Link-Quality and Traffic-Load Aware Routing for UAV Ad Hoc Networks

Cong Pu
Weisberg Division of Computer Science
Marshall University
Huntington, WV 25755
puc@marshall.edu

***Abstract*—With increasingly popular multi-sized unmanned aerial vehicles (UAVs), also referred to as drones, UAV Ad Hoc Networks (UANETs) play an essential role in the realization of coordinating the access of drones to controlled airspace, and providing navigation services between locations in the context of Internet-of-Drones (IoD). Because of the versatility, flexibility, easy installation and relatively small operating expenses of drones, UANETs are more efficient in completing complex tasks in harsh environments, e.g., search and destroy operations, border surveillance, disaster monitoring, etc. However, due to the high mobility, drastically changing network topology, and intermittently connected communication links, existing routing protocols and communication algorithms in Mobile Ad Hoc Networks and Vehicular Ad Hoc Networks cannot be directly applied in UANETs. In this paper, we propose a link-quality and traffic-load aware optimized link state routing protocol, also called *LTA-OLSR*, to provide efficient and reliable communication and data transmission in UANETs. A 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. A traffic load scheme is also proposed to assure a light load path by taking account of MAC layer channel contention information and the number of packets stored in the buffer. We evaluate the proposed schemes through extensive simulation experiments using OMNeT++ and compare their performance with the original OLSR and DSR protocols. The simulation results indicate that the proposed routing protocol can be a viable approach in UAV Ad Hoc Networks.**

***Index Terms*—Unmanned Aerial Vehicles, Ad Hoc Networks, Link Quality, Traffic Load, Routing Protocol, Internet-of-Drones**

I. Introduction

A rapidly growing number of unmanned aerial vehicles (UAVs), often referred to as drones, is leading the emergence of Internet-of-Drones (IoD) and its applications, 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 between locations [1]. It has been predicted that the hobbyist drone fleet will reach 3.55 million, and the number of commercial drones will grow tenfold to 442,000 by 2021 [2]. Economic growth of drone-based services and applications, including military scouting, goods freight in cities, road traffic management, aerial photography, urban safety, and so on, is also said to be considerable for businesses. It is probable that a $100 billion market opportunity helped by growing demand for drones from the commercial and civil government sectors will be available in the United States between 2016 and 2020 [3]. With the continuous miniaturization of sensors and processors, the prevalence of wireless connectivity and cloud computing, we envision a future in which seamlessly blended drones in the realm of Internet-of-Drones will lead to the further improvement of our lives.

In order to efficiently complete complex tasks, a swarm of drones can self-organize into a network, called UAV Ad Hoc Network (UANET), where all drones faithfully and collaboratively route data packets to a destination. Compared to the single drone paradigm used in those delay-tolerant applications [4], the advantages of UANETs will be faster multitasking capability, longer network lifetime, and higher scalability. However, many challenging problems still need further research in UANETs, and one is how to efficiently and cooperatively communicate and transmit data packets between drones. Over the last few years, many researchers have explicitly studied the communication algorithms and routing protocols in Mobile Ad Hoc Networks (MANETs) and Vehicular Ad Hoc Networks (VANETs), however, they cannot be directly applied in UANETs because of the high mobility of drones, drastically changing network topology, and intermittently connected communication links [5]. For example, network performance (e.g., packet delivery ratio) can significantly degrade in MANET with high mobility because of frequent link errors [6]. In sparse networks, network partitions may last for significantly long periods and lead to buffer contention because messages cannot be removed from buffers and new messages might be generated, resulting in longer transmission delay [7].

The Optimized Link State Routing (OLSR) [8] is a well-known proactive routing protocol based on the traditional link-state algorithm in MANETs, where each node maintains topology information about the network by periodically exchanging link-state messages. In order to reduce communication overhead in the network, each node selects a set of neighbor nodes, called multipoint relays (MPRs) of that node, to retransmit its broadcasted packets. Each node determines an optimal route with the least number of hops to every known destination using its topology information, and stores the route information in a routing table. Therefore, routes to every destination are

978-1-5386-9502-9/18/$31.00 ©2018 IEEE
DOI 10.1109/CIC.2018.00-38

immediately available when data transmission is needed. As it is well known, the hop count metrics does not take into account link quality and traffic load along the forwarding path. In [9], it has been shown that a route that minimizes the hop count does not necessarily maximize the throughput of a flow.

In this paper, we propose a novel routing protocol to provide efficient and reliable communication and data transmission in UANETs. Our major contribution is summarized in the following:

- First, we propose link quality and traffic load schemes to find the optimal path between source and destination in UANETs. 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 assures a light load path by taking account of MAC layer channel contention information and the number of packets stored in the buffer.
- Second, we integrate the link quality and traffic load schemes with Optimized Link State Routing (OLSR) and propose a link-quality and traffic-load aware optimized link state routing protocol, also called *LTA-OLSR*, to provide efficient and reliable communication and data transmission in UANETs.
- Third, we revisit existing routing protocols, Optimized Link State Routing (OLSR) and Dynamic Source Routing (DSR) [10], and modify them to work in UANETs for performance comparison.

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 and packet delivery latency. The simulation results indicate that the proposed routing protocol is a viable approach for efficient and reliable communication and data transmission in UANETs.

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

II. Related Work

Significant research efforts have been made to investigate routing protocols and communication algorithms in MANETs and VANETs. In this section, we categorize and analyze the existing routing schemes and techniques in terms of single-hop, multi-hop, and other approaches.

In the single-hop routing, drones are used as packet carriers, which transfer packets when flying from source to destination. In [4], 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. Since 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. However, the major disadvantage of this protocol is that drones do not use GPS information and trajectory calculation during route discovery and data forwarding. This can decrease the performance of the protocol in the case when the destination is not a fixed node.Without collaborative packet forwarding between multiple drones, the route planning problem is of great importance to drones in the single-hop routing. The [12] proposes a hybrid differential evolution with quantum-behaved particle swarm optimization to generate a safe and flyable path for drone in the presence of different threat environments on the sea, where the terrain pretreatment is performed and all islands are designated as terrain threats. To reduce the probability of detection by radar, the drone is required to execute a sea-skimming flight with a minimum constant clearance between drone and the sea level.

The single-hop routing is light-weight and mainly designed for fixed topology, and the disadvantages of this approach include poor fault tolerance and is not suitable for dynamic environments. Thus, multi-hop routing becomes the subject of focus, where packets are forwarded hop by hop. In [13], a directional optimized link state routing protocol (DOLSR) is proposed to minimize the number of multipoint relays in UANET, where each drone is equipped with directional and omni-directional antennas. In the DOLSR, each drone tests the distance between itself and destination. If the distance is larger than a threshold value, the drone will apply the DOLSR mechanism. If the distance is smaller than a threshold value, the drone will apply the original OLSR, where the omni-directional antenna is used. The [14] proposes a speed-aware predictive-optimized link state routing protocol by exploiting GPS information to aid routing operations, where the relative speed between two drones can be obtained based on GPS information, and is taken into account as a factor in the calculation of the expected transmission count metrics. However, there are still many issues worth studying such as the sudden disconnection and how to deal with this situation, and the recovery technique to apply in order to resume the normal function of the network. As future work, a maintenance mechanism needs to be conceived to deal with dis-connections when they occur. In [15], a mobility and load aware routing protocol is proposed for UANETs, where relative speed and position between neighbor 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 taken into account to discover more stable routes without congestion. The [16] proposes adaptive hybrid communication protocols including a novel position-prediction-based directional MAC protocol (PPMAC) and a self-learning routing protocol based on reinforcement learning (RLSRP). The performance results show that the proposed

PPMAC overcomes the directional deafness problem with directional antennas, and RLSRP provides an automatically evolving and more effective routing scheme. The [17] proposes a routing protocol using a directional antenna that has a longer transmission range and higher spatial reuse of the network in the MAC layer to overcome the problems under an omnidirectional antenna as well as the well-known deafness problem of directional MAC.

A time-slotted on-demand routing protocol is proposed in [18] for UANETs. It is basically a time-slotted version of Ad-hoc On-demand Distance Vector Routing (AODV). While AODV sends its control packets on random-access mode, time-slotted on-demand protocol uses dedicated time slots for each drone and allows collision-free communication between neighbors. In addition, in order to manage topology changes, the time-slot window is dynamically adjusted in case of link-failure detection. In [19], a UAV-assisted VANET routing protocol (UVAR) is proposed to support ad hoc routing between drones and VANET as well as between drones themselves. The UVAR 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. Secondly, 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 packets through vehicles. In [20], 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 [21] develops a cluster formation algorithm for UANETs to solve the problem of frequent cluster updates due to high-speed drones with the prediction of the network topology updates. It predicts the mobility structures of drones with the help of the dictionary trie structure prediction algorithm and link expiration time mobility mode. The [22] proposes a recovery strategy to salvage packets in void node situations, when a packet arrives at a node that has no neighbor node closer to the destination than it is. The proposed recovery strategy consists of three schemes: retrying greedy geographic forwarding, forwarding packet to the furthest neighbor node, and forwarding packet to best-moving node. A survey of routing protocols in UANETs is provided in [23].

In summary, various routing protocols and communication mechanisms have been well studied in UAV Ad Hoc Networks (UANETs). From the comparison, it is found that each protocol has its own definite strengths and weaknesses, and suitability for specific situation. In addition, we also find that most routing protocols do not take load balance into consideration. Some route matrices are proposed, such as the shortest path, the freshest path, the minimum-cost path, or the path with the best link quality. However, little attention has been shown to focus on the factors of link quality and traffic load together to provide efficient and reliable communication and data transmission in UANETs.

III. The Proposed Link-Quality and Traffic-Load Aware Optimized Link State Routing Protocol

In this section, we first introduce the system model, the limitations of OLSR and motivation, and present link quality and traffic load schemes. Then, we propose the link-quality and traffic-load aware optimized link state routing protocol, also referred to as *LTA-OLSR*, to provide efficient and reliable communication and data transmission in UAV Ad Hoc Networks (UANETs).

A. System Model

In this paper, we consider a set of drones (later nodes) that freely moves in a UANET, where each node is identified by its node address [24]. Each node is equipped with a GPS and digital map to obtain its current geographical position. We also assume that nodes have no energy restrictions since they are equipped with rechargeable batteries which can be recharged from environmental energy resources (e.g., solar energy, etc.) [19]. Most of drone-based services and applications that use drones like small quad-copters do not fly at high altitudes, therefore, we assume that drones have a low and constant altitude during the flight. In addition, IEEE 802.11p wireless interface with a large transmission range (i.e., 300 meters) are assumed to be used by drones.

B. Limitations of OLSR and Motivation

OLSR is currently one of the most popular proactive routing algorithms for ad hoc networks. The original OLSR design does not consider the quality of the wireless link, and the route selection is based on the hop count metric, which does not take into account link quality and traffic load along the forwarding path. Minimizing the hop count maximizes the distance traveled by each hop, which is likely to minimize signal strength and maximize the packet loss ratio. Even if the best route is a minimum hop count route, in a network there may be many routes of the same minimum length, with widely varying qualities. The arbitrary choice made by most minimum hop count metric is not likely to select the best route. The [9] has shown that minimizing the hop count does not necessary maximize the throughput of a flow. Additionally, in shortest path routing, nodes on the shortest path will get more heavily loaded than others since they are frequently chosen as the routing path. Having a heavy load can exhaust a nodes resource such as bandwidth, processing power, battery energy, and memory storage. Finally, if one of the heavily loaded nodes is congested, it can lead to packet loss and buffer overflow, resulting in longer end-to-end delay, degradation in throughput, and loss of transport connections. In light of these, in this paper, we investigate a link-quality and traffic-load aware optimized link state routing protocol, also referred to as LTA-OLSR, in UAV Ad Hoc Networks. The LTA-OLSR consists of two schemes: link quality scheme and traffic load scheme. A 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. A traffic load scheme is also proposed to assure a light load path by taking account of MAC layer channel contention information and the number of packets stored in the buffer.

C. 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) [25]. Compared to other three metrics, RSSI provides a quick and accurate estimate of whether a link is of very good quality [26]. The [27] has proved that higher RSSI values result in better PDRs, and as long as radio transceiver (e.g., DSRC compatible radio built upon the Atheros AR5000 chipset) maintains RSSI value above -55 dBm, the PDR 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 [28]. 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 [29], [30] 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:

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

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

$$var(X) = E(X^2) - E(X)^2. \quad (2)$$

and

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

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

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

Most radio transceivers contain an RSSI register, which provides the signal strength of the received packet [26]. 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

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

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 I, R_x, R_{x+1}, R_{x+2}, and R_{x+3} are the corresponding RSSI values of the most recently received messages 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 the 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 message 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 I: 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}

D. 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 [31]. 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 recently 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 it usually have 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 II: 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 II, 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.

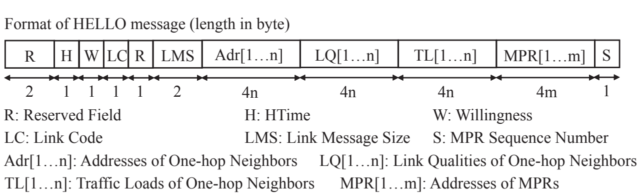


Fig. 1: The format of HELLO message. Here, the length is shown in byte.

E. The Proposed LTA-OLSR Protocol

First, each node periodically broadcasts HELLO messages piggybacked with its own address and traffic load to neighbor nodes. HELLO messages are transmitted in the broadcast mode, where they are received by all one-hop neighbor nodes, but not relayed or broadcasted further. Through periodically exchanged HELLO messages, each node can obtain its one-hop neighbor nodes, the link qualities associated with one-hop neighbor nodes, and the traffic loads of one-hop neighbor nodes, which are declared in the subsequent HELLO messages broadcasted by the node. Here, the format of HELLO message is shown in Fig. 1. These periodically exchanged HELLO messages permit each node to learn the knowledge of its two-hop neighbor nodes, and then build its neighbor table (NT), which contains one-hop neighbor node address (nid), the link quality associated with one-hop neighbor node (l_{qt}), the traffic load of one-hop neighbor node (t_{ld}), multipoint relay flag ($mpr^?$), two-hop neighbor nodes (nid^+), and entry expiration time (t_{exp}). Here, the entry expiration time indicates when the entry is no longer valid and hence removed.

Second, to minimize the flooding of broadcast packets in the network by reducing duplicate retransmissions in the same region, each node selects a set of one-hop neighbor nodes as multipoint relays (MPRs) based on the information of neighbor table to rebroadcast its packets [32]. Each node selects its MPR set among its one-hop neighbor nodes in such a manner that the MPR set covers (in terms of communication range) all its two-hop neighbor nodes. The one-hop neighbor nodes of a node which are not in the MPR set just read and process

---

Notations:

- $NT[nid, l_{qt}, t_{ld}, mpr^?, nid^+, t_{exp}]$, nid, l_{qt}, t_{ld}, $mpr^?$, nid^+, t_{exp}, $MST[mprsid, MPRseq]$, $mprsid$, $MPRseq$, MST^{seq}: Defined before.
- $HELLO[ADR, LQ, TL, MPR, MPRseq]$: A HELLO message with addresses of one-hop neighbors, ADR, link qualities of one-hop neighbors, LQ, traffic loads of one-hop neighbors, TL, addresses of multipoint relays, MPR, and multipoint relays sequence number, $MPRseq$.

◇ When a node, n_i, overhears a HELLO message from a node, n_j:

/∗ Update neighbor table NT ∗/
if $n_j \notin NT_i[nid]$
  Add new entry n_j along with $LQ_{i,j}$, TL_j, t_{exp} into NT_i;
  for $n_k \in HELLO[ADR]$
    $NT_i[n_j].nid^+ = NT_i[n_j].nid^+ \cup n_k$;
else
  Add $LQ_{i,j}$, TL_j into $NT_i[n_j].l_{qt}$ and $NT_i[n_j].t_{ld}$;
  Reset $NT_i[n_j].t_{exp}$; Clear up $NT_i[n_j].nid^+$;
  for $n_k \in HELLO[ADR]$
    $NT_i[n_j].nid^+ = NT_i[n_j].nid^+ \cup n_k$;
/∗ Update MRP Selector table MST ∗/
if $n_j \notin MST_i[mprsid]$
  $MST_i[mprsid] = MST_i[mprsid] \cup n_j$;
  $MST_i[n_j].MPRseq = HELLO[MPRseq]$;
  MST_i^{seq} += 1;
else
  if $MST_i[MPRseq] < HELLO[MPRseq]$;
    $MST_i[n_j].MPRseq = HELLO[MPRseq]$;

Fig. 2: The pseudo code of overhearing a HELLO message.

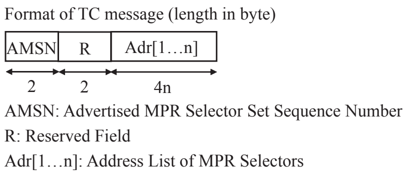


Fig. 3: The format of TC message. Here, the length is shown in byte.

Notations:

• $TT[mprsid, lid, MPRseq, t_{hold}]$, lid, t_{hold}: Defined before.

• $TC[mprsid, MPRseq]$: A Topology Control message with multipoint relay selector id, $mprsid$, and multipoint relay sequence number, $MPRseq$.

◇ When a node, n_i, overhears a TC message from a node, n_j:

```
for n_k ∈ TT_i[lid]
  if n_k == n_j and TT_i[n_k].MPRseq > TC[MPRseq]
    Discard TC message;
  if n_k == n_j and TT_i[n_k].MPRseq < TC[MPRseq]
    Remove TT_i[n_k] entry;
for n_k ∈ TC[mprsid]
  for n_x ∈ TT_i[mprsid]
    if n_k == n_x and TT_i[n_x].lid == n_j
      Reset TT_i[n_x].t_hold;
    else
      Add TC[n_k, MPRseq] entry into TT_i;
```

Fig. 4: The pseudo code of overhearing a TC message.

the packet but do not rebroadcast the packet received from this node. These selected multipoint relays are also indicated in the HELLO messages with the piggybacked multipoint relays (MPR) along with a sequence number ($MPRseq$), as shown in Fig. 1. Here, the MPR sequence number ($MPRseq$) specifies the most recent multipoint relays. When there is a change in one-hop or two-hop neighbor node set, the multipoint relays will be recalculated. On the reception of HELLO messages, each node can construct its MPR Selector table (MST) which consists of two components: the addresses of its one-hop neighbor nodes which have selected it as a multipoint relay ($mprsid$) and the sequence number which specifies the most recent MPR set of that neighbor node ($MPRseq$). A sequence number (MST^{seq}) is also associated with MPR Selector table which specifies that the MPR Selector table is most recently modified with that sequence number. A node updates its MPR Selector table according to the information it receives in the HELLO messages, and increases the sequence number MST^{seq} on each modification. Major operations of overhearing a Hello message is summarized in Fig. 2.

Third, in order to build the intra-forwarding database needed for routing packets, each node broadcasts specific control messages called Topology Control (TC) messages. TC messages are forwarded like usual broadcast messages in the entire network through multipoint relays. A TC message is sent periodically by each node in the network to declare its MPR Selector set with the associated sequence number. Here, the format of TC message is shown in Fig. 3. The interval between the transmission of two consecutive TC messages depends upon whether the MPR Selector set is changed or not, since the last TC message transmitted. When a change occurs in the MPR Selector set, the next TC message will be sent t^{λ} earlier than the scheduled time t^{out}. If the time (t^{out} - t^{λ}) has already elapsed, the next TC message will be transmitted immediately. All subsequent TC messages are sent with the normal default interval (t^{out}) until the MPR Selector set is changed again. Through periodically exchanged TC messages, each node can build and maintain a topology table (TT), in which it records the node address of MPR Selector ($mprsid$), the address of the last-hop node to the MPR Selector (lid), a MPR Selector sequence number ($MPRseq$), and entry holding time (t_{hold}). The topology table will be updated accordingly upon receipt of a TC message. When the originator address of the TC message equals to the last-hop address of certain entry in the topology table, if the sequence number in the received message is less than the MPR Selector sequence number in that entry, then the TC message is silently discarded. If the sequence number in the received message is larger than the MPR Selector sequence number in that entry, then this entry is removed from topology table. If there exist certain entry in the topology table whose destination address corresponds to the MPR Selector address and the last-hop address of that entry corresponds to the originator address of the TC message, the holding time of that entry is refreshed. Otherwise, a new topology entry is recorded in the topology table. Major operations of overhearing a TC message is summarized in Fig. 4.

Fourth, on the basis of topology table and neighbor table, each node can maintain a routing table which allows it to send the packets to other destination in the network, where the route is built by tracking the connected pairs in a descending order. For example, to find a path from a given source node (e.g., n_s) to a destination node (n_d), one has to find a connected pair (n_k, n_d), then a connected pair (n_j, n_k), and so forth until one finds a node n_i as one of neighbor node of n_s. In the original OLSR, the shortest path algorithm is used to find the route between source and destination with the minimum number of hops. In the TLA-OLSR, the link quality and traffic load are introduced in the selection of paths instead of the number of hops, and all candidates paths are compared with the value of Route Credit (R_{crt}). The R_{crt} is the comprehensive judgment factor, which can be represented as

$$R_{crt} = \varpi \cdot e^{\sum_{i=1}^{N+1} LQ_i} + (1 - \varpi) \cdot e^{\sum_{j=1}^{N} TL_j}, \tag{9}$$

where N is the number of intermediate nodes along the forwarding path between source and destination, $\sum_{i=1}^{N+1} LQ_i$ is the sum of link qualities of all links along the forwarding path, and $\sum_{j=1}^{N} TL_j$ is the sum of traffic loads of all intermediate nodes along the forwarding path. Based on the above calculation, the route with the minimum R_{crt} will be chosen as the routing path to forward the data packet. Major operations

Notations:

• $RT_i[des, next, dist]$: A routing table with destination address, des, next-hop address to destination, $next$, and distance to destination in terms of number of hops, $dist$.

• $OT[src, dist]$: A set of candidate routes from source *src* to destination *dist*.

```
◇ Node n_i calculates routing table RT_i:
  Clean up RT_i;
  for n_j ∈ NT_i[nid]
    if hop == 1
      RT_i.[des] = n_j; RT_i.[next] = n_j; hop++;
    else
      for n_x ∈ TT_i[mprsid]
        flag = true;
        for n_y ∈ RT_i[des]
          if n_x == n_y
            flag = false;
        if flag == true
          for n_z ∈ RT_i.[des]
            if TT_i[n_x].lid == n_z and RT_i.[n_z].dist == hop
              RT_i.[n_z].des = n_x;
              for n_t ∈ RT_i.[des]
                if n_t == TT_i[n_z].lid
                  RT_i.[n_z].next = RT_i.[n_t].next;
              hop++;
◇ When a source node, n_s, has data packet for a destination node, n_d:
  Apply Eq. 9 to all routes to destination node n_d;
  for r_i ∈ OT
    Calculate R^i_crt;
  Choose the route r_i with minimum R^i_crt;
  Send data packet to n_d through r_i;
```

Fig. 5: The pseudo code of calculation of routing table and selection of optimal route.

of calculation of routing table and selection of optimal route are summarized in Fig. 5.

IV. Performance Evaluation

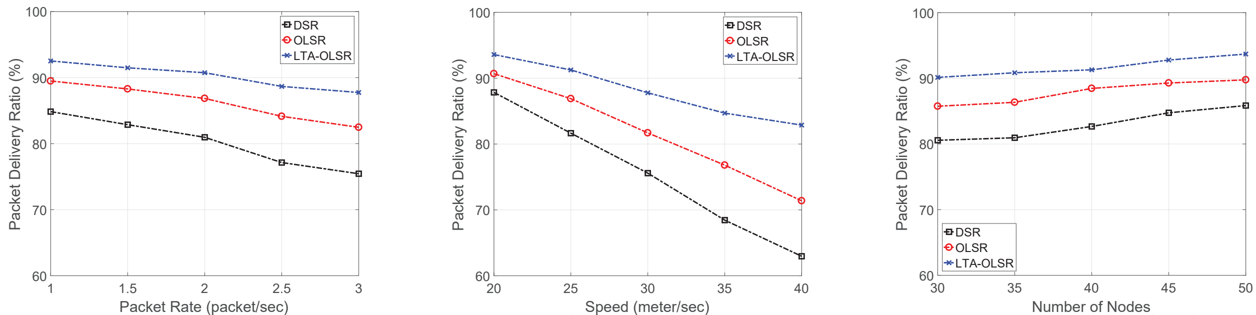
A. Simulation Testbed

We conduct extensive simulation experiments using the OMNeT++ [11] for performance evaluation and analysis. A 2000×2000 (m^2) square network area is considered, where 30-50 nodes are uniformly distributed. The communication range of each node is 300 (m) and the two-way ground propagation channel is assumed with a data rate of 2 Mbps. The random waypoint mobility model [33] is deployed in the network, where each node travels toward a randomly selected destination in the network with a constant speed of 20-40 meter/sec and a zero pause time. Nodes are equipped with IEEE 802.11p radio transceiver. The source nodes generate 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 2000 seconds, and each simulation scenario is repeated 10 times with different randomly generated seeds to obtain steady state performance metrics. In this paper, we measure the performance in terms of packet delivery ratio and packet delivery latency by changing key simulation parameters, including packet rate, speed, and number of nodes in the network. For performance comparison, we compare the proposed TLA-OLSR with the original OLSR and DSR, which is the classic proactive routing protocol and reactive routing protocol, respectively. In the original OLSR, the optimal route is selected with the minimum number of hops between source and destination.

B. Simulation Results

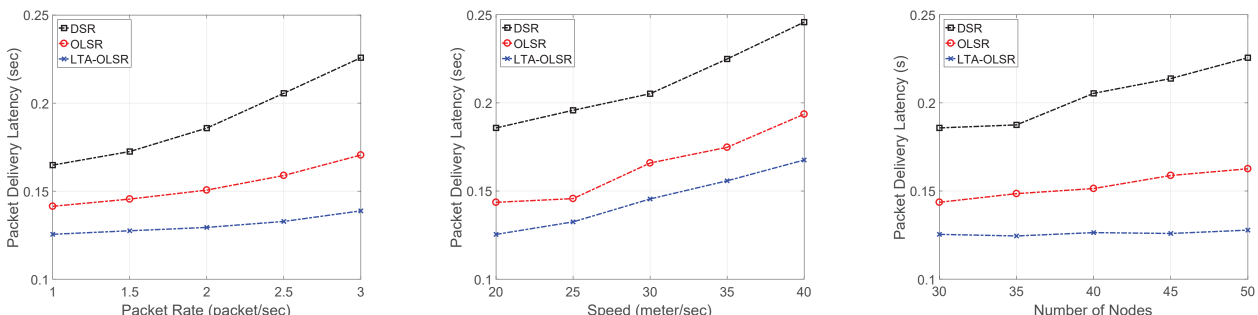
Packet Delivery Ratio: Fig. 6 shows the packet delivery ratio (PDR) with varying packet rate, moving speed, and number of nodes in the network. As shown in Subfig. 6(a), the PDRs of three different schemes decrease as the packet rate increases. With larger packet rate, more packets will be generated and forwarded by source node in the network. Thus, the shared wireless medium could experience more interference and contention, and more data packets could collide with each other and get lost. Additionally, each node moves with a high speed, the route between source and destination could become unavailable suddenly. Thus, ongoing packets cannot be forwarded and delivered to destination, a lower PDR is obtained. The DSR shows the lowest PDR because the route has to be built first before sending data packet. When the source node receives the route information (route reply packet) and sends data packet, the link along the forwarding path could be broken due to node mobility, and the data packet has to be dropped. The proposed LTA-OLSR shows the best performance as packet rate increases. This is because each node employs periodic exchange of messages to maintain topology information of the network, which provides optimal routes immediately when needed, the source node can send the data packet directly. On the other side, the LTA-OLSR considers the link quality and traffic load of forwarding path, and selects the stable and reliable route which provide high quality links and experience light traffic load to forward the data packet. As a result, more data packets can be delivered to destination, and a higher PDR is observed. The original OLSR provides higher and lower PDR compared to that of DSR and LTA-OLSR because each node proactively maintains the routes to all destinations, however, the route is chosen based on the number of hops, which is not always best metrics.

In Subfig. 6(b), as the node moving speed increases, the PDRs for three schemes significantly decrease as expected. When the node speed increases, the communication link between each two nodes becomes unstable, more packets are dropped or get lost during transmission due to unavailable or unstable routes. The PDR of DSR decreases significantly as speed increases, this is because the source node has to find the route to the destination node by issuing the route request packet before sending the data packet. When the source node receives the route reply packet from the destination node, and



(a) PDR vs. Packet Rate (b) PDR vs. Speed (c) PDR vs. Number of Nodes

Fig. 6: The performance of packet delivery ratio (PDR) against packet rate, speed, and number of nodes.



(a) Latency vs. Packet Rate (b) Latency vs. Speed (c) Latency vs. Number of Nodes

Fig. 7: The performance of packet delivery latency against packet rate, speed, and number of nodes.

sends out the data packet, the certain forwarding link along the route could be broken due to node mobility, and data packet has to be dropped. Our scheme shows the highest PDR compared to that of DSR and original OLSR. Since each node in the TLA-OLSR frequently exchanges HELLO and topology control messages to maintain up-to-date network topology, the data packet can be sent out immediately through the selected optimal route. Finally, more packets can be delivered to the destination node.

In Subfig. 6(c), by increasing the number of nodes in the network, the PDR of three routing protocols slightly increases. This is due to the fact that each node could have more number of neighbor nodes as node density increases, and more routes between source and destination will be available to forward the data packets. Thus, more data packets can be delivered to destination through stable route. The highest PDR is observed by the proposed scheme, because the TLA-OLSR evaluates each potential route to the destination by considering the link qualities of all links and the traffic load of all intermediate nodes along the forwarding path, the optimal route can be selected to forward the data packets, more data packets can be delivered.

Packet Delivery Latency: Fig. 7 shows the packet delivery latency with varying packet rates, moving speed, and number of nodes in the network. As shown in Subfig. 7(a), the packet delivery latency of DSR, OLSR, and TLA-OLSR increase as the packet rate increases. Due to the high mobility, the intermediate links along the forwarding path could be frequently broken, the data packet cannot be forwarded further and will be dropped. The source node has to retransmit the lost data packet, thus, the packet delivery latency will be increased. The DSR shows the highest packet delivery latency, which increases sharply as the packet rate increases. This is because the data packets frequently experience the broken link during the transmission and will be dropped by the intermediate node, the source node has to retransmit the lost data packets by finding another path, resulting in the increment of packet delivery latency. The proposed scheme shows the lowest packet delivery latency, because each node actively maintains the routes to all destination nodes, and the data packet can be sent immediately when needed. However, the selected route between source and destination could become unavailable during packet transmission because of node mobility, the source node has to retransmit the lost data packet, which makes packet delivery latency increased.

In Subfig. 7(b), the packet delivery latency significantly increases when the node moving speed increases. In DSR, the route has to be built first by issuing the route request packet before sending the data packet. However, the pre-found route could become unavailable during the packet transmission, as a result, the data packet will be dropped. The source node will need to find another route to send the data packet, thus, the packet delivery latency significantly increases. The TLA-OLSR actively maintains the routing table and frequently updates it whenever the node detects the changes in the neighborhood or when a route to any destination is expired, thus, each node can quickly send the data packet to the destination node. In Subfig. 7(c), as the number of nodes in the network increases, the packet delivery latency of DSR and original OLSR increases. This is because the network becomes more dense and each node has more neighbor nodes, more potential routes with the large number of hops could exist to deliver the data packets. The packet delivery latency of TLA-OLSR is not sensitive to the number of nodes, because the TLA-OLSR considers the link quality and traffic load to select route, rather than the number of hops.

V. Conclusion

In this paper, we propose a link-quality and traffic-load aware optimized link state routing protocol, called *LTA-OLSR*, to provide efficient and reliable communication and data transmission in UANETs. A 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. A traffic load scheme is also proposed to assure a light load path by taking account of MAC layer channel contention information and the number of packets stored in the buffer. 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 and packet delivery latency. The simulation results indicate that the proposed routing protocol is a viable approach for efficient and reliable communication and data transmission in UANETs. 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.

Acknowledgment

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

References

[1] M. Gharibi, R. Boutaba, and S. Waslander, "Internet of Drones," *IEEE Access*, vol. 4, pp. 1148–1162, 2016.

[2] *Almost 800,000 U.S. Drone Owners Have Registered to Fly in 15 Months*, https://www.recode.net/2017/3/27/15077998/us-drone-owners-registered-fly-15-months.

[3] *Drones: Reporting for Work*, http://www.goldmansachs.com/our-thinking/technology-driving-innovation/drones/.

[4] 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.

[5] 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, 2016.

[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] W. Zhao, M. Ammar, and E. Zegura, "Mobility Aware Hybrid Routing Protocol for Mobile Ad hoc Network," in *Proc. ACM MobiHoc*, 2004, pp. 187–198.

[8] 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.

[9] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing," *Wireless Networks*, vol. 11, no. 4, pp. 419–434, 2005.

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

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

[12] Y. Fu, M. Ding, C. Zhou, and H. Hu, "Route Planning for Unmanned Aerial Vehicle (UAV) on the Sea Using Hybrid Differential Evolution and Quantum-Behaved Particle Swarm Optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 6, pp. 1451–1465, 2013.

[13] A. Alshabtat and L. Dong, "Low Latency Routing Algorithm for Unmanned Aerial Vehicles Ad-Hoc Networks," *World Acad. Sci. Eng. & Technol*, vol. 80, pp. 705–711, 2011.

[14] S. Rosati, K. Kruzelecki, L. Traynard, and B. Rimoldi, "Speed-Aware Routing for UAV Ad-Hoc Networks," in *Proc. IEEE GLOBECOM Wkshps*, 2013, pp. 1367–1373.

[15] 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.

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

[17] 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.

[18] 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.

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

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

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

[22] J. Biomo, T. Kunz, and M. St-Hilaire, "Routing in Unmanned Aerial Ad hoc Networks: A Recovery Strategy for Greedy Geographic Forwarding Failure," in *Proc. IEEE WCNC*, 2014, pp. 2236–2241.

[23] J. Jiang and G. Han, "Routing Protocols for Unmanned Aerial Vehicles," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 58–63, 2018.

[24] 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.

[25] 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*, 2008, pp. 1–6.

[26] N. Baccour, A. Koubâa, L. Mottola, M. Zúñiga, H. Youssef, C. Boano, and M. Alves, "Radio Link Quality Estimation in Wireless Sensor Networks: A Survey," *ACM Trans. Sensor Networks*, vol. 8, no. 4, p. 34, 2012.

[27] 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.

[28] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "An Empirical Study of Low-Power Wireless," *ACM Trans. Sensor Networks*, vol. 6, no. 2, p. 16, 2010.

[29] V. Csiszár and T. Móri, "A Bienaym-Chebyshev Inequality for Scale Mixtures of the Multivariate Normal Distribution," *Math Inequal Appl*, vol. 12, no. 4, pp. 839–844, 2009.

[30] A. Moussaoui, F. Semchedine, and A. Boukerram, "A link-state QoS routing protocol based on link stability for Mobile Ad hoc Networks," *Elsevier Net. and Comp. Appl*, vol. 39, pp. 117–125, 2014.

[31] Y. Li and H. Man, "Three Load Metrics for Routing in Ad Hoc Networks," in *Proc. IEEE VTC*, 2004, pp. 2764–2768.

[32] O. Liang, Y. Sekercioglu, and N. Mani, "A Survey of Multipoint Relay based Broadcast Schemes in Wireless Ad Hoc Networks," *IEEE Commun. Surveys Tuts.*, vol. 8, no. 4, pp. 30–46, 2006.

[33] H. Pucha, S. Das, and Y. Hu, "The Performance Impact of Traffic Patterns on Routing Protocols in Mobile Ad Hoc Networks," *Elsevier Computer Networks*, vol. 51, no. 12, pp. 3595–3616, 2007.