IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 21, 2018, accepted October 31, 2018, date of publication November 5, 2018,
date of current version December 7, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2879758

Jamming-Resilient Multipath Routing Protocol for Flying Ad Hoc Networks

CONG PU, (Member, IEEE)
Weisberg Division of Computer Science, Marshall University, Huntington, WV 25755, USA
e-mail: puc@marshall.edu

This work was supported by Startup grant in the Weisberg Division of Computer Science and 2018 John Marshall University Summer Scholars Awards at Marshall University.

ABSTRACT As a result of the rapid technological advances on electronic, sensors and communication technologies, and increasingly popular multi-sized unmanned aerial vehicles, also referred to as drones, flying ad hoc networks (FANETs) are rapidly proliferating and leading the emergence of Internet of Drones and its applications. Because of the versatility, flexibility, easy installation, and relatively small operating expenses of drones, FANETs have enormous potential in the public and civil domains. However, due to unique characteristics of FANETs, routing demands of FANETs go beyond the needs of mobile ad hoc networks and vehicular ad hoc networks. In this paper, we propose a jamming-resilient multipath routing protocol, also called *JarmRout*, so that intentional jamming and disruption or isolated and localized failures do not interrupt the overall network performance of FANETs. To achieve this goal, the JarmRout relies on a combination of three major schemes that are link quality scheme, traffic load scheme, and spatial distance scheme. We present a simple analytical model and its numerical result in terms of *RREP* packet reception rate of source node. We also evaluate the proposed routing protocol through extensive simulation experiments using the OMNeT++ and compare its performance with three representative routing protocols that are dynamic source routing, optimized link state routing, and split multipath routing. Simulation results show that the JarmRout can not only improve packet delivery ratio and packet delivery latency but also can reduce end-to-end communication outage rate without introducing extra communication overhead, indicating a viable approach to improve network resiliency in the presence of malicious jammers in FANETs.

INDEX TERMS Internet-of-Drones (IoD), flying ad hoc networks, routing protocol, multipath routing, intentional jamming.

I. INTRODUCTION

Usage of unmanned aerial vehicles (UAVs), also referred to as drones, is expected to rise at unprecedented rates due to growing interest from hobbyists, researchers, and investors. As the number of drones rapidly increases, Internet-of-Drones (IoD) and its applications are expeditiously proliferating, where a myriad of multi-sized drones seamlessly interact with each other through zone service providers to realize the goal of coordinating the access of drones to controlled airspace and providing navigation services [1]. It has been predicted that hobbyist drones purchases and sales of drones for commercial purposes are expected to grow to 4.3 million and 2.7 million by 2020, respectively [2]. Economic growth of drone industry in the U.S., including on-demand package delivery, traffic and wild life surveillance, inspection of infrastructure, aerial photography, urban safety, military scouting, and so on, is also said to be considerable for businesses. In 2020, the U.S. drone industry is anticipated to generate some 4 billion U.S. dollars [3]. With the prevalence of wireless connectivity and fog computing as well as the rapid technological advances on electronic, sensors and communication technologies, we envision a future in which seamlessly blended drones in the realm of IoD will lead to the further improvement of our lives.

As a part of speedily emerging IoD, Flying Ad Hoc Networks (FANETs) are playing a remarkable role in the realization of ubiquitous computing and communications, where a set of drones faithfully and collaboratively route data packets to a destination in order to achieve the goal of sharing information and knowledge and coordinating decisions. Over the last decades, many researchers have explicitly studied the communication algorithms and routing protocols in Mobile

Ad Hoc Networks (MANETs) and Vehicular Ad Hoc Networks (VANETs), respectively. However, due to the high mobility, sparse deployment, drastically changing network topology, intermittently connected communication links, and intentional jamming and disruption, those mechanisms that were specifically designed for MANETs or VANETs cannot be directly applied in FANETs. In other words, routing demands of FANETs go beyond the needs of MANETs and VANETs [4]. For example, network performance (e.g., packet delivery ratio) can significantly degrade in MANETs with high mobility because of frequent link errors [5]. In sparse networks, network partitions may last for significantly long periods and lead to buffer contention because messages cannot be removed from buffer and new messages might be generated, resulting in longer transmission delay [6]. In IEEE 802.11 based ad hoc networks, a saboteur can easily degrade the network performance significantly by continually transmitting jamming signals on the shared wireless medium [7].

In light of these facts, we propose a novel multipath routing protocol to provide efficient and reliable communication and data transmission as well as improve network resiliency in the presence of malicious jammers in FANETs. The main contributions of this paper can be summarized as follows:

1) We propose a jamming-resilient multipath routing protocol, also called *JarmRout*, so that intentional jamming and disruption, or isolated and localized failures do not interrupt the overall network performance of FANETs.

   The JarmRout is designed based on three major schemes: link quality scheme, traffic load scheme, and spatial distance scheme.

   - The link quality scheme is proposed to differentiate link qualities between a node and its neighbor nodes by using the statistical information of received signal strength indication (RSSI) of received packets.
   - In the traffic load scheme, each node computes its current traffic load by taking account of MAC layer channel contention information and the number of packets stored in the buffer.
   - The spatial distance scheme calculates the spatial separation distance of multiple paths to find the maximally spatial node-disjoint multipath between source and destination nodes.

2) We present a simple analytical model and its numerical result in terms of *RREP* packet reception rate of source node.

   We also revisit three representative routing protocols, which are dynamic source routing (DSR) [8], optimized link state routing (OLSR) [9], and split multipath routing (SMR) [10], and modify them to work in FANETs for performance comparison.

3) We discuss the proposed JarmRout routing protocol in terms of its features, constraints, and possible extensions, and then investigate the immunity of JarmRout to other three well-known attacks in FANETs.

We develop a customized discrete event driven simulation framework by using OMNeT++ [11] and evaluate its performance through extensive simulation experiments in terms of packet delivery ratio, packet delivery latency, end-to-end communication outage rate, and energy consumption. The simulation results indicate that the JarmRout can not only improve packet delivery ratio and packet delivery latency, but also can reduce end-to-end communication outage rate without introducing extra communication overhead, indicating a viable approach to improve network resiliency in the presence of malicious jammers in FANETs.

The rest of the paper is organized as follows. Prior schemes and mechanisms are provided and analyzed in terms of five categories in Section II. A system model and the proposed JarmRout routing protocol are presented in Section III. A simple analytical model and its numeric result are presented in Section IV. Section V presents simulation results and their analyses. We further discuss the JarmRout in Section VI. Finally, concluding remarks are provided in Section VII.

II. RELATED WORK

Flying Ad Hoc Networks (FANETs) are considered as a subclass from Mobile Ad Hoc Networks (MANETs) and Vehicular Ad Hoc Networks (VANETs), therefore, common ideas and strategies could be shared for data delivery [12]. However, for the right functionality of data delivery, the techniques that are specifically designed for MANETs and VANETs have to be adapted to specific characteristics and challenges in FANETs, e.g., high mobility, sparse deployment, drastically changing network topology, intermittently connected communication links, and intentional jamming and disruption. In this section, we categorize and analyze existing routing protocols in FANETs in terms of static routing, proactive routing, reactive routing, hybrid routing, and other approaches.

A. STATIC ROUTING PROTOCOLS

Static routing tables are computed and loaded when the task starts, and these tables cannot be updated during the task operation. Drones are used as packet carriers, which transfer packets when flying from source to destination. This kind of routing is lightweight, however, the systems deployed with static routing are not fault tolerant or suitable for dynamically changing environment. In [13], a load-carry-and-deliver (LCAD) single-hop routing protocol is proposed to relay messages between two distant ground locations. Under LCAD, a drone will load data from the source ground location, carry it while flying towards the destination, and finally deliver it to the destination ground location. Because using single drone for packet transmissions can avoid interference and medium access contention, the proposed approach can provide high network throughput as well as high packet delivery latency with the increased distance between source and destination. Since static routing protocol is very sensitive to dynamically changing environment, the route planning problem is of great importance to drones. In [14], an optimal

fight path planning mechanism is proposed to determine the optimal flight path between neighboring acquisition points based on the obtained sensory data from data sensing points in wide IoT sensor networks. By using the proposed joint genetic algorithm and ant colony optimization from possible drone flight paths, an optimal flight path can be selected.

B. PROACTIVE ROUTING PROTOCOLS

The basic idea is that the 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. Thus, the routing paths can be selected to transmit data packets immediately without delay. The main advantage of proactive routing is that it contains the latest information of the routes. However, a large amount of control packets are needed to keep the routing tables up-to-date. In [15], a directional optimized link state routing protocol (DOLSR) is proposed to minimize the number of multi-point relays in FANET, where each drone is equipped with directional and omni-directional antennas. With the proposed protocol, the number of overhead packets as well as the end-to-end delay can be reduced. In [16], a mobility and load aware routing protocol is proposed for FANET, where relative speed and position between adjacent drones are considered to avoid selecting a high-speed drone as packet forwarder. Additionally, in order to avoid conflicts or interference when the packets are transmitted along the forwarding path, the packet load on each drone is considered to discover more stable routes without congestion. The [17] proposes a speed-aware predictive-optimized link state routing protocol (P-OLSR) by exploiting GPS information to aid routing operations. In the P-OLSR, the relative speed between two drones can be obtained based on GPS information, and is taken into account as a factor in the calculation of the expected transmission count metrics. Through the field experiments, the P-OLSR can follow rapid topology changes and provide a reliable multi-hop communication in situations where optimized link state routing (OLSR) protocol [9] fails.

C. REACTIVE ROUTING PROTOCOLS

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. Without periodic exchanges of control messages, reactive routing protocol can effectively reduce the communication overhead, but introduces high end-to-end delay. Dynamic source routing (DSR) [8] is a classic reactive routing protocol for multi-hop wireless mesh networks. In DSR, a source node floods a route request packet throughout the network. When the route request packet reaches the destination, the destination replies a route reply packet to source node. In addition, each node can quickly learn the routes of other nodes by aggressively overhearing on-flying packets and caching the piggybacked route information in its routing table. A time-slotted on-demand routing [18] is another representative reactive routing protocol, which assigns dedicated time slots for packet transmission to avoid congestion and improve packet delivery ratio. The [19] improves the procedure of route selection in reactive-greedy-reactive (RGR) protocol [20] by adding a criterion based on route reliability or stability, where the drones with high link reliability are preferred for packet forwarding. In [21], a drone-assisted VANET routing protocol is proposed to support ad hoc routing between drones and VANET as well as between drones themselves. The proposed scheme consists of two phases: ground-to-air communication and air-to-air communication. First, drones are used to estimate the vehicular density within a given road segment by monitoring and exchanging Hello messages with vehicles on the ground and assist vehicles in selecting communication routes for routing their data packets. Second, through air-to-air communication, drones are also used to route data packets when communication on the ground is deemed poor or when the vehicular density is not enough to route data packets through vehicles.

D. HYBRID ROUTING PROTOCOLS

To overcome the control message overhead problem of proactive routing protocols and the high end-to-end delay of reactive routing protocols, the hybrid routing protocol that is a combination of proactive and reactive routing is introduced. The [22] proposes a zone routing protocol (ZRP), which is a hybrid routing framework suitable for a wide variety of mobile ad-hoc networks, especially those with large network spans and diverse mobility patterns. Each node proactively maintains routes within a local region, also referred to as the routing zone. Knowledge of the routing zone topology is leveraged by the ZRP to improve the efficiency of a globally reactive route query/reply mechanism. A mobility prediction clustering algorithm (MPCA) relying on the attributes of drone is proposed to solve the problem of frequent cluster updates due to high-speed drones with the prediction of the network topology updates in [23]. The MPCA predicts the mobility structures of drones with the help of the dictionary Trie structure prediction algorithm and link expiration time mobility mode. In [24], a routing protocol named rapid-reestablish temporally ordered routing algorithm (RTORA) is proposed for FANET, where a reduced-overhead mechanism is adopted to overcome adverse effects caused by link reversal failure. In the reduced-overhead mechanism, a large amount of useless control packets from flooding are prevented.

E. OTHER APPROACHES

In [25], adaptive hybrid communication protocols including a position-prediction-based directional MAC protocol (PPMAC) and a self-learning routing protocol based on reinforcement learning (RLSRP) are proposed in FANETs. The PPMAC combines the directional antennas and position prediction in the MAC layer to overcome the directional deafness problem. The RLSRP allows updating the local routing policies with the position information of drones and a reward function defined based on the global network utility, while avoiding the necessity for other global knowledge of the networks. The [26] proposes a motion-driven packet forwarding

algorithm that applies delay-tolerant networking in case of disconnections, where near future drones' positions can be predicted by taking advantage of location and motion sensors provided by drones and a realistic mobility model. The [27] proposes an aerial network management protocol built on top of a software defined networking (SDN) architecture to address the needs of efficient and robust end-to-end data relaying in FANET, where each drone becomes a SDN switch that performs under directives sent by a centralized controller. The [28] proposes a predictive routing protocol based on three-dimensional estimation with a fast update mechanism for the flying path in FANET, where prediction mechanism is employed to determine the drone location and its trajectory to enhance the efficiency of the routing protocol.

In summary, various routing protocols and communication mechanisms have been well studied in FANETs and similar environments. Through analysis and comparison, it is found that each protocol has its own definite strengths and weaknesses, and suitable for specific situation. Most prior approaches focus on the shortest path, the freshest path, the minimum-cost path, the path with the best link quality, or mobility prediction. However, little attention has been paid for multipath routing with jamming-resilient capability in FANETs.

III. THE PROPOSED JAMMING-RESILIENT MULTIPATH ROUTING PROTOCOL

In this section, we first introduce the system model, and present link quality scheme, traffic load scheme, and spatial distance scheme, respectively. Then, we propose a jamming-resilient multipath routing protocol, also called *JarmRout*, to provide efficient and reliable communication and data transmission, and improve network resiliency in the presence of malicious jammers in FANETs.

A. SYSTEM MODEL

In this paper, we consider a set of drones (later nodes) that freely move in a FANET, where each node is identified by its node address. Each node is equipped with a global positioning system (GPS), inertial measurement units (IMUs), and digital map to obtain its current geographical position and mobility information [26]. Most of drone-based services and applications that use drones like small quad-copters do not fly at high altitudes [29], therefore, we assume that all drones have the same constant and low altitude during the flight. Thus, the mobility model and spatial distance scheme are designed based on two-dimensional (2D) space. For example, the 2D position coordinate and velocity vector of the i^{th} node, n_i, are denoted by $\{x_i,y_i\}$ and $\{v_x^i,v_y^i\}$, respectively. An extension to three-dimensional (3D) is possible, but it requires additional experiments. We also assume that nodes have no energy restrictions since they are equipped with rechargeable batteries which can be recharged from recharging stations or environmental energy resources (e.g., wireless power transfer, solar energy, etc.) [30], [31]. In addition, IEEE 802.11p wireless interface with a large transmission range (i.e., 300 meters) are assumed to be used by each node.

B. LINK QUALITY SCHEME

To estimate point-to-point link quality, most of prior studies typically employ one of the following four metrics: received signal strength indication (RSSI), signal-to-interference-plus-noise ratio (SINR), packet delivery ratio (PDR), and bit error rate (BER) [32]. Compared to the other three metrics, RSSI provides a quick and accurate estimate of whether a link is of very good quality [33]. The [34] has proved that higher RSSI values result in better packet delivery ratio, and as long as radio transceiver (e.g., DSRC compatible radio built upon the Atheros AR5000 chipset) maintains RSSI value above -55 dBm, the packet delivery ratio is almost a 100%. In addition, RSSI is shown very stable (standard deviation less than 1 dBm) over a short time period (e.g., 2 second), thereby a single RSSI reading is sufficient to determine if the link is stable or not [35]. Thus, the link quality can be estimated by using the statistical information of RSSI.

In this paper, we propose a function based on Chebyshev inequality [36], [37] to estimate the link quality. In probability theory, Chebyshev inequality guarantees that in any data sample or probability distribution, the strictly positive expectation $E(X)$ and the variance $var(X)$ have the following inequality with the discrete variable X:

$$P\{|X - E(X)| < \varepsilon\} \geq 1 - \frac{var(X)}{\varepsilon^2}. \tag{1}$$

When variance $var(X)$ tends to be zero, it reflects that the value of random variable X are always close to or equal to its expected value. In other words, a random variable X is relatively stable. By definition, we can obtain

$$var(X) = E(X^2) - E(X)^2. \tag{2}$$

and

$$E(X) = \sum_i \frac{X_i}{n}. \tag{3}$$

Thus, $var(X)$ can be represented as

$$var(X) = \Big(\sum_i \frac{X_i^2}{n}\Big) - \Big(\sum_i \frac{X_i}{n}\Big)^2. \tag{4}$$

Most radio transceivers contain an RSSI register, which provides the signal strength of the received packet [33]. Thus, each node can obtain the RSSI information when it receives the packet from neighbor node. Here, we use the RSSI to replace the variable X in Eq. 4. If the value of RSSI is very close to the expected value (e.g., -55 dBm), then it can be considered that the link between two nodes is stable. Finally, the link quality between two nodes (e.g., n_i and n_j), $LQ_{i,j}$, can be represented as

$$LQ_{i,j} = \Big(\sum_{x=1}^{N_{rssi}} \frac{R_x^2}{N_{rssi}}\Big) - \Big(\sum_{x=1}^{N_{rssi}} \frac{R_x}{N_{rssi}}\Big)^2. \tag{5}$$

Here, N_{rssi} is the total number of RSSI samples and R_x is the value of RSSI of the *x-th* sample.

For example, node n_a, n_b, and n_c are the neighbor nodes of n_i. As shown in Table 1, R_x, R_{x+1}, R_{x+2}, and R_{x+3} are the corresponding RSSI values of the most recently received packets from n_a, n_b, and n_c. Thus, n_i can calculate the link qualities according to Eq. 5, and then choose the neighbor node that provides the most stable link, where LQ is the minimum. Among three neighbor nodes, $LQ_{i,a}$ is the minimum, so the link between n_i and n_a is the most stable one. If there are two nodes with the same value of LQ, the node that has the closest value of the last packet to the expected RSSI value (e.g., −55 dBm) will be considered to provide a more stable link. For example, between n_b and n_c, n_c is assumed to have a more stable link with n_i.

TABLE 1. Calculation of link qualities between node n_i and its neighbor node n_a, n_b, and n_c.

Node	R_x	R_{x+1}	R_{x+2}	R_{x+3}	LQ	Ranking
n_a	-55	-56	-58	-62	7.1875	1^{st}
n_b	-65	-70	-68	-63	7.25	3^{rd}
n_c	-60	-65	-63	-58	7.25	2^{nd}

C. TRAFFIC LOAD SCHEME

The IEEE 802.11 Medium Access Control (MAC) protocol with request-to-send (RTS)/clear-to-send (CTS) exchange is used to reduce frame collisions due to the hidden terminal problem and the exposed terminal problem. The protocol not only uses physical carrier sensing, it also introduces the novel concept of virtual carrier sensing, which is implemented in the form of a Network Allocation Vector (NAV). The NAV contains a time value that represents the duration upto which the wireless medium is expected to be busy because of transmissions by other nodes. When the node receives RTS or CTS packet piggybacked with the duration information for the remainder of the messages, it will set its own NAV and defer any possible transmission to a later time. The NAV also indicates the busyness of the medium and can be considered as a useful metrics for contention and traffic load situation around the node [38]. For example, a node with three active neighbor nodes will get less chance to access the shared wireless medium than the node with only one active neighbor node. Thus, the average busy proportion of wireless channel can be used to represent the traffic load around a node in a short term. In order to mitigate the effect of traffic bursts, the average busy portion of wireless medium at node n_i, T_i^{busy}, is updated by the low-pass filer with a filer gain constant α,

$$T_i^{busy} = \alpha \cdot T_i^{busy} + (1-\alpha) \cdot NAV_i^{k-1}. \quad (6)$$

Here, NAV_i^{k-1} is the measurement from the most recent medium access.

Moreover, according to IEEE 802.11 mechanism, when the MAC layer cannot transmit the packets timely, the packets will be stored in the buffer. A node with more traffic load passing through usually has more waiting packets stored in its buffer. Thus, the average number of waiting packets stored in the buffer at node n_i, Q_i^{buf}, can indicate the traffic load around n_i in a long term, which can be represented as

$$Q_i^{buf} = \beta \cdot Q_i^{buf} + (1-\beta) \cdot B_i^{k-1}. \quad (7)$$

Here, B_i^{k-1} is the most recently measured number of waiting packets stored in the buffer. In this paper, α and β are the system parameters and can be configured depending on whether the current traffic condition has more influence on the calculation of the average value.

Finally, the overall traffic load of node n_i, TL_i, can be represented as

$$TL_i = \gamma \cdot T_i^{busy} + (1-\gamma) \cdot Q_i^{buf} + (T_i^{busy} + Q_i^{buf}) \cdot \varphi, \quad (8)$$

where γ is a filter gain constant and φ is an adjustment factor and $(T_i^{busy} + Q_i^{buf}) \cdot \varphi$ is added to consider the medium access and packet queue delay.

TABLE 2. Calculation of traffic load at node n_a, n_b, and n_c.

Node	T^{busy}	Q^{buf}	TL	Ranking
n_a	7.3619 msec	15	11.6282	3^{rd}
n_b	6.8976 msec	8	7.7468	1^{st}
n_c	7.1257 msec	10	8.9054	2^{nd}

For example, as shown in Table 2, T^{busy} and Q^{buf} is the average busy portion of wireless medium and the average number of waiting packets stored in the buffer for node n_a, n_b, and n_c, respectively. According to Eq. 8, the traffic load can be calculated for each node and n_b is considered to have the lightest traffic load, where TL is the minimum. Note that the traffic load of n_a, n_b, and n_c are 11.6282, 7.7468, and 8.9054, respectively.

D. SPATIAL DISTANCE SCHEME

Multipath routing is proposed to reduce end-to-end delay, perform load balancing, and consequently improve network throughput in ad hoc networks [39]. However, one of significant challenges to effective use of multipath routing protocol in this environment is the effects of route coupling. Route coupling occurs when two routes are located physically close enough to interfere with each other during data transmissions with the shared wireless medium. As a result, the nodes in multiple routes are constantly contending for access to the medium and can end up performing worse than a single path protocol. On the other side, solely utilizing multipath between source and destination without considering spatial separation distance between multipath is not enough. This is because FANET may suffer from malicious attacks that blanket out a mission-critical area by intentional jamming and disruption, where several drones located along multipath may be affected by the jamming signals concurrently, resulting in the fully disconnection of multipath between source and destination. Thus, leveraging maximally spatial node-disjoint

multipath between source and destination not only can help avoid radio collisions between alternate paths and reduce the effects of route coupling, but also avoid isolated and localized failures, or even intentional jamming and disruption.

In this paper, we propose a new spatial distance scheme along with a new distance metrics to measure the physical distance between multiple paths based on [40]. First, we define the distance of a node n_i on path p to path q as the minimum distance from node n_i on path p to all nodes of path q, which can be expressed as

$$dist^{node}_{path}(i, q) = \min_{j \in q}\{dist(i, j)\}. \tag{9}$$

Here, $dist(i, j)$ is the spatial distance between node n_i and n_j, and can be represented as

$$dist(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \tag{10}$$

where $\{x_i,y_i\}$ and $\{x_j,y_j\}$ is the two-dimensional position coordinate of node n_i and n_j, respectively. The distance from path p to path q is defined as the arithmetic mean of the distance of nodes of p to q, and can be expressed as

$$dist^{path}_{path}(p, q) = \frac{\sum_{i \in p} dist^{node}_{path}(i, q)}{size(p)}, \tag{11}$$

where $size(p)$ is the number of nodes excluding the source and destination nodes located along the path p. Since Eq. 11 is not symmetric, we use Eq. 12 to calculate the spatial distance between path p and q,

$$\Upsilon(p, q) = \frac{dist^{path}_{path}(p, q) + dist^{path}_{path}(q, p)}{2}. \tag{12}$$

For example in Fig. 1, three paths, p, q, and r, are available between source n_S and destination n_D, where the position coordinate of each node is shown in the pair of parentheses. In this example, we consider a horizontal two-dimensional space, i.e., in the X-Y plane. As shown in Table 3, the spatial distance between any two paths can be calculated according to Eq. 12, and path p and r has the largest spatial distance, where Υ is the largest, $\Upsilon(p, r) = 10$.

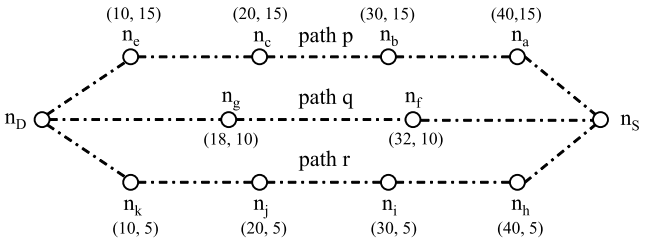


FIGURE 1. An example of computing the spatial distance between paths, where three paths, p, q, and r, are available between source n_S and destination n_D, and the position coordinate is shown in the pair of parentheses.

TABLE 3. Calculation of spatial distance among path p, q, and r.

Path	Spatial Distance
$\Upsilon(p, r)$	10.0
$\Upsilon(p, q)$	6.4
$\Upsilon(q, r)$	6.4

E. THE PROPOSED JARMROUT ROUTING PROTOCOL

First, when the source node has data packets to send, it first searches its routing table for the route to the destination node. If the route is not available, the source node initiates the route discovery procedure by broadcasting a route request (RREQ) packet. The RREQ packet contains source node ID (S_{id}), destination node ID (D_{id}), packet sequence number (pkt_{seq}), the number of hops to source node (C_{hop}), source route record (H_{route}), list of position coordinates of nodes in source route record (P_{coord}), worst link quality along the route (L_{qt}), and maximum traffic load along the route (T_{ld}). Here, the format of RREQ packet is shown in Fig. 2. Any intermediate node located between source and destination nodes receives a RREQ packet for the first time, it caches the packet sequence number pkt_{seq} and the number of hops to source node C_{hop}. In addition, it calculates the link quality between itself and RREQ packet sender and its current traffic load according to Eq. 5 and 8, respectively. If the calculated link quality is larger than the piggybacked link quality L_{qt}, or the calculated traffic load is larger than the piggybacked traffic load T_{ld}, it replaces the L_{qt} or T_{ld} with the newly calculated value, appends its node ID in the source route record H_{route}, adds its position coordinate in the list of position coordinates P_{coord}, increases the hop count C_{hop} by one, and rebroadcasts the RREQ packet. Otherwise, it just appends its node ID in H_{route}, adds its position coordinate in P_{coord}, increases C_{hop} by one, and rebroadcasts the RREQ packet. When a node receives duplicated RREQ packet, it first compares the piggybacked C_{hop} in the received RREQ packet with the previously cached hop count information. If the piggybacked C_{hop} is larger than the previously cached hop count, the node drops the RREQ packet directly. Otherwise, the node calculates the link quality between itself and RREQ packet sender and its current traffic load respectively, updates the L_{qt} if the newly calculated link quality is larger than L_{qt}, updates the T_{ld} if the newly calculated traffic load is larger than T_{ld}, appends its node ID in H_{route}, adds its position coordinate in P_{coord}, increases C_{hop} by one, and rebroadcasts the RREQ packet. In reactive routing protocols, i.e., DSR [8], each node can quickly learn the routes of other nodes by aggressively overhearing on-flying packets and caching the piggybacked route information in its routing table. This is because overhearing does help in improving the routing performance [41]. However, in the JarmRout, intermediate nodes are not allowed to send the route reply (RREP) packet back to the source node even

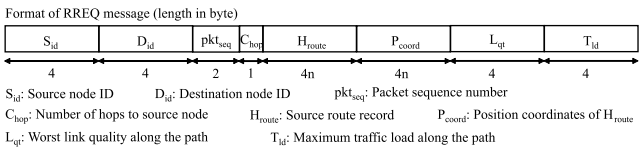


FIGURE 2. The format of *RREQ* message. Here, the length is shown in byte.

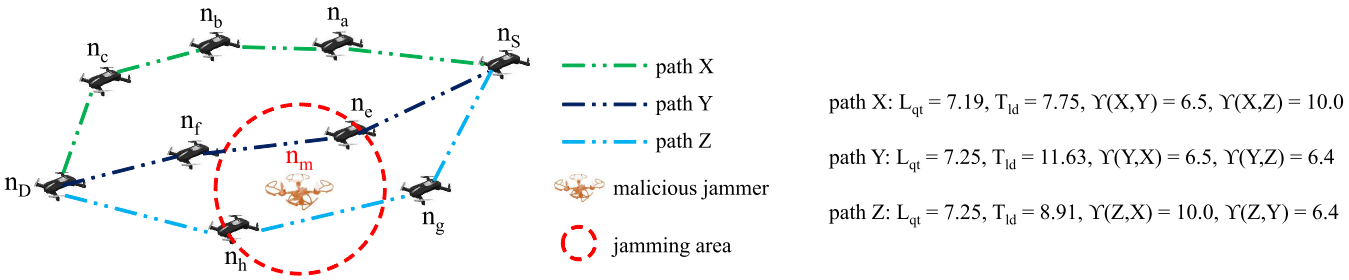


FIGURE 3. A snapshot of network, where three node-disjoint paths, *X*, *Y*, and *Z*, are available between source node n_S and destination node n_D. n_m is the malicious node and continuously broadcasts the jamming signals to disrupt the on-going communications of node n_e and n_h.

when they have route information to the destination node. This is because if intermediate nodes reply RREP packets from cache, it is difficult to establish maximally spatial node-disjoint multiple paths because not enough RREQ packets will reach the destination node and the destination node will not know the information of the route that is formed from the cache of intermediate nodes [10].

Second, the destination node maintains a route table to store the information of received multiple node-disjoint paths. When the destination node receives the first RREQ packet, it records the piggybacked pkt_{seq}, H_{route}, P_{coord}, L_{qt}, and T_{ld} in the route table. The destination node then waits for a certain time period (t_{wait}) to receive more RREQ packets and learn all possible routes. If the destination node receives a duplicated RREQ packet, it compares the H_{route} in the received RREQ packet to all of the existing node-disjoint paths in the route table so far. If there is not a common node (except source and destination nodes) between the H_{route} in the received RREQ packet and any existing node-disjoint path in the route table, it records the pkt_{seq}, H_{route}, P_{coord}, L_{qt}, and T_{ld} of the received RREQ packet in the route table. Otherwise, the destination node discards the received RREQ packet because it does not meet the requirement of node-disjoint path. Here, the path piggybacked in the first received RREQ packet, also called the shortest delay path, has a higher priority to be stored in the route table. The rationale behind this design is that the shortest delay path reflects the less number of hops between source and destination nodes, as well as the better link quality and light traffic load along the forwarding path. In addition, the shortest delay path also can minimize the latency of route discovery process, which is a practical need in reactive routing protocol.

When t_{wait} expires, the destination node selects two node-disjoint paths based on the metrics of link quality, traffic load, and spatial distance. More than two node-disjoint paths can be chosen, but we limit the number of paths to two in this paper. In the route table, every pair of candidate paths is assigned a Pair Priority (P_{prio}), which is the comprehensive judgment factor of two node-disjoint paths in terms of link quality, traffic load, and spatial distance. The Pair Priority $P^{p,q}_{prio}$ of two node-disjoint paths p and q can be represented as

$$P^{p,q}_{prio} = \varpi \cdot (L^p_{qt} + L^q_{qt}) + \varrho \cdot (T^p_{ld} + T^q_{ld}) + \varsigma \cdot e^{\frac{1}{\Upsilon(p,q)}}, \quad (13)$$

where L^p_{qt} and T^p_{ld}, and L^q_{qt} and T^q_{ld} are the worst link quality and largest traffic load along the path p and q, respectively, and $\Upsilon(p, q)$ is the spatial distance between path p and q. Here, ϖ, ϱ, and ς are weighting factors of each metrics for the calculation of Pair Priority, and $(\varpi + \varrho + \varsigma) = 1$. Based on the calculated P_{prio}, the pair of paths with the minimum P_{prio} will be chosen as the two node-disjoint paths to send the data packets. After this process, the destination node replies two RREP packets piggybacked with reversed route information to source node. Upon receiving the RREP packets, the source node caches the complete routes piggybacked in the RREP packets in its routing table, and then sends data packets along two node-disjoint paths.

Third, certain link of path can be disconnected frequently because of mobility of nodes, traffic congestion, package collisions, or even intentional jamming and interruption. In the JarmRout, if a node continuously fails to deliver the data packet to the next-hop node along the forwarding path, i.e., not overhearing implicit acknowledgment or receiving explicit acknowledgment [42], it considers the link to be disconnected and sends a route error (RERR) packet piggybacked with disconnected link to source node. Upon receiving the RERR packet, the source node removes the entire path that contains the broken link in its routing table. If only one of the two node-disjoint paths of the session is invalid, the source node uses the remaining valid path to deliver data packets. If both paths are invalid, the source node initiates the route discovery procedure again to find a new pair of node-disjoint paths to send data packets.

For example, as shown in Fig. 3, three node-disjoint paths, *X*, *Y*, and *Z*, are available between source node n_S and destination node n_D. However, a malicious node n_m is able to continuously broadcast jamming signals to interfere with the communication of node n_e and n_h in path *Y* and *Z*, and interrupt the entire communication of path *Y* and *Z* between n_S and n_D. If path *Y* and *Z* were selected as two node-disjoint paths for communication, this undesirable situation nullifies the benefit of multipath routing. To mitigate this situation, the JarmRout is to choose two node-disjoint paths that are physically far away. According to Eq. 13, the $P^{X,Y}_{prio}$, $P^{X,Z}_{prio}$, and $P^{Y,Z}_{prio}$ are 7.4638, 6.8831, and 7.7095, respectively. Thus, the pair of path *X* and *Z* has the highest priority to be chosen as two node-disjoint paths, where $P^{X,Z}_{prio}$ has the smallest value.

Notations:

- $RREQ[S_{id}, D_{id}, pkt_{seq}, C_{hop}, H_{route}, P_{coord}, L_{qt}, T_{ld}]$, $RREP$, $cache\_seq_i$, $coord_i$, $cache\_hop_i$, and RT_i: A *RREQ* packet, a *RREP* packet, cached packet sequence number of *RREQ* packet at node n_i, position coordinate of node n_i, cached hop count of *RREQ* packet at node n_i, and routing table at node n_i.
- S_{id}, D_{id}, pkt_{seq}, C_{hop}, H_{route}, P_{coord}, L_{qt}, T_{ld}, t_{wait}, P_{prio}: Defined before.

Event-driven JarmRout Algorithm:

◇ When a source node n_S has data packets to send to destination node n_D:
  if route $\notin RT_S$
    Broadcast *RREQ* packet;
  else
    Send data packets along cached route;
◇ When an intermediate node n_i receives a *RREQ* packet:
  if $RREQ.pkt_{seq} \notin cache\_seq_i$ /* Receive $RREQ$ first time. */
    Cache pkt_{seq} and C_{hot} in $cache\_seq_i$ and $cache\_hop_i$;
    Calculate L_{qt} and T_{ld} according to Eq. 5 and 8;
    if $L_{qt} > RREQ.L_{qt}$
      $RREQ.L_{qt} = L_{qt}$;
    if $T_{ld} > RREQ.T_{ld}$.
      $RREQ.T_{ld} = T_{ld}$;
    Append n_i in $RREQ.H_{route}$;
    Add $coord_i$ in $RREQ.P_{coord}$;
    Increase $RREQ.C_{hop}$ by one;
    Rebroadcast $RREQ$ packet;
  else /* Receive duplicated $RREQ$. */
    if $RREQ.C_{hop} > cache\_hop_i[pkt_{seq}]$
      Discard $RREQ$ packet;
    else
      Calculate L_{qt} and T_{ld} according to Eq. 5 and 8;
      if $L_{qt} > RREQ.L_{qt}$
        $RREQ.L_{qt} = L_{qt}$;
      if $T_{ld} > RREQ.T_{ld}$
        $RREQ.T_{ld} = T_{ld}$;
      Append n_i in $RREQ.H_{route}$;
      Add $coord_i$ in $RREQ.P_{coord}$;
      Increase $RREQ.C_{hop}$ by one;
      Rebroadcast $RREQ$ packet;
◇ When a destination node n_D receives $RREQ$ packet;
  if $RREQ.pkt_{seq} \notin RT_D$ /* Receive $RREQ.pkt_{seq}$ first time. */
    Record $RREQ[pkt_{seq}, C_{hop}, H_{route}, P_{coord}, L_{qt}, T_{ld}]$ in RT_D;
  else /* Check whether common node exists along multiple routes. */
    $flag_{com}$ = **false**;
    for $n_k \in RREQ.H_{route}$
      if $n_k \in RT_D[pkt_{seq}].H_{route}$ /* Common node exists. */
        $flag_{com}$ = **true**;
        break;
    if $flag_{com}$ is **false** /* No common node. */
      Record $RREQ[pkt_{seq}, C_{hop}, H_{route}, P_{coord}, L_{qt}, T_{ld}]$ in RT_D;
    else
      Discard $RREQ$ packet;
◇ When a timer t_{wait} expires at destination node n_D:
  Calculate $P_{prio}^{X,Y}$ for any two paths, i.e., X and Y, in RT_D;
  Select the pair of path X and Y with smallest P_{prio} as forwarding paths;
  Reply *RREP* packets to n_S through path X and Y, respectively;
◇ When a source node n_S receives *RREP* packets:
  Cache the complete route piggybacked in the RREP packets in RT_S;
  Send data packets to destination node along multiple node-disjoint paths;
◇ When an intermediate node n_i detects a link failure:
  Send *RERR* packet back to source node;

FIGURE 4. The pseudocode of the JarmRout routing protocol.

Here, ϖ, ϱ, and ς are set to 0.2, 0.2, and 0.6, respectively, where spatial distance has more weights than that of link quality and traffic load. In summary, the JarmRout aims to deter selection of physically closer paths to avoid them being disrupted by a single jamming source, in return, the end-to-end outage rate as well as network resiliency and performance can be improved in the presence of malicious jammers. Major operations of the JarmRout are summarized in Fig. 4.

IV. ANALYSIS OF THE PROPOSED JARMROUT ROUTING PROTOCOL

We further analyze the proposed JarmRout routing protocol in terms of *RREP* packet reception rate of source node, which is denoted by R_{rrep}. When source node receives a RREP packet corresponding to previously issued RREQ packet, it successfully finds one path to send data packets towards destination node. Suppose a network size is $X \times Y$ (m^2), where N nodes are uniformly distributed, and a packet loss rate is ζ due to bad channel quality or disconnected link. Let R_{dest}^{rreq} and R_{src}^{rrep} be the probability of destination node receiving RREQ packet and source node receiving RREP packet, respectively. Then R_{rrep} is expressed as,

$$R_{rrep} = R_{dest}^{rreq} \cdot R_{src}^{rrep}. \tag{14}$$

In this paper, the average number of hops between the source and destination nodes, c_{hop}, is approximated according to [41] and it is expressed as,

$$c_{hop} \approx \frac{d}{\ell} \approx \frac{\sqrt{X^2+Y^2}}{2\ell} \approx \frac{(2\xi+1)\cdot\sqrt{X^2+Y^2}}{4\xi R}. \tag{15}$$

Here, ℓ and d are the average progress of each hop and average distance between the source and destination nodes, respectively. R is the communication range of each node. ξ is the average number of nodes located within R and it is expressed as,

$$\xi = \frac{N}{X * Y} \cdot \pi R^2. \tag{16}$$

First, R_{dest}^{rreq} is expressed as,

$$R_{dest}^{rreq} = (1-\zeta)^{c_{hop}}. \tag{17}$$

which is the probability that a RREQ packet is relayed through c_{hop} number of hops and reaches destination node. Second, R_{src}^{rrep} is expressed as,

$$R_{src}^{rrep} = R_{dest}^{rreq} \cdot R_{sr}, \tag{18}$$

where

$$R_{sr} = (1-\zeta)^{c_{hop}}. \tag{19}$$

Here, R_{sr} is the probability that a RREP packet is forwarded back to source node through c_{hop} number of hops. Finally, R_{rrep} is expressed as,

$$\begin{aligned} R_{rrep} &= R_{dest}^{rreq} \cdot R_{src}^{rrep} \\ &= \left((1-\zeta)^{c_{hop}}\right)^3. \end{aligned} \tag{20}$$

In Fig. 5, we show a numerical result of the number of hops between source and destination nodes and *RREP* packet reception rate of source node against the number of nodes and channel error rate in the network. Here, 50 to 100 nodes are uniformly distributed in a 1000 × 1000 (m^2) network area, where the communication range of each node is 300 (meter) and channel error rate is between 5% and 10%. According to Subfig. 5(a), the number of hops between source and destination nodes is not sensitive to the number of nodes

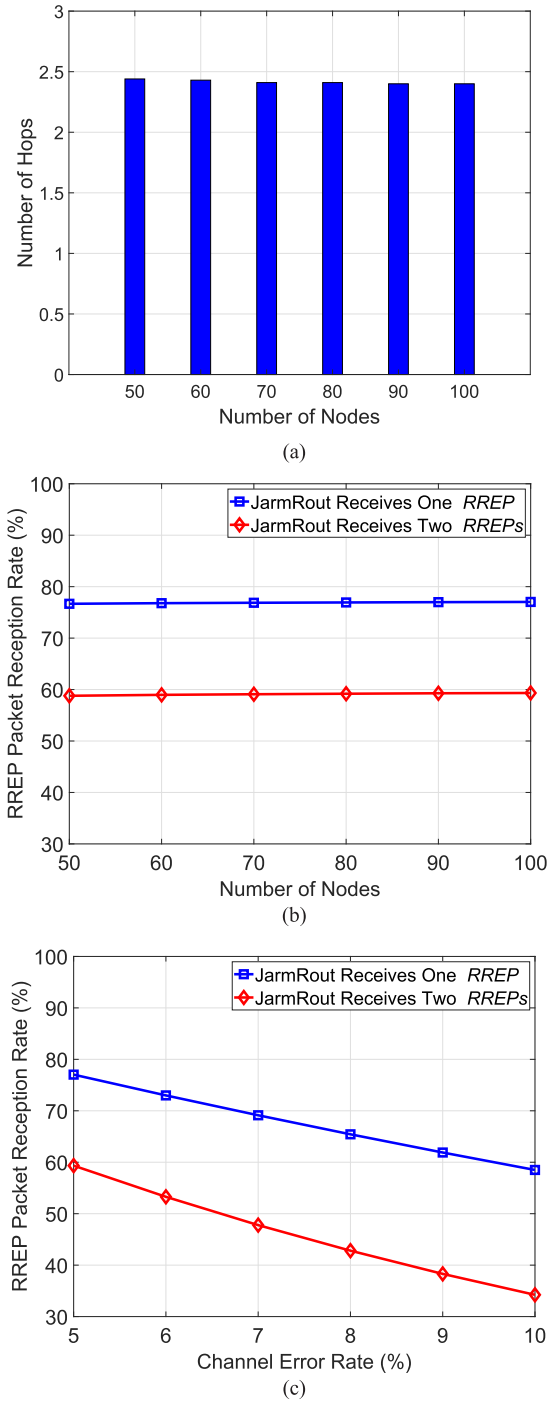


FIGURE 5. The number of hops between source and destination and the *RREP* packet reception rate against the number of nodes and channel error rate.

in the network, and remains steady around 2.415 hops. As shown in Subfig. 5(b), the *RREP* packet reception rate is not changing too much as the number of nodes increases, because the average number of hops between source node and destination node does not change significantly as the node density increases. However, the probability of receiving two RREP packets are lower than that of receiving one RREP packet, because either one of RREP packets could get lost during the transmission due to bad channel quality or disconnected link. In Subfig. 5(c), as the channel error rate increases, the *RREP* packet reception rate significantly decreases. This is because RREQ or RREP packet has more chances to get lost during the transmission with larger channel error rate.

V. PERFORMANCE EVALUATION

A. SIMULATION TESTBED

We conduct extensive simulation experiments using the OMNeT++ [11] for performance evaluation and analysis. A 1000 $\times$ 1000 (m^2) square network area is considered, where 50 to 100 nodes are uniformly distributed. Nodes are equipped with IEEE 802.11p radio transceiver. The communication range of each node is 300 (meter) and the two-way ground propagation channel is assumed with a data rate of 2 Mbps. The random waypoint mobility model [43] is deployed in the network, where each node travels toward a randomly selected destination in the network with a constant speed of 30 meter/sec and a zero pause time. The source node generates a constant bit rate (CBR) traffic at the packet rate of 1.0 to 3.0 packet/sec and each packet size is 512 Bytes. The total simulation time is 3000 seconds, and each simulation scenario is repeated 10 times with different randomly generated seeds to obtain steady state performance metrics. The simulation parameters are summarized in Table 4. In this paper, we measure the performance in terms of packet delivery ratio, packet delivery latency, end-to-end communication outage rate, and energy consumption by changing key simulation parameters, including packet rate, number of nodes, and number of malicious jammers.

- Packet Delivery Ratio (PDR): PDR is computed as the ratio of total number of received data packets to total number of generated data packets, showing the performance resiliency of the proposed JarmRout in the adversary scenarios.
- Packet Delivery Latency (PDL): PDL is the elapsed time from when source node initiates route discovery procedure to when the destination node receives the first data packet, indicating the packet transmission delay in the presence of malicious jammers.
- End-to-End Communication Outage Rate (COR): COR is the total number of built paths divided by the total number of affected paths due to jamming signals dur-

TABLE 4. Simulation parameters.

Parameter	Value
Network area	1000$\times$1000 m^2
Number of nodes	50 to 100
Moving speed	30 meter/sec
Mobility model	Random waypoint
Communication range	300 meter
Number of malicious jammers	1 to 6
Channel error rate	10%
Radio data rate	2 Mbps
Packet injection rate	1.0 to 3.0 packet/sec
Packet size	512 Bytes
Simulation time	3000 seconds

ing the entire simulation, showing the improvement of network resiliency of the proposed JarmRout.

- Energy Consumption (EC): EC is measured based on the number of forwarded and received control packets to build routing path [44], and it is used to show that the proposed JarmRout does not introduce extra communication overhead.

For performance comparison, we revisit three representative routing protocols, which are dynamic source routing (DSR) [8], optimized link state routing (OLSR) [9], and split multipath routing (SMR) [10], and modify them to work in FANET. The major operations of three benchmark routing protocols are briefly described below:

- Dynamic Source Routing (DSR): When a source node generates a data packet to send, it first searches its routing table for the route to a destination node. If the route is not available, the source node initiates the route discovery procedure by broadcasting a RREQ packet. Any intermediate node located between the source and destination nodes rebroadcasts the received RREQ by adding its node address in the packet header, if it does not have the route to destination node. When the destination node receives the RREQ, it replies a RREP packet back to source node. Upon receiving the RREP, the source node sends a data packet using the complete route piggybacked in the packet header.
- Optimized Link State Routing (OLSR): Each node periodically constructs and maintains the set of neighbor nodes that can be reached in one-hop and two-hop. Based on this information, the dedicated multi-point relays (MPR) algorithm minimizes the number of active relays needed to cover all two-hop neighbor nodes. A node forwards a packet if and only if it has been elected as MPR by the sender node. In order to construct and maintain routing table, OLSR periodically transmits link state information over the MPR backbone. Upon convergence, an active route is created at each node to reach any destination node in the network.
- Split Multipath Routing (SMR): When a source node wants to send a data packet to a destination node for which a route is not available, it broadcasts a RREQ packet into the network. When receiving the first RREQ packet, the destination node considers the piggybacked route in the first received RREQ packet as the first available path, and replies a RREP packet to source node. After that, the destination node waits for a certain duration of time to receive more RREQ packets. Then, the destination node selects the route that has the least number of common nodes with the route that is already replied. Finally, the destination node sends another RREP packet to source node via the second selected route.

B. SIMULATION RESULTS AND ANALYSIS

We first measure the packet delivery ratio (PDR) by changing packet injection rate, number of nodes, and number of malicious jammers in Fig. 6. In Subfig. 6(a), as packet injection rate increases, the PDR of four schemes decrease because more data packets collide with constant jamming signals or get lost during transmission due to bad channel quality. The DSR, SMR, and JarmRout show higher PDR

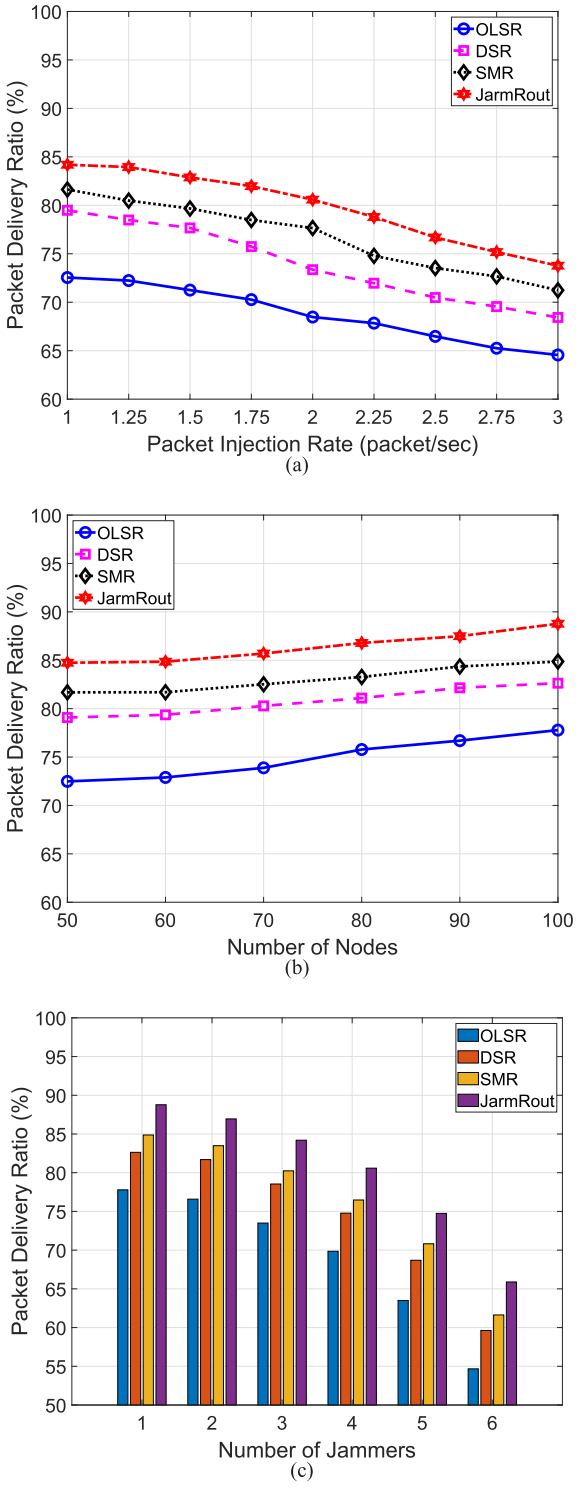


FIGURE 6. The performance of packet delivery ratio against packet injection rate, number of nodes, and number of jammers.

than that of OLSR because active route stored in the routing table of OLSR may not be available any more due to the frequent change of network topology when a source node has data packets to send, and data packets that are transmitted along the unavailable route cannot reach the destination node. The JarmRout shows the best performance because two maximally spatial node-disjoint paths are selected to send data packets. As the packet injection rate increases, more data packets are transmitted. If one of paths is disrupted by malicious jammers, data packets still can be transmitted along another path, and finally reach the destination node.

In Subfig. 6(b), the PDR of four schemes slightly increase as the number of nodes in the network increases. This is because each node has more neighbor nodes as node density increases, more available routes between source and destination nodes will be available to send data packets, and more data packets can be delivered to destination node. Both SMR and JarmRout show higher PDR than that of OLSR and DSR because data packets are transmitted along multiple paths, and a larger number of data packets can be received by destination node in the presence of malicious jammers. By considering maximally spatial distance, node-disjoint multiple paths have less chances to be affected within disruption radius or area concurrently, a larger number of data packets can be delivered by JarmRout, resulting in the highest PDR. As shown in Subfig. 6(c), the PDR significantly decreases as the number of malicious jammers increases. However, the JarmRout still outperforms OLSR, SMR and DSR as expected because the destination node selects multipath with high quality links, low traffic load, and maximally spatial separation distance to send data packets, more data packets can be delivered.

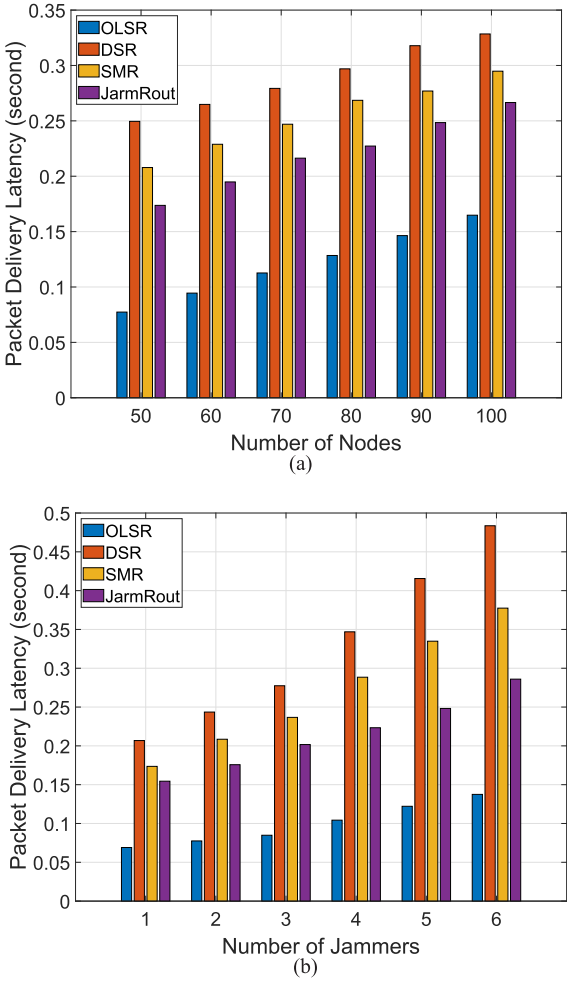


FIGURE 7. The performance of packet delivery latency against number of nodes and number of jammers.

Second, the packet delivery latency (PDL) is measured against the change of number of nodes and number of malicious jammers in Fig. 7. As shown in Subfig. 7(a), the OLSR achieves the lowest PDL compared to that of DSR, SMR, and JarmRout. This is because OLSR is a proactive routing protocol and the routing tables are updated and shared periodically, and the data packets can be transmitted immediately without delay with the stored routing path. As for DSR, SMR, and JarmRout, a route discovery procedure has to be initiated to find the routing path between source and destination nodes, as a result, a larger PDL is achieved compared to that of OLSR. The DSR shows a higher PDL than that of SMR and JarmRout because the destination node replies to the first received RREQ packet with a RREP packet to build the routing path between source and itself. However, this single routing path may be disrupted by the malicious jammer, the affected intermediate node cannot successfully deliver the data packet to the next-hop node, and then replies the RERR packet back to source node. Thus, route discovery procedure needs to be initiated again, which results in a higher PDL. The JarmRout shows a lower PDL than that of SMR. Since the SMR selects maximal node-disjoint paths rather than maximally spatial node-disjoint paths, the selected multipath has more chances to be affected by the malicious jamming signals concurrently, the route discovery process has to be repeated again and a longer latency is observed.

In Subfig. 7(b), the PDL of DSR, SMR, and JarmRout naturally increases as more malicious jammers exist in the network, while the PDL of OLSR is not very sensitive to the change of number of malicious jammers. In the OLSR, the routing paths are stored in each node in advance, and can be selected to transmit data packets without a long waiting time. However, data packets could get lost during the transmission because of packet collisions with jamming signals, thus, a slight increment of PDL is observed due to the retransmission of data packets. Among DSR, SMR, and JarmRout, the JarmRout still shows the best performance as the number of malicious jammers increases. This is because maximally spatial node-disjoint paths are selected to transmit data packets. If one path is disrupted, the source node still can use another active path to send data packets to destination node without initiating a new route discovery procedure. As a result, a lower PDL is achieved by JarmRout.

Third, end-to-end communication outage rate (COR) is observed with varying number of nodes and malicious

jammers in Fig. 8. In Subfig. 8(a), the overall COR of all schemes decreases as the number of nodes in the network increases. As the node density increases, each node has more adjacent nodes, thus, more potential routing paths are available between any two nodes. Even though the malicious jammers can disrupt a certain number of routes, however, more routing paths are still available and the COR decreases. The JarmRout still outperforms other three schemes, OLSR, DSR, and SMR, by providing the lowest COR as the number of nodes increases. This is because maximally spatial node-disjoint paths are chosen, and there are less chances that the radius of jamming signals cover both maximally spatially separated paths. Thus, the lowest COR is achieved by JarmRout. As shown in Subfig. 8(b), the COR is very sensitive to the number of malicious jammers. As the number of malicious jammers increases, the COR significant increases. As expected, the receiving and sending operations of a large number of intermediate nodes can be disrupted by jamming signals from malicious jammers, as a result, the entire end-to-end connections between source and destination nodes are affected, and finally the total number of affected routing paths increases. However, the JarmRout shows the lowest COR compared to that of OLSR, DSR, and SMR because two routing paths with maximally spatial distance are selected in each route discovery process, thus, the number of disrupted end-to-end connection decreases. In addition, as the number of malicious jammers increases, the COR difference between JarmRout and other three schemes increases.

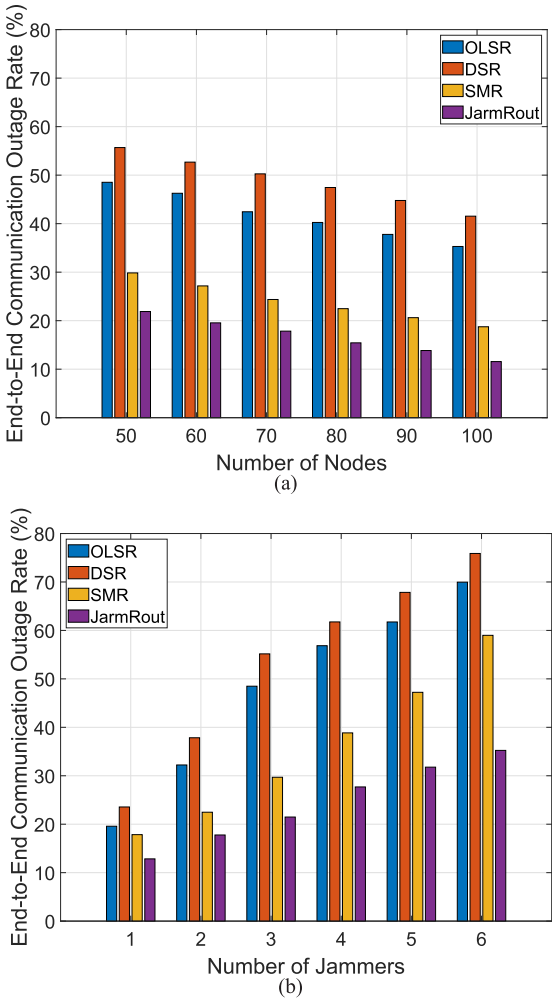


FIGURE 8. The performance of end-to-end communication outage rate against number of nodes and number of jammers.

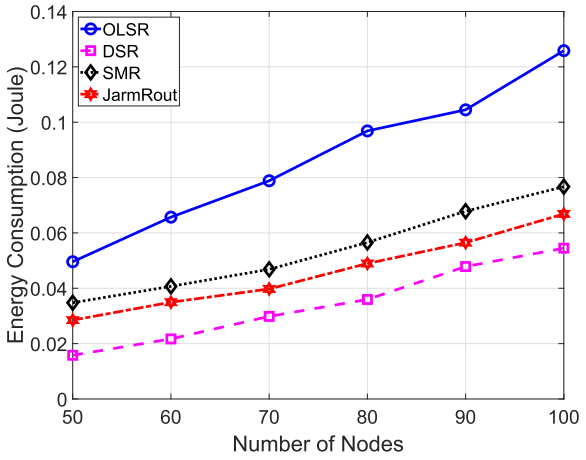


FIGURE 9. The performance of energy consumption against number of nodes.

Finally, we measure the energy consumption (EC) of building routing paths based on the number of forwarded and received control packets by varying number of nodes in Fig. 9. The OLSR shows the highest EC compared to that of other three schemes. In particular, as the number of nodes increases, the EC of OLSR increases quickly. In OLSR, each node periodically exchanges its routing information by broadcasting control packets to build the routing paths, thus, a large amount of control packets are broadcasted and received by each node, which introduces larger communication overhead. The lowest energy consumption is achieved by DSR because control packets (i.e., RREQ and RREP) are transmitted along a single path. The JarmRout does not bring extra communication overhead, and a slightly higher energy consumption compared to that of DSR is observed because control packets are transmitted along two paths.

VI. DISCUSSION

In this section, we first discuss the proposed JarmRout routing protocol in terms of its features, constraints, and possible enhancements. Then we investigate the immunity of the JarmRout to other three well-known attacks in FANETs.

A. FEATURES, CONSTRAINTS, AND POTENTIAL ENHANCEMENTS

We discuss the JarmRout by considering its features, constraints, and potential enhancements for improvement. The JarmRout is designed with three desirable features. First, each node utilizes the received signal strength indication (RSSI)

information to estimate the link quality, since each radio transceiver contains an RSSI register, which provides the signal strength of the received packet. Compared to other measurements, i.e., statistical information of the physical distance or relative movement between nodes, RSSI actually reflects the characteristics of the environment, such as rich signal reflection, blockages, route coupling, or even interference, and can accurately represent the link quality. Second, in order to avoid packet collisions, IEEE 802.11 CSMA/CA mechanism is used in FANETs. However, as the time interval of sending packets decreases, the MAC layer cannot transmit the packets timely because of the busyness of the wireless medium, and more packets will be cached in the buffer. Thus, a traffic load scheme is to assure a light load path by taking account of MAC layer channel contention information and the number of packets cached in the buffer. Third, the FANET may face malicious attacks that blanket out a mission-critical area by intentional jamming and disruption. Several nodes may be affected within the disrupted area concurrently, which may fully disconnect the routing path between source and destination nodes. Thus, maximally spatial node-disjoint multipath becomes a practical need to reduce end-to-end communication outage rate and improve network resiliency in these circumstances.

In the JarmRout, there are a few constraints that need to be further investigated. First, the JarmRout selects two node-disjoint paths to send data packets, where the two paths do not have common node except source and destination nodes. However, if the destination node only has one adjacent node, for example n_f in Fig. 3, it would be hard to find multiple paths. This is because the destination node will discard the received RREQ packet that does not meet the requirement of node-disjoint path, and only one path can be established in this case. Second, bidirectional links are implicitly assumed in this paper, and the proposed JarmRout may be incapable of functioning properly over unidirectional links. For example in Fig. 3, suppose that node n_b is within the communication range of node n_a, however, n_a is out of the communication range of n_b. In this case, n_b can successfully receive the RREQ packet from n_a, but it cannot forward RREP packet back to n_a. As a result, the routing path is unable to be built.

To see the full potential of the JarmRout, we plan to explore the followings for future extensions.

1) CONNECTIVITY-BASED MOBILITY MODEL

The basic idea of connectivity-based mobility model is to make use of the local geographical position and mobility information to extend connection time while changing the flying direction gracefully [45]. For example in Fig. 3, every T_{sec} seconds, node n_a first checks whether it is within the communication range of neighbor node n_b. If yes, it checks whether it would still be within the communication range of n_b after T_{sec} based on its current geographical position and mobility information. If n_a determines that it would not leave the communication range of n_b in a time period of T_{sec}, it does not change its flying direction. Otherwise, it changes its flying direction randomly toward the center of communication range of n_b.

2) ROUTE CACHING

Route caching via unconditional overhearing is one of the major features to improve routing performance in reactive routing protocols. Whenever a node forwards or overhears a RREQ, RREP, or data packet, it caches the route learned from the packet to its routing table. If a node forwards or overhears a RERR packet, it removes any route containing the broken link from its routing table. Thus, an intermediate node can send RREP packet back to the source node according to cached route information when it receives a RREQ packet, which can significantly reduce the delay of building the routing path. When the source node receives multiple RREP packets, it investigates the piggybacked routes in the received RREP packets and discards the RREP packet that does not meet the requirement of node-disjoint path.

B. IMMUNITY TO OTHER ATTACKS

We investigate the JarmRout and see whether it is immune to other three well-known attacks: selective forwarding attack, limited transmission power attack, and routing attack.

1) IMMUNITY TO SELECTIVE FORWARDING ATTACK

The selective forwarding attack primarily targets service availability by disrupting network routing protocols or interfering with on-going communications in multihop ad hoc networks, where a malicious node randomly or strategically drops the received packets without forwarding [46]. However, the proposed JarmRout is immune to the selective forwarding attack, because multiple node-disjoint paths are chosen to transmit the same data packets. For example, as shown in Fig. 3, suppose that node n_b is a compromised legitimate node and behaves maliciously to drop the received data packets along forwarding path X. However, since the same data packets are transmitted along another forwarding path Z, the destination node still can receive the data packets. On the other side, if node n_b continuously drops the received data packets, node n_a will consider the link with n_b to be disconnected and sends a RERR packet to the source node. This is because n_a does not overhear implicit acknowledgment or receive explicit acknowledgment from n_b. As a result, path X that contains the broken link will be removed from routing table by source node n_S.

2) IMMUNITY TO LIMITED TRANSMISSION POWER ATTACK

In limited transmission power attack, a malicious node may drop a data packet on purpose by transmitting it with reduced transmission power to exclude a legitimate next-hop node from its communication range [47]. This attack is similar to the selective forwarding attack and the network performance of the JarmRout will not be affected by this attack. For example in Fig. 3, a malicious node n_b receives the data packet from node n_a. Then n_b may forward the data packet by carefully reducing the communication range that does not reach node

n_c but the transmission of data packet can be overheard by n_a. As a result, the data packet that is transmitted along the forwarding path X is successfully dropped by the malicious node. However, the same data packet is also transmitted along path Z, thus, the data packet still can be received by destination node.

3) IMMUNITY TO ROUTING ATTACK

In routing attack, a malicious node falsely claims a fake shortest route to a destination node to attract network traffic on purpose [41], and then launches further attacks, such as selective forwarding attack or blackhole attack. For example, as shown in Fig. 3, suppose that a malicious node n_b replies a fake RREP packet back to source node to falsely claim that it has a route or the shortest route to the destination node. This could lead the malicious node to be involved in the future routing operation and have a chance to selectively or strategically drop or forward any incoming data packets on purpose. However, the proposed JarmRout can protect the network from routing attack. This is because intermediate nodes are not allowed to send the RREP packet back to the source node even when they have route information to the destination node. In other words, the source node will not accept the RREP packet generated by any intermediate node, thus, the network is free of routing attack.

VII. CONCLUSION AND FUTURE WORK

In this paper, we propose a jamming-resilient multipath routing protocol, also called *JarmRout*, so that intentional jamming and disruption, or isolated and localized failures do not interrupt the overall network performance of FANETs. The JarmRout is designed based on a combination of three major schemes, which are link quality scheme, traffic load scheme, and spatial distance scheme, to select maximally spatial node-disjoint multiple paths with high link quality and light traffic load to deliver the data packets from source to destination nodes. We develop a customized discrete event driven simulation framework by using OMNeT++ and evaluate its performance through extensive simulation experiments in terms of packet delivery ratio, packet delivery latency, end-to-end communication outage rate, and energy consumption. The simulation results indicate that the JarmRout can improve packet delivery ratio and packet delivery latency as well as reduce end-to-end communication outage rate. In the presence of malicious jammers, the JarmRout can significantly improve network resiliency without introducing extra communication overhead, which indicates a viable routing approach in FANETs.

As a future work, we plan to extend the JarmRout by including connectivity-based mobility model and route caching technique. Since radio propagation and its channel dynamics cannot easily be captured by simulation models, we plan to develop a small-scale testbed with small and safe quad-copters, e.g, Crazyflie 2.0, and deploy a real outdoor environment to see the full potential of the proposed scheme.

REFERENCES

[1] C. Lin, D. He, N. Kumar, K.-K. R. Choo, A. Vinel, and X. Huang, "Security and Privacy for the Internet of Drones: Challenges and Solutions," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 64–69, Jan. 2018.

[2] *FAA Releases 2016 to 2036 Aerospace Forecast*. Accessed: Nov. 5, 2018. [Online]. Available: https://www.faa.gov/news/updates/?newsId=85227

[3] *Projected Direct Economic Impact from the UAV Industry in the United States*. Accessed: Nov. 5, 2018. [Online]. Available: https://www.statista.com/statistics/536486/projecteddirect-economic-impact-from-the-uav-industry-united-states/

[4] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1123–1152, 2nd Quart., 2016.

[5] M.-S. Kang, D.-W. Kum, J.-S. Bae, Y.-Z. Cho, and A.-N. Le, "Mobility aware hybrid routing protocol for mobile ad hoc network," in *Proc. IEEE ICOIN*, Feb. 2012, pp. 410–414.

[6] M. Kang, D. Kum, J. Bae, Y. Cho, and A. Le, "Mobility aware hybrid routing protocol for mobile ad hoc network," in *Proc. IEEE ICOIN*, 2012, pp. 410–414.

[7] K. Pelechrinis, M. Iliofotou, and S. Krishnamurthy, "Denial of service attacks in wireless networks: The case of jammers," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 2, pp. 245–257, 2nd Quart., 2011.

[8] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*. New York, NY, USA: Springer, 1996, pp. 153–181.

[9] 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*, Dec. 2001, pp. 62–68.

[10] S. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," in *Proc. IEEE ICC*, Jun. 2001, pp. 3201–3205.

[11] A. Varga. (2014). *OMNeT++*. [Online]. Available: http://www.omnetpp.org/

[12] O. Oubbati, A. Lakas, F. Zhou, M. Güneş, and M. Yagoubi, "A survey on position-based routing protocols for Flying Ad hoc Networks (FANETs)," *Veh. Commun.*, vol. 10, pp. 29–56, Nov. 2017.

[13] C.-M. Cheng, P.-H. Hsiao, H. T. Kung, and D. Vlah, "Maximizing throughput of UAV-relaying networks with the load-carry-and-deliver paradigm," in *Proc. IEEE WCNC*, Mar. 2007, pp. 4417–4424.

[14] Q. Yang and S.-J. Yoo, "Optimal UAV path planning: Sensing data acquisition over IoT sensor networks using multi-objective bio-inspired algorithms," *IEEE Access*, vol. 6, pp. 13671–13684, 2018.

[15] A. Alshabtat, L. Dong, J. Li, and F. Yang, "Low latency routing algorithm for unmanned aerial vehicles ad-hoc networks," *Int. J. Elect. Comput. Eng.*, vol. 6, no. 1, pp. 48–54, 2010.

[16] 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.

[17] 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, Mar. 2016.

[18] J. Forsmann, R. E. Hiromoto, and J. Svoboda, "A time-slotted on-demand routing protocol for mobile ad hoc unmanned vehicle systems," *Proc. SPIE*, vol. 6561, p. 65611P, May 2007.

[19] J. Biomo, T. Kunz, and M. St-Hilaire, "Routing in unmanned aerial ad hoc networks: Introducing a route reliability criterion," in *Proc. IFIP WMNC*, May 2014, pp. 1–7.

[20] R. Shirani, "Reactive-greedy-reactive in unmanned aeronautical ad-hoc networks: A combinational routing mechanism," M.S. thesis, Dept. Syst. Comput. Eng., Carleton Univ., Ottawa, ON, USA, 2011.

[21] O. Oubbati, A. Lakas, F. Zhou, M. Güneş, N. Lagraa, and M. Yagoubi, "Intelligent UAV-assisted routing protocol for urban VANETs," *Comput. Commun.*, vol. 107, pp. 93–111, Jul. 2017.

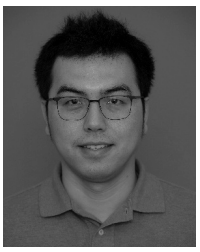
[22] Z. J. Haas, M. R. Pearlman, and P. Samar, "The zone routing protocol (ZRP) for ad-hoc networks," Internet Draft, draft-ietfmanet-zone-zrp-04.txt, Jul. 2002.

[23] C. Zang and S. Zang, "Mobility prediction clustering algorithm for UAV networking," in *Proc. IEEE GLOBECOM Wkshps*, Dec. 2011, pp. 1158–1161.

[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*, Jul. 2013, pp. 7–12.

[25] Z. Zheng, A. Sangaiah, and T. Wang, "Adaptive communication protocols in flying ad hoc network," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 136–142, Jan. 2018.

[26] M. Asadpour, K. Hummel, D. Giustiniano, and S. Draskovic, "Route or carry: Motion-driven packet forwarding in micro aerial vehicle networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 3, pp. 843–856, Mar. 2017.
[27] 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, Jan. 2018.
[28] G. Gankhuyag, A. P. Shrestha, and S.-J. Yoo, "Robust and reliable predictive routing strategy for flying ad-hoc networks," *IEEE Access*, vol. 5, pp. 643–654, 2017.
[29] Y. Zhou, N. Cheng, N. Lu, and X. S. Shen, "Multi-UAV-aided networks: Aerial-ground cooperative vehicular networking architecture," *IEEE Veh. Technol. Mag.*, vol. 10, no. 4, pp. 36–44, Dec. 2015.
[30] T. Long, M. Ozger, O. Cetinkaya, and O. Akan, "Energy neutral Internet of drones," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 22–28, Jan. 2018.
[31] C. Pu, T. Gade, S. Lim, M. Min, and W. Wang, "Lightweight forwarding protocols in energy harvesting wireless sensor networks," in *Proc. IEEE MILCOM*, Oct. 2014, pp. 1053–1059.
[32] A. Vlavianos, L. Law, I. Broustis, S. Krishnamurthy, and M. Faloutsos, "Assessing link quality in IEEE 802.11 wireless networks: Which is the right metric?" in *Proc. IEEE PIMRC*, Sep. 2008, pp. 1–6.
[33] N. Baccour *et al.*, "Radio link quality estimation in wireless sensor networks: A survey," *ACM Trans. Sensor Netw.*, vol. 8, no. 4, p. 34, 2012.
[34] F. Bai, D. Stancil, and H. Krishnan, "Toward understanding characteristics of dedicated short range communications (DSRC) from a perspective of vehicular network engineers," in *Proc. ACM MobiCom*, 2010, pp. 329–340.
[35] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "An empirical study of low-power wireless," *ACM Trans. Sensor Netw.*, vol. 6, no. 2, p. 16, 2010.
[36] V. Csiszár and T. Móri, "A Bienaymé-Chebyshev inequality for scale mixtures of the multivariate normal distribution," *Math. Inequal. Appl.*, vol. 12, no. 4, pp. 839–844, 2009.
[37] A. Moussaoui *et al.*, "A link-state QoS routing protocol based on link stability for mobile ad hoc networks," *J. Netw. Comput. Appl.*, vol. 39, no. 1, pp. 117–125, Mar. 2014.
[38] Y. Li and H. Man, "Three load metrics for routing in ad hoc networks," in *Proc. IEEE VTC*, Sep. 2004, pp. 2764–2768.
[39] D. Saha, S. Toy, S. Bandyopadhyay, T. Ueda, and S. Tanaka, "An adaptive framework for multipath routing via maximally zone-disjoint shortest paths in ad hoc wireless networks with directional antenna," in *Proc. IEEE GLOBECOM*, Dec. 2003, pp. 226–230.
[40] J. J. Gálvez, P. M. Ruiz, and A. F. G. Skarmeta, "Multipath routing with spatial separation in wireless multi-hop networks without location information," *Comput. Netw.*, vol. 55, no. 3, pp. 583–599, 2011.
[41] C. Pu, S. Lim, J. Chae, and B. Jung, "Active detection in mitigating routing misbehavior for MANETs," *Wireless Netw.*, pp. 1–15, Nov. 2017, doi: 10.1007/s11276-017-1621-z.
[42] C. Pu and S. Lim, "A light-weight countermeasure to forwarding misbehavior in wireless sensor networks: Design, analysis, and evaluation," *IEEE Syst. J.*, vol. 12, no. 1, pp. 834–842, Mar. 2018.
[43] H. Pucha, S. Das, and Y. Hu, "The performance impact of traffic patterns on routing protocols in mobile ad hoc networks," *Comput. Netw.*, vol. 51, no. 12, pp. 3595–3616, Mar. 2007.
[44] C. Pu and T. Song, "Hatchetman attack: A denial of service attack against routing in low power and lossy networks," in *Proc. IEEE CSCloud*, Jun. 2018, pp. 12–17.
[45] E. Yanmaz, "Connectivity versus area coverage in unmanned aerial vehicle networks," in *Proc. IEEE ICC*, Jun. 2012, pp. 719–723.
[46] C. Pu and S. Lim, "Spy vs. spy: Camouflage-based active detection in energy harvesting motivated networks," in *Proc. IEEE MILCOM*, Oct. 2015, pp. 903–908.
[47] C. Pu, S. Lim, B. Jung, and J. Chae, "EYES: Mitigating forwarding misbehavior in energy harvesting motivated networks," *Comput. Commun.*, vol. 124, pp. 17–30, Jun. 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. He is currently an Assistant Professor with the Weisberg Division of Computer Science, Marshall University. His research interests are in the areas of cybersecurity, wireless networks and mobile computing, energy-harvesting motivated networks, mobile ad hoc networks, low-power and lossy networks, flying ad hoc networks, and evacuation-assisting vehicular networks.

Dr. Pu served as a technical program committee member in many conferences. He is a member of the Computer Science Workgroup for the West Virginia Department of Education to increase and strengthen computer science education in West Virginia. He received the 2015 Helen Devitt Jones Excellence in Graduate Teaching Award at Texas Tech University, the NASA WVSGC Research Initiation Grant, and the John Marshall Summer Scholar Award in 2018. He was the Winner of 2017 Design for Delight (D4D) Innovation Challenge Competition as a Faculty Coach (Marshall University and Intuit Inc.). He was nominated by the Department of Education (WV) to participate in the Educational Testing Services Standard Setting Study in EST. He was a reviewer for several IEEE journals.