisions in an open environment. The goals of the SD-UAVNet framework are to optimize the movements of UAVs, determine the deployment of relay UAVs, select appropriate routes, and finally avoid UAV collisions. A survey focusing on FANET-specific communication schemes is provided in [44].

After conducting a systematic literature review, it is found that each routing protocol is designed for specific situation and has its own merits and demerits. For example, some approaches are built with delay-tolerant forwarding and/or end-to-end routing techniques to achieve the desired data delivery ratio. However, they also experience a high communication latency. To properly support the emerging FANET applications, it is mandatory to consider the unique characteristics of FANETs in the design of communication schemes. However, few effort has been paid to a stochastic packet forwarding algorithm in FANETs. Finally, *SPA* is compared with prior schemes in terms of seven criteria in Table 1.

III. STOCHASTIC PACKET FORWARDING ALGORITHM

A. SYSTEM MODEL

In this article, it is assumed that a fleet of drones (later nodes) are deployed in an area of interest for a search and rescue mission (i.e., locate a missing person). Each node takes some (high-resolution) images and send them to a ground station for further processing and analysis. Each node is associated with a unique identification number. In addition, all nodes are armed with inertial measurement unit and navigation system, global positioning system, as well as digital map so that they can obtain their real-time geographical location and mobility information. It is also assumed that each node has the knowledge of the geographical location of ground station [10]. Thus, the packet sender or forwarder only forwards the data packets towards the direction of destination node. Since the data packets will not be forwarded to the opposite direction of destination node, temporary forwarding loop can be avoided, where the data packet is forwarded backwards and forwards between two nodes because one node is a good forwarding candidate to the other and vice versa. Each node can harvest energy from ambient environmental (i.e., wireless charging [45] or solar energy [46], [47]) and store harvested energy in the rechargeable battery for future use. Thus, energy consumption is not a concern of this article.

B. A BRIEF OVERVIEW

In *SPA*, when a node (i.e., source node or intermediate node) plans to send the data packets, it first evaluates all forwarding candidate nodes with the information of different network metrics according to the entropy weight theory. Then, it calculates each forwarding candidate node's forwarding probability and randomly selects one forwarding node to send data packets. Since different real-time network metrics

TABLE 2: Forwarding Candidate Table

Node ID	Link Throughput	Link Expiration Time	...	Network Metrics^M	Expiration Time Period
n_1	11	12	...	$1M$	t_{exp}^1
n_2	21	22	...	$2M$	t_{exp}^2
n_3	31	32	...	$3M$	t_{exp}^3
...	...	...	...	...	...
n_N	$N1$	$N2$	...	NM	t_{exp}^N

***This table serves as a template of forwarding candidate table which contains node ID, M number of network metrics, and entry expiration time period, respectively. However, this article only considers two real-time network metrics: link throughput and link expiration time.*

has distinct scale and unit, a weight is objectively assigned to each network metrics, which can significantly reduce the negative effect from various metrologies in evaluating forwarding candidate nodes. In addition, by stochastically selecting forwarding node, each forwarding candidate node will have a chance to be involved in the packet forwarding operation. The rationale behind this design is to achieve load balancing and finally realize the goal of extending network lifetime. More details about *SPA* are presented below. All notations used in this article are provided in Table 3.

C. STOCHASTIC PACKET FORWARDING ALGORITHM

First, each node periodically broadcasts *HELLO* messages piggybacked with its own node ID, position coordinates, moving speed, and moving direction to neighbor nodes. *HELLO* messages are transmitted in the broadcast manner so that they are received or overheard by all one-hop neighbor nodes that are located within the communication range, but not relayed or broadcasted further. Through periodically exchanged *HELLO* messages, each node is aware of its one-hop neighbor nodes. When the node has a data packet to send, its one-hop neighbor nodes who are moving towards to the destination node will be considered as forwarding candidate nodes. Since each node is aware of the geographical location of ground station and the *HELLO* messages also contain the moving direction of neighbor nodes, thus, the packet sender knows which of its neighbor nodes is moving closer to the ground station. In addition, these periodically exchanged *HELLO* messages permit each node to learn the network condition associated with each neighbor node, and then build its forwarding candidate table (*FT*). In the *FT*, each entry contains neighbor node ID (n_{id}), multiple real-time network metrics such as link throughput, link expiration time, etc. Each entry is also associated with an expiration time period (t_{exp}). Here, the expiration time period t_{exp} is a system parameter, indicating that the node will remove the entry of neighbor node from the *FT* if it does not receive the *HELLO* message from this neighbor node before the t_{exp} expires. When the node receives the *HELLO* message from a neighbor node that is already in the *FT* before the t_{exp} expires, it updates the information of real-time network metrics associated with this neighbor node, and resets the t_{exp}. If the node receives a *HELLO* message from a new neighbor node, a new entry is added into the *FT*. Here, the template of forwarding candidate table *FT* is shown in Table. 2, where *M* number of network metrics are considered for *N* number of forwarding candidate (or neighbor) nodes.

TABLE 3: Notations

Notation	Meaning
FT	Forwarding candidate table
N	The number of nodes
M	The number of network metrics
n_{id}	Node ID
t_{exp}	Expiration time period
$\mathbb{R}^{N\times M}$	Normalized matrix
$_j$	The j^{th} network metrics of all nodes
$_{ij}$	The value of the j^{th} network metrics of n_i
$^*_{ij}$	The normalized value of the j^{th} network metrics of n_i
ψ_j	The efficiency coefficient of the j^{th} network metrics
$\mathbb{E}_j$	The entropy of the j^{th} network metrics
$\mathbb{P}_{ij}$	The proportion of the j^{th} network metrics of n_i
Φ_j	The entropy weight of the j^{th} network metrics
$\mathbb{A}_i^{fwd}$	The forwarding availability of n_i
$\mathbb{P}_i^{fwd}$	The forwarding probability of n_i
R	The communication range of nodes
(x_i, y_i)	The two-dimensional position coordinates of n_i
v_i	The moving speed of n_i
$LK_{i,j}^{exp}$	The link expiration time between n_i and n_j
$LK_{i,j}^{tp}$	The link throughput between n_i and n_j
$dist(i,j)$	The spatial distance between n_i and n_j
$X \times Y$	The size of network area
ϑ	Packet loss ratio
P_{deliv}	Packet delivery ratio
$route^{relay}$	The average number of relays of a route
$dist^{avg}$	The average distance between source and destination
pro^{relay}	The average progress of each relay along the route
N_R^{avg}	The average number of nodes located within R
π	Pi

Second, when a packet sender has data packets to send, it retrieves the information of the forwarding candidate table *FT* and removes the column of node ID n_{id} and candidate expiration time period t_{exp}. Then, the packet sender normalizes the value of each network metrics in the *FT* according to Eq. 1,

$$^*_{ij} = \frac{max\{_j\} - _{ij}}{max\{_j\} - min\{_j\}} \times \psi_j + (1 - \psi_j), \quad i = 1, 2, \ldots, N, \quad j = 1, 2, \ldots, M, \tag{1}$$

and generates a normalized matrix of the value of network metrics $\mathbb{R}^{N\times M}$, as shown in Fig. 1. Here, ψ_j is the efficiency coefficient used to control the value range of the j^{th} network metrics, and $\sum_{j=1}^{M} \psi_j = 1.0$. The rationale behind the design

$$\mathbb{R}^{N\times M} = \begin{pmatrix} æ^*_{11} & æ^*_{12} & æ^*_{13} & \cdots & æ^*_{1M} \\ æ^*_{21} & æ^*_{22} & æ^*_{23} & \cdots & æ^*_{2M} \\ æ^*_{31} & æ^*_{32} & æ^*_{33} & \cdots & æ^*_{3M} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ æ^*_{N1} & æ^*_{N2} & æ^*_{N3} & \cdots & æ^*_{NM} \end{pmatrix}$$

FIGURE 1: The normalized matrix $\mathbb{R}^{N\times M}$.

of ψ_j is to adjust the effect of the j^{th} network metrics for subjective preference, since the entropy is a pure objective weighting allocation method. For example, if the j^{th} network metrics weights more than the k^{th} network metrics in the process of making a decision to select forwarding node, then $\psi_j > \psi_k$. After that, the packet sender calculates the entropy of the j^{th} network metrics according to Eq. 2,

$$\mathbb{E}_j = -\frac{1}{\ln N}\sum_{i=1}^{N}\Big(\mathbb{P}_{ij}\cdot \ln \mathbb{P}_{ij}\Big), \quad j = 1, 2, \ldots, M, \tag{2}$$

where $\mathbb{P}_{ij}$ is the proportion of the j^{th} network metrics associated with the forwarding candidate node n_i, and is represented as

$$\mathbb{P}_{ij} = \frac{\overset{*}{ij}}{\sum_{k=1}^{N}\overset{*}{kj}}, \quad j = 1, 2, \ldots, M. \tag{3}$$

Based on the concept of entropy [48], the entropy weight of the j^{th} network metrics Φ_j can be defined as

$$\Phi_j = \frac{1-\mathbb{E}_j}{\sum_{k=1}^{M}(1-\mathbb{E}_k)}, \quad j = 1, 2, \ldots, M. \tag{4}$$

Thus, the forwarding availability of the forwarding candidate node n_i, denoted by $\mathbb{A}_i^{fwd}$, can be calculated according to the normalized value of network metrics and the entropy weight of network metrics, which is represented as

$$\mathbb{A}_i^{fwd} = \sum_{j=1}^{M}(\Phi_j \cdot \overset{*}{ij}), \quad i = 1, 2, \ldots, N. \tag{5}$$

Here, the forwarding availability is the comprehensive assessment of forwarding candidate node to send a data packet. The forwarding availability reflects the real-time forwarding node state. The higher forwarding availability a forwarding candidate node achieves, the more likely it is chosen to forward a data packet. Finally, the forwarding probability of the forwarding candidate node n_i, denoted by $\mathbb{P}_i^{fwd}$, can be obtained from

$$\mathbb{P}_i^{fwd} = 1 - \frac{\mathbb{A}_i^{fwd}}{\sum_{k=1}^{N}\mathbb{A}_k^{fwd}}, \quad i = 1, 2, \ldots, N. \tag{6}$$

Third, based on the calculated forwarding probability of each forwarding candidate node, the packet sender stochastically chooses the forwarding node, and then sends the data packets. In detail, the packet sender generates a random number (e.g., rand[0,1]) and compares it with the forwarding probability of one forwarding candidate node. If the forwarding probability of this forwarding candidate node is larger than the randomly generated number, the packet sender selects it as the forwarding node, and then sends the data packets. Otherwise, the packet sender will generate a new random number and compare it with the forwarding probability of another forwarding candidate node, at which the same aforementioned operations will be applied to select the forwarding node. This selection process will continue until the packet sender successfully chooses the forwarding node whose forwarding probability is larger than the randomly generated number. The rationale behind the design of randomly selecting the forwarding node is to achieve conditional load balancing. Through comparing the random number with the forwarding probability of each forwarding candidate node, each forwarding candidate node has some chance to be selected as the forwarding node and involves in the forwarding operation. Here, the conditional load balancing means that the forwarding candidate node with a better link will have more chance to be selected as the forwarding node, while other forwarding candidate nodes also have chance to involve in the forwarding operation. Another way to achieve the load balancing is to develop a selection algorithm based on the forwarding probability of each forwarding candidate node and the total number of forwarding candidate nodes. However, the main focus of this article is the stochastic packet forwarding algorithm, thus, the design of selection algorithm will be left for future work. In order to improve fault tolerance and network resiliency, more than one forwarding node can be chosen to send data packets, but the number of forwarding node is limited to one in this article.

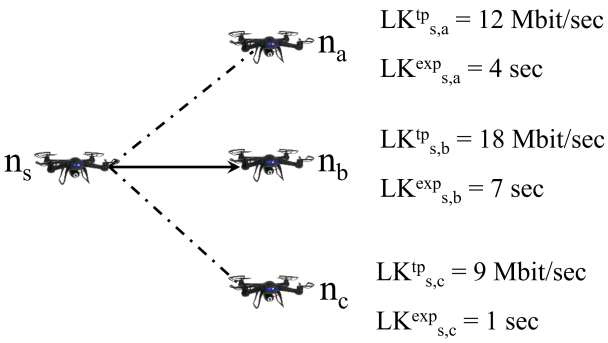


FIGURE 2: A snapshot of the network, where the packet sender n_s has three forwarding candidate nodes, n_a, n_b, and n_c, to select and send data packets. Here, $LK^{tp}_{s,i}$ and $LK^{exp}_{s,i}$ is the link throughput and link expiration time between node n_s and node n_i, and solid line represents the forwarding of data packets.

For example, as shown in Fig. 2, suppose that the packet sender n_s has three forwarding candidate nodes (or one-hop neighbor nodes), n_a, n_b, and n_c, to select and send data

packets. Link throughput and link expiration time are considered as real-time network metrics to calculate the forwarding probability of each forwarding candidate node. Here, the modeling of link throughput and link expiration time is introduced in Subsection III-D. Thus, n_s first generates the normalized matrix $\mathbb{R}^{3\times2}$ according to Eq. 1.

$$\mathbb{R}^{3\times2} = \begin{bmatrix} 0.83 & 0.75 \\ 0.5 & 0.5 \\ 1.0 & 1.0 \end{bmatrix}$$

After that, n_s calculates the entropy and the entropy weight of link throughput and link expiration time according to Eq. 2 and Eq. 4, respectively.

$$\mathbb{E}_{LK^{tp}} = 0.80 \quad \mathbb{E}_{LK^{exp}} = 0.77$$

$$\Phi_{LK^{tp}} = 0.47 \quad \Phi_{LK^{exp}} = 0.54$$

Then, n_s calculates the forwarding availability and forwarding probability of each forwarding candidate node based on Eq. 5 and Eq. 6, respectively.

$$\mathbb{A}_a^{fwd} = 0.79 \quad \mathbb{A}_b^{fwd} = 0.50 \quad \mathbb{A}_c^{fwd} = 1.0$$

$$\mathbb{P}_a^{fwd} = 0.66 \quad \mathbb{P}_b^{fwd} = 0.78 \quad \mathbb{P}_c^{fwd} = 0.56$$

Finally, n_s generates a random number rand[0,1], compares it with the forwarding probability of n_a, n_b, and n_c, and selects the forwarding node. Note that the forwarding candidate node n_b provides the largest throughput (18 Mbit/sec) and longest link expiration time (7 sec) as shown in Fig. 2. If n_b was selected as forwarding node, more data packets could be forwarded from n_s before the link between n_s and n_b is broken. Thus, n_b has the largest forwarding probability ($\mathbb{P}_b^{fwd} = 0.78$) to be selected as the forwarding node, compared to that of n_a and n_c. Major operations of *SPA* are summarized in Fig. 3.

D. LINK EXPIRATION TIME AND LINK THROUGHPUT MODELING

To accurately estimate the expiration time of the link between two adjacent nodes, the authors in [49] propose a link expiration time (LET) estimation scheme. Without loss of generality, two adjacent nodes which are denoted as n_i and n_j respectively are considered. In addition, the two-dimensional coordinates of n_i and n_j are represented as (x_i, y_i) and (x_j, y_j). The communication range of each node is R. It is also assumed that the moving speed of n_i and n_j are v_i and v_j, respectively. Their moving directions are denoted as θ_i and θ_j, where $0 \leq \theta_i, \theta_j < 2\pi$. According to [49], the LET between n_i and n_j, denoted as $LK_{i,j}^{exp}$, can be represented by

$$LK_{i,j}^{exp} = \frac{-(a\cdot b + c\cdot d) + \sqrt{(a^2+c^2)\cdot R^2 - (a\cdot d - b\cdot c)^2}}{a^2+c^2}, \tag{7}$$

where

$$a = v_i \cdot \cos\theta_i - v_j \cdot \cos\theta_j, \tag{8}$$

$$b = x_i - x_j, \tag{9}$$

Notations:

• *FT*, n_{id}, t_{exp}, $\mathbb{R}^{N\times M}$, $\mathbb{E}$, Φ, $\mathbb{A}^{fwd}$, and $\mathbb{P}^{fwd}$: Defined before.

• $FT_i[j].metrics$: The information of real-time network metrics associated with node n_j stored at node n_i.

• $pkt[n_{id}, type]$: A packet containing a node ID (n_{id}) and packet type (*type*). Here, *type* can be either *Data* or *HELLO*.

◇ When a node n_i receives a *HELLO* packet $pkt[n_j, HELLO]$ from node n_j:
if $n_j \in FT_i[n_{id}]$ /* n_j is the neighbor node of n_i */
Update $FT_i[j].metrics$;
Reset $FT_i[j].t_{exp}$;
else /* n_j is a new neighbor node of n_i */
Add a new entry of n_j into FT_i;

◇ When $FT_i[j].t_{exp}$ expires at node n_i:
Remove the entry $FT_i[j]$;

◇ When a packet sender n_i has data packets to send:
Retrieve FT_i; Remove the column of n_{id} and t_{exp} in FT_i;
Normalize the value of metrics in FT_i according to Eq. 1;
Generate $\mathbb{R}^{N\times M}$ as shown in Fig. 1;
Calculate $\mathbb{E}$ of each network metrics according to Eq. 2;
Calculate Φ of each network metrics according to Eq. 4;
Calculate $\mathbb{A}^{fwd}$ of candidate nodes according to Eq. 5;
Calculate $\mathbb{P}^{fwd}$ of candidate nodes according to Eq. 6;
for $n_k \in FT_i[n_{id}]$ /* Select forwarding node */
temp = rand[0,1];
if $\mathbb{A}_k^{fwd} < temp$
continue;
Select n_k as forwarding node;
break;
Send $pkt[n_i, Data]$ to n_k;

FIGURE 3: The pseudocode of the proposed *SPA* algorithm.

$$c = v_i \cdot \sin\theta_i - v_j \cdot \sin\theta_j, \tag{10}$$

$$d = y_i - y_j. \tag{11}$$

If n_i and n_j are moving at the same speed and direction, $LK_{i,j}^{exp}$ becomes ∞.

The link throughput between two adjacent nodes can be modeled as a function of geographical distance between these two nodes. According to [10], [22], the link throughput between n_i and n_j, denoted as $LK_{i,j}^{tp}$, is given as follows

$$LK_{i,j}^{tp} = 10^6 \cdot (-9.09 \cdot \log_2(dist(i,j)) + 72.58), \tag{12}$$

Here, $dist(i, j)$ is the spatial distance between n_i and n_j. The unit of spatial distance is meter, whereas the link throughput is measured in bit/second.

IV. ANALYSIS OF THE PROPOSED STOCHASTIC PACKET FORWARDING ALGORITHM

A mathematical model is developed to analyze *SPA* in terms of the number of relays between source node and destination node and packet delivery ratio, respectively. As shown in

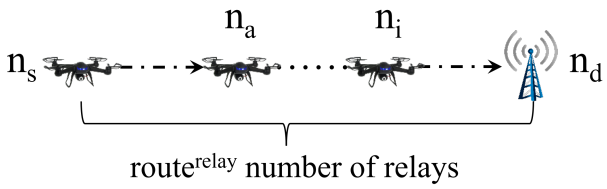


FIGURE 4: A snapshot of the packet forwarding along the route of $route^{relay}$ number of relays between source n_s and destination n_d.

Fig. 4, when the destination node (i.e., stationary ground station) n_d receives the data packet from the source node n_s through *h* number of relays, it indicates that the data packet is successfully delivered. It is assumed that *N* number of nodes are uniformly distributed in a $X \times Y$ network area, where a packet loss ratio is ϑ due to a bad channel quality. In the mathematical model, it is also assumed that the packet loss is mainly caused by bad channel quality or channel error. Suppose that P_{deliv} is the packet delivery ratio, which is the probability that the data packet is transmitted along a route and finally reaches the destination node. As shown in Fig. 4, it is clear that the average number of relays $route^{relay}$ along the route to calculate the P_{deliv} is needed. And the average number of relays $route^{relay}$ can be calculated as

$$route^{relay} = \frac{dist^{avg}}{pro^{relay}}. \tag{13}$$

Here, $dist^{avg}$ and pro^{relay} is the average distance between source node and destination node and the average progress of each relay along the route, respectively.

First, according to [50], the average distance between source and destination in a network area of size $X \times Y$ can be approximated by

$$dist^{avg} \approx \frac{\sqrt{X^2+Y^2}}{2}. \tag{14}$$

Second, the average progress of each relay along the route can be approximated as the average of the maximum distance between the packet sender and each of the neighbor nodes within its transmission range [51], which can be represented as

$$pro^{relay} \approx \frac{2 \cdot N_R^{avg} \cdot R}{2 \cdot N_R^{avg} + 1}. \tag{15}$$

Here, N_R^{avg} is the average number of nodes located within the communication range circle and it is expressed as,

$$N_R^{avg} = \frac{N}{X * Y} \cdot \pi R^2, \tag{16}$$

where *R* is the communication range of each node, $X * Y$ is the size of the network area, and $\frac{N}{X*Y}$ is the node density.

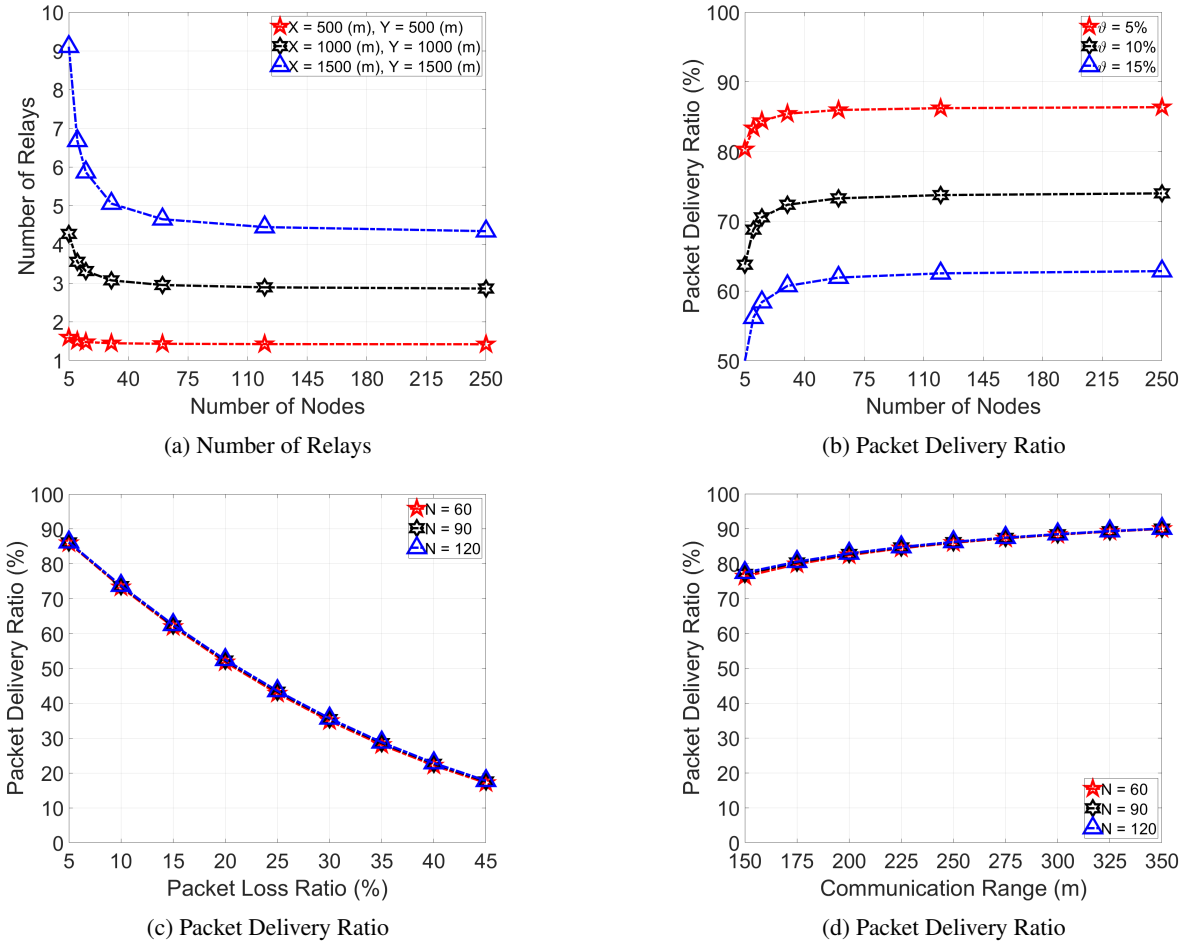
Third, the estimated number of relays along a route can be estimated as

$$\begin{aligned} route^{relay} &\approx \frac{dist^{avg}}{pro^{relay}} \\ &\approx \frac{\sqrt{X^2+Y^2}}{2 \cdot pro^{relay}} \\ &\approx \frac{(2 \cdot N_R^{avg} + 1) \cdot \sqrt{X^2+Y^2}}{4 \cdot N_R^{avg} \cdot R} \\ &\approx \frac{(2N\pi R^2 + XY) \cdot \sqrt{X^2+Y^2}}{4N\pi R^3} \end{aligned} \tag{17}$$

Therefore, the packet delivery ratio P_{deliv} of the data packet transmission along a route with $route^{relay}$ number of relays can be represented as,

$$P_{deliv} = (1-\vartheta)^{route^{relay}} = (1-\vartheta)^{\frac{(2N\pi R^2+XY)\cdot\sqrt{X^2+Y^2}}{4N\pi R^3}}. \tag{18}$$

In Fig. 5, the number of relays between source and destination and the packet delivery ratio are measured by varying the number of nodes (*N*), packet loss ratio (ϑ), and communication range (*R*). Subfig. 5(a) shows the number of relays between source and destination. Here, in a $X \times Y$ network area, 5 to 250 nodes are deployed uniformly. The communication range (*R*) of each node is set to 250 (m). As shown in Subfig. 5(a), when the node density reaches a certain level in the network, the number of relays between source node and destination node is no longer sensitive to the increasing number of nodes in the network. However, as the size of network area increases, the number of relays between source node and destination node significantly increases. This is because the physical distance between source node and destination node increases as the network size increases. In Subfig. 5(b), (c), and (d), the packet delivery ratio is measured against the number of nodes *N*, packet loss ratio ϑ, and communication range *R*. The packet loss ratio ϑ is set to be from 5% to 45%. As shown in Subfig. 5(b), the packet delivery ratio can be maintained above 85% when the packet loss ratio ϑ = 5%. When the packet loss ratio ϑ is increased, a lower packet delivery ratio is observed. This is because a larger ϑ causes more data packets to be lost during the transmission, a less number of data packets can be delivered and a lower packet delivery ratio is observed. In addition, the packet delivery ratio significantly increases when the number of nodes is increased from 5 to 60. However, as the number of nodes increases from 60 to 250, the packet delivery ratio barely increases. In Subfig. 5(c), the packet delivery ratio significantly decreases as the packet loss ratio ϑ increases from 5% to 45%. This is because more data packets will get lost during the transmission with a larger packet loss ratio. The packet delivery ratio is not sensitive to the change of the number of nodes in the network, which has been verified in Subfig. 5(a). Subfig. 5(d) shows that the packet delivery ratio increases as the communication range *R* of each node increases. Since the larger communication range *R* can significantly reduce the physical distance to destination node,



(a) Number of Relays

(b) Packet Delivery Ratio

(c) Packet Delivery Ratio

(d) Packet Delivery Ratio

FIGURE 5: The performance of the number of relays between source node and destination node and packet delivery ratio against the number of nodes, packet loss ratio, and communication range.

a less number of relays will be required to deliver the data packet to destination node. As a result, the data packet has a less chance to get lost during the transmission, and more data packets can be delivered and a higher packet delivery ratio is obtained. When $R = 350$ (m), the packet delivery ratio can reach 90%. However, when the number of nodes in the network changes, the packet delivery ratio is not changing significantly.

V. NETWORK SIMULATION EXPERIMENTS

A. NETWORK SIMULATION TESTBED

A network simulation framework is built in OMNeT++ [9] to evaluate *SPA*'s performance. 5 to 15 nodes are deployed uniformly in a 1000×1000 (m^2) network area. Each node is armed with a radio transceiver and can communicate with other nodes using two-ray ground reflection model. In addition, each node has the same communication range which is set to 300 meters. The random waypoint mobility model [44] is adopted in the framework, where the node randomly selects a destination and moves towards to it with a consistent speed. The moving speed is set to 25 to 50 meter/sec. One (1) data packet is being generated by the source node and its size is 512 Bytes. The entire experiment lasts for 2000 seconds. All network simulation parameters are presented in Table 4.

TABLE 4: Network Simulation Parameters

Parameter	Value
Network area	1000×1000 m^2
Number of nodes	5 to 15
Moving speed	25 to 50 meter/sec
Packet loss ratio	5%
Radio data rate	250 Kbps
Packet size	512 Bytes
Packet injection rate	1 packet/sec
Communication range	300 m
Radio model	CC2420
Simulation time	2000 seconds

Four performance metrics including packet delivery ratio, packet delivery latency, average link lifetime, and the number of delivered packets are measured.

- Packet Delivery Ratio: Packet delivery ratio is calculated as the number of received data packets divided by the total number of generated data packets.

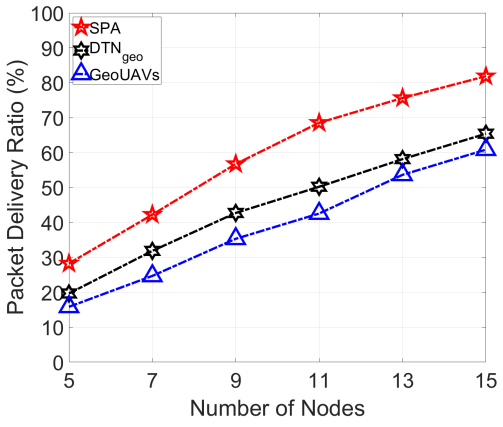
- Packet Delivery Latency: Packet delivery latency is the time from when the data packet is sent out by the source node until it is received by the destination node. If there is a packet loss, the retransmission time will be considered in the calculation of packet delivery latency.
- Average Link Lifetime: Average link lifetime is measured as the average time interval in which one node stays in the transmission area of the selected forwarding node.
- Number of Delivered Packets: The number of delivered packets is the total number of data packets received by the destination node.

The performance of *SPA* is compared with motion-driven packet forwarding algorithm (DTN_{geo}) [10] and geocast routing protocol (*GeoUAVs*) [11]. The fundamental concepts of these two schemes are briefly explained in the following:

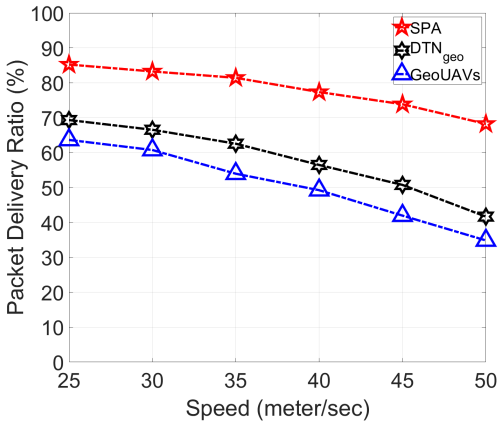
- Motion-Driven Packet Forwarding Algorithm (DTN_{geo}): DTN_{geo} is a packet forwarding approach with delay-tolerant networking support. Each node maintains a topology table which is periodically updated by node status messages transmitted through the out-of-band channel. The source node first tries to find the Dijkstra shortest path to the destination node based on the information of topology table. If the path does not exist, DTN_{geo} applies delay-tolerant forwarding technique, where the data packet will be forwarded to a neighbor node who is physically closer to the destination node.
- Geocast Routing Protocol (*GeoUAVs*): Before packet transmission, the packet sender has to check whether there is other packet relay nodes that are physically located closer to the destination. If the packet sender deduces that there is no other relay node closer to the destination in a timeout period, it designates itself as a relay node and transmits the data packet to other nodes physically closer to the destination. This verification can avoid redundant packet transmissions, and thus permit to reduce the number of duplicated data packets.

B. NETWORK SIMULATION RESULTS AND ANALYSIS

First, the performance of packet delivery ratio is measured by varying the number of nodes and moving speed in Fig. 6. As shown in Subfig. 6(a), when the number of nodes in the network increases from 5 to 15, the packet delivery ratio of *SPA*, DTN_{geo} and GeoUAVs increase significantly. Since the packet sender has more forwarding candidate nodes to choose with more nodes in the network, more data packets can be delivered to the destination. As a result, an increasing packet delivery ratio is obtained. In particular, the packet delivery ratio of *SPA* is higher than that of DTN_{geo} and GeoUAVs. The reason is that *SPA* has a chance to evaluate the link expiration time of adjacent nodes and chooses a more reliable and stable link to send data packets. Thus, a larger number of data packets can be delivered to the destination, which results in a higher packet delivery ratio. In Subfig. 6(b), the packet delivery ratio of *SPA*, DTN_{geo} and GeoUAVs de-



(a)



(b)

FIGURE 6: The performance of packet delivery ratio against the number of nodes and moving speed.

crease as the moving speed increases from 25 to 35 meter/sec. Since a higher velocity makes nodes move more quickly and suddenly, the communication channels that connect adjacent nodes become less stable or even get disconnected. Thus, a few number of data packets can be delivered to the destination along the unstable end-to-end forwarding path, and the packet delivery ratio decreases accordingly. However, *SPA* still outperforms DTN_{geo} and GeoUAVs. The rationale is that the packet sender has more chances (or high probability) to select the forwarding candidate node with more reliable and stable communication link. Therefore, more data packets can be delivered to the destination and a higher packet delivery ratio is observed.

Second, the packet delivery latency is measured by changing the number of nodes and moving speed in Fig. 7. As shown in Subfig. 7(a), the packet delivery latency is not sensitive to the number of nodes in the network. This is because the increasing node density in the network will not affect the number of hops between source node and destination node significantly, the packet delivery latency of three schemes do not have significant changes. However, the packet delivery

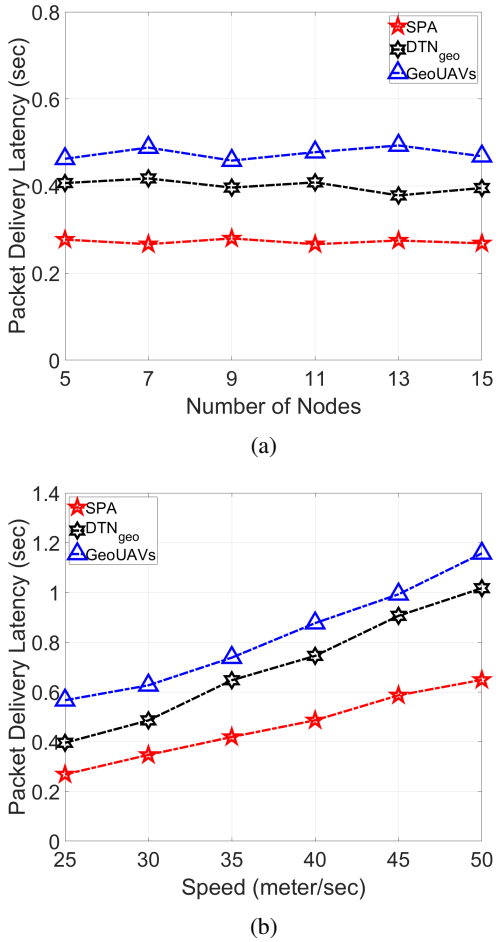


FIGURE 7: The performance of packet delivery latency against the number of nodes and moving speed.

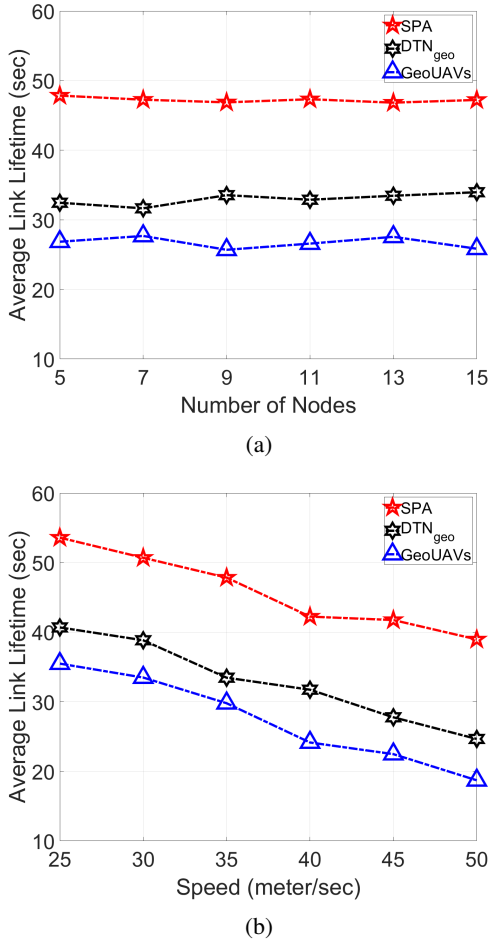
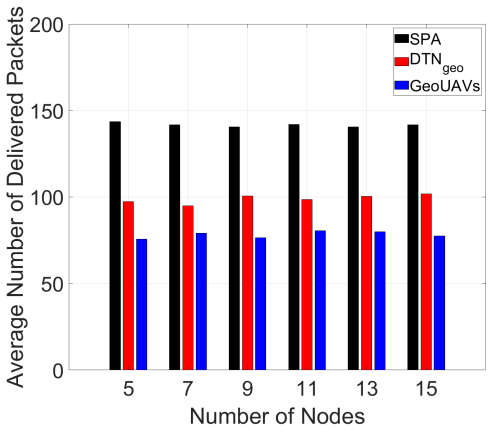


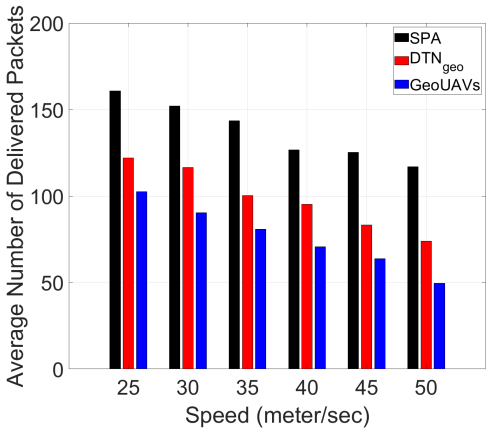
FIGURE 8: The performance of average link lifetime of selected links against the number of nodes and moving speed.

latency of DTN_{geo} and GeoUAVs are clearly higher than that of *SPA*. This is because GeoUAVs experiences a longer timeout period to select the next-hop relay node, as a result, the highest packet delivery latency is observed. In DTN_{geo}, if end-to-end routing path is not available, the packet sender needs to carry the data packet and move to the destination node to deliver the data packet, which result in a longer latency. In Subfig. 7(b), as the moving speed of each node increases, the overall packet delivery latency increases. Since a higher moving speed causes more unreliable links, as a result, more data packets get lost during transmission and packet delivery latency will increase due to retransmission. *SPA* still outperforms both DTN_{geo} and GeoUAVs because a less number of data packets will get lost, and a lower packet delivery latency can be achieved.

Third, the performance of average link lifetime of selected links is obtained by changing the number of nodes and moving speed in Fig. 8. According to Subfig. 8(a), the average link lifetime of *SPA*, DTN_{geo} and GeoUAVs do not change significantly as the number of nodes increases. *SPA* still can achieve 15% and 20% larger average link lifetime compared to DTN_{geo} and GeoUAVs, respectively. Since the link expiration time is considered as one of network metrics to evaluate the forwarding candidate nodes in *SPA*, it is more likely to choose the candidate nodes with larger link expiration time to send the data packets. In the DTN_{geo}, the source node tries to identify the shortest path in terms of transmission delay to send the data packets to the destination. However, the intermediate nodes along the forwarding path might not provide the longest link connection time. In addition, the GeoUAVs tries to find the next-hop relay node that is physically closest to the destination. However, the link between the packet sender and the next-hop relay node might not be stable because of the high mobility of next-hop relay node. Subfig. 8(b) shows that the average link lifetime of three schemes decline when the moving speed of network nodes increase from 25 to 50 meter/sec. This is because the communication link between two adjacent nodes becomes unstable and the link expiration time decreases with a higher moving speed. However, a higher average link lifetime is still obtained by *SPA* because it has preference to select the forwarding candidate node with a larger link expiration time



(a)



(b)

FIGURE 9: The performance of average number of delivered packets against the number of nodes and moving speed.

to send the data packets.

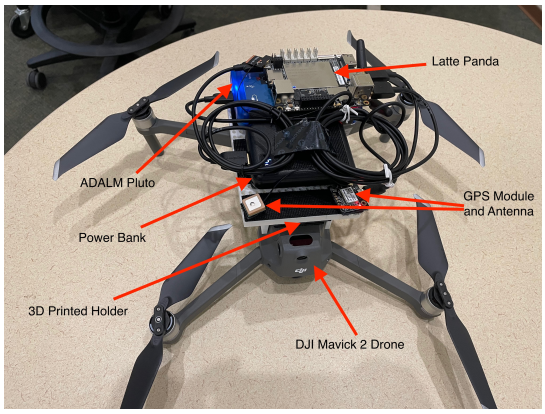
Fourth, the average number of delivered data packets is observed by changing the number of nodes and moving speed in Fig. 9. Please note, the average number of delivered data packets is measured at the simulation time 100 seconds, and does not consider the retransmitted data packets. In Subfig. 9(a), it is clearly shown that *SPA* can deliver more data packets than DTN_{geo} and GeoUAVs. In *SPA*, the forwarding candidate node with a larger link throughput as well as a longer link expiration time always has more chances to be chosen by the packet sender. Thus, a larger number of data packets can be forwarded along a more stable end-to-end path and finally delivered to the destination. The DTN_{geo} only takes into account of transmission delay to select the end-to-end forwarding path. However, the intermediate nodes along the selected forwarding path might provide shorter link connection time, which results in a smaller number of data packets to be delivered to the destination. Subfig. 9(b) shows that the average number of delivered data packets decreases linearly when the moving speed increases. Since the link is less stable and becomes disconnected suddenly with a higher mobility, a smaller number of data packets can be delivered. With a higher moving speed, the communication links become unstable and can easily get disconnected. Thus, some data packets might get lost during the transmission and less number of data packets can be finally delivered. However, *SPA* still outperforms DTN_{geo} and GeoUAVs.

VI. REAL-WORLD TESTBED IMPLEMENTATION AND EXPERIMENTS

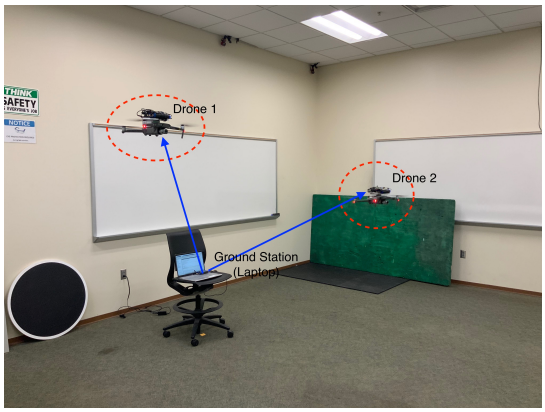
A. REAL-WORLD TESTBED

Hardware Configuration: A real-world testbed with two DJI Mavic 2 Pro drones (serving as packet forwarding candidates) [52], one laptop (serving as a packet sender ground station (GS)), two ADALM Pluto Software Defined Radio (SDR) modules [53], two Latte Panda development boards [54], and two micro GPS circuits [55] is built. ADALM Pluto SDR is a portable, cost-effective, and self-contained SDR that contains the analog transmitter and receiver chains for radio frequency (RF) communication systems. Among others, ADALM Pluto can support baseband implementation of transmitter and receiver signal processing blocks developed in MATLAB environment. Latte Panda is a development board that can run a full version of Windows 10, where the proposed *SPA* along with baseband transmitter and receiver signal processing blocks have been implemented in MATLAB. Note that Latte Panda connected with ADALM Pluto serves as a complete digital and analog chain for communication transceiver. Moreover, off-the-shelf omnidirectional antenna integrated with ADALM Pluto is used for transmission and reception of RF electromagnetic waves. As shown in Subfig. 10(a), each drone is equipped with an ADALM Pluto SDR, a Latte Panda development board, a micro GPS unit, and a power bank. Note that the power bank is capable enough to support the developed drone platform for an hour. The laptop, which serves as a GS, is also connected with an SDR. To bind ADALM Pluto SDR, Latte Panda development board, micro GPS circuit, power bank, and drone together, a specific plastic holder is built using 3D printer. The developed real-world testbed with two drone platforms and one laptop is shown in Subfig. 10(b).

Baseband Transceiver: An IEEE 802.11a WLAN transmitter and receiver chain [56] is developed, where the baseband (digital) processing is accomplished with MATLAB and analog signal processing is done in SDR. The transmitter and receiver chains are implemented at the GS and two drones. The WLAN Toolbox from MATLAB is exploited to generate 802.11a standard compliant medium access control (MAC) frames [57]. The data bits are first segmented into multiple MAC service data units (MSDU) piggybacked with header information, then MAC protocol data units (MPDUs) are created. MPDUs are then converted into physical layer service data units (PSDUs) and fed into WLAN waveform generator. Precisely, 64 QAM modulation and convolutional channel coding with rate 2/3 on 20 MHz bandwidth is used. These parameters are used for all transmission links (GS to drones and drones to GS). Orthogonal frequency division



(a) Developed Drone Platform



(b) Developed Real-World Testbed

FIGURE 10: (a) Developed drone platform with SDR, Latte Panda development board, micro GPS circuit, power bank bounded together with 3D-printed plastic holder. (b) Developed real-world testbed with two drone platforms and one laptop.

multiplexing (OFDM) scheme is deployed with 52 subcarriers. The transmit power gain and carrier frequency is set to 0 dBm and 2.432 GHz, respectively. The transmissions from GS to drones and from drones to GS are organized by time division duplex (TDD) method.

B. FIELD EXPERIMENT SCENARIOS, RESULTS, AND PERFORMANCE ANALYSIS

The basic setup is inspired by a generic packet forwarding scenario, where one packet sender GS is placed together with a couple of potential packet forwarding candidates (drones). Each packet forwarding candidate periodically sends its GPS position, moving speed, and orientation information to packet sender GS. After receiving location and motion information, the packet sender GS calculates link expiration time and link throughput according to Eq. (7) and (12), respectively, and updates the forwarding candidate table. Then, the packet sender GS selects the packet forwarding drone based on the proposed *SPA* and sends pre-defined data packets. The

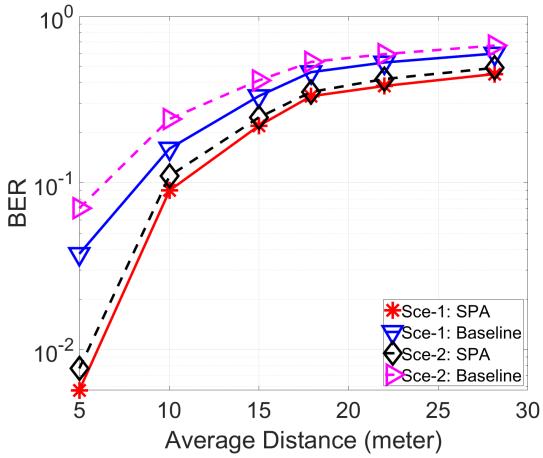


FIGURE 11: The performance of BER against average distance between packet sender GS and drones.

performance metrics that is being used to evaluate *SPA* in the field experiments is the overall bit error rate (BER). The BER is computed as the number of received error bits divided by the number of transmitted bits. In other words, the packet forwarding drone will compare the received data packet with the pre-defined data packet and calculate the BER. For performance comparison and analysis, a baseline scheme is also implemented. The basic idea of baseline scheme is that the packet sender GS randomly selects one of two packet forwarding candidates following uniform distribution (i.e., equal selection probability).

The performance of BER with varying average distance between packet sender GS and drones is shown in Fig. 11. Two scenarios with different drone velocities are considered. In the first scenario (denoted as Sce-1), both drones travel at a mean velocity of 1 meter/sec. In the second scenario (denoted as Sce-1), drones' mean velocity is set to 3 meter/sec. An operation cycle is defined as an time interval from when drones send out GPS/motion information to packet sender GS to the time when drones receive the pre-defined data packets from packet sender GS. An operation cycle is set to 1 minute and the entire experiment lasts for 30 minutes. In addition, each drone is carefully controlled to travel back-and-forth along a pre-defined path. It is observed that the BER increases as the average distance increases for both *SPA* and the baseline scheme. The lower BER observed by SPA in compared to the baseline scheme over the considered range of average distance (between GS and drones) demonstrates the usefulness of implementing the SPA. Please note that this observation holds for both scenarios with different drone velocities. Compared to the baseline scheme, it is worth mentioning that *SPA* exhibits much lower BER as the average distance increases. Furthermore, it is also shown that the BER of *SPA* and the baseline scheme increase as the velocity of drones increases. However, *SPA* still outperforms the baseline scheme.

C. INSIGHTFUL LESSONS FROM TESTBED DEVELOPMENT

1) Selection of SDR

Observations: We used ADALM-PLUTO as SDR (connected with baseband processing chain configured in MATLAB) for GS and drones. The main reason for choosing ADALM-PLUTO for the drone is a) being lightweight, b) having an option to use off-the-shelf MATLAB function, c) good performance of the (omnidirectional) antenna that comes with the module. The (light) weight of ADALM-PLUTO is within the recommended limit of DJI Mavic 2 Pro drones (to carry the SDR without any issue) used for the experiments. Moreover, the amenable physical structure of this SDR helps to tightly bind it with the drone along with the Windows development board (Latte Panda). Furthermore, the off-the-shelf MATLAB function demonstrates robust performance in interfacing the digital baseband signal processing chain (IEEE 802.11a WLAN configured in MATLAB) with the SDR module. Finally, the excellent performance of the off-the-shelf antenna results in an excellent performance to conduct our experiments. While adding a more directional antenna can enhance the performance, it can lead to issues and extra weight for the drones to carry over a longer distance.

Improvements/Alternatives: The objective of the experiments conducted in this paper is to show the effectiveness of the proposed SPA compared to the baseline scheme while considering practical limits in real-time communications. An SDR customized with the algorithms implemented in the field-programmable gate array (FPGA) level might be a better choice for commercial-level products. In such cases, baseband digital signal processing chain can be configured with high precision algorithms developed in embedded programming. Furthermore, custom-made antennas can be designed while considering the weight limits of the drones and radiation pattern requirements.

2) Selection of Signal Processing Algorithms

Observations: The experiments conducted from the developed testbed leveraged the signal processing algorithms that entail low computational complexity to maximize the battery life of the drones. For instance, we considered a minimum mean square error (MMSE) based signal detection algorithm at the receiver, balancing performance and complexity [58]. Moreover, linear MMSE (LMMSE) based estimation (at pilot signals) with a non-linear interpolation algorithm was used for channel estimation purposes. Moreover, convolutional codes were used for channel coding.

Improvements/Alternatives: While the considered signal processing algorithms showed reasonably good performance from these experiments, more advanced tools could enhance the error rate performance as long as the real-time computational complexities are ensured to stay within a prescribed limit. For example, a maximum likelihood (ML) detector can be used that efficiently equalizes the wireless channel and thereby shows superior performance in comparison to linear receivers, e.g., zero-forcing (ZF) and MMSE [58]. To alleviate the computational complexity of the ML detector, we could implement a sphere decoding (SD) algorithm for ML-based detection. Advanced channel estimation techniques like adaptive Wiener filter-based channel estimation technique can enhance the system performance along with angle-Doppler estimation algorithm for estimating and thereby nullifying the Doppler effect of the drones. To increase the fidelity of data transmission, advanced channel coding techniques, e.g., polar code, low-density parity-check (LDPC) codes, etc., can be deployed at the transmitter [58]. Artificial intelligence (AI) based signal processing algorithms can be a great option to enhance the system performance without increasing the real-time computational complexity. However, careful attention must be paid while training the models offline with appropriate training data and deploying the inference models on the Latte Panda boards (attached to the drones). A different carrier frequency can be used depending on the availability of the frequency spectrum.

3) Choice of Baseband Processing for Drone Transceivers

Observations: As mentioned earlier, we used Latte Panda, the development board on the Windows platform, to configure the baseband digital signal processing chain using MATLAB. This development board works as the central processing unit for the transceiver node attached to the drone. The main reason for using Latte Panda as the central processing unit is the ease of use and support of MATLAB. In addition, the power supply used for transceivers was capable and robust enough to support Latte Panda and SDR together.

Improvements/Alternatives: As an alternative to Latte Panda, we could use Raspberry Pi and configure the baseband processing chain there. Instead of IEEE 802.11a WLAN, LTE-A/NR can be used to develop a physical layer transmitter and receiver by leveraging different physical shared channels. Open-source tools, e.g., srsRAN [59], OpenAirInterface [60], etc., can be used as an alternative to MATLAB-based implementation.

4) Selection of Propagation Environment

Observations: The experiments were conducted in a specific location, and multiple trials were made to obtain the average quantity of the performance metric, BER. Our objective was to assess SPA over a random selection scheme, and hence, the considered propagation environment was adequate to observe the relative performances.

Improvements/Alternatives: To observe the performance of the proposed routing algorithm in different propagation environments like densely urban, urban, rural, etc., controlled experiments can be conducted in these locations. The relative performances can provide insights to the system designers and physical layer algorithm developers. Moreover, with appropriate Doppler spread estimation and nullification algorithm, drones running with higher velocities will be considered in the experimentation.

VII. DISCUSSION

A. DESIGN FEATURES, CONSTRAINTS, AND FUTURE EXTENSIONS

SPA is designed with four enticing features. First of all, a weight is assigned to different network metrics based on the entropy weight theory. In this process, the coefficient parameter is utilized to regulate the value range of network metrics. In addition, the coefficient parameter can also be used to adjust the effect of the particular network metrics for subjective preference. Thus, *SPA* can be easily adjusted for various applications with different network characteristics, such as high or low link throughput, and high or low moving speed. Second, the forwarding node will be stochastically selected according to the forwarding probability. Thus, each forwarding candidate node will have a chance to be involved in the packet forwarding operation. To some extent, stochastically selecting forwarding node will achieve the effect of conditional load balancing because network traffic can be relatively distributed by choosing different forwarding node. Third, frequently broadcasted *HELLO* messages piggybacked with geographical position and mobility information can help packet sender quickly detect the change of forwarding candidate nodes, and then re-evaluate all forwarding candidate nodes for further packet forwarding operations. Fourth, when *SPA* is designed, the potential extensibility and flexibility are considered. Therefore, additional real-time network metrics can be smoothly included in *SPA*. For example, each node is equipped with the limited amount of battery energy. In order to extend the operational time of each node and the network lifetime, the network traffic should be relatively balanced among all potential forwarding nodes. By considering the residual energy of nodes as a new factor to calculate forwarding probability, the node with a less amount of residual energy will have a less chance to be chosen to transfer data packets, resulting in less energy consumption.

In *SPA*, there are a few constraints that need to be further discussed. First, *SPA* is limited to select one forwarding node to send data packet. However, due to a bad channel quality or packet collisions, the data packet can be lost during the transmission. As a result, the data packet can not be delivered to the destination node. Second, due to the nature of high mobility of nodes in FANETs, it is always possible that the packet sender does not have any neighbor node to forward the data packet. In this case, the packet sender has to carry the data packet and physically move to a waypoint that might be farther away from the destination node. As a result, the packet delivery latency could increase.

To address the above-mentioned constraints, *SPA* can be further investigated with the following improvements.

1) Multipath Forwarding

In *SPA*, the number of forwarding node is limited to one, which does not provide strong fault tolerance and network resiliency for FANETs. Thus, there is a plan to extend *SPA* with multipath forwarding technique [22] so that communication failure or disruption does not affect the system performance. If there is any transmission error that causes the packet lost along one forwarding path, the data packet still can be received through another active forwarding path. Correspondingly, the network performance such as end-to-end delay and packet delivery ratio can be improved.

2) Greedy Forwarding

Since each node can access its geographic position and mobility information, the concept of greedy forwarding can be applied to make forwarding decision. To be specific, the packet sender can choose to forward the data packets to the adjacent node that is physically closer to the destination node [61]. Thus, the physical distance between the candidate forwarding node and the destination node can also be considered as real-time network metrics along with link throughput and link expiration time to select forwarding node to send data packets. Therefore, the data packet can be delivered to the node who is physically closer to the destination node and the less number of hops along a route can be achieved. As a result, the data packets have a low chance to get lost during the transmission and the end-to-end communication latency can be reduced as well.

B. SECURITY ISSUES AND CORRESPONDING COUNTERMEASURES

The security resilience of *SPA* to potential attacks is discussed and the corresponding countermeasures to secure *SPA* are proposed.

1) Jamming Attack

FANETs may suffer from jamming attack that blankets out a business-critical area by intentionally broadcasting disruption or interference signals [32]. As a result, the on-going communications within the disturbed area can be interrupted by the interference signals, which causes significant packet losses. For example, wireless communication channels are vulnerable to jamming attacks which can cause the degradation of network performance significantly [62]. In order to defend against jamming attack, node-disjoint multipath forwarding technique can be applied, where more than one forwarding path are established to deliver the data packets to the destination. If the data packet gets lost along one forwarding path because of jamming attack, the data packet can still reach the destination node along the other forwarding path. In addition to avoid intentional jamming, multipath forwarding can also help to reduce the negative effects of route coupling.

2) Selective Forwarding Attack

Similar to many other forwarding protocols widely used in ad hoc wireless networks, *SPA* might be vulnerable to well-known selective forwarding attack. In the selective forwarding attack, an adversary intentionally refuses to forward some packets or simply chooses to drop them, ensuring that the packets cannot be propagated towards destination [63]. In order to detect the selective forwarding attack, an explicit confirmation based approach can be deployed in FANETs.

However, since the end-to-end forwarding path might not exist between the source node and the destination, it is difficult or impossible for the source node to accurately detect the selective forwarding attack. Thus, an intermediate node detection scheme can be proposed. The basic idea is that certain intermediate nodes reply an acknowledgment packet to the previous packet sender/forwarder after receiving the data packets. The selective forwarding attack can be easily detected if an intermediate node does not receive the required number of acknowledgment packets from upstream nodes within a period of time.

VIII. CONCLUSION

This article proposed a stochastic packet forwarding algorithm to deliver data packets efficiently and reliably in FANETs. In order to show the effectiveness of the algorithm theoretically, an analytical model was also presented. In terms of performance evaluation, an OMNeT++ based network simulation framework was developed and extensive simulation experiments were conducted. In addition, a real-world testbed was established to investigate the proposed algorithm, which complemented the network simulation experiments. Extensive experimental finding showed that the proposed algorithm is a reliable and efficient approach to deliver data packets in FANETs. For the future work, other important network metrics can be included in the algorithm to efficiently balance traffic load and extend network lifetime. There is also a plan to expand the testbed (e.g., broader range of devices) on the campus so that the proposed algorithm can be further evaluated.

REFERENCES

[1] C. Pu, "Stochastic Packet Forwarding Algorithm in Flying Ad Hoc Networks," in *IEEE Proc. MILCOM*, 2019, pp. 494–499.

[2] *Commercial Drone Market*, Last accessed: Nov 03, 2021, https://www.grandviewresearch.com/.

[3] *$98 Billion Expected for Military Drone Market*, Last accessed: Nov 11, 2021, https://www.nationaldefensemagazine.org/articles/2020/1/6/98-billion-expected-for-military-drone-market.

[4] M. Yahuza, M. Idris, I. Ahmedy, A. Wahab, T. Nandy, N. Noor, and A. Bala, "Internet of Drones Security and Privacy Issues: Taxonomy and Open Challenges," *IEEE Access*, vol. 9, pp. 57 243–57 270, 2021.

[5] H. Asano, H. Okada, C. Naila, and M. Katayama, "Flight Model Using Voronoi Tessellation for a Delay-Tolerant Wireless Relay Network Using Drones," *IEEE Access*, vol. 9, pp. 13 064–13 075, 2021.

[6] G. Wang, B. Lee, J. Ahn, and G. Cho, "A UAV-assisted CH election framework for secure data collection in wireless sensor networks," *Future Generation Computer Systems*, vol. 102, pp. 152–162, 2020.

[7] V. Sharma, N. Sharma, M. Rehmani, and H. Pervaiz, "Control Over Skies: Survivability, Coverage, and Mobility Laws for Hierarchical Aerial Base Stations," *IEEE Pervasive Computing*, vol. 20, no. 3, pp. 51–59, 2021.

[8] K. Namuduri and R. Pendse, "Multicriteria UAV Base Stations Placement for Disaster Management," *IEEE Sensors J.*, vol. 12, no. 6, pp. 1828–1835, 2011.

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

[10] 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.

[11] F. Bousbaa, C. Kerrache, Z. Mahi, A. Tahari, N. Lagraa, and M. Yagoubi, "GeoUAVs: A new geocast routing protocol for fleet of UAVs," *Computer Communications*, vol. 149, pp. 259–269, 2020.

[12] 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.

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

[14] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized Link State Routing Protocol for Ad Hoc Networks," in *Proc. IEEE INMIC*, 2001, pp. 62–68.

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

[16] 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.

[17] Y. Zheng, Y. Wang, Z. Li, L. Dong, Y. Jiang, and H. Zhang, "A Mobility and Load Aware OLSR Routing Protocol for UAV Mobile Ad-hoc Networks," in *Proc. IETICT*, 2014, pp. 1–7.

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

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

[20] J. Forsmann, R. Hiromoto, and J. Svoboda, "A Time-Slotted On-Demand Routing Protocol for Mobile Ad Hoc Unmanned Vehicle Systems," in *Proc. SPIE*, 2007, pp. 1–11.

[21] J. Maxa, M. Mahmoud, and N. Larrieu, "Joint Model-Driven Design and Real Experiment-based Validation for a Secure UAV Ad Hoc Network Routing Protocol," in *IEEE Proc. ICNS*, 2016, pp. 1E2–1–1E2–16.

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

[23] Z. Haas, M. Pearlman, and P. Samar, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," *Internet Draft, draft-ietfmanet-zone-zrp-04.txt*, July 2002.

[24] 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.

[25] V. Park and M. Corson, "Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification," *Internet Draft, Internet Engineering Task Force*, 1997.

[26] X. Li and J. Yan, "LEPR: Link Stability Estimation-based Preemptive Routing protocol for Flying Ad Hoc Networks," in *IEEE Proc. ISCC*, 2017, pp. 1079–1084.

[27] C. Zang and S. Zang, "Mobility Prediction Clustering Algorithm for UAV Networking," in *Proc. IEEE GLOBECOM Wkshps*, 2011, pp. 1158–1161.

[28] B. Karp and H. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," in *Proc. MobiCom*, 2000, pp. 243–254.

[29] S. Choi, H. Hussen, J. Park, and J. Kim, "Geolocation-Based Routing Protocol for Flying Ad Hoc Networks (FANETs)," in *Proc. IEEE ICUFN*, 2018, pp. 50–52.

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

[31] G. Gankhuyag, A. Shrestha, and S. Yoo, "Robust and Reliable Predictive Routing Strategy for Flying Ad-hoc Networks," *IEEE Access*, vol. 5, pp. 643–654, 2017.

[32] 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.

[33] Z. Zhao *et al.*, "Software-defined unmanned aerial vehicles networking for video dissemination services," *Ad Hoc Networks*, vol. 83, pp. 68–77, 2019.

[34] 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.

[35] L. Lin, Q. Sun, S. Wang, and F. Yang, "A Geographic Mobility Prediction Routing Protocol for Ad Hoc UAV Network," in *Proc. IEEE GLOBECOM Wkshps*, 2012, pp. 1597–1602.

[36] D. Medina, F. Hoffmann, F. Rossetto, and C. Rokitansky, "A Geographic Routing Strategy for North Atlantic In-Flight Internet Access Via Airborne Mesh Networking," *IEEE/ACM TON*, vol. 20, no. 4, pp. 1231–1244, 2012.

[37] K. Liu, J. Zhang, and T. Zhang, "The Clustering Algorithm of UAV Networking in Near-space," in *Proc. IEEE ISAPE*, 2008, pp. 1550–1553.

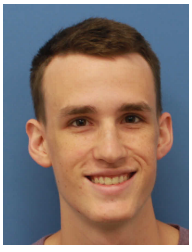
[38] W. Zafar and B. Khan, "A reliable, delay bounded and less complex communication protocol for multicluster FANETs," *Digital Communications and Networks*, vol. 3, no. 1, pp. 30–38, 2017.

[39] T. e Silva, C. de Melo, P. Cumino, D. Rosario, E. Cerqueira, and E. D. Freitas, "STFANET: SDN-based Topology Management for Flying Ad Hoc Network," *IEEE Access*, vol. 7, pp. 173 499–173 514, 2019.

[40] I. Khan, I. Qureshi, M. Aziz, T. Cheema, and S. Shah, "Smart IoT Control-Based Nature Inspired Energy Efficient Routing Protocol for Flying Ad Hoc Network (FANET)," *IEEE Access*, vol. 8, pp. 56 371–56 378, 2020.
[41] I. Khan, S. Shah, L. Wang, M. Aziz, T. Stephan, and N. Kumar, "Routing protocols & unmanned aerial vehicles autonomous localization in flying networks," *International Journal of Communication Systems*, p. e4885, 2021.
[42] I. Khan, M. A. Hassan, M. Fayaz, J. Gwak, and M. Aziz, "Improved Sequencing Heuristic DSDV Protocol Using Nomadic Mobility Model for FANETS," *CMC-COMPUTERS MATERIALS & CONTINUA*, vol. 70, no. 2, pp. 3653–3666, 2022.
[43] I. Khan, M. Hassan, M. Alshehri, M. Ikram, H. Alyamani, R. Alturki, and V. Hoang, "Monitoring System-Based Flying IoT in Public Health and Sports Using Ant-Enabled Energy-Aware Routing," *Journal of Healthcare Engineering*, vol. 2021, 2021.
[44] D. Lakew, U. Sa'ad, N. Dao, W. Na, and S. Cho, "Routing in Flying Ad Hoc Networks: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1071–1120, 2020.
[45] 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.
[46] 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.
[47] 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.
[48] C. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.
[49] 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.
[50] C. Pu, S. Lim, C. Jinseok, and J. Byungkwan, "Active Detection in Mitigating Routing Misbehavior for MANETs," *Wireless Network, Springer*, vol. 25, no. 4, pp. 1669–1683, 2019.
[51] K. Liu, J. Deng, P. Varshney, and K. Balakrishnan, "An Acknowledgment-Based Approach for the Detection of Routing Misbehavior in MANETs," *IEEE Trans. Mobile Comput.*, no. 5, pp. 536–550, 2007.
[52] *DJI Mavic 2*, https://www.dji.com/mavic-2.
[53] *ADALM Pluto Software Defined Radio*, https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/adalm-pluto.html.
[54] *Latte Panda*, https://www.lattepanda.com/.
[55] *Tiny GPS Tracker*, https://www.instructables.com/id/Tiny-GPS-Tracker/.
[56] "ISO/IEC/IEEE International Standard - Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *ISO/IEC/IEEE 8802-11:2012(E)*, pp. 1 – 2798, 2012.
[57] *MATLAB WLAN Toolbox*, https://www.mathworks.com/products/wlan.html.
[58] *Digital Communications, 5th expanded ed*, 2007.
[59] *srsRAN*, Last accessed: Nov 28, 2021, https://www.srslte.com/.
[60] *Open Air*, Last accessed: Nov 28, 2021, https://openairinterface.org/.
[61] C. Pu and L. Carpenter, "To Route or To Ferry: A Hybrid Packet Forwarding Algorithm in Flying Ad Hoc Networks," in *IEEE Proc. NCA*, 2019, pp. 367–374.
[62] A. Alipour-Fanid, M. Dabaghchian, and K. Zeng, "Impact of Jamming Attacks on Vehicular Cooperative Adaptive Cruise Control Systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12 679–12 693, 2020.
[63] C. Pu and S. Lim, "A Light-Weight Countermeasure to Forwarding Misbehavior in Wireless Sensor Networks: Design, Analysis, and Evaluation," *IEEE Systems Journal*, vol. 12, no. 1, pp. 834–842, 2018.

CONG PU (S'15–M'16) received the B.S. degree in Computer Science and Technology from Zhengzhou University, China, in 2009, and the M.S. and Ph.D. degrees in Computer Science from Texas Tech University in 2013 and 2016, respectively. From 2014 to 2016, he was an Instructor with the Department of Computer Science, Texas Tech University, while he was working towards Ph.D. degree. He is currently an Assistant Professor with the Department of Computer Sciences and Electrical Engineering, Marshall University, Huntington, WV, USA. His primary research interests include cryptography, network security, wireless networks, mobile computing, and information-centric networking. He served as a technical program committee member in many international conferences. He was a reviewer for many IEEE, ACM, Elsevier, and Springer journals. He is also serving as Associate Editor of several journals. He received 2015 Helen Devitt Jones Excellence in Graduate Teaching Award at Texas Tech University. He was the recipient of 2018 NASA WVSGC Research Initiation Grant, 2020 NASA EPSCoR Research Seed Grant, 2020 Open Education Resources (OER) Grant Award, 2018 John Marshall Summer Scholar Award. He received 2019 IEEE ICDIS Best Paper Award. He was the Winner of 2017 Design for Delight (D4D) Innovation Challenge Competition as a Faculty Coach (Marshall University and Intuit Inc.). He was a member of Computer Science Workgroup for West Virginia Department of Education to increase and strengthen computer science education in West Virginia. He was nominated by West Virginia Department of Education to participate in Educational Testing Services Standard Setting Study in EST.

IMTIAZ AHMED is an Assistant Professor in the Department of Electrical Engineering and Computer Science at Howard University, Washington, DC, USA. He works in the areas of wireless communications, signal processing, and computer networks. After finishing his Ph.D. in Electrical and Computer Engineering from the University of British Columbia, Vancouver, BC, Canada, Dr. Ahmed worked at Intel Corporation, San Diego, California, USA as a wireless systems engineer and Marshall University, Huntington, WV, USA as an Assistant Professor. Currently, he is working on artificial intelligence aided physical layer design, integration of aerial and terrestrial communication networks, communication with energy harvesting nodes, etc.



EVAN ALLEN was a student with the Department of Computer Sciences and Electrical Engineering, Marshall University, Huntington, WV, USA. His research interests include wireless networks and mobile computing.

KIM-KWANG RAYMOND CHOO (SM'15) received the Ph.D. in Information Security in 2006 from Queensland University of Technology, Australia. He currently holds the Cloud Technology Endowed Professorship at The University of Texas at San Antonio (UTSA). In 2016, he was named the Cybersecurity Educator of the Year - APAC (Cybersecurity Excellence Awards are produced in cooperation with the Information Security Community on LinkedIn), and in 2015 he and his team won the Digital Forensics Research Challenge organized by Germany's University of Erlangen-Nuremberg. He is the recipient of the 2019 IEEE Technical Committee on Scalable Computing (TCSC) Award for Excellence in Scalable Computing (Middle Career Researcher), 2018 UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for Tenured Faculty, Outstanding Associate Editor of 2018 for IEEE Access, British Computer Society's 2019 Wilkes Award Runner-up, 2019 EURASIP Journal on Wireless Communications and Networking (JWCN) Best Paper Award, Korea Information Processing Society's Journal of Information Processing Systems (JIPS) Survey Paper Award (Gold) 2019, IEEE Blockchain 2019 Outstanding Paper Award, Inscrypt 2019 Best Student Paper Award, IEEE TrustCom 2018 Best Paper Award, ESORICS 2015 Best Research Paper Award, 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, Fulbright Scholarship in 2009, 2008 Australia Day Achievement Medallion, and British Computer Society's Wilkes Award in 2008. His research has been funded by the National Science Foundation, NASA, CPS Energy, LGS Innovations, Texas National Security Network Excellence Fund, Australian Government National Drug Law Enforcement Research Fund, Australian Government Cooperative Research Centre for Data to Decision, auDA Foundation, Government of South Australia, BAE Systems stratsec, Australasian Institute of Judicial Administration Incorporated, Australian Research Council, etc. He is also a Fellow of the Australian Computer Society, and Co-Chair of IEEE Multimedia Communications Technical Committee's Digital Rights Management for Multimedia Interest Group.